



参考：构件块、表和过程

---

## **SAP Sybase IQ 16.0 SP03**

文档 ID: DC01134-01-1603-01

最后修订日期: 2013 年 11 月

© 2013 SAP 股份公司或其关联公司版权所有, 保留所有权利。

未经 SAP 股份公司明确许可, 不得以任何形式或为任何目的复制或传播本文的任何内容。本文包含的信息如有更改, 恕不另行事先通知。

由 SAP 股份公司及其分销商营销的部分软件产品包含其它软件供应商的专有软件组件。各国的产品规格可能不同。

上述资料由 SAP 股份公司及其关联公司 (统称“SAP 集团”) 提供, 仅供参考, 不构成任何形式的陈述或保证, 其中如若存在任何错误或疏漏, SAP 集团概不负责。与 SAP 集团产品和服务相关的保证仅限于该等产品和服务随附的保证声明 (若有) 中明确提出之保证。本文中的任何信息均不构成额外保证。

SAP 和本文提及的其它 SAP 产品和服务及其各自标识均为 SAP 股份公司在德国和其它国家的商标或注册商标。

如欲了解更多商标信息和声明, 请访问: <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>。

# 目录

读者 .....	1
文件位置和安装设置 .....	3
安装目录结构 .....	3
SAP Sybase IQ 如何定位文件 .....	4
简单文件搜索 .....	5
广泛文件搜索 .....	5
环境变量 .....	6
在 Windows 上设置环境变量 .....	6
运行 UNIX 环境源文件 .....	6
在 UNIX 上设置环境变量 .....	6
IQCHARSET 环境变量 .....	7
IQDIR16 环境变量 .....	7
IQLANG 环境变量 .....	8
IQLOGDIR16 环境变量 .....	9
IQTMP16 环境变量 .....	10
IQ_USE_DIRECTIO 环境变量 .....	10
JAVA_HOME 环境变量 .....	11
LIBPATH 或 LD_LIBRARY_PATH 环境变量 .....	11
PATH 环境变量 .....	12
SQLCONNECT 环境变量 .....	12
SYBASE 环境变量 .....	12
\$SYBASE_JRE7_64、\$SYBASE_JRE7_32 环 境变量 .....	13
SYBASE_OCS 环境变量 .....	13
注册表条目 .....	14
当前用户和本地计算机设置 .....	14
注册表结构 .....	14
安装时的注册表设置 .....	15
SQL 语言元素 .....	17
关键字 .....	17
保留字 .....	17

标识符 .....	24
数据库服务器命名限制 .....	25
字符串 .....	26
表达式 .....	27
表达式中的常量 .....	28
表达式中的列名 .....	29
表达式中的子查询 .....	29
SQL 运算符 .....	30
IF 表达式 .....	34
CASE 表达式 .....	35
表达式和常量的兼容性 .....	36
搜索条件 .....	39
比较条件 .....	40
三值逻辑 .....	42
搜索条件中的子查询 .....	43
ALL 或 ANY 条件 .....	46
BETWEEN 条件 .....	47
带有逻辑运算符的条件 .....	47
CONTAINS 条件 .....	48
EXISTS 条件 .....	48
IN 条件 .....	49
IS DISTINCT FROM 搜索条件 .....	49
IS NULL 条件 .....	50
LIKE 条件 .....	50
NOT 条件 .....	54
真值条件 .....	54
用户提供的条件提示 .....	54
特殊值 .....	62
CURRENT DATABASE 特殊值 .....	62
CURRENT DATE 特殊值 .....	62
CURRENT PUBLISHER 特殊值 .....	63
CURRENT TIME 特殊值 .....	63
CURRENT TIMESTAMP 特殊值 .....	63
CURRENT USER 特殊值 .....	64

LAST USER 特殊值 .....	64
SQLCODE 特殊值 .....	64
SQLSTATE 特殊值 .....	65
TIMESTAMP 特殊值 .....	65
USER 特殊值 .....	65
变量 .....	66
局部变量 .....	66
连接级变量 .....	67
全局变量 .....	68
注释 .....	72
空值 .....	73
<b>SQL 函数 .....</b>	<b>75</b>
集合函数 .....	75
分析函数 .....	77
窗口化集合函数的用法 .....	78
排名函数的用法 .....	79
统计聚合分析函数的用法 .....	80
分布函数的用法 .....	80
行间函数的用法 .....	81
数据类型转换函数 .....	83
日期和时间函数 .....	83
日期分量 .....	85
HTTP 函数 .....	86
数值函数 .....	87
字符串函数 .....	88
系统函数 .....	90
Adaptive Server 系统函数兼容性 .....	91
连接属性 .....	92
可用于服务器的属性 .....	92
各数据库可用的属性 .....	115
SQL 和外部环境用户定义的函数 .....	131
SQL 中用户定义的函数 .....	131
Java 中用户定义的函数 .....	132
杂项函数 .....	132

按字母顺序排列的函数列表 .....	132
ABS 函数 [数值] .....	133
ACOS 函数 [数值] .....	133
ARGN 函数 [杂项] .....	134
ASCII 函数 [字符串] .....	135
ASIN 函数 [数值] .....	136
ATAN 函数 [数值] .....	137
ATAN2 函数 [数值] .....	137
AVG 函数 [集合] .....	138
BFILE 函数 [数据提取] .....	139
BIGINTTOHEX 函数 [数据类型转换] .....	139
BIT_LENGTH 函数 [字符串] .....	140
BYTE_LENGTH 函数 [字符串] .....	141
BYTE_LENGTH64 函数 .....	142
BYTE_SUBSTR64 和 BYTE_SUBSTR 函数 .....	143
CAST 函数 [数据类型转换] .....	143
CEIL 函数 [数值] .....	144
CEILING 函数 [数值] .....	145
CHAR 函数 [字符串] .....	146
CHAR_LENGTH 函数 [字符串] .....	147
CHAR_LENGTH64 函数 .....	148
CHARINDEX 函数 [字符串] .....	148
COALESCE 函数 [杂项] .....	149
COL_LENGTH 函数 [系统] .....	150
COL_NAME 函数 [系统] .....	151
CONNECTION_PROPERTY 函数 [系统] .....	152
CONVERT 函数 [数据类型转换] .....	153
CORR 函数 [集合] .....	157
COS 函数 [数值] .....	158
COT 函数 [数值] .....	159
COVAR_POP 函数 [集合] .....	160
COVAR_SAMP 函数 [集合] .....	161
COUNT 函数 [集合] .....	162
CUME_DIST 函数 [分析] .....	163

DATALENGTH 函数 [系统] .....	164
DATE 函数 [日期和时间] .....	165
DATEADD 函数 [日期和时间] .....	165
DATECEILING 函数 [日期和时间] .....	167
DATEDIFF 函数 [日期和时间] .....	169
DATEFLOOR 函数 [日期和时间] .....	171
DATEFORMAT 函数 [日期和时间] .....	174
DATENAME 函数 [日期和时间] .....	175
DATEPART 函数 [日期和时间] .....	176
DATEROUND 函数 [日期和时间] .....	177
DATETIME 函数 [日期和时间] .....	180
DAY 函数 [日期和时间] .....	180
DAYNAME 函数 [日期和时间] .....	181
DAYS 函数 [日期和时间] .....	182
DB_ID 函数 [系统] .....	183
DB_NAME 函数 [系统] .....	184
DB_PROPERTY 函数 [系统] .....	185
DEGREES 函数 [数值] .....	186
DENSE_RANK 函数 [分析] .....	186
DIFFERENCE 函数 [字符串] .....	188
DOW 函数 [日期和时间] .....	189
ERRORMSG 函数 [杂项] .....	189
EVENT_CONDITION 函数 [系统] .....	190
EVENT_CONDITION_NAME 函数 [系统] .....	192
EVENT_PARAMETER 函数 [系统] .....	192
EXP 函数 [数值] .....	194
EXP_WEIGHTED_AVG 函数 [集合] .....	194
FIRST_VALUE 函数 [集合] .....	196
FLOOR 函数 [数值] .....	198
GETDATE 函数 [日期和时间] .....	199
GRAPHICAL_PLAN 函数 [字符串] .....	199
GROUPING 函数 [集合] .....	201
GROUP_MEMBER 函数 [系统] .....	202
HEXTOBIGINT 函数 [数据类型转换] .....	203

HEXTOINT 函数 [数据类型转换]	204
HOUR 函数 [日期和时间]	205
HOURS 函数 [日期和时间]	206
HTML_DECODE 函数 [Miscellaneous]	207
HTML_ENCODE 函数 [Miscellaneous]	208
HTML_PLAN 函数 [字符串]	209
HTTP_DECODE 函数 [Web 服务]	211
HTTP_ENCODE 函数 [Web 服务]	211
HTTP_HEADER 函数 [Web 服务]	212
HTTP_VARIABLE 函数 [Web 服务]	214
IFNULL 函数 [杂项]	216
INDEX_COL 函数 [系统]	216
INSERTSTR 函数 [字符串]	217
INTTOHEX 函数 [数据类型转换]	218
ISDATE 函数 [日期和时间]	219
ISNULL 函数 [杂项]	220
ISNUMERIC 函数 [杂项]	221
LAG 函数 [分析]	222
LAST_VALUE 函数 [集合]	223
LCASE 函数 [字符串]	225
LEAD 函数 [分析]	226
LEFT 函数 [字符串]	228
LEN 函数 [字符串]	229
LENGTH 函数 [字符串]	230
LIST 函数 [集合]	231
LN 函数 [数值]	233
LOCATE 函数 [字符串]	234
LOG 函数 [数值]	235
LOG10 函数 [数值]	236
LOWER 函数 [字符串]	237
LTRIM 函数 [字符串]	238
MAX 函数 [集合]	239
MEDIAN 函数 [集合]	240
MIN 函数 [集合]	241



MINUTE 函数 [日期和时间]	242
MINUTES 函数 [日期和时间]	242
MOD 函数 [数值]	244
MONTH 函数 [日期和时间]	244
MONTHNAME 函数 [日期和时间]	245
MONTHS 函数 [日期和时间]	246
NEWID 函数 [杂项]	247
NEXT_CONNECTION 函数 [系统]	248
NEXT_DATABASE 函数 [系统]	250
NEXT_HTTP_HEADER 函数 [Web 服务]	251
NEXT_HTTP_VARIABLE 函数 [Web 服务]	252
NOW 函数 [日期和时间]	253
NTILE 函数 [分析]	253
NULLIF 函数 [杂项]	255
NUMBER 函数 [杂项]	256
OBJECT_ID 函数 [系统]	257
OBJECT_NAME 函数 [系统]	257
OCTET_LENGTH 函数 [字符串]	258
PATINDEX 函数 [字符串]	259
PERCENT_RANK 函数 [分析]	261
PERCENTILE_CONT 函数 [分析]	262
PERCENTILE_DISC 函数 [分析]	264
PI 函数 [数值]	266
POWER 函数 [数值]	266
PROPERTY 函数 [系统]	267
PROPERTY_DESCRIPTION 函数 [系统]	268
PROPERTY_NAME 函数 [系统]	269
PROPERTY_NUMBER 函数 [系统]	270
QUARTER 函数 [日期和时间]	270
RADIANS 函数 [数值]	271
RAND 函数 [数值]	272
RANK 函数 [分析]	273
REGR_AVGX 函数 [集合]	274
REGR_AVGY 函数 [集合]	275

REGR_COUNT 函数 [集合]	276
REGR_INTERCEPT 函数 [集合]	277
REGR_R2 函数 [集合]	278
REGR_SLOPE 函数 [集合]	280
REGR_SXX 函数 [集合]	281
REGR_SXY 函数 [集合]	282
REGR_SYY 函数 [集合]	283
REMAINDER 函数 [数值]	284
REPEAT 函数 [字符串]	285
REPLACE 函数 [字符串]	286
REPLICATE 函数 [字符串]	288
REVERSE 函数 [字符串]	289
RIGHT 函数 [字符串]	290
ROUND 函数 [数值]	291
ROW_NUMBER 函数 [分析]	292
ROWID 函数 [杂项]	293
RTRIM 函数 [字符串]	295
SECOND 函数 [日期和时间]	296
SECONDS 函数 [日期和时间]	297
SIGN 函数 [数值]	298
SIMILAR 函数 [字符串]	298
SIN 函数 [数值]	299
SORTKEY 函数 [字符串]	300
SOUNDEX 函数 [字符串]	304
SP_HAS_ROLE 函数 [系统]	305
SPACE 函数 [字符串]	307
SQLFLAGGER 函数 [杂项]	307
SQRT 函数 [数值]	309
SQUARE 函数 [数值]	309
STDDEV 函数 [集合]	310
STDDEV_POP 函数 [集合]	311
STDDEV_SAMP 函数 [集合]	312
STR 函数 [字符串]	314
STR_REPLACE 函数 [字符串]	315

STRING 函数 [字符串].....	316
STRTOUUID 函数 [字符串].....	317
STUFF 函数 [字符串].....	318
SUBSTRING 函数 [字符串].....	319
SUBSTRING64 函数 [字符串].....	322
SUM 函数 [集合].....	322
SUSER_ID 函数 [系统].....	323
SUSER_NAME 函数 [系统].....	324
TAN 函数 [数值].....	324
TODAY 函数 [日期和时间].....	325
TRIM 函数 [字符串].....	326
TRUNCNUM 函数 [数值].....	326
UCASE 函数 [字符串].....	327
UPPER 函数 [字符串].....	328
USER_ID 函数 [系统].....	329
USER_NAME 函数 [系统].....	330
UUIDTOSTR 函数 [字符串].....	331
VAR_POP 函数 [集合].....	332
VAR_SAMP 函数 [集合].....	333
VARIANCE 函数 [集合].....	334
WEEKS 函数 [日期和时间].....	336
WEIGHTED_AVG 函数 [集合].....	337
WIDTH_BUCKET 函数 [数值].....	339
YEAR 函数 [日期和时间].....	341
YEARS 函数 [日期和时间].....	342
YMD 函数 [日期和时间].....	343
<b>SQL 数据类型.....</b>	<b>345</b>
字符数据类型.....	345
存储大小.....	346
字符集和代码页.....	346
索引.....	347
VARCHAR 数据和尾随空白.....	347
对于超过 255 字节的 CHAR 和 VARCHAR 数据 的限制.....	348

字符数据兼容性 .....	348
长字符串 .....	348
数值数据类型 .....	348
数值数据类型的用法 .....	349
二进制数据类型 .....	352
二进制数据类型的用法 .....	353
位数据类型 .....	357
位数据兼容性 .....	357
日期和时间数据类型 .....	358
日期和时间数据类型的用法 .....	358
域 .....	362
简单域 .....	363
CREATE DOMAIN 语句 .....	364
域兼容性 .....	366
数据类型转换 .....	366
字符串到日期时间转换的兼容性 .....	367
导出的日期的兼容性 .....	368
从 BIT 到 BINARY 数据类型的转换 .....	368
BIT 和 CHAR/VARCHAR 数据类型之间的转换 .....	369
<b>与其它 SQL 术语的区别 .....</b>	<b>371</b>
日期 .....	371
完整性 .....	371
连接 .....	372
更新 .....	372
更改表 .....	372
有时不允许使用子查询 .....	372
附加函数 .....	373
游标 .....	373
<b>物理限制 .....</b>	<b>375</b>
<b>系统过程 .....</b>	<b>379</b>
管理特许系统过程执行 .....	379
迁移兼容性角色的影响 .....	380
授予运行特许系统过程的权限 .....	381
撤消运行特许系统过程的权限 .....	381

确定数据库所使用的安全模型 .....	381
16.0 之前版本的特许系统过程 .....	382
存储过程的语法规则 .....	384
了解存储过程报告的统计信息 .....	385
SQL Anywhere 支持过程 .....	385
按字母顺序排列的系统存储过程列表 .....	385
sp_iqaddlogin 过程 .....	386
sp_iqbackupdetails 过程 .....	387
sp_iqbackupsummary 过程 .....	389
sp_iqcardinality_analysis 过程 .....	390
sp_iqcheckdb 过程 .....	392
sp_iqcheckoptions 过程 .....	398
sp_iqclient_lookup 过程 .....	400
sp_iqcolumn 过程 .....	401
sp_iqcolumnmetadata 过程 .....	404
sp_iqcolumnuse 过程 .....	404
sp_iqconnection 过程 .....	406
sp_iqconstraint 过程 .....	409
sp_iqcontext 过程 .....	410
sp_iqcopyloginpolicy 过程 .....	413
sp_iqcursorinfo 过程 .....	413
sp_iqdatatype 过程 .....	415
sp_iqdbsize 过程 .....	418
sp_iqdbspace 过程 .....	419
sp_iqdbspaceinfo 过程 .....	423
sp_iqdbspaceobjectinfo 过程 .....	426
sp_iqdbstatistics 过程 .....	430
sp_iqdroplogin 过程 .....	431
sp_iqemptyfile 过程 .....	432
sp_iqestdbspaces 过程 .....	433
sp_iqestspace 过程 .....	435
sp_iqevent 过程 .....	436
sp_iqfile 过程 .....	438
sp_iqhelp 过程 .....	441

sp_iqindex 和 sp_iqindex_alt 过程 .....	447
sp_iqindexadvice 过程 .....	450
sp_iqindexfragmentation 过程 .....	452
sp_iqindexinfo 过程 .....	453
sp_iqindexmetadata 过程 .....	455
sp_iqindexfragmentation 过程 .....	458
sp_iqindexsize 过程 .....	459
sp_iqindexuse 过程 .....	461
sp_iqlmconfig 过程 .....	463
sp_iqllocks 过程 .....	465
sp_iqmergerlvstore 过程 .....	468
sp_iqmodifyadmin 过程 .....	468
sp_iqmodifylogin 过程 .....	469
sp_iqmpxcheckdqpconfig 过程 .....	470
sp_iqmpxdumptlvlog 过程 .....	472
sp_iqmpxfilestatus 过程 .....	473
sp_iqmpxinconnpoolinfo 过程 .....	474
sp_iqmpxincheartbeatinfo 过程 .....	475
sp_iqmpxincstatistics 过程 .....	476
sp_iqmpxinfo 过程 .....	477
sp_iqmpxsuspendedconninfo 过程 .....	478
sp_iqmpxvalidate 过程 .....	480
sp_iqmpxversioninfo 过程 .....	480
sp_iqobjectinfo 过程 .....	481
sp_iqpassword 过程 .....	484
sp_iqpkeys 过程 .....	485
sp_iqprocedure 过程 .....	487
sp_iqprocparm 过程 .....	489
sp_iqrebuildindex 过程 .....	492
sp_iqrebuildindex 过程 .....	495
sp_iqrename 过程 .....	498
sp_iq_reset_identity 过程 .....	499
sp_iqrestoreaction 过程 .....	500
sp_iqrlvmemory 过程 .....	502

sp_iqrowdensity 过程 .....	503
sp_iqsetcompression 过程 .....	504
sp_iqsharedtempdistrib 过程 .....	505
sp_iqshowcompression 过程 .....	506
sp_iqshowpsexec 过程 .....	507
sp_iqspaceinfo 过程 .....	509
sp_iqspaceused 过程 .....	510
sp_iqstatistics 过程 .....	512
sp_iqstatus 过程 .....	515
sp_iqsysmon 过程 .....	518
sp_iqtable 过程 .....	539
sp_iqtablesize 过程 .....	543
sp_iqtableuse 过程 .....	544
sp_iqtransaction 过程 .....	545
sp_iqunusedcolumn 过程 .....	549
sp_iqunusedindex 过程 .....	550
sp_iqunusedtable 过程 .....	551
sp_iqversionuse 过程 .....	552
sp_iqview 过程 .....	554
sp_iqwho 过程 .....	556
sp_iqworkmon 过程 .....	559
按字母顺序排列的目录存储过程的列表 .....	560
sa_ansi_standard_packages 系统过程 .....	561
sa_audit_string 系统过程 .....	561
sa_char_terms 系统过程 .....	562
sa_checkpoint_execute 系统过程 .....	563
sa_conn_activity 系统过程 .....	564
sa_conn_info 系统过程 .....	565
sa_conn_list 系统过程 .....	568
sa_conn_properties 系统过程 .....	569
sa_db_info 系统过程 .....	570
sa_db_properties 系统过程 .....	572
sa_describe_shapefile 系统过程 .....	573
sa_dependent_views 系统过程 .....	574

sa_disable_auditing_type 系统过程 .....	576
sa_disk_free_space 系统过程 .....	577
sa_enable_auditing_type 系统过程 .....	578
sa_eng_properties 系统过程 .....	579
sa_external_library_unload 系统过程 .....	580
sa_flush_cache 系统过程 .....	580
sa_get_user_status 系统过程 .....	581
sa_get_ldapsrvr_status 系统过程 .....	582
sa_list_external_library 系统过程 .....	583
sa_locks 系统过程 .....	584
sa_make_object 系统过程 .....	587
sa_nchar_terms 系统过程 .....	588
sa_performance_diagnostics 系统过程 .....	589
sa_procedure_profile_summary 系统过程 .....	594
sa_procedure_profile 系统过程 .....	596
sa_report_deadlocks 系统过程 .....	598
sa_rowgenerator 系统过程 .....	599
sa_server_option 系统过程 .....	601
sa_set_http_header 系统过程 .....	613
sa_set_http_option 系统过程 .....	614
sa_table_page_usage 系统过程 .....	617
sa_text_index_stats 系统过程 .....	618
sa_text_index_vocab 系统过程 .....	619
sa_validate 系统过程 .....	621
sa_verify_password 系统过程 .....	622
sp_alter_secure_feature_key 系统过程 .....	622
sp_auth_sys_role_info 系统过程 .....	623
sp_create_secure_feature_key 系统过程 .....	623
sp_displayroles 系统过程 .....	624
sp_drop_secure_feature_key 系统过程 .....	627
sp_expireallpasswords 系统过程 .....	628
sp_list_secure_feature_keys 系统过程 .....	628
sp_login_environment 系统过程 .....	629
sp_objectpermission 系统过程 .....	629



sp_proc_priv 系统过程 .....	633
sp_remote_columns 系统过程 .....	634
sp_remote_exported_keys 系统过程 .....	636
sp_remote_imported_keys 系统过程 .....	637
sp_remote_primary_keys 系统过程 .....	639
sp_remote_tables 系统过程 .....	640
sp_servercaps 系统过程 .....	642
sp_sys_priv_role_info 系统过程 .....	643
sp_tsql_environment 系统过程 .....	643
sp_use_secure_feature_key 系统过程 .....	644
Adaptive Server 系统过程和目录过程 .....	644
Adaptive Server 系统过程 .....	645
Adaptive Server 目录过程 .....	646
<b>系统表和视图 .....</b>	<b>649</b>
系统表 .....	649
SYS.DUMMY 表与 IQ_DUMMY 表 .....	652
系统视图 .....	653
统一视图 .....	653
兼容性视图 .....	653
按字母顺排序的系统视图列表 .....	653
<b>与其它 Sybase 数据库的兼容性 .....</b>	<b>755</b>
SQL Anywhere 的说明 .....	755
Transact-SQL 支持概述 .....	755
Adaptive Server、SQL Anywhere 和 SAP Sybase IQ	
体系结构 .....	756
服务器和数据库 .....	756
空间分配和设备管理 .....	757
系统表、目录存储库和 IQ 主存储库 .....	757
数据类型 .....	758
位数据类型 .....	758
字符数据类型兼容性 .....	758
二进制数据类型 .....	759
日期、时间、日期时间和时间戳数据类型 .....	760
数值数据类型 .....	761

文本数据类型 .....	761
图像数据类型 .....	762
Java 数据类型 .....	762
Data Definition Language (数据定义语言) .....	762
使用 CREATE DATABASE 语句创建与 Transact- SQL 兼容的数据库 .....	762
区分大小写 .....	762
确保兼容的对象名 .....	763
使用 CREATE TABLE 语句时的注意事项 .....	764
使用 CREATE DEFAULT、CREATE RULE 和 CREATE DOMAIN 语句时的注意事项 .....	767
使用 CREATE TRIGGER 语句时的注意事项 .....	767
使用 CREATE INDEX 语句时的注意事项 .....	767
用户、组/角色以及权限 .....	768
装载格式 .....	770
Transact-SQL 兼容性选项 .....	770
Data Manipulation Language (数据操纵语言) .....	770
编写可移植 SQL 的一般原则 .....	770
编写兼容查询的标准 .....	770
子查询支持 .....	771
GROUP BY 子句支持 .....	772
COMPUTE 子句支持 .....	772
WHERE 子句支持 .....	772
Transact-SQL 外连接支持 .....	772
ANSI 连接支持 .....	773
空值比较支持 .....	773
零长度字符串支持 .....	773
HOLDLOCK、SHARED 和 FOR BROWSE 支持 .....	774
SQL 函数支持 .....	774
OLAP 函数支持 .....	775
系统函数支持 .....	776
用户定义的函数支持 .....	776
解释日期的算术表达式的区别 .....	776

SELECT INTO 语句支持 .....	776
可更新的视图支持 .....	777
支持 UPDATE 和 DELETE 中的 FROM 子句 .....	777
Transact-SQL 过程语言概述 .....	777
Transact-SQL 存储过程概述 .....	777
Transact-SQL 批处理概述 .....	778
过程和批处理中的 SQL 语句 .....	778
存储过程的自动转换 .....	779
Transact-SQL 过程的结果集 .....	780
Transact-SQL 过程中的变量 .....	780
Transact-SQL 过程中的错误处理 .....	781
在过程中使用 RAISERROR 语句 .....	782
Watcom-SQL 术语中与 Transact-SQL 类似的错 误处理 .....	783
SQL Anywhere 和 SAP Sybase IQ 的区别和共享功能 .....	783
SQL Anywhere 服务器和数据库启动与管理 .....	783
SQL Anywhere 数据定义语言 (DDL) 区别 .....	784
SQL Anywhere 数据操作语言 (DDL) 区别 .....	784
SAP Adaptive Server <sup>®</sup> Enterprise 和 SAP Sybase IQ 的区别和共享功能 .....	785
Adaptive Server 存储过程 .....	785
Adaptive Server 系统视图 .....	786
索引 .....	789



# 读者

本手册面向需要 SQL 语句、语言元素、数据类型、函数、系统过程和系统表等参考资料的 SAP® Sybase® IQ 用户。

其它手册提供有关如何执行特定任务的详细信息。可以使用本手册来获取有关 SQL 语法、参数和选项的信息。有关命令行实用程序启动参数，请参见《实用程序指南》。

读者

# 文件位置和安装设置

以下主题介绍 SAP Sybase IQ 所使用的安装和操作系统设置。

根据操作系统的不同，这些设置可能作为环境变量、初始化文件条目或注册表条目进行存储。

## 安装目录结构

---

安装 SAP Sybase IQ 时，可能会创建多个目录。本节介绍目录结构。

系统会创建哪些目录取决于在安装期间选择了哪些选项，以及 Sybase 目录（该目录在 UNIX 上由 `$SYBASE` 定义，在 Windows 上由 `%SYBASE%` 定义）中已经存在哪些目录。

缺省情况下，SAP Sybase IQ 软件将安装在 Sybase 目录下面的唯一子目录中。该子目录称为安装目录。SAP Sybase IQ 附带的其它工具在 Sybase 目录下面具有唯一子目录。本节只介绍 SAP Sybase IQ 的子目录结构。

缺省情况下，SAP Sybase IQ 目录为 `IQ-16_0`。`IQ-16_0` 的位置因 SAP Sybase IQ 的安装位置而异。此外，`IQ-16_0` 目录在 UNIX 上由环境变量 `$IQDIR16` 引用，在 Windows 上由 `%IQDIR16%` 引用。

SAP Sybase IQ 目录下有大量目录和文件：

- **Demo 目录** (`%ALLUSERSPROFILE%/SybaseIQ/demo`) - 包含建立 `iqdemo` 数据库所需的工具。`iqdemo` 数据库文件包括 `iqdemo.db`、`iqdemo.iq`、`iqdemo.iqmsg` 和 `iqdemo.iqtmp`。SAP Sybase IQ 并不提供演示数据库。
- **子目录** `/demo/adata` - 包含可用于创建 16.x `iqdemo` 数据库的 16.x 数据。子目录 `/demo/demodata` 包含 SAP Sybase IQ 12.7 数据，这些数据可用于创建具有与 `IQ 12.7 asiqdemo` 数据库相同的模式布局和数据的数据的 `iqdemo` 数据库。使用 Windows 上的 `/demo/mkiqdemo.bat` 和 UNIX 上的 `demo/mkiqdemo.sh` 创建 16.x `iqdemo` 数据库。`iqdemo` 数据库可以用于向技术支持人员演示问题。
- **Scripts 目录** (`IQ-16_0/scripts`) - 包含一些用于示例以及在创建存储过程等目录对象时使用的脚本。不要编辑这些脚本。如果编辑、删除或移动这些脚本，则服务器将无法正常工作。
- **samples 目录** - `samples` 目录包含 SQL 示例和用户定义的函数 (UDF) 示例。`%ALLUSERSPROFILE%/SybaseIQ/samples/sqlanywhere` 包含 SQL 示例的目录。`sqlanywhere/c directory` 包含说明如何将 `ESQL` (嵌入式 SQL) 和 C 用于 SAP Sybase SQL Anywhere® 的 C++ 示例。由于 `SQL Anywhere` 和 `SAP Sybase IQ` 共享公用代码，因此您可以修改这些示例以便用于 `SAP Sybase IQ`。

`%ALLUSERSPROFILES%/SybaseIQ/samples/udf` 目录包含 C++ 标量和集合 UDF 示例。

- **可执行文件目录** - 包含可执行文件、库、帮助文件等等。在 **UNIX** 上，可执行文件子目录包括 IQ-16\_0 子目录 `/bin64`、`/lib64`、`/logfiles`、`/res` 和 `/tix`。在 **Windows** 上，这些子目录包括 IQ-16\_0 子目录 `\h`、`\install`、`\java` 和 `\bin32`。

## SAP Sybase IQ 如何定位文件

---

SAP Sybase IQ 必须查找并访问几种类型的文件，才能启动和运行。系统上可能驻留一些同名的目录或文件。

为确保使用正确的文件，有必要了解 SAP Sybase IQ 如何查找这些文件。这些文件的类型包括但不限于：

- **库** - 可能包括产品库或系统库。在 **UNIX** 上，文件扩展名包括 `.so.nnn` 或 `.so`，而在 **Windows** 上则包括 `.dll` 或 `.lib`。这些文件对于运行 SAP Sybase IQ 必不可少。如果找到错误的 **DLL**，则有可能发生版本不匹配错误。例如，在 **UNIX** 上，库文件可能位于 `$IQDIR16/lib64` 或 `$SYBASE/$SYBASE_OCS/lib64` 中，而在 **Windows** 上则可能位于 `%IQDIR16%\bin32` 或 `%SYBASE%\SYBASE_OCS\dll` 中。空目录 `$IQDIR16/usrlib` 会使您用自定义库和补丁取代缺省库，因为 **start\_iq** 所包含的 `usrlib` 位于常规库目录之前。SAP Sybase IQ 使用 SAP Adaptive Server® Enterprise 和 SQL Anywhere 库。如果系统中已安装了任意一种产品，则应注意它们的安装目录，以避免混淆。
- **接口文件** - 对于运行 SAP Sybase IQ 必不可少。例如，在 **UNIX** 上为 `.odbc.ini` 和 `utility_db.ini`，在 **Windows** 上为 `util_db.ini`。有关这些文件的详细信息，请参见《管理：用户管理和安全》。
- **配置文件** - 用于指定连接参数。示例包括 **Windows** 上的 `default.cfg` 或 `iqdemo.cfg`。
- **数据库文件** - 存储数据和元数据。例如：`iqdemo.db`、`iqdemo.iq`、`iqdemo.iqmsg`、`iqdemo.iqtmp`。
- **日志文件** - 存储有关服务器和连接的数据库上的当前会话信息。例如，服务器日志可能命名为 `%ALLUSERSPROFILE%/logfiles/yourservername.0006.srvlog`。连接到数据库时将创建数据库日志（例如 `%ALLUSERSPROFILE%/demo/iqdemo.log`）。有关这些文件的详细信息，请参见《管理：数据库》和《实用程序指南》。
- **产品脚本** - 是显示如何创建、填充和升级数据库的示例文件。
- **用户文件** - 包括用于 **LOAD** 命令的平面文件，以及用于 **Interactive SQL** 等工具的 **SQL** 脚本。
- **临时文件** - 其创建目的是存储查询排序等操作执行过程中产生的临时信息。

某些文件名在 **SQL** 语句中指定，并且必须在运行时定位。使用文件名的 **SQL** 语句的示例包括以下语句：



- **INSTALL** 语句 - 包含 Java 类的文件的名称。
- **LOAD TABLE** 语句 - 应从其中装载数据的文件的名称。
- **CREATE DATABASE** 语句 - 此语句以及可以创建文件的相似语句所需的文件名。

在某些情况下，SAP Sybase IQ 使用简单的算法来定位文件。在其它情况下，则会执行更广泛的搜索。

## 简单文件搜索

在诸如 **LOAD TABLE** 或 **CREATE DATABASE** 这样的很多 SQL 语句中，系统会以相对于数据库服务器的当前工作目录（即服务器启动时所在的目录）的方式来解释文件名。

此外，当启动数据库服务器并提供数据库文件名（DBF 参数）时，会以相对于服务器启动时所在目录的方式来解释路径。

## 广泛文件搜索

SAP Sybase IQ 程序（包括数据库服务器和管理实用程序）执行广泛搜索来查找所需的文件，如 DLL 或共享库。

在上述情况下，SAP Sybase IQ 程序按以下顺序查找文件：

- **可执行文件目录** - 程序可执行文件所在的目录。同时也是与程序可执行文件目录相关的以下路径的目录：
  - 可执行文件目录的父目录。
  - 名为 `scripts.` 的父目录的子目录。
- **当前工作目录** - 程序启动时，其具有当前工作目录（从中启动程序的目录）。在此目录中搜索所需的文件。
- **Location 注册表条目** - 在 Windows 安装中，SAP Sybase IQ 添加一条 LOCATION 注册表条目。先搜索指示的目录，然后搜索：
  - 名为 `scripts` 的子目录
  - 具有操作系统名称的子目录（`bin32` 和 `bin` 等等）
- **系统特定的目录** - 包括保存公用操作系统文件的目录，例如 Windows 操作系统上的 `Windows` 目录和 `Windows\system` 目录。
- **CLASSPATH 目录** - 针对 Java 文件，系统将搜索 CLASSPATH 环境变量中列出的目录以定位文件。
- **PATH 目录** - 系统路径和用户路径中的目录，用于搜索文件。
- **LIBRARY PATH 目录** - LIBPATH 环境变量中列出的目录，用于搜索共享库。

## 环境变量

---

SAP Sybase IQ 使用环境变量来存储各种类型的信息；并不是在所有情况下都需要设置所有的变量。

### 在 Windows 上设置环境变量

在 Windows 平台上，安装程序会自动设置所有环境变量，因此不需要进行更改。但是，如果必须设置可选变量或更改缺省值，请使用以下过程。

1. 在桌面上，右键单击“我的电脑”，并从子菜单中选择“属性”。
2. 单击“高级”选项卡。
3. 单击“环境变量”按钮。

将打开“环境变量”对话框。

- a) 如果环境变量尚不存在，请单击“新建”，并在所提供的空白处键入变量的名称及其值；然后单击“确定”。
- b) 如果变量已经存在，请从“系统变量”或“用户变量”的列表中将其选定，单击“编辑”，并在“变量值”字段中进行修改。然后单击“确定”以捕获设置。

---

**注意：** 有关用户变量和系统变量的说明，请参见 Microsoft Windows 文档。

---

### 运行 UNIX 环境源文件

环境源文件设置 UNIX 上必需的环境变量。

发出以下命令以设置所有必需的环境变量。

1. 对于 Bourne/Korn shell:

```
. $SYBASE/IQ-16_0/IQ-16_0.sh
```

2. 对于 C shell:

```
source $SYBASE/IQ-16_0/IQ-16_0.csh;  
rehash
```

### 在 UNIX 上设置环境变量

在 UNIX 平台上，运行环境源文件可以设置必需的环境变量。但是，如果必须设置可选变量或更改缺省值，请使用以下过程。

1. 若要检查环境变量的设置，请使用：

```
echo $variable-name
```

例如，若要查看 \$SYBASE 变量的设置：

```
% echo $SYBASE
```

```
/server1/users/test/sybase
```

2. 在其中一个启动文件（.cshrc、.shrc、.login）中，添加一个设置变量的行。

在某些 shell（例如 sh、bash 和 ksh）中，该行如下所示：

```
VARIABLE=value;export VARIABLE
```

在其它 shell（例如 csh 和 tsch）中，该行如下所示：

```
setenv VARIABLE "
                    value"
```

## IQCHARSET 环境变量

IQCHARSET 设置缺省字符集。

*Charset* 是字符集名称。例如，设置 **IQCHARSET=cp1252** 可将缺省字符集设置为 cp1252。

以下值中，第一个设定的值将决定缺省的字符集。

- IQCHARSET 环境变量
- 查询操作系统

如果未指定任何字符集信息，则对 UNIX 使用 iso\_1，对其它操作系统使用 cp850。

*设置*

```
IQCHARSET=charset
```

## IQDIR16 环境变量

IQDIR16 用于标识 SAP Sybase IQ 目录的位置，以及该目录下的其它目录和文件的位置。

- \$IQDIR16/bin[64]/util\_db.ini 包含实用程序数据库 utility\_db 的登录 ID 和口令。安装程序允许您更改这些值的缺省值，登录 ID "DBA"，口令 "sql"。
- \$IQDIR16/logfiles 是服务器日志和备份/恢复日志（备份历史记录文件）的缺省位置。可以设置 IQLOGDIR16 环境变量以覆盖此缺省值。
- \$IQDIR16/demo 是 iqdemo 数据库文件所在的位置。

*设置*

```
IQDIR16 = ${SYBASE}/IQ-16_0
```

*操作系统*

（必需）由环境源文件或安装程序设置。在 Windows 上可以更改此缺省设置。

## IQLANG 环境变量

IQLANG 设置缺省语言。

*Language\_code* 是表示某种语言的双字母组合。例如，设置 **IQLANG=DE** 可将缺省语言设置为德语。

以下值中，第一个设定的值将决定缺省的语言。

- IQLANG 环境变量
- 由安装程序设置的注册表（仅限 Windows）
- 查询操作系统

如果未设置任何语言信息，则英语将是缺省语言。

### 设置

```
IQLANG=language_code
```

### 操作系统

可选，但在非英语环境中推荐使用。

### 语言标签值列表

有效的语言标签值以及等效的 ISO 639 语言代码。设置 IQ\_LANG 环境变量中的双字母 ISO\_639 语言代码。

语言	ISO_639 语言代码	语言标签	替代标签
阿拉伯语	AR	arabic	不适用
捷克语	CS	czech	不适用
丹麦语	DA	danish	不适用
荷兰语	NL	dutch	不适用
英语	EN	us_english	english
芬兰语	FI	finnish	不适用
法语	FR	french	不适用
德语	DE	german	不适用
希腊语	EL	greek	不适用
希伯来语	HE	hebrew	不适用
匈牙利语	HU	hungarian	不适用
意大利语	IT	italian	不适用

语言	ISO_639 语言代码	语言标签	替代标签
日语	JA	japanese	不适用
朝鲜语	KO	korean	不适用
立陶宛语	LT	lithuanian	不适用
挪威语	NO	norwegian	norweg
波兰语	PL	polish	不适用
葡萄牙语	PT	portuguese	portugue
俄语	RU	russian	不适用
简体中文	ZH	chinese	simpchin
西班牙语	ES	spanish	不适用
瑞典语	SV	swedish	不适用
泰语	TH	thai	不适用
繁体中文	TW	tchinese	tradchin
土耳其语	TR	turkish	不适用
乌克兰语	UK	ukrainian	不适用

## IQLOGDIR16 环境变量

IQLOGDIR16 环境变量定义各种日志文件的位置。IQLOGDIR16 不是由安装程序设置的。

- 服务器日志位于 \$IQLOGDIR16 指定的目录下的 `servername.nnnn.srvlog` 文件（其中，`nnnn` 为服务器已启动的次数）中。

如果未将 IQLOGDIR16 设置为有效且可写入的目录，则大多数实用程序（包括 `start_iq`）将使用所有服务器日志的缺省位置 `$IQDIR16/logfiles`。

### 设置

```
IQLOGDIR16 = path
```

### 操作系统

可选。

## **IQTMP16 环境变量**

IQTMP16 环境变量不是由安装程序设置的。SAP Sybase IQ 使用 IQTMP16 指示保存临时文件的目录。

IQTMP16 环境变量应当指向那些使用 NFS（网络文件系统）的文件的本地目录，这样可以允许 IQTMP16 目录清除客户端连接关闭后不再需要的目录和文件。每个客户端连接都会在临时目录中创建若干个目录和文件。这些目录和文件仅用于连接期间。连接到服务器的所有用户都必须对该目录具有写入权限。

---

**注意：**通过 IQTMP16 定义位置的临时文件是客户端和服务器中使用的文件。此变量不控制 IQ 临时存储库的缺省位置。**CREATE DATABASE** 语句控制 IQ 临时存储库的缺省位置。

---

**警告！** 请勿将 IQTMP16 设置为 \$SYBASE 或 \$IQDIR16。

---

如果未显式设置 IQTMP16，则 IQTMP16 将设置为 UNIX 目录 /tmp 下的子目录。

如果有多个数据库服务器正在同一计算机上运行，则每个服务器及其关联的本地客户端都需要单独的临时目录，以避免冲突。如果不指定连接的端口或引擎号，则 SAP Sybase IQ 将使用共享的内存连接，而不是网络连接。

若要避免使用共享内存时发生冲突，请使用以下操作：

- 创建每个服务器专用的临时目录。通过在两个环境中显式设置 IQTMP16 环境变量，确保每个客户端都与其服务器使用相同的临时目录。
- 在 .odbc.ini 文件（UNIX 上）中为每个服务器创建一个数据源名称，并提供详细的连接信息。
- 使用指定显式参数而不是依赖于缺省值的连接字符串。
- 通过发出以下命令确认连接：

```
SELECT "database name is" = db_name(), "servername_is" =  
@@servername
```

### *设置*

```
IQTMP16 = temp_directory
```

### *操作系统*

在 UNIX 上可选。在 Windows 平台上不使用。

## **IQ\_USE\_DIRECTIO 环境变量**

IQ\_USE\_DIRECTIO 在支持的 Simplex 平台上启用直接 I/O。

IQ\_USE\_DIRECTIO 环境变量缺省设置为 0。要在支持的平台上启用直接 I/O，将 IQ\_USE\_DIRECTIO 设置为 1。Multiplex 数据库不支持直接 I/O。

### *设置*

```
IQ_USE_DIRECTIO = value
```

### 操作系统

仅在 Sun Solaris UFS、Linux、Linux IBM、AIX 和 Windows 文件系统中为可选。在 Linux 中，内核版本 2.6.x 支持直接 I/O。

不在 HP-UX 和 HP-UXi 平台上使用。

IQ\_USE\_DIRECTIO 对 Sun Solaris 和 Windows 不起作用。

## JAVA\_HOME 环境变量

定义指向包含 bin/java 的目录的 JRE 主目录。

如果没有在 \$SYBASE\_JRE6\_32、\$SYBASE\_JRE6\_64 或 \$SYBASE\_JRE5\_64 环境变量中设置 Java VM 的位置，则使用此变量。

JAVA\_HOME 通常在安装 VM 时创建。

在 UNIX 上，运行 SYBASE.csh (C shell) 或 SYBASE.sh (Bourne 或 Korn shell) 环境源文件可查找并启动适用于 IQ 引擎的 JRE。在 JAVA\_HOME 中指定的 Java VM 位置优先于 SYBASE.csh 或 SYBASE.sh 返回的位置。如果 JAVA\_HOME、SYBASE.csh 或 SYBASE.sh 脚本均没有找到 Java VM，IQ 将不装载 Java VM。

### 设置

```
JAVA_HOME = Sybase/shared/JRE<version>
```

### 操作系统

必需。

## LIBPATH 或 LD\_LIBRARY\_PATH 环境变量

LIBPATH 或 LD\_LIBRARY\_PATH 指定 SAP Sybase IQ 共享库所在的目录。

在 UNIX 上，通过运行环境源文件可以设置库路径变量。

### 设置

```
For AIX:
LIBPATH = installation_path/lib
```

```
For all other UNIX/LINUX platforms:
LD_LIBRARY_PATH = installation_path/lib
```

### 操作系统

必需。变量名因平台而异。仅限于 UNIX。

## PATH 环境变量

PATH 是操作系统必需的变量，它包括 SAP Sybase IQ 可执行文件所在的目录。

在 Windows 上，安装程序会修改 PATH。在 UNIX 上，可以运行环境源文件以纳入必需的目录。

在 Windows 上，PATH 取代了 LIBRARY\_PATH 变量的位置，因此使用 PATH 变量可以找到可执行文件和 DLL。

### *设置*

```
PATH = installation_path
```

### *操作系统*

必需。

## SQLCONNECT 环境变量

SQLCONNECT 指定在连接到数据库服务器时几个数据库管理实用程序（例如 Interactive SQL、dbinfo 和 dbstop）使用的连接参数。

SQLCONNECT 环境变量是可选的，它不是由安装程序设置的。

此字符串是一个参数设置的列表，其格式为 **parameter=value**，以分号分隔。

井号“#”可以代替等号；如果您在 SQLCONNECT 环境变量中设置连接参数字符串，请使用井号。在环境变量设置中使用“=”是一个语法错误。=号仅允许在 Windows 中使用。

---

**注意：**对于 UNIX 系统，在 SQLCONNECT 中而不是在命令行上指定连接参数可以增加安全性。这样可防止用户通过 **ps -ef** 显示您的口令。如果在安静模式下运行 Interactive SQL 或其它实用程序，这一点尤为有用。请注意，在 SQLCONNECT 中而不是在命令行上指定连接参数可以提高安全性，但不能完全确保安全。因为口令采用纯文本形式，所以恶意操作可能从环境上下文中提取该口令。有关详细信息，请参见《管理：数据库》中的连接和通信参数参考。

---

### *设置*

```
SQLCONNECT = parameter#value ; ...
```

### *操作系统*

可选。

## SYBASE 环境变量

SYBASE 用于标识 Sybase 应用程序（比如 Open Client 和 Open Server）的位置。

必须先设置 SYBASE 变量，然后才能在 UNIX 系统中安装 SAP Sybase IQ。



*设置*

```
SYBASE = path
```

*操作系统*

必需。

**SYBASE\_JRE7\_64、SYBASE\_JRE7\_32 环境变量**

该变量指定 Sybase Control Center 所使用的 Java 运行时环境的位置。

启动时，Sybase Control Center 会检查用于 Java 版本定义的 SCC\_JAVA\_HOME。如果未定义 SCC\_JAVA\_HOME，Sybase Control Center 会按以下顺序检查安装的 JRE：

- SYBASE\_JRE7\_64
- SYBASE\_JRE7
- SYBASE\_JRE7\_32

然后，Sybase Control Center 会将 SCC\_JAVA\_HOME 设置为该列表中的第一个值。

*设置*

定位 IQ.sh (Bourne/Korn shell) IQ.csh (C shell) 文件。

---

**提示：** 或者可以手动设置 JRE：

```
SCC_JAVA_HOME=${SYBASE}/shared/JRE-7_(minor_version)_64BIT
```

或者：

```
SCC_JAVA_HOME=${SYBASE}/shared/JRE-7_(minor_version)_32BIT
```

---

**SYBASE\_OCS 环境变量**

SYBASE\_OCS 指定 Open Client 产品的主目录。

此变量仅在 Windows 上使用。在 Windows 上，安装程序会在安装 Open Client/Server 软件开发工具包时设置 SYBASE\_OCS。

*设置*

```
SYBASE_OCS = "OCS-16_0"
```

*操作系统*

必需。

## 注册表条目

---

在 Windows 操作系统上，SAP Sybase IQ 使用多个注册表设置。

本软件为您进行这些设置，而且在一般操作中您无需访问注册表。如果您修改操作环境，则可在此处找到设置。

---

**警告！** 不建议修改注册表，因为错误的更改可能损坏系统。

---

## 当前用户和本地计算机设置

某些操作系统（例如 Windows）具有两个级别的系统设置：用户设置和本地计算机设置。

当前用户设置是特定于单个用户的，仅当该用户登录时才会使用。本地计算机设置对计算机来说是全局的，不管哪个用户登录它都是可用的。您必须在计算机上有管理员权限，才能进行本地计算机设置。

SAP Sybase IQ 允许使用当前用户设置和本地计算机设置。对于 Windows，这些设置分别保存在 HKEY\_CURRENT\_USER 注册表和 HKEY\_LOCAL\_MACHINE 注册表中。

SAP Sybase IQ 安装程序允许您选择其设置是仅针对当前用户还是针对本地计算机级别。

如果在当前用户注册表和本地计算机注册表中都进行了某个设置，则当前用户设置优先于本地计算机设置。

如果要在 Windows 上将 SAP Sybase IQ 程序作为服务来运行，则应确保在本地计算机级别进行这些设置。

在您从计算机注销后，只要不彻底关闭计算机，服务就可以继续以特殊的帐户运行。可以使这些服务独立于单个帐户，因此需要访问本地计算机设置。

通常使用本地计算机设置。

## 注册表结构

在 Windows 上，可以直接使用注册表编辑器来访问注册表。

---

**注意：** 只读模式可防止注册表数据遭到意外更改。若要使用只读模式，请打开注册表编辑器，选择“编辑”|“权限”，然后再选中“读取”权限。

---

SAP Sybase IQ 注册表条目保存在 HKEY\_LOCAL\_MACHINE 注册表项中的以下位置：

SOFTWARE > SAP Sybase IQ 16.0

### 启动注册表编辑器

启动注册表编辑器，以访问 Windows 注册表。

1. 选择“开始” > “运行”。
2. 在“打开”框中，键入：

```
regedt32
```

### 安装时的注册表设置

安装程序会自动在 Sybase 注册表中进行以下注册表设置。

- Location - 在 SAP Sybase IQ 注册表中，此条目包含安装目录位置。例如：

```
Location:REG_SZ:C:\Program Files\Sybase  
\IQ-16_0
```

SAP Sybase IQ 注册表包括已安装应用程序的其它条目。



# SQL 语言元素

以下主题详细介绍 SAP Sybase IQ SQL 的语言元素和约定。

## 关键字

---

每个 SQL 语句都包含一个或多个关键字。

SQL 的关键字不区分大小写，但在整个 SAP Sybase IQ 文档中，关键字都以大写字母表示。例如，在以下语句中，SELECT 和 FROM 是关键字：

```
SELECT *  
FROM Employees
```

下列语句等同于上一语句：

```
Select *  
From Employees  
select * from Employees  
sELECT * FRoM Employees
```

## 保留字

SQL 中的一些关键字也是保留字。

要在 SQL 语句中使用保留字作为标识符，必须用双引号将它引起来。SQL 语句中出现的很多关键字都是保留字，但并非全部都是。例如，必须使用以下语法来检索名为 SELECT 的表的内容。

```
SELECT *  
FROM "SELECT"
```

如果使用嵌入式 SQL，则可以利用数据库库函数 `sql_needs_quotes` 确定某个字符串是否需要加上引号。如果一个字符串是保留字，或者包含标识符中通常不允许的字符，则该字符串需要加上引号。

下表列出了 SAP Sybase IQ 中的 SQL 保留字。由于 SQL 的关键字不区分大小写，因此下表中的各个保留字可以采用大写、小写或两者任意组合的形式显示。与这些关键字仅在大小写形式上有所不同的所有字符串都是保留字。

### SQL 保留字

- add
- all
- alter
- and

## SQL 语言元素

- any
- array
- as
- asc
- attach
- backup
- begin
- between
- bigint
- binary
- bit
- bottom
- break
- by
- call
- capability
- cascade
- case
- cast
- char
- char\_convert
- character
- check
- checkpoint
- close
- comment
- commit
- compressed
- conflict
- connect
- constraint
- contain
- continue
- convert
- create
- cross
- cube
- current
- current\_timestamp

- current\_user
- cursor
- date
- datetimeoffset
- dbspace
- deallocate
- dec
- decimal
- declare
- default
- delete
- deleting
- desc
- detach
- distinct
- do
- double
- drop
- dynamic
- else
- elseif
- encrypted
- end
- endif
- escape
- except
- exception
- exec
- execute
- existing
- exists
- externlogin
- fetch
- first
- float
- for
- force
- foreign
- forward

## SQL 语言元素

- from
- full
- goto
- grant
- group
- having
- holdlock
- identified
- if
- in
- index
- inner
- inout
- insensitive
- insert
- inserting
- install
- instead
- int
- integer
- integrated
- intersect
- into
- is
- isolation
- join
- json
- kerberos
- key
- lateral
- left
- like
- limit
- lock
- login
- long
- match
- membership
- merge



- message
- mode
- modify
- natural
- nchar
- new
- no
- noholdlock
- not
- notify
- null
- numeric
- nvarchar
- of
- off
- on
- open
- openstring
- openxml
- option
- options
- or
- order
- others
- out
- outer
- over
- passthrough
- precision
- prepare
- primary
- print
- privileges
- proc
- procedure
- publication
- raiserror
- readtext
- real

## SQL 语言元素

- reference
- references
- refresh
- release
- remote
- remove
- rename
- reorganize
- resource
- restore
- restrict
- return
- revoke
- right
- rollback
- rollup
- row
- rowtype
- save
- savepoint
- scroll
- select
- sensitive
- session
- set
- setuser
- share
- smallint
- some
- spatial
- sqlcode
- sqlstate
- start
- stop
- subtrans
- subtransaction
- synchronize
- table
- temporary

- then
- time
- timestamp
- tinyint
- to
- top
- tran
- treat
- trigger
- truncate
- tsequal
- unbounded
- union
- unique
- uniqueidentifier
- unknown
- unnest
- unsigned
- update
- updating
- user
- using
- validate
- values
- varbinary
- varbit
- varchar
- variable
- varray
- varying
- view
- wait
- waitfor
- when
- where
- while
- window
- with
- within

- work
- writetext
- xml

### 另请参见

- 标识符 (第 24 页)
- `quoted_identifier` 选项 (第 38 页)
- 搜索条件中的子查询 (第 43 页)
- 表达式中的列名 (第 29 页)

## 标识符

---

标识符是数据库中对象（如用户 ID、表和列）的名称。

标识符的最大长度是 128 个字节。当以下任一条件成立时，标识符必须用双引号引起来或用中括号括起来：

- 标识符包含空格。
- 标识符的首字符不是字母字符（定义将在后面提供）。
- 标识符包含保留字。
- 标识符包含字母和数字以外的其它字符。  
字母字符包括字母表中的字母，以及下划线(`_`)、at 符号(`@`)、井号(`#`)和美元符号(`$`)。数据库归类序列指出了哪些字符被视为字母字符或数字字符。

以下字符不允许用于标识符中：

- 双引号
- 控制字符（任何小于 `0x20` 的字符）
- 反斜线
- 方括号（开）
- 方括号（闭）
- 反引号/重音符

可在标识符中使用两个相连的撇号（单引号）来表示一个撇号。

如果 `QUOTED_IDENTIFIER` 数据库选项设置为 `OFF`，则双引号可用于分隔 SQL 字符串，而不能用于标识符中。但不管 `QUOTED_IDENTIFIER` 如何设置，您都可以用方括号来分隔标识符。

对于 Open Client 和 jConnect 连接，`QUOTED_IDENTIFIER` 选项的缺省设置为 `OFF`；对于其它连接，缺省设置为 `ON`。

### *限制*

标识符有以下限制：

- 表名不能包含双引号。
- 用户名不能包含双引号或分号（但允许包含单引号）。
- 数据库名称不能包含双引号字符、单引号字符和分号字符。
- 用户名和数据库名称不能以空格开头或结尾。
- 数据库空间名称始终不区分大小写，无论 **CREATE DATABASE**……**CASE IGNORE** 或 **CASE RESPECT** 规范如何。

使用 `-n start_iq` 服务器选项时存在数据库服务器命名限制。

### 示例

以下各种情况均为有效标识符。

```
Surname
"Surname"
[Surname]
SomeBigName
"Client Number"
```

### 另请参见

- 保留字（第 17 页）
- `quoted_identifier` 选项（第 38 页）
- 搜索条件中的子查询（第 43 页）
- 表达式中的列名（第 29 页）

## 数据库服务器命名限制

如果在 `start_iq [server-options]` 中使用 `-n` 开关，则会受到某些命名限制。

不对服务器名称执行字符集转换。如果客户端字符集和数据库服务器字符集不同，则在服务器名称中使用扩展字符会导致找不到该服务器。如果客户端和服务器运行在不同的操作系统或区域设置中，请在服务器名称中使用 7 位 ASCII 字符。

数据库服务器名称必须是有效的标识符。长数据库服务器名将被截断为不同的长度，具体长度取决于协议。数据库服务器名称不能：

- 以空格、单引号或双引号开头
- 以空格结尾
- 包含分号
- 超过 128 字节

服务器名称指定要用在客户端应用程序连接字符串或配置文件中的名称。建议不要使用同一名称运行多台数据库服务器。

## 字符串

---

字符串是文字字符串，或者是数据类型为 **CHAR** 或 **VARCHAR** 的表达式。

字符串是用撇号(‘单引号’)引起来的任何字符序列。属于字符数据类型的 **SQL** 变量可以包含字符串。下面是字符串的一个简单示例：

数据类型为 **CHAR** 的表达式可以是内置函数或用户定义的函数，也可以是其它多种可用表达式之一。

```
'This is a string.'
```

### 字符串中的特殊字符

在字符串中用转义序列表示特殊字符，如下所示：

- 要在字符串中表示撇号，需在一行中使用两个撇号。例如：

```
'John''s database'
```

- 要表示换行符，需使用后跟 **n** 的反斜杠 (**\n**)。例如：

```
'First line:\nSecond line:'
```

- 要表示反斜杠字符，需在一行中使用两个反斜杠 (**\\**)。例如：

```
'c:\\temp'
```

- 十六进制转义序列可用于任何字符（无论是否为可打印字符）。十六进制转义序列由反斜杠后跟一个 **x** 再跟两个十六进制数字构成（例如，**\x6d** 表示字母 **m**）。  
例如：

```
'\x00\x01\x02\x03'
```

### 兼容性

为与 **Adaptive Server® Enterprise** 兼容，可将 **QUOTED\_IDENTIFIER** 数据库选项设置为 **OFF**。在这种设置下，仍可以用双引号来标记字符串的开始和结束。缺省情况下，此选项为 **ON**。

### 另请参见

- 比较条件（第 40 页）
- 表达式（第 27 页）
- 空值（第 73 页）
- 搜索条件（第 39 页）
- 三值逻辑（第 42 页）
- **SQL** 运算符（第 30 页）
- 搜索条件中的子查询（第 43 页）

## 表达式

表达式由若干不同种类的元素（例如常量、列名、SQL 运算符和子查询）构成。

### 语法

```
expression:
case-expression
| constant
| [ correlation-name. ] column-name [ java-ref ]
| - expression
| expression operator expression
| ( expression )
| function-name ( expression, ... )
| if-expression
| [ java-package-name. ] java-class-name java-ref
| ( subquery )
| variable-name [ java-ref ]
```

### 参数

```
case-expression:
{ CASE search-condition
... WHEN expression
      THEN expression [ , ... ]
... [ ELSE expression ]
END
| CASE
... WHEN search-condition
      THEN expression [ , ... ]
... [ ELSE expression ]
END }
```

```
constant:
{ integer | number | 'string' | special-constant | host-variable }
```

```
special-constant:
{ CURRENT { DATE | TIME | TIMESTAMP | USER }
| LAST USER
| NULL
| SQLCODE
| SQLSTATE }
```

```
if-expression:
IF condition
... THEN expression
... [ ELSE expression ]
ENDIF
```

```
java-ref:
{ . field-name [ java-ref ]
| >> field-name [ java-ref ]
```

```
| . method-name ( [ expression ] [ , ... ] ) [ java-ref ]  
| >> method-name ( [ expression ] [ , ... ] ) [ java-ref ] }
```

*operator:*

```
{ + | - | * | / | || | % }
```

*用法*

任何位置。

*授权*

必须连接到数据库

*副作用*

无。

*兼容性*

- Adaptive Server 不支持 IF 条件。
- Adaptive Server 当前不支持 Java 表达式。
- 有关其它区别，请参见后续章节中对各类表达式的单独说明。

**另请参见**

- 比较条件 (第 40 页)
- 空值 (第 73 页)
- 搜索条件 (第 39 页)
- 字符串 (第 26 页)
- 三值逻辑 (第 42 页)
- SQL 运算符 (第 30 页)
- 搜索条件中的子查询 (第 43 页)
- 特殊值 (第 62 页)
- CASE 语句支持 (第 778 页)

### 表达式中的常量

常量是数字或字符串。

字符串常量用撇号引起来。在一行中使用两个撇号来表示字符串中的一个撇号。

**另请参见**

- 表达式中的列名 (第 29 页)
- 表达式中的子查询 (第 29 页)
- SQL 运算符 (第 30 页)
- IF 表达式 (第 34 页)
- CASE 表达式 (第 35 页)



- 表达式和常量的兼容性（第 36 页）

## 表达式中的列名

列名是一个跟在可选相关名后面的标识符。相关名通常是表名。

如果列名含有字母、数字和下划线以外的其它字符，则必须用引号（“”）将该名称引起来。例如，以下是有效的列名：

```
Employees.Surname
City
"StartDate"
```

### 另请参见

- 表达式中的常量（第 28 页）
- 表达式中的子查询（第 29 页）
- SQL 运算符（第 30 页）
- IF 表达式（第 34 页）
- CASE 表达式（第 35 页）
- 表达式和常量的兼容性（第 36 页）
- 搜索条件中的子查询（第 43 页）
- 保留字（第 17 页）
- 标识符（第 24 页）

## 表达式中的子查询

子查询是圆括号括起来的 **SELECT** 语句。**SELECT** 语句可包含一个，且只能包含一个选择列表项。用作表达式时，标量子查询只能返回一个值或不返回值。

在顶级 **SELECT** 的 **SELECT** 列表中，或在 **UPDATE** 语句的 **SET** 子句中，可将标量子查询用于可使用列名称的任何地方。但是，子查询不能出现在条件表达式中：

- **CASE**
- **IF**
- **NULLIF**
- **ARGN**
- **COALESCE**
- **ISNULL**

例如，以下语句将返回每个部门的雇员数（按部门名称分组）：

```
SELECT DepartmentName, COUNT(*), 'out of', (SELECT COUNT(*) FROM
Employees) FROM Departments AS D, Employees AS E WHERE D.DepartmentID
= E.DepartmentID GROUP BY DepartmentName;
```

### 另请参见

- 表达式中的常量（第 28 页）

## SQL 语言元素

- 表达式中的列名 (第 29 页)
- SQL 运算符 (第 30 页)
- IF 表达式 (第 34 页)
- CASE 表达式 (第 35 页)
- 表达式和常量的兼容性 (第 36 页)

## SQL 运算符

以下主题介绍了 SAP Sybase IQ 中可用的算术运算符、字符串运算符和逐位运算符。

正常的运算优先级顺序仍适用。括号中的表达式先进行运算；接着是乘除，然后是加减。字符串并置在加减之后进行。

### 另请参见

- 表达式中的常量 (第 28 页)
- 表达式中的列名 (第 29 页)
- 表达式中的子查询 (第 29 页)
- IF 表达式 (第 34 页)
- CASE 表达式 (第 35 页)
- 表达式和常量的兼容性 (第 36 页)
- 比较条件 (第 40 页)
- 表达式 (第 27 页)
- 空值 (第 73 页)
- 搜索条件 (第 39 页)
- 字符串 (第 26 页)
- 三值逻辑 (第 42 页)

### 算术运算符

SAP Sybase IQ 中提供以下算术运算符。

表 1. 算术运算符

运算符	描述
expression + expression	加法。如果其中任一表达式为空值，则结果为空值。
expression - expression	减法。如果其中任一表达式为空值，则结果为空值。
- expression	取非。如果表达式为空值，则结果为空值。
expression * expression	乘法。如果其中任一表达式为空值，则结果为空值。
expression / expression	除法。如果其中任一表达式为空值或第二个表达式为 0，则结果为空值。

运算符	描述
expression % expression	模运算用于计算两个整数相除后的整数余数。例如 21 % 11 = 10，因为 21 除以 11 等于 1 余 10。

### 另请参见

- 字符串运算符 (第 31 页)
- 逐位运算符 (第 32 页)
- 连接运算符 (第 34 页)
- 运算符优先级 (第 34 页)

### 字符串运算符

SAP Sybase IQ 中提供以下字符串运算符。

表 2. 字符串运算符

运算符	描述
expression    expres- sion	字符串并置 (两条竖线)。如果其中任一字符串为空值，则会在并置时将其视为空字符串。
expression + ex- pression	字符串并置运算的另一种表达方式。使用 + 并置运算符时，必须确保操作数已被显式设置为字符数据类型，而不是依赖隐式数据转换。

字符串串联运算符的结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用字符串串联运算符，则必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **LEFT** 设置为正确的数据类型和大小。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。|| 运算符是 ISO/ANSI SQL 的字符串并置运算符。
- Sybase - Adaptive Server 支持 + 运算符。

### 另请参见

- 算术运算符 (第 30 页)
- 逐位运算符 (第 32 页)
- 连接运算符 (第 34 页)
- 运算符优先级 (第 34 页)
- REVERSE 函数 [字符串] (第 289 页)

**逐位运算符**

在 SAP Sybase IQ 和 Adaptive Server 中，可以对所有无标度的整数数据类型使用以下逐位运算符。

运算符	描述
&	与
	逐位或
^	异或
~	非

**另请参见**

- 算术运算符 (第 30 页)
- 字符串运算符 (第 31 页)
- 连接运算符 (第 34 页)
- 运算符优先级 (第 34 页)

**与运算符 (&)**

与运算符会对两个位进行比较。如果两个位均为 1，则结果为 1。

位 1	位 2	位 1 & 位 2
0	0	0
0	1	0
1	0	0
1	1	1

**另请参见**

- 逐位或 (|) (第 32 页)
- 异或 (^) (第 33 页)
- 非 (~) (第 33 页)

**逐位或 (|)**

或运算符会对两个位进行比较。如果其中一位或另一位为 1，则结果为 1。

位 1	位 2	位 1   位 2
0	0	0
0	1	1

位 1	位 2	位 1   位 2
1	0	1
1	1	1

**另请参见**

- 与运算符 (&) (第 32 页)
- 异或 (^) (第 33 页)
- 非 (~) (第 33 页)

**异或 (^)**

如果两个操作数中任意一个 (但并非全部) 为 1, 异或运算符的计算结果将为 1。

位 1	位 2	位 1 ^ 位 2
0	0	0
0	1	1
1	0	1
1	1	0

**另请参见**

- 与运算符 (&) (第 32 页)
- 逐位或 (|) (第 32 页)
- 非 (~) (第 33 页)

**非 (~)**

非运算符是一元运算符, 将返回其操作数的反操作数。

位	~ 位
1	0
0	1

**另请参见**

- 与运算符 (&) (第 32 页)
- 逐位或 (|) (第 32 页)
- 异或 (^) (第 33 页)

### 连接运算符

除了使用 **FROM** 子句中的表表达式的 ISO/ANSI SQL 连接语法外，SAP Sybase IQ 还支持 Transact-SQL™ 外连接运算符 \*= 和 =\*。

#### 兼容性

- 模运算 - 对于新数据库，缺省值为 **OFF**。
- 字符串串联 - 在 SAP Sybase IQ 中使用 + 串联运算符时，需确保操作数已显式设置为字符串，而不是依赖于隐式数据转换。例如，以下查询返回整数值 579:

```
SELECT 123 + 456
```

而以下查询则返回字符串 123456:

```
SELECT '123' + '456'
```

可以使用 **CAST** 或 **CONVERT** 函数来显式转换数据类型。

---

**注意：**与 **BINARY** 或 **VARBINARY** 数据类型一起使用时，+ 运算符表示串联，而不是加法。

---

Adaptive Server 不支持 || 串联运算符。

#### 另请参见

- 算术运算符 (第 30 页)
- 字符串运算符 (第 31 页)
- 逐位运算符 (第 32 页)
- 运算符优先级 (第 34 页)

### 运算符优先级

请按照以下建议来显式设置运算顺序。

在一个表达式中使用多个运算符时，请使用括号显式设置运算顺序，而不是依赖于 Adaptive Server 和 SAP Sybase IQ 之间相同的运算符优先级。

#### 另请参见

- 算术运算符 (第 30 页)
- 字符串运算符 (第 31 页)
- 逐位运算符 (第 32 页)
- 连接运算符 (第 34 页)

## IF 表达式

IF 表达式提供 IF-THEN-ELSE SQL 表达式。

IF 表达式的语法如下:

```
IF condition  
  THEN expression1
```

```
[ ELSE expression2 ]
ENDIF
```

此表达式返回：

- 如果 *condition* 为 TRUE，则 IF 表达式返回 *expression1*。
- 如果 *condition* 为 FALSE，则 IF 表达式返回 *expression2*。
- 如果 *condition* 为 FALSE，并且没有 *expression2*，则 IF 表达式返回空值。
- 如果 *condition* 为空值，则 IF 表达式返回空值。

---

**注意：** IF 语句不同于 IF 表达式。

不要混淆 IF 表达式的语法和 IF 语句的语法。

---

### 另请参见

- 表达式中的常量（第 28 页）
- 表达式中的列名（第 29 页）
- 表达式中的子查询（第 29 页）
- SQL 运算符（第 30 页）
- CASE 表达式（第 35 页）
- 表达式和常量的兼容性（第 36 页）

## CASE 表达式

CASE 表达式提供条件 SQL 表达式。

在能够使用表达式的任何位置均可使用 CASE 表达式。CASE 表达式的语法如下：

```
CASE expression
WHEN expression THEN expression [, ...]
[ ELSE expression ] END
```

在 CASE 语句中不能将子查询用作值表达式。

如果 CASE 语句后的表达式等于 WHEN 语句后的表达式，则返回 THEN 语句后的表达式。否则，返回 ELSE 语句后的表达式（如果存在）。

例如，以下代码用 CASE 表达式作为 SELECT 语句的第二个子句。

```
SELECT ID,
       (CASE name
        WHEN 'Tee Shirt' THEN 'Shirt'
        WHEN 'Sweatshirt' THEN 'Shirt'
        WHEN 'Baseball Cap' THEN 'Hat'
        ELSE 'Unknown'
        END) as Type
FROM "GROUP0".Products
```

替代语法如下：

**CASE**

```
WHEN search-condition THEN expression [, ...]
[ ELSE expression ] END
```

如果满足 **WHEN** 语句后的搜索条件，则返回 **THEN** 语句后的表达式。否则，返回 **ELSE** 语句后的表达式（如果存在）。

例如，以下语句用 **CASE** 表达式作为 **SELECT** 语句的第三个子句，以将字符串与搜索条件相关联。

```
SELECT ID, name,
       (CASE
        WHEN name='Tee Shirt' THEN 'Sale'
        WHEN quantity >= 50 THEN 'Big Sale'
        ELSE 'Regular price'
        END) as Type
FROM "GROUPO".Products
```

**另请参见**

- 表达式中的常量（第 28 页）
- 表达式中的列名（第 29 页）
- 表达式中的子查询（第 29 页）
- SQL 运算符（第 30 页）
- IF 表达式（第 34 页）
- 表达式和常量的兼容性（第 36 页）
- NULLIF 函数 [杂项]（第 255 页）
- 用于简写 **CASE** 表达式的 NULLIF 函数（第 36 页）

**用于简写 **CASE** 表达式的 NULLIF 函数**

**NULLIF** 函数提供了一种方法来以简写形式编写某些 **CASE** 语句。

**NULLIF** 的语法如下：

```
NULLIF ( expression-1, expression-2 )
```

**NULLIF** 比较两个表达式的值。如果第一个表达式等于第二个表达式，则 **NULLIF** 返回空值。如果它们不相等，则 **NULLIF** 返回第一个表达式。

**另请参见**

- **CASE** 表达式（第 35 页）
- **NULLIF** 函数 [杂项]（第 255 页）

**表达式和常量的兼容性**

以下主题介绍了 Adaptive Server 和 SAP Sybase IQ 之间表达式和常量的兼容性。

**另请参见**

- 表达式中的常量（第 28 页）



- 表达式中的列名（第 29 页）
- 表达式中的子查询（第 29 页）
- SQL 运算符（第 30 页）
- IF 表达式（第 34 页）
- CASE 表达式（第 35 页）

### 表达式的兼容性

下表介绍了 Adaptive Server 和 SAP Sybase IQ 之间表达式的兼容性。

下表只是一个指南，**Both** 标记并不表示在任何情况、任何目的下，表达式均以相同的方式执行。有关详细说明，请参见 Adaptive Server 文档和 SAP Sybase IQ 文档中各表达式的相关内容。

在下表中，**expr** 表示表达式，**op** 表示运算符。

**表 3. Adaptive Server 和 SAP Sybase IQ 之间表达式的兼容性**

表达式	支持情况
常量	同时适用
列名称	同时适用
变量名	同时适用
函数 (expr)	同时适用
- expr	同时适用
expr op expr	同时适用
(expr)	同时适用
(子查询)	同时适用
if 表达式	仅限 SAP Sybase IQ

### 常量的兼容性

下表介绍了 Adaptive Server 和 SAP Sybase IQ 之间常量的兼容性。

下表只是一个指南，**Both** 的标记并不表示在任何情况、任何目的下，表达式均以相同的方式执行。有关详细说明，请参见 Adaptive Server 文档和 SAP Sybase IQ 文档中各表达式的相关内容。

**表 4. Adaptive Server 和 SAP Sybase IQ 之间常量的兼容性**

常量	支持情况
整数	同时适用
数字	同时适用

常量	支持情况
'字符串'	同时适用
特殊常量	同时适用
宿主变量	SAP Sybase IQ

### 分隔字符串的缺省解释

缺省情况下，Adaptive Server 和 SAP Sybase IQ 中的分隔字符串在含义上有所不同：即，用撇号（单引号）引起来的字符串和用引号（双引号）引起来的字符串。

SAP Sybase IQ 遵循 SQL92 约定：用撇号引起来的字符串是常量表达式，用引号（双引号）引起来的字符串是分隔标识符（数据库对象的名称）。Adaptive Server 则遵循如下约定：用引号引起来的字符串是常量，但缺省情况下不允许使用分隔标识符，标识符被视为字符串。

### quoted\_identifier 选项

Adaptive Server 和 SAP Sybase IQ 均提供 **quoted\_identifier** 选项，该选项允许更改对分隔字符串的解释。缺省情况下，**quoted\_identifier** 选项在 Adaptive Server 中设置为 OFF，而在 SAP Sybase IQ 中则设置为 ON。

如果 **quoted\_identifier** 选项为 Off，则不能将 SQL 保留字用作标识符。

尽管大多数 Adaptive Server 连接选项不支持 Transact-SQL SET 语句，但 **quoted\_identifier** 选项支持 SET。

在 SAP Sybase IQ 或 Adaptive Server 中，以下语句将 **quoted\_identifier** 选项设置更改为 ON：

```
SET quoted_identifier ON
```

**quoted\_identifier** 选项设置为 ON 后，Adaptive Server 允许用引号分隔表、视图和列名。在 Adaptive Server 中不能分隔其它对象名。

在 SAP Sybase IQ 或 Adaptive Server 中，以下语句将 **quoted\_identifier** 选项设置更改为 OFF：

```
SET quoted_identifier OFF
```

在 Adaptive Server 和 SAP Sybase IQ 中，只要各自 DBMS 中的 **quoted\_identifier** 选项均设置为相同的值，就可以选用 SQL92 或缺省 Transact-SQL 约定。

### 示例

如果在 **quoted\_identifier** 选项为 ON（缺省 SAP Sybase IQ 设置）时进行操作，则对于两种类型的 DBMS，以下包含 SQL 关键字 **user** 的语句均有效。

```
CREATE TABLE "user" (
    coll char(5)
);
```

```
INSERT "user" ( coll )
VALUES ( 'abcde' ) ;
```

如果在 `quoted_identifier` 选项为 OFF（缺省 Adaptive Server 设置）时进行操作，则对于两种类型的 DBMS，以下语句均有效。

```
SELECT *
FROM Employees
WHERE Surname = "Chin"
```

### 另请参见

- 保留字（第 17 页）
- 标识符（第 24 页）

## 搜索条件

条件用于从表中选择行的子集，或者在控制语句（如 **IF** 语句）中确定控制流。

SQL 条件不符合布尔逻辑，在这种逻辑中条件非真即假。在 SQL 中，每个条件的值都是 **TRUE**、**FALSE** 或 **UNKNOWN** 中的一个。这叫做三值逻辑。如果所比较的值中某一个为空值，则比较结果为 **UNKNOWN**。

当且仅当条件的结果为 **TRUE** 时，行才满足搜索条件。条件为 **UNKNOWN** 的行不满足搜索条件。

子查询构成了一类很重要的表达式，这种表达式可用在很多搜索条件中。

以下几节将论述不同类型的搜索条件。

您可以为 **WHERE** 子句、**HAVING** 子句、**CHECK** 子句、**JOIN** 子句或 **IF** 表达式指定搜索条件。

### 语法

```
{ expression compare expression
| expression compare { ANY | SOME | ALL } ( subquery )
| expression IS [ NOT ] NULL expression
| expression1 IS [ NOT ] DISTINCT FROM expression2
| expression [ NOT ] BETWEEN expression AND expression
| expression [ NOT ] LIKE expression [ ESCAPE expression ]
| expression [ NOT ] IN ( { expression | subquery |
... value-expr1 , value-expr2 [, value-expr3 ] ... } )
| column-name [ NOT ] CONTAINS ( ... word1 [ , word2, ] [ , word3 ] ... )
| CONTAINS ( column-name [ ,...], contains-query string)
| EXISTS ( subquery )
| NOT condition
| condition AND condition
| condition OR condition
| ( condition )
| ( condition , estimate )
| condition IS [ NOT ] { TRUE | FALSE | UNKNOWN }
```

参数

```
compare:  
{ = | > | < | >= | <= | <> | != | !< | !> }
```

用法  
任何位置

授权  
必须连接到数据库。

示例  
例如，下面的查询可检索年龄最大雇员的姓名和出生年份：

```
SELECT Surname, BirthDate FROM Employees WHERE BirthDate <= ALL  
(SELECT BirthDate FROM Employees);
```

为定量比较谓词提供比较值的子查询可检索多行，但只能有一列。

副作用  
无

另请参见

- 比较条件 (第 40 页)
- 表达式 (第 27 页)
- 空值 (第 73 页)
- 字符串 (第 26 页)
- 三值逻辑 (第 42 页)
- SQL 运算符 (第 30 页)
- 搜索条件中的子查询 (第 43 页)

比较条件

搜索条件中的比较条件使用比较运算符。

比较条件的语法如下：

```
expression compare expression
```

其中，*compare* 为比较运算符。下表列出了 SAP Sybase IQ 中可用的比较运算符。

运算符	描述
=	等于
>	大于
<	小于

运算符	描述
>=	大于或等于
<=	小于或等于
!=	不等于
<>	不等于
!>	不大于
!<	不小于

### 示例

例如，下面的查询可检索年龄最大雇员的姓名和出生年份：

```
SELECT Surname, BirthDate FROM Employees WHERE Surname <= ALL
(SELECT MIN(BirthDate) FROM Employees);
```

如上例所示，为定量比较谓词提供比较值的子查询可检索多行，但只能有一列。

---

**注意：** 所有字符串比较均遵循以下原则：

- 如果数据库创建为区分大小写，则区分大小写（缺省设置）
  - 如果数据库创建为忽略大小写，则不区分大小写
- 

### 兼容性

- 尾随空白 - **Adaptive Server** 在进行比较时会忽略字符数据中的任何尾随空白。在比较字符串时，**SAP Sybase IQ** 的行为由“比较字符串时忽略尾随空白”数据库创建选项控制。
- 区分大小写 - 缺省情况下，**SAP Sybase IQ** 数据库与 **Adaptive Server** 一样，均创建为区分大小写。执行比较时不仅注意所在的数据库，还注意大小写。可以在创建数据库时控制 **SAP Sybase IQ** 数据库的大小写区分。

### 另请参见

- 表达式（第 27 页）
- 空值（第 73 页）
- 搜索条件（第 39 页）
- 字符串（第 26 页）
- 三值逻辑（第 42 页）
- SQL 运算符（第 30 页）
- 搜索条件中的子查询（第 43 页）

## 三值逻辑

SQL 的 AND、OR、NOT 和 IS 逻辑运算符以三值逻辑的方式操作。

下表显示了三值逻辑。

### AND 运算符

AND	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

### OR 运算符

OR	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

### NOT 运算符

TRUE	FALSE	UNKNOWN
FALSE	TRUE	UNKNOWN

### IS 运算符

IS	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	FALSE
FALSE	FALSE	TRUE	FALSE
UNKNOWN	FALSE	FALSE	TRUE

### 另请参见

- 比较条件 (第 40 页)
- 表达式 (第 27 页)
- 空值 (第 73 页)
- 搜索条件 (第 39 页)
- 字符串 (第 26 页)
- SQL 运算符 (第 30 页)

- 搜索条件中的子查询（第 43 页）

## 搜索条件中的子查询

子查询是用括号括起来的 **SELECT** 语句。这种 **SELECT** 语句必须包含一个且只能包含一个选择列表项。

只要子查询返回的行不超过一个，就可以在比较条件（例如 >、< 或 !=）中对列和子查询进行比较。如果子查询（必须含有一列）返回一行，该行的值将与表达式进行比较。如果子查询未返回行，则其值为 **NULL**。

返回一列以及任意数量的行的子查询可用在 **IN** 条件、**ANY** 条件、**ALL** 条件或 **EXISTS** 条件中。以下几节将对这些条件进行论述。

SAP Sybase IQ 只支持在不相关子查询谓词中使用 **UNION**，而在标量值子查询或相关子查询谓词中则不支持。

不能在 **CONTAINS** 或 **LIKE** 谓词中使用子查询。

SAP Sybase IQ 不支持在一个 **OR** 子句中使用多个子查询。例如，以下查询具有两个通过 **OR** 连接的子查询：

```
CREATE VARIABLE @ln int;SELECT @ln = 1;select count(*) FROM
lineitemWHERE l_shipdate IN (select l_shipdate FROM lineitem WHERE
l_orderkey IN (2,4,6))OR l_shipdate IN (select l_shipdate FROM
lineitem WHERE l_orderkey IN (1,3,5))OR l_linenumber = @ln;
```

允许使用通过 **AND** 和 **BETWEEN** 连接的类似子查询。

### 另请参见

- 表达式中的列名（第 29 页）
- 保留字（第 17 页）
- 标识符（第 24 页）
- 比较条件（第 40 页）
- 表达式（第 27 页）
- 空值（第 73 页）
- 搜索条件（第 39 页）
- 字符串（第 26 页）
- 三值逻辑（第 42 页）

### 分离子查询谓词

SQL89 标准允许多种形式的子查询谓词。

每个子查询可以出现在带有其它谓词的 **WHERE** 或 **HAVING** 子句中，并且可以使用 **AND** 或 **OR** 运算符进行组合。SAP Sybase IQ 支持这些子查询，它们可以是相关的（包含对外部查询中出现且不能单独计算的表的引用），也可以是不相关的（不包含对远程表的引用）。

子查询谓词的形式包括：

- 非量化比较谓词:

```
<scalar-expression> <comparison-operator> <subquery>
```

比较运算符为: =、<>、>、>=、< 或 <=

非定量比较子查询只返回一个值。如果该子查询返回多个值, 则出现错误消息。此类查询也称为标量子查询谓词。

- **IN** 谓词:

```
<scalar-expression> [NOT] IN <subquery>
```

**IN** 子查询谓词返回值的列表或单个值。此类型也称为量化子查询谓词。

- 存在谓词:

```
[NOT] EXISTS <subquery>
```

**EXISTS** 谓词表示存在子查询。只有当子查询结果非空时, 表达式 **EXISTS** <subquery> 才计算为 true。**EXISTS** 谓词不将结果与外部查询块中的任何列或表达式进行比较, 它通常用于相关子查询。

- 量化比较谓词:

```
<scalar-expression> <comparison-operator> [ANY | ALL] <subquery>
```

量化比较谓词对从子查询返回的一个值或值的集合进行比较。

可运行的查询类型包括:

- 无法在 **WHERE** 或 **HAVING** 子句中垂直执行的非相关标量子查询或 **IN** 子查询的分离
- **WHERE** 或 **HAVING** 子句中相关/非相关 **EXISTS** 子查询的分离
- **WHERE** 或 **HAVING** 子句中任意相关/非相关标量子查询 (**IN** 或 **EXISTS** 子查询, 或者量化比较子查询) 的分离
- 与 **AND/OR** (结合/分离) 组合的任意非相关/相关子查询谓词, 以及简单谓词或子查询谓词
- 视图/派生表顶部的子查询谓词的结合/分离
- **UPDATE**、**DELETE** 和 **SELECT INTO** 语句中子查询谓词的分离

**SUBQUERY\_CACHING\_PREFERENCE** 选项使有经验的 DBA 可以选择要使用的子查询高速缓存方法。

示例

非相关 **EXISTS** 和 **IN** 子查询的分离:

```
SELECT COUNT(*)
FROM supplier
WHERE s_suppkey IN (SELECT MAX(l_suppkey)
                    FROM lineitem
                    GROUP BY l_linenumber)
OR EXISTS (SELECT p_brand
           FROM part
           WHERE p_brand = 'Brand#43');
```

非相关 **EXISTS** 子查询的分离:



```

SELECT COUNT(*)
FROM supplier
WHERE EXISTS (SELECT l_suppkey
              FROM lineitem
              WHERE l_suppkey = 12345)
OR EXISTS (SELECT p_brand
          FROM part
          WHERE p_brand = 'Brand#43' );

```

非相关标量或 **IN** 子查询谓词的分离:

```

SELECT COUNT(*)
FROM supplier
WHERE s_acctbal*10 > (SELECT MAX(o_totalprice)
                    FROM orders
                    WHERE o_custkey = 12345)
OR substring(s_name, 1, 6) IN (SELECT c_name
                              FROM Customers
                              WHERE c_nationkey = 10);

```

分离相关/非相关的定量比较子查询:

```

SELECT COUNT(*)
FROM lineitem
WHERE l_suppkey > ANY (SELECT MAX(s_suppkey)
                     FROM supplier
                     WHERE s_acctbal >100
                     GROUP BY s_nationkey)
OR l_partkey >= ANY (SELECT MAX(p_partkey)
                   FROM part
                   GROUP BY p_mfgr);

```

任意相关子查询谓词的分离:

```

SELECT COUNT(*)
FROM supplier S
WHERE EXISTS (SELECT l_suppkey
             FROM lineitem
             WHERE l_suppkey = S.s_suppkey)
OR EXISTS (SELECT p_brand FROM part
          WHERE p_brand = 'Brand#43'
          AND p_partkey > S.s_suppkey);

```

在支持子查询分离之前，要求用户分两个部分编写查询，然后使用 **UNION** 合并最终结果。

下面的查询说明了一个合并查询，该查询获取的结果与任意相关子查询谓词的分离示例结果相同。合并查询的性能不是最理想的，因为它扫描两次供应商表，然后将每个 **UNION** 的结果合并以返回最终结果。

```

SELECT COUNT(*)
FROM (SELECT s_suppkey FROM supplier S
      WHERE EXISTS (SELECT l_suppkey
                   FROM lineitem
                   WHERE l_suppkey = S.s_suppkey)

```

```
UNION

SELECT s_suppkey
FROM supplier S
WHERE EXISTS (SELECT p_brand
              FROM part
              WHERE p_brand = 'Brand#43'
                 AND p_partkey > S.s_suppkey) as UD;
```

### ALL 或 ANY 条件

在搜索条件的子查询中使用 **ALL** 或 **ANY** 条件。

**ALL** 条件的语法为:

```
expression compare
                ALL ( subquery )
```

其中, *compare* 为比较运算符。

**ANY** 条件的语法为:

```
expression compare
                ANY ( subquery )
```

其中, *compare* 为比较运算符。

例如, 如果 *expression* 等于子查询结果中的任何一个值, 则带有相等运算符的 **ANY** 条件为 **TRUE**; 如果表达式不是空值且不等于子查询中的任何一列, 则 **ANY** 条件为 **FALSE**:

```
expression = ANY ( subquery )
```

如果 *expression* 为空值, 则 **ANY** 条件为 **UNKNOWN**, 除非子查询的结果中没有行, 在这种情况下, 该条件始终为 **FALSE**。

您可以使用关键字 **SOME**, 而非 **ANY**。

#### *限制*

如果定量比较谓词的某一侧有多个表达式, 则系统会返回错误消息。例如:

```
Subquery allowed only one select list item
```

这种类型的查询始终可以用 **IN** 子查询或使用 **MIN** 和 **MAX** 集合函数的标量子查询表示。

#### *兼容性*

**ANY** 和 **ALL** 子查询在 Adaptive Server 和 SAP Sybase IQ 之间是兼容的。只有 SAP Sybase IQ 支持 **SOME** 作为 **ANY** 的同义词。

## BETWEEN 条件

在子查询中使用 **BETWEEN** 条件来检索范围内的值。

**BETWEEN** 条件的语法如下：

```
expr [ NOT ] BETWEEN start-expr
      AND end-expr
```

**BETWEEN** 条件的值可以是 **TRUE**、**FALSE** 或 **UNKNOWN**。在没有 **NOT** 关键字的情况下，如果 *expr* 介于 *start-expr* 和 *end-expr* 之间，则该条件的值为 **TRUE**。**NOT** 关键字会使条件的含义相反，但会使 **UNKNOWN** 保持不变。

**BETWEEN** 条件相当于两个不等式的组合：

```
expr >= start-expr
      AND expr <= end-expr
```

**BETWEEN** 谓词的形式为 “A between B and C”。“B” 或 “C” 可以为子查询，“B” 和 “C” 也可以同时为子查询。“A” 必须为值表达式或列。

*兼容性*

**BETWEEN** 条件在 SAP Sybase IQ 和 Adaptive Server 之间是兼容的。

## 带有逻辑运算符的条件

使用 **AND**、**OR** 和 **NOT** 在子查询中组合搜索条件。

使用 **AND** 组合的条件如下所示：

```
condition1
AND
condition2
```

如果两个条件都为 **TRUE**，则合并的条件为 **TRUE**。如果任何一个条件为 **FALSE**，则合并的条件为 **FALSE**。在其它情况下，合并的条件为 **UNKNOWN**。

使用 **OR** 组合的条件如下所示：

```
condition1
OR
condition2
```

如果两个条件都为 **TRUE**，则合并的条件为 **TRUE**。如果任何一个条件为 **FALSE**，则合并的条件为 **FALSE**。在其它情况下，合并的条件为 **UNKNOWN**。至于先评估哪个条件（是 *condition1* 还是 *condition2*），系统不保证任何顺序。

### 兼容性

**AND** 和 **OR** 运算符在 SAP Sybase IQ 和 Adaptive Server 之间是兼容的。

## CONTAINS 条件

在子查询中使用 **CONTAINS** 条件来定义文本匹配。

对于具有 **WD** 索引的列，**CONTAINS** 条件的语法如下所示：

```
{ column-name [ NOT ] CONTAINS ( ( word1 [ , word2 ] [ , word3 ]
... )
```

*column-name* 必须是基表中的 CHAR、VARCHAR 或 LONG VARCHAR (CLOB) 列，并且必须具有 **WD** 索引。*word1*、*word2* 和 *word3* 表达式必须是长度不超过 255 字节的字符串常量，每个表达式仅包含一个单词。该单词的长度不能超过单词列索引允许的最大单词长度。

在没有 **NOT** 关键字的情况下，如果 *column-name* 包含各个单词，则 **CONTAINS** 条件为 **TRUE**；如果 *column-name* 为空值，则该条件为 **UNKNOWN**；否则，该条件为 **FALSE**。**NOT** 关键字对这些值求反，但会使 **UNKNOWN** 保持不变。

例如，以下搜索条件：

```
varchar_col CONTAINS ('cat', 'mat')
```

在 *varchar\_col* 的值是 The cat is on the mat 时为 **TRUE**。如果 *varchar\_col* 的值为 The cat chased the mouse，则此条件为 **FALSE**。

当 SAP Sybase IQ 执行同时包含 **LIKE** 和 **CONTAINS** 的语句时，**CONTAINS** 条件将优先执行。

不要在具有用户定义的函数的视图中使用 **CONTAINS** 谓词，因为 **CONTAINS** 条件将被忽略。可用带有通配符的 **LIKE** 谓词代替，或者在视图外发出查询。

有关使用具有 **TEXT** 索引的 **CONTAINS** 条件的信息，请参见非结构化数据分析。

## EXISTS 条件

如果子查询至少返回一行，则满足 **EXISTS** 条件。

**EXISTS** 条件的语法如下：

```
EXISTS ( subquery )
```

如果子查询结果至少包含一行，则 **EXISTS** 条件为 **TRUE**；如果子查询结果一行也不包含，则条件为 **FALSE**。**EXISTS** 条件不能为 **UNKNOWN**。

### 兼容性

**EXISTS** 条件在 Adaptive Server 和 SAP Sybase IQ 之间是兼容的。

## IN 条件

在子查询中使用 **IN** 条件来减少使用多个 **OR** 条件的需求。

**IN** 条件的语法为：

```
{ expression [ NOT ] IN ( subquery )
| expression [ NOT ] IN ( expression )
| expression [ NOT ] IN ( value-expr1 , value-expr2
[ , value-expr3 ] ... ) }
```

在没有 **NOT** 关键字的情况下，如果 *expression* 等于列出的任意值，则 **IN** 条件为 **TRUE**；如果 *expression* 为空值，则该条件为 **UNKNOWN**；否则，该条件为 **FALSE**。**NOT** 关键字会使条件的含义相反，但会使 **UNKNOWN** 保持不变。

**IN** 条件列表中允许的最大值数为 250,000。

*兼容性*

**IN** 条件在 Adaptive Server 和 SAP Sybase IQ 之间是兼容的。

## IS DISTINCT FROM 搜索条件

将 **IS DISTINCT FROM** 和 **IS NOT DISTINCT FROM** 搜索条件用作比较运算符。

*语法*

```
expression1 IS [ NOT ] DISTINCT FROM expression2
```

*注释*

**IS DISTINCT FROM** 和 **IS NOT DISTINCT FROM** 搜索条件为可优化搜索，值可以是 **TRUE** 或 **FALSE**。

如果 *expression1* 等于 *expression2*，或者两个表达式均为 **NULL**，则 **IS NOT DISTINCT FROM** 搜索条件的值为 **TRUE**。这相当于两个搜索条件的组合，如下所示：

```
expression1 = expression2 OR ( expression1 IS NULL AND expression2 IS NULL )
```

**IS DISTINCT FROM** 语法的含义相反。即，如果 *expression1* 不等于 *expression2*，并且至少其中一个表达式不为 **NULL**，则 **IS DISTINCT FROM** 的值为 **TRUE**。这等效于以下语句：

```
NOT( expression1 = expression2 OR ( expression1 IS NULL AND expression2 IS NULL ) )
```

*标准和兼容性*

- **SQL/2008** – **IS [NOT] DISTINCT FROM** 谓词在 SQL/2008 标准中定义。**IS DISTINCT FROM** 谓词是 SQL/2008 标准的功能 T151，“**DISTINCT** 谓词”。**IS NOT DISTINCT FROM** 谓词是 SQL/2008 标准的功能 T152，“**取非的 DISTINCT** 谓词”。

## IS NULL 条件

在子查询中使用 **IS NULL** 条件，以便在表达式为空值时表示缺少未知的数据。

**IS NULL** 条件的语法为：

```
expression
      IS [ NOT ] NULL
```

在没有 **NOT** 关键字的情况下，如果表达式为空值，则 **IS NULL** 条件为 **TRUE**，否则为 **FALSE**。**NOT** 关键字使条件的含义相反。

*兼容性*

**IS NULL** 条件在 Adaptive Server 和 SAP Sybase IQ 之间是兼容的。

## LIKE 条件

在子查询中使用 **LIKE** 条件，以在 **WHERE** 子句中使用通配符来执行模式匹配。

**LIKE** 条件的语法为：

```
expression [ NOT ] LIKE pattern [ ESCAPE escape-expr ]
```

**LIKE** 条件的值可以是 **TRUE**、**FALSE** 或 **UNKNOWN**。只能对字符串数据使用 **LIKE**。

不能在 **LIKE** 谓词中使用子查询。

**LIKE** 谓词以非通配符字符开头，如果有 **HG** 或 **LF** 索引，则该谓词可能会执行得更快。

在以下条件下，特定 **LIKE** 谓词的执行速度会更快：

- 可以使用 **WD** 索引，前提条件是 **LIKE** 模式在空格左侧或该模式末尾包含至少一个词，或
- 可以使用 **NGRAM TEXT** 索引，前提条件是 **LIKE** 模式包含至少 *N* 个连续的非通配符

在没有 **NOT** 关键字的情况下，如果 *expression* 与 *pattern* 匹配，则该条件的值为 **TRUE**。如果 *expression* 或 *pattern* 为空值，则该条件为 **UNKNOWN**。**NOT** 关键字会使条件的含义相反，但会使 **UNKNOWN** 保持不变。

模式可包含任意数量的通配符。通配符如下：

通配符	匹配项
_ (下划线)	任意一个字符
% (百分号)	包含零个或多个字符的任意字符串
[ ]	指定范围或集合中的任意单个字符
[^]	不在指定范围或集合中的任意单个字符



在这种情况下，不折叠任何字符。字符串长度仍为 130 个字符，于是 SAP Sybase IQ 会生成错误。

*长度介于 127 和 254 个字符之间的模式：示例 3*

在此示例中，四个相邻下划线 \_\_\_\_\_（特殊字符）替换了 130 个字符的 **LIKE** 谓词末尾的四个常量字符 %%%%：

```
select col2 from tablen where col2 like
'1234567890123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890123456_____';
```

SAP Sybase IQ 不折叠相邻特殊字符。字符串长度仍为 130 个字符，于是 SAP Sybase IQ 会生成错误。

*长度介于 127 和 254 个字符之间的模式：示例 4*

在此示例中，范围 [1-3] 替换了 130 个字符的 **LIKE** 谓词末尾的四个常量字符 %%%%：

```
select col2 from tablen where col2 like
'1234567890123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890123456[1-3]';
```

**LIKE** 谓词的长度（以字节为单位）计算方式如下：126（常量字符）\* 2 + 1（方括号中的 1）+ 1（方括号中的 3）+ 2（Set 语句和 Range 语句表达式）。

计算结果为 256 字节，因此 SAP Sybase IQ 会生成错误。

*搜索字符集中的某个字符*

可以通过将字符列在方括号中来指定要查找的一组字符。例如，以下条件将查找到字符串 *smith* 和 *smyth*：

```
LIKE 'sm[iy]th'
```

*搜索字符范围中的某个字符*

通过将字符范围的两端列在方括号中（用连字符分隔），可以指定要查找的字符范围。例如，以下条件将查找到字符串 *bough* 和 *rough*，但不能查找到 *tough*：

```
LIKE '[a-r]ough'
```

字符范围 [a-z] 被解释为“大于等于 a 且小于等于 z”，其中大于和小于运算在数据库归类中执行。有关归类中字符排序的信息，请参见管理：全球化中的归类序列如何排序字符。

范围的下端必须位于上端之前。例如，包含表达式 [z-a] 的 **LIKE** 条件不返回任何行，因为没有字符与 [z-a] 范围匹配。

除非数据库创建为区分大小写，否则字符范围不区分大小写。例如，以下条件将查找到字符串 *Bough*、*rough* 和 *TOUGH*：

```
LIKE '[a-z]ough'
```



如果数据库创建为区分大小写的数据库，则搜索条件也区分大小写。

#### 将范围搜索和集合搜索组合起来

可以在方括号中将范围和集合组合起来。例如，以下条件将查找到字符串 *bough*、*rough* 和 *tough*：

```
LIKE '[a-rt]ough'
```

中括号 *[a-mpqs-z]* 解释为“正好一个字符，该字符处于 *a* 到 *m*（含这两者）的范围中，或者是 *p*，或者是 *q*，或者处于 *s* 到 *z*（含这两者）的范围中”。

#### 搜索不在一定范围内的某个字符

使用脱字符 (^) 可指定排除在搜索之外的字符范围。例如，以下条件将查找到字符串 *tough*，但不能查找到字符串 *rough* 和 *bough*：

```
LIKE '[^a-r]ough'
```

脱字符否定括号的所有内容。例如，中括号 *[^a-mpqs-z]* 解释为“正好一个字符，该字符不在 *a* 到 *m*（含这两者）的范围中，不是 *p*，不是 *q*，并且不在 *s* 到 *z*（含这两者）的范围中”。

#### 范围和集合的特殊情况

中括号中的任何单个字符都表示该字符。例如，*[a]* 仅与字符 *a* 匹配。*[^]* 仅与脱字符匹配，*[%]* 仅与百分号字符匹配（在此环境中，百分号字符不充当通配符），*[\_]* 仅与下划线字符匹配。同样，*[/]* 仅与字符 */* 匹配。

其它特殊情况包括：

- 表达式 *[a-]* 与字符 *a* 或 *-* 匹配。
- 表达式 *[]* 永远没有匹配项，而且始终不返回任何行。
- 表达式 *[/* 或 *[abp-q]* 是非法格式表达式，系统将显示语法错误。
- 不能在中括号内使用通配符。表达式 *[a%b]* 会查找 *a*、*%* 或 *b* 之一。
- 除非作为方括号内的第一个字符，否则脱字符不能用于对一个范围取非。表达式 *[a^b]* 会查找 *a*、*^* 或 *b* 之一。

#### 兼容性

只有 SAP Sybase IQ 支持 **ESCAPE** 子句。

---

**注意：** 有关对大对象数据和变量的 **LIKE** 谓词的支持信息，请参见非结构化数据分析中的非结构化数据查询。

用户必须经过专门授权才能使用大对象数据类型 **LONG BINARY** 和 **LONG VARCHAR**。有关非结构化数据分析选项的详细信息，请参见非结构化数据分析。

---

#### 另请参见

- **PATINDEX** 函数 [字符串]（第 259 页）
- **LOCATE** 函数 [字符串]（第 234 页）

## NOT 条件

NOT 条件的值可以是 TRUE、FALSE 或 UNKNOWN。

NOT 条件的语法为：

```
NOT condition1
```

如果 *condition1* 为 FALSE，则 NOT 条件为 TRUE；如果 *condition1* 为 TRUE，则该条件为 FALSE；如果 *condition1* 为 UNKNOWN，则该条件为 UNKNOWN。

## 真值条件

条件的真值为 TRUE 或 FALSE。

真值条件的语法如下：

```
IS [ NOT ] truth-value
```

在没有 NOT 关键字的情况下，如果 *condition* 的值为提供的 *truth-value*（必须是 TRUE、FALSE 或 UNKNOWN 之一），则该条件为 TRUE。否则，值为 FALSE。NOT 关键字会使条件的含义相反，但会使 UNKNOWN 保持不变。

*兼容性*

只有 SAP Sybase IQ 支持真值条件。

## 用户提供的条件提示

条件的选择性是表的行中满足该条件的部分。

SAP Sybase IQ 查询优化程序根据来自可用索引的信息选择用于执行查询的适当策略。对于查询中的每个条件，优化程序将决定是否可以使用索引执行该条件，如果可以，优化程序会选择使用的索引以及相对于该表中其它条件的执行顺序。在这些决策中最重要因素是条件的选择；即，表的行中满足该条件的部分。

优化程序通常在没有用户干预的情况下做出决定，并且一般可提供最佳决策。但在某些情况下，优化程序可能无法在条件执行之前准确确定其选择性。通常只有以下两种条件之一成立时，才会出现这些情况：一种是条件所处的列没有适当的可用索引，另一种是条件涉及某些算术或函数表达式，因而过于复杂，使优化程序无法准确估计。

如果有经常运行的查询，您可能希望进行试验以确定能否为优化程序提供其它信息以帮助其选择最佳执行策略，从而提高该查询的性能。

### 用户提供的条件选择性

条件提示的最简单形式是提供一个选择性值，该值将用于替代优化程序计算的值。

选择性提示是在查询文本中通过为条件加上括号提供的。随后可在括号内的条件后面添加一个逗号和一个要用作选择性的数字值。

此选择性值表示为表中满足条件的行的百分比。因此，选择性可能具有的数字值的范围从 100.0 到 0.0。

**注意：**在查询计划中，选择性表示为小数而不是百分比；因此用户提供的选择性 35.5 在该查询的计划中显示为选择性 0.355000。

### 示例

- 以下查询估计 1.5% 的 ship\_date 值早于 1994/06/30:

```
SELECT ShipDate
FROM SalesOrderItems
WHERE ( ShipDate < '2001/06/30', 1.5 )
ORDER BY ShipDate DESC
```

- 以下查询估计 0.5% 的行将满足条件:

```
SELECT *
FROM Customers c, SalesOrders o
WHERE (o.SalesRepresentative > 1000.0, 0.5)
AND c.ID = o.customerID
```

使用小数百分比可以指定更精确的用户估计值，这对于大表尤为重要。

### 兼容性

SQL Anywhere 支持用户提供的选择性估计。

Adaptive Server 不支持用户提供的选择性估计。

### 另请参见

- 用户提供的条件提示字符串 (第 55 页)
- 用户提供的关于连接等同性条件的提示 (第 60 页)
- 用户提供的条件提示的使用准则 (第 62 页)
- 选择性提示 (第 56 页)

### 用户提供的条件提示字符串

您可以通过条件提示字符串为优化程序提供其它提示信息。

使用这些针对各个条件的提示字符串，用户可以为优化程序关注的条件指定可能的附加执行首选项。这些首选项包括要用于条件的索引、条件的选择性、执行条件时的执行阶段，以及条件的有用性（对该条件在某一执行阶段内的一组执行条件中的排列顺序产生影响）。

与用户提供的选择性估计一样，条件提示字符串也是在查询文本中通过为条件加上括号提供的。随后可在括号内的条件后面添加一个逗号并提供包含所需提示的带引号的字符串。在带引号的字符串中，每个提示显示为一个提示类型标识符，后跟一个冒号和该提示类型的值。在同一提示字符串中，多个提示用逗号分隔，并且可以按任意顺序显示。在一个提示字符串中的任意两个元素之间可以使用空格。

**另请参见**

- 用户提供的条件选择性 (第 54 页)
- 用户提供的关于连接等同性条件的提示 (第 60 页)
- 用户提供的条件提示的使用准则 (第 62 页)

选择性提示

可以在提示字符串中出现的第一种提示类型为选择性提示。选择性提示以提示类型标识符“S”或“s”标识。

与用户提供的选择性估计一样，选择性值始终表示为表中满足条件的行的百分比。

*示例*

以下示例与第二个用户提供的条件选择性示例完全等效。

```
SELECT *
FROM Customers c, SalesOrders o
WHERE (o.SalesRepresentative > 1000.0, 's: 0.5)
      AND c.ID = o.CustomerID
```

**另请参见**

- 索引首选项提示 (第 56 页)
- INDEX\_PREFERENCE 选项 (第 57 页)
- 执行阶段提示 (第 58 页)
- 有用性提示 (第 59 页)
- 用户提供的条件选择性 (第 54 页)

索引首选项提示

第二个支持的提示类型为索引首选项提示，以提示类型标识符“I”或“i”标识。

索引首选项提示的值可以是介于 -10 和 10 之间的任意整数。每个正整数值的含义是选用某个特定索引类型，而负值则表示要避免选用该特定索引类型。

索引首选项提示的作用与 INDEX\_PREFERENCE 选项相同，区别在于首选项仅适用于与之关联的条件，而不是查询中的所有条件。只有在以下条件成立时索引首选项才能对条件的执行产生影响，条件为：该列中存在指定的索引类型，并且评估关联条件时该索引类型可用；并非所有索引类型对于所有条件均可用。

*示例*

以下示例指定 3% 的选择性，并指示应尽可能使用 HG 索引评估条件：

```
SELECT *
FROM Customers c, SalesOrders o
WHERE (o.SalesRepresentative > 1000.0, 'S:3.00, I:+2')
      AND c.ID = o.CustomerID
```

下一示例指定 37.5% 的选择性，并指示应尽可能避免使用 HG 索引评估条件：

```
SELECT *
FROM Customers c, SalesOrders o
```

```
WHERE (o.SalesRepresentative > 1000.0, 'i:-2, s:37.500')
      AND c.ID = o.CustomerID
```

### 另请参见

- 选择性提示 (第 56 页)
- INDEX\_PREFERENCE 选项 (第 57 页)
- 执行阶段提示 (第 58 页)
- 有用性提示 (第 59 页)

### INDEX\_PREFERENCE 选项

控制对用于查询的索引的选择。

#### 允许值

值	操作
0	让优化程序选择
1	首选 LF 索引
2	首选 HG 索引
3	首选 HNG 索引
4	首选 CMP 索引
5	首选缺省索引
6	首选 WD 索引
8	首选 DATE 索引
9	首选 TIME 索引
10	首选 DTTM 索引
-1	避免 LF 索引
-2	避免 HG 索引
-3	避免 HNG 索引
-4	避免 CMP 索引
-5	避免缺省索引
-6	避免 WD 索引
-8	避免 DATE 索引

值	操作
-9	避免 TIME 索引
-10	避免 DTTM 索引

*默认值*

0

*范围*

可在数据库 (PUBLIC) 或用户级别设置选项。在数据库级别进行设置时，值将变为任何新用户的缺省值，但不会对现有用户产生任何影响。在用户级别进行设置时，仅替换该用户的 PUBLIC 值。为自身设置选项无需任何系统特权。在数据库或用户级别为任何其他用户设置选项都需要系统特权。

必须具有 SET ANY PUBLIC OPTION 系统特权才能设置此选项。可针对个别连接或 PUBLIC 角色进行临时设置。设置立即生效。

*注释*

SAP Sybase IQ 优化程序一般会选择现有的最佳索引来处理局部 WHERE 子句谓词和其它能够在 IQ 索引内完成的操作。INDEX\_PREFERENCE 用来覆盖用于测试的优化程序选项；在大多数情况下，不应更改它。

**另请参见**

- 选择性提示 (第 56 页)
- 索引首选项提示 (第 56 页)
- 执行阶段提示 (第 58 页)
- 有用性提示 (第 59 页)

*执行阶段提示*

第三种支持的提示类型为执行阶段提示，以提示类型标识符“E”或“e”标识。

在 SAP Sybase IQ 查询引擎中，有几个可以评估条件的不同执行阶段：

- 不变
- 延迟
- 绑定
- 水平

缺省情况下，优化程序会选择对评估条件所需信息全部可用的最早执行阶段内的每个条件进行评估。因此，每个条件都有一个进行评估的缺省执行阶段。

由于在所需信息可用之前无法评估条件，因此执行阶段提示只能用于将条件的执行延迟到其缺省阶段之后的某个阶段，而不能用于在早于条件缺省阶段的任意阶段内强制评估该条件。

四个条件执行阶段（从最早到最晚）如下：

- **不变** - 只引用一列（或同一个表中的两列）且可以使用索引进行评估的条件一般称为简单不变条件。简单不变条件通常在优化过程的较早阶段进行评估。这意味着，满足所有这种不变条件的行数可用于指导优化程序选择要使用的最佳连接顺序和连接算法。由于这是最早的执行阶段，因此用户无法强制在此阶段输入条件，但可以使此阶段中的条件进入后面的阶段。
- **延迟** - 有些条件只有在查询的一些其它部分执行之后才能进行评估。这些延迟条件将在其附加到的查询节点首次读取时进行一次评估。这些条件分为两类：不相关子查询条件和优化程序创建的 **IN** 或 **PROBABLY\_IN** 下推连接条件。
- **绑定** - 有些条件必须进行多次评估。这些条件一般分为两类：相关子查询中包含外部引用的条件，以及优化程序创建的下推等值连接条件。例如，在查询执行过程中，外部引用条件会在每次外部引用值发生更改时重新进行评估。
- **水平** - 有些条件（如包含同一表中两个以上列的条件）必须一次评估一行，而不能使用索引进行评估。

执行阶段提示将接受标识用户希望评估条件的执行阶段的值。每个值都是不区分大小写的单个字符：

- **D** - 延迟
- **B** - 绑定
- **H** - 水平

#### 示例

以下示例显示的条件提示字符串指示应将条件移入“延迟”执行阶段，并且指示应尽可能使用 **LF** 索引评估该条件：

```
SELECT *
FROM Customers c, SalesOrders o
WHERE (o.SalesRepresentative > 10000.0, 'E:D, I:1')
      AND c.id = o.CustomerID
```

#### 另请参见

- 选择性提示（第 56 页）
- 索引首选项提示（第 56 页）
- **INDEX\_PREFERENCE** 选项（第 57 页）
- 有用性提示（第 59 页）

#### 有用性提示

最后一种支持的提示类型为有用性提示，以提示类型标识符“**U**”或“**u**”标识。

有用性提示的值可以是介于 **0.0** 和 **10.0** 之间的任意数字值。在优化程序中，会针对每个条件计算有用性值，随后使用该值确定同一执行阶段内要评估的一组条件的评估顺序。有用性值越大，在评估顺序中出现的越早。提供有用性提示使用户可以将条件放在评估顺序中的特定位置，但无法更改在其中评估条件的执行阶段。

*示例*

以下示例显示的条件提示字符串指示应将条件移入“延迟”执行阶段，并且指明应在“延迟”阶段内将其有用性设置为 3.25。

```
SELECT *
FROM Customers c, SalesOrders o
WHERE (co.SalesRepresentative > 10000.0, 'U: 3.25, E: D')
AND c.id = o.CustomerID
```

*兼容性*

SQL Anywhere 不支持用户提供的条件提示字符串。

Adaptive Server 不支持用户提供的条件提示字符串。

**另请参见**

- 选择性提示（第 56 页）
- 索引首选项提示（第 56 页）
- INDEX\_PREFERENCE 选项（第 57 页）
- 执行阶段提示（第 58 页）

**用户提供的关于连接等同性条件的提示**

用户可以指定不影响查询中的每个连接的连接算法首选项。

简单的等同性连接谓词可使用谓词提示标记，该提示支持仅针对该连接指定连接首选项。如果同一连接具有多个带有本地连接首选项的连接条件，并且如果这些提示不是相同值，那么，将忽略该连接的所有本地首选项。本地连接首选项不会影响优化程序选择的连接顺序。

*示例*

以下示例请求散列连接：

```
AND (T.X = 10 * R.x, 'J:4')
```

**表 5. JOIN\_PREFERENCE 值**

值	操作
0	让优化程序选择
1	首选排序合并
2	首选嵌套循环
3	首选嵌套循环叠加
4	首选散列
5	首选散列叠加



值	操作
6	首选非对称排序合并连接
7	首选排序合并叠加
8	首选非对称排序合并叠加连接
9	如果连接键包含散列分区表的所有分区键，则首选分区散列连接
10	如果连接键包含散列分区表的所有分区键，则首选分区散列叠加连接
11	如果连接键包含散列分区表的所有分区键，则首选分区排序合并连接
12	如果连接键包含散列分区表的所有分区键，则首选分区排序合并叠加连接
-1	避免排序合并
-2	避免嵌套循环
-3	避免嵌套循环叠加
-4	避免散列
-5	避免散列叠加
-6	避免非对称排序合并连接
-7	避免排序合并叠加
-8	避免非对称排序合并叠加连接
-9	如果连接键包含散列分区表的所有分区键，则避免分区散列连接
10	如果连接键包含散列分区表的所有分区键，则避免分区散列叠加连接
11	如果连接键包含散列分区表的所有分区键，则避免分区排序合并连接
12	如果连接键包含散列分区表的所有分区键，则避免分区排序合并叠加连接

### 有关详细信息

参考：语句和选项 > 数据库选项 > 按字母顺序排列的选项列表 > JOIN\_PREFERENCE 选项

### 另请参见

- 用户提供的条件选择性（第 54 页）
- 用户提供的条件提示字符串（第 55 页）
- 用户提供的条件提示的使用准则（第 62 页）

### 用户提供的条件提示的使用准则

一般来讲，条件提示仅适用于经常运行的查询。

只有高级用户才能尝试使用条件提示。优化程序通常会做出最佳决策，但它无法根据可用索引推断有关条件的准确信息。

优化程序经常重写或简化原始条件，还会根据原始条件推断新条件。条件提示不会从新条件一直应用到优化程序推断的条件，也不会应用到简化条件。

#### 另请参见

- 用户提供的条件选择性（第 54 页）
- 用户提供的条件提示字符串（第 55 页）
- 用户提供的关于连接等同性条件的提示（第 60 页）

## 特殊值

---

特殊值可用在表达式中，也可在创建表时用作列缺省值。

#### 另请参见

- 表达式（第 27 页）

## CURRENT DATABASE 特殊值

**CURRENT DATABASE** 返回当前数据库的名称。

*数据类型*

STRING

## CURRENT DATE 特殊值

**CURRENT DATE** 返回当前年月日。

*数据类型*

DATE

#### 另请参见

- **TIMESTAMP** 特殊值（第 65 页）
- **CURRENT TIMESTAMP** 特殊值（第 63 页）
- **CURRENT TIME** 特殊值（第 63 页）
- 日期和时间数据类型（第 358 页）
- 检索日期和时间（第 360 页）

## **CURRENT PUBLISHER 特殊值**

**CURRENT PUBLISHER** 返回一个字符串，其中包含 SQL Remote 复制数据库的发布者用户 ID。

*数据类型*

STRING

**CURRENT PUBLISHER** 可在具有字符数据类型的列中用作缺省值。

## **CURRENT TIME 特殊值**

**CURRENT TIME** 返回当前小时、分钟、秒和秒的小数部分。

*数据类型*

TIME

*描述*

秒的小数值存储至 6 个小数位，但当前时间的精确性受系统时钟的精确性限制。

**另请参见**

- **TIMESTAMP** 特殊值（第 65 页）
- **CURRENT TIMESTAMP** 特殊值（第 63 页）
- **CURRENT DATE** 特殊值（第 62 页）
- 日期和时间数据类型（第 358 页）
- 检索日期和时间（第 360 页）

## **CURRENT TIMESTAMP 特殊值**

组合 **CURRENT DATE** 和 **CURRENT TIME**，以构成包含年、月、日、小时、分钟、秒和秒的小数部分的 **TIMESTAMP** 值。

与 **CURRENT TIME** 一样，秒的小数部分的精确性受系统时钟的限制。

**CURRENT TIMESTAMP** 缺省为 3 位数。

*数据类型*

TIMESTAMP

**另请参见**

- **TIMESTAMP** 特殊值（第 65 页）
- **CURRENT TIME** 特殊值（第 63 页）
- **CURRENT DATE** 特殊值（第 62 页）
- 日期和时间数据类型（第 358 页）
- 检索日期和时间（第 360 页）

- **CURRENT USER** 特殊值 (第 64 页)
- **LAST USER** 特殊值 (第 64 页)
- **USER** 特殊值 (第 65 页)

### **CURRENT USER** 特殊值

**CURRENT USER** 返回一个字符串，其中包含当前连接的用户 ID。

在执行 **UPDATE** 操作时，系统不会更改缺省值为 **CURRENT USER** 的列。

#### *数据类型*

STRING

**CURRENT USER** 可在具有字符数据类型的列中用作缺省值。

#### 另请参见

- **CURRENT TIMESTAMP** 特殊值 (第 63 页)
- **LAST USER** 特殊值 (第 64 页)
- **USER** 特殊值 (第 65 页)

### **LAST USER** 特殊值

**LAST USER** 返回上次修改行的用户的名称。

在执行 **INSERT** 和 **LOAD** 操作时，此常量与 **CURRENT USER** 的效果相同。在执行 **UPDATE** 操作时，如果未显示修改缺省值为 **LAST USER** 的列，则该列将更改为当前用户的名称。

与 **DEFAULT TIMESTAMP** 组合时，**LAST USER** 缺省值可用于记录 (在单独的列中) 用户和上次更改行的日期和时间。

#### *数据类型*

STRING

**LAST USER** 可在具有字符数据类型的列中用作缺省值。

#### 另请参见

- **CURRENT USER** 特殊值 (第 64 页)
- **CURRENT TIMESTAMP** 特殊值 (第 63 页)
- **USER** 特殊值 (第 65 页)

### **SQLCODE** 特殊值

**SQLCODE** 返回当前的 **SQLCODE** 值。

**SQLCODE** 值在每个语句后进行设置。可以检查 **SQLCODE** 以查看语句是否成功。

*数据类型*

STRING

## **SQLSTATE 特殊值**

**SQLSTATE** 返回当前的 **SQLSTATE** 值。

**SQLSTATE** 值在每个语句后进行设置。可以检查 **SQLSTATE** 以查看语句是否成功。

*数据类型*

STRING

## **TIMESTAMP 特殊值**

**TIMESTAMP** 指示表中各行上次修改的时间。

当使用 **DEFAULT TIMESTAMP** 声明列时，系统将为插入和装载操作提供缺省值。只要更新此行，便会用当前日期和时间更新该值。

在执行 **INSERT** 和 **LOAD** 操作时，**DEFAULT TIMESTAMP** 与 **CURRENT TIMESTAMP** 的效果相同。在执行 **UPDATE** 操作时，如果未显示修改缺省值为 **TIMESTAMP** 的列，则该列的值将更改为当前日期和时间。

---

**注意：** SAP Sybase IQ 不支持 **UTC TIMESTAMP** 或 **CURRENT UTC TIMESTAMP** 的 **DEFAULT** 值，也不支持数据库选项 **DEFAULT\_TIMESTAMP\_INCREMENT**。SAP Sybase IQ 在每次尝试插入或更新类型为 **UTC TIMESTAMP** 或 **CURRENT UTC TIMESTAMP** 的列 **DEFAULT** 值时产生错误。

---

*数据类型*

TIMESTAMP

另请参见

- **CURRENT TIMESTAMP** 特殊值（第 63 页）
- **CURRENT TIME** 特殊值（第 63 页）
- **CURRENT DATE** 特殊值（第 62 页）
- 日期和时间数据类型（第 358 页）
- 检索日期和时间（第 360 页）

## **USER 特殊值**

**USER** 返回一个字符串，其中包含当前连接的用户 ID。

在执行 **UPDATE** 操作时，系统不会更改缺省值为 **USER** 的列。

*数据类型*

STRING

**USER** 可在具有字符数据类型的列中用作缺省值。

### 另请参见

- **CURRENT USER** 特殊值 (第 64 页)
- **CURRENT TIMESTAMP** 特殊值 (第 63 页)
- **LAST USER** 特殊值 (第 64 页)

## 变量

---

SAP Sybase IQ 支持局部变量、连接级变量和全局变量。

所有全局变量的名称都以两个 @ 符号开头。例如，全局变量 @@version 的值是数据库服务器的当前版本号。用户不能定义全局变量。

## 局部变量

局部变量由用户声明，并且可以在 SQL 语句的过程或批处理中用于保存信息。

局部变量是使用 **DECLARE** 语句声明的，只能在复合语句（即，两端带有 **BEGIN** 和 **END** 关键字的语句）中使用。这类变量最初设置为 **NULL**。您可以使用 **SET** 语句设置这类变量的值，也可以使用带有 **INTO** 子句的 **SELECT** 语句为其赋值。

**DECLARE** 语句的语法如下：

```
DECLARE variable-name data-type
```

只要是从复合语句内调用过程，就可以将局部变量作为参数传递给该过程。

### 示例

- 下面的批处理说明了如何使用局部变量：

```
BEGIN
    DECLARE local_var INT ;
    SET local_var = 10 ;
    MESSAGE 'local_var = ', local_var ;
END
```

从 **ISQL** 运行此批处理会在服务器窗口中显示以下消息：

```
local_var = 10
```

- 变量 `local_var` 在声明它的复合语句外部不存在。以下批处理语句无效，将显示未找到列的错误：

```
-- This batch is invalid.
BEGIN
    DECLARE local_var INT ;
    SET local_var = 10 ;
    MESSAGE 'local_var = ', local_var ;
END;
MESSAGE 'local_var = ', local_var ;
```

- 以下示例说明了如何使用带有 **INTO** 子句的 **SELECT** 来设置局部变量的值：

```
BEGIN
  DECLARE local_var INT ;
  SELECT 10 INTO local_var ;
  MESSAGE 'local_var = ', local_var ;
END
```

从 **ISQL** 运行此批处理会在服务器窗口中显示以下消息：

```
local_var = 10
```

### 兼容性

名称 - **Adaptive Server** 和 **SAP Sybase IQ** 均支持局部变量。在 **Adaptive Server** 中，所有变量都必须以符号 **@** 为前缀。在 **SAP Sybase IQ** 中，**@** 前缀是可选的。要编写兼容的 **SQL**，需确保所有变量均带有 **@** 前缀。

范围 - **SAP Sybase IQ** 和 **Adaptive Server** 之间的局部变量范围是不同的。**SAP Sybase IQ** 支持使用 **DECLARE** 语句在批处理内声明局部变量。但是，如果在复合语句中执行 **DECLARE**，则范围将限于该复合语句。

声明 - 在 **SAP Sybase IQ** 中，每个 **DECLARE** 语句只能声明一个变量。而在 **Adaptive Server** 中，一个语句可以声明多个变量。

## 连接级变量

连接级变量由用户声明，并且可以在 **SQL** 语句的过程或批处理中用于保存信息。

连接级变量是使用 **CREATE VARIABLE** 语句声明的。**CREATE VARIABLE** 语句可以在复合语句以外的任何位置使用。连接级变量可作为参数传递给过程。

**CREATE VARIABLE** 的语法为：

```
CREATE VARIABLE variable-name data-type
```

变量创建时最初设置为 **NULL**。您可以使用 **SET** 语句或带有 **INTO** 子句的 **SELECT** 语句，通过与局部变量相同的方式来设置连接级变量的值。

连接级变量会一直存在，直到连接终止，或者直到使用 **DROP VARIABLE** 语句显式删除该变量为止。以下语句会删除变量 *con\_var*：

```
DROP VARIABLE con_var
```

### 示例

- 下面的 **SQL** 批处理语句说明如何使用连接级变量。

```
CREATE VARIABLE con_var INT;
SET con_var = 10;
MESSAGE 'con_var = ', con_var;
```

从 **ISQL** 运行此批处理会在服务器窗口中显示以下消息：

```
con_var = 10
```

### 兼容性

Adaptive Server 不支持连接级变量。

## 全局变量

全局变量是系统提供的变量，用于提供系统提供的值。

SAP Sybase IQ 会设置全局变量的值。例如，全局变量 `@@version` 的值是数据库服务器的当前版本号。

全局变量与局部变量和连接级变量的区别在于名称前有两个 `@` 符号。例如，`@@error` 是全局变量。用户不能创建全局变量，也不能直接更新全局变量的值。

有些全局变量（如 `@@spid`）保存特定于连接的信息，因此具有特定于连接的值。有些变量（如 `@@connections`）具有所有连接共有的值。

### 全局变量和特殊常量

诸如 `CURRENT DATE`、`CURRENT TIME`、`USER`、`SQLSTATE` 等的特殊常量与全局变量很相似。

以下语句检索 `version` 全局变量的值：

```
SELECT @@version
```

在过程中，可以将全局变量选入变量列表。以下过程用 `ver` 参数返回服务器版本号。

```
CREATE PROCEDURE VersionProc ( OUT ver
                              VARCHAR ( 100) )
BEGIN
    SELECT @@version
    INTO ver;
END
```

在嵌入式 SQL 中，可以将全局变量选入宿主变量列表。

### 全局变量列表

下表列出了 SAP Sybase IQ 中可用的全局变量。

变量名	含义
<code>@@error</code>	常用于检查最近执行的语句的错误状态（成功或失败）。如果上一事务成功，则包含 0，否则包含系统生成的最后一个错误号。如果发生错误，则 <code>if @@error != 0 return</code> 这样的语句会导致退出。每个 SQL 语句都会重置 <code>@@error</code> ，因此在成功状态不确定的语句之后必须立即执行状态检查。



变量名	含义
<i>@@fetch_status</i>	包含从上一 <code>fetch</code> 语句所生成的状态信息。 <i>@@fetch_status</i> 可包含以下值 <ul style="list-style-type: none"> <li>• 0 - 读取语句成功完成。</li> <li>• -1 - 读取语句导致错误。</li> <li>• -2 - 结果集中没有其它数据。</li> </ul> <p>此功能与 <i>@sqlstatus</i> 相同，只不过它返回的值不同。这是为了与 Microsoft SQL Server 兼容。</p>
<i>@@identity</i>	通过插入、装载或更新语句插入标识/自动增量列的最后一个值。每次向表中插入一行时， <i>@@identity</i> 都会被重置。如果一个语句插入多行， <i>@@identity</i> 将反映插入的最后一行的标识/自动增量值。如果受影响的表不包含标识/自动增量列，则 <i>@@identity</i> 会设置为 0。 <i>@@identity</i> 的值不受插入、装载或更新语句失败的影响，也不受包含失败语句的事务回退的影响。 <i>@@identity</i> 保留插入标识/自动增量列的最后一个值，即使插入该值的语句未能提交也是如此。
<i>@@isolation</i>	当前隔离级别。 <i>@@isolation</i> 采用活动级别的值。
<i>@@procid</i>	当前执行的过程的存储过程 ID。
<i>@@rowcount</i>	受上一语句影响的行数。应在该语句后立即检查 <i>@@rowcount</i> 的值。插入、更新和删除会将 <i>@@rowcount</i> 设置为受影响的行数。对于游标， <i>@@rowcount</i> 表示从游标结果集返回到客户端（直到最后一个 <code>fetch</code> 命令）的累积行数。任何不影响行的语句（如 <code>IF</code> 语句）都不将 <i>@@rowcount</i> 重置为零。
<i>@@servername</i>	当前数据库服务器的名称。
<i>@@sqlstatus</i>	包含从上一 <code>FETCH</code> 语句所生成的状态信息。
<i>@@version</i>	当前 SAP Sybase IQ 版本的版本号。

### SAP Sybase IQ 中支持的 Adaptive Server 全局变量

下表包括 SAP Sybase IQ 中支持的所有 Adaptive Server 全局变量。SAP Sybase IQ 不支持的 Adaptive Server 全局变量不包括在列表中。

此列表包括返回值的所有全局变量，包括值固定为空值、1、-1 或 0 以及可能无意义的全局变量。

表 6. SAP Sybase IQ 中支持的 Adaptive Server 全局变量

全局变量	返回
<i>@@char_convert</i>	返回 0。

全局变量	返回
@@client_csname	在 Adaptive Server 中，返回客户端的字符集名称。如果客户端字符集从未初始化，则设置为 NULL；否则将包含最近使用的字符集的名称。在 SAP Sybase IQ 中返回 NULL。
@@client_csid	在 Adaptive Server 中，返回客户端的字符集 ID。如果客户端字符集从未初始化，则设置为 -1；否则将包含 syscharsets 中最近使用的客户端字符集 ID。在 SAP Sybase IQ 中返回 -1。
@@connections	自服务器上上次启动后的登录数。
@@cpu_busy	在 Adaptive Server 中，返回自上次 Adaptive Server 启动后 CPU 在 Adaptive Server 运行上所花费的时间（以时钟周期为单位）。在 SAP Sybase IQ 中返回 0。
@@error	<p>常用于检查最近执行的语句的错误状态（成功或失败）。如果上一事务成功，则包含 0，否则包含系统生成的最后一个错误号。如果发生类似如下的错误：</p> <pre>if @@error != 0 return</pre> <p>这样的语句会导致退出。每个语句（包括 PRINT 语句或 IF 测试）都会重置 @@error，因此在成功状态不确定的语句之后必须立即执行状态检查。</p>
@@identity	<p>在 Adaptive Server 中，返回通过 INSERT、LOAD 或 SELECT INTO 语句插入 IDENTITY 列的最后一个值。每次向表中插入一行时，都会重置 @@identity。如果一个语句插入多行，@@identity 将反映插入的最后一行的 IDENTITY 值。如果受影响的表不包含 IDENTITY 列，则 @@identity 会设置为 0。@@identity 的值不受 INSERT 或 SELECT INTO 语句的失败的影响，也不受包含失败语句的事务回退的影响。@@identity 保留插入 IDENTITY 列的最后一个值，即使插入该值的语句未能提交也是如此。</p>
@@idle	在 Adaptive Server 中，返回自服务器上上次启动后 Adaptive Server 的空闲时间（以时钟周期为单位）。在 SAP Sybase IQ 中返回 0。
@@io_busy	在 Adaptive Server 中，返回自服务器上上次启动后 Adaptive Server 在输入和输出操作上花费的时间（以时钟周期为单位）。在 SAP Sybase IQ 中返回 0。
@@isolation	当前连接的隔离级别。在 Adaptive Server 中，@@isolation 采用活动级别的值。
@@langid	在 Adaptive Server 中，定义当前所用语言的区域语言 ID。在 SAP Sybase IQ 中返回 0。
@@language	在 Adaptive Server 中，定义当前所用语言的名称。在 SAP Sybase IQ 中返回 "English"。
@@maxcharlen	在 Adaptive Server 中，返回 Adaptive Server 缺省字符集中的最大字符长度（以字节为单位）。在 SAP Sybase IQ 中返回 1。

全局变量	返回
@@max_connections	对于网络服务器，返回活动客户端（而非数据库连接，因为各客户端可支持多个连接）的最大数目。对于 Adaptive Server，返回与服务器连接的最大数量。
@@ncharsize	在 Adaptive Server 中，返回国家字符的平均长度（以字节为单位）。在 SAP Sybase IQ 中返回 1。
@@nestlevel	在 Adaptive Server 中，返回当前执行的嵌套级别（最初为 0）。每当存储过程调用另一个存储过程或触发器调用另一触发器时，嵌套级别就会增加。在 SAP Sybase IQ 中返回 -1。
@@pack_received	在 Adaptive Server 中，返回自服务器上次启动后 Adaptive Server 所读取的输入包数。在 SAP Sybase IQ 中返回 0。
@@pack_sent	在 Adaptive Server 中，返回自服务器上次启动后 Adaptive Server 所写入的输出包数。在 SAP Sybase IQ 中返回 0。
@@packet_errors	在 Adaptive Server 中，返回 Adaptive Server 在发送和接收数据包过程中出现的错误数。在 SAP Sybase IQ 中返回 0。
@@procid	当前执行的过程的存储过程 ID。
@@servername	本地 Adaptive Server 或 SAP Sybase IQ 服务器的名称。
@@spid	在 Adaptive Server 中，返回当前进程的服务器进程 ID 号。在 SAP Sybase IQ 中，返回当前连接的连接句柄。该值与 sa_conn_info 过程显示的值相同。
@@sqlstatus	包含从上一 <b>FETCH</b> 语句所生成的状态信息。@@sqlstatus 可能包含以下值： <ul style="list-style-type: none"> <li>• 0 - <b>FETCH</b> 语句成功完成。</li> <li>• 1 - <b>FETCH</b> 语句产生错误。</li> <li>• 2 - 结果集中没有其它数据。</li> </ul>
@@thresh_hysteresis	在 Adaptive Server 中，返回激活一个阈值所需的可用空间变化。在 SAP Sybase IQ 中返回 0。
@@timeticks	在 Adaptive Server 中，返回每个时钟周期的微秒数。每个时钟周期的时间量与计算机有关。在 SAP Sybase IQ 中返回 0。
@@total_errors	在 Adaptive Server 中，返回 Adaptive Server 在读取或写入过程中出现的错误数。在 SAP Sybase IQ 中返回 0。
@@total_read	在 Adaptive Server 中，返回自服务器上次启动后 Adaptive Server 所读取的磁盘数。在 SAP Sybase IQ 中返回 0。
@@total_write	在 Adaptive Server 中，返回自服务器上次启动后 Adaptive Server 所写入的磁盘数。在 SAP Sybase IQ 中返回 0。

全局变量	返回
@@tranchained	Transact-SQL 程序的当前事务模式。@@tranchained 为非链式模式返回 0，为链式模式返回 1。
@@trancount	事务的嵌套级别。批处理中的每个 <b>BEGIN TRANSACTION</b> 都会增加事务计数。
@@transtate	在 Adaptive Server 中，返回语句执行后事务的当前状态。在 SAP Sybase IQ 中返回 -1。
@@version	关于 Adaptive Server 或 SAP Sybase IQ 的当前版本的信息。

## 注释

使用注释在 SQL 语句或语句块中附加说明性文本。数据库服务器不执行注释。

SAP Sybase IQ 中提供以下注释指示符：

注释指示符	描述
-- (双连字符)	数据库服务器会忽略行中的所有其余字符。这是 SQL92 注释指示符。
// (双斜线)	双斜线与双连字符的含义相同。
/* ... */ (斜线加星号)	将忽略两个注释标记间的所有字符。两个注释标记可以位于同一行，也可以位于不同行。可以嵌套以此样式指示的注释。这种注释样式也称为 C 样式注释。
% (百分号)	百分号与双连字符的含义相同。SAP 建议不要将 % 用作注释指示符。

**注意：** 双连字符和斜线加星号的注释样式与 Adaptive Server 兼容。

### 示例

以下示例说明如何使用双横线注释：

```
CREATE FUNCTION fullname (firstname CHAR(30),                lastname
CHAR(30)) RETURNS CHAR(61) -- fullname concatenates the firstname and
lastname -- arguments with a single space between. BEGIN    DECLARE
name CHAR(61);        SET name = firstname || ' ' || lastname;        RETURN
( name ); END
```

以下示例说明如何使用 C 样式注释：

```
/*
    Lists the names and employee IDs of employees
    who work in the sales department.
*/
CREATE VIEW SalesEmployee AS
SELECT emp_id, emp_lname, emp_fname
FROM "GROUP0".Employees
WHERE DepartmentID = 200
```

## 空值

使用空值来指定未知、缺少或不适用的值。

空值是一个特殊值，它不同于任何数据类型的任何有效值。但空值在任何数据类型中都是合法值。以下是使用 **NULL** 的两种截然不同的情况：

情况	描述
缺少	字段具有值，但该值未知。
不适用	字段不适用于该特定的行。

SQL 允许列在创建时带有 **NOT NULL** 限制。这意味着那些特殊的列不能包含空值。

空值将三值逻辑的概念引入 SQL。使用任何比较运算符与任何值（包括空值）进行比较的空值都是 **UNKNOWN**。返回 **TRUE** 的唯一搜索条件是 **IS NULL** 谓词。在 SQL 中，只有 **WHERE** 子句中的搜索条件的值为 **TRUE** 时才选择行；不选择值为 **UNKNOWN** 或 **FALSE** 的行。

此外，也可以使用 **IS [ NOT ] truth-value** 子句（其中 *truth-value* 是 **TRUE**、**FALSE** 或 **UNKNOWN** 之一）选择包含空值的行。

在以下示例中，列 **Salary** 包含 **NULL** 值。

条件	真值	是否选择?
Salary = NULL	UNKNOWN	NO
Salary <> NULL	UNKNOWN	NO
NOT (Salary = NULL)	UNKNOWN	NO
NOT (Salary <> NULL)	UNKNOWN	NO
Salary = 1000	UNKNOWN	NO
Salary IS NULL	TRUE	YES
Salary IS NOT NULL	FALSE	NO
Salary = 1000 IS UNKNOWN	TRUE	YES

同样的规则也适用于比较两个不同表中的列。因此，将两个表连接在一起不会选择其中进行比较的任何列含有空值的行。

用于数值表达式时，**NULL** 值还有一个有趣的属性。任何涉及 **NULL** 值的数值表达式的结果都为 **NULL**。这意味着，如果将 **NULL** 值加上一个数字，结果为 **NULL** 值，而不是数字。如果要将 **NULL** 值视为 0，则必须使用 **ISNULL( expression, 0 )** 函数。

公式化 SQL 查询时的很多常见错误是由 **NULL** 的行为引起的。请小心避免这类问题。组合搜索条件时，请注意三值逻辑的效果。

### 语法

**NULL**

### 用法

任何位置

### 权限

必须连接到数据库

### 副作用

无

### 示例

以下 **INSERT** 语句将 **NULL** 插入到 Borrowed\_book 表的 date\_returned 列中。

```
INSERT  
INTO Borrowed_book  
( date_borrowed, date_returned, book )  
VALUES ( CURRENT DATE, NULL, '1234' )
```

### 另请参见

- 比较条件 (第 40 页)
- 表达式 (第 27 页)
- 搜索条件 (第 39 页)
- 字符串 (第 26 页)
- 三值逻辑 (第 42 页)
- **SQL** 运算符 (第 30 页)
- 搜索条件中的子查询 (第 43 页)

# SQL 函数

函数返回数据库中的信息并且可以在允许使用表达式的任意位置使用。

在 SAP Sybase IQ 中使用函数时，除非另外说明，否则，任何接收 NULL 值作为参数的函数都会返回 NULL 值。

如果省略 **FROM** 子句，或者查询中的所有表都位于 **SYSTEM** dbspace 中，则查询将由 SQL Anywhere 而非 SAP Sybase IQ 来处理且行为可能不同，特别是关于语法和语义限制以及选项设置的影响方面。

如果您的查询不需要 **FROM** 子句，则可以通过添加 "**FROM iq\_dummy**" 子句强制由 SAP Sybase IQ 处理查询，其中 **iq\_dummy** 是在数据库中创建的包含一行和一列的表。

另请参见

- 附加函数（第 373 页）

## 集合函数

---

集合函数可汇总数据库中一组行中的数据。这些组是使用 **SELECT** 语句的 **GROUP BY** 子句构成的。

*用法*

简单集合函数（如 **SUM()**、**MIN()**、**MAX()**、**AVG()** 和 **COUNT()**）只能在选择列表中以及 **SELECT** 语句的 **HAVING** 和 **ORDER BY** 子句中使用。这些函数可汇总数据库中一组行中的数据。这些组是使用 **SELECT** 语句的 **GROUP BY** 子句构成的。

一类名为 *窗口函数* 的新集合函数提供移动平均值和累计测量值来计算诸如 “What is the quarterly moving average of the Dow Jones Industrial average” 或 “List all employees and their cumulative salaries for each department” 等查询的答案。

- 简单集合函数（如 **AVG()**、**COUNT()**、**MAX()**、**MIN()** 和 **SUM()**）可对数据库的一组行中的数据进行汇总。这些组是使用 **SELECT** 语句的 **GROUP BY** 子句构成的。
- 采用一个参数的较新的统计集合函数包括 **STDDEV()**、**STDDEV\_SAMP()**、**STDDEV\_POP()**、**VARIANCE()**、**VAR\_SAMP()** 和 **VAR\_POP()**。

简单集合类别和较新的集合类别都可以用作将 **<window clause>** 纳入 SQL 查询规范（一个 *窗口*）中的窗口函数，SQL 查询规范在概念上针对所处理的结果集创建移动窗口。

另一类窗口集合函数支持对时序数据进行分析。与简单集合函数和统计集合函数一样，您可以结合 SQL 查询规范（或 *window-spec*）使用这些窗口集合。时序窗口集合函数可计算相关、线性回归、排名和加权平均值结果：

- 用于进行时序分析的 ISO/ANSI SQL:2008 OLAP 函数包括：**CORR()**、**COVAR\_POP()**、**COVAR\_SAMP()**、**CUME\_DIST()**、**FIRST\_VALUE()**、**LAST\_VALUE()**、**REGR\_AVGX()**、**REGR\_AVGY()**、**REGR\_COUNT()**、**REGR\_INTERCEPT()**、**REGR\_R2()**、**REGR\_SLOPE()**、**REGR\_SXX()**、**REGR\_SXY()** 和 **REGR\_SYY()**。
- 在数据库行业使用的非 ISO/ANSI SQL:2008 OLAP 集合函数扩展包括 **FIRST\_VALUE()**、**MEDIAN()** 和 **LAST\_VALUE()**。
- 用于计算加权移动平均值的加权 OLAP 集合函数包括 **EXP\_WEIGHTED\_AVG()** 和 **WEIGHTED\_AVG()**。

专门为财务时序预测和分析而设计的时序函数的名称以 “**TS\_**” 开头。

有关 OLAP 的详细信息，请参见编程中的附录：使用 OLAP。

有关 LONG BINARY 和 LONG VARCHAR 数据类型的集合函数支持的信息，请参见非结构化数据分析。

集合函数 **AVG**、**SUM**、**STDDEV** 和 **VARIANCE** 不支持二进制数据类型（**BINARY** 和 **VARBINARY**）。

### 另请参见

- 分析函数（第 77 页）
- 集合函数（第 75 页）
- **AVG** 函数 [集合]（第 138 页）
- **CORR** 函数 [集合]（第 157 页）
- **COVAR\_POP** 函数 [集合]（第 160 页）
- **COVAR\_SAMP** 函数 [集合]（第 161 页）
- **COUNT** 函数 [集合]（第 162 页）
- **CUME\_DIST** 函数 [分析]（第 163 页）
- **EXP\_WEIGHTED\_AVG** 函数 [集合]（第 194 页）
- **FIRST\_VALUE** 函数 [集合]（第 196 页）
- **LAST\_VALUE** 函数 [集合]（第 223 页）
- **LIST** 函数 [集合]（第 231 页）
- **MAX** 函数 [集合]（第 239 页）
- **MEDIAN** 函数 [集合]（第 240 页）
- **MIN** 函数 [集合]（第 241 页）
- **REGR\_AVGX** 函数 [集合]（第 274 页）
- **REGR\_AVGY** 函数 [集合]（第 275 页）
- **REGR\_COUNT** 函数 [集合]（第 276 页）
- **REGR\_INTERCEPT** 函数 [集合]（第 277 页）
- **REGR\_R2** 函数 [集合]（第 278 页）
- **REGR\_SLOPE** 函数 [集合]（第 280 页）



- REGR\_SXX 函数 [集合] (第 281 页)
- REGR\_SXY 函数 [集合] (第 282 页)
- REGR\_SYY 函数 [集合] (第 283 页)
- STDDEV 函数 [集合] (第 310 页)
- STDDEV\_POP 函数 [集合] (第 311 页)
- STDDEV\_SAMP 函数 [集合] (第 312 页)
- SUM 函数 [集合] (第 322 页)
- VAR\_POP 函数 [集合] (第 332 页)
- VAR\_SAMP 函数 [集合] (第 333 页)
- VARIANCE 函数 [集合] (第 334 页)
- WEIGHTED\_AVG 函数 [集合] (第 337 页)

## 分析函数

---

分析函数包括简单集合、窗口函数和数值函数。

- 简单集合 - **AVG**、**COUNT**、**MAX**、**MIN**、**SUM**、**STDDEV** 和 **VARIANCE**

---

**注意：** 除 Grouping() 函数外，您可以结合 OLAP 窗口函数使用所有简单集合。

---

- 窗口函数：
  - 窗口化集合 - **AVG**、**COUNT**、**MAX**、**MIN** 和 **SUM**。
  - 排名函数 - **RANK**、**DENSE\_RANK**、**PERCENT\_RANK**、**ROW\_NUMBER** 和 **NTILE**。
  - 统计函数 - **STDDEV**、**STDDEV\_SAMP**、**STDDEV\_POP**、**VARIANCE**、**VAR\_SAMP** 和 **VAR\_POP**。
  - 分布函数 - **PERCENTILE\_CONT** 和 **PERCENTILE\_DISC**。
  - 行间函数 - **LAG** 和 **LEAD**。
- 数值函数 - **WIDTH\_BUCKET**、**CEIL**、**LN**、**EXP**、**POWER**、**SQRT** 和 **FLOOR**。

---

**注意：** Adaptive Server 不支持秩函数和逆分布分析函数。

---

与某些集合函数不同的是，您不能在窗口函数中指定 **DISTINCT**。

\* OLAP SQL 标准仅允许在 **GROUP BYCUBE** 或 **GROUP BY ROLLUP** 操作中使用 Grouping()。

### 另请参见

- 集合函数 (第 75 页)
- AVG 函数 [集合] (第 138 页)
- COUNT 函数 [集合] (第 162 页)
- DENSE\_RANK 函数 [分析] (第 186 页)
- GROUPING 函数 [集合] (第 201 页)

- MAX 函数 [集合] (第 239 页)
- MIN 函数 [集合] (第 241 页)
- NTILE 函数 [分析] (第 253 页)
- PERCENT\_RANK 函数 [分析] (第 261 页)
- PERCENTILE\_CONT 函数 [分析] (第 262 页)
- PERCENTILE\_DISC 函数 [分析] (第 264 页)
- RANK 函数 [分析] (第 273 页)
- ROW\_NUMBER 函数 [分析] (第 292 页)
- STDDEV 函数 [集合] (第 310 页)
- STDDEV\_POP 函数 [集合] (第 311 页)
- STDDEV\_SAMP 函数 [集合] (第 312 页)
- SUM 函数 [集合] (第 322 页)
- VAR\_POP 函数 [集合] (第 332 页)
- VAR\_SAMP 函数 [集合] (第 333 页)
- VARIANCE 函数 [集合] (第 334 页)

### 窗口化集合函数的用法

OLAP 的 ISO/ANSI SQL 扩展的一个主要特征是：它具有一个名为 *窗口* 的结构。利用此窗口化扩展，用户可以将查询（或查询的逻辑分区）的结果集分成名为“分区”的行组，并确定要与当前行集合的行的子集。

您可以将以下三类窗口函数与窗口一起使用：排名函数、行计算函数和窗口集合函数。

窗口化扩展插件通过窗口名称或窗口说明指定窗口函数类型，并应用于单个查询表达式范围内的分区结果集。

使用窗口化操作，可以建立信息，诸如在行的分区中排列每行、在某个分区内的行中分布值，以及类似操作。利用窗口化操作，还可对数据计算移动平均值和总数，从而增强对数据及其对操作的影响进行评估的能力。

窗口分区是由查询返回的行的子集，由一个特殊的 **OVER()** 子句中的一个或多个列定义：

```
OVER (PARTITION BY col1, col2...)
```

有关支持 **LONG BINARY** 和 **LONG VARCHAR** 数据类型的分析函数的信息，请参见非结构化数据分析中的函数支持。

#### 另请参见

- CORR 函数 [集合] (第 157 页)
- COUNT 函数 [集合] (第 162 页)
- EXP\_WEIGHTED\_AVG 函数 [集合] (第 194 页)
- FIRST\_VALUE 函数 [集合] (第 196 页)

- GROUPING 函数 [集合] (第 201 页)
- LAST\_VALUE 函数 [集合] (第 223 页)
- MAX 函数 [集合] (第 239 页)
- MEDIAN 函数 [集合] (第 240 页)
- MIN 函数 [集合] (第 241 页)
- REGR\_AVGX 函数 [集合] (第 274 页)
- REGR\_COUNT 函数 [集合] (第 276 页)
- REGR\_INTERCEPT 函数 [集合] (第 277 页)
- REGR\_R2 函数 [集合] (第 278 页)
- REGR\_SLOPE 函数 [集合] (第 280 页)
- REGR\_SXX 函数 [集合] (第 281 页)
- REGR\_SXY 函数 [集合] (第 282 页)
- REGR\_SYY 函数 [集合] (第 283 页)
- STDDEV 函数 [集合] (第 310 页)
- SUM 函数 [集合] (第 322 页)
- STDDEV\_POP 函数 [集合] (第 311 页)
- STDDEV\_SAMP 函数 [集合] (第 312 页)
- VAR\_POP 函数 [集合] (第 332 页)
- VAR\_SAMP 函数 [集合] (第 333 页)
- VARIANCE 函数 [集合] (第 334 页)
- WEIGHTED\_AVG 函数 [集合] (第 337 页)

## 排名函数的用法

使用 OLAP 排名函数，应用程序开发人员可以编写单语句的 SQL 查询，来回答诸如“Name the top 10 products shipped this year by total sales”或“Give the top 5% of salespeople who sold orders to at least 15 different companies”之类的问题。

这些函数包括排名函数 **RANK()**、**DENSE\_RANK()**、**PERCENT\_RANK()**、**ROW\_NUMBER()** 和 **NTILE()**。

Rank 分析函数将组中的项进行排序、计算分布，并将结果集分为多个组。Rank 分析函数 **RANK()**、**DENSE\_RANK()**、**PERCENT\_RANK()**、**ROW\_NUMBER()** 和 **NTILE()** 都需要使用 **OVER (ORDER BY)** 子句。例如：

```
RANK() OVER ( [PARTITION BY] ORDER BY <expression>
[ ASC | DESC ] )
```

**ORDER BY** 子句指定要执行排序的参数以及每个组中行的排列顺序。此 **ORDER BY** 子句只在 **OVER** 子句中使用，而不是用于 **SELECT** 的 **ORDER BY**。不允许排序查询 **ROW** 中的任何集合函数指定 **DISTINCT**。

---

**注意：** **ROW\_NUMBER()** 函数的 **OVER (ORDER BY)** 子句不能包含 **ROWS** 或 **RANGE** 子句。

---

**OVER** 子句表示函数对查询结果集进行操作。结果集是在对 **FROM**、**WHERE**、**GROUP BY** 和 **HAVING** 子句求值完成之后返回的行。**OVER** 子句定义要包括在 rank 分析函数计算中的行数据集。

值 *expression* 为一个排序规范，可以是涉及列引用、集合的任意有效表达式，也可以是调用这些项的表达式。

**ASC** 或 **DESC** 参数指定排序顺序，如升序或降序。升序是缺省值。

Rank 分析函数只能在 **SELECT** 或 **INSERT** 语句的选择列表中使用，或者在 **SELECT** 语句的 **ORDER BY** 子句中使用。Rank 函数可以在视图中或并集中使用。不能在子查询中、**HAVING** 子句中，或者 **UPDATE** 或 **DELETE** 语句的选择列表中使用 rank 函数。在 SAP Sybase IQ 16.0 中，允许在一个查询中使用多个 rank 分析函数。

## 统计聚合分析函数的用法

统计聚合分析函数可汇总数据库中一组行中的数据。

这些组是使用 **SELECT** 语句的 **GROUP BY** 子句构成的。集合函数只能在选择列表中以及 **SELECT** 语句的 **HAVING** 和 **ORDER BY** 子句中使用。这些函数包括 **STDDEV**、**STDDEV\_POP**、**STDDEV\_SAMP**、**VARIANCE**、**VAR\_POP** 和 **VAR\_SAMP**。

OLAP 函数可作为窗口函数与 **OVER()** 子句一起用在 SQL 查询规范中，SQL 查询规范在概念上针对所处理的结果集创建移动窗口。

## 分布函数的用法

逆分布分析函数 **PERCENTILE\_CONT** 和 **PERCENTILE\_DISC** 将一个百分点值作为函数参数使用，并对 **WITHIN GROUP** 子句中指定的一组数据执行操作或对整个数据集执行操作。

这些函数为每个组返回一个值。对于 **PERCENTILE\_DISC**，结果的数据类型与在 **WITHIN GROUP** 子句中指定的它的 **ORDER BY** 项的数据类型相同。对于 **PERCENTILE\_CONT**，结果的数据类型为数值（如果 **WITHIN GROUP** 子句中的 **ORDER BY** 项为数值）或双精度（如果 **ORDER BY** 项为整数或浮点数）。

逆分布分析函数需要 **WITHIN GROUP (ORDER BY)** 子句。例如：

```
PERCENTILE_CONT ( expression1 ) WITHIN GROUP ( ORDER BY expression2
[ASC | DESC ] )
```

*expression1* 的值必须是数值数据类型的常量，范围从 0 到 1（包含这两个数）。如果参数为 NULL，将返回“wrong argument for percentile”（百分点的参数错误）错误。如果参数值小于 0 或大于 1，将返回“data value out of range”（数据值超出范围）错误。

**ORDER BY** 子句（必须存在）指定要执行百分点函数的表达式以及每个组中行的排列顺序。此 **ORDER BY** 子句只在 **WITHIN GROUP** 子句中使用，而不是 **SELECT** 的 **ORDER BY**。

**WITHIN GROUP** 子句将查询结果分布到排序数据集中，函数通过此数据集计算结果。

值 *expression2* 是一种排序规范，必须是涉及列引用的单个表达式。不允许多个表达式，并且在此排序表达式中不允许使用 **rank** 分析函数、集合函数或子查询。

**ASC** 或 **DESC** 参数指定排序顺序，如升序或降序。升序是缺省值。

子查询、**HAVING** 子句、视图或联合中允许使用逆分布分析函数。可以在使用简单非分析集合函数的任意位置使用逆分布函数。逆分布函数忽略数据集中的空值。

## 行间函数的用法

通过行间函数 **LAG** 和 **LEAD**，可以访问数据系列中的先前值或后续值。

通过这些函数可以同时访问表或分区中的多行，且无需自连接。通过 **LAG** 函数可以对位于表或分区中 **CURRENT ROW** 前面且与之相距给定物理偏移量的行进行访问。通过 **LEAD** 函数可以对位于表或分区中 **CURRENT ROW** 后面且与之相距给定物理偏移量的行进行访问。使用 **LAG** 和 **LEAD** 函数可以创建诸如 "What was the stock price two intervals before the current row" 和 "What was the stock price one interval after the current row" 之类的查询。

行间函数需要使用 **OVER (ORDER\_BY)** 子句。

### 行间函数

通过行间函数 **LAG** 和 **LEAD** 可以访问数据系列中的先前值或后续值，或表中的多个行。

行间函数同时还会进行分区，且无需自连接。通过 **LAG** 可以对位于表或分区中 **CURRENT ROW** 前面且与之相距给定物理偏移量的行进行访问。通过 **LEAD** 可以对位于表或分区中 **CURRENT ROW** 后面且与之相距给定物理偏移量的行进行访问。

**LAG** 和 **LEAD** 使用的语法相同。这两个函数都需要 **OVER (ORDER\_BY)** 窗口规范。例如：

```
LAG (value_expr) [, offset [, default]] OVER ([PARTITION BY
window partition] ORDER BY
window ordering)
```

和：

```
LEAD (value_expr) [, offset [, default]] OVER ([PARTITION
BY
window partition] ORDER BY
window ordering)
```

**OVER (ORDER\_BY)** 子句中的 **PARTITION BY** 子句是可选的。**OVER (ORDER\_BY)** 子句不能包含窗口构架 **ROWS/RANGE** 规范。

*value\_expr* 是定义要从表返回的偏移数据的表列或表达式。可以在 *value\_expr* 中定义其它函数（分析函数除外）。

对于这两个函数，可输入物理偏移量来指定目标行。*offset* 值是指当前行上面或下面的行数。请输入非负数值数据类型（输入负值会生成错误）。如果输入 0，SAP Sybase IQ 会返回当前行。

可选的 *default* 值定义 *offset* 值超出表范围时要返回的值。*default* 的缺省值为 **NULL**。*default* 的数据类型必须可以隐式转换为 *value\_expr* 值的数据类型，否则 SAP Sybase IQ 将生成转换错误。

**LAG** 示例 1 - 行间函数适用于对数据流进行计算的金融服务应用程序（如股票交易）。此示例使用 **LAG** 函数来计算特定股票的交易价格的百分比变化。假设有以下来自 `stock_trades` 虚构表的贸易数据：

traded at	symbol	price
2009-07-13 06:07:12	SQL	15.84
2009-07-13 06:07:13	TST	5.75
2009-07-13 06:07:14	TST	5.80
2009-07-13 06:07:15	SQL	15.86
2009-07-13 06:07:16	TST	5.90
2009-07-13 06:07:17	SQL	15.86

**注意：** 虚构表 `stock_trades` 在 `iqdemo` 数据库中不可用。

查询按照股票符号对交易进行分区，按照交易时间对它们进行排序，并使用 **LAG** 函数来计算当前交易和以前交易之间的交易价格增加或降低百分比：

```
select stock_symbol as 'Stock',
       traded_at    as 'Date/Time of Trade',
       trade_price  as 'Price/Share',
       cast ( ( ( (trade_price
                  - (lag(trade_price, 1)
                       over (partition by stock_symbol
                             order by traded_at)))
                  / trade_price)
              * 100.0) as numeric(5, 2) )
         as '% Price Change vs Previous Price'
from stock_trades
order by 1, 2
```

查询返回以下结果：

Stock symbol	Date/Time of Trade	Price/Share	% Price Change vs Previous Price
SQL	2009-07-13 06:07:12	15.84	NULL
SQL	2009-07-13 06:07:15	15.86	0.13
SQL	2009-07-13 06:07:17	15.86	0.00
TST	2009-07-13 06:07:13	5.75	NULL
TST	2009-07-13 06:07:14	5.80	0.87
TST	2009-07-13 06:07:16	5.90	1.72

第一个和第四个输出行中的 NULL 结果表明 **LAG** 函数同时超出了两个分区中第一行的范围。由于没有先前行可供比较，**SAP Sybase IQ** 返回 *default* 变量指定的 NULL。

## 数据类型转换函数

---

数据类型转换函数将参数从一种数据类型转换为另一种数据类型。

数据库服务器自动执行许多类型转换。例如，如果在需要数值表达式时提供了字符串，则会自动将该字符串转换为数字。

### 另请参见

- 数据类型转换 (第 366 页)
- 存储大小 (第 355 页)
- **BIGINTTOHEX** 函数 [数据类型转换] (第 139 页)
- **CAST** 函数 [数据类型转换] (第 143 页)
- **CONVERT** 函数 [数据类型转换] (第 153 页)
- **HEXTOBIGINT** 函数 [数据类型转换] (第 203 页)
- **HEXTOINT** 函数 [数据类型转换] (第 204 页)
- **INTTOHEX** 函数 [数据类型转换] (第 218 页)
- **ISNUMERIC** 函数 [杂项] (第 221 页)

## 日期和时间函数

---

日期和时间函数对日期和时间数据类型执行转换、提取或处理操作，并且可以返回日期和时间信息。

日期和时间函数允许操作时间单位。大多数时间单位（如 MONTH）都具有四个用于时间操作的函数，但只使用两个名称（如 MONTH 和 MONTHS）。

下列函数是 **Transact-SQL** 日期和时间函数。它们提供了另一种用于访问和操作日期和时间函数的方式：

- DATEADD
- DATEDIFF
- DATENAME
- DATEPART
- GETDATE

您应该先将日期函数的参数转换为日期，然后再使用。例如，下例是不正确的：

```
days ( '1995-11-17', 2 )
```

下面的示例是正确的：

```
days ( date( '1995-11-17' ), 2 )
```

SAP Sybase IQ 没有与 SQL Anywhere 相同的常量或数据类型升级，但与其共享公共用户接口。如果您发出不包含 FROM 子句的 **SELECT** 语句，则该语句会传递到 SQL Anywhere。以下语句由 SQL Anywhere 以独占方式处理：

```
SELECT WEEKS( '1998/11/01' );
```

以下语句由 SAP Sybase IQ 处理，它为 **WEEKS** 函数使用不同的起点，并且与上面的语句返回的结果不同：

```
SELECT WEEKS( '1998/11/01' ) FROM iq_dummy;
```

请考虑另一示例。**MONTHS** 函数返回自“任意开始日期”以来的月数。系统将选择 SAP Sybase IQ 的“任意开始日期”（虚构日期 0000-01-01）来生成最有效的日期计算，并且该日期将在各个数据部分之间保持一致。SQL Anywhere 没有单个开始日期。下列语句（第一个由 SQL Anywhere 处理，第二个由 SAP Sybase IQ 处理）均返回答案 12：

```
SELECT MONTHS( '0001/01/01' );
```

```
SELECT MONTHS( '0001/01/01' ) FROM iq_dummy;
```

但是，另请考虑以下语句：

```
SELECT DAYS( '0001/01/01' );
```

```
SELECT DAYS( '0001/01/01' ) FROM iq_dummy;
```

由 SQL Anywhere 处理的第一个语句生成值 307，但由 SAP Sybase IQ 处理的第二个语句则生成值 166。

因此，对于大多数一致的结果，无论是否需要，都要始终在 **FROM** 子句中包括表名。

**注意：** 创建一个只包含一列和一行的虚拟表。然后，可以在任何使用日期或时间函数的 **SELECT** 语句的 **FROM** 子句中引用该表，从而确保由 SAP Sybase IQ 处理并且生成一致的结果。

### 另请参见

- ISDATE 函数 [日期和时间]（第 219 页）
- DATE 函数 [日期和时间]（第 165 页）
- DATEADD 函数 [日期和时间]（第 165 页）
- DATECEILING 函数 [日期和时间]（第 167 页）
- DATEDIFF 函数 [日期和时间]（第 169 页）
- DATEFLOOR 函数 [日期和时间]（第 171 页）
- DATEFORMAT 函数 [日期和时间]（第 174 页）
- DATENAME 函数 [日期和时间]（第 175 页）
- DATEPART 函数 [日期和时间]（第 176 页）
- DATEROUND 函数 [日期和时间]（第 177 页）
- DATETIME 函数 [日期和时间]（第 180 页）



- DAY 函数 [日期和时间] (第 180 页)
- DAYNAME 函数 [日期和时间] (第 181 页)
- DAYS 函数 [日期和时间] (第 182 页)
- DOW 函数 [日期和时间] (第 189 页)
- GETDATE 函数 [日期和时间] (第 199 页)
- HOUR 函数 [日期和时间] (第 205 页)
- HOURS 函数 [日期和时间] (第 206 页)
- MINUTE 函数 [日期和时间] (第 242 页)
- MINUTES 函数 [日期和时间] (第 242 页)
- MONTH 函数 [日期和时间] (第 244 页)
- MONTHNAME 函数 [日期和时间] (第 245 页)
- MONTHS 函数 [日期和时间] (第 246 页)
- NOW 函数 [日期和时间] (第 253 页)
- QUARTER 函数 [日期和时间] (第 270 页)
- SECOND 函数 [日期和时间] (第 296 页)
- SECONDS 函数 [日期和时间] (第 297 页)
- TODAY 函数 [日期和时间] (第 325 页)
- WEEKS 函数 [日期和时间] (第 336 页)
- YEAR 函数 [日期和时间] (第 341 页)
- YEARS 函数 [日期和时间] (第 342 页)
- YMD 函数 [日期和时间] (第 343 页)

## 日期分量

许多日期函数都使用从日期分量生成的日期。

下表显示了 *date-part* 允许的值。

日期部分	缩写	值
Year	yy	0001 - 9999
Quarter	qq	1 - 4
Month	mm	1 - 12
Week	wk	1 - 54
Day	dd	1 - 31
Dayofyear	dy	1 - 366
Weekday	dw	1 - 7 (周日 - 周六)
Hour	hh	0 - 23

日期部分	缩写	值
Minute	mi	0 - 59
Second	ss	0 - 59
Millisecond	ms	0 - 999
Microsecond	mcs 或 us	0 - 999999
Calyearofweek	cyr	整型。一周开始的年份。包含一年中的前几天的周可以是上一年中最后一周的一部分，具体取决于它开始于哪一天。如果新年的开始日期为星期四到星期六，则它的第一周开始于上一年的最后一个星期日。如果新年的开始日期为星期日到星期三，则它的任何一天都不是上一年的一部分。
Calweekofyear	cwk	一个从 1 到 54 的整数，表示年中包含指定日期的周编号。
Caldayofweek	cdw	周中的日期编号（星期日 = 1，星期六 = 7）。

**注意：** 缺省情况下，星期日是一周中的第一天。若要将星期一用作第一天，请使用：  
`set option 'Date_First_Day_Of_Week' = '1'`

#### 兼容性

若要与 Adaptive Server 兼容，请使用 Transact-SQL 日期和时间函数。

#### 另请参见

- DATEADD 函数 [日期和时间]（第 165 页）
- DATECEILING 函数 [日期和时间]（第 167 页）
- DATEDIFF 函数 [日期和时间]（第 169 页）
- DATEFLOOR 函数 [日期和时间]（第 171 页）
- DATEPART 函数 [日期和时间]（第 176 页）
- DATENAME 函数 [日期和时间]（第 175 页）
- DATEROUND 函数 [日期和时间]（第 177 页）

## HTTP 函数

使用 HTTP 函数可便于处理 Web 服务中的 HTTP 请求。

**注意：** 确保您的 Web 服务使用最佳编码实践来防止跨站点脚本 (XSS) 攻击。可以使用组织中开放源的资源，如 OWASP。

#### 另请参见

- HTML\_DECODE 函数 [Miscellaneous]（第 207 页）
- HTML\_ENCODE 函数 [Miscellaneous]（第 208 页）

- HTTP\_DECODE 函数 [Web 服务] (第 211 页)
- HTTP\_ENCODE 函数 [Web 服务] (第 211 页)
- HTTP\_HEADER 函数 [Web 服务] (第 212 页)
- HTTP\_VARIABLE 函数 [Web 服务] (第 214 页)
- NEXT\_HTTP\_HEADER 函数 [Web 服务] (第 251 页)
- NEXT\_HTTP\_VARIABLE 函数 [Web 服务] (第 252 页)

## 数值函数

---

数值函数对数值数据类型执行数学运算或者返回数值信息。

SAP Sybase IQ 没有与 SQL Anywhere 相同的常量或数据类型升级，但与其共享公共用户接口。如果您发出不包含 **FROM** 子句的 **SELECT** 语句，则该语句会传递到 SQL Anywhere。对于大多数一致的结果，无论是否需要，都应在 **FROM** 子句中包括表名。

---

**注意：** 请考虑创建一个虚拟表以在这种情况下使用。

---

### 另请参见

- ABS 函数 [数值] (第 133 页)
- ACOS 函数 [数值] (第 133 页)
- ASIN 函数 [数值] (第 136 页)
- ATAN 函数 [数值] (第 137 页)
- ATAN2 函数 [数值] (第 137 页)
- CEIL 函数 [数值] (第 144 页)
- CEILING 函数 [数值] (第 145 页)
- COS 函数 [数值] (第 158 页)
- COT 函数 [数值] (第 159 页)
- DEGREES 函数 [数值] (第 186 页)
- EXP 函数 [数值] (第 194 页)
- FLOOR 函数 [数值] (第 198 页)
- LN 函数 [数值] (第 233 页)
- LOG 函数 [数值] (第 235 页)
- LOG10 函数 [数值] (第 236 页)
- MOD 函数 [数值] (第 244 页)
- PI 函数 [数值] (第 266 页)
- POWER 函数 [数值] (第 266 页)
- RADIANS 函数 [数值] (第 271 页)
- RAND 函数 [数值] (第 272 页)
- REMAINDER 函数 [数值] (第 284 页)
- ROUND 函数 [数值] (第 291 页)

- SIGN 函数 [数值] (第 298 页)
- SIN 函数 [数值] (第 299 页)
- SQRT 函数 [数值] (第 309 页)
- SQUARE 函数 [数值] (第 309 页)
- TAN 函数 [数值] (第 324 页)
- TRUNCNUM 函数 [数值] (第 326 页)
- WIDTH\_BUCKET 函数 [数值] (第 339 页)

## 字符串函数

---

字符串函数对字符串执行转换、提取或处理操作，或者返回有关字符串的信息。

当使用多字节字符集时，请仔细检查所使用的函数是否返回有关字符或字节的信息。

大多数字符串函数都接受在 *string-expr* 参数中使用二进制数据（十六进制字符串），但有些函数（如 **LCASE**、**UCASE**、**LOWER** 和 **LTRIM**）期望字符串表达式成为字符串。

除非为生成 LONG VARCHAR 结果的函数（如 **SPACE** 或 **REPEAT**）提供常量 **LENGTH** 参数，否则缺省长度即为允许的最大长度。

包含其中一个或多个函数的 SAP Sybase IQ 查询可能会返回下列错误之一：

```
ASA Error -1009080: Key doesn't fit on a single database page:
65560(4, 1)
```

```
ASA Error -1009119: Record size too large for database page size
```

例如：

```
SELECT COUNT(*) FROM test1 a WHERE (a.col1 + SPACE(4-LENGTH(a.col1))
+ a.col2 + space(2- LENGTH(a.col2))) IN (SELECT (b.col3) FROM test1
b);
```

若要避免出现这种错误，请转换具有适当的最大长度的函数结果；例如：

```
SELECT COUNT(*) FROM test1 a WHERE (a.col1 + CAST(SPACE(4-
LENGTH(a.col1)) AS VARCHAR(4)) + a.col2 + CAST(SPACE(2-LENGTH
(a.col2)) AS VARCHAR(4))) IN (SELECT (b.col3) FROM test1 b);
```

64K 的页面大小或多字节归类更易出错。

---

**注意：** 有关支持 LONG BINARY 和 LONG VARCHAR 数据类型的字符串函数的信息，请参见《非结构化数据分析》中的函数支持。

---

另请参见

- ASCII 函数 [字符串] (第 135 页)
- BIT\_LENGTH 函数 [字符串] (第 140 页)
- BYTE\_LENGTH 函数 [字符串] (第 141 页)
- CHAR 函数 [字符串] (第 146 页)

- CHAR\_LENGTH 函数 [字符串] (第 147 页)
- CHARINDEX 函数 [字符串] (第 148 页)
- DIFFERENCE 函数 [字符串] (第 188 页)
- GRAPHICAL\_PLAN 函数 [字符串] (第 199 页)
- HTML\_PLAN 函数 [字符串] (第 209 页)
- INSERTSTR 函数 [字符串] (第 217 页)
- LCASE 函数 [字符串] (第 225 页)
- LEFT 函数 [字符串] (第 228 页)
- LEN 函数 [字符串] (第 229 页)
- LENGTH 函数 [字符串] (第 230 页)
- LOCATE 函数 [字符串] (第 234 页)
- LOWER 函数 [字符串] (第 237 页)
- LTRIM 函数 [字符串] (第 238 页)
- PATINDEX 函数 [字符串] (第 259 页)
- REPEAT 函数 [字符串] (第 285 页)
- REPLACE 函数 [字符串] (第 286 页)
- REPLICATE 函数 [字符串] (第 288 页)
- REVERSE 函数 [字符串] (第 289 页)
- RIGHT 函数 [字符串] (第 290 页)
- RTRIM 函数 [字符串] (第 295 页)
- SIMILAR 函数 [字符串] (第 298 页)
- SORTKEY 函数 [字符串] (第 300 页)
- SOUNDEX 函数 [字符串] (第 304 页)
- SPACE 函数 [字符串] (第 307 页)
- STR 函数 [字符串] (第 314 页)
- STR\_REPLACE 函数 [字符串] (第 315 页)
- STRING 函数 [字符串] (第 316 页)
- STRTOUUID 函数 [字符串] (第 317 页)
- STUFF 函数 [字符串] (第 318 页)
- SUBSTRING 函数 [字符串] (第 319 页)
- SUBSTRING64 函数 [字符串] (第 322 页)
- UCASE 函数 [字符串] (第 327 页)
- UPPER 函数 [字符串] (第 328 页)
- UUIDTOSTR 函数 [字符串] (第 331 页)

## 系统函数

---

系统函数返回系统信息。

### *描述*

当前运行于服务器的数据库由数据库名称和数据库 ID 号标识。db\_id 和 db\_name 函数提供有关这些值的信息。

一组系统函数提供有关数据库服务器上当前运行的数据库或连接的属性的信息。这些系统函数采用数据库名称或 ID 或者连接名称作为可选参数，以标识为其请求该属性的数据库或连接。

### *性能*

系统函数的处理方式不同于其它 SAP Sybase IQ 函数。当对 SAP Sybase IQ 表的查询包括系统函数时，查询的性能会降低。

### 另请参见

- CEIL 函数 [数值] (第 144 页)
- COL\_LENGTH 函数 [系统] (第 150 页)
- COL\_NAME 函数 [系统] (第 151 页)
- CONNECTION\_PROPERTY 函数 [系统] (第 152 页)
- DATALENGTH 函数 [系统] (第 164 页)
- DB\_ID 函数 [系统] (第 183 页)
- DB\_NAME 函数 [系统] (第 184 页)
- DB\_PROPERTY 函数 [系统] (第 185 页)
- EVENT\_CONDITION 函数 [系统] (第 190 页)
- EVENT\_CONDITION\_NAME 函数 [系统] (第 192 页)
- EVENT\_PARAMETER 函数 [系统] (第 192 页)
- GROUP\_MEMBER 函数 [系统] (第 202 页)
- INDEX\_COL 函数 [系统] (第 216 页)
- NEXT\_CONNECTION 函数 [系统] (第 248 页)
- NEXT\_DATABASE 函数 [系统] (第 250 页)
- OBJECT\_ID 函数 [系统] (第 257 页)
- OBJECT\_NAME 函数 [系统] (第 257 页)
- PROPERTY 函数 [系统] (第 267 页)
- PROPERTY\_DESCRIPTION 函数 [系统] (第 268 页)
- PROPERTY\_NAME 函数 [系统] (第 269 页)
- PROPERTY\_NUMBER 函数 [系统] (第 270 页)
- SUSER\_ID 函数 [系统] (第 323 页)

- SUSER\_NAME 函数 [系统] (第 324 页)
- USER\_ID 函数 [系统] (第 329 页)
- USER\_NAME 函数 [系统] (第 330 页)

## Adaptive Server 系统函数兼容性

并非所有 Adaptive Server 系统函数均在 SAP Sybase IQ 中实现。

**表 7. SAP Sybase IQ 中 Adaptive Server 系统函数的状态**

函数	状态
col_length	已实现
col_name	已实现
db_id	已实现
db_name	已实现
index_col	已实现
object_id	已实现
object_name	已实现
proc_role	始终返回 0
show_role	始终返回 NULL
tsequal	未实现
user_id	已实现
user_name	已实现
suser_id	已实现
suser_name	已实现
datalength	已实现
curunreservedpgs	未实现
data_pgs	未实现
host_id	未实现
host_name	未实现
lct_admin	未实现
reserved_pgs	未实现
rowcnt	未实现

函数	状态
<b>used_pgs</b>	未实现
<b>valid_name</b>	未实现
<b>valid_user</b>	未实现

*注释*

- 有些系统函数在 SAP Sybase IQ 中作为系统存储过程来实现的。
- db\_id、db\_name、datalength、suser\_id 和 suser\_name 函数作为内置函数来实现。

**连接属性**

检索特定连接属性的值或所有连接属性的值。

*示例*

可以使用 connection\_property 系统函数检索连接属性的值。以下语句返回已通过当前连接从文件中读取的页数：

```
select connection_property ( 'DiskRead' )
```

可以使用 sa\_conn\_properties 系统过程检索所有连接属性的值。

```
call sa_conn_properties
```

将针对每个属性为每个连接显示一个单独的行。

**另请参见**

- PROPERTY 函数 [系统] (第 267 页)
- PROPERTY\_NAME 函数 [系统] (第 269 页)
- PROPERTY\_NUMBER 函数 [系统] (第 270 页)
- CONNECTION\_PROPERTY 函数 [系统] (第 152 页)

**可用于服务器的属性**

检索特定服务器属性的值或所有服务器属性的值。

服务器属性应用于整个服务器。

Server Edition 属性返回 SQL Anywhere 版本，而不是 SAP Sybase IQ 版本。若要显示 SQL Anywhere 许可证信息，请使用 **sp\_iqlmconfig** 系统过程。

*示例*

可以使用 property 系统函数检索服务器属性的值。以下语句返回用于保存主堆的高速缓存页数：

```
select property ( 'MainHeapPages' ) from iq_dummy
```



可以使用 `sa_eng_properties` 系统过程检索所有服务器属性的值。

```
call sa_eng_properties
```

### 另请参见

- `sp_iqlmconfig` 过程 (第 463 页)
- `PROPERTY` 函数 [系统] (第 267 页)
- `PROPERTY_NAME` 函数 [系统] (第 269 页)
- `PROPERTY_NUMBER` 函数 [系统] (第 270 页)
- `CONNECTION_PROPERTY` 函数 [系统] (第 152 页)

### 连接属性列表

连接属性可用于 SAP Sybase IQ 数据库的每个连接。连接属性名称不区分大小写。

#### 连接属性

属性名称	说明
<code>allow_nulls_by_default</code>	返回一个值，指示是否允许创建时未指定 NULL 或 NOT NULL 的列中包含空值。
<code>allow_read_client_file</code>	返回一个值，指示数据库服务器是否允许读取客户端计算机上的文件。
<code>allow_snapshot_isolation</code>	返回一个值，指示是启用还是禁用快照隔离。
<code>allow_write_client_file</code>	返回一个值，指示数据库服务器是否允许向客户端计算机写入文件。
<code>ansi_blanks</code>	返回一个值，指示何时在客户端截断字符数据。
<code>ansi_close_cursors_on_rollback</code>	返回一个值，指示是否在执行 ROLLBACK 时关闭通过 WITH HOLD 打开的游标。
<code>ansi_permissions</code>	返回一个值，指示是否针对 DELETE 和 UPDATE 语句检查特权。
<code>ansi_substring</code>	返回一个值，指示在为起始参数或长度参数提供了负值时 SUBSTRING (SUBSTR) 函数的行为。
<code>ansi_update_constraints</code>	返回一个值，指示允许的更新范围。
<code>ansinull</code>	返回一个值，指示如何解释空值。

属性名称	说明
AppInfo	<p>返回有关建立连接的客户端的信息。对于 HTTP 连接，这包括有关浏览器的信息。对于使用较旧版本 jConnect 或 Sybase Open Client 的连接，信息可能不完整。</p> <p>API 值可为 DBLIB、ODBC、OLEDB、ADO.NET、iAnywhereJDBC、PHP、PerlDBD 或 DBEXPRESS。</p>
ApproximateCPU-Time	<p>返回由给定连接累计的 CPU 时间量的估计值，单位为秒。该值与实际值的最大差异可达 50%，不过返回值的典型变动范围是 5-10%。在多处理器计算机上，每个 CPU（或是超线程或内核）都会累计时间，因此所有连接的累计时间总和可能大于所经过的时间。在 Windows 和 Linux 上支持此属性。</p>
auditing	<p>如果将 PUBLIC.auditing 选项设置为 On，则返回 On。否则，返回 Off。</p> <p>如果将 auditing 选项设置为 On，而将 conn_auditing 选项设置为 Off，则即使未对当前连接进行审计，auditing 连接属性仍将返回 On。</p>
auditing_options	<p>此属性保留供系统使用。请不要更改此选项的设置。</p>
Authenticated	<p>如果应用程序发送了有效的连接验证字符串，则返回 Yes。如果应用程序尚未发送有效的连接验证字符串，则返回 No。</p>
AuthType	<p>返回连接时所使用的验证类型。返回值可能是 Standard、Integrated、Kerberos、LDAPUA 或空字符串。对于内部连接以及使用 AUTHORIZATION OFF 的 HTTP 服务连接，将返回空字符串。</p>
auto_commit_on_create_local_temp_index	<p>返回 auto_commit_on_create_local_temp_index 选项的设置。</p>
background_priority	<p>不建议使用此属性。返回一个值，指示当前连接对其它连接的性能的影响程度。</p>
BlockedOn	<p>如果当前连接未被阻塞，则返回零；如果当前连接被阻塞，则返回由于锁定冲突而阻塞连接的连接号。</p>
blocking	<p>返回一个值，指示数据库服务器对锁定冲突的响应行为。</p>
blocking_others_timeout	<p>返回在回退当前连接之前另一个连接可以在当前连接的行锁和表锁上阻塞的时间长度。</p>
blocking_timeout	<p>返回事务为获得锁而等待的时间长度（以毫秒为单位）。</p>

属性名称	说明
BytesReceived	返回在客户端/服务器通信期间所收到的字节数。此值对于 HTTP 和 HTTPS 连接会进行更新。
BytesReceivedUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所收到的字节数。在禁用压缩的情况下，此值与 BytesReceived 的值相同。
BytesSent	返回在客户端/服务器通信期间所发送的字节数。此值对于 HTTP 和 HTTPS 连接会进行更新。
BytesSentUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间将发送的字节数。在禁用压缩的情况下，此值与 BytesSent 的值相同。
CacheHits	返回成功读取高速缓存的次数。
CacheRead	返回已在高速缓存中查找的数据库页数。
CacheReadIndInt	返回已从高速缓存中读取的索引内部节点页数。
CacheReadIndLeaf	返回已从高速缓存中读取的索引叶页数。
CacheReadTable	返回已从高速缓存中读取的表页数。
CacheReadWorkTable	返回高速缓存工作表读取次数。
CarverHeapPages	返回供短期使用（如查询优化）的堆页数。
chained	返回在缺少 BEGIN TRANSACTION 语句时使用的事务模式。
CharSet	返回连接所使用的 CHAR 字符集。此属性具有一些在查询属性值时可以指定的扩展项。
checkpoint_time	返回数据库服务器在不执行检查点操作的情况下运行的最长时间（以分钟为单位）。
cis_option	如果远程数据访问的调试信息出现在数据库服务器消息窗口中，则返回 7；如果远程数据访问的调试信息不出现在数据库服务器消息窗口中，则返回 0。
cis_rowset_size	返回每次读取从远程服务器返回的行数。
ClientLibrary	对于 jConnect 连接，返回 jConnect；对于 Sybase Open Client 连接，返回 CT_Library；对于 HTTP 连接，返回 None；对于 ODBC、嵌入式 SQL、OLE DB、ADO.NET 和 SAP Sybase IQ JDBC 驱动程序连接，返回 CmdSeq。

属性名称	说明
ClientNodeAddress	返回客户端/服务器连接中客户端的节点。当客户端和服务端均在一台计算机上时，将会返回一个空字符串。这是 NodeAddress 属性的同义词。 若当前执行的请求为事件处理程序的一部分，则该属性返回 NA。
ClientPort	返回客户端的 TCP/IP 端口号；或者如果连接不是 TCP/IP 连接，则返回 0。
ClientStmtCache-Hits	返回因客户端语句高速缓存而不需要对此连接执行的准备的数目。如果禁用了客户端语句高速缓存，则为必需的附加准备数。
ClientStmtCache-Misses	返回客户端语句高速缓存中为此连接重新准备的语句的数目。即为由于模式更改、数据库选项设置或 DROP VARIABLE 语句等原因而使本打算重用的某个高速缓存的语句无法重用的次数。
close_on_endtrans	返回 On 或 Off 以指示是否在事务结束时关闭游标。
collect_statistics_on_dml_updates	返回 On 或 Off 以指示是否在执行数据修改 DML 语句（如 INSERT、DELETE 和 UPDATE）期间收集统计信息。
Commit	返回已处理的提交请求数。
CommLink	返回连接的通信链接。这是 SAP Sybase IQ 所支持的网络协议之一，或者 local，如果为相同计算机连接。 若当前执行的请求为事件处理程序的一部分，则该属性返回 NA。
CommNetworkLink	返回连接的通信链接。这是 SAP Sybase IQ 所支持的网络协议之一。值包括 SharedMemory 和 TCPIP。CommNetworkLink 属性总是返回链接的名称，不管它是否为相同计算机。 若当前执行的请求为事件处理程序的一部分，则该属性返回 NA。
CommProtocol	对于 Sybase Open Client 和 jConnect 连接，返回 TDS；对于 HTTP 连接，返回 HTTP；对于 HTTPS 连接，返回 HTTPS；对于 ODBC、嵌入式 SQL、OLE DB、ADO.NET 和 SAP Sybase IQ JDBC 驱动程序连接，返回 CmdSeq。

属性名称	说明
Compression	返回 On 或 Off 以指示是否对连接启用通信压缩。 若当前执行的请求为事件处理程序的一部分，则该属性返回 NA。
conn_auditing	如果为连接启用了审计，则即使将 auditing 选项设置为 Off，也会返回 On。
ConnectedTime	返回已被连接的一项连接的总时间长度（以秒为单位）。
connection_authentication	返回用于验证客户端的字符串。修改数据库前需要进行验证。
continue_after_raiserror	返回 On 或 Off 以指示每当遇到 RAISERROR 语句时，是否停止执行过程或触发器。
conversion_error	返回 On 或 Off 以指示在从数据库读取信息时报告数据类型转换失败。
cooperative_commit_timeout	返回数据库服务器在写入磁盘之前等待其它连接填充日志页的时间（以毫秒为单位）。
cooperative_commits	返回 On 或 Off 以指示何时将提交写入磁盘。
CurrentLineNumber	返回连接正在执行的过程或复合语句的当前行号。可使用 CurrentProcedure 属性标识该过程。如果该行是来自客户端的复合语句的一部分，则返回一个空字符串。
CurrentProcedure	返回连接当前正在执行的过程的名称。如果该连接正在执行嵌套过程调用，则该名称为当前过程的名称。如果未在执行任何过程，则返回一个空字符串。
Cursor	返回数据库服务器当前维护的声明游标数。
CursorOpen	返回数据库服务器当前维护的打开游标数。
database_authentication	返回用于验证数据库的字符串。需要先对已验证的数据库服务器进行验证，才能修改数据库。
date_format	返回一个字符串，指示从数据库检索的日期格式。
date_order	返回一个字符串，指示如何设置日期格式。
db_publisher	返回数据库发布者的用户 ID。
DBNumber	返回数据库的 ID 编号。

属性名称	说明
debug_messages	返回 On 或 Off 以指示是否执行包含 DEBUG ONLY 子句的 MESSAGE 语句。
dedicated_task	返回 On 或 Off 以指示请求处理任务是否专门用于处理连接请求。
default_dbspace	返回缺省 dbspace 的名称，如果尚未指定缺省的 dbspace，则返回一个空字符串。
default_timestamp_increment	返回一个值（以毫秒为单位），该值将被增加到类型为 TIMESTAMP 的列中以使该列中的值保持唯一。
delayed_commit_timeout	返回数据库服务器在执行 COMMIT 后等待将控制返回到应用程序的时间（以毫秒为单位）。
delayed_commits	返回 On 或 Off 以指示数据库服务器在执行 COMMIT 后何时将控制返回到应用程序。
disk_sandbox	返回 On 或 Off，以指示数据库的读写文件操作是否局限于主数据库文件所在的目录。
DiskRead	返回已从磁盘读取的页数。
DiskReadHint	返回磁盘读取提示数。
DiskReadHintPages	返回磁盘读取提示页数。
DiskReadIndInt	返回已从磁盘读取的索引内部节点页数。
DiskReadIndLeaf	返回已从磁盘读取的索引叶页数。
DiskReadTable	返回已从磁盘读取的表页数。
DiskReadWorkTable	返回磁盘工作表读取数。
DiskSyncRead	返回同步发出的磁盘读取数。
DiskSyncWrite	返回同步发出的写入数。
DiskWaitRead	返回数据库服务器等待异步读取的次数。
DiskWaitWrite	返回数据库服务器等待异步写入的次数。
DiskWrite	返回已写入磁盘的修改页数。
DiskWriteHint	返回磁盘写入提示数。

属性名称	说明
DiskWriteHintPages	返回磁盘写入提示页数。
divide_by_zero_error	如果被零除导致错误，返回 On；如果被零除不导致错误，则返回 Off。
Encryption	返回一个值，指示是否对连接加密。
escape_character	此属性保留供系统使用。请不要更改此选项的设置。
EventName	如果连接正在运行事件处理程序，则返回相关事件的名称。否则，返回一个空字符串。
exclude_operators	此属性保留供系统使用。请不要更改此选项的设置。
ExprCacheAbandons	返回表达式高速缓存因命中率太低而被放弃的次数。
ExprCacheDropsToReadOnly	返回表达式高速缓存因命中率低而降到只读状态的次数。
ExprCacheEvicts	返回从表达式高速缓存中的退出次数。
ExprCacheHits	返回表达式高速缓存中的命中次数。
ExprCacheInserts	返回在表达式高速缓存中插入的值的个数。
ExprCacheLookups	返回在表达式高速缓存中执行的查找次数。
ExprCacheResumesOfReadWrite	返回表达式高速缓存因命中率上升而恢复读写状态的次数。
ExprCacheStarts	返回表达式高速缓存的启动次数。
extended_join_syntax	如果允许使用具有多表连接的重复相关名语法的查询，则返回 On，如果将此类查询作为错误报告，则返回 Off。
extern_login_credentials	返回在尝试远程连接时是使用已登录用户的外部登录证书还是使用有效用户的外部登录证书。
fire_triggers	如果在数据库中触发触发器，则返回 On；否则，返回 Off。
first_day_of_week	返回用于表示每周第一天的数字，其中 7=星期日，1=星期一。
for_xml_null_treatment	当在查询中使用了 FOR XML 子句时，如果从结果中省略包含空值的元素和属性，则返回 Omit；如果为空值生成空元素或属性，则返回 Empty。

属性名称	说明
force_view_creation	此属性保留供系统使用。请不要更改此选项的设置。
FullCompare	返回已执行的超过索引中散列值的比较次数。
GetData	返回 GETDATA 请求数。
global_database_id	返回用于以 DEFAULT GLOBAL AUTOINCREMENT 创建的列的起始值。
HashForcedPartitions	返回因内存争用而强制散列运算符进行分区的次数。
HashRowsFiltered	返回位矢量过滤器拒绝的探测行数。
HashRowsPartitioned	返回写入散列工作表的行数。
HashWorkTables	返回为基于散列的操作而创建的工作表数。
HeapsCarver	返回供短期使用（如查询优化）的堆数。
HeapsLocked	返回高速缓存中当前锁定的可重定位堆数。
HeapsQuery	返回用于查询处理（散列和排序操作）的堆数。
HeapsRelocatable	返回可重定位堆数。
http_connection_pool_basesize	返回数据库连接的额定阈值大小。
http_connection_pool_timeout	返回未使用的连接存储在连接池中的最大时间长度。
http_session_timeout	返回当前 HTTP 会话超时值（以分钟为单位）。
HttpServiceName	返回当前 HTTP 请求的服务名称入口点。该属性对于错误报告和控制流很有用。当从中选择此属性的存储过程并非源自于 HTTP 请求时，或者如果连接当前处于不活动状态或等待继续进行 HTTP 会话，将会返回一个空字符串。
IdleTimeout	返回连接的空闲超时值。 若当前执行的请求为事件处理程序的一部分，则该属性返回 NA。



属性名称	说明
IndAdd	返回已添加到索引的条目数。
IndLookup	返回已在索引中查找的条目数。
integrated_server_name	返回域控制器服务器的名称，该服务器用于查找集成登录的 Windows 用户组成员资格。
IsDebugger	返回 Yes 或 No 以区分出正用于运行 SAP Sybase IQ 调试程序的连接。如果当前连接编号对应于某调试程序的连接编号，此值为 Yes，否则为 No。
isolation_level	返回连接的隔离级别（0、1、2、3、快照、语句快照或只读语句快照）。
java_class_path	返回在其中搜索类的附加目录或 JAR 文件的列表。
java_location	返回数据库的 Java VM 路径（如果已指定）。
java_main_userid	不建议使用此属性。
java_vm_options	返回数据库服务器启动 Java VM 时使用的命令行选项。
Language	返回地区语言。
LastCommitRedoPos	在连接将最后一个 COMMIT 操作写入到事务日志后返回重做日志位置。
LastIdle	返回请求间隔时间。
LastPlanText	返回在连接上执行的最后一个查询的长文本计划。通过设置 sa_server_option 系统过程的 RememberLastPlan 选项或使用 -zp 服务器选项控制对最后一个计划的记忆情况。
LastReqTime	返回指定连接的最后一个请求开始的时间。此属性可以为内部连接（如事件）返回空字符串。

属性名称	说明
LastStatement	<p>返回最近为当前连接准备的 SQL 语句。</p> <p>当准备了语句时，将会设置 LastStatement 值；当删除了语句时，将会清除该值。每个连接仅会记忆一个语句字符串。</p> <p>如果 sa_conn_activity 对某个连接报告了一个非空值，它很可能为该连接当前执行的语句。假如该语句已完成，则很有可能已丢弃了它并清除了属性值。如果某个应用程序准备了多条语句并保有其语句句柄，则 LastStatement 值不反映连接当前进行的操作。</p> <p>如果启用客户端语句高速缓存并且重用了某个高速缓存的语句，则此属性返回空字符串。</p>
LivenessTimeout	<p>返回当前连接的活动超时期。</p> <p>若当前执行的请求为事件处理程序的一部分，则该属性返回 NA。</p>
lock_rejected_rows	此属性保留供系统使用。请不要更改此选项的设置。
LockCount	返回连接所持有的锁的个数。
LockIndexID	返回锁定索引的标识符。
LockName	返回一个 64 位无符号整数值，表示连接正在等待的锁。
LockRowID	返回锁定行的标识符。
LockTableOID	如果连接未被阻塞或连接与调用 CONNECTION_PROPERTY 连接位于不同的数据库，则返回零。否则，此值是与该连接正在等待的锁相应的那个表的对象 ID。可使用该对象 ID 通过 SYSTAB 系统视图查找表信息。
log_deadlocks	如果报告死锁信息，则返回 On；否则，返回 Off。
LogFreeCommit	返回重做无偿提交数。当请求提交事务日志而该日志已被写入（从而提交是无偿完成的）时，便会发生重做无偿提交。
login_mode	返回 Standard、Integrated、Kerberos 或 LDAPUA 中的一个或多个以指示支持的登录类型。
login_procedure	返回用于在启动时设置兼容性选项的存储过程的名称。
LoginTime	返回建立连接的日期和时间。
LogWrite	返回已写入事务日志的页数。

属性名称	说明
materialized_view_optimization	返回一个值，指示在查询优化期间是否使用实例化视图： Disabled Fresh Stale <i>N</i> Minute[s] <i>N</i> Hour[s] <i>N</i> Day[s] <i>N</i> Week[s] <i>N</i> Month[s]
max_client_statements_cached	返回由客户端高速缓存的语句数。
max_cursor_count	返回一个值，指定连接一次最多可以使用的游标数。
max_hash_size	不建议使用此属性。
max_plans_cached	返回一个值，指定最多要在高速缓存中存储的执行计划数。
max_priority	返回一个值，指示连接可以具有的最高优先级级别。
max_query_tasks	返回数据库服务器在处理查询时最多可以使用的请求数。
max_recursive_iterations	返回一个值，指定递归公用表表达式最多可以进行的迭代次数。
max_statement_count	返回一个值，指定连接最多可以同时使用的经过准备的语句数。
max_temp_space	返回一个值，指示连接可用的最大临时文件空间量。
MessageReceived	返回由导致 WAITFOR 语句中断的 MESSAGE 语句所生成的字符串。否则，返回一个空字符串。
min_password_length	返回数据库中新口令的最小长度。
min_role_admins	返回角色所需的最少管理员数。请参见 min_role_admins option。

属性名称	说明
Name	<p>返回当前连接的名称。</p> <p>可使用 <code>ConnectionName (CON)</code> 连接参数来指定连接的名称。</p> <p>以下名称用于数据库服务器建立的临时连接：</p> <ul style="list-style-type: none"> <li>INT:ApplyRecovery</li> <li>INT:BackupDB</li> <li>INT:Checkpoint</li> <li>INT:Cleaner</li> <li>INT:CloseDB</li> <li>INT&gt;CreateDB</li> <li>INT&gt;CreateMirror</li> <li>INT:DelayedCommit</li> <li>INT:DiagRcvr</li> <li>INT:DropDB</li> <li>INT:EncryptDB</li> <li>INT:Exchange</li> <li>INT:FlushMirrorLog</li> <li>INT:FlushStats</li> <li>INT:HTTPReq</li> <li>INT:PromoteMirror</li> <li>INT:PurgeSnapshot</li> <li>INT:ReconnectMirror</li> <li>INT:RecoverMirror</li> <li>INT:RedoCheckpoint</li> <li>INT:RefreshIndex</li> <li>INT:ReloadTrigger</li> <li>INT:RenameMirror</li> <li>INT:RestoreDB</li> <li>INT:StartDB</li> <li>INT:VSS</li> </ul>
NcharCharSet	<p>返回连接所使用的 <code>NCHAR</code> 字符集。此属性具有一些在查询属性值时可以指定的扩展项。</p>
nearest_century	<p>返回一个值，指示在字符串到日期转换中如何对两位数年份进行解释。</p>
NodeAddress	<p>返回客户端/服务器连接中客户端的节点。当客户端和服务端均在一台计算机上时，将会返回一个空字符串。</p>

属性名称	说明
non_keywords	返回为使其能够用作标识符而关闭的关键字列表（如果有）。
Number	返回当前连接的连接 ID（数字）。
NumLocalTempTables	返回连接使用的本地临时表的数量。当局部临时表被删除或超出范围时，在下一个 COMMIT 之前仍将此表视为正在使用中。
odbc_describe_binary_as_varbinary	如果 SAP Sybase IQ ODBC 驱动程序将 BINARY 和 VARBINARY 列都描述为 SQL_BINARY，则返回 Off；如果该 ODBC 驱动程序将 BINARY 和 VARBINARY 列都描述为 SQL_VARBINARY，则返回 On。
odbc_distinguish_char_and_varchar	如果将 CHAR 列描述为 SQL_VARCHAR，则返回 Off；如果将 CHAR 列描述为 SQL_CHAR，则返回 On。
oem_string	返回数据库文件头页中存储的字符串。
on_charset_conversion_failure	返回 Ignore、Warning 或 Error 之一，以指示在字符集转换期间遇到错误时的行为。
on_tsql_error	返回 Stop、Conditional 或 Continue 之一，以指示在执行存储过程或 T-SQL 批处理期间遇到错误时的行为。
optimization_goal	返回 First-row 或 All-rows 之一，以指示如何对查询处理进行优化。
optimization_level	返回一个介于 0 到 15 的值。此数用于控制 SAP Sybase IQ 查询优化程序为查找 SQL 语句的访问计划而付出的努力程度。
optimization_workload	返回一个值，指示 SAP Sybase IQ 查询优化程序为查找 SQL 语句的访问计划而付出的努力程度。
OSUser	返回与客户端进程相关联的操作系统用户名。如果客户端进程模拟另一个用户（或者设置的 ID 位在 Unix 上设置），则返回被模拟的用户名。对于 10.0.1 版及更早版本的客户端，以及 HTTP 和 TDS 客户端，将返回空字符串。
PacketSize	返回连接所使用的包大小（以字节为单位）。 若当前执行的请求为事件处理程序的一部分，则该属性返回 NA。
PacketsReceived	返回所收到的客户端/服务器通信包数。此值对于 HTTP 和 HTTPS 连接不会进行更新。

属性名称	说明
PacketsReceivedUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所收到的包数。（在禁用压缩的情况下，该值与 PacketsReceived 的值相同。）
PacketsSent	返回所发送的客户端/服务器通信包数。此值对于 HTTP 和 HTTPS 连接不会进行更新。
PacketsSentUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所发送的包数。（在禁用压缩的情况下，该值与 PacketsSent 的值相同。）
ParentConnection	返回已创建临时连接以执行数据库操作（如执行备份或创建数据库）的连接的连接 ID。对于其它类型的连接，此属性返回 NULL。
pinned_cursor_percent_of_cache	返回可用于固定游标的高速缓存百分比。
post_login_procedure	返回结果集中包含消息的过程名，这些消息应在用户连接时由应用程序显示。
precision	返回数字位数和数值精度设置。
prefetch	如果不进行预取，返回 Off；如果进行预取（但游标类型为 SENSITIVE 或查询中包含代理表时则不进行预取），返回 Conditional；如果即使对于 SENSITIVE 游标类型和包含代理表的游标也进行预取，返回 Always。
Prepares	返回为进行连接所执行的语句准备次数。
PrepStmt	返回数据库服务器当前维护的经过准备的语句数。
preserve_source_format	如果在系统表中保存过程、触发器、视图和事件处理程序的原始源代码，则返回 On；否则，返回 Off。
prevent_article_pkey_update	如果不允许对发布中所涉及的表的主键列进行更新，则返回 On；否则，返回 Off。
priority	返回一个值，指示连接的优先级级别。
Progress	返回关于查询已运行时长的信息。例如： 43% (9728 of 22230 pages) complete after 00:00:05; estimated 00:00:06 remaining 此属性具有一些在查询属性值时可以指定的扩展项。

属性名称	说明
progress_messages	返回 progress_messages 选项的值。
query_mem_timeout	返回 query_mem_timeout 选项的值。
QueryBypassed	返回不使用优化程序所优化的请求数。
QueryBypassedCosted	返回优化程序旁路使用开销计算所处理的请求数。
QueryBypassed-Heuristic	返回优化程序旁路使用试探法所处理的请求数。
QueryBypassedOptimized	返回最初由优化程序旁路处理、随后由 SAP Sybase IQ 优化程序完全优化的请求数。
QueryCachedPlans	返回当前为连接高速缓存的查询执行计划数。
QueryCachePages	返回用于高速缓存执行计划的高速缓存页数。
QueryDescribedBy-pass	返回优化程序旁路所处理的描述请求数。
QueryDescribedOptimizer	返回优化程序所处理的描述请求数。
QueryHeapPages	返回用于查询处理（散列和排序操作）的高速缓存页数。
QueryJHToJNLOptUsed	返回散列连接转换为嵌套循环连接的次数。
QueryLowMemory-Strategy	返回服务器由于内存不足的情况而在执行期间改变其执行计划的次数。策略可能会因当前可用内存比优化程序预计的少或执行计划所需的内存比优化程序预计的多而发生变化。
QueryMemActive-Curr	返回主动使用查询内存的请求数。
QueryMemGrant-Failed	返回某一请求等待但未能获得查询内存的总次数。
QueryMemGrant-Granted	返回当前授予请求的页数。
QueryMemGrantRequested	返回任一请求尝试获取查询内存的总次数。

属性名称	说明
QueryMemGrantWait- ted	返回任一请求等待查询内存的总次数。
QueryMemGrant- Waiting	返回等待查询内存的当前请求数。
QueryOpened	返回要执行的 <b>OPEN</b> 请求数。
QueryOptimized	返回经过完全优化的请求数。
QueryReused	返回已从计划高速缓存中重用的请求数。
QueryRowsFetched	使用顺序扫描或索引扫描为此连接返回已从基表中读取的行数。
QueryRowsMateri- alized	返回查询处理期间写入工作表的行数。
quoted_identifier	如果将括在双引号中的字符串解释为标识符，则返回 <b>On</b> ；如果将其解释为文字字符串，则返回 <b>Off</b> 。
read_past_deleted	如果顺序扫描在隔离级别 1 和 2 跳过未提交的已删除行，则返回 <b>On</b> ；如果顺序扫描在隔离级别 1 和 2 阻塞于未提交的已删除行，则返回 <b>Off</b> 。
recovery_time	返回数据库服务器从系统故障中恢复最长将要花费的时间（以分钟为单位）。
RecursiveItera- tions	返回递归联合的迭代次数。
RecursiveItera- tionsHash	返回递归散列连接使用散列策略的次数。
RecursiveItera- tionsNested	返回递归散列连接使用嵌套循环策略的次数。
RecursiveJNLMiss- es	返回递归散列连接的索引探测高速缓存未命中次数。
RecursiveJNLP- robes	返回递归散列连接尝试索引探测的次数。
remote_idle_time- out	返回 <b>Web</b> 服务客户端过程和函数允许的非活动时间（以秒为单位）。
replicate_all	仅供内部使用。



属性名称	说明
ReqCountActive	返回已处理的请求数；如果将 RequestTiming 服务器属性设置为 Off，则返回 NULL。
ReqCountBlockCon- tention	返回连接等待原子访问的次数；如果未指定 -zt 选项，则返回 NULL。
ReqCountBlockIO	返回连接等待 I/O 完成的次数；如果未指定 -zt 选项，则返回 NULL。
ReqCountBlockLock	返回连接等待锁的次数；如果未指定 -zt 选项，则返回 NULL。
ReqCountUnsched- uled	返回连接等待调度的次数；如果未指定 -zt 选项，则返回 NULL。
ReqStatus	<p>返回请求状态。它可以是以下各值之一：</p> <ul style="list-style-type: none"> <li>• <b>Idle</b> - 连接当前未处理请求。</li> <li>• <b>Unscheduled*</b> - 连接有工作要做并且正在等待可用的数据库服务器工作线程。</li> <li>• <b>BlockedIO*</b> - 连接被阻塞，正在等待 I/O。</li> <li>• <b>BlockedContention*</b> - 连接被阻塞，正在等待访问共享数据库服务器数据结构。</li> <li>• <b>BlockedLock</b> - 连接被阻塞，正在等待锁定的对象。</li> <li>• <b>Executing</b> - 连接正在执行请求。</li> </ul> <p>仅当已使用 -zt 服务器选项为数据库服务器开启了请求计时信息记录功能时，才会返回标有星号 (*) 的值。如果未记录请求计时信息（缺省设置），则会将这些值报告为 Executing。</p>
ReqTimeActive	返回处理请求所用的时间长度（以秒为单位）；如果未指定 -zt 选项，则返回 NULL。
ReqTimeBlockCon- tention	返回等待原子访问所用的时间长度（以秒为单位）；如果将 RequestTiming 服务器属性设置为 Off，则返回 NULL。
ReqTimeBlockIO	返回等待 I/O 完成所用的时间长度（以秒为单位）；如果未指定 -zt 选项，则返回 NULL。
ReqTimeBlockLock	返回等待锁所用的时间长度（以秒为单位）；如果未指定 -zt 选项，则返回 NULL。
ReqTimeUnsched- uled	返回未调度的时间长度；如果未指定 -zt 选项，则返回 NULL。
ReqType	返回上次请求的类型。如果连接已由连接池进行了高速缓存，则其 ReqType 值为 CONNECT_POOL_CACHE。

属性名称	说明
request_timeout	返回单个请求可以运行的最长时间。
RequestsReceived	返回客户端/服务器通信请求数或往返次数。它与 <b>PacketsReceived</b> 的不同之处在于：多包请求将计为一个请求，并且不包括活动包。
reserved_keywords	返回为数据库启用的非缺省保留关键字。
return_date_time_as_string	如果将 <b>DATE</b> 、 <b>TIME</b> 和 <b>TIMESTAMP</b> 值作为字符串返回给应用程序，则返回 <b>On</b> ；如果将其作为 <b>DATE</b> 或 <b>TIME</b> 数据类型返回，则返回 <b>Off</b> 。
Rlbk	已处理的回退请求数。
rollback_on_deadlock	当在 <b>UPDATE</b> 或 <b>DELETE</b> 之后执行参照完整性操作时，返回 <b>After</b> ；如果在 <b>UPDATE</b> 或 <b>DELETE</b> 之前执行参照完整性操作，则返回 <b>Before</b> 。
RollbackLogPages	返回回退日志中的页数。
row_counts	如果行计数始终准确，则返回 <b>On</b> ；如果行计数通常为估计值，则返回 <b>Off</b> 。
scale	返回连接的十进制和数值小数位数。
secure_feature_key	存储用于启用和禁用数据库服务器功能的键值。选择此属性值始终都会返回一个空字符串。
ServerNodeAddress	返回客户端/服务器连接中服务器的节点。当客户端和服务器均在一台计算机上时，将会返回一个空字符串。 若当前执行的请求为事件处理程序的一部分，则该属性返回 <b>NA</b> 。
ServerPort	返回数据库服务器的 <b>TCP/IP</b> 端口号或 <b>0</b> 。
SessionCreateTime	返回创建 <b>HTTP</b> 会话的时间。
SessionID	返回连接的会话 <b>ID</b> （如果存在），否则，返回一个空字符串。
SessionLastTime	返回 <b>HTTP</b> 会话的上次请求时间。
SessionTimeout	返回 <b>HTTP</b> 会话在非活动期间持续的时间（单位为分钟）。
SnapshotCount	返回与连接相关联的快照数。
sort_collation	如果 <b>ORDER BY</b> 子句保持不变，则返回 <b>Internal</b> ；否则，返回归类名或归类 <b>ID</b> 。

属性名称	说明
SortMergePasses	返回在排序过程中使用的合并遍数。
SortRowsMaterialized	返回写入排序工作表的行数。
SortRunsWritten	返回在排序过程中写入的排序运行数。
SortSortedRuns	返回在运行形成过程中创建的排序运行数。
SortWorkTables	返回为排序创建的工作表数。
sql_flagger_error_level	<p>返回以下各值之一，指示将哪个不属于指定 SQL/2003 集的 SQL 标记为错误：</p> <ul style="list-style-type: none"> <li>• <b>E</b> - 标记非入门级 SQL/2003 语法的语法</li> <li>• <b>I</b> - 标记非中级 SQL/2003 语法的语法</li> <li>• <b>F</b> - 标记非完整 SQL/2003 语法的语法</li> <li>• <b>W</b> - 允许所有支持的语法</li> </ul>
sql_flagger_warning_level	<p>返回以下各值之一，指示将哪个不属于指定 SQL/2003 集的 SQL 标记为警告：</p> <ul style="list-style-type: none"> <li>• <b>E</b> - 标记非入门级 SQL/2003 语法的语法</li> <li>• <b>I</b> - 标记非中级 SQL/2003 语法的语法</li> <li>• <b>F</b> - 标记非完整 SQL/2003 语法的语法</li> <li>• <b>W</b> - 允许所有支持的语法</li> </ul>
st_geometry_asbinary_format	返回一个值，该值指示如何将空间值从几何格式转换为二进制格式。
st_geometry_as_text_format	返回一个值，该值指示如何将空间值从几何形式转换为文本形式。
st_geometry_asxml_format	返回一个值，该值指示如何将空间值从几何形式转换为 xml 形式。
st_geometry_describe_type	返回一个值，该值指示已描述空间值。
st_geometry_interpolation	返回 ST_CircularString 几何的内插值设置。
st_geometry_on_invalid	返回一个值，该值指示几何表面校验失败时采取的行为。

属性名称	说明
StatementDe-scribes	返回各个 <b>DESCRIBE</b> 请求所处理的语句总数。
StatementPostAn-notates	返回由语义查询转换阶段所处理的语句数。
StatementPostAn-notatesSimple	返回由语义查询转换阶段处理但跳过了某些语义转换的语句数。
StatementPostAn-notatesSkipped	返回已完全跳过语义查询转换阶段的语句数。
string_rtrunca-tion	如果在截断字符串时产生错误，则返回 <b>On</b> ；如果不产生错误且字符串是以静默方式截断的，则返回 <b>Off</b> 。
subsume_row_locks	如果数据库服务器获得表的各个行锁，则返回 <b>On</b> ；否则，返回 <b>Off</b> 。
suppress_tds_de-bugging	如果 <b>TDS</b> 调试信息出现在数据库服务器消息窗口中，则返回 <b>Off</b> ；如果调试信息不出现在数据库服务器消息窗口中，则返回 <b>On</b> 。
synchronize_mir-ror_on_commit	如果在提交时同步数据库镜像服务器，则返回 <b>On</b> ；否则，返回 <b>Off</b> 。
tds_empty_string_is_null	如果对于 <b>TDS</b> 连接将空字符串作为 <b>NULL</b> 返回，则返回 <b>On</b> ；如果对于 <b>TDS</b> 连接返回包含单个空字符的字符串，则返回 <b>Off</b> 。
temp_space_limit_check	如果数据库服务器检查连接可用的临时空间量，则返回 <b>On</b> ；如果数据库服务器不检查连接可用的空间量，则返回 <b>Off</b> 。
TempFilePages	返回连接使用的临时文件页面的数量。
TempTablePages	返回用于临时表的临时文件中的页数。
time_format	返回从数据库中检索的时间所使用的字符串格式。
time_zone_adjust-ment	返回为显示连接本地时间而必须对协调通用时间（ <b>Coordinated Universal Time</b> ，简称 <b>UTC</b> ）增加的分钟数。
timestamp_format	返回从数据库检索得来的时间戳的格式。
timestamp_with_time_zone_format	返回从数据库检索的 <b>TIMESTAMP WITH TIME ZONE</b> 值的格式。

属性名称	说明
TimeZoneAdjustment	返回为显示连接本地时间而必须对协调通用时间 (Coordinated Universal Time, 简称 UTC) 增加的分钟数。
TransactionStartTime	返回一个字符串, 其中包含在 COMMIT 或 ROLLBACK 之后首次修改数据库的时间; 如果自上次执行 COMMIT 或 ROLLBACK 以来未对数据库进行过任何修改, 则返回一个空字符串。
truncate_timestamp_values	如果 TIMESTAMP 值中使用的小数位数受限制, 则返回 On; 否则, 返回 Off。
trusted_certificates_file	当数据库服务器充当 LDAP 服务器的客户端时, 返回包含受信任证书颁发机构证书列表的文件。
tsql_outer_joins	如果可以在 DML 语句中使用 Transact-SQL 外连接, 则返回 On。
tsql_variables	如果可以使用 @ 符号代替冒号作为嵌入式 SQL 中主机变量名的前缀, 则返回 On; 否则, 返回 Off。
UncommitOp	返回未提交的操作数。
updatable_statement_isolation	返回当 isolation_level 选项设置为 Readonly-statement-snapshot (只读语句快照) 时, 可更新语句使用的隔离级别 (0、1、2 或 3)。
update_statistics	如果此连接能向统计信息调控器返回查询反馈, 则返回 On。当 update_statistics 选项设置为 Off 时, 统计信息调控器不会收到来自当前连接的查询反馈。
upgrade_database_capability	此属性保留供系统使用。请不要更改此选项的设置。
user_estimates	<p>返回以下各值之一, 这些值控制查询优化程序是考虑还是忽略查询谓语句中的选择性估计:</p> <ul style="list-style-type: none"> <li>• <b>Enabled</b> - 考虑所有由用户提供的选择性估计。也可以使用 On 开启此选项。</li> <li>• <b>Override-Magic</b> - 只有优化程序要不然会选择采取最后的手段, 即试探值 (也称魔术值), 才会考虑并使用用户选择性估计。</li> <li>• <b>Disabled</b> - 忽略所有用户估计, 并在没有其它可用的估计数据时使用魔术值。也可以使用 Off 关闭此选项。</li> </ul>
UserAppInfo	返回连接字符串中的 AppInfo 连接参数所指定的字符串。

属性名称	说明
UserDefinedCounterRate01	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器值随着时间的推移而发生的变化。
UserDefinedCounterRate02	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器值随着时间的推移而发生的变化。
UserDefinedCounterRate03	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器值随着时间的推移而发生的变化。
UserDefinedCounterRate04	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器值随着时间的推移而发生的变化。
UserDefinedCounterRate05	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器值随着时间的推移而发生的变化。
UserDefinedCounterRaw01	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器的绝对值。
UserDefinedCounterRaw02	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器的绝对值。
UserDefinedCounterRaw03	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器的绝对值。
UserDefinedCounterRaw04	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器的绝对值。
UserDefinedCounterRaw05	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器的绝对值。
UserID	返回连接的用户 ID。
UtilCmdsPermitted	返回 On 或 Off, 指示连接是否允许使用实用程序命令 (如 CREATE DATABASE、DROP DATABASE 和 RESTORE DATABASE)。

属性名称	说明
uuid_has_hyphens	该属性控制将唯一标识符值转换为字符串时所采用的格式。
verify_password_function	返回用于口令验证的函数名（如果已指定）。
wait_for_commit	如果数据库要在下一条 <b>COMMIT</b> 语句之后才会检查外键完整性，则返回 <b>On</b> 。否则，返回 <b>Off</b> ，并且在插入、更新或删除所有不是使用 <b>check_on_commit</b> 选项创建的外键时都会对其进行检查。
WaitStartTime	返回连接开始等待的时间（如果连接没有处于等待状态则返回一个空字符串）。
WaitType	返回等待的原因（如果可用）。WaitType 的可能值有： <ul style="list-style-type: none"> <li>• <b>lock</b> - 如果连接在等待锁定则返回此值。</li> <li>• <b>waitfor</b> - 如果连接在执行 <b>waitfor</b> 语句则返回此值。</li> <li>• <b>空字符串</b> - 如果连接没有处于等待状态，或等待的原因没有提供，则返回此值。</li> </ul>
webservice_name-space_host	返回要在生成的 <b>WSDL</b> 文档内用作 <b>XML</b> 命名空间的主机名（如果已指定）。
webservice_sessionid_name	返回会话标识符名称，该会话标识符名称供 <b>Web</b> 服务器用于确定是否正在使用会话管理。

## 各数据库可用的属性

您可以检索特定数据库属性的值或所有数据库属性的值。数据库属性适用于整个数据库。

服务器属性 **QueryBypassedCosted**、**QueryBypassedOptimized**、**QueryDescribedOptimizer** 和 **StatementPostAnnotatesSimple** 仅针对目录存储表的查询进行更新。

### 示例

可以使用 `db_property` 系统函数检索数据库属性的值。以下语句返回当前数据库的页面大小：

```
select db_property ( 'PageSize')
```

可以使用 `sa_db_properties` 系统过程检索所有数据库属性的值：

```
call sa_db_properties
```

### 另请参见

- **PROPERTY** 函数 [系统]（第 267 页）

- PROPERTY\_NAME 函数 [系统] (第 269 页)
- PROPERTY\_NUMBER 函数 [系统] (第 270 页)
- CONNECTION\_PROPERTY 函数 [系统] (第 152 页)

### 数据库服务器属性列表

#### 数据库服务器属性

属性名称	说明
ActiveReq	返回当前处理客户端请求的服务器工作线程数。
ApproximateCPUtime	返回由数据库服务器累计的 CPU 时间量的估计值，单位为秒。该值与实际值的最大差异可达 50%，不过返回值的典型变动范围是 5-10%。在多处理器计算机上，每个 CPU（或是超线程或内核）都会累计时间，因此所有连接的累计时间总和可能大于所经过的时间。在 Windows 和 Linux 上支持此属性。
AutoMultiProgrammingLevel	返回一个值，该值表示数据库服务器是否在自动调整它的进程并发水平。
AutoMultiProgrammingLevelStatistics	返回一个值，该值表示关于数据库服务器进程并发水平自动调整的消息是否显示在数据库服务器的消息日志中。
AvailIO	返回当前可用的 I/O 控制块数。
BuildChange	此属性保留供系统使用。请不要更改此属性的设置。
BuildClient	此属性保留供系统使用。请不要更改此属性的设置。
BuildProduction	如果数据库服务器是为了用于生产而编译的，则返回 Yes；如果数据库服务器为调试内部版本，则返回 No。
BuildReproducible	此属性保留供系统使用。请不要更改此属性的设置。
BytesReceived	返回在客户端/服务器通信期间所收到的字节数。此值对于 HTTP 和 HTTPS 连接会进行更新。
BytesReceivedUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所收到的字节数。（在禁用压缩的情况下该值与 BytesReceived 的值相同。）
BytesSent	返回在客户端/服务器通信期间所发送的字节数。此值对于 HTTP 和 HTTPS 连接会进行更新。



属性名称	说明
BytesSentUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所发送的字节数。（在禁用压缩的情况下该值与 BytesSent 的值相同。）
CacheAllocated	返回已为服务器数据结构分配的高速缓存页数。
CacheFile	返回用于存放数据库文件中数据的高速缓存页数。
CacheFileDirty	返回处于脏状态的高速缓存页数（需要写操作）。
CacheFree	返回未在使用的高速缓存页数。
CacheHits	返回数据库页查找次数。
CachePanics	返回高速缓存管理器未能找到要分配页的次数。
CachePinned	返回固定高速缓存页数。
CacheRead	返回高速缓存读取次数。
CacheReplacements	返回高速缓存中已替换的页数。
CacheScavenges	返回高速缓存管理器清除回收以分配页的次数。
CacheScavengeVisited	返回在清除回收以分配页期间访问的页数。
CacheSizingStatistics	如果服务器在重新调整高速缓存大小时显示高速缓存大小调整统计信息，则返回 Yes；否则，返回 No。
CarverHeapPages	返回供短期使用（如查询优化）的堆页数。
CharSet	返回数据库服务器所使用的 CHAR 字符集。
ClientStmtCacheHits	返回因客户端语句高速缓存而不需要进行的准备的数目。如果禁用了客户端语句高速缓存，则为必需的附加准备数。
ClientStmtCacheMisses	返回客户端语句高速缓存中重新准备的语句数。即为由于模式更改、数据库选项设置或 DROP VARIABLE 语句等原因而使本打算重用的某个高速缓存的语句无法重用的次数。
CollectStatistics	返回 Yes 或 No，指示数据库服务器是否正在收集性能统计信息。
CommandLine	返回用于启动数据库服务器的命令行参数。  如果数据库的加密密钥是使用 -ek 选项指定的，则该密钥在由此属性返回的值中将以星号字符串常量代替。

属性名称	说明
Commit	返回已处理的提交请求数。
CompactPlatformVer	返回 PlatformVer 属性的精简版本。
CompanyName	返回拥有此软件的公司的名称。
ConnectedTime	返回所有已连接到数据库服务器的连接总时间长度（以秒为单位）。  仅当连接请求完成时或连接断开时更新该值。因此，在数据库服务器中，不管是空闲还是忙于执行的连接，该值均滞后很长时间。该值包括由任何连接产生的时间，其中包括数据库事件和后台服务器连接（如数据库清理程序）。
ConnsDisabled	返回 Yes 或 No，指示用于禁用新连接的服务器选项的当前设置。
ConsoleLogFile	如果指定了 -o 选项，则返回用于记录数据库服务器消息的文件的名称；否则，返回一个空字符串。
ConsoleLogMaxSize	返回用于记录数据库服务器消息的文件的最大大小（以字节为单位）。
CurrentCacheSize	返回当前高速缓存大小（以千字节为单位）。
CurrentMultiProgrammingLevel	返回数据库服务器可同时处理的当前任务数。
Cursor	返回数据库服务器当前维护的声明游标数。
CursorOpen	返回数据库服务器当前维护的打开游标数。
DebuggingInformation	如果服务器显示用于进行排除故障的诊断消息，则返回 Yes；否则，返回 No。
DefaultCollation	返回如果没有明确指定，将用于新数据库的归类。
DefaultNcharCollation	返回服务器计算机上缺省 NCHAR 归类的名称（如果安装了 ICU，则为 UCA；否则为 UTF8BIN）。
DiskRead	返回磁盘读取次数。
DiskReadHintScatterLimit	返回对分散读取提示大小（以字节为单位）的强加限制。
DiskRetryRead	返回磁盘读取重试的次数。

属性名称	说明
DiskRetryReadScatter	返回分散读取的磁盘读取重试次数。
DiskRetryWrite	返回磁盘写入重试的次数。
DiskSandbox	返回 On 或 Off，以指示数据库的读写文件操作是否局限于主数据库文件所在的目录。
EventTypeDesc	返回与给定事件类型 ID 相关联的事件类型说明。
EventTypeName	返回与给定事件类型 ID 相关联的系统事件类型名称。
ExchangeTasks	返回当前用于并行执行查询的任务数。
ExchangeTasksCompleted	返回自数据库服务器启动以来已用于查询内并行机制的内部任务总数。
FipsMode	如果在启动数据库服务器时指定了 -fips 选项，则返回 Yes；否则，返回 No。
FirstOption	返回一个数字，表示与某个数据库选项相对应的第一个连接属性。
FreeBuffers	返回可用的网络缓冲区数。
FunctionMaxParms	返回最多可由函数指定的参数个数。函数通过 <i>function-number</i> 所指定的值进行标识，该值是一个正整数。例如： <pre>SELECT PROPERTY ( 'FunctionMaxParms', function-number );</pre> <i>function-number</i> 在不同的版本之间可能会发生变化。
FunctionMinParms	返回必须由函数指定的最少参数个数。函数通过 <i>function-number</i> 所指定的值进行标识，该值是一个正整数。例如： <pre>SELECT PROPERTY ( 'FunctionMinParms', function-number );</pre> <i>function-number</i> 在不同的版本之间可能会发生变化。
FunctionName	返回通过 <i>function-number</i> 指定的值（该值是一个正整数）所标识的函数的名称： <pre>SELECT PROPERTY ( 'FunctionName', function- number );</pre> <i>function-number</i> 在不同的版本之间可能会发生变化。
HeapsCarver	返回供短期使用（如查询优化）的堆数。

属性名称	说明
HeapsLocked	返回高速缓存中当前锁定的可重定位堆数。
HeapsQuery	返回用于查询处理（散列和排序操作）的堆数。
HeapsRelocatable	返回可重定位堆数。
HttpAddresses	返回服务器为 HTTP 连接正在监听的 TCP/IP 地址的列表（以分号分隔）。例如： (::1):80;127.0.0.1:80
HttpNumActiveReq	返回正在主动处理 HTTP 请求的 HTTP 连接数。不包括已发送其响应的 HTTP 连接。
HttpNumConnections	返回数据库服务器中当前打开的 HTTP 连接数。这些连接可能正在主动处理请求，或者正在长寿命（保持活动状态）连接队列中等待。
HttpNumSessions	返回数据库服务器中活动的和休眠的 HTTP 会话数量。
HttpPorts	返回 Web 服务器的 HTTP 端口号列表（以逗号分隔）。
HttpsAddresses	返回服务器为 HTTPS 连接正在监听的 TCP/IP 地址的列表（以分号分隔）。例如： (::1):443;127.0.0.1:443
HttpsNumActiveReq	返回正在主动处理 HTTPS 请求的安全 HTTPS 连接数量。不包括已发送其响应的 HTTPS 连接。
HttpsNumConnections	返回数据库服务器中当前打开的 HTTPS 连接数。这些连接可能正在主动处理请求，或者正在长寿命（保持活动状态）连接队列中等待。
HttpsPorts	返回 Web 服务器的 HTTPS 端口号列表（以逗号分隔）。
IdleTimeout	返回缺省空闲超时值。
IPAddressMonitor-Period	返回数据库服务器检查新 IP 地址的时间，以秒为单位。
IsAesniAvailable	如果数据库服务器计算机的 CPU 支持 Intel AES-NI 指令集，并且此计算机上正在运行受支持的操作系统，则返回 Yes，否则返回 No。
IsFipsAvailable	如果安装了 FIPS 认证的 DLL，则返回 Yes；否则，返回 No。
IsIQ	此属性保留供系统使用。请不要更改此属性的设置。

属性名称	说明
IsNetworkServer	如果连接到网络数据库服务器，则返回 Yes；如果连接到个人数据库服务器，则返回 No。
IsPortableDevice	如果数据库服务器运行于膝上型计算机、笔记本或其它便携式设备上，则返回 Yes，否则返回 No。在 Windows Mobile 上始终返回 Yes。不考虑 VMWare，如果在 VM 中运行的数据库服务器运行在膝上型计算机上，则返回 No。  在 Windows 上，如果无法确定是否为便携式设备，此属性返回 NULL。  在 Unix 上，此属性始终返回 NULL。
IsRsaAvailable	如果安装了 RSA DLL，则返回 Yes；否则，返回 No。
IsRuntimeServer	为所有版本的数据库服务器返回 No。
IsService	如果数据库服务器是作为服务来运行，则返回 Yes，否则返回 No。
Language	返回服务器的地区语言。
LastConnection-Property	返回表示最后一个连接属性的数字。
LastDatabaseProperty	返回表示最后一个数据库属性的数字。
LastOption	返回一个数字，表示与某个数据库选项相对应的最后一个连接属性。
LastServerProperty	返回表示最后一个服务器属性的数字。
LegalCopyright	返回软件的版权字符串。
LegalTrademarks	返回软件的商标信息。
LicenseCount	返回被授权的存取设备或处理器数量。
LicensedCompany	返回被授权公司的名称。
LicensedUser	返回被授权用户的名称。
LicenseKey	仅供内部使用。
LicenseType	返回许可类型。可以是联网存取设备（每存取设备）或基于 CPU 的类型。
LivenessTimeout	返回客户端活动超时缺省值。

属性名称	说明
LockedCursorPages	返回用于使游标堆在内存中保持固定的页数。
LockedHeapPages	返回高速缓存中锁定的堆页数。
MachineName	返回运行数据库服务器的计算机名称。通常，这是计算机的主机名。
MainHeapBytes	返回用于全局服务器数据结构的字节数。
MainHeapPages	返回用于全局服务器数据结构的页数。
MapPhysicalMemoryEng	返回使用 Address Windowing Extensions 映射到高速缓存中物理内存的数据库页地址空间窗口的数量。
MaxCacheSize	返回所允许的最大高速缓存大小（以千字节为单位）。
MaxConnections	返回服务器允许的最大并发连接数。对于个人服务器，此值在缺省情况下为 10。对于网络服务器，此值在缺省情况下约为 32000。可使用 -gm 服务器选项降低此值。 受计算机资源所限，网络服务器的连接数通常低于缺省值。
MaxEventType	返回最大有效事件类型 ID。
MaxMessage	不建议使用此属性。返回可从数据库服务器消息窗口检索的当前最大行号。它表示数据库服务器消息窗口中显示的最新消息。
MaxMultiProgrammingLevel	返回数据库服务器可同时处理的最大任务数。
MaxRemoteCapability	返回最大有效功能 ID。
Message	不建议使用此属性。返回数据库服务器消息窗口中的某一行，前面带有消息出现的日期和时间。第二个参数指定行号。 由 PROPERTY( "message" ) 返回的值是写入数据库服务器消息窗口的第一行输出。调用 PROPERTY( "message", n ) 会返回服务器输出的第 n 行 (0 为第一行)。缓冲区是有限的，因此随着消息不断被生成，前面的行可能被删除且不再存储在内存中。在这种情况下，返回 NULL。
MessageCategoryLimit	返回可使用 sa_server_messages 系统过程检索的各个严重级和类别的最小消息数。缺省值是 400。

属性名称	说明
MessageText	不建议使用此属性。返回与数据库服务器消息窗口中所指定的行号相关联的文本，前面不带日期和时间。第二个参数指定行号。
MessageTime	不建议使用此属性。返回与数据库服务器消息窗口中所指定的行号相关联的日期和时间。第二个参数指定行号。
MessageWindowSize	不建议使用此属性。返回可从数据库服务器消息窗口检索的最大行数。
MinCacheSize	返回所允许的最小高速缓存大小（以千字节为单位）。
MinMultiProgrammingLevel	返回数据库服务器可同时处理的最小任务数。
MultiPacketsReceived	返回在客户端/服务器通信期间所收到的多包请求数。
MultiPacketsSent	返回在客户端/服务器通信期间所发送的多包请求数。
MultiPageAllocs	返回多页高速缓存分配数。
MultiProgrammingLevel	返回服务器最多可以处理的并发任务数。如果并发任务数大于此值，则会将请求送入队列。可使用 <b>-gn</b> 服务器选项对此进行更改。
Name	如果指定了用于连接到数据库的服务器的替代名称，则返回该名称；否则，返回实际服务器名。  如果客户端连接到复制节点并在连接字符串中指定 <b>NodeType=COPY</b> ，则此属性的值可能与 <b>ServerName</b> 连接参数在客户端连接字符串中指定的数据库服务器名称不同。

属性名称	说明
NativeProcessorArchitecture	<p>返回一个字符串，该字符串标识了正在运行软件的本地处理器类型。对于可仿真处理器的平台（如 x86 或 x64），将返回实际处理器类型（而不是操作系统体系结构类型）。</p> <p>此属性不返回表示操作系统是 32 位还是 64 位的值。</p> <p>值可以包括：</p> <ul style="list-style-type: none"> <li>Windows Mobile - ARM</li> <li>Solaris - SPARC 或 X86_64</li> <li>AIX - PPC</li> <li>HP - IA64</li> <li>Linux - X86 或 X86_64</li> </ul> <p>X86 表示 32 位硬件体系结构。X86_64 表示 64 位硬件体系结构。</p> <p>有关所支持平台的详细信息，请参见 <a href="http://www.sybase.com/detail?id=1002288">http://www.sybase.com/detail?id=1002288</a>。</p>
NumLogicalProcessors	返回服务器计算机上已启用的逻辑处理器数（包括内核和超线程）。
NumLogicalProcessorsUsed	返回数据库服务器将要使用的逻辑处理器数。在 Windows 上，可使用 <code>-gtc</code> 选项更改所使用的逻辑处理器数。
NumPhysicalProcessors	返回服务器计算机上已启用的物理处理器数。此值是用 <code>NumLogicalProcessors</code> 除以每个物理处理器的内核或超线程数得到的。在某些非 Windows 平台上，可能会将内核或超线程计为物理处理器。
NumPhysicalProcessorsUsed	返回数据库服务器将要使用的物理处理器数。在某些平台上，个人服务器的一个处理器上最多只能有四个内核。在 Windows 上，可使用 <code>-gt</code> 选项更改网络数据库服务器所使用的物理处理器数。
ObjectType	返回数据库对象的类型。该值用于 <code>SYSOBJECT</code> 系统视图。
OmniIdentifier	此属性保留供系统使用。请不要更改此属性的设置。
PacketsReceived	返回所收到的客户端/服务器通信包数。此值对于 HTTP 和 HTTPS 连接不会进行更新。
PacketsReceivedUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所收到的包数。（在禁用压缩的情况下，该值与 <code>PacketsReceived</code> 的值相同。）



属性名称	说明
PacketsSent	返回所发送的客户端/服务器通信包数。此值对于 HTTP 和 HTTPS 连接不会进行更新。
PacketsSentUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所发送的包数。（在禁用压缩的情况下，该值与 PacketsSent 的值相同。）
PageSize	返回数据库服务器高速缓存页的大小。可以使用 -gp 选项设置该值，否则为命令行中指定数据库的数据库页面的最大值。
PeakCacheSize	返回高速缓存在当前会话中已达到的最大值（以千字节为单位）。
Platform	返回运行软件的操作系统。例如，如果您正运行在 Windows XP 上，则此属性返回 WindowsXP。
PlatformVer	返回运行软件的操作系统，其中包括内部版本号、服务包等。
PrepStmt	返回数据库服务器当前维护的经过准备的语句数。
ProcessCPU	<p>返回数据库服务器进程的 CPU 使用情况。这些值以秒为单位。在 Windows 和 Unix 上支持此属性。在 Windows Mobile 上不支持此属性。</p> <p>此属性的返回值是自启动数据库服务器以来的累计值。该值与应用程序（如 Windows 任务管理器或 Windows 性能监控器）所返回的瞬时值不匹配。</p>
ProcessCPUSystem	<p>返回数据库服务器进程 CPU 的系统 CPU 使用情况。这是数据库服务器花费在操作系统内核中的 CPU 时间。这些值以秒为单位。在 Windows 和 Unix 上支持此属性。在 Windows Mobile 上不支持此属性。</p> <p>此属性的返回值是自启动数据库服务器以来的累计值。该值与应用程序（如 Windows 任务管理器或性能监控器）所返回的瞬时值不匹配。</p>
ProcessCPUUser	<p>返回数据库服务器进程的用户 CPU 使用情况。这些值以秒为单位。这包括数据库服务器花费在操作系统内核中的 CPU 时间。在 Windows 和 Unix 上支持此属性。在 Windows Mobile 上不支持此属性。</p> <p>此属性的返回值是自启动数据库服务器以来的累计值。该值与应用程序（如 Windows 任务管理器或性能监控器）所返回的瞬时值不匹配。</p>
ProcessID	返回数据库服务器进程的进程 ID。

属性名称	说明
ProcessorAffinity	返回数据库服务器所使用的逻辑处理器，该逻辑处理器由 <code>-gta</code> 选项或者由 <code>sa_server_option</code> 系统过程和 <code>ProcessorAffinity</code> 选项所指定。
ProcessorArchitecture	<p>返回一个字符串，该字符串标识当前软件所对应的处理器类型。值包括：</p> <p>Windows Mobile - ARM Solaris - SPARC 或 X86_64 AIX - PPC HP - IA64 Linux - X86 或 X86_64</p> <p>X86 表示 32 位数据库服务器。X86_64 表示 64 位数据库服务器。</p>
ProductName	返回软件的名称。
ProductVersion	返回正在运行的软件版本。
ProfileFilterConn	如果开启对某特定连接的过程分析，则返回被监控连接的 ID。否则，返回一个空字符串。可使用 <code>sa_server_option</code> 过程来控制用户所进行的过程分析。
ProfileFilterUser	如果开启对某特定用户的过程分析，则返回被监控的用户 ID。否则，返回一个空字符串。可使用 <code>sa_server_option</code> 过程来控制用户所进行的过程分析。
QueryHeapPages	返回用于查询处理（散列和排序操作）的高速缓存页数。
QueryMemActiveCurr	返回主动使用查询内存的请求数。
QueryMemActiveEst	返回在数据库服务器处于稳定状态时对主动使用查询内存的请求数平均值的估计。
QueryMemActiveMax	返回主动允许使用查询内存的最大请求数。
QueryMemExtraAvail	返回可用于在基础内存密集型授权之外进行授权的内存量。
QueryMemGrantBase	返回授权给所有请求的最小内存量。
QueryMemGrantBaseMI	返回授权给内存密集型请求的最小内存量。
QueryMemGrantExtra	返回除 <code>QueryMemGrantBaseMI</code> 以外可分发到活动内存密集型请求的查询内存页数。

属性名称	说明
QueryMemGrantFailed	返回某一请求等待但未能获得查询内存的总次数。
QueryMemGrantGranted	返回当前授予请求的页数。
QueryMemGrantRequested	返回任一请求尝试获取查询内存的总次数。
QueryMemGrantWaited	返回任一请求等待查询内存的总次数。
QueryMemGrantWaiting	返回等待查询内存的当前请求数。
QueryMemPages	返回可用于查询执行算法的内存量，以页数表示。
QueryMemPercentageOfCache	返回可用于查询执行算法的内存量，以最大高速缓存大小的百分比表示。
QuittingTime	返回服务器的关机时间。如果没有指定，则该值为 none。
RememberLastPlan	如果服务器正在记录优化程序返回的最后一个查询优化计划，则返回 Yes。
RememberLastStatement	如果服务器正在记录每个连接所准备的最后一条语句，则返回 Yes；否则，返回 No。
RemoteCapability	返回与给定功能 ID 相关联的远程功能的名称。
RemoteputWait	返回发送通信包时服务器必然发生阻塞的次数。通常情况下，只有当数据库服务器发送数据的速度比客户端或网络接收数据的速度快时才会发生阻塞。并不是指错误情况。
Req	返回为使服务器可以处理新请求或继续处理现有请求而进入该服务器的次数。
ReqCountActive	返回为使服务器可以处理新请求或继续处理现有请求而进入该服务器的次数。
ReqCountBlockContention	返回由于争用内部服务器资源造成的任何连接发生阻塞的次数。
ReqCountBlockIO	返回在等待 IO 请求完成时任何连接发生阻塞的次数。
ReqCountBlockLock	返回在等待由另一个连接持有的行锁定时任何连接发生阻塞的次数。

属性名称	说明
ReqCountUnscheduled	返回在等待服务器线程处理连接时任何连接发生阻塞的次数。
ReqTimeActive	返回服务器花费在直接服务请求上的总时间量。
ReqTimeBlockContention	返回由于争用内部服务器资源造成的任何连接发生阻塞的总时间量。
ReqTimeBlockIO	返回在等待 IO 请求完成时任何连接发生阻塞的总时间量。
ReqTimeBlockLock	返回在等待由另一个连接持有的行锁定时任何连接发生阻塞的总时间量。
ReqTimeUnscheduled	返回在等待服务器线程处理连接时任何连接发生阻塞的总时间量。
RequestFilterConn	返回正在为其过滤记录信息的连接的 ID，否则，返回 -1。
RequestFilterDB	返回正在为其过滤记录信息的数据库的 ID，否则，返回 -1。
RequestLogFile	返回请求记录文件名。如果没有任何请求记录，则会返回一个空字符串。
RequestLogging	返回 SQL、PLAN、HOSTVARS、PROCEDURES、TRIGGERS、OTHER、BLOCKS、REPLACE、ALL 或 NONE 之一，指示请求记录的当前设置。
RequestLogMaxSize	返回请求日志文件的最大大小。
RequestLogNumFiles	返回保留的请求日志文件数。
RequestsReceived	返回客户端/服务器通信请求数或往返次数。它与 <b>PacketsReceived</b> 的不同之处在于：多包请求将计为一个请求，并且不包括活动包。
RequestTiming	如果开启请求计时，则返回 Yes；否则，返回 No。可使用 <b>-zt</b> 数据库服务器选项开启请求计时。
Rlbk	已处理的回退请求数。
SendFail	返回底层通信协议发送数据包失败的次数。

属性名称	说明
ServerEdition	<p>返回描述数据库服务器类型的词汇列表，列表词汇用空格分隔。值包括：</p> <p>Evaluation Developer Web Educational Standard Advanced Workgroup OEM Authenticated</p> <p>如果您对于以下任一功能具有单独的许可，则相应的字符串会被添加到返回的许可字符串：</p> <p>HighAvailability InMemory FIPS</p>
ServerName	返回实际服务器名（永不返回替代服务器名）。可使用此值确定哪个工作服务器当前在担当数据库镜像配置中的主服务器。
StartDBPermission	返回 -gd 服务器选项的设置，该值可以为 DBA、all 或 none 之一。
StartTime	返回服务器启动的日期/时间。
StreamsUsed	返回正在使用的数据库服务器流的数量。
TcpIpAddresses	<p>返回服务器为命令序列和 TDS 连接正在监听的 TCP/IP 地址的列表（以分号分隔）。例如：</p> <p>(::1):2638;127.0.0.1:2638</p>
TempDir	返回服务器存储临时文件的目录。

属性名称	说明
ThreadDeadlocksAvoided	<p>返回检测到线程死锁错误却未向客户端应用程序报告的次数。当数据库服务器启动时，此属性的值为 0。</p> <p>为避免返回线程死锁错误，数据库服务器会动态提高进程并发水平。如果无法提高进程并发水平，将会向客户端应用程序返回线程死锁错误，且 ThreadDeadlocksReported 属性将递增。在个人服务器上，由于进程并发水平的动态调整缺省为关闭，ThreadDeadlocksAvoided 属性恒为 0。</p>
ThreadDeadlocksReported	返回向客户端应用程序报告线程死锁错误的次数。当数据库服务器启动时，此属性的值为 0。
TimeZoneAdjustment	返回为显示服务器本地时间而必须对协调通用时间 (Coordinated Universal Time, 简称 UTC) 增加的分钟数。
TotalBuffers	返回网络缓冲区的总数。
UniqueClientAddresses	返回连接到网络服务器的唯一客户端网络地址数。
UnschReq	返回当前排队等候可用服务器工作线程的请求数。
UserDefinedCounterRate01	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器值随着时间的推移而发生的变化。
UserDefinedCounterRate02	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器值随着时间的推移而发生的变化。
UserDefinedCounterRate03	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器值随着时间的推移而发生的变化。
UserDefinedCounterRate04	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器值随着时间的推移而发生的变化。
UserDefinedCounterRate05	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器值随着时间的推移而发生的变化。

属性名称	说明
UserDefinedCounterRaw01	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器的绝对值。
UserDefinedCounterRaw02	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器的绝对值。
UserDefinedCounterRaw03	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器的绝对值。
UserDefinedCounterRaw04	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器的绝对值。
UserDefinedCounterRaw05	返回用户定义的性能计数器的当前值。此属性的语义由客户端应用程序定义。此计数器也可通过 "性能监控器" 进行访问。"性能监控器" 显示计数器的绝对值。
WebClientLogFile	返回 Web 服务客户端日志文件的名称。
WebClientLogging	返回一个值，指示是否正在将 Web 服务客户端信息记录到文件中。

## SQL 和外部环境用户定义的函数

在 SAP Sybase IQ 中有两种创建用户定义函数的机制。可以使用 SQL 语言编写函数，或者可以使用 ESQL、ODBC、Java、Perl 或 PHP 外部环境。

不要将 SQL UDF 与外部 C 和 C++ UDF 相混淆。外部 UDF 需要特殊的许可证。有关外部 UDF 的信息，请参见用户定义的函数手册。

## SQL 中用户定义的函数

您可以使用 **CREATE FUNCTION** 语句在 SQL 中实现自己的函数。**CREATE FUNCTION** 语句内的 **RETURN** 语句决定函数的数据类型。

创建了 SQL 用户定义的函数后，可以在任何使用相同数据类型的内置函数的位置使用该函数。

**注意：**不要在具有用户定义函数的视图中使用 **CONTAINS** 谓词，因为 **CONTAINS** 条件将被忽略。可用 **LIKE** 谓词代替，或者在视图外发出查询。

## Java 中用户定义的函数

虽然 SQL 函数非常有用，但 Java 类提供了更加强大灵活的方法来实现用户定义的函数，而且还有一个额外的优点：如果您愿意，可以将它们从数据库服务器移动到客户端应用程序。

已安装的 Java 类的任何类方法都可以作为用户定义的函数，在任何使用相同数据类型的内置函数的位置使用。

实例方法与类的特定实例相关，所以其行为与标准用户定义的函数不同。

有关创建 Java 类及类方法的详细信息，请参见《编程》指南中的数据库中的 Java。

## 杂项函数

---

杂项函数对算术、字符串或日期/时间表达式（包括其它函数的返回值）执行操作。

### 兼容性

Adaptive Server 仅支持 COALESCE、ISNULL 和 NULLIF 函数。

### 另请参见

- ARGN 函数 [杂项]（第 134 页）
- COALESCE 函数 [杂项]（第 149 页）
- IFNULL 函数 [杂项]（第 216 页）
- ISNULL 函数 [杂项]（第 220 页）
- ISNUMERIC 函数 [杂项]（第 221 页）
- NEWID 函数 [杂项]（第 247 页）
- NULLIF 函数 [杂项]（第 255 页）
- NUMBER 函数 [杂项]（第 256 页）
- ROWID 函数 [杂项]（第 293 页）

## 按字母顺序排列的函数列表

---

本节单独介绍了各个 SQL 函数。

例如，函数类型（如 Numeric 或 String）在函数名称旁边的中括号中指出。

示例中有些结果已经被舍入或截断。

数据库对象 ID（如表的对象 ID 或列的列 ID）的实际值可能会不同于示例中显示的值。



**ABS 函数 [数值]**

返回数值表达式的绝对值。

语法

```
ABS ( numeric-expression )
```

参数

参数	描述
numeric-expression	要返回其绝对值的数字。

返回

数值表达式的绝对值。

数值表达式数据类型	返回
INT	INT
FLOAT	FLOAT
DOUBLE	DOUBLE
NUMERIC	NUMERIC

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

示例

以下语句返回值 66:

```
SELECT ABS ( -66 ) FROM iq_dummy
```

**ACOS 函数 [数值]**

返回数值表达式的反余弦值（以弧度表示）。

语法

```
ACOS ( numeric-expression )
```

参数

表 8. 参数

参数	描述
numeric-expression	角度的余弦值。

返回

DOUBLE

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

示例

以下语句返回值 1.023945:

```
SELECT ACOS ( 0.52 ) FROM iq_dummy
```

另请参见

- COS 函数 [数值] (第 158 页)
- ATAN2 函数 [数值] (第 137 页)
- ATAN 函数 [数值] (第 137 页)
- ASIN 函数 [数值] (第 136 页)
- COT 函数 [数值] (第 159 页)
- SIN 函数 [数值] (第 299 页)
- TAN 函数 [数值] (第 324 页)

**ARGN 函数 [杂项]**

从参数列表中返回所选参数。

语法

```
ARGN ( integer-expression, expression [ , ... ] )
```

参数

表 9. 参数

参数	描述
integer-expression	表达式列表中的参数位置。

参数	描述
表达式	传递给函数的任意数据类型的表达式。提供的所有表达式必须具有相同的数据类型。

### 返回

使用 *integer-expression* 的值作为 *n*，从其余参数列表中返回第 *n* 个参数（从 1 开始）。

### 注释

使用 *integer-expression* 的值作为 *n*，从其余参数列表中返回第 *n* 个参数（从 1 开始）。虽然表达式可以是任意数据类型，但它们必须属于相同的数据类型。整数表达式必须是从 1 到列表中的表达式数目，否则返回空值。多个表达式之间用逗号分隔。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下语句返回值 6:

```
SELECT ARGN( 6, 1,2,3,4,5,6 ) FROM iq_dummy
```

## ASCII 函数 [字符串]

返回字符串表达式中第一个字节的整数 ASCII 值。

### 语法

```
ASCII ( string-expression )
```

### 参数

参数	描述
string-expression	字符串。

### 返回

SMALLINT

### 注释

如果字符串为空，则 **ASCII** 返回零。文字字符串必须用引号引起来。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

## SQL 函数

- Sybase - 与 Adaptive Server 兼容。

### 示例

当归类序列设置为缺省值 ISO\_BINENG 时，以下语句返回值 90:

```
SELECT ASCII( 'Z' ) FROM iq_dummy
```

## ASIN 函数 [数值]

返回一个数字的反正弦值（以弧度表示）。

### 语法

```
ASIN ( numeric-expression )
```

### 参数

参数	描述
numeric-expression	角度的正弦值。

### 返回

DOUBLE

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下语句返回值 0.546850。

```
SELECT ASIN( 0.52 ) FROM iq_dummy
```

### 另请参见

- COS 函数 [数值]（第 158 页）
- ATAN2 函数 [数值]（第 137 页）
- ATAN 函数 [数值]（第 137 页）
- ACOS 函数 [数值]（第 133 页）
- COT 函数 [数值]（第 159 页）
- SIN 函数 [数值]（第 299 页）
- TAN 函数 [数值]（第 324 页）

## **ATAN 函数 [数值]**

返回一个数字的反正切值（以弧度表示）。

语法

```
ATAN ( numeric-expression )
```

参数

参数	描述
numeric-expression	角度的正切值。

返回

DOUBLE

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

示例

以下语句返回值 0.479519:

```
SELECT ATAN( 0.52 ) FROM iq_dummy
```

另请参见

- COS 函数 [数值]（第 158 页）
- ATAN2 函数 [数值]（第 137 页）
- ASIN 函数 [数值]（第 136 页）
- ACOS 函数 [数值]（第 133 页）
- COT 函数 [数值]（第 159 页）
- SIN 函数 [数值]（第 299 页）
- TAN 函数 [数值]（第 324 页）

## **ATAN2 函数 [数值]**

返回两个数字的比率的反正切值（以弧度表示）。

语法

```
ATAN2 ( numeric-expression1, numeric-expression2 )
```

## 参数

参数	描述
numeric-expression1	要计算其反正切值的比率中的分子。
numeric-expression2	要计算其反正切值的比率中的分母。

## 返回

DOUBLE

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - ATAN2 不受 Adaptive Server 支持。

## 示例

以下语句返回值 0.00866644968879073143:

```
SELECT ATAN2( 0.52, 060 ) FROM iq_dummy
```

## 另请参见

- COS 函数 [数值] (第 158 页)
- ATAN 函数 [数值] (第 137 页)
- ASIN 函数 [数值] (第 136 页)
- ACOS 函数 [数值] (第 133 页)
- COT 函数 [数值] (第 159 页)
- SIN 函数 [数值] (第 299 页)
- TAN 函数 [数值] (第 324 页)

**AVG 函数 [集合]**

为一组行计算数值表达式的平均值，或计算一组唯一值的平均值。

## 语法

```
AVG ( numeric-expression | DISTINCT column-name )
```

## 参数

参数	描述
numeric-expression	要针对一组行计算其平均值的值。
DISTINCT column-name	计算 column-name 中的唯一值的平均值。其用途有限，但出于完整性考虑将其包括在内。

### 返回

对于不包含任何行的组，返回空值。

如果参数为 **DOUBLE**，则返回 **DOUBLE**；否则返回 **NUMERIC**。

### 注释

此平均值不包括 *numeric-expression* 为空值的行。对于不包含任何行的组，返回空值。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下语句返回值 49988.6:

```
SELECT AVG ( salary ) FROM Employees
```

### 另请参见

- **COUNT** 函数 [集合] (第 162 页)
- **SUM** 函数 [集合] (第 322 页)

## **BFILE** 函数 [数据提取]

将各个 **LONG BINARY** 和 **LONG VARCHAR** 单元提取到服务器上的各个操作系统文件中。

### 注释

**IQ** 数据提取工具包含 **BFILE** 函数，可用于将各个 **LONG BINARY** 和 **LONG VARCHAR** 单元提取到服务器上的各个操作系统文件中。**BFILE** 不一定要与数据提取工具一起使用。

如果您有权使用非结构化数据分析功能，则可以将此函数与大对象数据一起使用。

## **BIGINTTOHEX** 函数 [数据类型转换]

返回 **VARCHAR(16)** 中十进制整数的十六进制等效数字。

### 语法

```
BIGINTTOHEX ( integer-expression )
```

## 参数

参数	描述
integer-expression	要转换成十六进制的整数。

## 注释

**BIGINTTOHEX** 接受取值为 **BIGINT** 的整数表达式并返回十六进制等效数字。返回值的左侧会附加零，最多可达 16 位数字。所有无标度的整数数据类型都将用作整数表达式。

如果需要，将会自动执行转换。仅当小数值为零时，才会截断常量。如果列中包含正小数位数值，则无法截断该列。如果转换失败，**SAP Sybase IQ** 会返回错误，但如果 **CONVERSION\_ERROR** 选项设置为 **OFF** 则例外。在这种情况下，结果为 **NULL**。

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 与 Adaptive Server 兼容。

## 示例

返回值 00000000000000009:

```
SELECT BIGINTTOHEX(9) FROM iq_dummy
```

返回值 FFFFFFFFFFFFFFFF7:

```
SELECT BIGINTTOHEX (-9) FROM iq_dummy
```

## 另请参见

- **HEXTOBIGINT** 函数 [数据类型转换] (第 203 页)
- **HEXTOINT** 函数 [数据类型转换] (第 204 页)
- **INTTOHEX** 函数 [数据类型转换] (第 218 页)

**BIT\_LENGTH 函数 [字符串]**

返回不带符号的 64 位值，该值包含列参数的位长度。

## 语法

```
BIT_LENGTH( column-name )
```

## 参数

参数	描述
column-name	列的名称



返回

INT

注释

空值参数的返回值为空。

**BIT\_LENGTH** 函数支持所有 SAP Sybase IQ 数据类型。

如果您有权使用非结构化数据分析功能，则可以将此函数与大对象数据一起使用。

请参见《非结构化数据分析》中的函数支持。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - SQL Anywhere 或 Adaptive Server 不支持。

另请参见

- **BYTE\_LENGTH** 函数 [字符串] (第 141 页)
- **CHAR\_LENGTH** 函数 [字符串] (第 147 页)
- **COL\_LENGTH** 函数 [系统] (第 150 页)
- **DATALENGTH** 函数 [系统] (第 164 页)
- **LEN** 函数 [字符串] (第 229 页)
- **LENGTH** 函数 [字符串] (第 230 页)
- **OBJECT\_NAME** 函数 [系统] (第 257 页)
- **OCTET\_LENGTH** 函数 [字符串] (第 258 页)
- **STR\_REPLACE** 函数 [字符串] (第 315 页)

## **BYTE\_LENGTH** 函数 [字符串]

返回字符串中的字节数。

语法

```
BYTE_LENGTH ( string-expression )
```

参数

参数	描述
string-expression	要计算其长度的字符串。

返回

INT

### 注释

返回的长度中包括尾随空格字符。

空字符串的返回值为空值。

如果字符串使用的是多字节字符集，则 **BYTE\_LENGTH** 值不同于 **CHAR\_LENGTH** 返回的字符数。

如果您有权使用非结构化数据分析功能，则可以将此函数用于大对象数据：

- 只有在查询返回小于 2GB 时，**BYTE\_LENGTH** 函数才支持 LONG BINARY 列和变量以及 LONG VARCHAR 列和变量。如果返回的 LONG BINARY 或 LONG VARCHAR 数据的字节长度大于或等于 2GB，**BYTE\_LENGTH** 就会返回错误，提示您必须使用 **BYTE\_LENGTH64** 函数。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的服务商扩充。
- SAP Sybase - 不受 Adaptive Server 支持。

### 示例

返回值 12:

```
SELECT BYTE_LENGTH( 'Test Message' ) FROM iq_dummy
```

### 另请参见

- BIT\_LENGTH 函数 [字符串] (第 140 页)
- CHAR\_LENGTH 函数 [字符串] (第 147 页)
- COL\_LENGTH 函数 [系统] (第 150 页)
- DATALENGTH 函数 [系统] (第 164 页)
- LEN 函数 [字符串] (第 229 页)
- LENGTH 函数 [字符串] (第 230 页)
- OBJECT\_NAME 函数 [系统] (第 257 页)
- OCTET\_LENGTH 函数 [字符串] (第 258 页)
- STR\_REPLACE 函数 [字符串] (第 315 页)

## **BYTE\_LENGTH64** 函数

**BYTE\_LENGTH64** 返回不带符号的 64 位值，该值包含 LONG BINARY 列参数的字节长度。

### 注释

**BYTE\_LENGTH64** 也支持 LONG VARCHAR 数据类型以及任意数据大小的 LONG BINARY 和 LONG VARCHAR 变量。

如果您有权使用非结构化数据分析功能，则可以将此函数与大对象数据一起使用。

请参见《非结构化数据分析》中的函数支持。

## **BYTE\_SUBSTR64 和 BYTE\_SUBSTR 函数**

**BYTE\_SUBSTR64** 和 **BYTE\_SUBSTR** 返回 LONG BINARY 列参数的长二进制字节子字符串。

### 注释

**BYTE\_SUBSTR64** 和 **BYTE\_SUBSTR** 函数也支持 LONG VARCHAR 数据类型以及任意数据大小的 LONG BINARY 和 LONG VARCHAR 变量。

**CHAR\_LENGTH64** 也支持任意数据大小的 LONG VARCHAR 变量。

如果您有权使用非结构化数据分析功能，则可以将此函数与大对象数据一起使用。

请参见《非结构化数据分析》中的 **BYTE\_SUBSTR64** 和 **BYTE\_SUBSTR** 函数。

## **CAST 函数 [数据类型转换]**

返回转换为提供的数据类型的表达式的值。

### 语法

```
CAST ( expression
      AS data type )
```

### 参数

参数	描述
表达式	要转换的表达式。
数据类型	目标数据类型。

### 返回

指定的数据类型。

### 注释

如果不指示字符串类型的长度，SAP Sybase IQ 将选择适当的长度。如果没有为 **DECIMAL** 转换指定精度和小数位数，数据库服务器会选择适当的值。

如果针对 **NULL** 到 **NUMERIC** 的显式转换既未指定精度也未指定小数位数，则缺省值为 **NUMERIC(1,0)**。例如，

```
SELECT CAST( NULL AS NUMERIC ) A,
       CAST( NULL AS NUMERIC(15,2) ) B
```

被描述为：

```
A NUMERIC(1,0)
B NUMERIC(15,2)
```

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下函数确保字符串被用作日期:

```
CAST( '2000-10-31' AS DATE )
```

计算表达式 **1+2** 的值, 并将结果转换为单字符字符串 (即数据服务器分配的长度):

```
CAST( 1 + 2 AS CHAR )
```

可以使用 **CAST** 函数缩短字符串长度:

```
SELECT CAST( lname AS CHAR(5) ) FROM Customers
```

### 另请参见

- CONVERT 函数 [数据类型转换] (第 153 页)
- HOURS 函数 [日期和时间] (第 206 页)
- MINUTES 函数 [日期和时间] (第 242 页)
- MONTHS 函数 [日期和时间] (第 246 页)
- REPLACE 函数 [字符串] (第 286 页)
- SECOND 函数 [日期和时间] (第 296 页)
- WEEKS 函数 [日期和时间] (第 336 页)
- YEAR 函数 [日期和时间] (第 341 页)
- YEARS 函数 [日期和时间] (第 342 页)
- DAYS 函数 [日期和时间] (第 182 页)

## CEIL 函数 [数值]

返回大于或等于指定表达式的最小整数。

**CEIL** 是 **CEILING** 的同义词。

### 语法

```
CEIL ( numeric-expression )
```

### 参数

参数	描述
表达式	数据类型为精确数值、近似数值、货币或任何可隐式转换为这些类型之一的类型的列、变量或表达式。对于其它数据类型, <b>CEIL</b> 会产生错误。返回值与所提供的值属于相同的数据类型。

### 注释

对于给定表达式，**CEIL** 函数将采用一个参数。例如，**CEIL (-123.45)** 返回 -123。**CEIL (123.45)** 返回 124。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 另请参见

- FLOOR 函数 [数值] (第 198 页)
- CEILING 函数 [数值] (第 145 页)

## **CEILING 函数 [数值]**

返回一个数字的上限（不小于的最小整数）。

**CEIL** 是 **CEILING** 的同义词。

### 语法

```
CEILING ( numeric-expression )
```

### 参数

参数	描述
numeric-expression	要计算其上限的数字。

### 返回

DOUBLE

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下语句返回值 60.00000:

```
SELECT CEILING( 59.84567 ) FROM iq_dummy
```

以下语句返回值 123:

```
SELECT CEILING( 123 ) FROM iq_dummy
```

以下语句返回值 124.00:

```
SELECT CEILING( 123.45 ) FROM iq_dummy
```

## SQL 函数

以下语句返回值 -123.00:

```
SELECT CEILING( -123.45 ) FROM iq_dummy
```

### 另请参见

- FLOOR 函数 [数值] (第 198 页)
- CEIL 函数 [数值] (第 144 页)

## CHAR 函数 [字符串]

返回含数字所表示的 ASCII 值的字符。

### 语法

```
CHAR ( integer-expression )
```

### 参数

参数	描述
integer-expression	要转换成 ASCII 字符的数字。该数字必须在 0 到 255 (包括 0 和 255) 之间。

### 返回

VARCHAR

### 注释

将会返回与所提供的数值表达式模 256 相对应的当前数据库字符集中的字符。

对于值大于 255 或小于 0 的整数表达式, **CHAR** 返回空值。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下语句返回值 "Y":

```
SELECT CHAR( 89 ) FROM iq_dummy
```

以下语句返回值 "S":

```
SELECT CHAR( 83 ) FROM iq_dummy
```

## **CHAR\_LENGTH** 函数 [字符串]

返回字符串中的字符数。

### 语法

```
CHAR_LENGTH ( string-expression )
```

### 参数

参数	描述
<i>string-expression</i>	要计算其长度的字符串。

### 返回

INT

### 注释

返回的长度中包括尾随空格字符。

NULL 字符串的返回值为 NULL。

如果字符串使用的是多字节字符集，则 **CHAR\_LENGTH** 值可能会小于 **BYTE\_LENGTH** 值。

**CHAR\_LENGTH64** 也支持任意数据大小的 LONG VARCHAR 变量。

如果您有权使用非结构化数据分析功能，则可以将此函数用于大对象数据。请参见非结构化数据分析中的 **CHAR\_LENGTH** 函数。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下语句返回值 8:

```
SELECT CHAR_LENGTH( 'Chemical' ) FROM iq_dummy
```

### 另请参见

- **BIT\_LENGTH** 函数 [字符串] (第 140 页)
- **BYTE\_LENGTH** 函数 [字符串] (第 141 页)
- **COL\_LENGTH** 函数 [系统] (第 150 页)
- **DATALENGTH** 函数 [系统] (第 164 页)
- **LEN** 函数 [字符串] (第 229 页)
- **LENGTH** 函数 [字符串] (第 230 页)

- OBJECT\_NAME 函数 [系统] (第 257 页)
- OCTET\_LENGTH 函数 [字符串] (第 258 页)
- STR\_REPLACE 函数 [字符串] (第 315 页)

### **CHAR\_LENGTH64 函数**

**CHAR\_LENGTH64** 函数返回不带符号的 64 位值，该值包含 LONG VARCHAR 列参数的字符长度（包括尾随空白）。

*注释*

**CHAR\_LENGTH64** 也支持任意数据大小的 LONG VARCHAR 变量。

如果您有权使用非结构化数据分析功能，则可以将此函数用于大对象数据。请参见非结构化数据分析中的 **CHAR\_LENGTH64** 函数。

### **CHARINDEX 函数 [字符串]**

返回一个指定字符串在另一个字符串中第一次出现的位置。

*语法*

```
CHARINDEX ( string-expression1, string-expression2 )
```

*参数*

参数	描述
<i>string-expression1</i>	要搜索的字符串。该字符串限制为 255 个字节。
<i>string-expression2</i>	要搜索的字符串。所搜索的字符串中第一个字符的位置是 1。

*返回*

INT

*注释*

在 **CHARINDEX** 函数中返回或指定的所有位置或偏移量始终为字符偏移量，并且可能不同于多字节数据的字节偏移量。

如果所搜索的字符串包含指定字符串的多个实例，则 **CHARINDEX** 将返回第一个实例的位置。

如果所搜索的字符串不包含指定的字符串，则 **CHARINDEX** 将返回零 (0)。

搜索零长度的字符串将返回 1。

如果任一参数为 NULL，结果则为 NULL。

**CHARINDEX** 为 CHAR 和 VARCHAR 列返回 32 位带符号的整数位置。



如果您有权使用非结构化数据分析功能，则可以将此函数用于大对象数据。请参见非结构化数据分析中的 **CHARINDEX** 函数。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

#### 示例

语句:

```
SELECT Surname, GivenName
FROM Employees
WHERE CHARINDEX('K', Surname) = 1
```

返回以下值:

Surname	GivenName
Klobucher	James
Kuo	Felicia
Kelly	Moira

#### 另请参见

- SUBSTRING 函数 [字符串] (第 319 页)

## **COALESCE** 函数 [杂项]

返回列表中的第一个非空表达式。

#### 语法

```
COALESCE ( expression, expression [ , ... ] )
```

#### 参数

参数	描述
表达式	任意表达式。

#### 返回

ANY

#### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下语句返回值 34:

```
SELECT COALESCE( NULL, 34, 13, 0 ) FROM iq_dummy
```

### 另请参见

- ISNULL 函数 [杂项] (第 220 页)

## **COL\_LENGTH 函数 [系统]**

返回定义的列长度。

### 语法

```
COL_LENGTH ( table-name, column-name )
```

参数	描述
table-name	表名。
column-name	列名称。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。

### 示例

返回列长度 35:

```
SELECT COL_LENGTH ( 'CUSTOMERS', 'ADDRESS' ) FROM iq_dummy
```

### 另请参见

- BIT\_LENGTH 函数 [字符串] (第 140 页)
- BYTE\_LENGTH 函数 [字符串] (第 141 页)
- CHAR\_LENGTH 函数 [字符串] (第 147 页)
- DATALENGTH 函数 [系统] (第 164 页)
- LEN 函数 [字符串] (第 229 页)
- LENGTH 函数 [字符串] (第 230 页)
- OBJECT\_NAME 函数 [系统] (第 257 页)
- OCTET\_LENGTH 函数 [字符串] (第 258 页)
- STR\_REPLACE 函数 [字符串] (第 315 页)

## COL\_NAME 函数 [系统]

返回列名。

语法

```
COL_NAME ( table-id, column-id [ , database-id ] )
```

参数

表 10. 参数

参数	描述
table-id	表的对象 ID。
column-id	列的列 ID。
database-id	数据库 ID。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。

示例

以下语句返回列名 lname。Customers 表的对象 ID 为 100209，即 **OBJECT\_ID** 函数返回值。列 ID 存储在 syscolumn 系统表的 column\_id 列中。iqdemo 数据库的数据库 ID 为 0，即 **DB\_ID** 函数返回值。

```
SELECT COL_NAME( 100209, 3, 0 ) FROM iq_dummy
```

以下语句返回列名 city。

```
SELECT COL_NAME ( 100209, 5 ) FROM iq_dummy
```

另请参见

- **DB\_ID** 函数 [系统] (第 183 页)
- **DB\_NAME** 函数 [系统] (第 184 页)
- **DB\_PROPERTY** 函数 [系统] (第 185 页)
- **NEXT\_DATABASE** 函数 [系统] (第 250 页)
- **OBJECT\_ID** 函数 [系统] (第 257 页)
- **OBJECT\_NAME** 函数 [系统] (第 257 页)

**CONNECTION\_PROPERTY 函数 [系统]**

以字符串形式返回给定连接属性的值。

*语法*

```
CONNECTION_PROPERTY ( { integer-expression1 | string-expression }
                    ... [ , integer-expression2 ] )
```

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

*参数*

参数	描述
integer-expression1	大多数情况下，提供字符串表达式作为第一个参数更方便。如果您确实提供了 integer-expression1，则它将是连接属性 ID。可以使用 PROPERTY_NUMBER 函数来确定这一点。
string-expression	连接属性名称。您必须指定属性 ID 或属性名称。
integer-expression2	当前数据库连接的连接 ID。如果省略此参数，则使用当前连接。

*返回*

VARCHAR

*注释*

如果省略第二个参数，则使用当前连接。

*标准和兼容性*

- ISO/ANSI SQL 语法的供应商扩展。
- 与 Adaptive Server 兼容。

*示例*

以下语句返回所维护的预准备语句的数目，例如 4：

```
SELECT connection_property( 'PrepStmnt' ) FROM iq_dummy
```

**另请参见**

- PROPERTY 函数 [系统] (第 267 页)
- PROPERTY\_NAME 函数 [系统] (第 269 页)
- PROPERTY\_NUMBER 函数 [系统] (第 270 页)
- 可用于服务器的属性 (第 92 页)

- 各数据库可用的属性 (第 115 页)
- 连接属性 (第 92 页)
- sp\_iqshowpsexec 过程 (第 507 页)
- sp\_iqcontext 过程 (第 410 页)

## **CONVERT 函数 [数据类型转换]**

返回转换成提供的数据类型的表达式。

语法

```
CONVERT ( data-type, expression [ , format-style ] )
```

参数

参数	描述
<i>data-type</i>	表达式将要转换成的数据类型。
表达式	要转换的表达式。
<i>format-style</i>	对于将字符串转换为日期或时间数据类型以及相反的转换过程, <i>format-style</i> 是描述要使用的日期格式字符串的样式代码编号。

如果未提供任何 *format-style* 参数, 系统将使用数据库选项设置。

**表 11. CONVERT 格式样式代码输出**

不含世纪 (yy)	含世纪 (yyyy)	输出
-	0 或 100	mmm dd yyyy hh:nnAM (或 PM)
1	101	mm/dd/yy[yy]
2	102	[yy]yy.mm.dd
3	103	dd/mm/yy[yy]
4	104	dd.mm.yy[yy]
5	105	dd-mm-yy[yy]
6	106	dd mmm yy[yy]
7	107	mmm dd, yy[yy]
8	108	hh:nn:ss
-	9 或 109	mmm dd yyyy hh:nn:ss:sssAM (或 PM)

不含世纪 (yy)	含世纪 (yyyy)	输出
10	110	mm-dd-yy[yy]
11	111	[yy]yy/mm/dd
12	112	[yy]yymmdd
-	13 或 113	dd mmm yyyy hh:nn:ss:sss (24 小时制, 欧洲缺省时间 + 毫秒, 4 位数年份)
14	114	hh:nn:ss (24 小时制)
-	20 或 120	yyyy-mm-dd hh:nn:ss (24 小时制, ODBC 规范, 4 位数年份)
-	21 或 121	yyyy-mm-dd hh:nn:ss:sss (24 小时制, ODBC 规范加毫秒, 4 位数年份)
36	136	hh:nn:ss.sssssAM (或 PM)
37	137	hh:nn:ss.sssss
38	138	mmm dd yy[yy] hh:nn:ss.sssssAM (或 PM)
39	139	mmm dd yy[yy] hh:nn:ss.sssss
40	140	[yy]yy-mm-dd hh:nn:ss.sssss
-	365	yyyyjjj (采用字符串或整数形式, 其中 jjj 是一年中的儒略天数, 从 1 到 366)

**CONVERT** 格式样式表中日期分量的缩写和值:

缩写	日期分量	值
hh	hour	0 - 23
nn	minute	0 - 59
ss	second	0 - 59
sss	millisecond	0 - 999
sssss	microsecond	0 - 999999
mmm	month	一月到十二月
dd	day	1 - 31
yyyy	year	0001 - 9999

缩写	日期分量	值
mm	month	1 - 12

*返回*

指定的数据类型。

*注释*

**CONVERT** 函数的结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **CONVERT**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **CONVERT** 设置为正确的数据类型和大小。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 和 SQL Anywhere 兼容，但格式样式 365 除外，该样式是仅限于 SAP Sybase IQ 的扩展。

*示例*

以下语句举例说明了格式样式的用法：

```
SELECT CONVERT( CHAR( 20 ), order_date, 104 )
FROM sales_order
```

order_date
16.03.1993
20.03.1993
23.03.1993
25.03.1993
...

```
SELECT CONVERT( CHAR( 20 ), order_date, 7 )
FROM sales_order
```

order_date
mar 16, 93
mar 20, 93
mar 23, 93
mar 25, 93

<b>order_date</b>
-------------------

...
-----

```
SELECT order_datetime, CONVERT (CHAR(30), order_datetime, 40)
order_datetime40,
CONVERT (CHAR(30), order_datetime, 140) order_datetime140
FROM sales_order;
```

order_datetime	order_datetime40	order_datetime140
03/05/2009 01:03:05.123456	09-03-05 01:03:05.123456	2009-03-05 01:03:05.123456
03/05/2009 13:05:07.654321	09-03-05 13:05:07.654321	2009-03-05 13:05:07.654321

```
SELECT CONVERT (CHAR(50), DATETIME ('2009-11-03
11:10:42.033189'), 136) FROM iq_dummy 返回 11:10:42.033189AM
```

```
SELECT CONVERT (CHAR(50), NOW(), 137) FROM iq_dummy 返回
14:54:48.794122
```

以下语句举例说明了格式样式 365 的用法，通过该样式可将 DATE 和 DATETIME 类型的数据与字符串或整数类型的数据相互进行转换：

```
CREATE TABLE tab
(date_col DATE, int_col INT, char7_col CHAR(7));
INSERT INTO tab (date_col, int_col, char7_col)
VALUES ( 'Dec 17, 2004' , 2004352, '2004352' );
```

```
SELECT CONVERT (VARCHAR(8), tab.date_col, 365) FROM tab; 返回
'2004352'
```

```
SELECT CONVERT (INT, tab.date_col, 365) from tab; 返回 2004352
```

```
SELECT CONVERT (DATE, tab.int_col, 365) FROM TAB; 返回 2004-12-17
```

```
SELECT CONVERT (DATE, tab.char7_col, 365) FROM tab; 返回
2004-12-17
```

以下语句举例说明了到整数的转换，并返回值 5。

```
SELECT CONVERT ( integer, 5.2 ) FROM iq_dummy
```

### 另请参见

- CAST 函数 [数据类型转换] (第 143 页)
- HOURS 函数 [日期和时间] (第 206 页)
- MINUTES 函数 [日期和时间] (第 242 页)
- MONTHS 函数 [日期和时间] (第 246 页)
- REPLACE 函数 [字符串] (第 286 页)
- SECOND 函数 [日期和时间] (第 296 页)



- WEEKS 函数 [日期和时间] (第 336 页)
- YEAR 函数 [日期和时间] (第 341 页)
- YEARS 函数 [日期和时间] (第 342 页)

## **CORR 函数 [集合]**

返回一组数字对的相关系数。

### 语法 1

**CORR** (*dependent-expression*, *independent-expression*)

### 语法 2

**CORR** (*dependent-expression*, *independent-expression*)

**OVER** (*window-spec*)

### 参数

表 12. 参数

参数	描述
<i>dependent-expression</i>	受 <i>independent-expression</i> 影响的变量。
<i>independent-expression</i>	影响结果的变量。

### 返回

DOUBLE

### 注释

**CORR** 函数将其参数转换为 **DOUBLE**，执行双精度浮点计算，并返回一个 **DOUBLE** 作为结果。如果将 **CORR** 应用于空集，它将返回空值。

*dependent-expression* 和 *independent-expression* 都是数值。在消除 *dependent-expression* 或 *independent-expression* 为空值的值对之后，系统会将此函数应用于 (*dependent-expression*, *independent-expression*) 的集合。将进行以下计算：

$$\text{COVAR\_POP}(y, x) / (\text{STDDEV\_POP}(x) * \text{STDDEV\_POP}(y))$$

其中 *x* 表示 *dependent-expression*，*y* 表示 *independent-expression*。

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 **ROLLUP** 和 **CUBE**。

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。除核心 SQL 功能以外的 SQL 基础功能。
- Sybase - 与 SQL Anywhere 兼容。

### 示例

以下示例执行相关操作，用来查找收入水平是否与年龄相关联。此函数返回值 0.440227:

```
SELECT CORR( Salary, ( YEAR( NOW( ) ) - YEAR( BirthDate ) ) ) FROM Employees;
```

### 另请参见

- 窗口化集合函数的用法 (第 78 页)

## **COS 函数 [数值]**

返回一个数字的余弦值（以弧度表示）。

### 语法

```
COS ( numeric-expression )
```

### 参数

表 13. 参数

参数	描述
numeric-expression	角度（以弧度为单位）。

### 返回

此函数将其参数转换为 DOUBLE，以双精度浮点执行计算，然后返回 DOUBLE 值作为结果。如果参数为 NULL，则结果为 NULL。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 示例

以下语句返回值 0.86781:

```
SELECT COS( 0.52 ) FROM iq_dummy
```

### 另请参见

- ATAN2 函数 [数值] (第 137 页)

- ATAN 函数 [数值] (第 137 页)
- ASIN 函数 [数值] (第 136 页)
- ACOS 函数 [数值] (第 133 页)
- COT 函数 [数值] (第 159 页)
- SIN 函数 [数值] (第 299 页)
- TAN 函数 [数值] (第 324 页)

## **COT 函数 [数值]**

返回一个数字的余切值 (以弧度表示)。

*语法*

```
COT ( numeric-expression )
```

*参数*

**表 14. 参数**

参数	描述
numeric-expression	角度 (以弧度为单位)。

*返回*

此函数将其参数转换为 DOUBLE，以双精度浮点执行计算，然后返回 DOUBLE 值作为结果。如果参数为 NULL，则结果为 NULL。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

以下语句返回值 1.74653:

```
SELECT COT( 0.52 ) FROM iq_dummy
```

**另请参见**

- COS 函数 [数值] (第 158 页)
- ATAN2 函数 [数值] (第 137 页)
- ATAN 函数 [数值] (第 137 页)
- ASIN 函数 [数值] (第 136 页)
- ACOS 函数 [数值] (第 133 页)
- SIN 函数 [数值] (第 299 页)
- TAN 函数 [数值] (第 324 页)

## COVAR\_POP 函数 [集合]

返回一组数字对的总体协方差。

### 语法 1

```
COVAR_POP (dependent-expression, independent-expression)
```

### 语法 2

```
COVAR_POP (dependent-expression, independent-expression)
```

```
OVER (window-spec)
```

### 参数

参数	描述
<i>dependent-expression</i>	受独立变量影响的变量。
<i>independent-expression</i>	影响结果的变量。

### 注释

此函数将其参数转换为 **DOUBLE**，执行双精度浮点计算，并返回一个 **DOUBLE** 作为结果。如果将 **COVAR\_POP** 应用于空集，它将返回空值。

*dependent-expression* 和 *independent-expression* 都是数值。在消除 *dependent-expression* 或 *independent-expression* 为空值的值对之后，系统会将此函数应用于 (*dependent-expression*, *independent-expression*) 的集合。将进行以下计算：

$$(\text{SUM}(x*y) - \text{SUM}(x) * \text{SUM}(y) / n) / n$$

其中 *x* 表示 *dependent-expression*，*y* 表示 *independent-expression*。

---

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 **ROLLUP** 和 **CUBE**，也不支持 **DISTINCT**。

---

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。除核心 SQL 功能以外的 SQL 基础功能。
- Sybase - 与 SQL Anywhere 兼容。

### 示例

以下示例衡量雇员的年龄与薪水之间的相关程度。此函数返回值 73785.840059：

```
SELECT COVAR_POP( Salary, ( YEAR( NOW( ) ) - YEAR( BirthDate ) ) )
FROM Employees;
```

## COVAR\_SAMP 函数 [集合]

返回一组数字对的样本协方差。

### 语法 1

```
COVAR_SAMP (dependent-expression, independent-expression)
```

### 语法 2

```
COVAR_SAMP (dependent-expression, independent-expression)
```

```
OVER (window-spec)
```

### 参数

参数	描述
dependent-expression	受独立变量影响的变量。
independent-expression	影响结果的变量。

### 注释

此函数将其参数转换为 **DOUBLE**，执行双精度浮点计算，并返回一个 **DOUBLE** 作为结果。如果将 **COVAR\_SAMP** 应用于空集，它将返回空值。

dependent-expression 和 independent-expression 都是数值。在消除 dependent-expression 或 independent-expression 为空值的值对之后，系统会将此函数应用于 (dependent-expression, independent-expression) 的集合。

```
(SUM(x*y) - SUM(x) * SUM(y) / n) / (n-1)
```

其中 x 表示 dependent-expression，y 表示 independent-expression。

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 **ROLLUP** 和 **CUBE**，也不支持 **DISTINCT**。

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。除核心 SQL 功能以外的 SQL 基础功能。
- Sybase - 与 SQL Anywhere 兼容。

### 示例

以下示例衡量雇员的年龄与薪水之间的相关程度。此函数返回值 74782.946005：

```
SELECT COVAR_SAMP( Salary, ( 2008 - YEAR( BirthDate ) ) ) FROM Employees;
```

**COUNT 函数 [集合]**

根据指定的参数计算组中的行数。

*语法*

```
COUNT ( * | expression | DISTINCT column-name )
```

*参数*

参数	描述
*	返回每组中的行数。
表达式	返回每一组中 <i>expression</i> 不为空值的行数。
<b>DISTINCT</b> <i>column-name</i>	返回 <i>column-name</i> 中的不同值的数量。计数中不包括值为空值的行。

**注意：** 当显示查询结果时，\* 不会显示在列标题中，而是显示为：

```
Count ()
```

*返回*

INT

*标准和兼容性*

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

返回每个唯一的城市和含有该城市值的行数：

```
SELECT city , Count(*)
FROM Employees
GROUP BY city
```

**另请参见**

- **AVG 函数 [集合]** (第 138 页)
- **SUM 函数 [集合]** (第 322 页)
- **窗口化集合函数的用法** (第 78 页)

## CUME\_DIST 函数 [分析]

**CUME\_DIST** 函数是一个 **rank** 分析函数，用于计算一个值在一组行中的相对位置。它会返回一个介于 0 和 1 之间的小数值。

### 语法

```
CUME_DIST () OVER (window-spec)
```

### 返回

介于 0 和 1 之间的 **DOUBLE** 值。

### 注释

SAP Sybase IQ 计算大小为 **N** 的 **S** 集合中 **x** 值的累计分布，方法是使用：**CUME\_DIST(x)** = 按照指定的顺序 **S** 中 **x**（包括 **x**）之前的值个数 / **N**

当前，不得在 **CUME\_DIST** 函数中使用组合排序键。您可以对任何其它 **rank** 函数使用排序键组合。

您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。*window-spec* 必须包含 **ORDER BY** 子句，并且不能包含 **ROWS** 或 **RANGE** 子句。

---

**注意：** **DISTINCT** 不受支持。

---

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

### 示例

以下示例返回一个包含居住在 California 的职员的薪水累计分布的结果集：

```
SELECT DepartmentID, Surname, Salary, CUME_DIST() OVER (PARTITION BY
DepartmentID ORDER BY Salary DESC) "Rank" FROM Employees WHERE State IN
('CA');
```

返回的结果集为：

表 15. CUME\_DIST 结果集

DepartmentID	Surname	Salary	Rank
200	Savarino	72,300.000	0.333333
200	Clark	45,000.000	0.666667
200	Overbey	39,300.000	1.000000

**DATALENGTH 函数 [系统]**

返回表达式的长度（以字节为单位）。

*语法*

```
DATALENGTH ( expression )
```

*参数*

参数	描述
表达式	表达式通常是列名。如果表达式是字符串常量，则它必须用引号引起来。

*返回*

UNSIGNED INT

*注释*

**表 16. DATALENGTH 返回值**

数据类型	DATALENGTH
SMALLINT	2
INTEGER	4
DOUBLE	8
CHAR	数据的长度
BINARY	数据的长度

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。

*示例*

返回值 35，即 `company_name` 列中最长的字符串：

```
SELECT MAX( DATALENGTH( company_name ) )
FROM Customers
```

*另请参见*

- BIT\_LENGTH 函数 [字符串]（第 140 页）
- BYTE\_LENGTH 函数 [字符串]（第 141 页）
- CHAR\_LENGTH 函数 [字符串]（第 147 页）
- COL\_LENGTH 函数 [系统]（第 150 页）



- LEN 函数 [字符串] (第 229 页)
- LENGTH 函数 [字符串] (第 230 页)
- OBJECT\_NAME 函数 [系统] (第 257 页)
- OCTET\_LENGTH 函数 [字符串] (第 258 页)
- STR\_REPLACE 函数 [字符串] (第 315 页)

## DATE 函数 [日期和时间]

将表达式转换为日期，并删除任何小时、分钟或秒。

语法

```
DATE ( expression )
```

参数

表 17. 参数

参数	描述
表达式	要转换成日期格式的值。表达式通常是字符串。

返回

DATE

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

示例

以下语句返回值 1988-11-26 作为日期。

```
SELECT DATE ( '1988-11-26 21:20:53' ) FROM iq_dummy
```

## DATEADD 函数 [日期和时间]

返回通过将指定数量的指定日期分量添加到日期中而生成的日期。

语法

```
DATEADD ( date-part, numeric-expression, date-expression )
```

参数

参数	描述
日期分量	要添加到日期中的日期分量。

参数	描述
numeric-expression	要添加到日期中的日期部分的数量。numeric-expression 可以是任何数值类型；值截断为整数。numeric-expression 中的最大微秒为 2147483647，即 35:47.483647（35 分 47 秒 483647 微秒）。
date-expression	要修改的日期。

*返回*

TIMESTAMP

*注释*

DATEADD 是与 Transact-SQL 兼容的数据操作函数。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

以下语句返回值 1995-11-02 00:00:00.000:

```
SELECT DATEADD( MONTH, 102, '1987/05/02' ) FROM iq_dummy
```

以下语句返回值 2009-11-10 14:57:52.722016:

```
SELECT DATEADD(MICROSECOND, 15, '2009-11-10  
14:57:52.722001') FROM iq_dummy
```

以下语句返回值 1985-05-02 00:00:00.123456:

```
SELECT DATEADD(MICROSECOND, 123456, '1985/05/02')  
FROM iq_dummy
```

以下语句返回值 1985-05-01 23:59:59.876544:

```
SELECT DATEADD(MICROSECOND, -123456, '1985/05/02')  
FROM iq_dummy
```

以下语句返回值 2009-11-03 11:10:42.033192:

```
SELECT DATEADD(MCS, 2, '2009-11-03 11:10:42.033190')  
FROM iq_dummy
```

**另请参见**

- DATECEILING 函数 [日期和时间] (第 167 页)
- DATEDIFF 函数 [日期和时间] (第 169 页)
- DATEFLOOR 函数 [日期和时间] (第 171 页)
- DATEPART 函数 [日期和时间] (第 176 页)
- DATENAME 函数 [日期和时间] (第 175 页)
- DATEROUND 函数 [日期和时间] (第 177 页)

- 日期分量 (第 85 页)

## **DATECEILING** 函数 [日期和时间]

通过将所提供的值最大增加至与指定粒度最接近的较大值，计算新日期、时间或 datetime 值。

### 语法

```
DATECEILING ( date-part, datetime-expression [,multiple-expression])
```

### 参数

参数	描述
<i>date-part</i>	要添加到日期中的日期分量。
<i>datetime-expression</i>	包含要计算的值的日期、时间或日期时间表达式。
<i>multiple-expression</i>	(可选)。一个非零的正整数值表达式，用于指定要在计算中使用的由 <i>date-part</i> 参数指定的单位的倍数是多少。例如，您可以使用 <i>multiple-expression</i> 指定要将数据的间隔调整为 200 微秒或 10 分钟。  如果 <i>multiple-expression</i> 计算结果为零、计算结果为负数、是一个显式 NULL 常量，或者不是指定 <i>date-part</i> 的有效值，则 SAP Sybase IQ 会生成错误。如果 <i>multiple-expression</i> 计算结果为 NULL，则该函数的结果也为 NULL。

### 注释

此函数通过将所提供的值最大增加至具有指定粒度的最接近较大值，计算新日期、时间或日期时间值。如果包括可选 *multiple-expression* 参数，则此函数会将日期和时间最大增加至指定粒度的最接近指定倍数。

计算的日期和时间的数据类型与 *multiple-expression* 参数的数据类型匹配。

以下日期分量与 **DATECEILING** 不兼容：

- DayofYear
- WeekDay
- CalYearofWeek
- CalWeekofYear
- CalDayofWeek

如果对微秒、毫秒、秒、分钟或小时日期部分指定 *multiple-expression*，则 SAP Sybase IQ 会假定从下一个较大的粒度单位开始应用倍数：

- 微秒的倍数从当前秒开始
- 毫秒的倍数从当前秒开始
- 秒的倍数从当前分钟开始

- 分钟的倍数从当前小时开始
- 小时的倍数从当天开始

例如，如果指定一个 2 分钟的倍数，则 SAP Sybase IQ 会从当前小时开始应用两分钟的时间间隔。

对于微秒、毫秒、秒、分钟和小时日期分量，指定一个会平均划分到指定的日期分量范围的 *multiple-expression* 值：

- 对于小时，有效的 *multiple-expression* 值为：1、2、3、4、6、8、12、24
- 对于秒和分钟，有效的 *multiple-expression* 值为：1、2、3、4、5、6、10、12、15、20、30、60
- 对于毫秒，有效的 *multiple-expression* 值为：1、2、4、5、8、10、20、25、40、50、100、125、200、250、500、1000
- 对于微秒，有效的 *multiple-expression* 值为：

1	40	400	4000	40000
2	50	500	5000	50000
4	64	625	6250	62500
5	80	800	8000	100000
8	100	1000	10000	125000
10	125	1250	12500	200000
16	160	1600	15625	250000
20	200	2000	20000	500000
25	250	2500	25000	1000000
32	320	3125	31250	

如果您对天、周、月、季度或年日期部分指定 *multiple-expression*，则 SAP Sybase IQ 会假定时间间隔从最小日期值 (0001-01-01)、最小时间值 (00:00:00.000000) 或最小日期时间值 (0001-01-01.00:00:00.000000) 开始。例如，如果指定一个 10 天的倍数，则 SAP Sybase IQ 会从 0001-01-01 开始计算 10 天的时间间隔。

对于天、周、月、季度或年日期分量，不需要指定一个会平均划分到下一个较大的单位时间粒度的倍数。

如果 SAP Sybase IQ 舍入为周日期部分的倍数，则日期值始终为星期日。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 或 SQL Anywhere 支持。

### 示例

以下语句返回值 August 13, 2009 10:40.00.000AM:

```
SELECT DATECEILING( MI, 'August 13, 2009, 10:32.00.132AM', 10) FROM iq_dummy
```

以下语句返回值 August 13, 2009 10:32.35.456800 AM:

```
SELECT DATECEILING( US, 'August 13, 2009, 10:32.35.456789AM', 200 ) FROM iq_dummy
```

以下语句返回值 August 13, 2009 10:32.35.600000 AM:

```
SELECT DATECEILING( US, 'August 13, 2009, 10:32.35.456789AM', 200000 ) FROM iq_dummy
```

以下语句返回值 August 13, 2009 10:32.35.456789 AM:

```
SELECT DATECEILING( US, 'August 13, 2009, 10:32.35.456789AM') FROM iq_dummy
```

### 另请参见

- DATEADD 函数 [日期和时间] (第 165 页)
- DATEDIFF 函数 [日期和时间] (第 169 页)
- DATEFLOOR 函数 [日期和时间] (第 171 页)
- DATEPART 函数 [日期和时间] (第 176 页)
- DATENAME 函数 [日期和时间] (第 175 页)
- DATEROUND 函数 [日期和时间] (第 177 页)
- 日期分量 (第 85 页)

## **DATEDIFF 函数 [日期和时间]**

返回两个日期之间的时间间隔。

### 语法

```
DATEDIFF ( date-part, date-expression1, date-expression2 )
```

### 参数

参数	描述
date-part	指定要测量其时间间隔的日期分量。
date-expression1	时间间隔的开始日期。从 date-expression2 中减去该值，以返回两个参数之间日期分量的数量。
date-expression2	时间间隔的结束日期。从此值中减去 date-expression1 以返回两个参数间的日期分量的数量。

*返回*

INT

*注释*

此函数计算两个指定日期之间日期分量的数量。结果是带符号的整数值（以日期分量表示），它等于 **(date2 - date1)**。

当结果不是日期分量的偶数倍时，**DATEDIFF** 结果将被截断而不是被舍入。

当使用 **day** 作为日期部分时，**DATEDIFF** 返回两个指定时间之间（包括第二个日期但不包括第一个日期）的午夜数。例如，以下语句返回值 5。第一天 2003/08/03 的午夜未包括在结果中。第二天的午夜包括在内，即使指定的时间在午夜之前也是如此。

```
SELECT DATEDIFF( DAY, '2003/08/03 14:00', '2003/08/08 14:00' ) FROM iq_dummy
```

当使用 **month** 作为日期分量时，**DATEDIFF** 返回两个日期之间（包括第二个日期但不包括第一个日期）出现的月第一天的个数。例如，下面的两条语句都返回值 9：

```
SELECT DATEDIFF( MONTH, '2003/02/01', '2003/11/15' ) FROM iq_dummy;
SELECT DATEDIFF( MONTH, '2003/02/01', '2003/11/01' ) FROM iq_dummy;
```

第一个日期 2003/02/01 是月份第一天，但未包括在任意一个查询的结果中。第二个查询中的第二个日期 2003/11/01 也是月份第一天，并且包括在结果中。

当使用 **week** 作为日期分量时，**DATEDIFF** 返回两个日期之间（包括第二个日期但不包括第一个日期）的星期日的个数。例如，在月份 2003/08 中，星期日所在的日期是 03、10、17、24 和 31。下面的查询返回值 4：

```
SELECT DATEDIFF( week, '2003/08/03', '2003/08/31' ) FROM iq_dummy;
```

第一个星期日 (2003/08/03) 未包括在结果中。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

以下语句返回 1：

```
SELECT DATEDIFF( HOUR, '4:00AM', '5:50AM' )
FROM iq_dummy
```

以下语句返回 102：

```
SELECT DATEDIFF( MONTH, '1987/05/02', '1995/11/15' )
FROM iq_dummy
```

以下语句返回 0：

```
SELECT DATEDIFF( DAY, '00:00', '23:59' ) FROM iq_dummy
```

以下语句返回 4:

```
SELECT DATEDIFF( DAY, '1999/07/19 00:00', '1999/07/23
23:59' ) FROM iq_dummy
```

以下语句返回 0:

```
SELECT DATEDIFF( MONTH, '1999/07/19', '1999/07/23' )
FROM iq_dummy
```

以下语句返回 1:

```
SELECT DATEDIFF( MONTH, '1999/07/19', '1999/08/23' )
FROM iq_dummy
```

以下语句返回 4:

```
SELECT DATEDIFF(MCS, '2009-11-03 11:10:42.033185',
'2009-11-03 11:10:42.033189') FROM iq_dummy
```

以下语句返回 15:

```
SELECT DATEDIFF(MICROSECOND, '2009-11-10
14:57:52.722001', '2009-11-10 14:57:52.722016')
FROM iq_dummy
```

以下语句返回 1,500,000:

```
SELECT DATEDIFF(MCS, '2000/07/07/07 07:07:06.277777',
'2000/07/07/07 07:07:07.777777') FROM iq_dummy
```

另请参见

- DATEADD 函数 [日期和时间] (第 165 页)
- DATECEILING 函数 [日期和时间] (第 167 页)
- DATEFLOOR 函数 [日期和时间] (第 171 页)
- DATEPART 函数 [日期和时间] (第 176 页)
- DATENAME 函数 [日期和时间] (第 175 页)
- DATEROUND 函数 [日期和时间] (第 177 页)
- 日期分量 (第 85 页)

## **DATEFLOOR 函数 [日期和时间]**

通过将所提供的值最大减小至与具有指定粒度的指定倍数最接近的较小值，计算新日期、时间或 `datetime` 值。

语法

```
DATEFLOOR ( date-part, datetime-expression [,multiple-expression])
```

## 参数

参数	描述
日期分量	要添加到日期中的日期分量。
datetime-expression	包含要计算的值的日期、时间或日期时间表达式。
multiple-expression	<p>(可选)。一个非零的正整数值表达式，用于指定要在计算中使用的由 <b>date-part</b> 指定的单位的倍数是多少。例如，您可以使用 <b>multiple-expression</b> 指定要将数据的间隔调整为 200 微秒或 10 分钟。</p> <p>如果 <b>multiple-expression</b> 计算结果为零、计算结果为负数、是一个显式 NULL 常量，或者不是指定 <b>date-part</b> 的有效值，则 SAP Sybase IQ 会生成错误。如果 <b>multiple-expression</b> 计算结果为 NULL，则该函数的结果也为 NULL。</p>

## 注释

此函数通过将所提供的值最大减小至具有指定粒度的最接近较小值，计算新日期、时间或日期时间值。如果包括可选 *multiple-expression* 参数，则此函数会将日期和时间最大减小至指定粒度的最接近指定倍数。

计算的日期和时间的数据类型与 *multiple-expression* 参数的数据类型匹配。

以下日期分量与 **DATEFLOOR** 不兼容：

- DayofYear
- WeekDay
- CalYearofWeek
- CalWeekofYear
- CalDayofWeek

如果对微秒、毫秒、秒、分钟或小时日期部分指定 *multiple-expression*，则 SAP Sybase IQ 会假定从下一个较大的粒度单位开始应用倍数：

- 微秒的倍数从当前秒开始
- 毫秒的倍数从当前秒开始
- 秒的倍数从当前分钟开始
- 分钟的倍数从当前小时开始
- 小时的倍数从当天开始

例如，如果指定一个 2 分钟的倍数，则 SAP Sybase IQ 会从当前小时开始应用两分钟的时间间隔。

对于微秒、毫秒、秒、分钟和小时日期分量，指定一个会平均划分到指定的日期分量范围的 *multiple-expression* 值：



- 对于小时，有效的 *multiple-expression* 值为：1、2、3、4、6、8、12、24
- 对于秒和分钟，有效的 *multiple-expression* 值为：1、2、3、4、5、6、10、12、15、20、30、60
- 对于毫秒，有效的 *multiple-expression* 值为：1、2、4、5、8、10、20、25、40、50、100、125、200、250、500、1000
- 对于微秒，有效的 *multiple-expression* 值为：

1	40	400	4000	40000
2	50	500	5000	50000
4	64	625	6250	62500
5	80	800	8000	100000
8	100	1000	10000	125000
10	125	1250	12500	200000
16	160	1600	15625	250000
20	200	2000	20000	500000
25	250	2500	25000	1000000
32	320	3125	31250	

如果您对天、周、月、季度或年日期部分指定 *multiple-expression*，则 SAP Sybase IQ 会假定时间间隔从最小日期值 (0001-01-01)、最小时间值 (00:00:00.000000) 或最小日期时间值 (0001-01-01.00:00:00.000000) 开始。例如，如果指定一个 10 天的倍数，则 SAP Sybase IQ 会从 0001-01-01 开始计算 10 天的时间间隔。

对于天、周、月、季度或年日期分量，不需要指定一个会平均划分到下一个较大的单位时间粒度的倍数。

如果 SAP Sybase IQ 舍入为周日期部分的倍数，则日期值始终为星期日。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 或 SQL Anywhere 支持。

#### 示例

- 以下语句返回值 August 13, 2009 10:35:00.000AM:

```
SELECT DATEFLOOR( MINUTE, 'August 13, 2009 10:35:22.123AM') FROM iq_dummy
```

- 以下语句返回值 August 13, 2009 10:32:35.456600 AM:

```
SELECT DATEFLOOR( US, 'August 13, 2009, 10:32:35.456789AM', 200 ) FROM iq_dummy
```

## SQL 函数

- 以下语句返回值 August 13, 2009 10:32:35.400000 AM:

```
SELECT DATEFLOOR( US, 'August 13, 2009, 10:32:35.456789AM',  
200000 ) FROM iq_dummy
```

- 以下语句返回值 August 13, 2009 10:32:35.456789 AM:

```
SELECT DATEFLOOR( US, 'August 13, 2009, 10:32:35.456789AM') FROM  
iq_dummy
```

### 另请参见

- DATEADD 函数 [日期和时间] (第 165 页)
- DATECEILING 函数 [日期和时间] (第 167 页)
- DATEDIFF 函数 [日期和时间] (第 169 页)
- DATEPART 函数 [日期和时间] (第 176 页)
- DATENAME 函数 [日期和时间] (第 175 页)
- DATEROUND 函数 [日期和时间] (第 177 页)
- 日期分量 (第 85 页)

## DATEFORMAT 函数 [日期和时间]

以指定的格式返回表示日期表达式的字符串。

### 语法

```
DATEFORMAT ( datetime-expression, string-expression )
```

### 参数

参数	描述
<i>datetime-expression</i>	要转换的日期/时间。必须是日期、时间、时间戳或字符串。
<i>string-expression</i>	转换后的日期格式。

### 返回

VARCHAR

### 注释

要转换的 *datetime-expression* 必须是日期、时间或时间戳数据类型，但也可以是 CHAR 或 VARCHAR 字符串。如果日期是字符串，SAP Sybase IQ 会隐式将字符串转换为日期、时间或时间戳数据类型，因此不需要像上例中那样执行显式转换。

任何允许的日期格式都可用于 *string-expression*。日期格式字符串不能包含任何多字节字符。即使数据库的归类顺序是多字节归类顺序（如 932JPN），也只能在日期/时间/日期时间格式的字符串中使用单字节字符。

如果 “?” 表示多字节字符，则下面的查询会失败：

```
SELECT DATEFORMAT ( start_date, 'yy?' ) FROM Employees;
```

而应使用并置运算符移动日期格式字符串外面的多字节字符：

```
SELECT DATEFORMAT (start_date, 'yy' ) + '?' FROM Employees;
```

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 或 SQL Anywhere 支持。

### 示例

以下语句返回类似 "Jan 01, 1989" 的字符串值:

```
SELECT DATEFORMAT( start_date, 'Mmm dd, yyyy' ) from Employees;
```

以下语句返回字符串 "Feb 19, 1987":

```
SELECT DATEFORMAT( CAST ( '1987/02/19' AS DATE ), 'Mmm Dd, yyyy' )
FROM iq_dummy
```

## DATENAME 函数 [日期和时间]

以字符串形式返回日期/时间值中指定部分的名称（如月份 “June”）。

### 语法

```
DATENAME ( date-part, date-expression )
```

### 参数

表 18. 参数

参数	描述
date-part	要指定的日期分量。
date-expression	要返回其日期分量名称的日期。日期必须包含请求的 date-part。

### 返回

VARCHAR

### 注释

**DATENAME** 返回一个字符串，即使结果是数值（如表示日期的 23）也是如此。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 示例

以下语句返回值 May:

```
SELECT DATENAME( MONTH , '1987/05/02' )
FROM iq_dummy
```

## SQL 函数

以下语句返回值 722,001:

```
SELECT DATENAME(MICROSECOND, '2009-11-10  
14:57:52.722001') FROM iq_dummy
```

以下语句返回值 777,777:

```
SELECT DATENAME(MICROSECOND, '2000/07/07  
07:07:07.777777') FROM iq_dummy
```

以下语句返回值 33,189:

```
SELECT DATENAME(MCS, '2009-11-03 11:10:42.033189')  
FROM iq_dummy
```

### 另请参见

- DATEADD 函数 [日期和时间] (第 165 页)
- DATECEILING 函数 [日期和时间] (第 167 页)
- DATEDIFF 函数 [日期和时间] (第 169 页)
- DATEFLOOR 函数 [日期和时间] (第 171 页)
- DATEPART 函数 [日期和时间] (第 176 页)
- DATEROUND 函数 [日期和时间] (第 177 页)
- 日期分量 (第 85 页)

## DATEPART 函数 [日期和时间]

返回日期/时间值的指定部分的整数值。

### 语法

```
DATEPART ( date-part, date-expression )
```

### 参数

表 19. 参数

参数	描述
date-part	要返回的日期分量。
date-expression	要返回其日期分量的日期。日期必须包含 date-part 字段。

### 返回

INT

### 注释

**DATE**、**TIME** 和 **DTTM** 索引不支持某些日期分量 (Calyearofweek、Calweekofyear、Caldayofweek、Dayofyear、Millisecond、Microsecond)。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 示例

以下语句返回值 5:

```
SELECT DATEPART( MONTH, '1987/05/02' )
FROM iq_dummy
```

以下语句返回值 722,001:

```
SELECT DATEPART(MICROSECOND, '2009-11-10
14:57:52.722001') FROM iq_dummy
```

以下语句返回值 777,777:

```
SELECT DATEPART(MICROSECOND, '2000/07/07
07:07:07.777777') FROM iq_dummy
```

以下语句返回值 33,189:

```
SELECT DATEPART(MCS, '2009-11-03 11:10:42.033189')
FROM iq_dummy
```

### 另请参见

- DATEADD 函数 [日期和时间] (第 165 页)
- DATECEILING 函数 [日期和时间] (第 167 页)
- DATEDIFF 函数 [日期和时间] (第 169 页)
- DATEFLOOR 函数 [日期和时间] (第 171 页)
- DATENAME 函数 [日期和时间] (第 175 页)
- DATEROUND 函数 [日期和时间] (第 177 页)
- 日期分量 (第 85 页)

## **DATEROUND 函数 [日期和时间]**

通过将所提供的值向上或向下舍入为与具有指定粒度的指定值最接近的倍数，计算新日期、时间或 `datetime` 值。

### 语法

```
DATEROUND (date-part, datetime-expression [,multiple-expression] )
```

### 参数

参数	描述
日期分量	要返回的日期分量。

参数	描述
datetime-expression	包含要计算的值的日期、时间或日期时间表达式。
multiple-expression	<p>(可选)。一个非零的正整数值表达式，用于指定要在计算中使用的由 <i>date-part</i> 指定的单位的倍数是多少。例如，您可以使用 <i>multiple-expression</i> 指定要将数据的间隔调整为 200 微秒或 10 分钟。</p> <p>如果 <i>multiple-expression</i> 计算结果为零、计算结果为负数、是一个显式 NULL 常量，或者不是指定 <i>date-part</i> 的有效值，则 SAP Sybase IQ 会生成错误。如果 <i>multiple-expression</i> 计算结果为 NULL，则该函数的结果也为 NULL。</p>

#### 注释

此函数通过将所提供的值向上或向下舍入为具有指定粒度的最接近的值，计算新日期、时间或者日期时间值。如果包括可选 *multiple-expression* 参数，则此函数会将日期和时间舍入至指定粒度的最接近指定倍数。

计算的日期和时间的数据类型与 *multiple-expression* 参数的数据类型匹配。

以下日期分量与 **DATEROUND** 不兼容：

- DayofYear
- WeekDay
- CalYearofWeek
- CalWeekofYear
- CalDayofWeek

如果对微秒、毫秒、秒、分钟或小时日期部分指定 *multiple-expression*，则 SAP Sybase IQ 会假定从下一个较大的粒度单位开始应用倍数：

- 微秒的倍数从当前秒开始
- 毫秒的倍数从当前秒开始
- 秒的倍数从当前分钟开始
- 分钟的倍数从当前小时开始
- 小时的倍数从当天开始

例如，如果指定一个 2 分钟的倍数，则 SAP Sybase IQ 会从当前小时开始应用两分钟的时间间隔。

对于微秒、毫秒、秒、分钟和小时日期分量，指定一个会平均划分到指定的日期分量范围的 *multiple-expression* 值：

- 对于小时，有效的 *multiple-expression* 值为：1、2、3、4、6、8、12、24
- 对于秒和分钟，有效的 *multiple-expression* 值为：1、2、3、4、5、6、10、12、15、20、30、60

- 对于毫秒，有效的 *multiple-expression* 值为：1、2、4、5、8、10、20、25、40、50、100、125、200、250、500、1000
- 对于微秒，有效的 *multiple-expression* 值为：

1	40	400	4000	40000
2	50	500	5000	50000
4	64	625	6250	62500
5	80	800	8000	100000
8	100	1000	10000	125000
10	125	1250	12500	200000
16	160	1600	15625	250000
20	200	2000	20000	500000
25	250	2500	25000	1000000
32	320	3125	31250	

如果您对天、周、月、季度或年日期部分指定 *multiple-expression*，则 SAP Sybase IQ 会假定时间间隔从最小日期值 (0001-01-01)、最小时间值 (00:00:00.000000) 或最小日期时间值 (0001-01-01.00:00:00.000000) 开始。例如，如果指定一个 10 天的倍数，则 SAP Sybase IQ 会从 0001-01-01 开始计算 10 天的时间间隔。

对于天、周、月、季度或年日期分量，不需要指定一个会平均划分到下一个较大的单位时间粒度的倍数。

如果 SAP Sybase IQ 舍入为周日期部分的倍数，则日期值始终为星期日。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 或 SQL Anywhere 支持。

#### 示例

以下语句返回值 August 13, 2009, 10:30.000AM:

```
SELECT DATEROUND( MI, 'August 13, 2009 10:33.123AM', 10) FROM
iq_dummy
```

以下语句返回值 August 13, 2009 10:32:35.456600 AM:

```
SELECT DATEROUND( US, 'August 13, 2009, 10:32:35.456500AM', 200 )
FROM iq_dummy
```

以下语句返回值 August 13, 2009 10:32:35.456789 AM:

```
SELECT DATEROUND( US, 'August 13, 2009, 10:32:35.456789AM') FROM
iq_dummy
```

## SQL 函数

以下语句返回值 **August 13, 2009 10:32:35.456400 AM**:

```
SELECT DATEROUND( US, 'August 13, 2009, 10:32:35.456499AM', 200 )  
FROM iq_dummy
```

### 另请参见

- DATEADD 函数 [日期和时间] (第 165 页)
- DATECEILING 函数 [日期和时间] (第 167 页)
- DATEDIFF 函数 [日期和时间] (第 169 页)
- DATEFLOOR 函数 [日期和时间] (第 171 页)
- DATEPART 函数 [日期和时间] (第 176 页)
- DATENAME 函数 [日期和时间] (第 175 页)
- 日期分量 (第 85 页)

## **DATETIME 函数 [日期和时间]**

将表达式转换成时间戳。

### 语法

```
DATETIME ( expression )
```

### 参数

参数	描述
表达式	要转换的表达式。表达式通常是字符串。可能会报告转换错误。

### 返回

TIMESTAMP

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 或 SQL Anywhere 支持。

### 示例

以下语句返回值为 **1998-09-09 12:12:12.000** 的时间戳:

```
SELECT DATETIME( '1998-09-09 12:12:12.000' ) FROM iq_dummy
```

## **DAY 函数 [日期和时间]**

返回 1 到 31 之间的一个整数，该整数对应于指定日期中的每月几号。

### 语法

```
DAY ( date-expression )
```



参数

表 20. 参数

参数	描述
date-expression	日期。

返回

SMALLINT

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

示例

以下语句返回值 12:

```
SELECT DAY( '2001-09-12' ) FROM iq_dummy
```

## **DAYNAME 函数 [日期和时间]**

返回指定日期中的星期几的名称。

语法

```
DAYNAME ( date-expression )
```

参数

表 21. 参数

参数	描述
date-expression	日期。

返回

VARCHAR

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

示例

以下语句返回值 Saturday:

```
SELECT DAYNAME ( '1987/05/02' ) FROM iq_dummy
```

**DAYS 函数 [日期和时间]**

返回自任意开始日期以来的天数，返回两个指定日期之间的天数，或者将指定的 *integer-expression* 数量的天添加到给定日期中。

**DAYS** 忽略小时、分钟和秒。

*语法*

```
DAYS ( datetime-expression )
| ( datetime-expression, datetime-expression )
| ( datetime-expression, integer-expression )
```

*参数*

表 22.

参数	描述
<i>datetime-expression</i>	日期和时间。
<i>integer-expression</i>	要添加到 <i>datetime-expression</i> 中的天数。如果 <i>integer-expression</i> 是负数，则从日期/时间中减去相应的天数。如果提供整数表达式，必须将 <i>datetime-expression</i> 显式转换为日期。

*返回*

INT，如果指定两个日期时间表达式。

TIMESTAMP，如果指定的第二个参数为整数。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句返回值 729948:

```
SELECT DAYS( '1998-07-13 06:07:12' ) FROM iq_dummy
```

以下语句返回整数 -366，表示两个日期之间的差值:

```
SELECT DAYS( '1998-07-13 06:07:12',
'1997-07-12 10:07:12' ) FROM iq_dummy
```

以下语句返回值 1999-07-14:

```
SELECT DAYS( CAST('1998-07-13' AS DATE ), 366 )
FROM iq_dummy
```

另请参见

- CAST 函数 [数据类型转换] (第 143 页)

## **DB\_ID 函数 [系统]**

返回数据库 ID 号。

语法

```
DB_ID ( [ database-name ] )
```

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

参数

参数	描述
database-name	包含数据库名称的字符串表达式。如果 database-name 是字符串常量，则它必须用引号引起来。如果未提供 database-name，则返回当前数据库的 ID 号。

返回

INT

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。

示例

如果 iqdemo 是唯一正在运行的数据库，则返回值 0:

```
SELECT DB_ID( 'iqdemo' ) FROM iq_dummy
```

如果针对唯一正在运行的数据库执行，则返回值 0:

```
SELECT DB_ID() FROM iq_dummy
```

另请参见

- COL\_NAME 函数 [系统] (第 151 页)
- DB\_NAME 函数 [系统] (第 184 页)
- DB\_PROPERTY 函数 [系统] (第 185 页)
- NEXT\_DATABASE 函数 [系统] (第 250 页)
- OBJECT\_ID 函数 [系统] (第 257 页)
- OBJECT\_NAME 函数 [系统] (第 257 页)

**DB\_NAME 函数 [系统]**

返回数据库名称。

语法

```
DB_NAME ( [ database-id ] )
```

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

参数

参数	描述
database-id	数据库的 ID。database-id 必须是数值表达式。

返回

VARCHAR

注释

如果未提供 *database-id*，则返回当前数据库的名称。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。

示例

如果针对演示数据库执行，则返回数据库名称 **iqdemo**：

```
SELECT DB_NAME( 0 ) FROM iq_dummy
```

另请参见

- COL\_NAME 函数 [系统] (第 151 页)
- DB\_ID 函数 [系统] (第 183 页)
- DB\_PROPERTY 函数 [系统] (第 185 页)
- NEXT\_DATABASE 函数 [系统] (第 250 页)
- OBJECT\_ID 函数 [系统] (第 257 页)
- OBJECT\_NAME 函数 [系统] (第 257 页)

## DB\_PROPERTY 函数 [系统]

返回给定属性的值。

语法

```
DB_PROPERTY ( { property-id | property-name }  
[ , { database-id | database-name } ] )
```

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

参数

表 23. 参数

参数	描述
property-id	数据库属性 ID。
property-name	数据库属性名称。
database-id	数据库 ID 号，由 DB_ID 返回。通常情况下，使用数据库名称。
database-name	数据库的名称，由 DB_NAME 返回。

返回

VARCHAR

注释

返回字符串。如果省略第二个参数，则使用当前数据库。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

示例

以下语句返回当前数据库的页面大小（以字节为单位）。

```
SELECT DB_PROPERTY( 'PAGESIZE' ) FROM iq_dummy
```

另请参见

- COL\_NAME 函数 [系统]（第 151 页）
- DB\_ID 函数 [系统]（第 183 页）
- DB\_NAME 函数 [系统]（第 184 页）
- NEXT\_DATABASE 函数 [系统]（第 250 页）
- OBJECT\_ID 函数 [系统]（第 257 页）

## SQL 函数

- OBJECT\_NAME 函数 [系统] (第 257 页)

### **DEGREES 函数 [数值]**

将数字从弧度转换为度数。

语法

```
DEGREES ( numeric-expression )
```

参数

表 24. 参数

参数	描述
numeric-expression	以弧度表示的角度。

返回

返回由 *numeric-expression* 所指定的角的度数。

DOUBLE

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

示例

以下语句返回值 29.793805:

```
SELECT DEGREES( 0.52 ) FROM iq_dummy
```

### **DENSE\_RANK 函数 [分析]**

排列组中的项目。

语法

```
DENSE_RANK () OVER ( ORDER BY expression [ ASC | DESC ] )
```

参数

表 25. 参数

参数	描述
表达式	排序规范，可以是涉及列引用、集合的任意有效表达式，也可以是调用这些项目的表达式。

*返回*

INTEGER

*注释*

**DENSE\_RANK** 是 rank 分析函数。行 R 的密集排名是指位于该行前面（包括行 R）并且在 **OVER** 子句指定的组中或者在整个结果集中保持不重复的行数。**DENSE\_RANK** 和 **RANK** 之间的区别是：当存在并列排名时，**DENSE\_RANK** 不会保留空位，而 **RANK** 会保留空位。

**DENSE\_RANK** 需要使用 **OVER (ORDER BY)** 子句。**ORDER BY** 子句指定要执行排序的参数以及每个组中行的排列顺序。此 **ORDER BY** 子句只在 **OVER** 子句中使用，而不是 **SELECT** 的 **ORDER BY**。不允许排序查询中的任何集合函数指定 **DISTINCT**。

**OVER** 子句表示函数对查询结果集进行操作。结果集是在对 **FROM**、**WHERE**、**GROUP BY** 和 **HAVING** 子句求值完成之后返回的行。**OVER** 子句定义要包括在 rank 分析函数计算中的行数数据集。

ASC 或 DESC 参数用于指定升序或降序排序序列。升序是缺省值。

**DENSE\_RANK** 只能在 **SELECT** 或 **INSERT** 语句的选择列表中或者 **SELECT** 语句的 **ORDER BY** 子句中使用。**DENSE\_RANK** 可以在视图中或联合中使用。您不能在子查询中、**HAVING** 子句中，或者 **UPDATE** 或 **DELETE** 语句的选择列表中使用 **DENSE\_RANK** 函数。每个查询仅允许使用一个 rank 分析函数。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 或 SQL Anywhere 支持。

*示例*

以下语句说明了 **DENSE\_RANK** 函数的用法：

```
SELECT s_suppkey, DENSE_RANK()
OVER ( ORDER BY ( SUM(s_acctBal) ) DESC )
AS rank_dense FROM supplier GROUP BY s_suppkey;
```

s_suppkey	sum_acctBal	rank_dense
supplier#011	200,000	1
supplier#002	200,000	1
supplier#013	123,000	2
supplier#004	110,000	3
supplier#035	110,000	3
supplier#006	50,000	4
supplier#021	10,000	5

*另请参见*

- **RANK** 函数 [分析]（第 273 页）

**DIFFERENCE 函数 [字符串]**

比较两个字符串，评估它们之间的相似性，并返回 0 到 4 之间的一个值。  
最佳匹配值是 4。

*语法*

```
DIFFERENCE ( string-expression1, string-expression2 )
```

*参数*

表 26. 参数

参数	描述
string-expression1	要比较的第一个字符串。
string-expression2	要比较的第二个字符串。

*返回*

SMALLINT

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*另请参见*

- SOUNDEX 函数 [字符串] (第 304 页)

**DIFFERENCE 函数示例**

使用示例作为 **DIFFERENCE** 函数用法的参考。

下面的语句返回值 4:

```
SELECT DIFFERENCE( 'Smith', 'Smith' ) FROM iq_dummy
```

下面的语句返回值 4:

```
SELECT DIFFERENCE( 'Smith', 'Smyth' ) FROM iq_dummy
```

下面的语句返回值 3:

```
SELECT DIFFERENCE( 'Smith', 'Sweeney' ) FROM iq_dummy
```

下面的语句返回值 2:

```
SELECT DIFFERENCE( 'Smith', 'Jones' ) FROM iq_dummy
```

下面的语句返回值 1:



```
SELECT DIFFERENCE( 'Smith', 'Rubin' ) FROM iq_dummy
```

下面的语句返回值 0:

```
SELECT DIFFERENCE( 'Smith', 'Wilkins' ) FROM iq_dummy
```

## **DOW 函数 [日期和时间]**

返回 1 到 7 之间的一个数字，表示指定日期中的星期几，星期日=1、星期一=2，依此类推。

*语法*

```
DOW ( date-expression )
```

*参数*

表 27. 参数

参数	描述
date-expression	日期。

*返回*

SMALLINT

*注释*

如果要使星期一（或另一天）成为一周中的第一天，请使用 DATE\_FIRST\_DAY\_OF\_WEEK 选项。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句返回值 5:

```
SELECT DOW( '1998-07-09' ) FROM iq_dummy
```

## **ERRORMSG 函数 [杂项]**

提供当前错误或指定的 SQLSTATE 或 SQLCODE 值的错误消息。

*语法*

```
ERRORMSG  
( [ sqlstate | sqlcode ] )
```

```
sqlstate: string
```

```
sqlcode: integer
```

### 参数

参数	定义
sqlstate	要返回其错误消息的 SQLSTATE 值。
sqlcode	要返回其错误消息的 SQLCODE 值。

### 返回

包含错误消息的字符串。

### VARCHAR

### 注释

如果未提供参数，则提供当前状态的错误消息。将进行任何替代（如表名和列名）。

如果提供了参数，则返回提供的 SQLSTATE 或 SQLCODE 的错误消息，没有任何替代项。将作为占位符（“???”）提供表名和列名。

**ERRORMSG** 函数返回 SQL Anywhere 和 SAP Sybase IQ 错误消息。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 支持。

### 示例

以下语句返回 SQLCODE -813 的错误消息：

```
select errmsg( -813 )
```

## **EVENT\_CONDITION 函数 [系统]**

指定何时触发事件处理程序。

若要定义事件及其关联的处理程序，请使用 **CREATE EVENT** 语句。

### 语法

```
EVENT_CONDITION ( condition-name )
```

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

参数

表 28. 参数

参数	定义
condition-name	触发事件的条件。可能的值已经在数据库中预设，并且不区分大小写。每个条件仅对某些特定的事件类型有效。

表 29. 事件的有效条件

条件名称	单位	适用于	注释
DBFreePercent	不适用	DBDiskSpace	DBDiskSpace 显示了系统数据库文件 (.db 文件) 而不是 IQ 存储中的可用空间。
DBFreeSpace	兆字节	DBDiskSpace	
DBSize	兆字节	GrowDB	
ErrorNumber	不适用	RAISERROR	
IdleTime	秒	ServerIdle	
间隔	秒	全部	自上次执行处理程序以来的时间。
LogFreePercent	不适用	LogDiskSpace	
LogFreeSpace	兆字节	LogDiskSpace	
LogSize	兆字节	GrowLog	
RemainingValues	整数	GlobalAutoincrement	剩余值的数目。
TempFreePercent	不适用	TempDiskSpace	TempDiskSpace 显示 TEMP 或 IQTMP16 环境变量所指向的系统临时文件的可用空间，而不是 IQ 临时存储库中的可用空间。
TempFreeSpace	兆字节	TempDiskSpace	
TempSize	兆字节	GrowTemp	

返回

INT

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

示例

以下事件定义使用了 **EVENT\_CONDITION** 函数:

```
create event LogNotifier
type LogDiskSpace
where event_condition( 'LogFreePercent' ) < 50
handler
begin
    message 'LogNotifier message'
end
```

另请参见

- **EVENT\_PARAMETER** 函数 [系统] (第 192 页)

**EVENT\_CONDITION\_NAME** 函数 [系统]

可用于列出 **EVENT\_CONDITION** 的可能参数。

若要定义事件及其关联的处理程序，请使用 **CREATE EVENT** 语句。

语法

```
EVENT_CONDITION_NAME ( integer )
```

---

**注意:** 需要考虑 CIS 函数补偿性能方面的一些事宜。

---

参数

参数	描述
整数	必须大于或等于零。

返回

VARCHAR

注释

可以使用 **EVENT\_CONDITION\_NAME** 获得所有 **EVENT\_CONDITION** 参数的列表，方法是：在整数中循环，直到函数返回空值。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

**EVENT\_PARAMETER** 函数 [系统]

提供事件处理程序的上下文信息。

若要定义事件及其关联的处理程序，请使用 **CREATE EVENT** 语句。

## 语法

```
EVENT_PARAMETER ( context-name )
```

```
context-name:
'ConnectionID'
| 'User'
| 'EventName'
| 'Executions'
| 'IQDBMainSpaceName'
| 'NumActive'
| 'TableName'
| condition-name
```

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

## 参数

表 30. 参数

参数	描述
context-name	其中一个预设字符串。这些字符串不区分大小写，它们传递下列信息：
ConnectionId	连接 ID，由以下语句返回 <code>connection_property( 'id' )</code>
User	导致事件被触发的用户的用户 ID。
EventName	已触发的事件的名称。
Executions	事件处理程序执行的次数。
NumActive	事件处理程序的活动实例数。如果想限制事件处理程序以便在任何给定的时间仅执行一个实例，则这很有用。
TableName	表名，与 RemainingValues 一起使用。

此外，您可以从 **EVENT\_PARAMETER** 函数访问 **EVENT\_CONDITION** 函数的任何有效的 *condition-name* 参数。

## 返回

VARCHAR

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

另请参见

- `EVENT_CONDITION` 函数 [系统] (第 190 页)

## **EXP 函数 [数值]**

返回指数函数，即 e 的指定数字次乘方。

语法

```
EXP ( numeric-expression )
```

参数

表 31. 参数

参数	描述
<code>numeric-expression</code>	指数。

返回

DOUBLE

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

示例

以下语句返回值 3269017.3724721107:

```
SELECT EXP( 15 ) FROM iq_dummy
```

## **EXP\_WEIGHTED\_AVG 函数 [集合]**

计算指数加权移动平均值。

加权确定构成平均值的每个数量的相对重要性。

语法

```
EXP_WEIGHTED_AVG ( expression, period-expression )
```

```
OVER ( window-spec )
```

参数

参数	描述
表达式	计算其加权值的数值表达式。
<code>period-expression</code>	指定要计算平均值的时间段的数值表达式。

### 注释

与 **WEIGHTED\_AVG** 函数类似，随着时间推移，**EXP\_WEIGHTED\_AVG** 的权重会减小。但是，**WEIGHTED\_AVG** 的权重按算术级数减小，而 **EXP\_WEIGHTED\_AVG** 的权重按指数级数减小。指数加权会为最新的值应用较多权重，而减小较旧值的权重，同时仍旧为较旧值应用一些权重。

SAP Sybase IQ 使用以下公式计算指数移动平均值：

$$S * C + (1 - S) * PEMA$$

在上面的计算中，SAP Sybase IQ 应用平滑系数，方法是将当前收盘价格 (C) 乘以平滑常量 (S)，然后加上前一天指数移动平均值 (PEMA) 与 1 减去平滑系数的差的乘积。

SAP Sybase IQ 在 **OVER** 子句指定的整个时间段中计算指数移动平均值。*period-expression* 指定指数移动平均值的移动范围。

您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。*window-spec* 必须包含 **ORDER BY** 语句，不能包含构架规范。

---

**注意：** **GROUP BY** 子句不支持 **ROLLUP** 和 **CUBE**。**DISTINCT** 不受支持。

---

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

### 示例

以下示例返回佛罗里达州雇员的指数加权平均薪水，其中最近雇用员工的薪水在该平均值中占最大权重。以下三行用在加权之中：

```
SELECT DepartmentID, Surname, Salary, EXP_WEIGHTED_AVG(Salary, 3)
OVER (ORDER BY YEAR(StartDate) DESC) as "W_AVG" FROM Employees WHERE
State IN ('FL') ORDER BY StartDate DESC
```

返回的结果集为：

**表 32. EXP\_WEIGHTED\_AVG 结果集**

DepartmentID	Surname	Salary	W_AVG
400	Evans	68,940.000	34,470.000000
300	Litton	58,930.000	46,700.000000
200	Sterling	64,900.000	55,800.000000
200	Kelly	87,500.000	71,650.000000
400	Charlton	28,300.000	49,975.000000
100	Lull	87,900.000	68,937.500000

DepartmentID	Surname	Salary	W_AVG
100	Gowda	59,840.000	60,621.875000
400	Francis	53,870.000	61,403.750000

另请参见

- `WEIGHTED_AVG` 函数 [集合] (第 337 页)
- 窗口化集合函数的用法 (第 78 页)

## **FIRST\_VALUE** 函数 [集合]

返回一组值中的第一个值。

语法

**FIRST\_VALUE** (*expression* [IGNORE NULLS | RESPECT NULLS])

**OVER** (*window-spec*)

参数

参数	描述
<i>expression</i>	要对其确定有序集合中第一个值的表达式。

返回

参数的数据类型。

注释

**FIRST\_VALUE** 返回一组值中的第一个值，该组通常是有序集合。如果该集合中的第一个值为空值，则此函数返回空值，除非您指定 **IGNORE NULLS**。如果您指定 **IGNORE NULLS**，则 **FIRST\_VALUE** 返回该集合中的第一个非空值，或者，如果所有值均为空值，则返回空值。

返回值的数据类型与输入值相同。

不能对 *expression* 使用 **FIRST\_VALUE** 或任何其它分析函数。换言之，不能嵌套使用分析函数，但可以将其它内置函数表达式用作 *expression*。

如果 *window-spec* 不包含 **ORDER BY** 表达式，或者 **ORDER BY** 表达式不够精确，无法保证唯一顺序，则结果将为任意值。如果没有 *window-spec*，结果将为任意值。

您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

---

**注意：** **DISTINCT** 不受支持。

---



## 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

## 示例

以下示例返回每个雇员的工资和同一部门中最近雇用的雇员的工资之间的关系，以百分比表示：

```
SELECT DepartmentID, EmployeeID,
100 * Salary / ( FIRST_VALUE( Salary ) OVER (
PARTITION BY DepartmentID ORDER BY Year(StartDate) DESC ) )
AS percentage
FROM Employees order by DepartmentID DESC;
```

返回的结果集为：

**表 33. FIRST\_VALUE 结果集**

DepartmentID	EmployeeID	百分比
500	1,658	100.0000000000000000000000
500	1,570	138.842709713689113761394
500	1,615	110.428462434244870095972
500	1,013	109.585190539292454724330
500	750	137.734409508894510701521
500	921	167.449704854836766654619
500	868	113.239368750752921334778
500	703	222.867927558928643135365
500	191	119.664297474199895594908
400	1,684	100.0000000000000000000000
400	1,740	76.128652163477274215016
400	1,751	76.353400685155687446813
400	1,607	133.758100765890593292456
400	1,507	77.996465120338650199655
400	1,576	150.428767810774836893669

## SQL 函数

在此示例中，雇员 1658 是部门 500 的第一行，表示雇员 1658 是该部门最近雇用的员工，因此获得的百分比为 100%。相对于雇员 1658 的百分比计算部门 500 中其余雇员的百分比。例如，雇员 1570 的收入大约是雇员 1658 的收入的 139%。

### 另请参见

- 窗口化集合函数的用法（第 78 页）

## **FLOOR** 函数 [数值]

返回一个数字的下限（不大于的最大整数）。

### 语法

```
FLOOR ( numeric-expression )
```

### 参数

表 34. 参数

参数	描述
numeric-expression	数字，通常是浮点数。

### 返回

DOUBLE

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 示例

以下语句返回值 123.00:

```
SELECT FLOOR ( 123 ) FROM iq_dummy
```

以下语句返回值 123:

```
SELECT FLOOR ( 123.45 ) FROM iq_dummy
```

以下语句返回值 -124.00。

```
SELECT FLOOR ( -123.45 ) FROM iq_dummy
```

### 另请参见

- CEILING 函数 [数值]（第 145 页）
- CEIL 函数 [数值]（第 144 页）

## GETDATE 函数 [日期和时间]

返回当前的日期和时间。

语法

```
GETDATE ( )
```

返回

TIMESTAMP

注释

**GETDATE** 是与 Transact-SQL 兼容的数据操作函数。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

示例

以下语句返回系统日期和时间。

```
SELECT GETDATE ( ) FROM iq_dummy
```

## GRAPHICAL\_PLAN 函数 [字符串]

以 XML 格式字符串的形式向 Interactive SQL 返回图形查询计划。

语法

```
GRAPHICAL_PLAN ( string-expression  
[, statistics-level  
[, cursor-type  
[, update-status ]]])
```

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

参数

参数	描述
string-expression	生成计划所使用的 SQL 语句。string-expression 通常是 <b>SELECT</b> 语句，但也可以是 <b>UPDATE</b> 或 <b>DELETE</b> 、 <b>INSERT SELECT</b> 或 <b>SELECT INTO</b> 语句。
statistics-level	一个整数。statistics-level 可以是： <ul style="list-style-type: none"> <li>• 0 - 仅优化程序估计值（缺省值）。</li> <li>• 2 - 包括节点统计在内的详细统计。</li> <li>• 3 - 详细统计。</li> </ul>

参数	描述
cursor-type	一种游标类型，以字符串表示。可能的值有：asensitive、insensitive、sensitive 或 keyset-driven。如果未指定 cursor-type，则缺省情况下使用 asensitive。
update-status	字符串参数，它采用下列值之一来指示优化程序如何处理给定的游标： READ-ONLY - 游标是只读的。 READ-WRITE (缺省值) - 可以读取或写入游标。

*返回*

## LONG VARCHAR

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **GRAPHICAL\_PLAN**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **GRAPHICAL\_PLAN** 设置为正确的数据类型和大小。

*注释*

如果未向 **GRAPHICAL\_PLAN** 函数提供参数，则系统会从高速缓存将查询计划返回给您。如果高速缓存中没有查询计划，则会显示下面的消息：

```
plan not available
```

**GRAPHICAL\_PLAN** 函数的行为由数据库选项 **QUERY\_PLAN\_TEXT\_ACCESS** 和 **QUERY\_PLAN\_TEXT\_CACHING** 控制。如果 **QUERY\_PLAN\_TEXT\_ACCESS** 为 **OFF** (缺省值)，则系统将显示以下消息：

```
Plan not available. The database option QUERY_PLAN_TEXT_ACCESS is OFF
```

如果用户需要访问计划，则必须由具备 **SET ANY SYSTEM OPTION** 系统特权的用户为该用户将选项 **QUERY\_PLAN\_TEXT\_ACCESS** 设置为 **ON**。

如果 **QUERY\_PLAN\_TEXT\_ACCESS** 为 **ON**，且该字符串表达式的查询计划可用于服务器所维护的高速缓存中，则会返回高速缓存中的查询计划。

如果查询计划不可用于高速缓存中，而您有权在客户端上查看计划，则会生成带有优化程序估计的查询计划 (**NOEXEC** 选项为 **ON** 的查询计划)，并在 **Interactive SQL** 客户端计划窗口中显示该查询计划。

当用户请求尚未执行的查询计划时，该查询计划不可用于高速缓存中。而是会返回带有优化程序估计但不带 **QUERY\_PLAN\_AFTER\_RUN** 统计信息的查询计划。

无法使用 **GRAPHICAL\_PLAN** 函数访问存储过程的查询计划。

用户可以查看针对 **SAP Sybase IQ** 查询打开游标的查询计划。使用 **DECLARE CURSOR** 和 **OPEN CURSOR** 来声明并打开游标。要获取最近打开的游标的查询计划，请使用：

```
SELECT GRAPHICAL_PLAN ( );
```

当 `QUERY_PLAN_AFTER_RUN` 选项为 `OFF` 时，计划会显示在 **OPEN CURSOR** 或 **CLOSE CURSOR** 之后。但如果 `QUERY_PLAN_AFTER_RUN` 为 `ON`，则必须先执行 **CLOSE CURSOR**，然后才能请求计划。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 支持。

#### 示例

以下示例将 **SELECT** 语句作为字符串参数传递，并返回计划以便执行查询。本示例将计划保存在文件 `gplan.xml` 中。

**注意：** 如果使用 **OUTPUT** 语句的 **HEXADECIMAL** 子句（设置为 **ASIS**）来获取已设置格式的 plan 输出，则字符值写入时不发生任何转义，即使值中包含控制字符。**ASIS** 对于包含格式设置字符（如制表符或回车符）的文本很有用。

```
SELECT GRAPHICAL_PLAN ('SELECT * FROM Employees');OUTPUT to 'C:\gplan.xml' HEXADECIMAL ASIS quote '';
```

以下示例会返回高速缓存中的查询计划（如果可用）：

```
SELECT GRAPHICAL_PLAN ( );
```

#### 另请参见

- `HTML_PLAN` 函数 [字符串]（第 209 页）

## GROUPING 函数 [集合]

标识 **ROLLUP** 或 **CUBE** 操作结果集中的某一列是否由于是小计行的一部分或由于基础数据而为空值。

#### 语法

```
GROUPING ( group-by-expression )
```

#### 参数

参数	描述
group-by-expression	表达式（作为分组列出现在使用带有 <b>ROLLUP</b> 或 <b>CUBE</b> 关键字的 <b>GROUP BY</b> 子句的查询结果集中）。该函数标识通过 <b>ROLLUP</b> 或 <b>CUBE</b> 操作添加到结果集中的小计行。

目前，SAP Sybase IQ 不支持 **PERCENTILE\_CONT** 或 **PERCENTILE\_DISC** 函数进行 **GROUP BY CUBE** 操作。

## 返回

值	描述
1	表示 group-by-expression 由于为小计行的一部分而为空值。该列不是该行的前缀列。
0	表示 group-by-expression 为小计行的前缀列。

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 支持。

## 另请参见

- 窗口化集合函数的用法 (第 78 页)

**GROUP\_MEMBER 函数 [系统]**

标识用户是否属于指定的组。

## 语法

```
GROUP_MEMBER ( group-name-string-expression [ , user-name-string-expression ] )
```

## 参数

参数	描述
<i>group-name-string-expression</i>	标识要考虑的组。
<i>user-name-string-expression</i>	标识要考虑的用户。如果未提供, 则假设是当前用户名。

## 返回值

表 35. 返回值

值	描述
0	如果符合以下条件中的任何一个条件, 则返回 0: 组不存在、用户不存在或者用户不属于指定组。
1	如果用户是指定组的成员, 则返回 0 以外的整数。

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

## HEXTOBIGINT 函数 [数据类型转换]

返回十六进制字符串的 BIGINT 等效项。

### 语法

```
HEXTOBIGINT ( hexadecimal-string )
```

### 参数

参数	描述
hexadecimal-string	要转换为大整数 (BIGINT) 的十六进制值。输入内容可采用以下形式，可以带有含小写或大写 "0x" 的前缀，也可以不带前缀： <i>0xhex-string</i> <i>0Xhex-string</i> <i>hex-string</i>

### 注释

**HEXTOBIGINT** 函数接受十六进制整数，并返回 BIGINT 等效项。十六进制整数可以以 CHAR 和 VARCHAR 值表达式以及 BINARY 和 VARBINARY 表达式的形式提供。

**HEXTOBIGINT** 函数接受有效的十六进制字符串，字符串带或不带 "0x" 或 "0X" 前缀，并用单引号引起来。

对于少于 16 位数的输入，假定将用零对其左侧进行填充。

如果输入时出现数据类型转换失败，会返回错误，但 `CONVERSION_ERROR` 选项设置为 OFF 时例外。当 `CONVERSION_ERROR` 为 OFF 时，无效的十六进制输入将返回 NULL。

请参见《参考：语句和选项》>“数据库选项”>“按字母顺序排列的选项列表”>“`CONVERSION_ERROR` 选项 [TSQL]”。

如果 BINARY 或 VARBINARY 值超过 8 个字节，CHAR 或 VARCHAR 值超过 16 个字符，会返回错误，但附加了 "0x" 的值除外。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下语句返回值 4294967287:

```
SELECT HEXTOBIGINT ( '0xffffffff7' ) FROM iq_dummy
```

```
SELECT HEXTOBIGINT ( '0Xxffffffff7' ) FROM iq_dummy
```

```
SELECT HEXTOBIGINT ( 'fffffff7' ) FROM iq_dummy
```

### 另请参见

- [BIGINTTOHEX 函数 \[数据类型转换\]](#) (第 139 页)
- [HEXTOINT 函数 \[数据类型转换\]](#) (第 204 页)
- [INTTOHEX 函数 \[数据类型转换\]](#) (第 218 页)

## HEXTOINT 函数 [数据类型转换]

返回十六进制字符串的不带符号的 `BIGINT` 等效项。

### 语法

```
HEXTOINT ( hexadecimal-string )
```

### 参数

参数	描述
<code>hexadecimal-string</code>	要转换成整数的字符串。输入内容可以采用以下形式，前缀中包括小写或大写“x”，或者不使用前缀：  <pre>0x<i>hex-string</i></pre> <pre>0X<i>hex-string</i></pre> <pre><i>hex-string</i></pre>

### 返回

`HEXTOINT` 函数返回十六进制字符串的等效平台无关 `SQL INTEGER`。如果从右侧数第 8 位数是数字 8-9 以及大写或小写字母 A-F 中的某一个，并且前面的前导位都是大写或小写字母 F，则该十六进制值代表负整数。以下语句中对 `HEXTOINT` 的使用无效，因为该参数表示的正整数值无法表示为有符号 32 位整数：

```
SELECT HEXTOINT( '0x0080000001' );
```

### INT

#### 注释

对于无效的十六进制输入，`SAP Sybase IQ` 会返回错误，但 `CONVERSION_ERROR` 选项为 `OFF` 时除外。当 `CONVERSION_ERROR` 为 `OFF` 时，无效的十六进制输入将返回 `NULL`。

请参见《参考：语句和选项》中的 `CONVERSION_ERROR` 选项 [TSQL]。

数据库选项 `ASE_FUNCTION_BEHAVIOR` 指定 `SAP Sybase IQ` 函数（包括 `INTTOHEX` 和 `HEXTOINT`）的输出与 `Adaptive Server` 函数的输出一致。



请参见《参考：语句和选项》中的 ASE\_FUNCTION\_BEHAVIOR 选项。

ASE\_FUNCTION\_BEHAVIOR 选项为 ON 时：

- SAP Sybase IQ **HEXTOINT** 假定输入是由 8 个字符组成的十六进制字符串；如果长度小于 8 个字符，则会用零填充字符串左侧。
- SAP Sybase IQ **HEXTOINT** 接受带前缀 0x 的最大字符长度为 16（共 18 个字符）；使用时应小心谨慎，因为较大的输入值产生的整数值可能会溢出带符号的 32 位整数输出的大小。
- 将 SAP Sybase IQ **HEXTOINT** 函数的输出数据类型假定为带符号的 32 位整数。
- SAP Sybase IQ **HEXTOINT** 接受 32 位十六进制整数作为带符号的表示形式。
- 对于 8 个以上的十六进制字符，SAP Sybase IQ **HEXTOINT** 只考虑相关字符。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

#### 示例

以下语句返回值 420：

```
SELECT HEXTOINT ( '0x1A4' ) FROM iq_dummy
```

```
SELECT HEXTOINT ( '0X1A4' ) FROM iq_dummy
```

```
SELECT HEXTOINT ( '1A4' ) FROM iq_dummy
```

#### 另请参见

- BIGINTTOHEX 函数 [数据类型转换]（第 139 页）
- HEXTOBIGINT 函数 [数据类型转换]（第 203 页）
- INTTOHEX 函数 [数据类型转换]（第 218 页）

## HOUR 函数 [日期和时间]

返回 0 到 23 之间的一个数字，该数字对应于指定日期/时间的小时部分。

#### 语法

```
HOUR ( datetime-expression )
```

表 36. 参数

参数	定义
datetime-expression	日期/时间。

#### 返回

SMALLINT

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句返回值 21:

```
SELECT HOUR( '1998-07-09 21:12:13' ) FROM iq_dummy
```

**HOURS 函数 [日期和时间]**

返回自任意开始日期和时间以来的小时数，返回两个指定时间之间的完整小时数，或者将指定的 *integer-expression* 数量的小时添加到某个时间中。

*语法*

```
HOURS ( datetime-expression
| datetime-expression, datetime-expression
| datetime-expression, integer-expression )
```

*参数*

**表 37. 参数**

参数	描述
<i>datetime-expression</i>	日期和时间。
<i>integer-expression</i>	要添加到 <i>datetime-expression</i> 中的小时数。如果 <i>integer-expression</i> 是负数，则从日期/时间中减去相应的小时数。如果提供整数表达式，则必须将 <i>datetime-expression</i> 显式转换为 <i>datetime</i> 数据类型。

*返回*

INT

*注释*

第二个语法返回从第一个日期/时间到第二个日期/时间之间的完整小时数。数值可能为负。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句返回值 17518758:

```
SELECT HOURS ( '1998-07-13 06:07:12' ) FROM iq_dummy
```

以下语句返回值 4，表示两个时间之间的差值：

```
SELECT HOURS ( '1999-07-13 06:07:12',
  '1999-07-13 10:07:12' ) FROM iq_dummy
```

以下语句返回日期时间值 1999-05-13 02:05:07.000：

```
SELECT HOURS ( CAST ( '1999-05-12 21:05:07'
AS DATETIME ), 5 ) FROM iq_dummy
```

### 另请参见

- CAST 函数 [数据类型转换] (第 143 页)
- CONVERT 函数 [数据类型转换] (第 153 页)
- MINUTES 函数 [日期和时间] (第 242 页)
- MONTHS 函数 [日期和时间] (第 246 页)
- REPLACE 函数 [字符串] (第 286 页)
- SECOND 函数 [日期和时间] (第 296 页)
- WEEKS 函数 [日期和时间] (第 336 页)
- YEAR 函数 [日期和时间] (第 341 页)
- YEARS 函数 [日期和时间] (第 342 页)

## HTML\_DECODE 函数 [Miscellaneous]

对出现在 HTML 文字字符串中的特殊字符实体进行解码。

### 语法

```
HTML_DECODE ( string )
```

### 参数

- **string** - HTML 文档所使用的任意文字字符串。

### 返回值

LONG VARCHAR 或 LONG NVARCHAR。

---

**注意：** 结果数据类型为 LONG VARCHAR。要在 SELECT INTO 语句中使用 HTML\_DECODE，您必须拥有“非结构化数据分析选项”许可或使用 CAST 并将 HTML\_DECODE 设置为正确数据类型和大小。

---

### 注释

此函数在进行相应替代后返回字符串参数。下表包含可接受字符实体的示例。

字符	替代
&quot;	"

字符	替代
&#39;	'
&amp;	&
&lt;	<
&gt;	>
&#x 十六进制数;	Unicode 代码点, 指定为十六进制数。例如, &#x27; 返回一个撇号。
&# 十进制数;	Unicode 代码点, 指定为十进制数。例如, &#8482; 返回商标符号。

指定 Unicode 代码点时, 如果可以将值转换为数据库字符集中的字符, 则将值转换为字符。否则返回未解释的值。

SAP Sybase IQ 支持 HTML 4.01 规范中指定的所有字符实体引用。请参见 <http://www.w3.org/TR/html4/> 和 <http://www.w3.org/TR/html4/sgml/entities.html#h-24.2>。

#### 标准和兼容性

- **SQL/2008** - 服务商扩充。

#### 示例

以下语句返回字符串 `<p>The piano was made for 'Steinway & Sons'.</p>`:

```
SELECT HTML_DECODE('&lt;p&gt;The piano was made ' ||
'by &lsquo;Steinway &amp; Sons&rsquo;.&lt;/p&gt;')
```

以下语句返回字符串 `<p>It cost &#8364;85.000,000. </p>`:

```
SELECT HTML_DECODE('&lt;p&gt;It cost &euro;85.000,000.&lt;/p&gt;')
```

## **HTML\_ENCODE 函数 [Miscellaneous]**

对要插入到 HTML 文档中的字符串中的特殊字符进行编码。

#### 语法

```
HTML_ENCODE( string )
```

#### 参数

- **string** - 要在 HTML 文档中使用的任意字符串。

#### 返回值

LONG VARCHAR 或 LONG NVARCHAR。

---

**注意：** 结果数据类型为 `LONG VARCHAR`。要在 `SELECT INTO` 语句中使用 `HTML_ENCODE`，您必须拥有“非结构化数据分析选项”许可或使用 `CAST` 并将 `HTML_DECODE` 设置为正确数据类型和大小。

---

#### 注释

此函数在进行以下一组替代后返回字符串参数：

字符	替代
"	&quot;
'	&#39;
&	&amp;
<	&lt;
>	&gt;
小于 0x20 的字符 <i>nn</i>	&#xnn;

此函数支持 `NCHAR` 输入和/或输出。

#### 标准和兼容性

- **SQL/2008** – 服务商扩充。

#### 示例

以下示例返回字符串 `'&lt;!DOCTYPE HTML PUBLIC &quot;-//W3C//DTD HTML 4.01//EN&quot;&gt;'`。

```
SELECT HTML_ENCODE('<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">')
```

## HTML\_PLAN 函数 [字符串]

以 `HTML` 格式字符串的形式返回查询计划。

#### 语法

```
HTML_PLAN ( string-expression )
```

---

**注意：** 需要考虑 `CIS` 函数补偿性能方面的一些事宜。

---

#### 参数

参数	描述
string-expression	生成计划所使用的 <code>SQL</code> 语句。该语句主要是 <code>SELECT</code> 语句，但也可以是 <code>UPDATE</code> 或 <code>DELETE</code> 语句。

如果未向 **HTML\_PLAN** 函数提供参数，则系统会从高速缓存将查询计划返回给您。如果高速缓存中没有查询计划，则系统将显示以下消息：

```
No plan available
```

**HTML\_PLAN** 函数的行为由数据库选项 `QUERY_PLAN_TEXT_ACCESS` 和 `QUERY_PLAN_TEXT_CACHING` 控制。如果 `QUERY_PLAN_TEXT_ACCESS` 为 **OFF** (缺省值)，则系统将显示以下消息：

```
Plan not available. The database option QUERY_PLAN_TEXT_ACCESS is OFF
```

如果 `QUERY_PLAN_TEXT_ACCESS` 为 **ON**，且该字符串表达式的查询计划可用于服务器所维护的高速缓存中，则会返回高速缓存中的查询计划。

借助 **SELECT**、**UPDATE**、**DELETE**、**INSERT SELECT** 和 **SELECT INTO**，**HTML\_PLAN** 函数可用于向 **Interactive SQL** 返回查询计划。

用户可以查看针对 **SAP Sybase IQ** 查询打开游标的查询计划。要获取最近打开的游标的查询计划，请使用：

```
SELECT HTML_PLAN ( );
```

当 `QUERY_PLAN_AFTER_RUN` 选项为 **OFF** 时，计划会显示在 **OPEN CURSOR** 或 **CLOSE CURSOR** 之后。但如果 `QUERY_PLAN_AFTER_RUN` 为 **ON**，则必须先执行 **CLOSE CURSOR**，然后才能请求计划。

当针对 **SQL Anywhere** 查询或 **OMNI/CIS** 分解查询请求 **HTML\_PLAN** 时，系统将返回以下消息：

```
No plan. HTML_PLAN function is not supported for this type of statement or database.
```

### 标准和兼容性

- **SQL - ISO/ANSI SQL** 语法的供应商扩展。
- **Sybase** - 不受 **Adaptive Server** 支持。

### 示例

以下示例将 **SELECT** 语句作为字符串参数传递，并返回 **HTML** 计划以便执行查询。本示例将计划保存在文件 `hplan.html` 中。

```
SELECT HTML_PLAN ('SELECT * FROM Employees');  
OUTPUT to 'C:\hplan.html' HEXADECIMAL ASIS QUOTE '';
```

**OUTPUT TO** 子句 **HEXADECIMAL ASIS** 对于含有格式设置字符（如制表符或回车符）的文本很有用。设置为 **ASIS** 时，值按原样写入而不发生任何转义，即使值中包含控制字符。

以下示例会返回高速缓存中的 **HTML** 查询计划（如果可用）。

```
SELECT HTML_PLAN ( );
```

另请参见

- GRAPHICAL\_PLAN 函数 [字符串] (第 199 页)

## HTTP\_DECODE 函数 [Web 服务]

解码 HTTP 编码的字符串。这也称为 URL 解码。

语法

```
HTTP_DECODE ( string )
```

参数

- **string** - 从 URL 或 URL 编码的请求主体中获取的任意字符串。

返回值

LONG VARCHAR 或 LONG NVARCHAR

注释

此函数在将所有 `%nn` 形式的字符串序列替换为编码为 `nn` 的字符后返回字符串参数，其中 `nn` 为一个十六进制值。此外，所有加号 (+) 都替换为空格。

标准和兼容性

- **SQL/2008** - 服务商扩充。

示例

以下语句返回字符串 `http://dcx.sybase.com`:

```
SELECT HTTP_DECODE ('http%3A%2F%2Fdcx.sybase.com')
```

## HTTP\_ENCODE 函数 [Web 服务]

对字符串进行编码以在 HTTP 中使用。这也称为 URL 编码。

语法

```
HTTP_ENCODE ( string )
```

参数

- **string** - 要进行编码以用于 HTTP 传输的任意字符串。

返回值

LONG VARCHAR 或 LONG NVARCHAR

注释

此函数在进行以下一组替代后返回字符串参数。此外，所有十六进制编码小于 20 或大于 7E 的字符都将替换为 `%nn`，其中 `nn` 为字符编码。

字符	替代
空格	%20
"	%22
#	%23
%	%25
&	%26
,	%2C
;	%3B
<	%3C
>	%3E
[	%5B
\	%5C
]	%5D
`	%60
{	%7B
	%7C
}	%7D
小于 0x20 且大于 0x7f 的字符编码 <i>nn</i> 。	% <i>nn</i>

此函数支持 NCHAR 输入和/或输出。

#### 标准和兼容性

- **SQL/2008** – 服务商扩充。

#### 示例

以下语句返回字符串 /opt%26id=123%26text='oid:c%09d%20ef':

```
SELECT HTTP_ENCODE('/opt&id=123&text=''oid:c\x09d ef''')
```

## **HTTP\_HEADER 函数 [Web 服务]**

返回 HTTP 请求标头的值。

#### 语法

```
HTTP_HEADER( header-field-name )
```



### 参数

- **header-field-name** - HTTP 请求标头字段的名称。

### 返回值

LONG VARCHAR。

---

**注意：** 结果数据类型为 LONG VARCHAR。要在 SELECT INTO 语句中使用 HTTP\_HEADER，您必须拥有“非结构化数据分析选项”许可或使用 CAST 并将 HTTP\_HEADER 设置为正确数据类型和大小。

---

### 注释

此函数返回指定的 HTTP 请求标头字段的值，如果该字段不存在或不是从 HTTP 服务进行调用，则返回 NULL。当通过 Web 服务处理 HTTP 请求时，将使用该函数。

以下是可能会在处理 HTTP Web 服务请求时值得关注的一些标头：

- **Cookie** - cookie 值（如果有），由客户端存储，并与请求的 URI 关联。
- **Referer** - 包含指向所请求 URI 链接的页面的 URL（例如 `http://documents.sample.com:80/index.html`）。
- **Host** - 所要请求资源的 Internet 主机名或 IP 地址和端口号，可从用户或相关资源提供的原始 URI 获得（例如 `webserver.sample.com:8082`）。
- **User-Agent** - 客户端应用程序的名称（例如 `Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0`）。
- **Accept-Encoding** - 客户端应用程序可接受的响应的编码列表（例如 `gzip, deflate`）。

有关这些标头的详细信息可在 <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html> 中找到。

可以通过以下特殊标头对客户端请求的请求行中的元素进行访问。

- **@HttpMethod** - 返回要处理的请求的类型。可接受的值包括 DELETE、HEAD、GET、PUT 或 POST。
- **@HttpURI** - 请求的完整 URI，即 HTTP 请求中指定的 URI（例如，`/myservice?&id=-123&version=109&lang=en`）。
- **@HttpVersion** - 请求的 HTTP 版本（例如，`HTTP/1.0` 或 `HTTP/1.1`）。
- **@HttpQueryString** - 返回所请求 URI 的查询部分（如果存在），例如 `id=-123&version=109&lang=en`。

### 标准和兼容性

- **SQL/2008** - 服务商扩充。

## 示例

在由 HTTP Web 服务所调用的存储过程内，使用以下语句会检索 Cookie 标头的值：

```
SET cookie_value = HTTP_HEADER( 'Cookie' );
```

在由 HTTP Web 服务所调用的存储过程内，使用以下语句会在数据库服务器消息窗口中显示 HTTP 请求标头的名称和值：

```
BEGIN
  declare header_name long varchar;
  declare header_value long varchar;
  set header_name = NULL;
header_loop:
  LOOP
    SET header_name = NEXT_HTTP_HEADER( header_name );
    IF header_name IS NULL THEN
      LEAVE header_loop
    END IF;
    SET header_value = HTTP_HEADER( header_name );
    MESSAGE 'HEADER: ', header_name, '=',
            header_value TO CONSOLE;
  END LOOP;
END;
```

## HTTP\_VARIABLE 函数 [Web 服务]

返回一个 HTTP 变量的值。

### 语法

```
HTTP_VARIABLE( var-name [ , instance [ , attribute ] ] )
```

### 参数

- **var-name** - HTTP 变量的名称。
- **instance** - 当有多个变量具有相同的名称时，则为字段实例的实例编号；当要获取第一个变量时，则为 NULL。对允许进行多个选择的 SELECT 列表很有用。
- **attribute** - 在有多个部分的请求中，属性可指定标头字段名称，该名称返回多部分名称的标头值。

如果未指定任何属性，返回值将为百分比解码数据，并且字符集将被转换为数据库字符集。在此模式下支持 UTF 百分比编码数据。

属性也可以是下列模式之一：

- '@BINARY' - 返回 x-www-form-urlencoded 二进制数据值。此模式表示返回值为百分比解码数据且不是转换的字符集。此模式不支持 UTF-8 百分比编码，因为百分比编码数据只能解码为其等效的字节表示。
- '@TRANSPORT' - 返回原始 HTTP 传送格式的值（保留百分比编码）。

*返回值*

LONG VARCHAR。

---

**注意：** 结果数据类型为 LONG VARCHAR。要在 SELECT INTO 语句中使用 HTTP\_VARIABLE，您必须拥有“非结构化数据分析选项”许可或使用 CAST 并将 HTTP\_VARIABLE 设置为正确数据类型和大小。

---

*注释*

此函数返回指定的 HTTP 变量的值。在 Web 服务内处理 HTTP 请求时，将使用该函数。

如果 *var-name* 不存在，则返回值为 NULL。

当 Web 服务请求为 POST 且变量数据作为多部分/形式数据进行发送时，HTTP 服务器会分别接收每个变量的 HTTP 标头。如果指定了 *attribute* 参数，HTTP\_VARIABLE 函数会从 POST 请求返回特定变量的关联多部分/形式数据标头值。对于表示文件的变量，Content-Disposition、Content-Type 和 @BINARY 的属性将分别返回文件名、媒体类型和文件内容。

通常，所有输入数据都会经历客户端（例如，浏览器）字符集与数据库字符集之间的字符集转换。不过，如果为 *attribute* 指定了 @BINARY，则返回变量值而不进行字符集转换或百分比解码。从客户端接收二进制数据（如图像数据）时，这可能会有用处。

如果指定的实例不存在或从 Web 服务之外调用函数，则此函数返回 NULL。

*标准和兼容性*

- SQL/2008 - 服务商扩充。

*示例*

在由 HTTP Web 服务所调用的存储过程内，使用以下语句会检索示例 URL 中所指示的 HTTP 变量的值：

```
-- http://sample.com/demo/ShowDetail?product_id=300&customer_id=101
BEGIN
  DECLARE v_customer_id LONG VARCHAR;
  DECLARE v_product_id LONG VARCHAR;
  SET v_customer_id = HTTP_VARIABLE( 'customer_id' );
  SET v_product_id = HTTP_VARIABLE( 'product_id' );
  CALL ShowSalesOrderDetail( v_customer_id, v_product_id );
END;
```

在由 HTTP Web 服务所调用的存储过程内，使用以下语句会请求图像变量的 Content-Disposition 和 Content-Type 标头：

```
SET v_name = HTTP_VARIABLE( 'image', NULL, 'Content-Disposition' );
SET v_type = HTTP_VARIABLE( 'image', NULL, 'Content-Type' );
```

在由 HTTP Web 服务所调用的存储过程内，使用以下语句会请求图像变量在其当前字符集中的值（即无需进行字符集转换）：

## SQL 函数

```
SET v_image = HTTP_VARIABLE( 'image', NULL, '@BINARY' );
```

### IFNULL 函数 [杂项]

返回第一个非空值表达式或空值。

如果第一个表达式为空值，则返回第二个表达式的值。如果第一个表达式不为空值，则返回第三个表达式的值。如果第一个表达式不为空值，并且没有第三个表达式，则返回空值。

#### 语法

```
IFNULL ( expression1, expression2 [ , expression3 ] )
```

#### 参数

表 38. 参数

参数	描述
expression1	要计算其值的表达式。它的值决定是返回 <i>expression2</i> 还是 <i>expression3</i> 。
expression2	当 <i>expression1</i> 为空值时的返回值。
expression3	当 <i>expression1</i> 不为空值时的返回值。

#### 返回

返回的数据类型取决于 *expression-2* 和 *expression-3* 的数据类型。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

#### 示例

下面的语句返回值 -66:

```
SELECT IFNULL( NULL, -66 ) FROM iq_dummy
```

下面的语句返回空值，这是因为第一个表达式不为空值，并且没有第三个表达式:

```
SELECT IFNULL( -66, -66 ) FROM iq_dummy
```

### INDEX\_COL 函数 [系统]

返回索引列的名称。

#### 语法

```
INDEX_COL ( table-name, index-id, key_# [ , user-id ] )
```

## 参数

参数	定义
table-name	表名。
index-id	<i>table-name</i> 的索引的索引 ID。
key_#	<i>index-id</i> 指定的索引中的关键字。此参数指定索引中的列号。如果是单列索引， <i>key_#</i> 等于 0。如果是多列索引，对于第一列， <i>key_#</i> 等于 0，对于第二列，它等于 1，依此类推。
user-id	<i>table-name</i> 的所有者的用户 ID。如果未指定 <i>user-id</i> ，则此值缺省为调用者的用户 ID。

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。

## 另请参见

- OBJECT\_ID 函数 [系统] (第 257 页)

**INSERTSTR 函数 [字符串]**

将一个字符串插入到另一个字符串中的指定位置。

## 语法

```
INSERTSTR ( numeric-expression, string-expression1, string-expression2 )
```

## 参数

参数	定义
numeric-expression	要在其后插入 <i>string-expression2</i> 的位置。使用零可将字符串插入到开头。
string-expression1	要在其中插入 <i>string-expression2</i> 的字符串。
string-expression2	要插入的字符串。

## 返回

LONG VARCHAR

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **INSERTSTR**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **INSERTSTR** 设置为正确的数据类型和大小。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 支持。STUFF 函数是等效的，该函数在 Adaptive Server 和 SAP Sybase IQ 中均受支持。

*示例*

以下语句返回值 "backoffice":

```
SELECT INSERTSTR( 0, 'office ', 'back' ) FROM iq_dummy
```

**INTTOHEX 函数 [数据类型转换]**

返回十进制整数的十六进制等效数字。

*语法*

```
INTTOHEX ( integer-expression )
```

*参数*

参数	描述
integer-expression	要转换成十六进制的整数。

*返回*

VARCHAR

*注释*

如果使用 INTTOHEX 转换进行输入数据转换时失败，则 SAP Sybase IQ 返回错误，但 CONVERSION\_ERROR 选项为 OFF 时除外。在这种情况下，结果为 NULL。

数据库选项 ASE\_FUNCTION\_BEHAVIOR 指定 SAP Sybase IQ 函数（包括 INTTOHEX 和 HEXTOINT）的输出与 Adaptive Server 函数的输出一致。ASE\_FUNCTION\_BEHAVIOR 的缺省值为 OFF。

当禁用 ASE\_FUNCTION\_BEHAVIOR 选项时（值为 OFF），以下内容成立：

- INTTOHEX 的输出与 SQL Anywhere 兼容。
- 根据输入，INTTOHEX 的输出可以是 8 位数或 16 位数，并用零进行左侧填充；返回数据类型为 VARCHAR。
- INTTOHEX 的输出没有“0x”或“0X”前缀。
- INTTOHEX 的输入可以是一个最多为 64 位的整数。

当启用 ASE\_FUNCTION\_BEHAVIOR 选项时（值为 ON），以下内容成立：

- INTTOHEX 的输出与 Adaptive Server 兼容。

- **INTTOHEX** 的输出始终为 8 位数，并用零进行左侧填充；返回数据类型为 VARCHAR。
- **INTTOHEX** 的输出没有 “0x” 或 “0X” 前缀。
- **SAP Sybase IQ INTTOHEX** 假定输入是带符号的 32 位整数；更大的值可能会溢出，并且可能导致转换错误。例如，语句：

```
SELECT INTTOHEX( 1000000000 ) FROM iq_dummy
```

返回值 3B9ACA00。但语句：

```
SELECT INTTOHEX( 10000000000 ) FROM iq_dummy
```

则导致转换错误。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下语句返回值 3B9ACA00：

```
SELECT INTTOHEX( 1000000000 ) FROM iq_dummy
```

以下语句返回值 00000002540BE400：

```
SELECT INTTOHEX ( 10000000000) FROM iq_dummy
```

### 另请参见

- **BIGINTTOHEX** 函数 [数据类型转换] (第 139 页)
- **HEXTOBIGINT** 函数 [数据类型转换] (第 203 页)
- **HEXTOINT** 函数 [数据类型转换] (第 204 页)

## **ISDATE** 函数 [日期和时间]

测试字符串参数是否可以转换为日期。

如果转换能够进行，该函数返回 1；否则返回 0。如果参数为空，则返回 0。

### 语法

```
ISDATE ( string )
```

### 参数

表 39. 参数

参数	描述
string	要分析的字符串（分析目的是确定该字符串是否代表有效日期）。

返回

INT

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase:
  - SQL Anywhere 使用 ISO 8601 日期交换格式。
  - 受 Adaptive Server 15.0.1 及更高版本支持。

示例

以下示例测试 birth\_date 列中是否保存着有效日期，它以 NULL 返回无效日期，并以 date 格式返回有效日期。

```
select birth_date from MyData;
-----
1990/32/89
0101/32/89
1990/12/09

select
  case when isdate(birth_date)=0 then NULL
  else cast(birth_date as date)
  end
  from MyData;
-----
(NULL)
(NULL)
1990-12-09
```

### ISNULL 函数 [杂项]

返回参数列表中的第一个非空表达式的值。

必须向此函数至少传递两个表达式。

语法

```
ISNULL ( expression, expression [ ..., expression ] )
```

参数

表 40. 参数

参数	描述
expression	要测试是否为空值的表达式。

返回

此函数的返回类型取决于指定的表达式。即，数据库服务器计算此函数时，首先搜索可用于比较所有表达式的数据类型。如果找到通用类型，则数据库服务器比较表达



式，然后以用于比较的类型返回结果。如果数据库服务器找不到通用比较类型，则返回错误。

#### 注释

**ISNULL** 函数与 **COALESCE** 函数相同。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

#### 示例

以下语句返回值 -66:

```
SELECT ISNULL( NULL , -66, 55, 45, NULL, 16 ) FROM iq_dummy
```

#### 另请参见

- **COALESCE** 函数 [杂项] (第 149 页)

## **ISNUMERIC** 函数 [杂项]

测试字符串参数是否可以转换为数字。

如果转换能够进行，该函数返回 1；否则返回 0。如果参数为空，则返回 0。

#### 语法

```
ISNUMERIC ( string )
```

#### 参数

参数	描述
string	要分析的字符串（分析目的是确定该字符串是否代表有效的数值）。

#### 返回

INT

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 受 Adaptive Server 15.0.1 及更高版本支持。

#### 示例

以下示例测试 `height_in_cms` 列中是否保存着有效的数值数据，它以 `NULL` 返回无效的数值数据，并以 `int` 格式返回有效的数值数据。

```
data height_in_cms
-----
```

## SQL 函数

```
asde
asde
180
156

select case
  when isnumeric(height_in_cms)=0
  then NULL
  else cast(height_in_cms as int)
end
from MyData
```

## LAG 函数 [分析]

返回表或表分区的上一行中的属性值的行间函数。

### 语法

```
LAG (value_expr) [, offset [, default]] OVER ((PARTITION BY window partition) ORDER BY window ordering)
```

### 参数

参数	描述
<i>value_expr</i>	定义要从表返回的偏移量数据的表列或表达式。
<i>offset</i>	当前行上方的行数，表示为一个准确的非负数字文本，或表示为一个包含准确数值数据的 SQL 变量。允许的范围为 0 到 231。
<i>default</i>	<i>offset</i> 值超过表或分区的基数范围时要返回的值。
<i>window partition</i>	(可选) 一个或多个用逗号分隔的值表达式，表示您希望如何分隔一组结果行。
<i>window ordering</i>	如果指定窗口分区，则定义用于对窗口分区中的行进行排序的表达式；如果未指定窗口分区，则定义用于对结果集中的行进行排序的表达式。

### 注释

**LAG** 函数需要 **OVER (ORDER BY)** 窗口规范。**OVER (ORDER BY)** 子句中的窗口分区子句是可选的。**OVER (ORDER BY)** 子句不能包含窗口构架 **ROWS/RANGE** 规范。

不能在 *value\_expr* 中定义分析表达式。换言之，不能嵌套使用分析函数，但可以将其其它内置函数表达式用作 *value\_expr*。

必须为 *offset* 输入非负数值数据类型。输入 0 将返回当前行。输入负数将产生错误。

*default* 的缺省值为 **NULL**。*default* 的数据类型必须可以隐式转换为 *value\_expr* 值的数据类型，否则 SAP Sybase IQ 将生成转换错误。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

### 示例

以下示例返回雇员表中的薪水数据，按部门 ID 对数据进行分区，并根据雇用开始日期对数据进行排序。**LAG** 函数返回上一行中的薪水值（一行的物理偏移量），并在 **LAG (Salary)** 列的下方显示该值：

```
SELECT DepartmentID dID, StartDate, Salary, LAG(Salary, 1)
OVER(PARTITION BY dID ORDER BY StartDate) FROM Employees ORDER BY
1,2;
```

返回的结果集为：

dID	StartDate	Salary	Lag(Salary)
100	1984-08-28	45,700.000	NULL
100	1985-01-01	62,000.000	45,700.000
100	1985-06-17	57,490.000	62,000.000
100	1986-06-07	72,995.000	57,490.000
100	1986-07-01	48,023.690	72,995.000
...			
200	1985-02-03	38,500.000	NULL
200	1985-12-06	54,800.000	38,500.000
200	1987-02-19	39,300.000	54,800.000
200	1987-07-10	49,500.000	39,300.000
200	1988-10-04	54,600.000	49,500.000
200	1988-11-12	39,800.000	54,600.000
...			

### 另请参见

- LEAD 函数 [分析]（第 226 页）

## LAST\_VALUE 函数 [集合]

返回一组值中的最后一个值。

### 语法

```
LAST_VALUE (expression [IGNORE NULLS | RESPECT NULLS])
```

```
OVER (window-spec)
```

### 参数

参数	定义
<code>expression</code>	用于确定有序集合中最后一个值的表达式。

### 返回

参数的数据类型。

*注释*

**LAST\_VALUE** 返回一组值中的最后一个值，该组通常是有序集合。如果这组值中的最后一个值是空值，则除非指定 **IGNORE NULLS**，否则函数将返回空值。如果您指定 **IGNORE NULLS**，则 **LAST\_VALUE** 返回该集合中的最后一个非空值，或者，如果所有值均为空值，则返回空值。

返回值的数据类型与输入值相同。

不能对 **expression** 使用 **LAST\_VALUE** 或任何其它分析函数。换言之，不能嵌套使用分析函数，但可以将其它内置函数表达式用作 **expression**。

如果 **window-spec** 不包含 **ORDER BY** 表达式，或者 **ORDER BY** 表达式不够精确，无法保证唯一顺序，则结果将为任意值。如果没有 **window-spec**，结果将为任意值。

您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

---

**注意：** **DISTINCT** 不受支持。

---

*标准和兼容性*

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

*示例*

以下示例返回每位雇员的薪水，以及该部门中薪水最高的雇员的姓名：

```
SELECT GivenName + ' ' + Surname AS employee_name,
       Salary, DepartmentID,
       LAST_VALUE( employee_name ) OVER Salary_Window AS
highest_paid
FROM Employees
WINDOW Salary_Window AS ( PARTITION BY DepartmentID ORDER BY Salary
RANGE BETWEEN UNBOUNDED PRECEDING
AND UNBOUNDED FOLLOWING )
ORDER BY DepartmentID DESC;
```

返回的结果集为：

**表 41. LAST\_VALUE 结果集**

<b>employee_name</b>	<b>Salary</b>	<b>DepartmentID</b>	<b>highest_paid</b>
Michael Lynch	24,903.000	500	Jose Martinez
Joseph Barker	27,290.000	500	Jose Martinez
Sheila Romero	27,500.000	500	Jose Martinez
Felicia Kuo	28,200.000	500	Jose Martinez

employee_name	Salary	DepartmentID	highest_paid
Jeannette Bertrand	29,800.000	500	Jose Martinez
Jane Braun	34,300.000	500	Jose Martinez
Anthony Rebeiro	34,576.000	500	Jose Martinez
Charles Crowley	41,700.000	500	Jose Martinez
Jose Martinez	55,500.800	500	Jose Martinez
Doug Charlton	28,300.000	400	Scott Evans
Elizabeth Lambert	29,384.000	400	Scott Evans
Joyce Butterfield	34,011.000	400	Scott Evans
Robert Nielsen	34,889.000	400	Scott Evans
Alex Ahmed	34,992.000	400	Scott Evans
Ruth Wetherby	35,745.000	400	Scott Evans
...	...	...	...

另请参见

- 窗口化集合函数的用法 (第 78 页)

## **LCASE 函数 [字符串]**

将字符串中的所有字符转换成小写形式。

语法

**LCASE** ( *string-expression* )

参数

参数	描述
string-expression	要转换成小写形式的字符串。

返回

CHAR

NCHAR

LONG VARCHAR

VARCHAR

NVARCHAR

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **LCASE**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **LCASE** 设置为正确的数据类型和大小。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - Adaptive Server 不支持 **LCASE**；您可以使用 **LOWER** 来实现相同的功能。

### 示例

以下语句返回值 "lower case":

```
SELECT LCASE( 'LOWER CasE' ) FROM iq_dummy
```

### 另请参见

- LEFT 函数 [字符串] (第 228 页)
- LOWER 函数 [字符串] (第 237 页)
- REPLACE 函数 [字符串] (第 286 页)
- REVERSE 函数 [字符串] (第 289 页)
- RIGHT 函数 [字符串] (第 290 页)
- UCASE 函数 [字符串] (第 327 页)
- UPPER 函数 [字符串] (第 328 页)

## LEAD 函数 [分析]

返回表或表分区中的后续行中的属性值的行间函数。

### 语法

```
LEAD (value_expr) [, offset [, default]] OVER ([PARTITION BY window partition] ORDER BY window ordering)
```

### 参数

参数	描述
value_expr	定义要从表返回的偏移量数据的表列或表达式。
offset	当前行下方的行数，表示为一个准确的非负数字文本，或表示为一个包含准确数值数据的 SQL 变量。允许的范围为 0 到 231。
default	offset 值超过表或分区的范围时要返回的值。
window partition	(可选) 一个或多个用逗号分隔的值表达式，表示您希望如何分隔一组结果行。

参数	描述
window ordering	如果指定窗口分区，则定义用于对窗口分区中的行进行排序的表达式；如果未指定窗口分区，则定义用于对结果集中的行进行排序的表达式。

### 注释

**LEAD** 函数需要 **OVER (ORDER\_BY)** 窗口规范。**OVER (ORDER\_BY)** 子句中的窗口分区子句是可选的。**OVER (ORDER\_BY)** 子句不能包含窗口构架 **ROWS/RANGE** 规范。

不能在 *value\_expr* 中定义分析表达式。换言之，不能嵌套使用分析函数，但可以将其其它内置函数表达式用作 *value\_expr*。

必须为 *offset* 输入非负数值数据类型。输入 **0** 将返回当前行。输入负数将产生错误。

*default* 的缺省值为 **NULL**。*default* 的数据类型必须可以隐式转换为 *value\_expr* 值的数据类型，否则 SAP Sybase IQ 将生成转换错误。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

### 示例

以下示例返回雇员表中的薪水数据，按部门 **ID** 对数据进行分区，并根据雇用开始日期对数据进行排序。**LEAD** 函数返回下一行中的薪水值（一行的物理偏移量），并在 **LEAD (Salary)** 列的下方显示该值：

```
SELECT DepartmentID dID, StartDate, Salary, LEAD(Salary, 1)
OVER(PARTITION BY dID ORDER BY StartDate) FROM Employees ORDER BY
1,2;
```

返回的结果集为：

dID	StartDate	Salary	Lead(Salary)
100	1984-08-28	45,700.000	62,000.000
100	1985-01-01	62,000.000	57,490.000
100	1985-06-17	57,490.000	72,995.000
100	1986-06-07	72,995.000	48,023.690
...			
100	1990-08-19	54,900.000	NULL
200	1985-02-03	38,500.000	39,300.000
200	1987-02-19	39,300.000	49,500.000
200	1987-07-10	49,500.000	54,600.000
200	1988-11-28	46,200.000	34,892.000
200	1989-06-01	34,892.000	87,500.000
...			
200	1993-08-12	47,653.000	NULL

### 另请参见

- **LAG** 函数 [分析]（第 222 页）

**LEFT 函数 [字符串]**

从字符串开头返回指定数量的字符。

*语法*

```
LEFT ( string-expression, numeric-expression )
```

*参数*

表 42. 参数

参数	描述
string-expression	字符串。
numeric-expression	要返回的字符数。

*返回*

LONG VARCHAR

LONG NVARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **LEFT**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **LEFT** 设置为正确的数据类型和大小。

---

*注释*

如果字符串包含多字节字符，并且使用了适当的归类，则返回的字节数可能大于指定的字符数。

---

**注意：** **LEFT** 函数的结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **LEFT**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **LEFT** 设置为正确的数据类型和大小。

---

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句返回值 "choco":

```
SELECT LEFT( 'chocolate', 5 ) FROM iq_dummy
```

**另请参见**

- LCASE 函数 [字符串] (第 225 页)
- LOWER 函数 [字符串] (第 237 页)



- REPLACE 函数 [字符串] (第 286 页)
- REVERSE 函数 [字符串] (第 289 页)
- RIGHT 函数 [字符串] (第 290 页)
- UCASE 函数 [字符串] (第 327 页)
- UPPER 函数 [字符串] (第 328 页)

## LEN 函数 [字符串]

采用一个参数作为 BINARY 或 STRING 类型的输入，并按照数据库归类序列中的定义返回指定字符串表达式的字符数（不包括尾随空白）。

结果可能不同于多字节字符集的字符串字节长度。

此外，也允许使用 BINARY 和 VARBINARY，在这种情况下，LEN() 返回输入的字节数。

LEN 是 LENGTH 函数的别名。

语法

```
LEN ( string_expr )
```

参数

表 43. 参数

参数	描述
string_expr	要计算其值的字符串表达式。

注释

此函数等效于 CHAR\_LENGTH ( *string\_expression* )。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。

示例

以下示例返回值 3152:

```
select len(Photo) from Productswhere ID = 500
```

另请参见

- BIT\_LENGTH 函数 [字符串] (第 140 页)
- BYTE\_LENGTH 函数 [字符串] (第 141 页)
- CHAR\_LENGTH 函数 [字符串] (第 147 页)
- COL\_LENGTH 函数 [系统] (第 150 页)
- DATALENGTH 函数 [系统] (第 164 页)

## SQL 函数

- **LENGTH** 函数 [字符串] (第 230 页)
- **OBJECT\_NAME** 函数 [系统] (第 257 页)
- **OCTET\_LENGTH** 函数 [字符串] (第 258 页)
- **STR\_REPLACE** 函数 [字符串] (第 315 页)

### **LENGTH** 函数 [字符串]

返回指定字符串中的字符数。

#### 语法

```
LENGTH ( string-expression )
```

#### 参数

表 44. 参数

参数	描述
string-expression	字符串。

#### 返回

INT

#### 注释

如果字符串包含多字节字符，并且使用了适当的归类，则 **LENGTH** 返回字符数而不是字节数。如果字符串为 **BINARY** 数据类型，则 **LENGTH** 函数与 **BYTE\_LENGTH** 行为相同。

**LENGTH** 函数与 **CHAR\_LENGTH** 函数相同。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。请改用 **CHAR\_LENGTH** 函数。

#### 示例

以下语句返回值 9:

```
SELECT LENGTH( 'chocolate' ) FROM iq_dummy
```

#### 另请参见

- **BIT\_LENGTH** 函数 [字符串] (第 140 页)
- **BYTE\_LENGTH** 函数 [字符串] (第 141 页)
- **CHAR\_LENGTH** 函数 [字符串] (第 147 页)
- **COL\_LENGTH** 函数 [系统] (第 150 页)
- **DATALength** 函数 [系统] (第 164 页)

- LEN 函数 [字符串] (第 229 页)
- OBJECT\_NAME 函数 [系统] (第 257 页)
- OCTET\_LENGTH 函数 [字符串] (第 258 页)
- STR\_REPLACE 函数 [字符串] (第 315 页)

## LIST 函数 [集合]

返回组中每行值的分隔列表。

### 语法

```
LIST(
[ALL | DISTINCT] string-expression
[, 'delimiter-string']
[ORDER BY order-by-expression [ ASC | DESC ], ... ] )
```

---

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

---

### 参数

- **string-expression** - 字符串表达式，通常是列名。如果指定了 ALL (缺省设置)，对于组中的每一行，string-expression 的值会添加到结果字符串，并用 delimiter-string 分隔这些值。如果指定了 DISTINCT，则只添加唯一的 string-expression 值。
- **delimiter-string** - 列表项的分隔符字符串。缺省设置是逗号。如果设为 NULL 值或空字符串，则没有分隔符。delimiter-string 必须是常量。
- **order-by-expression** - 对该函数返回的项进行排序。此参数之前没有逗号，以便在未提供 delimiter-string 的情况下使用。

*order-by-expression* 不能是整数值。但可以是包含整数值的变量。

当 ORDER BY 子句包含常量时，由优化程序对常量进行解释，然后替换为等效的 ORDER BY 子句。例如，优化程序将 ORDER BY 'a' 解释为 ORDER BY 表达式。

对含多个集合函数的查询块，如果几个有效的 ORDER BY 子句可以在逻辑上组合成一个 ORDER BY 子句，则可执行该 ORDER BY 子句。例如，以下子句：

```
ORDER BY expression1, 'a', expression2
```

```
ORDER BY expression1, 'b', expression2, 'c', expression3
```

可归入下个子句：

```
ORDER BY expression1, expression2, expression3
```

### 返回

LONG VARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 SELECT INTO 语句中使用 LIST，您必须具有“非结构化数据分析选项”许可证，或使用 CAST 并将 LIST 设置为正确的数据类型和大小。

---

### 注释

**LIST** 函数返回一串组中每一行 **X** 的所有非 **NULL** 值（带分隔符）。如果组中没有任何一行具有明确的 **X** 值，则 **LIST(X)** 返回空字符串。

**LIST** 函数会忽略 **NULL** 值和空字符串。

**LIST** 函数不能用作窗口函数，但可用作窗口函数的输入。

此函数支持 **NCHAR** 输入和/或输出。

### 标准和兼容性

- **SQL/2008** – 服务商扩充。

**SAP Sybase IQ** 支持 **SQL/2008** 语言功能 **F441** “扩展集函数支持”，它允许集合函数的操作数是列引用以外的任意表达式。

**SAP Sybase IQ** 不支持可选的 **SQL/2008** 功能 **F442** “集函数中的混合列引用”。**SAP Sybase IQ** 不允许集合函数参数既含有来自包含 **LIST** 函数的查询块的列引用，又含有外部引用。

### 示例

以下语句返回值 **487 Kennedy Court, 547 School Street**。

```
SELECT LIST( Street ) FROM Employees  
WHERE GivenName = 'Thomas';
```

以下语句用于列出雇员 **ID**。结果集中的每一行都包含某一部门雇员 **ID** 的逗号分隔列表。

```
SELECT LIST( EmployeeID )  
FROM Employees  
GROUP BY DepartmentID;
```

<b>LIST( EmployeeID )</b>
102,105,160,243,247,249,266,278,...
129,195,299,467,641,667,690,856,...
148,390,586,757,879,1293,1336,...
184,207,318,409,591,888,992,1062,...
191,703,750,868,921,1013,1570,...

以下语句按雇员的姓氏对雇员 **ID** 排序：

```
SELECT LIST( EmployeeID ORDER BY Surname ) AS "Sorted IDs"  
FROM Employees  
GROUP BY DepartmentID;
```

Sorted IDs
1013,191,750,921,868,1658,...
1751,591,1062,1191,992,888,318,...
1336,879,586,390,757,148,1483,...
1039,129,1142,195,667,1162,902,...
160,105,1250,247,266,249,445,...

以下语句返回以分号分隔的列表。请注意 **ORDER BY** 子句的位置和列表分隔符：

```
SELECT LIST( EmployeeID, ';' ORDER BY Surname ) AS "Sorted IDs"
FROM Employees
GROUP BY DepartmentID;
```

Sorted IDs
1013;191;750;921;868;1658;703;...
1751;591;1062;1191;992;888;318;...
1336;879;586;390;757;148;1483;...
1039;129;1142;195;667;1162;902; ...
160;105;1250;247;266;249;445;...

务必要将以下语句与前一个语句加以区分，以下语句返回以逗号分隔并按复合排序关键字 ( **Surname, ;** ) 排序的雇员 ID 列表：

```
SELECT LIST( EmployeeID ORDER BY Surname, ';' ) AS "Sorted IDs"
FROM Employees
GROUP BY DepartmentID;
```

## LN 函数 [数值]

返回指定表达式的自然对数。

语法

```
LN ( numeric-expression )
```

参数

参数	描述
numeric-expression	数据类型为精确数值、近似数值、货币或可隐式转换为这些类型的任何类型的列、变量或表达式。对于其它数据类型， <b>LN</b> 函数会产生错误。返回值的类型为 <b>DOUBLE</b> 。

*注释*

**LN** 采用一个参数。例如，**LN (20)** 返回 2.995732。

**LN** 函数是 **LOG** 函数的别名。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。请改用 LOG 函数。

*另请参见*

- LOG 函数 [数值] (第 235 页)
- LOG10 函数 [数值] (第 236 页)

**LOCATE 函数 [字符串]**

返回一个字符串在另一个字符串中的位置。

*语法*

```
LOCATE ( string-expression1, string-expression2
[ , numeric-expression ] )
```

*参数*

参数	描述
<i>string-expression1</i>	要搜索的字符串。
<i>string-expression2</i>	要搜索的字符串。该字符串限制为 255 个字节。
<i>numeric-expression</i>	字符串中作为搜索起始位置的字符位置。第一个字符为位置 1。如果起始偏移量是负数， <b>LOCATE</b> 返回上一匹配字符串偏移量，而不是第一个字符。负数偏移量表示要从搜索中排除的字符串末尾的字符数。排除的字节数计算公式为 $(-1 * \text{偏移}) - 1$ 。

对于 CHAR、VARCHAR 和 BINARY 列，*numeric-expression* 为带符号的 32 位整数。

*返回*

INT

*注释*

如果指定了 *numeric-expression*，则在搜索字符串中的该偏移位置开始搜索。

如果未指定 *numeric-expression*，**LOCATE** 将仅返回指定字符串第一个实例的位置。

第一个字符串可以是长字符串（长于 255 个字节），但第二个字符串的长度不能超过 255 个字节。第二个字符串长度超过 255 个字节时会导致错误。

如果任一参数为 **NULL**，结果则为 **NULL**。

搜索零长度的字符串将返回 1。

如果字符串不包含指定字符串，**LOCATE** 函数将返回零 (0)。

在 **LOCATE** 函数中返回或指定的所有位置或偏移量始终为字符偏移量，并且可能不同于多字节数据的字节偏移量。

如果您有权使用非结构化数据分析功能，则可以将此函数与大对象数据一起使用。

请参见《非结构化数据分析》中的 **LOCATE** 函数。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 支持。

### 示例

以下语句返回值 8:

```
SELECT LOCATE( 'office party this week - rsvp as soon as possible',
'party', 2 ) FROM iq_dummy
```

在第二个示例中，搜索的 *numeric-expression* 起始偏移为负数。

```
CREATE TABLE t1(name VARCHAR(20), dirname VARCHAR(60));
INSERT INTO t1 VALUES( 'm1000', ' c:\test\functions
\locate.sql' );
INSERT INTO t1 VALUES( 'm1001', ' d:\test\functions\trim.sql' );
COMMIT;

SELECT LOCATE(dirname, '\', -1), dirname FROM t1;
```

结果为:

```
18 c:\test\functions\locate.sql
18 d:\test\functions\trim.sql
```

### 另请参见

- PATINDEX 函数 [字符串]（第 259 页）
- LIKE 条件（第 50 页）

## **LOG** 函数 [数值]

返回一个数字的自然对数。

**LN** 是 **LOG** 的别名。

语法

```
LOG ( numeric-expression )
```

参数

表 45. 参数

参数	描述
numeric-expression	数字。

返回

此函数将其参数转换为 **DOUBLE**，以双精度浮点执行计算，然后返回 **DOUBLE** 值作为结果。如果参数为 **NULL**，则结果为 **NULL**。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

示例

以下语句返回值 3.912023:

```
SELECT LOG( 50 ) FROM iq_dummy
```

另请参见

- LN 函数 [数值] (第 233 页)
- LOG10 函数 [数值] (第 236 页)

### **LOG10 函数 [数值]**

返回一个数字的以 10 为底的对数。

语法

```
LOG10 ( numeric-expression )
```

参数

表 46. 参数

参数	描述
numeric-expression	数字。



### 返回

此函数将其参数转换为 **DOUBLE**，并以双精度浮点执行计算。如果参数为 **NULL**，则结果为 **NULL**。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 示例

以下语句返回值 1.698970。

```
SELECT LOG10( 50 ) FROM iq_dummy
```

### 另请参见

- LN 函数 [数值] (第 233 页)
- LOG 函数 [数值] (第 235 页)

## **LOWER** 函数 [字符串]

将字符串中的所有字符转换成小写形式。

### 语法

```
LOWER ( string-expression )
```

### 参数

表 47. 参数

参数	描述
string-expression	要转换的字符串。

### 返回

CHAR

NCHAR

LONG VARCHAR

VARCHAR

NVARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **LOWER**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **LOWER** 设置为正确的数据类型和大小。

---

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 示例

以下语句返回值 "lower case":

```
SELECT LOWER( 'LOWER Case' ) FROM iq_dummy
```

### 另请参见

- LCASE 函数 [字符串] (第 225 页)
- LEFT 函数 [字符串] (第 228 页)
- REPLACE 函数 [字符串] (第 286 页)
- REVERSE 函数 [字符串] (第 289 页)
- RIGHT 函数 [字符串] (第 290 页)
- UCASE 函数 [字符串] (第 327 页)
- UPPER 函数 [字符串] (第 328 页)

## LTRIM 函数 [字符串]

删除字符串中的前导空白。

### 语法

```
LTRIM ( string-expression )
```

### 参数

参数	描述
string-expression	要剪裁的字符串。

### 返回

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

---

**注意:** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **LTRIM**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **LTRIM** 设置为正确的数据类型和大小。

---

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 示例

以下语句返回值 "Test Message"，其中的所有前导空白都已删除：

```
SELECT LTRIM( '      Test Message' ) FROM iq_dummy
```

### 另请参见

- RTRIM 函数 [字符串] (第 295 页)

## MAX 函数 [集合]

返回在每一组行中找到的最大 *expression* 值。

### 语法

```
MAX ( expression
| DISTINCT column-name )
```

### 参数

参数	描述
<i>expression</i>	要计算其最大值的表达式。这通常是列名称。
DISTINCT <i>column-name</i>	返回的值与 MAX ( <i>expression</i> ) 相同，将其列出是为了保持完整性。

### 返回

与参数相同的数据类型。

### 注释

忽略 *expression* 为空值的行。对于不包含任何行的组返回空值。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 示例

以下语句返回值 138948.000，表示 Employees 表中的最高薪水：

```
SELECT MAX ( Salary )
FROM Employees
```

## 另请参见

- 窗口化集合函数的用法 (第 78 页)
- MIN 函数 [集合] (第 241 页)

**MEDIAN 函数 [集合]**

返回表达式的中值。

## 语法 1

```
MEDIAN ([ALL | DISTINCT] expression)
```

## 语法 2

```
MEDIAN ([ALL | DISTINCT] expression)
```

```
OVER (window-spec)
```

## 参数

参数	描述
expression	要为其计算中值的数值表达式。

## 注释

中值是样本、总体或概率分布中分隔较高一半与较低一半的数字。

返回值的数据类型与输入值相同。计算中值时会忽略空值。可以在使用集合函数之前使用可选关键字 **DISTINCT** 消除重复值。缺省设置为 **ALL**，即对所有行执行运算。

---

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 **ROLLUP** 和 **CUBE**。

---

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

---

**注意：** *window-spec* 不能包含 **ROW**、**RANGE** 或 **ORDER BY** 规范；*window-spec* 只能指定 **PARTITION** 子句。如果使用 **WINDOW** 子句，则不支持 **DISTINCT**。

---

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

## 示例

以下查询返回佛罗里达各部门的薪水中间值：

```
SELECT DepartmentID, Surname, Salary,
MEDIAN(Salary) OVER (PARTITION BY DepartmentID) "Median"
FROM Employees
WHERE State IN ('FL')
```

返回的结果为：

表 48. MEDIAN 结果集

DepartmentID	Surname	Salary	Median
100	Lull	87,900.000	73,870.000
100	Gowda	59,840.000	73,870.000
200	Sterling	64,900.000	76,200.000
200	Kelly	87,500.000	76,200.000
300	Litton	58,930.000	58,930.000
400	Francis	53,870.000	38,70.000
400	Charlton	28,300.000	53,870.000
400	Evans	68,940.000	53,870.000

另请参见

- 窗口化集合函数的用法 (第 78 页)

## MIN 函数 [集合]

返回在每一组行中找到的最小表达式值。

语法

```
MIN ( expression
| DISTINCT column-name )
```

参数

参数	描述
expression	要计算其最小值的表达式。这通常是列名称。
DISTINCT column-name	返回的值与 MIN (expression) 相同，将其列出是为了保持完整性。

返回

与参数相同的数据类型。

注释

忽略 *expression* 为空值的行。对于不包含任何行的组返回空值。

标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

以下语句返回值 24903.000，表示 Employees 表中的最低薪水：

```
SELECT MIN ( Salary )  
FROM Employees
```

**另请参见**

- 窗口化集合函数的用法 (第 78 页)
- MAX 函数 [集合] (第 239 页)

## **MINUTE 函数 [日期和时间]**

返回 0 到 59 之间的一个数字，该数字对应于指定日期/时间值的分钟部分。

*语法*

```
MINUTE ( datetime-expression )
```

*参数*

参数	描述
datetime-expression	日期/时间值。

*返回*

SMALLINT

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

以下语句返回值 22：

```
SELECT MINUTE ( '1998-07-13 12:22:34' ) FROM iq_dummy
```

## **MINUTES 函数 [日期和时间]**

返回自任意开始日期和时间以来的分钟数，返回两个指定时间之间的完整分钟数，或者将指定的 integer-expression 数量的分钟添加到某个时间中。

*语法*

```
MINUTES ( datetime-expression  
| datetime-expression, datetime-expression  
| datetime-expression, integer-expression )
```

## 参数

参数	描述
datetime-expression	日期和时间。
integer-expression	要添加到 <i>datetime-expression</i> 中的分钟数。如果 <i>integer-expression</i> 是负数，则从日期/时间中减去相应的分钟数。如果提供整数表达式，则必须将 <i>datetime-expression</i> 显式转换为 datetime 数据类型

## 返回

INT

TIMESTAMP

## 注释

第二个语法返回从第一个日期/时间到第二个日期/时间之间的完整分钟数。数值可能为负。

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

## 示例

返回值 1051125487:

```
SELECT MINUTES( '1998-07-13 06:07:12' ) FROM iq_dummy
```

返回值 240，表示两个时间之间的差值:

```
SELECT MINUTES( '1999-07-13 06:07:12',
  '1999-07-13 10:07:12' ) FROM iq_dummy
```

返回日期时间值 1999-05-12 21:10:07.000:

```
SELECT MINUTES( CAST( '1999-05-12 21:05:07'
AS DATETIME ), 5) FROM iq_dummy
```

## 另请参见

- CAST 函数 [数据类型转换] (第 143 页)
- CONVERT 函数 [数据类型转换] (第 153 页)
- HOURS 函数 [日期和时间] (第 206 页)
- MONTHS 函数 [日期和时间] (第 246 页)
- REPLACE 函数 [字符串] (第 286 页)
- SECOND 函数 [日期和时间] (第 296 页)
- WEEKS 函数 [日期和时间] (第 336 页)

## SQL 函数

- YEAR 函数 [日期和时间] (第 341 页)
- YEARS 函数 [日期和时间] (第 342 页)

### **MOD 函数 [数值]**

返回一个整数除以另一个整数之后产生的余数。

#### 语法

```
MOD ( dividend, divisor )
```

#### 参数

参数	描述
dividend	被除数，即除法的分子。
divisor	除数，即除法的分母。

#### 返回

SMALLINT

INT

NUMERIC

#### 注释

*dividend* 为负数时，除法的结果为负数或零。*divisor* 的符号没有影响。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 支持。% 运算符在 Adaptive Server 中用作模运算符。

#### 示例

以下语句返回值 2:

```
SELECT MOD( 5, 3 ) FROM iq_dummy
```

#### 另请参见

- REMAINDER 函数 [数值] (第 284 页)

### **MONTH 函数 [日期和时间]**

返回 1 到 12 之间的一个数字，该数字对应于给定日期的月份。

#### 语法

```
MONTH ( date-expression )
```



*参数*

参数	描述
date-expression	日期/时间值。

*返回*

SMALLINT

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句返回值 7:

```
SELECT MONTH( '1998-07-13' ) FROM iq_dummy
```

**MONTHNAME 函数 [日期和时间]**

返回指定日期表达式中的月份名称。

*语法*

```
MONTHNAME ( date-expression )
```

*参数*

参数	描述
date-expression	日期时间值。

*返回*

VARCHAR

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*当 DATE\_ORDER 选项设置为缺省值 *ymd* 时，以下语句返回值 **September**。

```
SELECT MONTHNAME( '1998-09-05' ) FROM iq_dummy
```

## MONTHS 函数 [日期和时间]

返回自任意开始日期和时间以来的月数，返回两个指定日期/时间之间的完整月数，或者将指定的 `integer-expression` 数量的月添加到某个时间中。

### 语法

```
MONTHS ( date-expression
| date-expression, datetime-expression
| date-expression, integer-expression )
```

### 参数

参数	描述
<code>date-expression</code>	日期和时间。
<code>integer-expression</code>	要添加到 <code>date-expression</code> 中的月数。如果 <code>integer-expression</code> 是负数，则从日期/时间值中减去相应的月数。如果提供整数表达式，则必须将 <code>date-expression</code> 显式转换为 <code>datetime</code> 数据类型。

### 返回

INT

TIMESTAMP

### 注释

第一种语法返回自任意开始日期以来的月数。此数字通常用于确定两个日期/时间表达式是否在同一年的同一月中。

```
MONTHS( invoice_sent ) = MONTHS( payment_received )
```

比较 **MONTH** 函数时通常会错误地包括发票发出后的 12 个月内进行的付款。

第二种语法返回从第一个日期到第二个日期之间的月数。数值可能为负。该值是根据两个日期之间一个月的第一天的数目计算出的。会忽略小时、分钟和秒钟。

第三种语法将 `integer-expression` 个月添加到给定日期中。如果新日期超出月末（如 **MONTHS** ('1992-01-31', 1)），则结果设置为该月的最后一天。如果 `integer-expression` 是负数，则从日期中减去相应的月数。会忽略小时、分钟和秒钟。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回值 23982:

```
SELECT MONTHS( '1998-07-13 06:07:12' ) FROM iq_dummy
```

以下语句返回值 2，表示两个日期之间的差值：

```
SELECT MONTHS ( '1999-07-13 06:07:12',
                '1999-09-13 10:07:12' ) FROM iq_dummy
```

以下语句返回日期时间值 1999-10-12 21:05:07.000：

```
SELECT MONTHS ( CAST ( '1999-05-12 21:05:07'
                      AS DATETIME ), 5) FROM iq_dummy
```

### 另请参见

- **CAST** 函数 [数据类型转换] (第 143 页)
- **CONVERT** 函数 [数据类型转换] (第 153 页)
- **HOURS** 函数 [日期和时间] (第 206 页)
- **MINUTES** 函数 [日期和时间] (第 242 页)
- **REPLACE** 函数 [字符串] (第 286 页)
- **SECOND** 函数 [日期和时间] (第 296 页)
- **WEEKS** 函数 [日期和时间] (第 336 页)
- **YEAR** 函数 [日期和时间] (第 341 页)
- **YEARS** 函数 [日期和时间] (第 342 页)

## **NEWID** 函数 [杂项]

生成 UUID (通用唯一标识符) 值。

返回的 UUID 值为二进制值。UUID 与 GUID (全局唯一标识符) 相同。

### 语法

```
NEWID ( )
```

### 参数

没有任何与 **NEWID()** 关联的参数。

### 返回

UNIQUEIDENTIFIER

### 注释

**NEWID()** 函数生成一个唯一标识符值。

UUID 可用来唯一地标识数据库中的对象。这样生成值会导致在一台计算机上生成的值与在另一台计算机上生成的该值不匹配，因此，它们还可以作为键用在复制和同步环境中。

**NEWID** 函数仅在下列位置受支持：

- 顶级查询块的 **SELECT** 列表
- **UPDATE** 语句的 **SET** 子句

## SQL 函数

- **INSERT...VALUES** 的 **VALUES** 子句

可以使用 **NEWID** 函数生成的值作为表中的列缺省值。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 支持。

### 示例

以下语句创建表 t1，然后更新该表，并将列 uid\_col 的值设置为由 **NEWID** 函数生成的唯一标识符，前提是该列的当前值为 **NULL**。

```
CREATE TABLE t1 (uid_col int);
UPDATE t1
    SET uid_col = NEWID()
    WHERE uid_col IS NULL
```

如果执行以下语句，

```
SELECT NEWID()
```

唯一标识符会以 **BINARY (16)** 形式返回。例如，其值可以是 **0xd3749fe09cf446e399913bc6434f1f08**。可以使用 **UIDTOSTR()** 函数将此字符串转换为可读格式。

### 另请参见

- 二进制数据类型 (第 352 页)
- **STRTOUUID** 函数 [字符串] (第 317 页)
- **UIDTOSTR** 函数 [字符串] (第 331 页)
- 字符数据类型 (第 345 页)
- 二进制数据类型 (第 759 页)

## **NEXT\_CONNECTION** 函数 [系统]

返回下一个连接编号，或第一个连接（如果参数为空）。

### 语法

```
NEXT_CONNECTION ( { connection-id }, { database-id } )
```

---

**注意：** 需要考虑 **CIS** 函数补偿性能方面的一些事宜。

---

### 返回

**INT**

## 参数

参数	描述
connection-id	一个整数，通常是从上一个 <b>NEXT_CONNECTION</b> 调用返回。如果 <i>connection-id</i> 为空值，则 <b>NEXT_CONNECTION</b> 返回最近的连接 ID。
database-id	一个表示当前服务器上某一数据库的整数。如果未提供 <i>database-id</i> ，则使用当前数据库。如果提供空值，则 <b>NEXT_CONNECTION</b> 返回下一个连接，而不考虑数据库。

## 注释

您可以使用 **NEXT\_CONNECTION** 来枚举数据库的连接。若要获取第一个连接，请传递空值；若要获取每个后续连接，请传递上一个返回值。当没有其它的连接时，此函数返回空值。

**NEXT\_CONNECTION** 可用于枚举数据库的连接。连接 ID 通常是按单调递增顺序创建的。此函数按相反顺序返回下一个连接 ID。

若要获取最近连接的连接 ID 值，请输入空值作为 *connection-id*。若要获取后续连接，请输入上一个返回值。当不存在按一定顺序排列的其它连接时，此函数返回 NULL。

如果要断开在特定时间之前创建的所有连接，可以使用 **NEXT\_CONNECTION**。不过，因为 **NEXT\_CONNECTION** 是以相反顺序返回连接 ID，所以不会返回在启动函数后建立的连接。如果想要确保所有连接都断开，请避免在运行 **NEXT\_CONNECTION** 前创建新连接。

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

## 示例

下面的语句返回当前数据库中第一个连接的标识符。该标识符是类似 10 的整数值。

```
SELECT NEXT_CONNECTION( NULL );
```

下面的语句返回类似 5 的值。

```
SELECT NEXT_CONNECTION( 10 );
```

下面的调用按相反顺序返回当前数据库上指定 *connection-id* 中的下一个连接 ID。

```
SELECT NEXT_CONNECTION( connection-id );
```

下面的调用按相反顺序返回指定 *connection-id* 中的下一个连接 ID（不考虑数据库）。

```
SELECT NEXT_CONNECTION( connection-id, NULL );
```

下面的调用按相反顺序返回指定数据库上指定 *connection-id* 中的下一个连接 ID。

```
SELECT NEXT_CONNECTION( connection-id, database-id );
```

## SQL 函数

下面的调用返回第一个（最早的）连接（不考虑数据库）。

```
SELECT NEXT_CONNECTION( NULL, NULL );
```

下面的调用返回指定数据库上的第一个（最早的）连接。

```
SELECT NEXT_CONNECTION( NULL, database-id );
```

### **NEXT\_DATABASE 函数 [系统]**

返回下一个数据库 ID 号，或第一个数据库（如果参数为空）。

*语法*

```
NEXT_DATABASE ( { NULL | database-id } )
```

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

*参数*

参数	描述
database-id	一个整数，它指定数据库的 ID 号。

*返回*

INT

*注释*

您可以使用 **NEXT\_DATABASE** 来枚举数据库服务器上运行的数据库。若要获取第一个数据库，请传递空值；若要获得每个后续数据库，请传递上一个返回值。当没有其它的数据库时，此函数返回空值。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句返回值 0，这是第一个数据库值：

```
SELECT NEXT_DATABASE( NULL ) FROM iq_dummy
```

以下语句返回 NULL，表示服务器上不再有数据库：

```
SELECT NEXT_DATABASE( 0 ) FROM iq_dummy
```

**另请参见**

- COL\_NAME 函数 [系统]（第 151 页）
- DB\_ID 函数 [系统]（第 183 页）
- DB\_NAME 函数 [系统]（第 184 页）

- DB\_PROPERTY 函数 [系统] (第 185 页)
- OBJECT\_ID 函数 [系统] (第 257 页)
- OBJECT\_NAME 函数 [系统] (第 257 页)

## NEXT\_HTTP\_HEADER 函数 [Web 服务]

返回下一个 HTTP 标头名称。

### 语法

```
NEXT_HTTP_HEADER( header-name )
```

### 参数

- **header-name** - 上一个请求标头的名称。如果 header-name 为 NULL，则此函数返回第一个 HTTP 请求标头的名称。

### 返回值

LONG VARCHAR。

---

**注意：** 结果数据类型为 LONG VARCHAR。要在 SELECT INTO 语句中使用 NEXT\_HTTP\_HEADER，您必须拥有“非结构化数据分析选项”许可或使用 CAST 并将 HTML\_DECODE 设置为正确数据类型和大小。

---

### 注释

此函数用于对 HTTP 请求标头进行迭代并返回下一个 HTTP 标头名称。当使用 NULL 调用时该函数，该函数返回第一个头名称。通过向此函数传递上一个标头的名称来检索后续的头。使用上一标头的名称调用此函数或不是从 Web 服务进行调用时，此函数返回 NULL。

重复调用该函数将会返回所有的头字段且仅返回一次，但不一定与其出现在 HTTP 请求中的顺序相同。

### 标准和兼容性

- SQL/2008 - 服务商扩充。

### 示例

在由 HTTP Web 服务所调用的存储过程内，使用以下语句会在数据库服务器消息窗口中显示 HTTP 请求标头的名称和值：

```
BEGIN
  declare header_name long varchar;
  declare header_value long varchar;
  set header_name = NULL;
header_loop:
  LOOP
    SET header_name = NEXT_HTTP_HEADER( header_name );
    IF header_name IS NULL THEN
```

```

    LEAVE header_loop
  END IF;
  SET header_value = HTTP_HEADER( header_name );
  MESSAGE 'HEADER: ', header_name, '=',
          header_value TO CONSOLE;
END LOOP;
END;
```

## NEXT\_HTTP\_VARIABLE 函数 [Web 服务]

返回下一个 HTTP 变量名称。

### 语法

```
NEXT_HTTP_VARIABLE ( var-name )
```

### 参数

- **var-name** - 上一个变量的名称。如果 **var-name** 为 NULL，则此函数返回第一个 HTTP 变量的名称。

### 返回值

LONG VARCHAR。

---

**注意：** 结果数据类型为 LONG VARCHAR。要在 SELECT INTO 语句中使用 NEXT\_HTTP\_VARIABLE，您必须拥有“非结构化数据分析选项”许可或使用 CAST 并将 NEXT\_HTTP\_HEADER 设置为正确数据类型和大小。

---

### 注释

此函数对请求内包含的 HTTP 变量进行迭代。当使用 NULL 调用时，该函数返回第一个变量的名称。使用上一个变量名称调用该函数则可检索到后续的变量名称。使用最后一个变量的名称调用此函数或不是从 Web 服务进行调用时，此函数返回 NULL。

重复调用此函数时将会返回所有变量且仅返回一次，但返回的顺序不一定是它们在 HTTP 请求中出现的顺序。如果 URL PATH 分别设置为 ON 或 ELEMENTS，则包括变量 url 或 url1、url2、……、url10。

### 标准和兼容性

- **SQL/2008** - 服务商扩充。

### 示例

在由 HTTP Web 服务所调用的存储过程内，使用以下语句会返回第一个 HTTP 变量的名称：

```

BEGIN
DECLARE variable_name LONG VARCHAR;
DECLARE variable_value LONG VARCHAR;
SET variable_name = NULL;
SET variable_name = NEXT_HTTP_VARIABLE( variable_name );
```



```
SET variable_value = HTTP_VARIABLE( variable_name );
END;
```

## **NOW** 函数 [日期和时间]

返回当前的日期和时间。这是 **CURRENT\_TIMESTAMP** 的传统语法。

语法

```
NOW ( * )
```

返回

TIMESTAMP

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

示例

以下语句返回当前日期和时间。

```
SELECT NOW(*) FROM iq_dummy
```

## **NTILE** 函数 [分析]

将查询结果分配到指定数量的表元中，并为表元中的每一行指派相应的表元号。

语法

```
NTILE ( expression1 )
OVER ( ORDER BY expression2 [ ASC | DESC ] )
```

参数

参数	描述
expression1	1 到 32767 之间的一个常量整数，该整数指定表元的数量。
expression2	排序规范，可以是涉及列引用、集合的任意有效表达式，也可以是调用这些项目的表达式。

注释

**NTILE** 为一个 **rank** 分析函数，可将查询结果分配到指定数量的表元中，并为表元中的每一行指派相应的表元号。您可以将结果集拆分为一百个（百分点）、十个（十分点）、四个（四分点）或其它数量的组。

**NTILE** 需要使用 **OVER (ORDER BY)** 子句。**ORDER BY** 子句指定要执行排序的参数以及每个组中行的排列顺序。此 **ORDER BY** 子句只在 **OVER** 子句中使用，而不是 **SELECT** 的 **ORDER BY**。不允许排序查询中的任何集合函数指定 **DISTINCT**。

**OVER** 子句表示函数对查询结果集进行操作。结果集是在对 **FROM**、**WHERE**、**GROUP BY** 和 **HAVING** 子句求值完成之后返回的行。**OVER** 子句定义要包括在 rank 分析函数计算中的行数据集。

**ASC** 或 **DESC** 参数用于指定升序或降序排序序列。升序是缺省值。

**NTILE** 只能在 **SELECT** 或 **INSERT** 语句的选择列表中或者 **SELECT** 语句的 **ORDER BY** 子句中使用。**NTILE** 可以在视图中或联合中使用。您不能在子查询中、**HAVING** 子句中，或者 **UPDATE** 或 **DELETE** 语句的选择列表中使用 **NTILE** 函数。每个查询仅允许使用一个 **NTILE** 函数。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 或 SQL Anywhere 支持

### 示例

以下示例使用 **NTILE** 函数来确定汽车经销商的销售状态。根据各位经销商售出的汽车数量将其分为四组。**ntile** 为 1 的经销商在汽车销售量排名中位于前 25%。

```
SELECT dealer_name, sales,
NTILE(4) OVER ( ORDER BY sales DESC )
FROM carSales;
```

dealer_name	sales	ntile
Boston	1000	1
Worcester	950	1
Providence	950	1
SF	940	1
Lowell	900	2
Seattle	900	2
Natick	870	2
New Haven	850	2
Portland	800	3
Houston	780	3
Hartford	780	3
Dublin	750	3
Austin	650	4
Dallas	640	4
Dover	600	4

要按销售量查找位于前 10% 的汽车经销商，请在示例 **SELECT** 语句中指定 **NTILE(10)**。类似地，要按销售量查找位于前 50% 的汽车经销商，请指定 **NTILE(2)**。

### 另请参见

- PERCENTILE\_CONT 函数 [分析] (第 262 页)
- PERCENTILE\_DISC 函数 [分析] (第 264 页)
- YEAR 函数 [日期和时间] (第 341 页)

## NULLIF 函数 [杂项]

通过比较表达式提供缩写的 **CASE** 表达式。

### 语法

```
NULLIF ( expression1, expression2 )
```

### 参数

参数	描述
expression1	要比较的表达式。
expression2	要比较的表达式。

### 返回

第一个参数的数据类型。

### 注释

**NULLIF** 比较两个表达式的值。

如果第一个表达式等于第二个表达式，则 **NULLIF** 返回空值。

如果第一个表达式不等于第二个表达式，或者第二个表达式为空值，则 **NULLIF** 返回第一个表达式。

**NULLIF** 函数提供了一种简便的方法来编写某些 **CASE** 表达式。**NULLIF** 等效于：

```

CASE WHEN expression1 = expression2
THEN NULL
ELSE expression1 END

```

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回 a：

```
SELECT NULLIF( 'a', 'b' ) FROM iq_dummy
```

以下语句返回 NULL：

```
SELECT NULLIF( 'a', 'a' ) FROM iq_dummy
```

### 另请参见

- CASE 表达式（第 35 页）

- 用于简写 CASE 表达式的 NULLIF 函数（第 36 页）

## NUMBER 函数 [杂项]

为查询结果中的各个连续行生成从 1 开始的编号。

语法

```
NUMBER ( * )
```

返回

INT

注释

仅在选择列表中或 UPDATE 语句的 SET 子句中使用 NUMBER 函数。例如，以下语句使用比前一行大 1 的数字来更新 seq\_id 列中的每一行。该数字的应用顺序由 ORDER BY 子句指定。

```
update empl  
set seq_id = number(*)  
order by empl_id
```

在 UPDATE 语句中，如果 NUMBER(\*) 函数用在 SET 子句中，并且 FROM 子句指定一对多连接，则 NUMBER(\*) 会生成唯一的数字，这些数字会增加，但由于删除行，可能不会按顺序递增。

从 SELECT 语句中使用 INSERT 时，还可以使用 NUMBER 生成主键，但生成顺序主键的首选机制是使用 IDENTITY/AUTOINCREMENT。

---

**注意：**如果在以下位置使用 NUMBER，则系统会产生语法错误：DELETE 语句、WHERE 子句、HAVING 子句、ORDER BY 子句、子查询、涉及集合的查询、任意约束、GROUP BY、DISTINCT、包含 UNION ALL 的查询，或派生表。

---

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

示例

以下语句返回此编号列表：

number(*)
1
2
3
4

<b>number(*)</b>
5

```
SELECT NUMBER ( * )
FROM Departments
WHERE DepartmentID > 10
```

## **OBJECT\_ID 函数 [系统]**

返回对象 ID。

语法

```
OBJECT_ID ( object-name )
```

参数

参数	描述
<i>object-name</i>	对象的名称。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。

示例

以下语句返回 *Customers* 表的对象 ID 100209:

```
SELECT OBJECT_ID ('CUSTOMERS') FROM iq_dummy
```

另请参见

- COL\_NAME 函数 [系统] (第 151 页)
- DB\_ID 函数 [系统] (第 183 页)
- DB\_NAME 函数 [系统] (第 184 页)
- DB\_PROPERTY 函数 [系统] (第 185 页)
- NEXT\_DATABASE 函数 [系统] (第 250 页)
- OBJECT\_NAME 函数 [系统] (第 257 页)
- INDEX\_COL 函数 [系统] (第 216 页)

## **OBJECT\_NAME 函数 [系统]**

返回对象名称。

语法

```
OBJECT_NAME ( object-id [ , database-id ] )
```

## 参数

参数	描述
object-id	对象 ID。
database-id	数据库 ID。

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。

## 示例

以下语句返回名称 "customer":

```
SELECT OBJECT_NAME ( 100209 ) FROM iq_dummy
```

## 另请参见

- BIT\_LENGTH 函数 [字符串] (第 140 页)
- BYTE\_LENGTH 函数 [字符串] (第 141 页)
- CHAR\_LENGTH 函数 [字符串] (第 147 页)
- COL\_LENGTH 函数 [系统] (第 150 页)
- DATALENGTH 函数 [系统] (第 164 页)
- LEN 函数 [字符串] (第 229 页)
- LENGTH 函数 [字符串] (第 230 页)
- OCTET\_LENGTH 函数 [字符串] (第 258 页)
- STR\_REPLACE 函数 [字符串] (第 315 页)
- COL\_NAME 函数 [系统] (第 151 页)
- DB\_ID 函数 [系统] (第 183 页)
- DB\_NAME 函数 [系统] (第 184 页)
- DB\_PROPERTY 函数 [系统] (第 185 页)
- NEXT\_DATABASE 函数 [系统] (第 250 页)
- OBJECT\_ID 函数 [系统] (第 257 页)

**OCTET\_LENGTH 函数 [字符串]**

返回不带符号的 64 位值，该值包含列参数的字节长度。

## 语法

```
OCTET_LENGTH ( column-name )
```

*参数*

参数	描述
column-name	列的名称。

*注释*

空值参数的返回值为空。

**OCTET\_LENGTH** 函数支持所有 SAP Sybase IQ 数据类型。

如果您有权使用非结构化数据分析功能，则可以将此函数与大对象数据一起使用。

请参见《非结构化数据分析》中的 **OCTET\_LENGTH** 函数。

*标准和兼容性*

- Sybase - SQL Anywhere 或 Adaptive Server 不支持。

*另请参见*

- **BIT\_LENGTH** 函数 [字符串] (第 140 页)
- **BYTE\_LENGTH** 函数 [字符串] (第 141 页)
- **CHAR\_LENGTH** 函数 [字符串] (第 147 页)
- **COL\_LENGTH** 函数 [系统] (第 150 页)
- **DATALENGTH** 函数 [系统] (第 164 页)
- **LEN** 函数 [字符串] (第 229 页)
- **LENGTH** 函数 [字符串] (第 230 页)
- **OBJECT\_NAME** 函数 [系统] (第 257 页)
- **STR\_REPLACE** 函数 [字符串] (第 315 页)

**PATINDEX 函数 [字符串]**

返回指定模式在第一次出现时的起始位置。

*语法*

```
PATINDEX ( '% pattern %', string-expression )
```

## 参数

参数	描述
pattern	要搜索的模式。对于具有通配符的模式，此字符串限制为 126 字节。如果省略前导百分号通配符，则模式出现在字符串起始处时， <b>PATINDEX</b> 返回 1 (1)，否则返回零。如果 <i>pattern</i> 以百分号通配符开头，则系统会将两个前导百分号通配符视为一个。  不带通配符（百分号 % 或下划线 _）的模式的长度最大可达 255 字节。
string-expression	要在其中搜索模式的字符串。

## 返回

INT

## 注释

**PATINDEX** 返回模式第一次出现的起始位置。如果搜索的字符串包含字符串模式的多个实例，**PATINDEX** 将仅返回第一个实例的位置。

模式使用的通配符与 **LIKE** 比较相同。下表列出了模式通配符。

表 49. PATINDEX 模式通配符

通配符	匹配项
_ (下划线)	任意一个字符
% (百分号)	包含零个或多个字符的任意字符串
[ ]	指定范围或集合中的任意单个字符
[^]	不在指定范围或集合中的任意单个字符

如果未找到模式，则 **PATINDEX** 返回零 (0)。

搜索长度超过 126 个字节的模式将返回 NULL。

搜索零长度的字符串将返回 1。

如果任一参数为 NULL，结果则为零 (0)。

在 **PATINDEX** 函数中返回或指定的所有位置或偏移量始终为字符偏移量，并且可能不同于多字节数据的字节偏移量。

**PATINDEX** 为 CHAR 和 VARCHAR 列返回不带符号的 32 位整数位置。

如果您有权使用非结构化数据分析功能，则可以将此函数与大对象数据一起使用。

请参见《非结构化数据分析》中的 **PATINDEX** 函数。



### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容。

### 示例

以下语句返回值 2:

```
SELECT PATINDEX( '%hoco%', 'chocolate' ) FROM iq_dummy
```

以下语句返回值 11:

```
SELECT PATINDEX ( '%4_5_', '0a1A 2a3A 4a5A' ) FROM iq_dummy
```

### 另请参见

- LIKE 条件 (第 50 页)
- LOCATE 函数 [字符串] (第 234 页)

## PERCENT\_RANK 函数 [分析]

根据 **ORDER BY** 子句的定义，计算查询返回的一行相对于该查询返回的其它行的 (小数) 位置。

它返回介于 0 和 1 之间的小数值。

### 语法

```
PERCENT_RANK ( ) OVER ( ORDER BY expression [ ASC | DESC ] )
```

### 参数

参数	描述
expression	排序规范，可以是涉及列引用、集合的任意有效表达式，也可以是调用这些项目的表达式。

### 返回

**PERCENT\_RANK** 函数返回介于 0 和 1 之间的 DOUBLE 值。

### 注释

**PERCENT\_RANK** 是 rank 分析函数。行 R 的百分比秩是指在 **OVER** 子句中指定的组中某个行的秩减去一，再除以在 **OVER** 子句中指定的组中的总行数减去一。

**PERCENT\_RANK** 返回介于 0 和 1 之间的值。第一行的百分比秩为零。

行的 **PERCENT\_RANK** 的计算公式为

```
(Rx - 1) / (NtotalRow - 1)
```

其中 *R<sub>x</sub>* 是组中某行的秩位置，*NtotalRow* 是 **OVER** 子句指定的组中的总行数。

**PERCENT\_RANK** 需要使用 **OVER (ORDER BY)** 子句。**ORDER BY** 子句指定要执行排序的参数以及每个组中行的排列顺序。此 **ORDER BY** 子句只在 **OVER** 子句中使用，而不是 **SELECT** 的 **ORDER BY**。不允许排序查询中的任何集合函数指定 **DISTINCT**。

**OVER** 子句表示函数对查询结果集进行操作。结果集是在对 **FROM**、**WHERE**、**GROUP BY** 和 **HAVING** 子句求值完成之后返回的行。**OVER** 子句定义要包括在 rank 分析函数计算中的行数据集。

**ASC** 或 **DESC** 参数用于指定升序或降序排序序列。升序是缺省值。

**PERCENT\_RANK** 只能在 **SELECT** 或 **INSERT** 语句的选择列表中或者 **SELECT** 语句的 **ORDER BY** 子句中使用。**PERCENT\_RANK** 可以在视图中或联合中使用。您不能在子查询中、**HAVING** 子句中，或者 **UPDATE** 或 **DELETE** 语句的选择列表中使用 **PERCENT\_RANK** 函数。每个查询仅允许使用一个 rank 分析函数。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 或 SQL Anywhere 支持。

### 示例

以下语句说明了 **PERCENT\_RANK** 函数的用法：

```
SELECT s_suppkey, SUM(s_acctBal) AS sum_acctBal,
PERCENT_RANK() OVER ( ORDER BY SUM(s_acctBal) DESC )
AS percent_rank_all FROM supplier GROUP BY s_suppkey;
```

s_suppkey	sum_acctBal	percent_rank_all
supplier#011	200000	0
supplier#002	200000	0
supplier#013	123000	0.3333
supplier#004	110000	0.5
supplier#035	110000	0.5
supplier#006	50000	0.8333
supplier#021	10000	1

## PERCENTILE\_CONT 函数 [分析]

给出一个百分点，返回与该百分点对应的值。假定是连续分布数据模型。

**注意：** 如果仅需要计算百分点，请改用 **NTILE** 函数，将值设为 100。

### 语法

```
PERCENTILE_CONT ( expression1 )
WITHIN GROUP ( ORDER BY expression2 [ ASC | DESC ] )
```

## 参数

参数	描述
expression1	一个 numeric 数据类型的常量，范围从 0 到 1（包含这两个数）。如果参数为 NULL，则返回 “wrong argument for percentile”（百分点的参数错误）错误。如果参数值小于 0 或大于 1，则返回 “data value out of range”（数据值超出范围）错误。
expression2	排序规范，必须为涉及列引用的单个表达式。不允许多个表达式，并且在此排序表达式中不允许使用 rank 分析函数、集合函数或子查询。

## 注释

逆分布分析函数返回第 k 个百分点值，该值可用于帮助为一组数据建立可接受阈值。函数 **PERCENTILE\_CONT** 采用一个百分点值作为函数参数，并对在 **WITHIN GROUP** 子句中指定的一组数据或整个数据集执行操作。该函数为每组返回一个值。如果查询中的 **GROUP BY** 列不存在，则结果为单个行。结果的数据类型与在 **WITHIN GROUP** 子句中指定的它的 **ORDER BY** 项的数据类型相同。**PERCENTILE\_CONT** 的 **ORDER BY** 表达式的数据类型必须是数值。

**PERCENTILE\_CONT** 需要使用 **WITHIN GROUP (ORDER BY)** 子句。

**ORDER BY** 子句（必须存在）指定要执行百分点函数的表达式以及每个组中行的排列顺序。对于 **PERCENTILE\_CONT** 函数，此表达式的数据类型必须是数值。此 **ORDER BY** 子句只在 **WITHIN GROUP** 子句中使用，而不是 **SELECT** 的 **ORDER BY**。

**WITHIN GROUP** 子句将查询结果分布到排序数据集中，函数通过此数据集计算结果。**WITHIN GROUP** 子句必须包含单个排序项。如果 **WITHIN GROUP** 子句包含多个或者 0 个排序项，系统将报告错误。

ASC 或 DESC 参数用于指定升序或降序排序序列。升序是缺省值。

子查询、**HAVING** 子句、视图或联合中允许使用 **PERCENTILE\_CONT** 函数。

**PERCENTILE\_CONT** 可在使用简单非分析集合函数的任意位置使用。

**PERCENTILE\_CONT** 函数忽略数据集中的空值。

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 或 SQL Anywhere 支持。

## 示例

以下示例使用 **PERCENTILE\_CONT** 函数来确定某区域前 10 个百分点值所对应的汽车销售量。

示例中使用了以下数据集：

sales	region	dealer_name
900	Northeast	Boston

800	Northeast	Worcester
800	Northeast	Providence
700	Northeast	Lowell
540	Northeast	Natick
500	Northeast	New Haven
450	Northeast	Hartford
800	Northwest	SF
600	Northwest	Seattle
500	Northwest	Portland
400	Northwest	Dublin
500	South	Houston
400	South	Austin
300	South	Dallas
200	South	Dover

以下 **SELECT** 语句包含 **PERCENTILE\_CONT** 函数：

```
SELECT region, PERCENTILE_CONT(0.1)
WITHIN GROUP ( ORDER BY sales DESC )
FROM carSales GROUP BY region;
```

**SELECT** 语句的结果列出了某区域前 10 个百分点值所对应的汽车销售量：

region	percentile_cont
Northeast	840
Northwest	740
South	470

另请参见

- [NTILE 函数 \[分析\]](#) (第 253 页)
- [PERCENTILE\\_DISC 函数 \[分析\]](#) (第 264 页)

## **PERCENTILE\_DISC 函数 [分析]**

给出一个百分点，返回与该百分点对应的值。假定是离散分布数据模型。

---

**注意：** 如果仅需要计算百分点，请改用 **NTILE** 函数，将值设为 100。

---

语法

```
PERCENTILE_DISC ( expression1 )
WITHIN GROUP ( ORDER BY expression2 [ ASC | DESC ] )
```

参数

参数	描述
expression1	一个 numeric 数据类型的常量，范围从 0 到 1 (包含这两个数)。如果参数为 NULL，将返回 “wrong argument for percentile” (百分点的参数错误) 错误。如果参数值小于 0 或大于 1，将返回 “data value out of range” (数据值超出范围) 错误。

参数	描述
expression2	排序规范，必须为涉及列引用的单个表达式。不允许多个表达式，并且在此排序表达式中不允许使用 <b>rank</b> 分析函数、集合函数或子查询。

### 注释

逆分布分析函数返回第 **k** 个百分点值，该值可用于帮助为一组数据建立可接受阈值。函数 **PERCENTILE\_DISC** 采用一个百分点值作为函数参数，并对在 **WITHIN GROUP** 子句中指定的一组数据或整个数据集执行运算。该函数为每个组返回一个值。如果查询中的 **GROUP BY** 列不存在，则结果为单个行。结果的数据类型与在 **WITHIN GROUP** 子句中指定的 **ORDER BY** 项的数据类型相同。**PERCENTILE\_DISC** 支持可在 SAP Sybase IQ 中排序的所有数据类型。

**PERCENTILE\_DISC** 需要使用 **WITHIN GROUP (ORDER BY)** 子句。

**ORDER BY** 子句（必须存在）指定要执行百分点函数的表达式以及每个组中行的排列顺序。此 **ORDER BY** 子句只在 **WITHIN GROUP** 子句中使用，而不是 **SELECT** 的 **ORDER BY**。

**WITHIN GROUP** 子句将查询结果分布到排序数据集中，函数通过此数据集计算结果。**WITHIN GROUP** 子句必须包含单个排序项。如果 **WITHIN GROUP** 子句包含多个或者 0 个排序项，系统将报告错误。

**ASC** 或 **DESC** 参数用于指定升序或降序排序序列。升序是缺省值。

子查询、**HAVING** 子句、视图或联合中允许使用 **PERCENTILE\_DISC** 函数。

**PERCENTILE\_DISC** 可在使用简单非分析集合函数的任意位置使用。**PERCENTILE\_DISC** 函数忽略数据集中的空值。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 或 SQL Anywhere 支持。

### 示例

以下示例使用 **PERCENTILE\_DISC** 函数来确定某区域前 10 个百分点值所对应的汽车销售量。

示例中使用了以下数据集：

sales	region	dealer_name
900	Northeast	Boston
800	Northeast	Worcester
800	Northeast	Providence
700	Northeast	Lowell
540	Northeast	Natick
500	Northeast	New Haven
450	Northeast	Hartford
800	Northwest	SF
600	Northwest	Seattle
500	Northwest	Portland

## SQL 函数

400	Northwest	Dublin
500	South	Houston
400	South	Austin
300	South	Dallas
200	South	Dover

以下 **SELECT** 语句包含 **PERCENTILE\_DISC** 函数：

```
SELECT region, PERCENTILE_DISC(0.1)
WITHIN GROUP ( ORDER BY sales DESC )
FROM carSales GROUP BY region;
```

**SELECT** 语句的结果列出了某区域前 10 个百分点值所对应的汽车销售量：

region	percentile_cont
Northeast	900
Northwest	800
South	500

另请参见

- [NTILE 函数 \[分析\]](#) (第 253 页)
- [PERCENTILE\\_CONT 函数 \[分析\]](#) (第 262 页)

## **PI 函数 [数值]**

返回数值 PI。

语法

```
PI ( * )
```

返回

DOUBLE

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - Adaptive Server 支持 **PI()** 函数，但不支持 **PI(\*)**。

示例

以下语句返回值 3.141592653……

```
SELECT PI ( * ) FROM iq_dummy
```

## **POWER 函数 [数值]**

以一个数字为底数另一个数字为指数计算乘方值。

语法

```
POWER ( numeric-expression1, numeric-expression2 )
```

## 参数

参数	描述
numeric-expression1	底数。
numeric-expression2	指数。

## 返回

DOUBLE

## 注释

将 *numeric-expression1* 增加至乘方 *numeric-expression2*。

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

## 示例

以下语句返回值 64:

```
SELECT Power( 2, 6 ) FROM iq_dummy
```

**PROPERTY 函数 [系统]**

以字符串形式返回指定的服务器级属性值。

## 语法

```
PROPERTY ( { property-id | property-name } )
```

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

## 参数

表 50.

参数	描述
property-id	一个整数，该整数表示服务器级属性的属性号。通过 <b>PROPERTY_NUMBER</b> 函数可以确定此数字。当在一组属性间循环时，常使用 <i>property-id</i> 。
property-name	给出属性名称的字符串。

## 返回

VARCHAR

### 注释

每个属性都有编号和名称，但编号在不同的版本间可能有变化，因此不应用作给定属性的可靠标识符。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回当前数据库服务器的名称：

```
SELECT PROPERTY( 'Name' ) FROM iq_dummy
```

### 另请参见

- PROPERTY\_NAME 函数 [系统] (第 269 页)
- PROPERTY\_NUMBER 函数 [系统] (第 270 页)
- CONNECTION\_PROPERTY 函数 [系统] (第 152 页)
- 可用于服务器的属性 (第 92 页)
- 各数据库可用的属性 (第 115 页)
- 连接属性 (第 92 页)

## **PROPERTY\_DESCRIPTION 函数 [系统]**

返回有关属性的说明。

### 语法

```
PROPERTY_DESCRIPTION ( { property-id | property-name } )
```

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

### 参数

参数	描述
<i>property-id</i>	一个整数，该整数表示属性的属性号。通过 <b>PROPERTY_NUMBER</b> 函数可以确定此数字。当在一组属性间循环时，常使用 <i>property-id</i> 。
<i>property-name</i>	给出属性名称的字符串。

### 返回

VARCHAR



### 注释

每个属性都有编号和名称，但编号在不同的版本间可能有变化，因此不应用作给定属性的可靠标识符。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回说明“索引插入数”：

```
SELECT PROPERTY_DESCRIPTION( 'IndAdd' ) FROM iq_dummy
```

## **PROPERTY\_NAME 函数 [系统]**

返回具有所提供属性号的属性的名称。

### 语法

```
PROPERTY_NAME ( property-id )
```

---

**注意：** 需要考虑 CIS 函数补偿性能方面的一些事宜。

---

### 参数

参数	描述
property-id	属性的属性号。

### 返回

VARCHAR

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回与属性号 126 关联的属性。它所指的实际属性因版本而异。

```
SELECT PROPERTY_NAME( 126 ) FROM iq_dummy
```

### 另请参见

- PROPERTY 函数 [系统] (第 267 页)
- PROPERTY\_NUMBER 函数 [系统] (第 270 页)
- CONNECTION\_PROPERTY 函数 [系统] (第 152 页)

## SQL 函数

- 可用于服务器的属性 (第 92 页)
- 各数据库可用的属性 (第 115 页)
- 连接属性 (第 92 页)

### **PROPERTY\_NUMBER 函数 [系统]**

返回具有所提供属性名称的属性的属性号。

#### 语法

```
PROPERTY_NUMBER ( property-name )
```

**注意:** 需要考虑 CIS 函数补偿性能方面的一些事宜。

#### 参数

参数	描述
<i>property-name</i>	属性名称。

#### 返回

INT

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

#### 示例

以下语句返回一个整数值。实际值因版本而异。

```
SELECT PROPERTY_NUMBER( 'PAGESIZE' ) FROM iq_dummy
```

#### 另请参见

- PROPERTY 函数 [系统] (第 267 页)
- PROPERTY\_NAME 函数 [系统] (第 269 页)
- CONNECTION\_PROPERTY 函数 [系统] (第 152 页)
- 可用于服务器的属性 (第 92 页)
- 各数据库可用的属性 (第 115 页)
- 连接属性 (第 92 页)

### **QUARTER 函数 [日期和时间]**

返回一个表示所提供的日期表达式中年的季度的数字。

#### 语法

```
QUARTER( date-expression )
```

## 参数

参数	描述
date-expression	日期。

## 返回

INT

## 注释

下表列出了一年的各个季度中的日期。

表 51. 年的季度值

Quarter	期限 (含起止日期)
1	1 月 1 日到 3 月 31 日
2	4 月 1 日到 6 月 30 日
3	7 月 1 日到 9 月 30 日
4	10 月 1 日到 12 月 31 日

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

## 示例

如果将 **DATE\_ORDER** 选项设置为 *ymd* 的缺省值，则下列语句返回值 2:

```
SELECT QUARTER ( '1987/05/02' ) FROM iq_dummy
```

**RADIANS 函数 [数值]**

将数字由角度转换成弧度。

## 语法

```
RADIANS ( numeric-expression )
```

## 参数

参数	描述
numeric-expression	数字 (以角度为单位)。该角度将转换成弧度。

## 返回

DOUBLE

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回近似值 0.5236:

```
SELECT RADIANS( 30 ) FROM iq_dummy
```

## **RAND 函数 [数值]**

返回一个 DOUBLE 精度的随机数字  $x$ ，其中  $0 \leq x < 1$ ，使用可选种子。

### 语法

```
RAND ( [ integer-expression ] )
```

### 参数

参数	描述
integer-expression	用于创建随机数字的可选种子值。该参数可用于创建可重复的随机数字序列。

### 返回

DOUBLE

### 注释

如果在只包含 IQ 存储中的表的查询中使用 **FROM** 子句以及某个参数调用 **RAND**，则该函数返回任意可重复值。

如果不调用任何参数，则 **RAND** 是非确定性函数。连续调用 **RAND** 可能会返回不同的值。查询优化程序不会对 **RAND** 函数的结果进行高速缓存。

**注意：** 由 **RAND** 返回的值因您是否使用 **FROM** 子句以及引用的表是在 **SYSTEM** 中创建还是在 **IQ** 存储库中创建而异。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 示例

以下语句返回某个表的 5% 的样本:

```
SELECT AVG(table1.number_of_cars), AVG(table1.number_of_tvs) FROM  
table1 WHERE RAND(ROWID(table1)) < .05 and table1.income < 50000;
```

以下语句返回近似值 941392926249216914:

```
SELECT RAND( 4 ) FROM iq_dummy
```

## RANK 函数 [分析]

排列组中的项目。

### 语法

```
RANK ( ) OVER ( [ PARTITION BY ] ORDER BY expression [ ASC | DESC ] )
```

### 参数

参数	描述
expression	排序规范，可以是涉及列引用、集合的任意有效表达式，也可以是调用这些项目的表达式。

### 返回

INTEGER

### 注释

**RANK** 是 rank 分析函数。行 R 的秩是指位于 R 前面但不是 R 的对等行的行数。如果两个或更多个行在 **OVER** 子句指定的组中或者在整个结果集中是不重复的，则连续秩编号中存在一个或多个空位。**RANK** 和 **DENSE\_RANK** 之间的区别是：当存在并列排名时，**DENSE\_RANK** 不会保留空位，而 **RANK** 会保留空位。

**RANK** 需要使用 **OVER (ORDER BY)** 子句。**ORDER BY** 子句指定要执行排序的参数以及每个组中行的排列顺序。此 **ORDER BY** 子句只在 **OVER** 子句中使用，而不是 **SELECT** 的 **ORDER BY**。不允许排序查询中的任何集合函数指定 **DISTINCT**。

**OVER (ORDER BY)** 子句中的 **PARTITION BY** 窗口分区子句是可选的。

**ASC** 或 **DESC** 参数用于指定升序或降序排序序列。升序是缺省值。

**OVER** 子句表示函数对查询结果集进行操作。结果集是在对 **FROM**、**WHERE**、**GROUP BY** 和 **HAVING** 子句求值完成之后返回的行。**OVER** 子句定义要包括在 rank 分析函数计算中的行数据集。

**RANK** 只能在 **SELECT** 或 **INSERT** 语句的选择列表中或者 **SELECT** 语句的 **ORDER BY** 子句中使用。**RANK** 可以在视图中或联合中使用。您不能在子查询中、**HAVING** 子句中，或者 **UPDATE** 或 **DELETE** 语句的选择列表中使用 **RANK** 函数。每个查询仅允许使用一个 rank 分析函数。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - Adaptive Server 或 SQL Anywhere 不支持。

*示例*

下面的语句说明了 **RANK** 函数的用法：

```
SELECT Surname, Sex, Salary, RANK() OVER (PARTITION BY Sex
ORDER BY Salary DESC) AS RANK FROM Employees
WHERE State IN ('CA', 'AZ') AND DepartmentID IN (200, 300)
ORDER BY Sex, Salary DESC;
```

以上查询的结果：

Surname	Sex	Salary	RANK
Savarino	F	72300.000	1
Jordan	F	51432.000	2
Clark	F	45000.000	3
Coleman	M	42300.000	1
Overbey	M	39300.000	2

**另请参见**

- **DENSE\_RANK** 函数 [分析] (第 186 页)

**REGR\_AVGX 函数 [集合]**

计算回归线的独立变量的平均值。

*语法 1*

```
REGR_AVGX (dependent-expression, independent-expression)
```

*语法 2*

```
REGR_AVGX (dependent-expression, independent-expression)
```

```
OVER (window-spec)
```

*参数*

参数	描述
dependent-expression	受独立变量影响的变量。
independent-expression	影响结果的变量。

*返回*

**DOUBLE**

*注释*

此函数将其参数转换为 **DOUBLE**，执行双精度浮点计算，并返回一个 **DOUBLE** 作为结果。如果将 **REGR\_AVGX** 应用于空集，它将返回空值。

在消除所有 dependent-expression 或 independent-expression 为 NULL 的值对之后，会将此函数应用于 dependent-expression 和 independent-expression 对的集合。此函数在

一次数据传递的同时进行计算。消除空值后，将进行以下计算，其中 *x* 表示 *independent-expression*：

```
AVG (x)
```

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 **ROLLUP** 和 **CUBE**，也不支持 **DISTINCT**。

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

#### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

#### 示例

以下示例计算相关变量（雇员年龄）的平均值：

```
SELECT REGR_AVGX( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) ) FROM Employees;
```

#### 另请参见

- 窗口化集合函数的用法（第 78 页）

## REGR\_AVGY 函数 [集合]

计算回归线的相关变量的平均值。

#### 语法 1

```
REGR_AVGY (dependent-expression, independent-expression)
```

#### 语法 2

```
REGR_AVGY (dependent-expression, independent-expression)
```

```
OVER (window-spec)
```

#### 参数

参数	描述
<i>dependent-expression</i>	受独立变量影响的变量。
<i>independent-expression</i>	影响结果的变量。

#### 返回

DOUBLE

*注释*

此函数将其参数转换为 **DOUBLE**，执行双精度浮点计算，并返回一个 **DOUBLE** 作为结果。如果将 **REGR\_AVGY** 应用于空集，它将返回空值。

在消除所有 **dependent-expression** 或 **independent-expression** 为 **NULL** 的值对之后，会将此函数应用于 **dependent-expression** 和 **independent-expression** 对的集合。此函数在一次数据传递的同时进行计算。消除空值后，将进行以下计算，其中 **y** 表示 **dependent-expression**：

```
AVG (y)
```

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 **ROLLUP** 和 **CUBE**，也不支持 **DISTINCT**。

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

*标准和兼容性*

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

*示例*

以下示例计算不相关变量（雇员薪水）的平均值。此函数返回值 49988.6232：

```
SELECT REGR_AVGY( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) )FROM
Employees;
```

**REGR\_COUNT 函数 [集合]**

返回一个整数，该整数表示用于拟合回归线的非 Null 数字对的数量。

*语法 1*

```
REGR_COUNT(dependent-expression, independent-expression)
```

*语法 2*

```
REGR_COUNT(dependent-expression, independent-expression)
```

```
OVER (window-spec)
```

*参数*

参数	描述
<b>dependent-expression</b>	受独立变量影响的变量。
<b>independent-expression</b>	影响结果的变量。



返回

INTEGER

注释

此函数返回一个 UNSIGNED BIGINT 值作为结果。

---

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 ROLLUP 和 CUBE，也不支持 DISTINCT。

---

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

示例

以下示例返回一个值，表示用于拟合回归线的非 NULL 值对的数量。此函数返回值 75：

```
SELECT REGR_COUNT( Salary, ( YEAR( NOW() ) -
YEAR( BirthDate ) ) ) FROM Employees;
```

另请参见

- 窗口化集合函数的用法（第 78 页）

## **REGR\_INTERCEPT** 函数 [集合]

计算可以最好地拟合相关和独立变量的线性回归线的 y 截距。

语法 1

```
REGR_INTERCEPT (dependent-expression, independent-expression)
```

语法 2

```
REGR_INTERCEPT (dependent-expression, independent-expression)
```

```
OVER (window-spec)
```

参数

参数	描述
dependent-expression	受独立变量影响的变量。
independent-expression	影响结果的变量。

*返回*

DOUBLE

*注释*

此函数将其参数转换为 **DOUBLE**，执行双精度浮点计算，并返回一个 **DOUBLE** 作为结果。如果将 **REGR\_INTERCEPT** 应用于空集，它将返回空值。

在消除所有 *dependent-expression* 或 *independent-expression* 为空值的值对之后，系统会将此函数应用于 *dependent-expression* 和 *independent-expression* 对的集合。此函数在一次数据传递的同时进行计算。消除空值后，系统将进行以下计算，其中 *y* 表示 *dependent-expression*，*x* 表示 *independent-expression*：

$$\text{AVG}(y) - \text{REGR\_SLOPE}(y, x) * \text{AVG}(x)$$

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 **ROLLUP** 和 **CUBE**，也不支持 **DISTINCT**。

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

*标准和兼容性*

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

*示例*

以下示例返回值 1874.5805688517603：

```
SELECT REGR_INTERCEPT( Salary, ( YEAR( NOW() ) -
YEAR( BirthDate ) ) ) FROM Employees;
```

**另请参见**

- 窗口化集合函数的用法（第 78 页）

**REGR\_R2 函数 [集合]**

计算回归线的决定系数（也称为 R 平方或拟合优度统计）。

*语法 1*

$$\text{REGR\_R2}(\text{dependent-expression}, \text{independent-expression})$$
*语法 2*

$$\text{REGR\_R2}(\text{dependent-expression}, \text{independent-expression})$$

$$\text{OVER}(\text{window-spec})$$

## 参数

参数	描述
dependent-expression	受独立变量影响的变量。
independent-expression	影响结果的变量。

## 返回

DOUBLE

## 注释

此函数将其参数转换为 **DOUBLE**，执行双精度浮点计算，并返回一个 **DOUBLE** 作为结果。如果将 **REGR\_R2** 应用于空集，它将返回空值。

**REGR\_R2** 应用于排除所有 dependent-expression 或 independent-expression 为 Null 的对后剩下的 (dependent-expression 和 independent-expression) 对的集合。然后对 SAP Sybase IQ 应用以下算法：

- 如果  $\text{VAR\_POP}(x) = 0$ ，**REGR\_R2** 将计算  $\text{VAR\_POP}(x)$  并返回空值；如果  $\text{VAR\_POP}(y) = 0$ ，则会计算  $\text{VAR\_POP}(y)$  并返回值 1。
- 如果  $\text{VAR\_POP}(x)$  或  $\text{VAR\_POP}(y)$  都不是零，则返回值为  $\text{POWER}(\text{CORR}(y,x),2)$ 。

其中， $y$  表示 *dependent-expression*， $x$  表示 *independent-expression*。

---

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 **ROLLUP** 和 **CUBE**，也不支持 **DISTINCT**。

---

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

## 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

## 示例

以下示例返回值 0.19379959710325653：

```
SELECT REGR_R2( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) )FROM
Employees;
```

## 另请参见

- 窗口化集合函数的用法（第 78 页）

## REGR\_SLOPE 函数 [集合]

计算与非空值对拟合的线性回归线的斜率。

### 语法 1

```
REGR_SLOPE (dependent-expression, independent-expression)
```

### 语法 2

```
REGR_SLOPE (dependent-expression, independent-expression)
```

```
OVER (window-spec)
```

### 参数

参数	描述
dependent-expression	受独立变量影响的变量。
independent-expression	影响结果的变量。

### 返回

DOUBLE

### 注释

此函数将其参数转换为 DOUBLE，执行双精度浮点计算，并返回一个 DOUBLE 作为结果。如果将 REGR\_SLOPE 应用于空集，它将返回空值。

在消除所有 dependent-expression 或 independent-expression 为空值的值对之后，系统会将 REGR\_SLOPE 应用于 dependent-expression 和 independent-expression 对的集合。此函数在一次数据传递的同时进行计算。消除空值后，系统将进行以下计算，其中 y 表示 dependent-expression，x 表示 independent-expression：

```
COVAR_POP (x, y) / VAR_POP (y)
```

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 ROLLUP 和 CUBE，也不支持 DISTINCT。

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

### 示例

以下示例返回值 935.3429749445614：

```
SELECT REGR_SLOPE( Salary, ( YEAR( NOW() ) -
YEAR( BirthDate ) ) )FROM Employees;
```

另请参见

- 窗口化集合函数的用法（第 78 页）

## **REGR\_SXX 函数 [集合]**

计算与非空值对拟合的线性回归线的斜率。

语法 1

```
REGR_SXX(dependent-expression, independent-expression)
```

语法 2

```
REGR_SXX(dependent-expression, independent-expression)
```

```
OVER (window-spec)
```

参数

参数	描述
<i>dependent-expression</i>	受独立变量影响的变量。
<i>independent-expression</i>	影响结果的变量。

返回

DOUBLE

注释

此函数将其参数转换为 DOUBLE，执行双精度浮点计算，并返回一个 DOUBLE 作为结果。如果将 **REGR\_SXX** 应用于空集，它将返回空值。

在消除所有 *dependent-expression* 或 *independent-expression* 为空值的值对之后，系统会将此函数应用于 *dependent-expression* 和 *independent-expression* 对的集合。此函数在一次数据传递的同时进行计算。消除空值后，系统将进行以下计算，其中 *y* 表示 *dependent-expression*，*x* 表示 *independent-expression*：

```
REGR_COUNT(y, x) * VAR_POP(x)
```

**注意：**使用语法 1 时，**GROUP BY** 子句不支持 ROLLUP 和 CUBE，也不支持 DISTINCT。

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

## 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

## 示例

以下示例返回值 5916.4800000000105:

```
SELECT REGR_SXX( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) FROM Employees;
```

## 另请参见

- 窗口化集合函数的用法 (第 78 页)

## REGR\_SXY 函数 [集合]

返回相关和独立变量的乘积之和。使用 REGR\_SXY 可以计算回归模型的统计有效性。

### 语法 1

```
REGR_SXY(dependent-expression, independent-expression)
```

### 语法 2

```
REGR_SXY(dependent-expression, independent-expression)
```

```
OVER (window-spec)
```

## 参数

参数	描述
<i>dependent-expression</i>	受独立变量影响的变量。
<i>independent-expression</i>	影响结果的变量。

## 返回

DOUBLE

## 注释

此函数将其参数转换为 DOUBLE，执行双精度浮点计算，并返回一个 DOUBLE 作为结果。如果将此函数应用于空集，它将返回 NULL。

在消除所有 *dependent-expression* 或 *independent-expression* 为空值的值对之后，系统会将此函数应用于 *dependent-expression* 和 *independent-expression* 对的集合。此函数在一次数据传递的同时进行计算。消除空值后，系统将进行以下计算，其中 y 表示 *dependent-expression*，x 表示 *independent-expression*：

```
REGR_COUNT(x, y) * COVAR_POP(x,
y)
```

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 ROLLUP 和 CUBE，也不支持 DISTINCT。

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

#### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

#### 示例

以下示例返回值 5533938.004400015。

```
SELECT REGR_SXY( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) )FROM
Employees;
```

#### 另请参见

- 窗口化集合函数的用法（第 78 页）

## REGR\_SYY 函数 [集合]

返回可以计算回归模型的统计有效性的值。

#### 语法 1

```
REGR_SYY(dependent-expression, independent-expression)
```

#### 语法 2

```
REGR_SYY(dependent-expression, independent-expression)
```

```
OVER (window-spec)
```

#### 参数

参数	描述
dependent-expression	受独立变量影响的变量。
independent-expression	影响结果的变量。

#### 返回

DOUBLE

#### 注释

此函数将其参数转换为 **DOUBLE**，执行双精度浮点计算，并返回一个 **DOUBLE** 作为结果。如果将 **REGR\_SYY** 应用于空集，它将返回空值。

## SQL 函数

在消除所有 *dependent-expression* 或 *independent-expression* 为空值的值对之后，系统会将此函数应用于 *dependent-expression* 和 *independent-expression* 对的集合。此函数在一次数据传递的同时进行计算。消除空值后，系统将进行以下计算，其中 *y* 表示 *dependent-expression*，*x* 表示 *independent-expression*：

```
REGR_COUNT(x, y) * VAR_POP(y)
```

---

**注意：** 使用语法 1 时，**GROUP BY** 子句不支持 ROLLUP 和 CUBE，也不支持 DISTINCT。

---

语法 2 表示作为 **SELECT** 语句中的窗口函数的用法。这样，您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。
- Sybase - 与 SQL Anywhere 兼容。

### 示例

以下示例返回值 26、708、672、843.3002：

```
SELECT REGR_SYY( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) )FROM Employees;
```

### 另请参见

- 窗口化集合函数的用法（第 78 页）

## REMAINDER 函数 [数值]

返回一个整数除以另一个整数之后产生的余数。

### 语法

```
REMAINDER ( dividend, divisor )
```

### 参数

参数	描述
dividend	被除数，即除法的分子。
divisor	除数，即除法的分母。

### 返回

INTEGER

NUMERIC

### 注释

**REMAINDER** 与 **MOD** 函数相同。



### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。% (模) 运算符和除法运算符可用于产生余数。

### 示例

以下语句返回值 2:

```
SELECT REMAINDER( 5, 3 ) FROM iq_dummy
```

### 另请参见

- MOD 函数 [数值] (第 244 页)

## REPEAT 函数 [字符串]

以指定的次数连接字符串。

### 语法

```
REPEAT ( string-expression, integer-expression )
```

### 参数

参数	描述
<i>string-expression</i>	要重复的字符串。
<i>integer-expression</i>	字符串的重复次数。如果 <i>integer-expression</i> 是负数，则返回空字符串。

### 返回

LONG VARCHAR

LONG NVARCHAR

**注意:** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **REPEAT**，则您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **REPEAT** 设置为正确的数据类型和大小。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持，但 **REPLICATE** 提供了相同的功能。

### 示例

以下语句返回值 "repeatrepeatrepeat":

```
SELECT REPEAT( 'repeat', 3 ) FROM iq_dummy
```

另请参见

- REPLACE 函数 [字符串] (第 286 页)
- REPLICATE 函数 [字符串] (第 288 页)

## REPLACE 函数 [字符串]

用另一个子串替换在各个位置出现的某个子串。

语法

```
REPLACE ( original-string, search-string, replace-string )
```

参数

如果有参数为空值，此函数返回空值。

参数	描述
original-string	待搜索的字符串。此字符串可以为任意长度。
search-string	要搜索并以 <i>replace-string</i> 替换的字符串。此字符串的长度不应超过 255 个字节。如果 <i>search-string</i> 是空字符串，则按原样返回原始字符串。
replace-string	替代字符串，用于替换 <i>search-string</i> 。可为任意长度。如果 <i>replace-string</i> 为空字符串，则删除出现的所有 <i>search-string</i> 。

返回

LONG VARCHAR

LONG NVARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **REPLACE**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **REPLACE** 设置为正确的数据类型和大小。

---

注释

**REPLACE** 函数的结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **REPLACE**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **REPLACE** 设置为正确的数据类型和大小。

有两种方法可以解决这一问题：

- 声明一个局部临时表，然后执行 **INSERT**：

```
DECLARE local temporary table #mytable
  (name column char(10)) on commit preserve rows;
INSERT INTO #mytable SELECT REPLACE(name,'0','1') FROM
dummy_table01;
```

- 使用 **CAST**：

```
SELECT CAST(replace(name, '0', '1') AS Char(10)) into #mytable
from dummy_table01;
```

如果需要在 *replace-string* 宽于 *search-string* 时控制结果列的宽度，可使用 **CAST** 函数。例如：

```
CREATE TABLE aa(a CHAR(5));
INSERT INTO aa VALUES( 'CCCC' );
COMMIT;
SELECT a, CAST(REPLACE(a,'C','ZZ') AS CHAR(5)) FROM aa;
```

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的服务商扩充。
- Sybase - 与 Adaptive Server Enterprise 兼容。

### 示例

以下语句返回值 "xx.def.xx.ghi"：

```
SELECT REPLACE( 'abc.def.abc.ghi', 'abc', 'xx' ) FROM iq_dummy
```

以下语句生成包含 **ALTER PROCEDURE** 语句的结果集，这些语句在执行时会修复那些引用已重命名的表的存储过程。（表名必须唯一才有用。）

```
SELECT REPLACE(
    replace(proc_defn,'OldTableName','NewTableName'),
    'create procedure',
    'alter procedure')
FROM SYS.SYSPROCEDURE
WHERE proc_defn LIKE '%OldTableName%'
```

对于 **LIST** 函数，使用逗号以外的分隔符：

```
SELECT REPLACE( list( table_id ), ',', '--')
FROM SYS.ISYSTAB
WHERE table_id <= 5
```

### 另请参见

- **CAST** 函数 [数据类型转换]（第 143 页）
- **CONVERT** 函数 [数据类型转换]（第 153 页）
- **HOURS** 函数 [日期和时间]（第 206 页）
- **MINUTES** 函数 [日期和时间]（第 242 页）
- **MONTHS** 函数 [日期和时间]（第 246 页）
- **SECOND** 函数 [日期和时间]（第 296 页）
- **WEEKS** 函数 [日期和时间]（第 336 页）
- **YEAR** 函数 [日期和时间]（第 341 页）
- **YEARS** 函数 [日期和时间]（第 342 页）
- **LCASE** 函数 [字符串]（第 225 页）
- **LEFT** 函数 [字符串]（第 228 页）
- **LOWER** 函数 [字符串]（第 237 页）
- **REVERSE** 函数 [字符串]（第 289 页）

## SQL 函数

- RIGHT 函数 [字符串] (第 290 页)
- UCASE 函数 [字符串] (第 327 页)
- UPPER 函数 [字符串] (第 328 页)
- REPEAT 函数 [字符串] (第 285 页)
- REPLICATE 函数 [字符串] (第 288 页)

### **REPLICATE 函数 [字符串]**

以指定的次数连接字符串。

#### 语法

```
REPLICATE ( string-expression, integer-expression )
```

#### 参数

参数	描述
string-expression	要重复的字符串。
integer-expression	字符串的重复次数。

#### 返回

LONG VARCHAR

LONG NVARCHAR

---

**注意:** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **REPLICATE**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **REPLICATE** 设置为正确的数据类型和大小。

---

#### 注释

**REPLICATE** 与 **REPEAT** 函数相同。

---

**注意:** **REPLICATE** 函数的结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **REPLICATE**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **REPLICATE** 设置为正确的数据类型和大小。

---

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

#### 示例

以下语句返回值 "repeatrepeatrepeat":

```
SELECT REPLICATE( 'repeat', 3 ) FROM iq_dummy
```

另请参见

- REPEAT 函数 [字符串] (第 285 页)
- REPLACE 函数 [字符串] (第 286 页)

## **REVERSE 函数 [字符串]**

采用一个参数作为类型 BINARY 或 STRING 的输入，并返回字符以反向顺序列出的指定字符串。

语法

```
REVERSE ( expression | uchar_expr )
```

参数

参数	描述
expression	类型为 CHAR、VARCHAR、NCHAR、NVARCHAR、BINARY 或 VARBINARY 的字符类型或二进制类型列名表达式、变量表达式或常量表达式。

返回

LONG VARCHAR

LONG NVARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **REVERSE**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **REVERSE** 设置为正确的数据类型和大小。

---

注释

- **REVERSE** 是一个字符串函数，它返回 expression 的逆序形式。
- 如果表达式为空，则逆序返回空。
- 代理对被视为不可拆分，因此不能逆序。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。

示例

```
select reverse("abcd")
-----
dcba
```

```
select reverse(0x12345000)
-----
0x00503412
```

另请参见

- 字符串运算符 (第 31 页)
- LCASE 函数 [字符串] (第 225 页)
- LEFT 函数 [字符串] (第 228 页)
- LOWER 函数 [字符串] (第 237 页)
- REPLACE 函数 [字符串] (第 286 页)
- RIGHT 函数 [字符串] (第 290 页)
- UCASE 函数 [字符串] (第 327 页)
- UPPER 函数 [字符串] (第 328 页)

**RIGHT 函数 [字符串]**

返回字符串中最右边的字符。

语法

**RIGHT** ( *string-expression*, *numeric-expression* )

参数

参数	描述
string-expression	要从左边截断的字符串。
numeric-expression	要返回的从字符串结尾处开始的字符数。

返回

LONG VARCHAR

LONG NVARCHAR

**注意:** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **RIGHT**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **RIGHT** 设置为正确的数据类型和大小。

注释

如果字符串包含多字节字符，并且使用了适当的归类，则返回的字节数可能大于指定的字符数。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

示例

以下语句返回值 "olate":

```
SELECT RIGHT( 'chocolate', 5 ) FROM iq_dummy
```

**另请参见**

- LCASE 函数 [字符串] (第 225 页)
- LEFT 函数 [字符串] (第 228 页)
- LOWER 函数 [字符串] (第 237 页)
- REPLACE 函数 [字符串] (第 286 页)
- REVERSE 函数 [字符串] (第 289 页)
- UCASE 函数 [字符串] (第 327 页)
- UPPER 函数 [字符串] (第 328 页)

**ROUND 函数 [数值]**

将 *numeric-expression* 舍入到小数点后指定的 *integer-expression* 位数。

**语法**

```
ROUND ( numeric-expression, integer-expression )
```

**参数**

参数	描述
<i>numeric-expression</i>	传递给函数的待舍入数字。
<i>integer-expression</i>	正整数指定小数点右边舍入到的有效位数。负数表达式指定小数点左边舍入到的有效位数。

**返回**

NUMERIC

如果将 ROUND\_TO\_EVEN 数据库选项设置为 ON，**ROUND** 函数会对 SAP Sybase IQ 表中的数据取整到最接近 *integer-expression* 的偶数，以便与 SQL Anywhere 表数据的行为匹配。如果将此选项设置为 OFF，**ROUND** 函数会对 SAP Sybase IQ 数据四舍五入。

**标准和兼容性**

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

**示例**

以下语句返回值 123.200:

```
SELECT ROUND( 123.234, 1 ) FROM iq_dummy
```

下表中显示了 **ROUND** 函数的其它结果:

值	ROUND (值)
123.4567	round (a.n,4)

值	ROUND (值)
123.4570	round (a.n,3)
123.4600	round (a.n,2)
123.5000	round (a.n,1)
123.0000	round (a.n, 0)
120.0000	round (a.n, -1)
100.0000	round (a.n, -2)
0.0000	round (a.n, -3)

在以下示例中，ROUND\_TO\_EVEN 设置影响返回值。

ROUND (值)	ROUND_TO_EVEN ON	ROUND_TO_EVEN OFF	注意
ROUND (convert (double, 123.45001), 1)	123.5	123.5	0.05001 大于 0.1 的一半
ROUND (convert (double, 123.45000), 1)	123.4	123.5	0.0500 是 0.1 的一半

另请参见

- TRUNCNUM 函数 [数值] (第 326 页)

## ROW\_NUMBER 函数 [分析]

一个排名函数，它为窗口分区中的每一行返回一个唯一行号，以便在每个窗口分区的开始位置对行重新开始编号。

如果窗口分区不存在，则该函数按从 1 到表基数的顺序对结果集中的行进行编号。

语法

```
ROW_NUMBER () OVER ([PARTITION BY window partition] ORDER BY window ordering)
```

参数

参数	描述
window partition	(可选) 一个或多个用逗号分隔的值表达式，表示您希望如何分隔一组结果行。



参数	描述
window ordering	如果指定窗口分区，则定义用于对窗口分区中的行进行排序的表达式；如果未指定窗口分区，则定义用于对结果集中的行进行排序的表达式。

### 注释

**ROW\_NUMBER** 函数需要 **OVER (ORDER\_BY)** 窗口规范。**OVER (ORDER\_BY)** 子句中的窗口分区子句是可选的。**OVER (ORDER\_BY)** 子句不能包含窗口构架 **ROWS/RANGE** 规范。

### 标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T611。

### 示例

以下示例返回雇员表中的薪水数据，按部门 ID 对结果集进行分区，并根据雇用开始日期对数据进行排序。**ROW\_NUMBER** 函数为每一行指派一个行号，并对每个窗口分区中的行重新开始编号：

```
SELECT DepartmentID dID, StartDate, Salary,
ROW_NUMBER()OVER(PARTITION BY dID ORDER BY StartDate) FROM Employees
ORDER BY 1,2;
```

返回的结果集为：

dID	StartDate	Salary	Row_number()
=====	=====	=====	=====
100	1986-10-14	42,998.000	1
100	1987-07-23	39,875.500	2
100	1988-03-23	37,400.000	3
100	1989-04-20	42,500.000	4
100	1990-01-15	42,100.000	5
200	1985-02-03	38,500.000	1
200	1987-02-19	39,300.000	2
200	1988-11-22	39,800.000	3
200	1989-06-01	34,892.000	4
200	1990-05-13	33,890.000	5
200	1990-07-11	37,803.000	6

## ROWID 函数 [杂项]

返回表的每一行的内部行 ID 值。

### 语法

```
ROWID ( table-name ) ...FROM table-name
```

参数

参数	描述
table-name	表的名称。指定括号内不带引号或带双引号的表名。

返回

UNSIGNED BIGINT

注释

您可以结合使用 **ROWID** 函数与其它子句来操作表的特定行。  
必须指定 **FROM table-name** 子句。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

示例

以下语句返回行 ID 值 1 到 10:

```
SELECT ROWID( "PRODUCTS" ) FROM PRODUCTS
```

rowid(Products)
1
2
3
.
.
.
10

以下语句返回产品 ID 值小于 400 的所有行的产品 ID 和行 ID 值:

```
SELECT PRODUCTS.ID, ROWID ( PRODUCTS )  
FROM PRODUCTS  
WHERE PRODUCTS.ID < 400
```

ID	rowid(Products)
300	1

ID	rowid(Products)
301	2
302	3

以下语句删除行 ID 值大于 50 的所有行：

```
DELETE FROM PRODUCTS
WHERE ROWID ( PRODUCTS ) > 50
```

## RTRIM 函数 [字符串]

返回尾随空白已删除的字符串。

*语法*

```
RTRIM ( string-expression )
```

*参数*

参数	描述
string-expression	要剪裁的字符串。

*返回*

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **RTRIM**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **RTRIM** 设置为正确的数据类型和大小。

---

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

以下语句返回字符串 "Test Message"，其中的所有末尾空格都已删除。

```
SELECT RTRIM( 'Test Message      ' ) FROM iq_dummy
```

**另请参见**

- LTRIM 函数 [字符串] (第 238 页)

**SECOND 函数 [日期和时间]**

返回 0 到 59 之间的一个数字，该数字对应于给定日期/时间值的秒部分。

*语法*

```
SECOND ( datetime-expression )
```

*参数*

参数	描述
<i>datetime-expression</i>	日期/时间值。

*返回*

SMALLINT

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

以下语句返回值 5:

```
SELECT SECOND ( '1998-07-13 08:21:05' ) FROM iq_dummy
```

**另请参见**

- CAST 函数 [数据类型转换] (第 143 页)
- CONVERT 函数 [数据类型转换] (第 153 页)
- HOURS 函数 [日期和时间] (第 206 页)
- MINUTES 函数 [日期和时间] (第 242 页)
- MONTHS 函数 [日期和时间] (第 246 页)
- REPLACE 函数 [字符串] (第 286 页)
- WEEKS 函数 [日期和时间] (第 336 页)
- YEAR 函数 [日期和时间] (第 341 页)
- YEARS 函数 [日期和时间] (第 342 页)

## SECONDS 函数 [日期和时间]

返回自任意开始日期和时间以来的秒数，两个时间之间的秒数，或者向某时间添加整数数量的秒。

### 语法

```
SECONDS ( datetime-expression
| datetime-expression, datetime-expression
| datetime-expression, integer-expression )
```

### 参数

参数	描述
datetime-expression	日期和时间。
integer-expression	要添加到 datetime-expression 中的秒数。如果 integer-expression 是负数，则从日期/时间值中减去相应的分钟数。如果提供整数表达式，则必须将 datetime-expression 显式转换为 datetime 数据类型。

### 返回

INTEGER

TIMESTAMP

### 注释

第二个语法返回从第一个日期/时间到第二个日期/时间之间的完整秒数。数值可能为负。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回值 3600:

```
SELECT ( SECONDS( '1998-07-13 06:07:12' ) -
SECONDS( '1998-07-13 05:07:12' ) ) FROM iq_dummy
```

以下语句返回值 14400，表示两个时间之间的差值:

```
SELECT SECONDS( '1999-07-13 06:07:12',
'1999-07-13 10:07:12' ) FROM iq_dummy
```

以下语句返回日期时间值 1999-05-12 21:05:12.000:

```
SELECT SECONDS( CAST( '1999-05-12 21:05:07'
AS TIMESTAMP ), 5) FROM iq_dummy
```

**SIGN 函数 [数值]**

返回一个数字的符号。

语法

```
SIGN ( numeric-expression )
```

参数

参数	描述
numeric-expression	要返回其符号的数字。

返回

SMALLINT

注释

对于负数，**SIGN** 返回 -1。

对于零，**SIGN** 返回 0。

对于正数，**SIGN** 返回 1。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

示例

以下语句返回值 -1:

```
SELECT SIGN( -550 ) FROM iq_dummy
```

**SIMILAR 函数 [字符串]**

返回 0 和 100 之间的一个整数，该整数表示两个字符串之间的相似性。

语法

```
SIMILAR ( string-expression1, string-expression2 )
```

参数

参数	描述
string-expression1	要比较的第一个字符串。
string-expression2	要比较的第二个字符串。

*返回***SMALLINT***注释*

此函数返回 0 和 100 之间的一个整数，该整数表示两个字符串之间的相似性。结果可以被解释成两个字符串之间匹配字符的百分比。值 100 表示两个字符串相同。

此函数可用于校正名称（如客户）列表。某些客户可能以稍微不同的名称多次添加到列表中。将表与其自身连接，生成一个关于所有大于 90% 但小于 100% 的相似性的报告。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句返回值 75:

```
SELECT SIMILAR( 'toast', 'coast' ) FROM iq_dummy
```

这表示两个值的相似度为 75%。

**SIN 函数 [数值]**

返回一个数字的正弦值，以弧度表示。

*语法*

```
SIN ( numeric-expression )
```

*参数*

参数	描述
numeric-expression	角度（以弧度为单位）。

*返回***DOUBLE***标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

以下语句返回值 0.496880:

```
SELECT SIN( 0.52 ) FROM iq_dummy
```

## 另请参见

- COS 函数 [数值] (第 158 页)
- ATAN2 函数 [数值] (第 137 页)
- ATAN 函数 [数值] (第 137 页)
- ASIN 函数 [数值] (第 136 页)
- ACOS 函数 [数值] (第 133 页)
- COT 函数 [数值] (第 159 页)
- TAN 函数 [数值] (第 324 页)

**SORTKEY 函数 [字符串]**

生成可用于根据替代归类规则对字符串进行排序的值。

## 语法

```
SORTKEY ( string-expression
[, { collation-id
| collation-name [(collation-tailoring-string)] } ]
)
```

## 参数

参数	描述
string-expression	<p>字符串表达式必须包含以数据库字符集编码的字符，且数据类型必须是 <b>STRING</b>。</p> <p>如果 <b>string-expression</b> 为空值，则 <b>SORTKEY</b> 函数返回空值。空字符串的排序顺序值不同于数据库列中的空值字符串。</p> <p><b>SORTKEY</b> 函数可处理的输入字符串没有长度限制。<b>SORTKEY</b> 的结果始终限制为 1024 字节和 <b>VARBINARY</b> 数据类型。如果实际结果超过 1024 字节，则只包含前 1024 字节。</p>
collation-name	<p>一个字符串或字符变量，用于指定要使用的排序顺序的名称。您也可以指定别名 <b>char_collation</b> 或 <b>db_collation</b> (两者等效) 以生成与数据库正在使用的 <b>CHAR</b> 归类所用排序键相同的排序键。</p> <p>类似地，可以指定别名 <b>NCHAR_COLLATION</b> 以生成数据库正在使用的 <b>NCHAR</b> 归类所使用的排序键。但是，<b>SAP Sybase IQ</b> 不支持对 <b>SAP Sybase IQ</b> 的特定对象使用 <b>NCHAR_COLLATION</b>。支持对 <b>SAP Sybase IQ</b> 服务器上的 <b>SQL Anywhere</b> 对象使用 <b>NCHAR_COLLATION</b>。</p>
collation-id	<p>一个变量、整数常量或字符串，用于指定要使用的排序顺序的 ID 号。此参数仅适用于按照对应的归类 ID 进行引用的 <b>Adaptive Server</b> 归类。</p>



参数	描述
collation-tailoring-string	<p>(可选) 指定归类定制选项 (<i>collation-tailoring-string</i>) 以便对字符的排序和比较进行进一步控制。这些选项采用跟在归类名称后, 放在括号内的关键字=值对形式。例如,</p> <pre>'UCA(locale=es;case=LowerFirst;accent=respect)'</pre> <p>指定这些选项的语法与 <b>CREATE DATABASE</b> 语句的 <b>COLLATION</b> 子句相同。.</p> <p><b>注意:</b> 指定 Unicode 归类算法 (UCA) 归类时, SQL Anywhere 数据库支持所有归类定制选项。对于所有其它归类, 仅支持区分大小写定制。</p>

### 返回

BINARY

### 注释

**SORTKEY** 函数生成的值可用于根据预定义的排序顺序行为对结果进行排序。这样, 您可以使用数据库归类可能并未提供的字符排序顺序行为。返回值是一个二进制值, 包含通过 **SORTKEY** 函数保留的输入字符串的已编码排序顺序信息。

例如, 您可以将 **SORTKEY** 函数返回的值存储在含源字符串的列中。下面的 **SELECT** 语句根据泰语字典按 **c1** 排序顺序从表 **T1** 中检索数据:

```
SELECT rid, c1 from T1 ORDER BY SORTKEY(c1)
```

您改而将 **SORTKEY** 返回的值存储在含源字符串的列中。若要按所需的顺序检索字符串数据, 在包含 **SORTKEY** 函数运行结果的列中 **SELECT** 语句只需包含一个 **ORDER BY** 子句:

```
UPDATE T1 SET shadowc1=SORTKEY(c1) FROM T1; SELECT rid, c1 FROM T1 ORDER BY shadowc1
```

**SORTKEY** 函数保证它为一组给定的排序顺序条件返回的值适用于对 **VARBINARY** 数据类型执行的二进制比较。

为查询生成排序键的成本可能很高。作为频繁请求排序键的替代方法, 可以考虑创建一个保存排序键值的计算列, 然后在查询的 **ORDER BY** 子句中引用该列。

如果不指定归类名称或归类 ID, 缺省设置为缺省 Unicode 多语言。

有效归类如下:

- 要查看按标签列出的 SAP Sybase IQ 所支持的归类, 请执行 `iqinit -l`。
- 下列表列出了 Adaptive Server 归类。

说明	归类名	归类 ID
缺省 Unicode 多语言	default	0
CP 850 方案: 不区分变音	altnoacc	39

## SQL 函数

说明	归类名	归类 ID
CP 850 方案: 小写优先	altdict	45
CP 850 西欧: 不区分大小写, 优先	altnocsp	46
CP 850 斯堪的纳维亚语字典	scandict	47
CP 850 斯堪的纳维亚语: 不区分大小写, 优先	scannocp	48
GB 拼音	gbpinyin	无
二进制排序	binary	50
拉丁语-1 英语、法语、德语字典	dict	51
拉丁语-1 英语、法语、德语不区分大小写	nocase	52
拉丁语-1 英语、法语、德语不区分大小写, 优先	nocasep	53
拉丁语-1 英语、法语、德语不区分变音	noaccent	54
拉丁语-1 西班牙语字典	espdict	55
拉丁语-1 西班牙语不区分大小写	espnocs	56
拉丁语-1 西班牙语不区分变音	espnoac	57
ISO 8859-5 俄语字典	rusdict	58
ISO 8859-5 俄语不区分大小写	rusnocs	59
ISO 8859-5 古斯拉夫语字典	cyrdict	63
ISO 8859-5 古斯拉夫语不区分大小写	cyrnocs	64
ISO 8859-7 希腊语字典	elldict	65
ISO 8859-2 匈牙利语字典	hundict	69
ISO 8859-2 匈牙利语不区分变音	hunnoac	70
ISO 8859-2 匈牙利语不区分大小写	hunnoacs	71
ISO 8859-5 土耳其语字典	turdict	72
ISO 8859-5 土耳其语不区分变音	turnoac	73
ISO 8859-5 土耳其语不区分大小写	turnocs	74
CP 874 (TIS 620) 皇家泰语字典	thaidict	1
ISO 14651 排序标准	14651	22

说明	归类名	归类 ID
Shift-JIS 二进制顺序	sjisbin	179
Unicode UTF-8 二进制排序	utf8bin	24
EUC JIS 二进制顺序	eucjisbn	192
GB2312 二进制顺序	gb2312bn	137
CP932 MS 二进制顺序	cp932bin	129
Big5 二进制顺序	big5bin	194
EUC KSC 二进制顺序	euckscbn	161

对于归类定制，创建排序键时通常以实现所有区分为目的，因此在指定非 UCA 归类时，所用的缺省定制等效于 `case=Respect`。例如，下面两个语句是等效的：

```
SELECT SORTKEY( 'abc', '1252LATIN1' ); SELECT SORTKEY( 'abc',
'1252LATIN1(case=Respect)' );
```

在指定非 UCA 归类时，缺省情况下，归类定制区分变音和大小写。但对于非 UCA 归类，使用归类定制只能覆盖区分大小写的设置。例如：

```
SELECT SORTKEY( 'abc', '1252LATIN1(case=LowerFirst)' );
```

如果创建数据库时未指定定制选项，即使为 **SORTKEY** 函数指定了数据库归类名称，下面两个子句也可能生成不同的排序顺序：

```
ORDER BY string-expression
```

```
ORDER BY SORTKEY( string-expression, database-collation-name )
```

系统可能生成不同的排序顺序，因为用于数据库创建和 **SORTKEY** 函数的缺省定制设置不同。要使 **SORTKEY** 获得与数据库归类相同的行为，可以为 *collation-tailoring-string* 提供与数据库归类设置匹配的定制语法，也可以为 `collation-name` 指定 `db_collation`。例如：

```
SORTKEY( expression, 'db_collation' )
```

**注意：** 使用 15.0 之前版本的 SAP Sybase IQ 创建的排序键值与使用 15.0 及更高版本创建的值不同。如果在 15.0 之前版本的数据库中存储了排序键值，则应用程序可能会出现问題，尤其是应用程序需要进行排序键值比较时。应在数据库中重新生成使用 15.0 之前版本的 SAP Sybase IQ 生成的任何排序键值。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

### 示例

以下语句查询 **Employees** 表并返回所有雇员的 **FirstName** 和 **Surname**，按 **Surname** 列的排序键值排序，使用 **dict** 归类（拉丁文-1、英语、法语、德语字典）：

```
SELECT Surname, GivenName FROM Employees ORDER BY SORTKEY( Surname, 'dict' );
```

### **SOUNDEX 函数 [字符串]**

返回表示字符串声音的数字。

语法

```
SOUNDEX ( string-expression )
```

参数

参数	描述
string-expression	字符串。

返回

**SMALLINT**

注释

字符串的 **SOUNDEX** 函数值基于第一个字母和随后三个除 **H**、**Y** 和 **W** 之外的辅音。两个重复的字母按一个字母计算。例如：

```
SOUNDEX( 'apples' ) FROM iq_dummy
```

基于字母 **A**、**P**、**L** 和 **S**。

**SOUNDEX** 函数会忽略多字节字符。

尽管不很完善，但 **SOUNDEX** 对于那些听起来相似以及以相同字母开头的单词，通常返回相同的数字。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server 兼容，但是 Adaptive Server 会返回 **CHAR(4)** 类型的结果，而 SAP Sybase IQ 会返回一个整数。

示例

以下语句返回两个数字，分别表示每个名称的声音。每个参数的 **SOUNDEX** 值为 3827。

```
SELECT SOUNDEX( 'Smith' ), SOUNDEX( 'Smythe' ) FROM iq_dummy
```

**SOUNDEX** ( 'Smith' ) 等同于 **SOUNDEX** ( 'Smythe' )。

另请参见

- **DIFFERENCE** 函数 [字符串] (第 188 页)

## SP\_HAS\_ROLE 函数 [系统]

返回一个整数值，表示是为调用用户授予了指定系统特权还是用户定义的角色。当用于对用户定义的存储过程执行特权检查时，**SP\_HAS\_ROLE** 会在用户特权检查失败时返回一条错误消息。

### 语法

```
dbo.sp_has_role( [rolename], [grant_type], [throw_error] )
```

### 参数

参数	描述
rolename	系统特权或用户定义角色的名称。
grant_type	有效值包括：ADMIN 和 NO ADMIN。如果为 NULL 或未指定，则缺省使用 NO ADMIN。
throw_error	有效值包括： <ul style="list-style-type: none"> <li>“1” - 如果没有为调用用户授予指定系统特权或用户定义角色，则显示错误消息。</li> <li>“0” - （缺省值）如果没有为调用用户授予指定系统特权或用户定义角色，将不显示错误消息。</li> </ul>

### 返回值

值	描述
1	已为调用用户授予系统特权或用户定义角色。
0 或权限被拒绝： 您没有执行此命令/过程的权限。	没有为调用用户授予系统特权或用户定义角色。throw_error 参数设置为 1 时，将返回错误消息以代替值 0。
-1	指定的系统特权或用户定义角色不存在。即使 throw_error 参数设置为 1，也不显示任何错误消息。

### 注释

如果 grant\_type 参数的值为 ADMIN，该函数将检查调用用户是否具有系统特权的管理特权。如果 grant\_type 参数的值为 NO ADMIN，该函数将检查调用用户是否具有对系统特权或角色的使用特权。

如果未指定 grant\_type 参数，则缺省使用 NO ADMIN，且输出仅指示是为调用用户授予（直接或间接）了指定的系统特权还是用户定义角色。

如果 `rolename` 和 `grant_type` 参数都为 `NULL`，且 `throw_error` 参数为 `1`，则显示错误消息。对于从目录表读取特定值而不是检查是否存在调用用户的系统特权之后出现错误消息的存储过程来说，此功能非常有用。

---

**注意：** 如果参数 `rolename` 和 `grant_type` 设置为 `NULL` 且 `throw_error` 设置为 `1`，或三个参数全部设置为 `NULL`，则返回权限被拒绝的错误消息。

---

### 示例

假定以下情形：

- 已使用 `WITH NO ADMIN OPTION` 子句授予 `u1` `CREATE ANY PROCEDURE` 系统特权。
- 已授予 `u1` `CREATE ANY TABLE` 系统特权。
- 已使用 `WITH ADMIN ONLY OPTION` 子句授予 `u1` 用户定义角色 `Role_A`。
- `Role_B` 存在，但未被授予 `u1`
- 角色 `Role_C` 不存在。

基于以上情形，以下命令

- `sp_has_role 'create any procedure'`

返回值 `1`，表示已授予 `u1` `CREATE ANY PROCEDURE` 系统特权。

- `sp_has_role 'create any table'`

返回值 `0`，表示未授予 `u1` `CREATE ANY TABLE` 系统特权。未返回错误消息，因为未指定 `throw_error` 参数。

- `sp_has_role 'create any procedure','admin',1`

返回 `Permission denied` 错误消息 (`throw_error=1`)。即使已授予 `u1` `CREATE ANY PROCEDURE` 系统特权，但未授予 `u1` 对系统特权的管理权限。

- `sp_has_role 'Role_A'`

返回值 `1`，表示已授予 `u1` 角色 `Role_A`。

- `sp_has_role 'Role_A','admin',1`

返回值 `1`，表示已授予 `u1` 具有管理权限的角色 `Role_A`。

- `sp_has_role 'Role_B'`

返回值 `0`，表示未授予 `u1` 角色 `ROLE_B`。未返回错误消息，因为未指定 `throw_error` 参数。

- `sp_has_role 'Role_C'`

返回值 `-1`，表示角色 `ROLE_C` 不存在。

- `sp_has_role 'Role_C',NULL,1`

返回值 `-1`，表示角色 `ROLE_C` 不存在。

## **SPACE 函数 [字符串]**

返回指定数目的空格。

语法

```
SPACE ( integer-expression )
```

参数

参数	描述
<i>integer-expression</i>	要返回的空格数。

返回

**LONG VARCHAR**

**注意：** 结果数据类型为 **LONG VARCHAR**。如果在 **SELECT INTO** 语句中使用 **SPACE**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **SPACE** 设置为正确的数据类型和大小。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

示例

以下语句返回包含 10 个空格的字符串：

```
SELECT SPACE( 10 ) FROM iq_dummy
```

## **SQLFLAGGER 函数 [杂项]**

返回给定 SQL 语句与指定标准之间的偏差。

语法

```
SQLFLAGGER ( sql-standard-string, sql-statement-string )
```

## 参数

参数	描述
sql-standard-string	<p>测试遵从性所使用的标准级别。可能的值与 <code>SQL_FLAGGER_ERROR_LEVEL</code> 数据库选项的值相同：</p> <ul style="list-style-type: none"> <li>• SQL:2003/Core 测试核心 SQL/2003 语法的一致性。</li> <li>• SQL:2003/Package 测试完整 SQL/2003 语法的一致性。</li> <li>• SQL:1999/Core 测试核心 SQL/1999 语法的一致性。</li> <li>• SQL:1999/Package 测试完整 SQL/1999 语法的一致性。</li> <li>• SQL:1992/Entry 测试初级 SQL/1992 语法的一致性。</li> <li>• SQL:1992/Intermediate 测试中级 SQL/1992 语法的一致性。</li> <li>• SQL:1992/Full 测试完整 SQL/1992 语法的一致性。</li> </ul>
sql-statement-string	用于检查一致性的 SQL 语句。

## 返回

LONG VARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **SQLFLAGGER**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **SQLFLAGGER** 设置为正确的数据类型和大小。

---

## 注释

您也可以使用 `iqsqlpp` SQL 预处理器实用程序标记不属于 SQL92 指定集合的任意嵌入式 SQL。请参见实用程序指南中的 `iqsqlpp` SQL 预处理器实用程序。

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

## 示例

以下语句显示发现不允许的扩充时返回的消息的示例：

```
SELECT SQLFLAGGER( 'SQL:2003/Package', 'SELECT top 1 dummy_col FROM
sys.dummy ORDER BY dummy_col' );
```

此语句返回消息 '0AW03 Disallowed language extension detected in syntax near 'top' on line 1'。

以下语句返回 '00000'，因为它不包含禁用扩展：



```
SELECT SQLFLAGGER( 'SQL:2003/Package', 'SELECT dummy_col FROM
sys.dummy' );
```

## SQRT 函数 [数值]

返回一个数字的平方根。

语法

```
SQRT ( numeric-expression )
```

参数

参数	描述
numeric-expression	要计算其平方根的数字。

返回

DOUBLE

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

示例

以下语句返回值 3:

```
SELECT SQRT( 9 ) FROM iq_dummy
```

## SQUARE 函数 [数值]

将指定表达式的平方以浮点数的形式返回。

语法

```
SQUARE ( numeric-expression )
```

参数

参数	描述
expression	是数据类型为精确数值、近似数值、货币（或任何可隐式转换为这些类型的类型）的列、变量或表达式。对于其它数据类型， <b>SQUARE</b> 函数会产生错误。返回值为 <b>DOUBLE</b> 数据类型。

注释

**SQUARE** 函数采用一个参数。例如，**SQUARE (12.01)** 返回 144.240100。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

## STDDEV 函数 [集合]

返回一组数字的标准偏差。

语法

```
STDDEV ( [ ALL ] expression )
```

参数

参数	描述
expression	任何数值数据类型 (FLOAT、REAL 或 DOUBLE 精度) 的表达式。

返回

DOUBLE

注释

用于计算 **STDDEV** 的公式是：

$$stddev = \sqrt{variance}$$

**STDDEV** 返回 DOUBLE 精度浮点数据类型的结果。如果应用于空集，则结果为 NULL，对于一个元素的输入集，它将返回 NULL。

**STDDEV** 不支持关键字 DISTINCT。如果将 DISTINCT 与 **STDDEV** 一起使用，则系统会返回语法错误。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

示例

假定有以下数据：

```
SELECT Salary FROM Employees WHERE DepartmentID = 300
```

Salary
51432.000
57090.000

Salary
42300.000
43700.00
36500.000
138948.000
31200.000
58930.00
75400.00

以下语句返回值 32617.8446712838471:

```
SELECT STDDEV ( Salary ) FROM Employees
WHERE DepartmentID = 300
```

假定有以下数据:

```
SELECT UnitPrice FROM Products WHERE Name = 'Tee Shirt'
```

Name	UnitPrice
Tee Shirt	9.00
Tee Shirt	14.00
Tee Shirt	14.00

以下语句返回值 2.88675134594813049:

```
SELECT STDDEV ( UnitPrice ) FROM Products
WHERE Name = 'Tee Shirt'
```

### 另请参见

- 窗口化集合函数的用法 (第 78 页)
- STDDEV\_SAMP 函数 [集合] (第 312 页)
- VARIANCE 函数 [集合] (第 334 页)

## **STDDEV\_POP 函数 [集合]**

计算由数值表达式组成的总体的标准偏差, 类型为 DOUBLE。

### 语法

```
STDDEV_POP ( [ ALL ] expression )
```

*参数*

参数	描述
expression	要对一组行计算其基于总体的标准偏差的表达式（通常是列名）。

*返回*

DOUBLE

*注释*

计算所提供的对每个组行或分区行求值的值表达式的总体标准偏差（如果指定了 **DISTINCT**，则为在消除重复项后剩余的各行），其定义为：总体方差的平方根。

$$\sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$$

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句列出不同时间段每个订单的产品数目的平均值和方差：

```
SELECT year( ship_date ) AS Year, quarter( ship_date )
      AS Quarter, AVG( quantity ) AS Average,
      STDDEV_POP ( quantity ) AS Variance
FROM SalesOrderItems GROUP BY Year, Quarter
ORDER BY Year, Quarter;
```

Year	Quarter	Average	Variance
2000	1	25.775148	14.2794
2000	2	27.050847	15.0270
...	...	...	...

*另请参见*

- 窗口化集合函数的用法（第 78 页）

**STDDEV\_SAMP 函数 [集合]**

计算由数值表达式组成的样本的标准偏差，类型为 DOUBLE。

*语法*

```
STDDEV_SAMP ( [ ALL ] expression )
```

## 参数

参数	描述
expression	要对一组行计算其基于样本的标准偏差的表达式（通常是列名）。

## 返回

DOUBLE

## 注释

**注意：** `STDDEV_SAMP` 是 `STDDEV` 的别名。

计算所提供的对每个组行或分区行求值的 *value expression* 的样本标准偏差（如果指定了 `DISTINCT`，则为在消除重复项后剩余的各行），其定义为：样本方差的平方根。

对于包含一个元素的输入集，空值将返回空值。

根据下面的公式计算标准偏差，其中假定正态分布：

$$\sqrt{\frac{\sum (x_i - \bar{x})^2}{(n-1)}}$$

## 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

## 示例

以下语句列出不同时间段每个订单的产品数目的平均值和方差：

```
SELECT year( ship_date ) AS Year, quarter( ship_date )
       AS Quarter, AVG( quantity ) AS Average,
       STDDEV_SAMP( quantity ) AS Variance
FROM SalesOrderItems GROUP BY Year, Quarter
ORDER BY Year, Quarter;
```

Year	Quarter	Average	Variance
2000	1	25.775148	14.3218
2000	2	27.050847	15.0696
...	...	...	...

## 另请参见

- 窗口化集合函数的用法（第 78 页）

## SQL 函数

- STDDEV 函数 [集合] (第 310 页)
- VARIANCE 函数 [集合] (第 334 页)

### **STR 函数 [字符串]**

返回一个数字的等效字符串。

#### 语法

```
STR ( numeric-expression [ , length [ , decimal ] ] )
```

#### 参数

参数	描述
numeric-expression	任何近似数值 (FLOAT、REAL 或 DOUBLE 精度) 的表达式。
length	要返回的字符数 (包括小数点、小数点右侧和左侧的所有数字、符号 (如果有) 以及空白)。缺省值为 10, 最大长度为 255。
decimal	要返回的小数点右边的位数。缺省值为 0。

#### 返回

VARCHAR

#### 注释

如果数字的整数部分不符合指定的长度, 则结果为空。例如, 下面的语句返回空值:

```
SELECT STR( 1234.56, 3 ) FROM iq_dummy
```

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

#### 示例

以下语句返回的字符串共有十个字符, 前面是六个空格, 后面是 1234:

```
SELECT STR( 1234.56 ) FROM iq_dummy
```

以下语句返回结果 1234.5:

```
SELECT STR( 1234.56, 6, 1 ) FROM iq_dummy
```

## STR\_REPLACE 函数 [字符串]

采用三个参数作为类型 BINARY 或 STRING 的输入，并将第一个字符串表达式 (*string\_expr1*) 中出现的第二个字符串表达式 (*string\_expr2*) 的所有实例替换为第三个表达式 (*string\_expr3*)。

### 语法

```
REPLACE ( string_expr1, string_expr2, string_expr3 )
```

### 参数

参数	描述
<i>string_expr1</i>	源字符串或要搜索的字符串表达式，表示为 CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY 或 BINARY 数据类型。
<i>string_expr2</i>	模式字符串或要在第一个表达式 ( <i>string_expr1</i> ) 中查找的字符串表达式，表示为 CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY 或 BINARY 数据类型。
<i>string_expr3</i>	替换字符串表达式，表示为 CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY 或 BINARY 数据类型。

### 注释

**STR\_REPLACE** 是 **REPLACE** 函数的别名。

- 采用任何数据类型作为输入并返回 **STRING** 或 **BINARY**。  
例如，在进一步评估之前，作为参数传递的空字符串 ("") 将替换为一个空格 (" ")。对于 BINARY 和 STRING 类型，这一点均适用。
- 所有参数都可以组合使用 BINARY 和 STRING 数据类型。
- 结果的长度可能会有所不同，具体取决于编译表达式时对参数值的了解情况。如果所有参数都是指派给常量的列或主机变量，则 SAP Sybase IQ 按如下方式计算结果长度：

```
result_length = ((s/p)*(r-p)+s)
WHERE
  s = length of source string
  p = length of pattern string
  r = length of replacement string
IF (r-p) <= 0, result length = s
```

- 如果 SAP Sybase IQ 因在编译表达式时参数值未知而无法计算结果长度，则使用结果长度 255。
- **RESULT\_LEN** 永远不会超过 32767。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。

#### 示例

将字符串 *cdefghi* 中的字符串 *def* 替换为 *yyy*。

```
select replace("cdefghi", "def", "yyy")
-----
cyyyghi
(1 row(s) affected)
```

将所有空格替换为 "toyota"

```
select str_replace ("chevy, ford, mercedes", "", "toyota")
-----
chevy,toyotaford,toyotamercedes
(1 row(s) affected)
```

接受第三个参数中的 NULL 并将其视为尝试用 NULL 替换 *string\_expr2*，有效地将 STR\_REPLACE 调整为“字符串剪切”操作。返回 "abcghijklm":

```
select str_replace("abcdefghijklm", "def", NULL)
-----
abcghijklm
(1 row affected)
```

### 另请参见

- BIT\_LENGTH 函数 [字符串] (第 140 页)
- BYTE\_LENGTH 函数 [字符串] (第 141 页)
- CHAR\_LENGTH 函数 [字符串] (第 147 页)
- COL\_LENGTH 函数 [系统] (第 150 页)
- DATALENGTH 函数 [系统] (第 164 页)
- LEN 函数 [字符串] (第 229 页)
- LENGTH 函数 [字符串] (第 230 页)
- OBJECT\_NAME 函数 [系统] (第 257 页)
- OCTET\_LENGTH 函数 [字符串] (第 258 页)

## **STRING 函数 [字符串]**

将一个或多个字符串连接为一个大字符串。

### 语法

```
STRING ( string-expression [ , ... ] )
```



### 参数

参数	描述
string-expression	一个字符串。如果只提供一个参数，则它将转换为单个表达式。如果提供多个参数，则会将它们连接为单个字符串。对于空值，将作为空字符串 ("") 处理。

### 返回

LONG VARCHAR

LONG NVARCHAR

LONG BINARY

**注意：**结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **STRING**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **STRING** 设置为正确的数据类型和大小。

### 注释

数字或日期参数在连接前转换为字符串。您也可以使用 **STRING** 函数将任何单个表达式转换为字符串，方法是提供该表达式作为唯一的参数。

如果所有参数都为空值，则 **STRING** 返回空值。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回值 testing123:

```
SELECT STRING( 'testing', NULL, 123 )
FROM iq_dummy
```

## **STRTOUUID 函数 [字符串]**

将字符串值转换为唯一标识符 (UUID 或 GUID) 值。

### 语法

```
STRTOUUID ( string-expression )
```

### 参数

参数	描述
string-expression	格式为 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 的字符串

*返回*

## UNIQUEIDENTIFIER

*注释*

将 `XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX` (其中 *x* 是十六进制数字) 格式的字符串转换为一个唯一标识符值。如果该字符串不是有效的 UUID 字符串, 则返回空值。

可以使用 **STRTOUUID**, 将 UUID 值插入 SAP Sybase IQ 数据库。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase - 不受 Adaptive Server 支持。

*示例*

```
CREATE TABLE T (
  pk uniqueidentifier primary key,
  c1 int);
INSERT INTO T (pk, c1)
VALUES (STRTOUUID
('12345678-1234-5678-9012-123456789012'), 1);
```

*另请参见*

- 二进制数据类型 (第 352 页)
- NEWID 函数 [杂项] (第 247 页)
- UUIDTOSTR 函数 [字符串] (第 331 页)
- 字符数据类型 (第 345 页)
- 二进制数据类型 (第 759 页)

**STUFF 函数 [字符串]**

从一个字符串中删除一些字符, 并用另一个字符串替换它们。

*语法*

```
STUFF ( string-expression1, start, length, string-expression2 )
```

*参数*

参数	描述
<code>string-expression1</code>	要用 <b>STUFF</b> 函数修改的字符串。
<code>start</code>	字符位置, 从此处开始删除字符。字符串中第一个字符是位置 1。

参数	描述
length	要删除的字符数。
string-expression2	要插入的字符串。若要使用 <b>STUFF</b> 函数删除字符串的一部分，请使用空值作为替换字符串。

*返回*

LONG NVARCHAR

*注释*

若要使用 **STUFF** 删除字符串的一部分，请使用空值作为替换字符串。若要使用 **STUFF** 插入字符串，请将长度设为零。

在以下情况下，**STUFF** 函数将返回空值结果：

- 前三个参数中的任一参数为空值。
- start 或 length 参数为负值。
- start 参数大于 string-expression1 的长度。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

以下语句返回值 "chocolate pie"：

```
SELECT STUFF( 'chocolate cake', 11, 4, 'pie' )
FROM iq_dummy
```

**SUBSTRING 函数 [字符串]**

返回字符串的子字符串。

*语法*

```
{ SUBSTRING | SUBSTR } ( string-expression, start [ , length ] )
```

*参数*

参数	描述
string-expression	从中返回子串的字符串。
start	要返回的子字符串的起始位置（以字符为单位）。负起始位置指定从字符串结尾处（而不是开始处）开始的字符数。字符串中第一个字符的位置是 1。

参数	描述
length	要返回的子字符串的长度（以字符为单位）。正 <i>length</i> 表示子字符串在起始位置的右侧 <i>length</i> 个字符处结束，而负 <i>length</i> 指定子字符串在起始位置的左侧 <i>length</i> 个字符处结束。

*返回*

LONG VARCHAR

LONG NVARCHAR

LONG BINARY

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **STRING**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **STRING** 设置为正确的数据类型和大小。

*注释*

如果指定了 *length*，则子字符串不应超过该长度。如果未指定长度，则从 *start* 位置开始返回字符串的其余部分。

*start* 和 *length* 都可以是负数。使用适当的负数和正数组合，可以从字符串的开始处或结束处获取子字符串。

如果您有权使用非结构化数据分析功能，则可以将此函数与大对象数据一起使用。

**ansi\_substring** 数据库选项设置为 ON（缺省）时，负值无效。

请参见《非结构化数据分析》中的 SUBSTRING 函数。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - **SUBSTR** 不受 Adaptive Server 支持。请改用 **SUBSTRING**

*示例*

以下语句返回 "back":

```
SELECT SUBSTRING ( 'back yard', 1 , 4 )
FROM iq_dummy
```

以下语句返回 yard:

```
SELECT SUBSTR ( 'back yard', -1 , -4 )
FROM iq_dummy
```

以下语句返回 0x2233:

```
SELECT SUBSTR ( 0x112233445566, 2, 2 )
FROM iq_dummy
```

另请参见

- CHARINDEX 函数 [字符串] (第 148 页)

### **ANSI SUBSTRING 选项 [TSQL]**

控制在为 start 或 length 参数提供负值时 SUBSTRING (SUBSTR) 函数的行为。

*允许值*

ON、OFF

*默认值*

ON

*范围*

可在数据库 (PUBLIC) 或用户级别设置选项。在数据库级别进行设置时，值将变为任何新用户的缺省值，但不会对现有用户产生任何影响。在用户级别进行设置时，仅替换该用户的 PUBLIC 值。为自身设置选项无需任何系统特权。在数据库或用户级别为任何其他用户设置选项都需要系统特权。

必须具有 SET ANY PUBLIC OPTION 系统特权才能设置此选项。可针对个别连接或 PUBLIC 角色进行临时设置。设置立即生效。

*注释*

**ANSI\_SUBSTRING** 选项设置为 ON 时，**SUBSTRING** 函数的行为相当于 ANSI/ISO SQL/2003 行为。如果起始偏移为负或零，则视为字符串左侧用非字符填补，并且在提供负值长度的情况下会出现错误。

当此选项设置为 OFF 时，**SUBSTRING** 函数的行为与先前版本的 SAP Sybase IQ 的行为相同：负的起始偏移表示从字符串末尾开始的偏移，而负的长度表示所需的子串在起始偏移左侧第 N 个字符处结束，其中 N 为长度。起始偏移 0 等效于起始偏移 1。

避免在 **SUBSTRING** 函数中使用非正起始偏移量或负长度。请尽可能改用 **LEFT** 或 **RIGHT** 函数。

*示例*

以下示例说明了基于不同 **ANSI\_SUBSTRING** 选项设置的不同 **SUBSTRING** 函数返回值：

```
SUBSTRING( 'abcdefgh', -2, 4 );
ansi_substring = Off ==> 'gh'
// substring starts at second-last character
ansi_substring = On ==> 'gh'
// takes the first 4 characters of
// ???abcdefgh and discards all ?
```

```
SUBSTRING( 'abcdefgh', 4, -2 );
ansi_substring = Off ==> 'cd'
ansi_substring = On ==> value -2 out of range
for destination
```

```
SUBSTRING( 'abcdefgh', 0, 4 );
ansi_substring = Off ==> 'abcd'
ansi_substring = On  ==> 'abcd'
```

## **SUBSTRING64 函数 [字符串]**

**SUBSTRING64** 函数返回大对象列或变量参数的可变长度字符串。

### *注释*

**SUBSTRING64** 支持搜索 LONG VARCHAR 和 LONG BINARY 列以及任意数据大小的 LONG VARCHAR 和 LONG BINARY 变量。目前，一个 SQL 变量可以容纳长度最大为 2GB - 1 的数据。

如果您有权使用非结构化数据分析功能，则可以将此函数与大对象数据一起使用。

请参见《非结构化数据分析》中的 SUBSTRING64 函数。

## **SUM 函数 [集合]**

返回每一组行的指定表达式总数。

### *语法*

```
SUM ( expression | DISTINCT column-name )
```

### *参数*

参数	描述
<i>expression</i>	要总计的对象。这通常是列名称。
<b>DISTINCT</b> <i>column-name</i>	计算每组行的 <i>column-name</i> 中唯一值的总和。其用途有限，但出于完整性考虑将其包括在内。

### *返回*

INTEGER

DOUBLE

NUMERIC

### *注释*

不包括指定表达式为空值的行。

对于不包含任何行的组返回空值。

### *标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

以下语句返回值 3749146.740:

```
SELECT SUM( salary )
FROM Employees
```

**另请参见**

- AVG 函数 [集合] (第 138 页)
- COUNT 函数 [集合] (第 162 页)
- 窗口化集合函数的用法 (第 78 页)

**SUSER\_ID 函数 [系统]**

返回整数用户标识号。

*语法*

```
SUSER_ID ( [ user-name ] )
```

*参数*

参数	描述
user-name	用户名。

*返回*

INT

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。

*示例*

以下语句返回用户标识号 1:

```
SELECT SUSER_ID ('DBA') FROM iq_dummy
```

以下语句返回用户标识号 0:

```
SELECT SUSER_ID ('SYS') FROM iq_dummy
```

**另请参见**

- SUSER\_NAME 函数 [系统] (第 324 页)
- USER\_ID 函数 [系统] (第 329 页)
- USER\_NAME 函数 [系统] (第 330 页)

**SUSER\_NAME 函数 [系统]**

返回用户名。

语法

```
SUSER_NAME ( [ user-id ] )
```

参数

参数	描述
user-id	用户标识号。

返回

LONG VARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **SUSER\_NAME**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **SUSER\_NAME** 设置为正确的数据类型和大小。

---

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。在 Adaptive Server 中，**SUSER\_NAME** 返回服务器用户名。

示例

以下语句返回值 DBA：

```
SELECT SUSER_NAME ( 1 ) FROM iq_dummy
```

以下语句返回值 SYS：

```
SELECT SUSER_NAME ( 0 ) FROM iq_dummy
```

另请参见

- SUSER\_ID 函数 [系统] (第 323 页)
- USER\_ID 函数 [系统] (第 329 页)
- USER\_NAME 函数 [系统] (第 330 页)

**TAN 函数 [数值]**

返回一个数字的正切值。

语法

```
TAN ( numeric-expression )
```



*参数*

参数	描述
numeric-expression	角度（以弧度为单位）。

*返回*

DOUBLE

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

*示例*

返回值 0.572561:

```
SELECT TAN( 0.52 ) FROM iq_dummy
```

*另请参见*

- COS 函数 [数值]（第 158 页）
- ATAN2 函数 [数值]（第 137 页）
- ATAN 函数 [数值]（第 137 页）
- ASIN 函数 [数值]（第 136 页）
- ACOS 函数 [数值]（第 133 页）
- COT 函数 [数值]（第 159 页）
- SIN 函数 [数值]（第 299 页）

**TODAY 函数 [日期和时间]**返回当前日期。这是 **CURRENT DATE** 的传统语法。*语法*

```
TODAY ( * )
```

*返回*

DATE

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句根据系统时钟返回当前日期。

```
SELECT TODAY( * ) FROM iq_dummy
```

## **TRIM 函数 [字符串]**

删除字符串中的前导和尾随空白。

*语法*

```
TRIM ( string-expression )
```

*参数*

参数	描述
string-expression	要剪裁的字符串。

*返回*

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **TRIM**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **TRIM** 设置为正确的数据类型和大小。

---

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。请改用 **LTRIM** 和 **RTRIM**。

*示例*

以下语句返回没有前导或尾随空白的值 "chocolate"。

```
SELECT TRIM( ' chocolate ' ) FROM iq_dummy
```

## **TRUNCNUM 函数 [数值]**

在小数点后的指定位数截断数字。

*语法*

```
TRUNCNUM ( numeric-expression, integer-expression )
```

### 参数

参数	描述
numeric-expression	要截断的数字。
integer-expression	正整数指定小数点右边舍入到的有效位数。负数表达式指定小数点左边舍入到的有效位数。

### 返回

NUMERIC

### 注释

此函数与 **TRUNCATE** 相同，但不会导致关键字冲突。

可以通过将 **ROUND**、**FLOOR** 和 **CEILING** 组合使用来提供类似的功能。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回值 600:

```
SELECT TRUNCNUM( 655, -2 ) FROM iq_dummy
```

以下语句返回值 655.340:

```
SELECT TRUNCNUM( 655.348, 2 ) FROM iq_dummy
```

### 另请参见

- **ROUND** 函数 [数值] (第 291 页)

## **UCASE** 函数 [字符串]

将字符串中的所有字符转换成大写形式。

### 语法

```
UCASE ( string-expression )
```

### 参数

参数	描述
string-expression	要转换成大写形式的字符串。

返回

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **UCASE**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **UCASE** 设置为正确的数据类型和大小。

---

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - Adaptive Server 不支持 **UCASE**，但是 **UPPER** 可通过兼容方式提供相同功能。

示例

以下语句返回值 "CHOCOLATE":

```
SELECT UCASE ( 'ChocoLate' ) FROM iq_dummy
```

另请参见

- LCASE 函数 [字符串] (第 225 页)
- LEFT 函数 [字符串] (第 228 页)
- LOWER 函数 [字符串] (第 237 页)
- REPLACE 函数 [字符串] (第 286 页)
- REVERSE 函数 [字符串] (第 289 页)
- RIGHT 函数 [字符串] (第 290 页)
- UPPER 函数 [字符串] (第 328 页)

## UPPER 函数 [字符串]

将字符串中的所有字符转换成大写形式。

语法

```
UPPER ( string-expression )
```

参数

参数	描述
string-expression	要转换成大写形式的字符串。

返回

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

---

**注意：** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **UPPER**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **UPPER** 设置为正确的数据类型和大小。

---

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 与 Adaptive Server Enterprise 兼容。

示例

以下语句返回值 "CHOCOLATE":

```
SELECT UPPER( 'ChocoLate' ) FROM iq_dummy
```

另请参见

- LCASE 函数 [字符串] (第 225 页)
- LEFT 函数 [字符串] (第 228 页)
- LOWER 函数 [字符串] (第 237 页)
- REPLACE 函数 [字符串] (第 286 页)
- REVERSE 函数 [字符串] (第 289 页)
- RIGHT 函数 [字符串] (第 290 页)
- UCASE 函数 [字符串] (第 327 页)

## **USER\_ID 函数 [系统]**

返回整数用户标识号。

语法

```
USER_ID ( [ user-name ] )
```

参数

参数	描述
user-name	用户名。

返回

INT

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。

示例

以下语句返回用户标识号 1:

```
SELECT USER_ID ('DBA') FROM iq_dummy
```

以下语句返回用户标识号 0:

```
SELECT USER_ID ('SYS') FROM iq_dummy
```

另请参见

- SUSER\_ID 函数 [系统] (第 323 页)
- SUSER\_NAME 函数 [系统] (第 324 页)
- USER\_NAME 函数 [系统] (第 330 页)

## USER\_NAME 函数 [系统]

返回用户名。

语法

```
USER_NAME ( [ user-id ] )
```

参数

参数	描述
user-id	用户标识号。

返回

LONG VARCHAR

---

**注意:** 结果数据类型为 LONG VARCHAR。如果在 **SELECT INTO** 语句中使用 **USER\_NAME**，您必须具有“非结构化数据分析选项”许可证，或使用 **CAST** 并将 **USER\_NAME** 设置为正确的数据类型和大小。

---

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

- Sybase - 为 SAP Sybase IQ 实现的 Adaptive Server 函数。在 Adaptive Server 中，USER\_NAME 返回用户名，而不是服务器用户名。

示例

以下语句返回值 "DBA":

```
SELECT USER_NAME ( 1 ) FROM iq_dummy
```

以下语句返回值 "SYS":

```
SELECT USER_NAME ( 0 ) FROM iq_dummy
```

另请参见

- SUSER\_ID 函数 [系统] (第 323 页)
- SUSER\_NAME 函数 [系统] (第 324 页)
- USER\_ID 函数 [系统] (第 329 页)

## UUIDTOSTR 函数 [字符串]

将唯一标识符值 (UUID, 又称 GUID) 转换为字符串值。

语法

```
UUIDTOSTR ( uuid-expression )
```

参数

表 52. 参数

参数	描述
uuid-expression	唯一标识符值。

返回

VARCHAR

注释

将唯一标识符转换为 `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` (其中 x 是十六进制数字) 格式的字符串值。如果此二进制值不是有效的唯一标识符, 则返回 NULL。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

示例

要将唯一标识符值转换为可读格式, 请执行类似下面的查询:

```
CREATE TABLE T3 (
pk uniqueidentifier primary key, c1 int);
```

## SQL 函数

```
INSERT INTO T3 (pk, c1)
values (0x12345678123456789012123456789012, 1)
SELECT UIDTOSTR(pk) FROM T3
```

### 另请参见

- 二进制数据类型 (第 352 页)
- NEWID 函数 [杂项] (第 247 页)
- STRTOUUID 函数 [字符串] (第 317 页)
- 字符数据类型 (第 345 页)
- 二进制数据类型 (第 759 页)

## VAR\_POP 函数 [集合]

计算由数值表达式组成的总体的统计方差，类型为 DOUBLE。

### 语法

```
VAR_POP ( [ ALL ] expression )
```

### 参数

参数	描述
expression	要对一组行计算其基于总体的方差的表达式 (通常是列名)。

### 返回

DOUBLE

### 注释

计算所提供的对每个组行或分区行求值的值表达式的总体方差 (如果指定了 **DISTINCT**, 则为在消除重复项后剩余的各行), 其定义为: 值表达式与值表达式均值之差的平方和, 然后再除以组中或分区中剩余的行数。

基于总体的方差是根据以下公式计算的:

$$\frac{\sum (x_i - \bar{x})^2}{n}$$

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句列出不同时间段每个订单的产品数目的平均值和方差:

```
SELECT year( ShipDate ) AS Year, quarter( ShipDate )
AS Quarter, AVG( Quantity ) AS Average,
```



```
VAR_POP( Quantity ) AS Variance
FROM SalesOrderItems GROUP BY Year, Quarter
ORDER BY Year, Quarter
```

Year	Quarter	Average	Variance
2000	1	25.775148	203.9021
2000	2	27.050847	225.8109
...	...	...	...

另请参见

- 窗口化集合函数的用法（第 78 页）

## **VAR\_SAMP 函数 [集合]**

计算由数值表达式组成的样本的统计方差，类型为 DOUBLE。

语法

```
VAR_SAMP ( [ ALL ] expression )
```

参数

参数	描述
expression	要对一组行计算其基于样本的方差的表达式（通常是列名）。

返回

DOUBLE

注释

**注意：**VAR\_SAMP 是 VARIANCE 的别名。

计算对每个组行或分区行求值的值表达式的样本方差（如果指定了 DISTINCT，则为在消除重复项后剩余的各行），其定义为：值表达式与值表达式均值之差的平方和，然后再除以组中或分区中的剩余行数减去一。

对于 SAP Sybase IQ 中的一个元素的输入集，NULL 会返回 NULL。

根据下面的公式计算方差，其中假定正态分布：

$$\frac{\sum (x_i - \bar{x})^2}{n}$$

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server 支持。

### 示例

以下语句列出不同时间段每个订单的产品数目的平均值和方差：

```
SELECT year( ShipDate ) AS Year, quarter( ShipDate )
      AS Quarter, AVG( Quantity ) AS Average,
      VAR_SAMP( Quantity ) AS Variance
FROM SalesOrderItems GROUP BY Year, Quarter
      ORDER BY Year, Quarter
```

Year	Quarter	Average	Variance
2000	1	25.775148	205.1158
2000	2	27.050847	227.0939
...	...	...	...

### 另请参见

- 窗口化集合函数的用法 (第 78 页)

## VARIANCE 函数 [集合]

返回一组数字的方差。

### 语法

```
VARIANCE ( [ ALL ] expression )
```

### 参数

参数	描述
expression	任何数值数据类型 (FLOAT、REAL 或 DOUBLE) 的表达式。 要对一组行计算其基于样本的方差的表达式 (通常是列名)。

### 返回

DOUBLE

### 注释

用于计算 **VARIANCE** 的公式是

$$var = \frac{n \sum x^2 - (\sum x)^2}{n(n-1)}$$

**VARIANCE** 返回 double-precision floating-point 数据类型的结果。如果应用于空集，则结果为 NULL，对于一个元素的输入集，它将返回 NULL。

**VARIANCE** 不支持关键字 **DISTINCT**。如果 **DISTINCT** 与 **VARIANCE** 一起使用，则会返回语法错误。

#### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

#### 示例

假定有以下数据:

```
SELECT Salary FROM Employees WHERE DepartmentID = 300
```

salary
51432.000
57090.000
42300.000
43700.00
36500.000
138948.000
31200.000
58930.00
75400.00

以下语句返回值 1063923790.99999994:

```
SELECT VARIANCE ( Salary ) FROM Employees
WHERE DepartmentID = 300
```

假定有以下数据:

```
SELECT UnitPrice FROM Products WHERE name = 'Tee Shirt'
```

UnitPrice
9.00
14.00
14.00

以下语句返回值 8.333333333333334327:

```
SELECT VARIANCE ( UnitPrice ) FROM Products
WHERE name = 'Tee Shirt'
```

### 另请参见

- 窗口化集合函数的用法 (第 78 页)
- STDDEV 函数 [集合] (第 310 页)
- STDDEV\_SAMP 函数 [集合] (第 312 页)

## WEEKS 函数 [日期和时间]

返回自任意开始日期/时间以来的周数，返回两个指定日期/时间之间的周数，或者将指定的 *integer-expression* 数量的周添加到某日期/时间中。

### 语法

```
WEEKS ( datetime-expression
| datetime-expression, datetime-expression
| datetime-expression, integer-expression )
```

### 参数

参数	描述
<i>datetime-expression</i>	日期和时间。
<i>integer-expression</i>	要添加到 <i>datetime-expression</i> 中的周数。如果 <i>integer-expression</i> 是负数，则从日期/时间值中减去相应的周数。忽略小时、分钟和秒。如果提供整数表达式，则必须将 <i>datetime-expression</i> 显式转换为 DATETIME 数据类型。

### 返回

语法 1 返回 INTEGER。

语法 2 返回 TIMESTAMP。

### 注释

周是指从星期日到星期六，这与北美日历中一样。第一种语法返回的数字通常用于确定两个日期是否在同一周中。

```
WEEKS ( invoice_sent ) = WEEKS ( payment_received ) FROM iq_dummy
```

在第二个语法中，系统将根据两个日期之间的星期日的数量计算 **WEEKS** 的值。忽略小时、分钟和秒。此函数不受 DATE\_FIRST\_DAY\_OF\_WEEK 选项的影响。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

- Sybase - 不受 Adaptive Server Enterprise 支持。

#### 示例

以下语句返回值 104278:

```
SELECT WEEKS( '1998-07-13 06:07:12' ) FROM iq_dummy
```

以下语句返回值 9, 表示两个日期之间的差值:

```
SELECT WEEKS( '1999-07-13 06:07:12',
              '1999-09-13 10:07:12' ) FROM iq_dummy
```

以下语句返回时间戳值 1999-06-16 21:05:07.000:

```
SELECT WEEKS( CAST( '1999-05-12 21:05:07'
                   AS TIMESTAMP ), 5) FROM iq_dummy
```

#### 另请参见

- CAST 函数 [数据类型转换] (第 143 页)
- CONVERT 函数 [数据类型转换] (第 153 页)
- HOURS 函数 [日期和时间] (第 206 页)
- MINUTES 函数 [日期和时间] (第 242 页)
- MONTHS 函数 [日期和时间] (第 246 页)
- REPLACE 函数 [字符串] (第 286 页)
- SECOND 函数 [日期和时间] (第 296 页)
- YEAR 函数 [日期和时间] (第 341 页)
- YEARS 函数 [日期和时间] (第 342 页)

## WEIGHTED\_AVG 函数 [集合]

计算算术 (或线性) 加权平均值。

加权平均值是一个平均值, 对于该平均值, 将为每个要平均的数量分配一个权重。加权确定组成平均值的每个数量的相对重要性。

#### 语法

```
WEIGHTED_AVG (expression)
```

```
OVER (window-spec)
```

window-spec: 请参见下面的“用法”部分。

## 参数

参数	描述
expression	计算其加权值的数值表达式。

## 注释

使用 **WEIGHTED\_AVG** 函数创建加权移动平均值。在加权移动平均值中，权重随着时间的推移按算术级数减小。对于最新的数据点，权重从最高值减小，直到减小为零。

图 1: **WEIGHTED\_AVG** 计算

$$WMA_M = \frac{np_M + (n-1)p_{M-1} + \dots + 2p_{M-n+2} + p_{M-n+1}}{n + (n-1) + \dots + 2 + 1}$$

若要增大加权，可以平均两个或多个加权移动平均值，或改用 **EXP\_WEIGHTED\_AVG** 函数。

您可以在函数语法（行内）中或使用 **SELECT** 语句中的 **WINDOW** 子句指定 *window-spec* 的元素。

*window-spec*:

- 必须包含 **ORDER BY** 指示符。
- 不能包含 **FOLLOWING** 或 **RANGE** 指示符。
- **ROW** 指示符（如果提供）的第二个参数必须为 **CURRENT ROW**。
- 不能包含空值。
- 不能包含 **DISTINCT** 指示符。
- 支持 **UNBOUNDED PRECEDING**，但如果使用，可能导致性能下降。

## 标准和兼容性

- **SQL - ISO/ANSI SQL** 语法的供应商扩展。

## 示例

以下示例按部门返回佛罗里达州雇员的加权平均薪水，其中最近雇用员工的薪水对平均值的加权贡献最大：

```
SELECT DepartmentID, Surname, Salary,
WEIGHTED_AVG(Salary) OVER (PARTITION BY DepartmentID
ORDER BY YEAR(StartDate) DESC) as "W_AVG"
FROM Employees
WHERE State IN ('FL') ORDER BY DepartmentID
```

返回的结果集为：

表 53. WEIGHTED\_AVG 结果集

DepartmentID	Surname	Salary	W_AVG
100	Lull	87,900.000	87,900.000000
100	Gowda	59,840.000	69,193.333333
200	Sterling	64,900.000	64,900.000000
200	Kelly	87,500.000	79,966.666667
300	Litton	58,930.000	58,930.000000
400	Evans	68,940.000	68,940.000000
400	Charlton	28,300.000	41,846.666667
400	Francis	53,870.000	47,858.333333

另请参见

- EXP\_WEIGHTED\_AVG 函数 [集合] (第 194 页)
- 窗口化集合函数的用法 (第 78 页)

## WIDTH\_BUCKET 函数 [数值]

对于给定表达式，**WIDTH\_BUCKET** 函数返回表元号，即在计算此表达式后将为其结果分配的表元号。

语法

```
WIDTH_BUCKET ( expression, min_value, max_value, num_buckets )
```

参数

参数	描述
<code>expression</code>	为其创建直方图的表达式。此表达式的计算结果必须是数值或日期时间值，或者是可以隐式转换为数值或日期时间值的值。如果 <i>expr</i> 的计算结果为空值，则表达式返回空值。
<code>min_value</code>	解析为 <i>expr</i> 的可接受范围的端点的表达式。还必须计算为数值或日期时间值，不能计算为空。
<code>max_value</code>	解析为 <i>expr</i> 的可接受范围的端点的表达式。还必须计算为数值或日期时间值，不能计算为空。
<code>num_buckets</code>	是解析为表示表元数的常量的表达式。此表达式的计算结果必须是正整数。

*注释*

您可以使用 **WIDTH\_BUCKET** 函数生成等宽直方图。等宽直方图将数据集分为区间大小（最大值到最小值）相等的表元。每个表元含有的行数将会不同。相关函数 **NTILE** 创建等高表元。

仅能为数字、日期或日期时间数据类型生成等宽直方图；因此，前三个参数应该全部是数值表达式或者全部是日期表达式。不允许其它类型的表达式。如果第一个参数为 **NULL**，则结果为 **NULL**。如果第二个或第三个参数为 **NULL**，则返回错误消息，原因是：空值无法以日期或数值形式表示某个范围的终点（或任意点）。最后一个参数（表元数）应该是计算结果为正整数值的数值表达式；0、**NULL** 或负值将导致出现错误。

表元的编号为 0 到 (n+1)。表元 0 含有的值数小于最小数量。表元 (n+1) 含有的值数大于或等于指定的最大值。

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下示例基于 `credit_limit` 列为示例表中马萨诸塞州的客户创建十域桶直方图，并为每名客户返回域桶号 ("Credit Group")。信用额度超过最大值的客户将被分配到溢出域桶 11：

```
select EmployeeID, Surname, Salary, WIDTH_BUCKET(Salary, 29000,
60000, 4) "Wages" from Employees where State = 'FL' order by "Wages"
```

EMPLOYEEID	SURNAME	SALARY	Wages
888	Charlton	28300.000	0
1390	Litton	58930.000	4
207	Francis	53870.000	4
266	Gowda	59840.000	4
445	Lull	87900.000	5
1021	Sterling	64900.000	5
902	Kelly	87500.000	5
1576	Evans	68940.000	5

如果将这些限定反转过来，桶将成为半开半闭区间。例如：**WIDTH\_BUCKET** (`credit_limit`, 5000, 0, 5)。在此示例中，桶号 1 的上下限为 (4000, 5000]，桶号 2 的上下限为 (3000, 4000]，桶号 5 的上下限为 (0, 1000]。上溢桶的编号为 0 (5000, 正无穷)，下溢桶的编号为 6 (负无穷, 0]。



## YEAR 函数 [日期和时间]

返回与给定日期/时间的年份相对应的 4 位数数字。

### 语法

```
YEAR ( datetime-expression )
```

### 参数

参数	描述
datetime-expression	日期和时间。

### 返回

SMALLINT

### 注释

**YEAR** 函数与 **YEARS** 函数的第一种语法相同。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回值 1998:

```
SELECT YEAR( '1998-07-13 06:07:12' ) FROM iq_dummy
```

### 另请参见

- CAST 函数 [数据类型转换] (第 143 页)
- CONVERT 函数 [数据类型转换] (第 153 页)
- HOURS 函数 [日期和时间] (第 206 页)
- MINUTES 函数 [日期和时间] (第 242 页)
- MONTHS 函数 [日期和时间] (第 246 页)
- REPLACE 函数 [字符串] (第 286 页)
- SECOND 函数 [日期和时间] (第 296 页)
- WEEKS 函数 [日期和时间] (第 336 页)
- YEARS 函数 [日期和时间] (第 342 页)
- NTILE 函数 [分析] (第 253 页)

## YEARS 函数 [日期和时间]

返回与给定日期/时间的年份相对应的 4 位数数字，返回两个指定日期/时间之间的年数，或向某日期/时间添加指定的 *integer-expression* 数量的年。

### 语法

```
YEARS ( datetime-expression
| datetime-expression, datetime-expression
| datetime-expression, integer-expression )
```

### 参数

参数	描述
<i>datetime-expression</i>	日期和时间。
<i>integer-expression</i>	要添加到 <i>datetime-expression</i> 中的年数。如果 <i>integer-expression</i> 是负数，则从日期时间值中减去相应的年数。如果提供整数表达式，则必须将 <i>datetime-expression</i> 显式转换为 DATETIME 数据类型。

### 返回

语法 1 返回 INTEGER。

语法 2 返回 TIMESTAMP。

### 注释

**YEARS** 函数的第一种语法与 **YEAR** 函数相同。

第二种语法返回从第一个日期到第二个日期的年数，该数量是根据两个日期间年的第一天的数量计算的。数值可能为负。会忽略小时、分钟和秒钟。例如，下面的语句返回 2，这是两个指定日期之间年的第一天的数量：

```
SELECT YEARS ( '2000-02-24', '2002-02-24' ) FROM iq_dummy
```

下面的语句也返回 2，即使两个指定日期之间的差值不是两个完整的日历年也是如此。值 2 是两个日期间年的第一天（在此示例中为 2001 年 1 月 1 日和 2002 年 1 月 1 日）的数量。

```
SELECT YEARS ( '2000-02-24', '2002-02-20' ) FROM iq_dummy
```

第三种语法将 *integer-expression* 数量的年添加到给定日期中。如果新日期超出月末（如 **SELECT YEARS (CAST ( '1992-02-29' AS TIMESTAMP), 1)**），则结果设置为该月的最后一天。如果 *integer-expression* 是负数，则从日期中减去相应的年数。会忽略小时、分钟和秒钟。

### 标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

### 示例

以下语句返回值 1998:

```
SELECT YEARS( '1998-07-13 06:07:12' ) FROM iq_dummy
```

以下语句返回值 2, 表示两个日期之间的差值。

```
SELECT YEARS( '1997-07-13 06:07:12',
              '1999-09-13 10:07:12' ) FROM iq_dummy
```

以下语句返回 YEARS(cast('1999-05-12 21:05:07' as timestamp), 5) 值 2004-05-12 21:05:07.000:

```
SELECT YEARS( CAST( '1999-05-12 21:05:07'
                   AS TIMESTAMP ), 5) FROM iq_dummy
```

### 另请参见

- CAST 函数 [数据类型转换] (第 143 页)
- CONVERT 函数 [数据类型转换] (第 153 页)
- HOURS 函数 [日期和时间] (第 206 页)
- MINUTES 函数 [日期和时间] (第 242 页)
- MONTHS 函数 [日期和时间] (第 246 页)
- REPLACE 函数 [字符串] (第 286 页)
- SECOND 函数 [日期和时间] (第 296 页)
- WEEKS 函数 [日期和时间] (第 336 页)
- YEAR 函数 [日期和时间] (第 341 页)

## YMD 函数 [日期和时间]

返回对应于给定年、月、日的日期值。

### 语法

```
YMD ( integer-expression1, integer-expression2, integer-expression3 )
```

### 参数

参数	描述
integer-expression1	年。

参数	描述
integer-expression2	表示月份的数字。如果月份超出 1-12 这一范围，则相应地调整年份。
integer-expression3	表示日的数字。日可以是任意整数，将相应地调整日期。

*返回*

DATE

*标准和兼容性*

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。

*示例*

以下语句返回值 1998-06-12:

```
SELECT YMD( 1998, 06, 12 ) FROM iq_dummy
```

如果值超出其正常范围，则相应地调整日期。例如，以下语句返回值 1993-03-01:

```
SELECT YMD( 1992, 15, 1 ) FROM iq_dummy
```

以下语句返回值 1993-02-28:

```
SELECT YMD ( 1992, 15, 1-1 ) FROM iq_dummy
```

以下语句返回值 1992-02-29:

```
SELECT YMD ( 1992, 3, 1-1 ) FROM iq_dummy
```

# SQL 数据类型

SQL 数据类型定义了要存储的数据的类型，例如字符串、数字和日期。

## 字符数据类型

使用字符数据类型来存储字母、数字和符号字符串。

### 语法

对于字符数据类型，请使用以下语法。

```
CHAR [ ( max-length ) ]
```

```
CHARACTER [ ( max-length ) ]
```

```
CHARACTER  
                  VARYING [ ( max-length ) ]
```

```
VARCHAR [ ( max-length ) ]
```

```
UNIQUEIDENTIFIERSTR
```

### 用法

下表介绍了字符数据类型。

表 54. 字符数据类型

字符数据类型	描述
CHAR	<p>最长 <i>max-length</i> 字节的字符数据。如果省略 <i>max-length</i>，则缺省值为 1。允许的最大大小为 32KB - 1。有关对超过 255 字节的 CHAR 数据的限制，请参见注释部分。</p> <p>请参见下文中关于数据库中字符数据的表示形式以及长字符串存储的说明。</p> <p>无论是否指定 BLANK PADDING 选项，系统均会用空白填补所有 CHAR 值以达到 <i>max-length</i>。将多字节字符串保存为 CHAR 类型时，最大长度仍以字节（而不是字符）为单位。</p>
CHARACTER	与 CHAR 相同。
CHARACTER VARYING	与 VARCHAR 相同。

字符数据类型	描述
LONG VARCHAR	任意长度的字符数据。最大大小受数据库文件的最大大小（当前为 2 GB）限制。
TEXT	这是用户定义数据类型。它是作为允许 NULL 的 LONG VARCHAR 实现的。
VARCHAR	除了不向这些字符串的存储中添加空白填充，且 VARCHAR 字符串的最大长度为 (32KB - 1) 之外，其它与 CHAR 相同。有关对超过 255 字节的 VARCHAR 数据的限制，请参见“注释”。
UNIQUEIDENTIFIERSTR	作为 CHAR ( 36 ) 实现的域。此数据类型用于在映射 Microsoft SQL Server uniqueidentifier 列时进行远程数据访问。

**注意：** 作为单独许可的组件，SAP Sybase IQ 支持字符大对象 (CLOB) 数据，对于 128KB 的 SAP Sybase IQ 页面大小，所支持的长度范围为零 (0) 到 512TB（千吉字节）；对于 512KB 的 SAP Sybase IQ 页面大小，则为零 (0) 到 2PB（千万亿字节）。最大长度为 4 GB 乘以数据库页面大小。请参见非结构化数据分析。

#### 另请参见

- 二进制数据类型（第 352 页）
- NEWID 函数 [杂项]（第 247 页）
- STRTOUUID 函数 [字符串]（第 317 页）
- UUIDTOSTR 函数 [字符串]（第 331 页）
- 二进制数据类型（第 759 页）
- 字符数据类型兼容性（第 758 页）

## 存储大小

字符数据的存储大小、给定的列定义大小和输入数据大小。

表 55. 字符数据的存储大小

数据类型	列定义	输入数据	存储
CHARACTER, CHAR	(32K - 1) 字节的宽度	(32K - 1) 字节	(32K - 1) 字节
VARCHAR, CHARACTER VARYING	(32K - 1) 字节的宽度	(32K - 1) 字节	(32K - 1) 字节

## 字符集和代码页

字符数据使用从应用程序传递而来的准确的二进制表示形式放置在数据库中。

这通常意味着字符数据是以系统使用的字符集的二进制表示形式存储在数据库中的。在操作系统的文档中可以找到有关字符集的文档。

在 Windows 上，对于前 128 个字符代码页均相同。如果您使用代码页上半部分中的特殊字符（变音国际语言字符），则必须当心您的数据库。具体说来，如果您将数据库复制到使用另一种代码页的不同计算机上，则从数据库中检索的那些特殊字符将采用原来的代码页表示形式。对于新代码页，它们在窗口上将显示为错误字符。

如果您有两个客户端，它们使用同一台多用户服务器，但运行不同的代码页，也会出现此问题。一个客户端插入或更新的数据在另一个客户端看来可能是不正确的。

如果跨平台使用数据库，也会出现此问题。PowerBuilder 和许多其它 Windows 应用程序将数据以标准 ANSI 字符集的形式插入到数据库中。如果非 Windows 应用程序试图使用这些数据，它们将无法正确地显示或更新扩展字符。

此问题十分复杂。如果您的某些应用程序使用代码页上半部分中的扩展字符，请确保使用数据库的所有客户端和所有计算机都使用相同或兼容的代码页。

## 索引

长度等于或小于 255 字节的 CHAR 数据和 VARCHAR 数据支持除 DATE、TIME 和 DTTM 以外的所有索引类型。

## VARCHAR 数据和尾随空白

对于数据类型为 VARCHAR 的列，系统将根据正在插入的数据是否用引号引起来，以不同的方式处理该数据中的尾随空白。

使用 **INSERT**、**UPDATE** 或 **LOAD TABLE** 插入的数据：

- 可以用引号引起来
- 可以不用引号引起来
- 可以是二进制数据

对于数据类型为 VARCHAR 的列，系统将以如下方式处理正在插入的数据中的尾随空白：

- 对于用引号引起来的数据，永远不会剪裁尾随空白。
- 对于未用引号引起来的数据：
  - 在插入和更新时始终剪裁尾随空白。
  - 对于 **LOAD** 语句，可以使用 **STRIP RTRIM/OFF LOAD** 选项来指定是否剪裁尾随空白。STRIP RTRIM/OFF 选项仅适用于长度可变的非二进制数据。例如，采用以下模式：

```
CREATE TABLE t( c1 VARCHAR(3) ); LOAD TABLE t( c1 ',' ) .....
STRIP RTRIM      // trailing blanks trimmed LOAD TABLE t( c1
',' ) ..... STRIP OFF      // trailing blanks not trimmed
LOAD TABLE t( c1 ASCII(3) ) ... STRIP RTRIM      // trailing
blanks not trimmed LOAD TABLE t( c1 ASCII(3) ) ... STRIP
OFF      // trailing blanks trimmed LOAD TABLE t( c1
BINARY ) ..... STRIP RTRIM      // trailing blanks trimmed LOAD
TABLE t( c1 BINARY ) ..... STRIP OFF      // trailing blanks
trimmed
```

## SQL 数据类型

- 对于二进制数据，始终剪裁尾随空白。

当编写应用程序时，不能依赖于 VARCHAR 列中存在的尾随空白。如果应用程序依赖于尾随空白，请使用 CHAR 列，而非 VARCHAR 列。

### 对于超过 255 字节的 CHAR 和 VARCHAR 数据的限制

长度超过 255 字节的 CHAR 和 VARCHAR 列仅支持缺省索引以及 **WD**、**TEXT** 和 **CMP** 索引类型。

您不能为这些列创建 **LF**、**HG**、**HNG**、**DATE**、**TIME** 或 **DTTM** 索引。

### 字符数据兼容性

SAP Sybase IQ、Adaptive Server 和 SQL Anywhere 之间存在字符数据兼容性的区别。

- Adaptive Server 不支持 CHAR 的替换形式 CHARACTER (*n*)。
- SAP Sybase IQ 不支持 Adaptive Server 提供的 NCHAR、NVARCHAR、UNICHAR 和 UNIVARCHAR 数据类型。SAP Sybase IQ 支持 CHAR 和 VARCHAR 数据类型中的 Unicode。
- 与 SQL Anywhere 相比，SAP Sybase IQ 支持更长的 LONG VARCHAR 数据类型。请参见非结构化数据分析。
- 为使 SAP Sybase IQ 和 Adaptive Server 兼容，请始终指定字符数据类型的长度。

### 长字符串

不超过 254 个字符的值存储为短字符串，并且前面带有一个长度字节。任何长度超过 255 个字节的值都被视为长字符串。第 255 个字节之后的字符与包含长字符串值的行分开存储。

SQL Anywhere 将 CHAR、VARCHAR 和 LONG VARCHAR 列全部视为相同的类型。

有几种函数将忽略任何字符串中第 255 个字符之后的部分。它们是 **soundex**、**similar** 和所有日期函数。另外，任何涉及长字符串到数字的转换的运算都只对前 255 个字符有效。出现上述限制的可能性非常小。

所有其它函数和所有其它运算符都对长字符串的全部长度起作用。

## 数值数据类型

使用数值数据类型来存储数值数据。

### 语法

对于数值数据类型，请使用以下语法。

```
[ UNSIGNED ] BIGINT
```



[ UNSIGNED ] { INT | INTEGER }

SMALLINT

TINYINT

DECIMAL [ ( precision [ , scale ] ) ]

NUMERIC [ ( precision [ , scale ] ) ]

DOUBLE

FLOAT [ ( precision ) ]

REAL

## 数值数据类型的用法

使用数值数据类型时，请注意以下几点。

- INTEGER、NUMERIC，和 DECIMAL 数据类型有时称为精确数值数据类型，与之相对的是近似数值数据类型 FLOAT、DOUBLE 和 REAL。只有精确数值数据才能确保会在算术运算后精确到指定的最小有效位。
- 不要将 TINYINT 列提取到定义为 CHAR 或 UNSIGNED CHAR 的嵌入式 SQL 变量中，因为这样做会导致系统试图将列的值转换为字符串，然后将第一个字节分配给程序中的变量。
- 句点是唯一的小数分隔符（小数点）；系统不支持逗号作为小数分隔符。

表 56. 数值数据类型

数值数据类型	描述
BIGINT	带符号的 64 位整数，需要 8 个字节的存储空间。 您可以将整数指定为 UNSIGNED。缺省情况下，该数据类型是带符号的。它的范围介于 -9223372036854775808 到 9223372036854775807（带符号）之间，或者 0 到 18446744073709551615（不带符号）。
INT 或 INTEGER	带符号的 32 位整数，值的范围介于 -2147483648 和 2147483647 之间，需要 4 字节的存储空间。 INTEGER 数据类型是精确数值数据类型；其精度在算术运算后保持不变。 您可以将整数指定为 UNSIGNED；缺省情况下，该数据类型是带符号的。不带符号的整数的值范围介于 0 到 4294967295 之间。
SMALLINT	16 位有符号整数，范围介于 -32768 和 32767 之间，需要 2 字节的存储空间。 SMALLINT 数据类型是一种精确数值数据类型，其精度在算术运算后保持不变。
TINYINT	不带符号的 8 位整数，范围介于 0 和 255 之间，需要 1 字节的存储空间。 TINYINT 数据类型是精确数值数据类型；其精度在算术运算后保持不变。

数值数据类型	描述
DECIMAL	总位数为 <i>precision</i> 、小数点后位数为 <i>scale</i> 的带符号十进制数字。精度值的范围是 1 到 126，而小数位数的范围是 0 到精度值。缺省值是：小数位数 = 38，精度 = 126。为了确保精确，将根据列的实际数据类型计算结果，但您可以使用 MAX_CLIENT_NUMERIC_SCALE 选项来设置返回到应用程序的结果的最大小数位数。
NUMERIC	与 DECIMAL 相同。
DOUBLE	占用 8 个字节存储空间的双精度浮点数（带符号）。绝对非零值的范围介于 2.2250738585072014e-308 和 1.797693134862315708e+308 之间。保存为 DOUBLE 的值精确到 15 个有效位，但超出第十五位后可能会产生舍入误差。  DOUBLE 数据类型是近似数值数据类型；它在算术运算后容易产生舍入误差。
FLOAT	如果未提供 <i>precision</i> ，则 FLOAT 数据类型与 REAL 数据类型相同。如果提供了 <i>precision</i> ，则 FLOAT 数据类型与 REAL 或 DOUBLE 数据类型相同，具体取决于精度值。REAL 和 DOUBLE 之间的分界值取决于平台，它是平台上单精度浮点数的尾数中使用的位数。  当使用 FLOAT 数据类型创建列时，所有平台上的列所保存的值一定会至少精确为指定的最小精度。相反，REAL 和 DOUBLE 不能保证与平台无关的最小精度。  FLOAT 数据类型是近似数值数据类型；它在算术运算后容易产生舍入误差。
REAL	占用 4 个字节存储空间单精度浮点数（带符号）。绝对非零值的范围介于 1.175494351e-38 和 3.402823466e+38 之间。保存为 REAL 的值精确到 6 个有效位，但超出第六位后可能会产生舍入误差。  REAL 数据类型是近似数值数据类型；它在算术运算后容易产生舍入误差。

下表列出了十进制数字所需的存储空间。

表 57. 十进制数字的存储大小

精度	存储
1 到 4	2 字节
5 到 9	4 字节
10 到 18	8 字节
19 到 126	请参见下文

对于精度大于 18 的十进制值的存储空间要求（以字节为单位），可以使用以下公式计算：

```
4 + 2 * (int(((prec - scale) + 3) / 4) +
int((scale + 3) / 4) + 1)
```

其中 *int* 取其参数的整数部分。列所使用的存储空间大小取决于列的精度和标度。列中的每个单元格都有足够的空间来存储该精度和标度的最大值。例如：

```
NUMERIC(18,4) takes 8 bytes per cell
NUMERIC(19,4) takes 16 bytes per cell
```

DECIMAL 数据类型是精确数值数据类型；其精度在算术运算后保留到最小有效位。它的最大绝对值是 [*precision - scale*] 定义的 9 的个数，接着依次是小数点、*scale* 定义的 9 的个数。最小绝对非零值是小数点后接 [*scale - 1*] 定义的 0 的个数，然后再接一个 1。例如：

```
NUMERIC (3,2) Max positive = 9.99 Min non-zero = 0.01 Max negative =
-9.99
```

如果针对 NULL 到 NUMERIC 的显式转换既未指定精度也未指定小数位数，则缺省值为 NUMERIC(1,0)。例如，

```
SELECT CAST ( NULL AS NUMERIC ) A,
        CAST ( NULL AS NUMERIC (15,2) ) B
```

被描述为：

```
A NUMERIC (1,0)
B NUMERIC (15,2)
```

**注意：** 对于数值函数，SQL Anywhere 中支持的最大值是 255。如果数值函数的精度超过 SQL Anywhere 中支持的最大值，则会出现以下错误：“函数 '\_funcname' 的结果数据类型超出支持的最大数字精度 255。请为数字函数 'location' 精度设置正确的值”

### 数值数据兼容性

SAP Sybase IQ、Adaptive Server 和 SQL Anywhere 之间存在数值数据兼容性区别。

- 在嵌入式 SQL 中，将 TINYINT 列读入 2 字节或 4 字节的整数列中。此外，要将 TINYINT 值发送到数据库，C 变量应该为整数。
- Adaptive Server 12.5.x 版不支持不带符号的整数。（15.x 版本支持不带符号的整数数据类型。）您可以将 SAP Sybase IQ 中不带符号的整数映射到 Adaptive Server 中带符号的整数或数值数据，数据将进行隐式转换。
  - 将 SAP Sybase IQ UNSIGNED SMALLINT 数据映射到 ASE INT
  - 如果存在负值，请将 SAP Sybase IQ UNSIGNED BIGINT 映射到 ASE NUMERIC (*precision*, *scale*)  
若要避免 UNSIGNED BIGINT 列的跨数据库连接出现性能问题，最好在 SAP Sybase IQ 端转换为（带符号的）BIGINT。
- 应该避免为 NUMERIC 和 DECIMAL 数据类型使用缺省精度和小数位数设置，因为这些缺省设置是因产品而异的：

数据库	缺省精度	缺省小数位数
SAP Sybase IQ	126	38
Adaptive Server	18	0
SQL Anywhere	30	6

- `FLOAT(p)` 数据类型是 `REAL` 或 `DOUBLE` 的同义词，具体取决于  $p$  的值。对于 Adaptive Server， $p$  小于等于 15 时使用 `REAL`，而  $p$  大于 15 时使用 `DOUBLE`。对于 SAP Sybase IQ，分界值则与平台有关，但所有平台上的分界值均大于 22。
- SAP Sybase IQ 包括两种用户定义数据类型 `MONEY` 和 `SMALLMONEY`，它们分别作为 `NUMERIC(19,4)` 和 `NUMERIC(10,4)` 实现。之所以提供它们，主要是为了与 Adaptive Server 兼容。

### 索引

本节介绍了索引类型和数值数据类型之间的关系。

- **CMP** 和 **HNG** 索引类型不支持 `FLOAT`、`DOUBLE` 和 `REAL` 数据类型，不建议您使用 **HG** 索引类型。
- **WD**、**DATE**、**TIME** 和 **DTTM** 索引类型不支持数值数据类型。

## 二进制数据类型

使用二进制数据类型来存储采用类似十六进制表示形式的原始二进制数据（如图片），最大长度为 (32K - 1) 字节。

### 语法

```
BINARY [ ( length ) ]
```

```
VARBINARY [ ( max-length ) ]
```

```
UNIQUEIDENTIFIER
```

### 另请参见

- `NEWID` 函数 [杂项]（第 247 页）
- `STRTOUUID` 函数 [字符串]（第 317 页）
- `UIDTOSTR` 函数 [字符串]（第 331 页）
- 字符数据类型（第 345 页）
- 二进制数据类型（第 759 页）

## 二进制数据类型的用法

二进制数据以字符“0x”或“0X”开头，并且可以包括数字和大小写字母（A 到 F）的任意组合。

您可以指定列长度（以字节为单位），也可以使用缺省长度（1 字节）。每个字节可以存储 2 位十六进制数字。虽然缺省长度为 1 字节，但建议您始终指定偶数数量的字符作为 BINARY 和 VARBINARY 列的长度。如果输入的值长于指定的列长度，则 SAP Sybase IQ 会在不显示警告或错误的情况下将输入值截断为指定的长度。

二进制数据类型	描述
BINARY	长度为 <i>length</i> 字节的二进制数据。如果省略 <i>length</i> ，则缺省值为 1 字节。允许的最大大小为 32767 个字节。对于预计所含所有条目的长度大致相等的的数据，可使用固定长度二进制类型 BINARY。因为 BINARY 列中的条目将用零填补到列长度 <i>length</i> ，所以，它们需要的存储空间可能比 VARBINARY 列中的条目需要的存储空间多。
VARBINARY	最大长度为 <i>max-length</i> 字节的二进制数据。如果省略 <i>max-length</i> ，则缺省值为 1 字节。允许的最大大小为 (32K - 1) 字节。对于预计长度变化会很大的数据，可使用可变长度二进制类型 VARBINARY。
UNIQUEIDENTIFIER	UNIQUEIDENTIFIER 数据类型用于存储 UUID（也称为 GUID）值。

### 尾随零的处理

所有 BINARY 列均用零填补以达到列的完整宽度。在所有 VARBINARY 列中，系统均会截断尾随零。

以下示例创建一个表，该表中含有使用 NULL 和 NOT NULL 定义的 BINARY 和 VARBINARY 数据类型的所有四个变体。在所有四个列中插入相同数据，并根据列的数据类型对数据进行填补或截断。

```
CREATE TABLE zeros (bnot BINARY(5) NOT NULL,
                    bnull BINARY(5) NULL,
                    vbnot VARBINARY(5) NOT NULL,
                    vbnull VARBINARY(5) NULL);
INSERT zeros VALUES (0x12345000, 0x12345000,
                    0x12345000, 0x12345000);
INSERT zeros VALUES (0x123, 0x123, 0x123, 0x123);
INSERT zeros VALUES (0x0, 0x0, 0x0, 0x0);
INSERT zeros VALUES ('002710000000ae1b',
                    '002710000000ae1b', '002710000000ae1b',
                    '002710000000ae1b');
SELECT * FROM zeros;
```

bnot	bnull	vbnot	vbnull
0x1234500000	0x1234500000	0x12345000	0x12345000

bnot	bnull	vbnot	vbnnull
0x0123000000	0x0123000000	0x0123	0x0123
0x0000000000	0x0000000000	0x00	0x00
0x3030323731	0x3030323731	0x3030323731	0x3030323731

因为每个存储字节都包含 2 位十六进制数字，所以 SAP Sybase IQ 希望二进制条目包含字符 "0x"，其后紧跟偶数个数字。如果 "0x" 后跟奇数个数字，SAP Sybase IQ 会认为您省略了前导 0，并会为您添加。

输入值 "0x00" 和 "0x0" 以 "0x00" 的形式存储在可变长度二进制列 (VARBINARY) 中。在固定长度二进制列 (BINARY) 中，系统会使用零填补值，直到达到字段的完整长度：

```
INSERT zeros VALUES (0x0, 0x0, 0x0, 0x0);
SELECT * FROM zeros
```

bnot	bnull	vbnot	vbnnull
0x0000000000	0x0000000000	0x00	0x00

如果输入值不包括 "0x"，SAP Sybase IQ 会假定该值为 ASCII 值并对其进行转换。例如：

```
CREATE TABLE sample (col_bin BINARY(8));
INSERT sample VALUES ('002710000000ae1b');
SELECT * FROM sample;
```

col_bin
0x3030323731303030

**注意：**在上例中，确保将 `string_rtruncation` 选项设置为 "off"。

选择 BINARY 值时，必须以填补零指定值或使用 **CAST** 函数。例如：

```
SELECT * FROM zeros WHERE bnot = 0x0123000000;
```

或者：

```
SELECT * FROM zeros WHERE bnot = CAST(0x0123 as binary(5));
```

### 来自平面文件的 ASCII 数据

从平面文件装载到二进制类型列 (BINARY 或 VARBINARY) 的任何 ASCII 数据都会存储为半字节。

例如，如果将 0x1234 或 1234 从平面文件读入二进制列中，SAP Sybase IQ 会将值存储为十六进制 1234。SAP Sybase IQ 忽略 "0x" 前缀。如果输入数据包含任何超出范围 0 - 9、a - f 和 A - F 的字符，则该数据会被拒绝。

## 存储大小

熟悉二进制数据的存储大小。

表 58. 二进制数据的存储大小

数据类型	列定义	输入数据	存储
VARBINARY	(32K - 1) 字节的宽度	(32K - 1) 字节的二进制数据	(32K - 1) 字节
VARBINARY	(32K - 1) 字节的宽度	(64K - 2) 字节的 ASCII 数据	(32K - 1) 字节
BINARY	(32K - 1) 字节的宽度	(32K - 1) 字节的二进制数据	(32K - 1) 字节
BINARY	(32K - 1) 字节的宽度	(64K - 2) 字节的 ASCII 数据	(32K - 1) 字节

输入某个特定值所采用的确切格式取决于您所使用的平台。因此，涉及二进制数据的计算可能会在不同的计算机上产生不同的结果。

对于在十六进制字符串和整数之间执行的与平台无关的转换，请使用 **INTTOHEX** 和 **HEXTOINT** 函数，而不是特定于平台的 **CONVERT** 函数。

## 另请参见

- 数据类型转换函数（第 83 页）
- 数据类型转换（第 366 页）

## 字符串运算符

并置字符串运算符 **||** 和 **+** 都支持二进制类型的数据。

将二进制操作数显式转换为字符数据类型时不必使用 **||** 运算符。但是，显式和隐式数据转换生成的结果不同。

## 对 **BINARY** 和 **VARBINARY** 数据的限制

限制适用于包含 **BINARY** 和 **VARBINARY** 数据的列。

- 不能将集合函数 **SUM**、**AVG**、**STDDEV** 或 **VARIANCE** 与二进制数据类型一起使用。集合函数 **MIN**、**MAX** 和 **COUNT** 支持二进制数据类型 **BINARY** 和 **VARBINARY**。
- **HNG**、**WD**、**DATE**、**TIME** 和 **DTTM** 索引不支持 **BINARY** 或 **VARBINARY** 数据。
- 长度超过 255 字节的 **BINARY** 和 **VARBINARY** 仅支持缺省索引以及 **CMP** 和 **TEXT** 索引类型。
- 长度等于小于 8 字节的 **BINARY** 和 **VARBINARY** 数据支持位操作。

## 二进制数据兼容性

SAP Sybase IQ、SQL Anywhere 和 Adaptive Server 对二进制数据尾随零的处理各不相同。

表 59. 尾随零的处理

数据类型	SAP Sybase IQ	SQL Anywhere	Adaptive Server
BINARY NOT NULL	填充	不填充	填充
BINARY NULL	填充	不填充	不填充
VARBINARY NOT NULL	截断, 不填充	截断, 不填充	截断, 不填充
VARBINARY NULL	截断, 不填充	截断, 不填充	截断, 不填充

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 均支持 **STRING\_RTRUNCATION** 数据库选项, 该选项会影响 **INSERT** 或 **UPDATE** 字符串被截断时所报告的错误消息。对于与 Transact-SQL 兼容的字符串比较, 请在两个数据库中将 **STRING\_RTRUNCATION** 选项设置为同一值。

您也可以在向表中装载数据时将 **STRING\_RTRUNCATION** 选项设置为 ON, 以便在数据太大而无法装载到字段中时提醒您。缺省值为 ON。

Adaptive Server 不支持对二进制类型数据进行位操作。SQL Anywhere 仅支持对二进制类型数据前四字节进行位操作。SAP Sybase IQ 支持对二进制类型数据前八字节进行位操作。

## UNIQUEIDENTIFIER

UNIQUEIDENTIFIER 数据类型用于存储 UUID (也称为 GUID) 值。

UNIQUEIDENTIFIER 数据类型经常用于主键或其它要保存 UUID (通用唯一标识符) 值 (可用于唯一标识行) 的唯一列。**NEWID** 函数生成 UUID 值的方式会使得在一台计算机上生成的值与在另一台计算机上生成的 UUID 不匹配。因此, 使用 **NEWID** 生成的 UNIQUEIDENTIFIER 值可在同步环境中作为键使用。

例如, 以下语句将更新表 mytab 并将列 uid\_col 的值设置为由 **NEWID** 函数生成的唯一标识符 (如果该列的当前值为空值)。

```
UPDATE mytab
  SET uid_col = NEWID()
  WHERE uid_col IS NULL
```

如果您执行以下语句,

```
SELECT NEWID()
```

系统会以 BINARY(16) 类型返回唯一标识符。例如, 其值可以是 0xd3749fe09cf446e399913bc6434f1f08。您可以使用 **UIDTOSTR()** 函数将此字符串转换为可读的格式。



UUID 值又称 GUID（全局唯一标识符）。

**STRTOUUID** 和 **UUIDTOSTR** 函数用于将值在 **UNIQUEIDENTIFIER** 和字符串表示形式之间转换。

**UNIQUEIDENTIFIER** 值会作为 **BINARY (16)** 存储并返回。

因为 **UNIQUEIDENTIFIER** 值很大，所以使用 **UNSIGNED BIGINT** 或 **UNSIGNED INT** 标识列（而不是 **UNIQUEIDENTIFIER**）将更加有效（如果您不需要跨数据库唯一标识符）。

### **UNIQUEIDENTIFIER 的标准和兼容性**

以下标准和兼容性适用于 **UNIQUEIDENTIFIER** 值。

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 受 SQL Anywhere 支持。不受 Adaptive Server 支持。

### **二进制大对象数据**

作为单独许可的组件，SAP Sybase IQ 支持二进制大对象 (BLOB) 数据，对于 128KB 的页面大小，所支持的长度范围为零 (0) 到 512TB（千吉字节）；对于 512KB 的页面大小，则为零 (0) 到 2PB（千万亿字节）。

最大长度为 4 GB 乘以数据库页面大小。请参见《非结构化数据分析》

## **位数据类型**

---

使用 **BIT** 数据类型来存储布尔值。

数据类型	值	支持情况
BIT	0 或 1	SQL Anywhere 和 Adaptive Server

### **用法**

**BIT** 仅存储值 0 或 1。

将任意非零值插入到 **BIT** 列中都会导致该列中存储一个 1。将任意零值插入到 **BIT** 列中都会存储一个 0。

**BIT** 数据仅支持缺省索引类型。

### **位数据兼容性**

Adaptive Server **BIT** 数据类型仅允许值 0 或 1。

## 日期和时间数据类型

使用日期和时间数据类型来存储日期和时间。

### 语法

对于日期和时间数据类型，请使用以下语法。

**DATE**

**DATETIME**

**SMALLDATETIME**

**TIME**

**TIMESTAMP**

### 另请参见

- **TIMESTAMP** 特殊值 (第 65 页)
- **CURRENT TIMESTAMP** 特殊值 (第 63 页)
- **CURRENT TIME** 特殊值 (第 63 页)
- **CURRENT DATE** 特殊值 (第 62 页)
- 检索日期和时间 (第 360 页)
- **BIGTIME** 和 **BIGDATETIME** 支持 (第 761 页)

## 日期和时间数据类型的用法

使用日期和时间数据类型之前，请熟悉以下用法的注意事项。

日期和时间数据类型	描述
DATE	日历日期，如年、月、日。年可以从 0001 到 9999。日必须是非零值，因而最早日期为 0001-01-01。DATE 值需要 4 字节的存储空间。
DATETIME	以 <b>TIMESTAMP</b> 形式实现的一个域。之所以提供 <b>DATETIME</b> ，主要是为了与 <b>Adaptive Server</b> 兼容。
SMALLDATETIME	以 <b>TIMESTAMP</b> 形式实现的一个域。之所以提供 <b>SMALLDATETIME</b> ，主要是为了与 <b>Adaptive Server</b> 兼容。
TIME	一天中的时间，包含小时、分钟、秒，以及秒的小数部分。小数存储到 6 个小数位。TIME 值需要 8 字节的存储空间。(ODBC 标准将 TIME 数据类型限制为精确到秒。为此，请勿在需要精确度高于秒的 <b>WHERE</b> 子句比较中使用 TIME 数据类型。)

日期和时间数据类型	描述
TIMESTAMP	时间点, 包含年、月、日、小时、分钟、秒, 以及秒的小数部分。小数存储到 6 个小数位。日必须为非零值。TIMESTAMP 值需要 8 字节的存储空间。

TIMESTAMP 数据类型的有效范围介于 0001-01-01 00:00:00.000000 和 9999-12-31 23:59:59.999999 之间。超出 1600-02-28 23:59:59 到 7911-01-01 00:00:00 范围的 TIMESTAMP 数据可能显示不完整, 但完整的日期时间值存储在数据库中; 可以先将该数据转换为字符串, 然后再查看完整的值。您可以使用 **CAST()** 函数执行此操作, 如下例中所示, 这会首先创建包含 DATETIME 和 TIMESTAMP 列的表, 然后在日期晚于 7911-01-01 的位置插入值。

```
create table mydates (id int, descript char(20),
    datetime_null datetime, timestamp_null timestamp);

insert into mydates values (1, 'example', '7911-12-30
    23:59:59', '7911-12-30 06:03:44');
commit;
```

如果选择不使用 **CAST**, 小时和分钟会设置为 00:00:

```
select * from mydates;

1, 'example', '7911-12-30 00:00:59.000', '7911-12-30
00:00:44.000'
```

如果选择使用转换, 则会看到完整的时间戳:

```
select id, descript, cast(datetime_null as char(21)),
    cast(timestamp_null as char(21)) from mydates;

1, 'example', '7911-12-30 23:59:59.0', '7911-12-30 06:03:44.0'
```

### 另请参见

- 字符串到日期时间转换的兼容性 (第 367 页)

### 支持的索引类型

日期和时间数据支持以下索引类型。

- 所有日期和时间数据类型支持 **CMP**、**HG**、**HNG** 和 **LF** 索引类型; 不支持 **WD** 索引类型。
- DATE 数据支持 **DATE** 索引。
- TIME 数据支持 **TIME** 索引。
- DATETIME 和 TIMESTAMP 数据支持 **DTTM** 索引。

### 发送日期和时间

通过以下方式向数据库发送日期和时间。

- 使用任何接口，以字符串形式
- 使用 ODBC，以 `TIMESTAMP` 结构形式
- 使用嵌入式 SQL，以 `SQLDATETIME` 结构形式

将时间作为字符串（对于 `TIME` 数据类型）或字符串的一部分（对于 `TIMESTAMP` 或 `DATE` 数据类型）发送到数据库时，小时、分钟和秒必须用冒号分隔开（格式为 *hh:mm:ss.sss*），但可以出现在字符串中的任何位置。可以选择用句点将秒与秒的小数部分相互分隔开，如 *hh:mm:ss.sss* 所示。以下是用于指定时间的有效而明确的字符串：

```
21:35 -- 24 hour clock if no am or pm specified
10:00pm -- pm specified, so interpreted as 12 hour clock
10:00 -- 10:00am in the absence of pm
10:23:32.234 -- seconds and fractions of a
                second included
```

将日期以字符串形式发送到数据库时，其到日期数据类型的转换是自动发生的。您可以通过下面两种方式之一提供字符串：

- 作为数据库明确解释的 *yyyy/mm/dd* 或 *yyyy-mm-dd* 格式字符串
- 作为根据 `DATE_ORDER` 数据库选项解释的字符串

日期格式字符串不能包含任何多字节字符。即使数据库的归类顺序是多字节归类顺序（如 `932JPN`），也只能在日期/时间/日期时间格式的字符串中使用单字节字符。

### 检索日期和时间

通过以下方式从数据库中检索日期和时间。

- 使用任何接口，以字符串形式
- 使用 ODBC，以 `TIMESTAMP` 结构形式
- 使用嵌入式 SQL，以 `SQLDATETIME` 结构形式

### 用法

以字符串形式检索日期或时间时，按照数据库选项 `DATE_FORMAT`、`TIME_FORMAT` 和 `TIMESTAMP_FORMAT` 所指定的格式进行检索。

日期中允许使用下列运算符：

表 60. 运算符

运算符	描述
时间戳 + 整数	向日期或时间戳中添加指定的天数。

运算符	描述
时间戳 - 整数	从日期或时间戳中减去指定的天数。
日期 - 日期	计算两个日期或时间戳之间的天数。
日期 + 时间	创建将给定日期和时间组合在一起的时间戳。

### 另请参见

- `TIMESTAMP` 特殊值 (第 65 页)
- `CURRENT_TIMESTAMP` 特殊值 (第 63 页)
- `CURRENT_TIME` 特殊值 (第 63 页)
- `CURRENT_DATE` 特殊值 (第 62 页)
- 日期和时间数据类型 (第 358 页)

### 比较日期和时间

要将日期作为字符串与某个字符串进行比较，请在比较前使用 `DATEFORMAT` 函数或 `CAST` 函数将日期转换为字符串。

### 用法

```
DATEFORMAT(invoice_date, 'yyyy/mm/dd') = '1992/05/23'
```

您可以为 `DATEFORMAT` 字符串表达式使用任意允许的日期格式。

日期格式字符串不得包含任何多字节字符。即使数据库的归类顺序是多字节归类顺序 (如 `932JPN`)，也只能在日期/时间/日期时间格式的字符串中使用单字节字符。

如果“?”表示多字节字符，则下面的查询会失败：

```
SELECT DATEFORMAT ( StartDate, 'yy?' ) FROM Employees;
```

而应使用并置运算符移动日期格式字符串外面的多字节字符：

```
SELECT DATEFORMAT (StartDate, 'yy' ) + '?' FROM Employees;
```

### 明确的日期和时间

使用明确的日期格式可以防止用户的 `DATE_ORDER` 设置影响对日期的正确解释。

### 用法

无论 `DATE_ORDER` 如何设置，格式为 `yyyy/mm/dd` 或 `yyyy-mm-dd` 的日期始终被视为日期。您可以使用其它字符作为分隔符；例如，问号、空格或逗号。如果不同用户可能采用不同的 `DATE_ORDER` 设置，则应使用此格式。例如，在存储过程中，使用明确的日期格式可以防止用户的 `DATE_ORDER` 设置影响对日期的正确解释。

格式为 `hh:mm:ss.sss` 的字符串也会被明确地解释为时间。

对于日期和时间的组合，任何明确的日期和任何明确的时间都会生成一个明确的日期时间值。此外，下列格式是明确的日期时间值：

```
YYYY-MM-DD HH.MM.SS.SSSSSS
```

只有当时间与日期组合使用时，才可以在时间中使用句点。

在其它环境中，可以使用更灵活的日期格式。SAP Sybase IQ 可以将多种字符串解释为格式。具体的解释取决于 DATE\_ORDER 数据库选项的设置。DATE\_ORDER 数据库选项可能包括值 'MDY'、'YMD' 或 'DMY'。例如，要将 DATE\_ORDER 选项设置为 'DMY'，请输入：

```
SET OPTION DATE_ORDER = 'DMY' ;
```

缺省 DATE\_ORDER 设置为 'YMD'。每当建立连接时，ODBC 驱动程序就会将 DATE\_ORDER 选项设置为 'YMD'。使用 **SET OPTION** 语句更改该值。

数据库选项 DATE\_ORDER 决定者数据库将字符串 10/11/12 解释为 1912 年 10 月 11 日、1910 年 11 月 12 日，还是 1912 年 11 月 10 日。日期字符串的年、月和日应该由特定字符（例如，"/"、"-" 或空格）分隔开并按 DATE\_ORDER 选项指定的顺序显示。

您可以 2 位数或 4 位数的形式提供年份。NEAREST\_CENTURY 选项 [TSQL] 的值影响对 2 位年份的解释：对于小于 NEAREST\_CENTURY 的值，系统将为其加上 2000；对于所有其它值，系统将为其加上 1900。此选项的缺省值为 50。因此，缺省情况下，年份 50 将解释为 1950，而年份 49 则解释为 2049。

月可以是月份名称或编号。小时和分钟用冒号分隔开，但可以出现在字符串中的任何位置。

使用 4 位数格式指定年份。

只要 DATE\_ORDER 的设置正确，以下字符串便均为有效日期：

```
99-05-23 21:35
99/5/23
1999/05/23
May 23 1999
23-May-1999
Tuesday May 23, 1999 10:00pm
```

如果字符串只包含部分日期说明，则使用缺省值填写日期。使用下列缺省值：

- 年 - 1900
- 月 - 无缺省值
- 日 - 1（对月份字段很有用；例如，“1999 年 5 月”代表日期“1999-05-01 00:00”）
- 小时、分钟、秒、小数 - 0

## 域

域是内置数据类型的别名，在适用的时候还会包括精度值和小数位数值。

域（也称为用户定义数据类型）允许针对同一数据类型，使用相同的 NULL 或 NOT NULL 条件自动定义数据库中的所有列。这有助于在整个数据库中保持一致性。域名

不区分大小写。如果试图创建与现有域同名只是大小写不同的域，则 SAP Sybase IQ 会返回错误。

## 简单域

可使用 `CREATE DOMAIN` 语句创建域。

以下语句创建一个名为 `street_address` 的数据类型，这是一个有 35 个字符的字符串：

```
CREATE DOMAIN street_address CHAR( 35 )
```

可以使用 `CREATE DATATYPE` 替代 `CREATE DOMAIN`，但不建议这样做，因为 `CREATE DOMAIN` 是 ISO/ANSI SQL 标准中使用的语法。

需要 `CREATE DATATYPE` 系统特权。创建某个数据类型后，执行 `CREATE DOMAIN` 语句的用户 ID 就是该数据类型的所有者。任何用户都可以使用该数据类型，并且，与其它数据库对象不同的是，所有者名称永远不会用作数据类型名称的前缀。

在定义列时，`street_address` 数据类型的使用方式可能与任何其它数据类型完全相同。例如，下表有两列，第二列为 `street_address` 列：

```
CREATE TABLE twocol (id INT,
street street_address)
```

所有者或 DBA 可以通过发出 `COMMIT`，然后使用 `DROP DOMAIN` 语句来删除域：

```
DROP DOMAIN street_address
```

仅当数据库中的任何表都未使用数据类型时，才可以执行此语句。

### 用户定义数据类型的约束和缺省值

用户定义数据类型中可以内置许多与列关联的属性，如允许空值、拥有缺省值等等。以这样的数据类型定义的任何列都会自动继承空值设置、检查条件和缺省值。这样就可以通过让数据库中的列保持一致，都具有类似的含义。

例如，演示数据库中的许多主键列都是含 ID 号的整数列。以下语句将创建对这些列可能有用的数据类型：

```
CREATE DOMAIN id INT
NOT NULL
DEFAULT AUTOINCREMENT
CHECK( @col > 0 )
```

使用数据类型 ID 创建的任何列都不得包含空值，应缺省为自动递增的值，并且必须包含正数。在 `@col` 变量中，可以使用任何标识符代替 `col`。

在需要时，可以通过显式提供列的属性来覆盖数据类型的属性。以数据类型 ID 创建且使用显式允许的空值的列允许空值（无论 ID 数据类型的设置是什么）。

## CREATE DOMAIN 语句

在数据库中创建用户定义数据类型。

快速链接:

[转至参数](#) (第 364 页)

[转至示例](#) (第 364 页)

[转至用法](#) (第 365 页)

[转至标准](#) (第 365 页)

[转至权限](#) (第 366 页)

### 语法

```
CREATE { DOMAIN | DATATYPE } domain-name data-type
... [ NOT ] NULL ]
... [ DEFAULT default-value ]
```

**default-value** - (back to Syntax)

**special-value**

```
| string
| global variable
| [ - ] number
| ( constant-expression )
| built-in-function( constant-expression )
| AUTOINCREMENT
| CURRENT DATABASE
| CURRENT REMOTE USER
| NULL
| TIMESTAMP
| LAST USER
```

**special-value** (back to default-value)

**CURRENT**

```
{ DATE
| TIME
| TIMESTAMP
| USER
| PUBLISHER }
| USER
```

### 参数

[\(返回顶部\)](#) (第 364 页)

- **data-type** - 具有精度和小数位数的内置数据类型

### 示例

[\(返回顶部\)](#) (第 364 页)



- **示例 1** - 创建名为 `address` 的数据类型，它包含 35 个字符长的字符串并且可以为 `NULL`：

```
CREATE DOMAIN address CHAR ( 35 ) NULL
```

## 用法

(返回顶部) (第 364 页)

用户定义数据类型是内置数据类型的别名，在适用的时候还会包括精度值和小数位数值。它们提高了方便性并促进了数据库的一致性。

---

**注意：** 使用 **CREATE DOMAIN** 而不是 **CREATE DATATYPE**，因为 **CREATE DOMAIN** 是 **ANSI/ISO SQL3** 术语。

---

创建数据类型的用户会自动成为此数据类型的所有者。在 **CREATE DATATYPE** 语句中不能指定所有者。用户定义数据类型名称必须是唯一的，所有用户都可以访问这个数据类型且不必将所有者作为前缀。

用户定义数据类型是数据库内的对象。其名称必须遵守标识符规则。与内置数据类型名称一样，用户定义数据类型名称始终不区分大小写。

缺省情况下，用户定义的数据类型允许 `NULL`，除非 **allow\_nulls\_by\_default** 数据库选项设置为 **OFF**。这种情况下，新的用户定义数据类型的缺省设置为不允许 `NULL`。基于用户定义数据类型创建的列可否为空值取决于该用户定义数据类型定义的设置，而不取决于引用该列时 **allow\_nulls\_by\_default** 选项的设置。任何在列定义中显式设置的 `NULL` 或 `NOT NULL` 都将覆盖用户定义数据类型设置。

使用 **CREATE DOMAIN** 语句可以对用户定义数据类型指定 **DEFAULT** 值。指定的 **DEFAULT** 值将由基于数据类型定义的任何列继承。在列上显式指定的任何 **DEFAULT** 值都将覆盖为该数据类型指定的值。

使用 **CREATE DOMAIN** 语句可以将称为 **CHECK** 条件的规则添加至用户定义数据类型的定义中。

对于基表、全局临时表、局部临时表和用户定义的数据类型，**SAP Sybase IQ** 将强制执行 **CHECK** 约束。

要从数据库中删除数据类型，请使用 **DROP** 语句。您必须是数据类型的所有者或者具有 **CREATE DATATYPE** 或 **CREATE ANY OBJECT** 系统特权才能删除用户定义的数据类型。

副作用：

- 自动提交

## 标准

(返回顶部) (第 364 页)

- **SQL** - 符合 **ISO/ANSI SQL** 标准。

- SAP Sybase 数据库产品 - 不受 Adaptive Server 支持。Transact-SQL 使用 **sp\_addtype** 系统过程以及 **CREATE DEFAULT** 和 **CREATE RULE** 语句提供类似功能。

### 权限

(返回顶部) (第 364 页)

需要以下特权之一：

- **CREATE DATATYPE** 系统特权。
- **CREATE ANY OBJECT** 系统特权。

### 域兼容性

SAP Sybase IQ、Adaptive Server 和 SQL Anywhere 之间存在域兼容性的区别。

- 命名约束和缺省值 - 在 SAP Sybase IQ 中，使用基本数据类型（并可以选择使用 **NULL** 或 **NOT NULL** 条件）创建用户定义数据类型。不支持命名约束和命名缺省值。
- 创建数据类型 - 在 SAP Sybase IQ 中，可以使用 **sp\_addtype** 系统过程或 **CREATE DOMAIN** 语句来添加域。在 Adaptive Server 中，必须使用 **sp\_addtype**。**sp\_addtype** 以及从 SQL Anywhere 继承的其它存储过程的所有者是 **dbo**。**dbo** 也是使用 SQL Anywhere 存储过程创建的所有对象的创建者，因此，没有 **ALTER DATATYPE** 或 **ALTER ANY OBJECT** 系统特权的用户将无法更改通过 **sp\_addtype** 创建的域，而没有 **DROP DATATYPE** 或 **DROP ANY OBJECT** 系统特权则将无法删除该域。必须具有这些系统特权才能相应更改或删除 **dbo** 所拥有的域。

## 数据类型转换

类型转换会自动进行，或者您可以使用 **CAST** 或 **CONVERT** 函数显式请求类型转换。

### 用法

如果字符串在数值表达式中使用或者作为期望数值参数的函数的参数使用，则该字符串在使用前会转换为数字。

如果数字在字符串表达式中使用或者作为字符串函数参数使用，则该数字在使用前会转换为字符串。

所有日期常量都被指定为字符串。字符串先自动转换为日期，然后再使用。

在一些情况下，不适合进行自动数据类型转换。

```
'12/31/90' + 5 -- Tries to convert the string to a number  
'a' > 0      -- Tries to convert 'a' to a number
```

您可以使用 **CAST** 或 **CONVERT** 函数强制进行类型转换。

也可以使用下列函数强制进行类型转换：

- **DATE( expression )** - 将表达式转换为日期，并删除任何小时、分钟或秒。可能会报告转换错误。
- **DATETIME( expression )** - 将表达式转换为时间戳。可能会报告转换错误。
- **STRING( expression )** - 类似于 `CAST(value AS CHAR)`，但 `string(NULL)` 是空字符串 (")，而 `CAST(NULL AS CHAR)` 是 `NULL` 值。

---

**注意：** 如果在将 `TINYINT`、`SMALLINT`、`INT`、`DOUBLE`、`DATE`、`DATETIME`、`TIME`、`BIGINT`、`UNSIGNED BIGINT`、`UNSIGNED INT` 转换为 `CHAR` 或 `VARCHAR` 时出现问题，SAP Sybase IQ 会在 `.iqmsg` 文件中将失败记录为错误或警告。如果 `CONVERSION_ERROR` 选项为 `ON`，SAP Sybase IQ 会将问题报告为错误。如果 `CONVERSION_ERROR` 选项为 `OFF`，将问题报告为警告

---

### 另请参见

- 数据类型转换函数（第 83 页）
- 存储大小（第 355 页）

## 字符串到日期时间转换的兼容性

---

在 SAP Sybase IQ 和 Adaptive Server 中，将字符串转换为日期和时间数据类型时的行为有所区别。

如果将仅包含时间值（不包含日期）的字符串转换为日期/时间数据类型，则 SAP Sybase IQ 和 Adaptive Server 均使用缺省日期 1900 年 1 月 1 日。而 SQL Anywhere 使用当前日期。

### 用法

如果时间的毫秒部分少于 3 位数，则 Adaptive Server 会根据值以句点还是冒号开头而采用不同方式解释值。如果以冒号开头，则值表示秒的千分位。如果以句点开头，则 1 位数表示十分位，2 位数表示百分位，3 位数表示千分位。无论分隔符是什么，SAP Sybase IQ 和 SQL Anywhere 都会以相同的方式解释值。

- Adaptive Server 按下面所显示的方式转换值。

```
12:34:56.7 to 12:34:56.700
12.34.56.78 to 12:34:56.780
12:34:56.789 to 12:34:56.789
12:34:56:7 to 12:34:56.007
12.34.56:78 to 12:34:56.078
12:34:56:789 to 12:34:56.789
```

- 在两种情况下，SAP Sybase IQ 转换毫秒值的方式都与 Adaptive Server 转换以句点开头的值的方式相同：

```
12:34:56.7 to 12:34:56.700
12.34.56.78 to 12:34:56.780
12:34:56.789 to 12:34:56.789
```

```
12:34:56:7 to 12:34:56.700
12.34.56:78 to 12:34:56.780
12:34:56:789 to 12:34:56.789
```

### 另请参见

- 日期和时间数据类型的用法（第 358 页）

## 导出的日期的兼容性

对于某个月的前 9 天以及小于 10 的小时，Adaptive Server 支持使用空格作为第一位数；SAP Sybase IQ 则支持零或空格。

有关 Adaptive Server 支持以及不支持的数据类型的详细信息，请参见管理：装载管理。

## 从 BIT 到 BINARY 数据类型的转换

SAP Sybase IQ 支持 BIT 到 BINARY 以及 BIT 到 VARBINARY 的隐式转换和显式转换，并且与支持这些转换的 Adaptive Server 兼容。

对于比较运算符、算术运算符以及 INSERT 和 UPDATE 语句，SAP Sybase IQ 支持数据类型 BIT 到 BINARY 以及 BIT 到 VARBINARY 的隐式转换。

对于从 BIT 到 BINARY 的转换，位值 'b' 将复制到二进制字符串的第一个字节，其余字节将用零填补。例如，BIT 值 1 转换为 BINARY(n) 字符串 0x0100...00（拥有 2<sup>n</sup> 个半字节）。BIT 值 0 转换为 BINARY 字符串 0x00...00。

对于从 BIT 到 VARBINARY 的转换，BIT 值 'b' 将复制到 BINARY 字符串的第一个字节，并且不使用其余字节；即，只使用一个字节。例如，BIT 值 1 转换为 VARBINARY(n) 字符串 0x01（拥有 2 个半字节）。

对于从 BIT 到 BINARY 以及从 BIT 到 VARBINARY 数据类型的隐式转换和显式转换，其结果是相同的。下表包含从 BIT 到 BINARY 和 VARBINARY 的转换示例。

将 BIT 值 “1” 转换为	结果
BINARY(3)	0x010000
VARBINARY(3)	0x01
BINARY(8)	0x0100000000000000
VARBINARY(8)	0x01

以下示例说明了从 BIT 到 BINARY 以及从 BIT 到 VARBINARY 数据类型的隐式转换和显式转换。

假定有以下表和数据：

```
CREATE TABLE tbin(c1 BINARY(9)) CREATE TABLE tvarbin(c2
VARBINARY(9)) CREATE TABLE tbar(c2 BIT) INSERT tbar VALUES (1) INSERT
tbar VALUES (0)
```

从 BIT 到 BINARY 的隐式转换:

```
INSERT tbin SELECT c2 FROM tbar

c1
---
0x010000000000000000000000 (18 nibbles)
0x000000000000000000000000 (18 nibbles)
```

从 BIT 到 VARBINARY 的隐式转换:

```
INSERT tvarbin SELECT c2 FROM tbar

c2
---
0x01
0x00
```

从 BIT 到 BINARY 的显式转换:

```
INSERT tbin SELECT CONVERT (BINARY(9), c2) FROM tbar

c1
---
0x010000000000000000000000 (18 nibbles)
0x000000000000000000000000 (18 nibbles)
```

从 BIT 到 VARBINARY 的显式转换:

```
INSERT tvarbin SELECT CONVERT (VARBINARY(9), c2) FROM tbar

c2
---
0x01
0x00
```

## **BIT 和 CHAR/VARCHAR 数据类型之间的转换**

对于比较运算符、算术运算符以及 **INSERT** 和 **UPDATE** 语句，SAP Sybase IQ 支持数据类型 BIT 与 CHAR 以及 BIT 与 VARCHAR 的隐式转换

以下示例说明了 BIT 与 CHAR 以及 BIT 与 VARCHAR 数据类型之间的隐式转换和显式转换。

假定有以下表和数据:

```
CREATE TABLE tchar(c1 CHAR(9)) CREATE TABLE tvarchar(c2 VARCHAR(9))
CREATE TABLE tbar(c2 BIT) CREATE TABLE tbit(c2 BIT) INSERT tbar
VALUES(1) INSERT tbar VALUES(0)
```

从 BIT 到 VARCHAR/从 VARCHAR 到 BIT 的隐式转换以及从 BIT 到 VARCHAR 的隐式转换:

```
INSERT tvarchar SELECT c2 FROM tbar
SELECT c2, char_length(c2) FROM tvarchar

c2,char_length(tvarchar.c2)
-----
```

```
'1',1
'0',1
```

从 VARCHAR 到 BIT 的隐式转换:

```
INSERT tbit SELECT c2 FROM tvarchar
SELECT c2 FROM tbit
```

```
c2
--
0
1
```

从 BIT 到 CHAR/从 CHAR 到 BIT 的显式转换以及从 BIT 到 CHAR 的显式转换:

```
INSERT tchar SELECT CONVERT (CHAR(9), c2) FROM tbar
SELECT c1, char_length(c1) FROM tchar
```

```
c1,char_length(tchar.c1)
-----
'1',9
'0',9
```

从 CHAR 到 BIT 的显式转换:

```
INSERT tbit SELECT CONVERT (BIT, c1) FROM tchar
SELECT c2 FROM tbit
```

```
c2
--
0
1
```

从 BIT 到 VARCHAR/从 VARCHAR 到 BIT 的显式转换以及从 BIT 到 VARCHAR 的显式转换:

```
INSERT tvarchar SELECT CONVERT (VARCHAR(9), c2)
FROM tbar
SELECT c2, char_length(c2) FROM tvarchar
```

```
c2,char_length(tvarchar.c2)
-----
'1',1
'0',1
```

从 VARCHAR 到 BIT 的显式转换:

```
INSERT tbit SELECT CONVERT (BIT, c2) FROM tvarchar
SELECT c2 FROM tbit
```

```
c2
--
0
1
```

## 与其它 SQL 术语的区别

SAP Sybase IQ 符合 ANSI SQL89 标准，但其中很多附加功能是按照 IBM DB2 和 SAA 规范以及 ANSI SQL92 标准定义的。

某些 SAP Sybase IQ 功能是一些其它 SQL 实现未能提供的。

### 日期

SAP Sybase IQ 具有日期、时间和时间戳类型，包括年、月、日、小时、分钟、秒和秒的小数值。插入或更新日期字段或者与日期字段进行比较时，支持使用自由格式日期。

此外，还可以对日期进行以下运算：

表 61. 日期运算

日期运算	描述
日期 + 整数	在日期中加上指定的天数。
日期 - 整数	从日期中减去指定的天数。
日期 - 日期	计算两个日期之间的天数。
日期 + 时间	用日期和时间创建一个时间戳。

另外，还提供了许多用于操作日期和时间的函数。

### 完整性

SAP Sybase IQ 同时支持实体完整性和参照完整性。

该支持通过对 **CREATE TABLE** 和 **ALTER TABLE** 语句进行以下两种扩展实现。

```
PRIMARY KEY ( column-name, ... )
[NOT NULL] FOREIGN KEY [role-name]
    [(column-name, ...)]
    REFERENCES table-name [(column-name, ...)]
    [ CHECK ON COMMIT ]
```

**PRIMARY KEY** 子句声明关系的主键。然后 SAP Sybase IQ 将实施主键的惟一性，并确保主键中的列不包含 **NULL** 值。

**FOREIGN KEY** 子句定义此表与另一个表之间的关系。这个关系由此表中的一列（或多列）表示，这些列必须包含另一个表的主键中的值。系统随后将确保这些列的参照

## 与其它 SQL 术语的区别

完整性；只要修改这些列或在此表中插入行，便会对这些列进行检查，以确保有一列或多列为 **NULL**，或这些值与另一个表的主键中某行的对应列相匹配。

## 连接

---

SAP Sybase IQ 允许在表之间使用 **automatic joins**。

除其它实现中支持的 **NATURAL** 和 **OUTER** 连接运算符之外，SAP Sybase IQ 还允许在表之间基于外键关系建立 **KEY** 连接。这降低了执行连接时 **WHERE** 子句的复杂程度。

## 更新

---

SAP Sybase IQ 允许 **UPDATE** 引用多个表。

此外，还可以更新在多个表上定义的视图。许多 SQL 实现都不允许更新连接的表。

## 更改表

---

**ALTER TABLE** 已扩展。

除用于实现实体完整性和参照完整性的更改外，还支持以下类型的变更：

```
DELETE column
RENAME new-table-name
RENAME old-column TO new-column
```

**提示：** 创建列后，则无法修改列数据类型。要更改数据类型，需将列删除然后再使用正确的数据类型重新创建。

---

## 有时不允许使用子查询

---

与 SQL Anywhere 不同，SAP Sybase IQ 不允许子查询出现在允许使用表达式的任何位置。

SAP Sybase IQ 仅支持 SQL-1989 语法中允许的子查询，以及顶级查询块的 **SELECT** 列表或 **UPDATE** 语句的 **SET** 子句中允许的子查询。SAP Sybase IQ 不支持 SQL Anywhere 扩展。

许多 SQL 实现只允许将子查询放置在比较运算符的右侧。例如，以下命令在 SAP Sybase IQ 中有效，但在大多数其它 SQL 实现中无效。

```
SELECT      SurName,
            BirthDate,
            ( SELECT DepartmentName
              FROM Departments
```



```
WHERE DepartmentID = Employees.EmployeeID  
AND DepartmentID = 200 )  
FROM Employees
```

## 附加函数

---

SAP Sybase IQ 支持几个不在 ANSI SQL 定义中的函数。

另请参见

- SQL 函数 (第 75 页)

## 游标

---

使用嵌入式 SQL 时，可以使用 **FETCH** 语句随意移动游标位置。可以相对于当前游标位置前移或后移游标，也可以从游标开头或末尾向前或向后移动给定数量的记录。

与其它 SQL 术语的区别

## 物理限制

SAP Sybase IQ 数据库中的对象大小和对象数量存在一定的限制。在大多数情况下，计算机内存和磁盘驱动器是主要的限制因素。

有关仅适用于一个平台的限制，请参见该平台特定的文档。

**表 62. 数据库对象大小和数量限制**

项目	限制
Catalog 文件大小	所有平台的最大值均为 1TB。带 NTFS 的 Windows 系统支持 1TB 的最大大小。
数据库名称大小	250 字节
数据库大小	数据库最大大小约等于文件数乘以特定平台上的文件大小，具体取决于最大磁盘配置。 有关影响文件最大数目的内核参数，请参见操作系统文档。
Dbfile 大小	文件系统设备：最大大小为 4TB 操作系统文件：操作系统支持的最大大小
dbspace 大小	每个 dbspace 最多有 2000 个 dbfile。 不建议在 NAS（网络附加存储）设备上创建 dbspace。
字段大小	针对 BINARY 为 255 字节，针对 VARBINARY 为 32,767 字节 针对 CHAR、VARCHAR 为 32,767 针对 LONG BINARY 和 LONG VARCHAR，上限为 512 TB（适用于 128 KB 页）或 1 PB（适用于 512 KB 页）
IQ 页大小	必须介于 64KB 与 512KB 之间。
最大键大小	对于单列索引为 255 字节。对于多列索引为 5300 字节。
字符串文字的最大长度	32KB
SQL 语句的最大长度	SQL 语句的最大长度限制在可用于 IQ 目录的内存量以及目录堆栈的大小。 如果 SQL 语句较长，可使用 <b>-gss</b> 增加目录堆栈大小，并使用 <b>-c</b> 或 <b>-ch</b> 加 <b>-cl</b> 的组合增加目录内存高速缓存大小。 在打印错误消息中的 SQL 语句时，文本限制为 IQ 目录页大小。要打印长命令，可以使用增加的 <b>-gp</b> 设置启动服务器，不过通常 SAP 建议使用 <b>-gp</b> 的缺省值 4096。

项目	限制
可变长度的 FILLER 列的最大长度	512 字节
Dbfile 的最大数目	可以打开的文件总数 取决于操作系统可以支持的 唯一文件描述符数量。
最大用户数（已连接的并发用户）	1000（64 位平台 AIX、HP、Linux 和 Sun Solaris）。 200（Windows 上的 64 位平台）。
临时提取文件的最大大小	由 TEMP_EXTRACT_SIZE <sub>n</sub> 选项设置。平台限制如下： AIX 和 HP-UX: 0 - 64GB Sun Solaris: 0 - 512GB Windows: 0 - 128GB Linux: 0 - 512GB
Table_name 大小	128 个字节
每个表的列数	SAP Sybase IQ 最多支持一个表中包含 45,000 列。如果表中列数多于 10,000，则性能会下降。每个表中允许 NULL 的列数存在限制，最多为大约 $8 * (\text{database-page-size} - 30)$ 。
Column_name 大小	128 个字节
每个数据库的事件数	$2^{31} - 1 = 2\,147\,483\,647$
每个数据库的文件数	用户可以调整的操作系统限制；例如，可使用 NOFILE 进行调整。每个数据库通常可包含 2047 个文件
索引数	每个表有 $2^{32}$ （约 4,000,000,000）个
Index_name 大小	128 个字节
每个表的行数	受表大小的限制，上限为 $2^{48} - 1$ 。
每个数据库的存储过程数	$2^{32} - 1 = 4\,294\,967\,295$
一个 FROM 子句中的表数或视图数	在打开连接优化程序的情况下为 16 至 64，具体取决于查询。
每个查询的引用表数或视图数	512
每个数据库的表数	4,293,918,719
每个事务的引用表数	无限制。
每个查询的 UNION 分支数	512. 如果每个分支在 FROM 子句中有多个表，每个查询的表数限制将使允许的 UNION 分支数减少。
一个 IN 列表中的值数	250,000

项目	限制
行宽	SAP 建议限制为页大小的一半
表大小	受数据库大小的限制。

物理限制

# 系统过程

在 SAP Sybase IQ 数据库中使用系统提供的存储过程来检索系统信息。

SAP Sybase IQ 包括以下几种系统过程：

- 作为存储过程实现的系统函数。
- 目录存储过程，用于以表格形式显示系统信息。
- 用于 Multiplex 服务器操作的 Multiplex 存储过程，包括上述两类过程。请参见《管理：Multiplex》中的系统过程。
- Transact-SQL 系统和类过程。

专门与大对象数据相关的系统存储过程（包括 `sp_iqsetcompression` 和 `sp_iqshowcompression`）在《非结构化数据分析》的存储过程支持中介绍。

## 管理特许系统过程执行

---

特许系统过程可以两种安全模型运行。每个模型都会授予以不同方式运行系统过程的权限。

**注意：** 以下信息只适用于 SAP Sybase IQ 特许系统过程，不适合用户定义的存储过程。

---

第一个模型称为 **SYSTEM PROCEDURE DEFINER** 模型，以特许系统过程所有者（通常是 `dbo`）特权来运行该过程。第二个模型称为 **SYSTEM PROCEDURE INVOKER** 模型，以执行特许系统过程的人员的特权来运行该过程。

要使用 **SYSTEM PROCEDURE DEFINER** 模型来运行特许系统过程，需要授予对该过程的显式 **EXECUTE** 对象级特权。运行系统过程的任何基础已授权任务所需的所有系统特权都自动从所有者（系统过程的定义者）继承。

对于使用 **SYSTEM PROCEDURE INVOKER** 模型的特许系统过程，系统会向 **PUBLIC** 角色授予 **EXECUTE** 对象级特权，由于在缺省情况下，每个用户都是 **PUBLIC** 角色的成员，因此，每个用户都会自动继承 **EXECUTE** 对象级特权。但是，由于 **PUBLIC** 角色并不是系统过程的所有者，并且未被授予任何系统特权，因此，必须将运行任何基础已授权任务所需的系统特权直接或间接授予该用户。

缺省情况下，16.0 及更高版本中所创建的数据库使用 **SYSTEM PROCEDURE INVOKER** 模型来运行所有特许系统过程。16.0 之前版本中以及升级到 16.0 或更高版本的版本中所创建的数据库组合使用 **SYSTEM PROCEDURE DEFINER** 和 **SYSTEM PROCEDURE INVOKER** 模型来运行特许系统过程。在组合模型中，所有 16.0 之前版本的特许系统过程都使用 **SYSTEM PROCEDURE DEFINER** 模型，而在 16.0（或任何之后的版本）中引入的所有特许系统过程都使用 **SYSTEM PROCEDURE INVOKER** 模型。在创建或升级数据库期间，可以替换缺省安全模型，也可以在此后

的任何时间更改此模型。但 SAP 不建议这么做，因为这可能会造成自定义存储过程和应用程序的功能丢失。

当使用 **SYSTEM PROCEDURE DEFINER** 模型运行特许系统过程，**dbo** 系统角色通常是过程所有者。缺省情况下，为 **dbo** 系统角色授予 **SYS\_AUTH\_DBA\_ROLE** 兼容性角色。这样便可确保为该角色间接授予执行系统过程所需的所有特权。迁移 **SYS\_AUTH\_DBA\_ROLE** 兼容性角色可导致 **dbo** 系统角色无法执行特许系统过程。有关详细信息，请参阅“迁移兼容性角色对系统角色的影响”。

## 迁移兼容性角色的影响

通过兼容性角色的成员资格向一些系统角色间接授予执行特许任务所需的系统特权。

无法撤消兼容性角色的基础系统特权；必须先将其迁移到用户定义角色。只有这样才能撤消新角色的各个基础系统特权，并根据组织的安全要求将其授予其它用户定义的角色。这会实施职责分离。

可自动或手动迁移兼容性角色。迁移方式可以影响 系统角色 继续执行已授权任务的能力。

无论使用哪种迁移方法，删除兼容性角色后，如果撤消新用户定义角色的系统特权并将其授予另一个用户定义角色，则必须执行以下其中一种操作以确保系统角色保留执行适当的特许任务：

- 将从已迁移的用户定义角色撤消的每个系统特权直接授予系统角色；或
- 将获得系统特权的用户定义角色的成员资格授予系统角色。

作为兼容性角色成员且可能受迁移影响的系统角色包括：

系统角色	兼容性角色
dbo	SYS_AUTH_DBA_ROLE
	SYS_AUTH_RESOURCE_ROLE
SYS_RUN_REPLICATION_ROLE	SYS_AUTH_DBA_ROLE

### 自动迁移

**ALTER ROLE** 语句可创建新的用户定义角色，自动将兼容性角色的所有基础系统特权授予新的用户定义角色，使兼容性角色的每个成员成为新用户定义角色的成员，然后删除该兼容性角色。

自动迁移假设目标用户定义角色尚不存在，并且所有系统特权都将迁移到相同的新用户定义角色。

### 手动迁移

使用 **CREATE ROLE** 语句创建新的用户定义角色。使用 **GRANT** 语句为一个或多个用户或角色授予各个基础系统特权。在将所有的基础系统特权授予至少一个其他用户或角色后，使用 **DROP** 语句删除兼容性角色。



不会为所迁移兼容性角色的成员自动授予新用户定义角色的成员资格。因此，删除兼容性角色后，有些系统角色的成员可能无法再执行预期的特许任务。必须将新用户定义角色的成员资格授予受影响的系统角色，或者直接为受影响的成员授予所需的系统特权。

## 授予运行特许系统过程的权限

授予运行特许系统过程的权限所用的进程取决于该过程以哪种安全模型运行。

对于使用 **SYSTEM PROCEDURE DEFINER** 模型的特许系统过程，将系统过程的 **EXECUTE** 对象级特权授予用户：

```
GRANT EXECUTE ON sys_procedure_name
TO grantee [,...]
```

对于使用 **SYSTEM PROCEDURE INVOKER** 模型的特许系统过程，将系统过程所需的基础系统特权授予用户。使用 **sp\_proc\_priv()** 可标识运行系统过程所需的系统特权。

```
GRANT system_privilege_name
TO grantee [,...]
```

## 撤消运行特许系统过程的权限

撤消运行特许系统过程的权限所用的进程取决于该过程以哪种安全模型运行。

对于使用 **SYSTEM PROCEDURE DEFINER** 模型的特许系统过程，撤消用户对系统过程的 **EXECUTE** 对象级特权：

```
REVOKE EXECUTE ON sys_procedure_name
FROM grantee [,...]
```

对于使用 **SYSTEM PROCEDURE INVOKER** 模型的特许系统过程，撤消用户具有的系统过程所需的基础系统特权：

```
REVOKE system_privilege_name
FROM grantee [,...]
```

## 确定数据库所使用的安全模型

有两种安全模型可供数据库使用。

要确定数据库正在使用的安全模型，请执行：

```
select IF ((HEXTOINT(substring(db_property('Capabilities'),
1,length(db_property('Capabilities'))-20)) & 8) = 8)
THEN 1
ELSE 0
END IF
```

1 表示数据库正在使用 **SYSTEM PROCEDURE INVOKER** 模型。0 表示数据库正在使用组合模型。

在组合模型中，只有 16.0 之前版本的特许系统过程才使用 **SYSTEM PROCEDURE DEFINER** 来运行。请参考 16.0 之前版本的特许系统过程列表来标识这些系统过程。

无法将新的或升级的 16.0 版本及更高版本的数据库配置为使用 **SYSTEM PROCEDURE DEFINER** 模型来运行所有系统过程。

### **16.0 之前版本的特许系统过程**

16.0 之前版本的特许系统过程列表。

#### *使用组合安全模型的特许系统过程*

对于这些特许系统过程，如果将数据库配置为使用 **SYSTEM PROCEDURE DEFINER**，您只需要该过程的 **EXECUTE** 对象级特权便可运行它。如果将数据库配置为使用 **SYSTEM PROCEDURE INVOKER**，您还需要具有运行每个过程所需的各个系统特权。有关运行各个系统过程所需的系统特权，请参见《参考：构件块、表和过程》指南。

<ul style="list-style-type: none"> <li>• sa_audit_string</li> <li>• sa_checkpoint_execute</li> <li>• sa_disable_auditing_type</li> <li>• sa_disk_free_space</li> <li>• sa_enable_auditing_type</li> <li>• sa_external_library_unload</li> <li>• sa_flush_cache</li> <li>• sa_list_external_library</li> <li>• sa_server_option</li> <li>• sa_procedure_profile</li> <li>• sa_procedure_profile_summary</li> <li>• sa_table_page_usage</li> <li>• sa_validate</li> <li>• sp_iq_reset_identity</li> <li>• sp_iqaddlogin</li> <li>• sp_iqbackupdetails</li> <li>• sp_iqbackupsummary</li> <li>• sp_iqcardinality_analysis</li> <li>• sp_iqcheckdb</li> <li>• sp_iqcheckoptions</li> <li>• sp_iqclient_lookup</li> <li>• sp_iqcolumn</li> <li>• sp_iqcolumnuse</li> <li>• sp_iqconnection</li> <li>• sp_iqconstraint</li> <li>• sp_iqcontext</li> <li>• sp_iqconstraint</li> <li>• sp_iqcontext</li> <li>• sp_iqcursorinfo</li> <li>• sp_iqdatatype</li> <li>• sp_iqdbsize</li> </ul>	<ul style="list-style-type: none"> <li>• sp_iqdbspace</li> <li>• sp_iqdbspaceinfo</li> <li>• sp_iqdbspaceobjectinfo</li> <li>• sp_iqdbstatistics</li> <li>• sp_iqdroplogin</li> <li>• sp_iqemptyfile</li> <li>• sp_iqestdbspaces</li> <li>• sp_iqestspace</li> <li>• sp_iqevent</li> <li>• sp_iqfile</li> <li>• sp_iqhelp</li> <li>• sp_iqindex</li> <li>• sp_iqindex_alt</li> <li>• sp_iqindexadvice</li> <li>• sp_iqindexfragmentation</li> <li>• sp_iqindexinfo</li> <li>• sp_iqindexmetadata</li> <li>• sp_iqindexsize</li> <li>• sp_iqindexuse</li> <li>• sp_iqlmconfig</li> <li>• sp_iqlocks</li> <li>• sp_iqmodifyadmin</li> <li>• sp_iqmodifylogin</li> <li>• sp_iqmpxcheckdqpconfig</li> <li>• sp_iqmpxdumpltvlog</li> <li>• sp_iqmpxfilestatus</li> <li>• sp_iqmpxinconnpoolinfo</li> <li>• sp_iqmpxincheartbeatinfo</li> <li>• sp_iqcopyloginpolicy</li> <li>• sp_iqmpxinconnpoolinfo</li> <li>• sp_iqmpxincheartbeatinfo</li> </ul>	<ul style="list-style-type: none"> <li>• sp_iqmpxinfo</li> <li>• sp_iqmpxversioninfo</li> <li>• sp_iqobjectinfo</li> <li>• sp_iqkeys</li> <li>• sp_iqprocedure</li> <li>• sp_iqprocparm</li> <li>• sp_iqrebuildindex</li> <li>• sp_iqrename</li> <li>• sp_iqrestoreaction</li> <li>• sp_iqrowdensity</li> <li>• sp_iqsetcompression</li> <li>• sp_iqsharedtempdistrib</li> <li>• sp_iqshowcompression</li> <li>• sp_iqshowpsex</li> <li>• sp_iqspaceinfo</li> <li>• sp_iqspaceused</li> <li>• sp_iqstatistics</li> <li>• sp_iqstatus</li> <li>• sp_iqsysmon</li> <li>• sp_iqtable</li> <li>• sp_iqtablesize</li> <li>• sp_iqtableuse</li> <li>• sp_iqtransaction</li> <li>• sp_iqunusedcolumn</li> <li>• sp_iqunusedindex</li> <li>• sp_iqunusedtable</li> <li>• sp_iqversionuse</li> <li>• sp_iqview</li> <li>• sp_iqwho</li> <li>• sp_iqworkmon</li> </ul>
---	---	--

### 使用调用者特权的特许系统过程

16.0 之前版本的这些特许系统过程以运行该过程的用户（而不是过程的所有者）的权限来运行，而不考虑安全模型设置如何。因此，除了需要对系统过程的 EXECUTE 对象级特权（缺省情况下通过 PUBLIC 角色的成员资格授予）之外，还必须为用户授予系统过程所需的其它系统特权。有关运行各个系统过程所需的系统特权，请参见《参考：构件块、表和过程》指南。

- sa\_describe\_shapefile
- sa\_get\_user\_status
- sa\_locks
- sa\_performance\_diagnostics
- sa\_report\_deadlocks
- sa\_text\_index\_stats

## 存储过程的语法规则

调用存储过程时括号和引号的使用情况不同，具体取决于是在 Interactive SQL 中所做的那样直接输入过程名称还是使用 **CALL** 语句调用它。

允许一些变体，因为产品支持 SAP Sybase IQ SQL 和 Transact-SQL 语法。如果需要 Transact-SQL 兼容性，请确保使用 Transact-SQL 语法。

表 63. 存储过程语法变化形式

语法	语法类型	解释
<code>procedure_name ('param')</code>	SAP Sybase IQ	如果将参数包含在括号中，则需要使用引号。
<code>procedure_name 'param'</code>	SAP Sybase IQ	如果将参数包含在引号中，则括号是可选项。
<code>procedure_name param</code>	Transact-SQL	<p>如果省略 参数两侧的引号，则还必须省略括号。</p> <p><b>注意：</b> 指定所有者时，参数始终需要用引号引起来。例如，假设所有者为 <i>dba</i>，</p> <pre>iqtablesize 'dba.emp1' <span style="float:right">SP_</span></pre> <p>需要用引号将参数引起来，而</p> <pre>iqtablesize emp1 <span style="float:right">SP_</span></pre> <p>则不需要。</p>

语法	语法类型	解释
<code>procedure_name</code>	SAP Sybase IQ 或 Transact- SQL	如果在 <b>Interactive SQL</b> 中不使用参数直接运行过程，以及当过程不包含参数时，请使用此语法。
<code>call procedure_name (param='value')</code>	SAP Sybase IQ	使用此语法可以调用传递参数值的过程。

使用 **Transact-SQL** 存储过程时，必须使用 **Transact-SQL** 语法。

## 了解存储过程报告的统计信息

许多存储过程都会报告过程执行时 **SAP Sybase IQ** 的状态信息。

这意味着可以获取快照视图。例如，用于列出某连接使用的空间的报告列将只显示在执行过程时使用的空间，而不是该连接使用的最大空间。

要监控 **SAP Sybase IQ** 在更长时段内的使用情况，请使用 **SAP Sybase IQ** 监控器，此监控器将按您指定的时间间隔在监控器启动到停止时段内收集和报告统计信息。

## SQL Anywhere 支持过程

**SAP Sybase IQ** 支持 **SQL Anywhere** 系统过程。

**提示：** **SQL Anywhere** 存储过程的过程名内不包含 **iq**。

**sa\_get\_table\_definition** 过程仅支持 **SQL Anywhere** 表。如果针对 **SAP Sybase IQ** 表运行，该过程会返回错误 **not implemented for IQ tables**。

## 按字母顺序排列的系统存储过程列表

系统存储过程用于执行 **IQ** 主存储区中的系统管理员任务。

系统存储过程由用户 **ID dbo** 所有。

**注意：** 缺省情况下，由 **Interactive SQL Classic** 显示的列值的最大长度是 30 个字符。这可能不足以显示诸如 **sp\_iqstatus** 这样的存储过程的输出。若要避免截断输出，可增加长度，方法是从“**Interactive SQL**”菜单中选择“命令”>“选项”，然后为“限制显示列”和/或“限制输出列”选择并输入较大的值。

## sp\_iqaddlogin 过程

向指定的登录策略添加一个新的 SAP Sybase IQ 用户帐户。

### 语法 1

```
call sp_iqaddlogin ( 'username_in' , 'pwd' ,
[ , 'password_expiry_on_next_login' ] [ , 'policy_name' ] )
```

### 语法 2

```
sp_iqaddlogin 'username_in' , 'pwd' ,
[ , 'password_expiry_on_next_login' ] [ , 'policy_name' ]
```

### 语法 3

```
sp_iqaddlogin username_in, pwd, [ password_expiry_on_next_login ] [ ,
policy_name ]
```

### 参数

- **username\_in** - 用户的登录名。登录名必须符合标识符的规则。
- **pwd** - 用户的口令。口令必须符合口令规则，即它们必须是有效标识符。
- **password\_expiry\_on\_next\_login** - (可选) 指定是否在创建用户登录后该用户的口令立即到期。缺省设置为 OFF (口令不到期)。
- **policy\_name** - (可选) 按照命名登录策略创建用户。如果未指定，则按照根登录策略创建用户。

使用 **sp\_iqaddlogin** 创建、并设置为一天后到期的 *username\_in/pwd* 第二天全天有效，随后一天无效。换言之，今天创建并设置为 *n* 天后到期的登录在日期变为第 (*n+1*) 天时将不再可用。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY USER 系统特权。

### 注释

添加新的 SAP Sybase IQ 用户帐户，为该用户分配登录策略，并将该用户添加到 ISYSUSER 系统表中。如果此用户已经拥有针对数据库的用户 ID，但不在 ISYSUSER 中 (例如，如果是使用 **GRANT CONNECT** 语句或 SAP Control Center 添加的)，则 **sp\_iqaddlogin** 会将该用户添加到此表中。

如果在调用过程时未指定登录策略名，SAP Sybase IQ 会为用户分配根登录策略。

---

**注意：** 如果登录策略的最大登录数没有限制，则属于该登录策略的用户可以有无限数目的连接。

---

第一次用户登录将强制更改口令，并为新创建的用户分配登录策略。使用 **CREATE USER** 创建新用户，但为了能够向后兼容，仍支持 **sp\_iqaddlogin**。

### 示例

以下调用将按照 `expired_password` 登录策略添加用户 `rose`，口令为 `irk324`。本示例假设 `expired_password` 登录策略已存在。

```
call sp_iqaddlogin('rose', 'irk324', 'ON', 'expired_password')
```

```
sp_iqaddlogin 'rose','irk324', 'ON', 'expired_password'
```

### 另请参见

- `sp_expireallpasswords` 系统过程 (第 628 页)
- `sp_iqcopyloginpolicy` 过程 (第 413 页)
- `sp_iqmodifylogin` 过程 (第 469 页)
- `sp_iqpassword` 过程 (第 484 页)
- `sp_iqdroplogin` 过程 (第 431 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqbackupdetails 过程

显示特定备份中包括的所有 `dbfile`。

### 语法

```
sp_iqbackupdetails backup_id
```

### 参数

- **backup\_id** - 指定备份操作的事务标识符。

**注意：** 通过执行以下查询可获得 `SYSIQBACKUPHISTORY` 表中的 `backup_id` 值：

```
select * from sysiqbackuphistory
```

### 特权

您必须具有系统过程的 `EXECUTE` 特权。

### 注释

`sp_iqbackupdetails` 返回：

表 64. `sp_iqbackupdetails` 列

列名	描述
<code>backup_id</code>	备份事务的标识符。
<code>backup_time</code>	备份的时间。

列名	描述
backup_type	备份的类型: "full"、"Incremental since incremental" 或 "Incremental since full"。
selective_type	备份的子类型: "All inclusive"、"All RW files in RW dbspaces"、"Set of RO dbspace/file"。
depends_on_id	备份所依赖的上次备份的标识符。
dbspace_id	正在备份的 dbspace 的标识符。
dbspace_name	如果 dbspace 名与给定 dbspace_id 的 SYSDBSPACE 中的 dbspace 名相匹配, 则为 SYSIQBACKUPHISTORYDETAIL 的 dbspace 名。否则为 "null"。
dbspace_rwstatus	"ReadWrite" 或 "Read Only"。
dbspace_createid	DbSPACE 创建的事务标识符。
dbspace_alterid	变更 DBSPACE 读写模式的事务标识符。
dbspace_online	状态 "Online" 或 "Offline"。
dbspace_size	备份时 dbspace 的大小 (KB)。
dbspace_backup_size	在 dbspace 中备份的数据的大小 (KB)。
dbfile_id	要备份的 dbfile 的标识符。
dbfile_name	如果备份操作后未重命名, 则为逻辑文件名。如果重命名, 则为 "null"。
dbfile_rwstatus	"ReadWrite" 或 "Read Only"。
dbfile_createid	Dbfile 创建的事务标识符。
dbfile_alterid	变更 FILE 读写模式的变更 DBSPACE 的事务标识符。
dbfile_size in MB	dbfile 的大小 (MB)。
dbfile_backup_size	dbfile 备份的大小 (KB)。
dbfile_path	SYSBACKUPDETAIL 的 dbfile 路径, 如果路径与给定 dbspace_id 和 dbfile_id 的 SYSDBFILE 中的物理文件路径 ("file_name") 相匹配, 则为 SYSBACKUPDETAIL 的 dbfile 路径。否则为 "null"。

### 示例

**sp\_iqbackupdetails** 的输出样本:

```
backup_id  backup_time  backup_type  selective_type  d
depends_on_id
```



```

883      2008-09-23 13:58:49.0    Full      All
inclusive      0

dbspace_id      dbspace_name      dbspace_rwstatus      dbspace_createid
0               system          ReadWrite              0

dbspace_alterid  dbspace_online  dbspace_size  dbspace_backup_size
dbfile_id
0               0               2884          2884          0

dbfile_name      dbfile_rwstatus  dbfile_createid  dbfile_alterid
dbfile_size
system          ReadWrite              0               0               2884

dbfile_backup_size  dbfile_path
2884 C:\\Documents and Settings\\All Users\\SybaseIQ\\
\\demo\\iqdemo.db

```

### 另请参见

- [SYSIQBACKUPHISTORY 系统视图](#) (第 682 页)
- [确定数据库所使用的安全模型](#) (第 381 页)

## sp\_iqbackupsummary 过程

总结执行的备份操作。

### 语法

```
sp_iqbackupsummary [ timestamp or backup_id ]
```

### 参数

- **timestamp 或 backup\_id** - 指定报告备份操作的间隔。如果指定时间戳或备份 ID，则只返回 backup\_time 大于或等于输入时间的那些记录。如果不指定时间戳，该过程返回 ISYSIQBACKUPHISTORY 中的所有备份记录。

### 特权

您必须具有系统过程的 EXECUTE 特权。

### 注释

**表 65. sp\_iqbackupsummary 列**

列名	描述
backup_id	备份事务的标识符
backup_time	备份的时间

列名	描述
backup_type	备份的类型: “Full”、 “Incremental since incremental” 或 “Incremental since full”
selective_type	备份的子类型: "All Inclusive"、 "All RW files in RW dbspaces"、 "Set of RO dbspace/file"
virtual_type	虚拟备份的类型: “Non-virtual”、 “Decoupled” 或 “Encapsulated”
depends_on_id	备份所依赖的备份的标识符
creator	备份的创建者
backup_size	备份的大小 (KB)
user_comment	用户注释
backup_command	发出的备份语句 (减去注释)

示例

**sp\_iqbackupsummary** 的输出样本:

```

backup_id      backup_time      backup_type      selective_type    v
virtual_type
      883    2008-09-23 13:58:49.0    Full              All inclusive     Non
virtual

depends_on_id   creator    backup_size    user_comment    backup_command
          0    DBA           10864              backup database to
                                     'c:\\\\temp
\\\\\\b1'
```

另请参见

- 确定数据库所使用的安全模型 (第 381 页)

### **sp\_iqcardinality\_analysis** 过程

分析表中列的基数。

**注意:** **sp\_iqcardinality\_analysis** 不再返回索引类型值或索引建议。建议用户运行索引顾问以获得关于附加列索引的建议。不建议使用 **sp\_iqcardinality\_analysis**, 将在未来版本中将其移除。

语法

```
sp_iqcardinality_analysis ([ 'table_name' ], [ 'table_owner' ], [ 'script' ])
```

参数

- **table\_name** - 表的名称。

- **table\_owner** – 表所有者的姓名。如果未指定此参数，那么该过程将查找当前用户所拥有的表。
- **script** – 脚本：
  - table\_name
  - table\_owner
  - column\_name
  - cardinality
  - index\_type
  - index recommendation

#### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种：

- SELECT ANY TABLE 系统特权
- 您拥有该表

此外，您必须还要至少拥有以下系统特权之一：

- CREATE ANY INDEX
- ALTER ANY INDEX
- CREATE ANY OBJECT
- ALTER ANY OBJECT

#### 注释

如果不指定任何参数，则 SAP Sybase IQ 将对当前用户拥有的所有表中的所有列显示 **create\_index** SQL 语句。

如果指定 *script*，则可以重新定向输出以生成脚本文件：

```
OUTPUT TO 'indexfile.sql' FORMAT ASCII QUOTE '';
```

#### 示例

```
sp_iqcardinality_analysis 'Departments', 'GROUPO'
```

table_name	table_owner	column_name	cardinality	index type	Index Recommendation
Departments	GROUPO	DepartmentID	5		运行索引顾问
Departments	GROUPO	DepartmentName	5		运行索引顾问
Departments	GROUPO	DepartmentHeadID	5		运行索引顾问

#### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqcheckdb 过程

检查当前数据库的有效性。（可选）更正 `dbspace` 或数据库的分配问题。如果分区数据存在于脱机 `dbspace` 上，则 `sp_iqcheckdb` 不检查分区表。

`sp_iqcheckdb` 读取数据库中的所有存储。成功完成后，系统将更新数据库空闲列表（内部分配映射），以反映数据库的真实存储分配情况。然后，`sp_iqcheckdb` 生成一个报告，列出它已经执行的操作。

如果发现错误，则 `sp_iqcheckdb` 会报告对象的名称和错误的类型。如果检测到错误，`sp_iqcheckdb` 不会更新空闲列表。

使用 `sp_iqcheckdb` 还可以检查指定的表、索引、索引类型或整个数据库的一致性。

---

**注意：** `sp_iqcheckdb` 是 SAP Sybase IQ 数据库一致性检查程序 (DBCC) 的用户界面，有时称为 **DBCC**。

---

### 语法

```
sp_iqcheckdb 'mode target [ ... ] [ resources resource-percent ]'

mode:
  { allocation
  | check
  | verify }
  | dropleaks

target:
  [ indextype index-type [...] ] database
  | database resetclocks
  | { [ indextype index-type ] [...] table table-name [ partition partition-
name ] [...]
  | index index-name
  | [...] dbspace dbspace-name}
  | cache main-cache-name
```

有三种检查数据库一致性的模式，以及一种重置分配映射的模式。如果参数字符串中既没有指定模式也没有指定目标，则 SAP Sybase IQ 会返回错误消息：

```
At least one mode and target must be specified to DBCC.
```

### 参数

- **database** - 如果目标是数据库，则所有 `dbspace` 都必须处于联机状态。
- **index-type** - 以下索引类型之一：**FP**、**CMP**、**LF**、**HG**、**HNG**、**WD**、**DATE**、**TIME**、**DTTM**、**TEXT**。

如果指定的 `index-type` 在目标中不存在，则返回错误消息。如果指定多个索引类型，并且目标仅包含这些索引类型中的一部分，则 `sp_iqcheckdb` 将处理现有索引类型。

- **index-name** - 可能包含所有者和表限定符：[[owner.]table-name.]index-name

如果未指定 *owner*，则系统将以该顺序替代当前用户及数据库所有者 (dbo)。如果未指定 *table*，则 *index-name* 必须唯一。

- **table-name** - 可能包含所有者限定符: [owner.]table-name

如果未指定 *owner*，则系统将以该顺序替代当前用户及数据库所有者 (dbo)。*table-name* 不能是临时表或预连接表。

---

**注意：** 如果表名称或索引名称包含空格，请将 *table-name* 或 *index-name* 参数用双引号引起来：

```
sp_iqcheckdb 'check index "dbo.sstab.i2" resources 75'
```

- **partition-name** - *partition-name* 参数不包含限定符。如果它包含空格，请将其放在双引号内。

分区过滤器会使 **sp\_iqcheckdb** 检查属于该分区的相应表行的子集。当表只有一个分区时，表中的分区过滤器和没有分区过滤器的表目标在语义上是等同的。

- **dbspace-name** - *dbspace-name* 参数不包含限定符。如果它包含空格，请将其放在双引号内。

dbspace 目标检查属于该 dbspace 的数据库页的子集。dbspace 必须联机。当表只有一个 dbspace 时，dbspace 和数据库目标在语义上是等同的。

- **resource-percent** - 输入参数 *resource-percent* 必须为大于零的整数。通过控制相对于 CPU 数的线程数，资源百分比可用来限制数据库一致性检查程序的 CPU 利用率。如果 *resource-percent* = 100 (缺省值)，则为每个 CPU 创建一个线程。如果 *resource-percent* > 100，则有比 CPU 数更多的线程，对于某些计算机配置，这可能会提高性能。线程数的最小值为 1。
- **main-cache-name** - 高速缓存目标将主高速缓存 dbspace 中的页面与 IQ 主存储库中的原始页面进行比较。

---

**注意：** **sp\_iqcheckdb** 参数字符串必须用单引号引起来，并且长度不能超过 255 字节。

分配问题可以在沙漏模式下进行修复。

---

适用于

Simplex 和 Multiplex。

特权

您必须具有系统过程的 EXECUTE 特权，以及 ALTER DATABASE 系统特权。

注释

**sp\_iqcheckdb** 检查数据库中每个块的分配情况，并在当前会话中保存该信息，直至发出下一个 **sp\_iqdbstatistics** 过程。**sp\_iqdbstatistics** 显示最近一次执行 **sp\_iqcheckdb** 的最新结果。

**sp\_iqcheckdb** 可以执行几种不同的功能，具体取决于指定的参数。

模式	描述
分配	<p>根据整个数据库、特定索引、特定索引类型、特定分区、特定表或特定 <b>dbspace</b> 的块映射信息检查分配情况。不检查索引一致性。</p> <p>检测重复块（两个或多个对象声明对其的所有权的块）或额外块（对象所拥有的未分配块）。</p> <p>检测数据库或 <b>dbspace</b> 目标的泄漏块（在指定目标中任何对象都未声明的已分配块）。</p> <p>当目标为分区表时，<b>分配模式</b>：</p> <ul style="list-style-type: none"> <li>• 检查所有表分区分配位图的元数据</li> <li>• 检查表分配位图的元数据</li> <li>• 验证块映射条目与表的分配位图是否一致</li> <li>• 验证表的分区分配位图没有重叠</li> <li>• 检查在表的分区分配位图中定义的行是否构成表的存在位图的超集</li> <li>• 检查在表的分区分配位图中定义的行是否构成表的分配位图的超集</li> <li>• 验证主高速缓存页与 <b>IQ</b> 主存储库页是否一致。</li> </ul> <hr/> <p><b>注意：</b>如果在输入参数字符串中指定单个索引、索引类型或表的名称，则 <b>sp_iqcheckdb</b> 不能检查所有分配问题。</p> <hr/> <p>以分配模式运行：</p> <ul style="list-style-type: none"> <li>• 检测重复块或无主块（使用数据库或特定表或索引作为目标）</li> <li>• 如果遇到页头错误</li> </ul> <p><b>DBCC</b> 选项 <b>resetclocks</b> 仅用于分配模式。<b>resetclocks</b> 与强制恢复一起使用，以将 <b>Multiplex</b> 辅助服务器转换成协调器。有关 <b>Multiplex</b> 功能的信息，请参见《管理：<b>Multiplex</b>》。一旦内部数据库版本控制时钟落后，<b>resetclocks</b> 将更正这些时钟的值。除非咨询过 <b>SAP Sybase IQ</b> 技术支持，否则请勿将 <b>resetclocks</b> 选项用于任何其它目的。</p> <p><b>resetclocks</b> 选项必须在单用户模式下运行，并且只能与 <b>DBCC</b> 语句 <b>allocation database</b> 一起使用。<b>resetclocks</b> 的语法为：</p> <pre>sp_iqcheckdb 'allocation database resetclocks'</pre>
检查	<p>验证可以读取整个数据库、主高速缓存、特定索引、特定索引类型、特定表、特定分区或特定 <b>dbspace</b> 的所有数据库页。如果表已分区，则检查模式将检查表的分区分配位图。</p> <p>如果在运行查询时返回元数据、空计数或非重复计数错误，则会在检查模式下运行。</p>

模式	描述
验证	<p>验证整个数据库、主高速缓存、特定索引、特定索引类型、特定表、特定分区或特定 dbspace 的非 FP 索引及其相应 FP 索引的内容。如果指定的目标包含 FP 及相应的非 FP 索引的所有数据页，则验证模式会检测以下不一致问题：</p> <ul style="list-style-type: none"> <li>• <b>缺少键</b> - 某键在 FP 中，但不在非 FP 索引中。</li> <li>• <b>额外键</b> - 某键在非 FP 索引中，但不在 FP 索引中。</li> <li>• <b>缺少行</b> - 某行在 FP 中，但不在非 FP 索引中。</li> <li>• <b>额外行</b> - 某行在非 FP 索引中，但不在 FP 索引中。</li> </ul> <p>如果指定的目标仅包含 FP 页的一个子集，则验证模式仅可以检测以下不一致问题：</p> <ul style="list-style-type: none"> <li>• 缺少键</li> <li>• 缺少行</li> </ul> <p>如果目标是分区表，则验证模式还将验证是否已将表或表分区中的每一行分配给正确的分区。</p> <p>如果在运行查询时返回元数据、空计数或非重复计数错误，则会在验证模式下运行。</p> <p><b>注意：</b> <code>sp_iqcheckdb</code> 不检查参照完整性或修复参照完整性违规。</p>
沙漏	<p>当 SAP Sybase IQ 服务器以单节点模式运行时，您可以将沙漏模式与数据库或 dbspace 目标结合使用，以重置整个数据库或指定 dbspace 目标的分配映射。如果目标是 dbspace，则沙漏操作还必须阻止对已命名 dbspace 的读写操作。数据库或 dbspace 列表中的所有 dbspace 都必须处于联机状态。</p> <p>在 Multiplex 协调器节点上，沙漏模式还检测整个 Multiplex 中的泄漏块、重复块或额外块。</p>

DBCC 性能：

DBCC 的执行时间根据进行全面数据库检查的数据库的大小、指定的表或索引的数目以及计算机的大小而有所不同。如果仅检查数据库的子集（即仅检查指定的表、索引或索引类型），则需要的时间要比检查整个数据库少。

`sp_iqcheckdb` 沙漏模式的处理时间取决于 dbspace 目标数。

下表总结了四种 `sp_iqcheckdb` 模式的操作和输出。

表 66. `sp_iqcheckdb` 模式的操作和输出

模式	检测到的错误	输出	速度
分配	分配错误	仅限分配统计信息	每小时 4 TB

模式	检测到的错误	输出	速度
检查	分配错误 大部分索引错误	所有可用统计信息	每小时 60 GB
验证	分配错误 所有索引错误	所有可用统计信息	每小时 15 GB
沙漏	分配错误	仅限分配统计信息	每小时 4 TB

输出:

根据执行模式, **sp\_iqcheckdb** 的输出包括摘要结果、错误、信息性统计信息和修复统计信息。如果在单个会话中指定多个模式, 则输出最多可能包含三个结果集。错误统计信息用星号(\*\*\*\*\*)表示, 并且仅当检测到错误时才会显示。

**sp\_iqcheckdb** 的输出也将被复制到 SAP Sybase IQ 消息文件 .iqmsg。如果 **DBCC\_LOG\_PROGRESS** 选项是 ON, 则 **sp\_iqcheckdb** 会将进度消息发送到此 IQ 消息文件, 从而使用户可以跟踪 DBCC 操作的执行进度。

*示例*

检查整个数据库的分配:

```
sp_iqcheckdb 'allocation database'
```

对索引 i1、i2 和 dbo.t1.i3 执行详细检查。如果不指定新模式, 则 **sp\_iqcheckdb** 将对剩余的目标应用相同模式, 如以下命令所示:

```
sp_iqcheckdb 'verify index i1 index i2 index dbo.t1.i3'
```

可以组合所有模式, 并可以在单个会话中对数据库运行多项检查。对表 t2 中的分区 p1 执行快速检查, 对索引 i1 执行详细检查, 并使用一半 CPU 对整个数据库执行分配检查:

```
sp_iqcheckdb 'check table t2 partition p1 verify index i1 allocation database resources 50'
```

检查数据库中类型为 **FP** 的所有索引:

```
sp_iqcheckdb 'check indextype FP database'
```

验证表 t1 中的 **FP** 和 **HG** 索引以及表 t2 中的 **LF** 索引:

```
sp_iqcheckdb 'verify indextype FP indextype HG table t1 indextype LF table t2'
```

检查 LVC 单元不一致问题:

```
sp_iqcheckdb 'check index EFG2JKL.ASIQ_IDX_T208_C504_FP'
-----
Index Statistics:
** Inconsistent Index: abcd.EFG2JKL.ASIQ_IDX_T208_C504_FP ***** FP
```



```
Indexes Checked: 1
** Unowned LVC Cells: 212 *****
```

**sp\_iqcheckdb** LVC 单元消息包括:

- 无主 LVC 单元
- 重复的 LVC 单元行
- 未分配的 LVC 单元行

这些消息表明与 VARCHAR、VARBINARY、LONG BINARY (BLOB) 或 LONG VARCHAR (CLOB) 列的不一致。无主 LVC 单元表示少量的不可用磁盘空间，可安全地忽略。重复和未分配的 LVC 单元为严重的错误，只能通过删除已损坏的列解决。

若要删除已损坏的列，请使用旧列的副本创建一个新列，然后删除原始列并将新列重命名为旧列。

---

**注意:** LVC 是一个宽度大于 255 的 VARCHAR 或 VARBINARY 列。LONG BINARY (BLOB) 和 LONG VARCHAR (CLOB) 也使用 LVC。

---

输出示例:

运行 **sp\_iqcheckdb 'allocation database':**

```
=====
DBCC Allocation Mode Report
=====
      DBCC Status                               No Errors Detected
=====
Allocation Summary
=====
      Blocks Total                               25600
      Blocks in Current Version                 5917
      Blocks in All Versions                   5917
      Blocks in Use                             5917
      % Blocks in Use                           23
=====
Allocation Statistics
=====
      Marked Logical Blocks                     8320
      Marked Physical Blocks                   5917
      Marked Pages                             520
      Blocks in Freelist                       2071196
      Imaginary Blocks                         2014079
      Highest PBN in Use                       1049285
      Total Free Blocks                        19683
      Usable Free Blocks                       19382
      % Total Space Fragmented                 1
      % Free Space Fragmented                 1
      Max Blocks Per Page                     16
      1 Block Page Count                      165
      3 Block Page Count                      200
      4 Block Page Count                      1
      10 Block Page Count                     1
      16 Block Page Count                     153
```

2	Block Hole Count	1
3	Block Hole Count	19
6	Block Hole Count	12
7	Block Hole Count	1
10	Block Hole Count	1
15	Block Hole Count	1
16	Block Hole Count	1220
Partition Summary		
	Database Objects Checked	2
	Blockmap Identity Count	2
	Bitmap Count	2
=====		
Connection Statistics		
=====		
	Sort Records	3260
	Sort Sets	2
=====		
DBCC Info		
=====		
	DBCC Work units Dispatched	197
	DBCC Work units Completed	197
	DBCC Buffer Quota	255
	DBCC Per-Thread Buffer Quota	255
	Max Blockmap ID found	200
	Max Transaction ID found	404

**注意：** 报告可指示泄漏的空间。泄漏的空间是按照数据库空闲列表（内部分配映射）分配的块，但 DBCC 发现它不属于任何数据库对象。

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqcheckoptions 过程

对于已连接的用户，**sp\_iqcheckoptions** 将显示已更改为非缺省值的数据库和服务器启动选项的当前值和缺省值列表。

### 语法

**sp\_iqcheckoptions**

### 特权

您必须具有系统过程的 EXECUTE 特权。

DBA 用户会看到为所有角色和用户设置的所有永久性选项，以及为 DBA 设置的临时选项。非 DBA 用户只能看到他们自己的临时选项。所有用户都会看到非缺省的服务器启动选项。

### 注释

为已更改为非缺省值的每个选项返回一行。输出将先按选项名然后再按用户名排序。

对于已连接的用户，**sp\_iqcheckoptions** 存储过程显示已更改为非缺省值的数据库和服务器启动选项的当前值和缺省值的列表。**sp\_iqcheckoptions** 会考虑所有 SAP Sybase IQ 和 SQL Anywhere 数据库选项。SAP Sybase IQ 修改了一些 SQL Anywhere 选项的缺省值，这些经过修改的值将成为新的缺省值。除非再次更改新的 SAP Sybase IQ 缺省值，否则 **sp\_iqcheckoptions** 不会列出选项。

运行 **sp\_iqcheckoptions** 时，DBA 可看到为所有角色和用户设置的所有永久性选项，以及为 DBA 设置的临时选项。非 DBA 用户只能看到他们自己的临时选项。所有用户都会看到非缺省的服务器启动选项。

表 67. **sp\_iqcheckoptions** 列

列名	描述
User_name	已设置选项的用户或角色的名称。创建数据库时，系统将为 PUBLIC 角色设置所有选项。为除 PUBLIC 以外的角色或用户设置的所有选项都将显示出来。
Option_name	选项的名称。
Current_value	选项的当前值。
Default_value	选项的缺省值。
Option_type	对 TEMPORARY 选项是“Temporary”，其它则是“Permanent”。

#### 示例

在这些示例中，临时选项 APPEND\_LOAD 设置为 ON，角色 myrole 的选项 MAX\_WARNINGS 设置为 9。用户 joel 的 MAX\_WARNINGS 设置为临时值 55。

在第一个示例中，由 DBA 运行 **sp\_iqcheckoptions**。

User_name	Option_name	Current_value	Default_value	Option_type
DBA	Ansi_update_constr	CURSORS	Off	Permanent
PUBLIC	Ansi_update_constr	Cursors	Off	Permanent
DBA	Checkpoint_time	20	60	Temporary
DBA	Connection_authent	Company=MyComp;		Temporary
DBA	Login_procedure	Application=DBTools;Signa	DBA.sp_iq_proce	Permanent
PUBLIC	Login_procedure	DBA.sp_iq_proce	sp_login_envir	Permanent
myrole	Max Warnings	9	281474976710655	Permanent
DBA	Thread_count	25	0	Temporary

在第二个示例中，由用户 joel 运行 **sp\_iqcheckoptions**。

User_name	Option_name	Current_value	Default_value	Option_type
joel	Ansi_update_constr	CURSORS	Off	Permanent
PUBLIC	Ansi_update_constr	Cursors	Off	Permanent
joel	Checkpoint_time	20	60	Temporary

joel	Connection_authent	Company=MyComp; Application=DBTools;Signa	Temporary
joel	Login_procedure	DBA.sp_iq_proce	sp_login_envir Perma
joel	Login_procedure	DBA.sp_iq_proce	sp_login_envir Perma
joel	Max_Warnings	55	281474976710655 Temporary
joel	Thread_count	25	0 Temporary

## sp\_iqclient\_lookup 过程

客户端应用程序可以确定负责某特定数据流的 SAP Sybase IQ 用户帐户，如源自某特定客户端 IP 地址/端口的网络分析程序中所示。

### 语法

```
sp_iqclient_lookup [ 'IPAddress' ], [ Port ], [ UserID ]
```

### 参数

- **IPAddress** - 指定始发客户端应用程序的 IP 地址。
- **Port** - 指定始发客户端应用程序的端口号。
- **UserID** - 指定 SAP Sybase IQ 用户 ID。

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种 系统特权：

- SELECT ANY TABLE
- MONITOR
- DROP CONNECTION
- SERVER OPERATOR

### 注释

**sp\_iqclient\_lookup** 过程获取客户端 IP 地址和端口号，然后返回一个包含 Number（连接 ID）、IPAddress、Port 和 UserID 的行。

```
1> sp_iqclient_lookup '158.76.235.71',3360
2> go
```

Number	IPAddress	Port	UserID
15	158.76.235.71	3360	rdeniro

也可以传递第三个参数以仅选择 UserID。如果没有传递任何参数，sp\_iqclient\_lookup 将返回所有当前登录名及其 IP 地址和端口号。例如：

```
sp_iqclient_lookup
```

Number	IPAddress	Port	UserID
11	162.66.131.36	2082	mbrando
21	162.66.100.233	1863	apacino

```

22      162.66.100.206    8080    jcaan
23      162.66.100.119    6901    rduvall
24      162.66.100.125    7001    dkeaton
25      162.66.100.124    6347    jcazale

```

```

(6 rows affected)
(return status = 0)

```

如果某个客户端应用程序并未使用 **TCP/IP** 或者不是用于内部连接，它的地址将显示为 127.0.0.1。

---

**注意：** 此信息仅对已登录用户可用。服务器中不为此保留任何历史登录数据。

---

### 副作用

**sp\_iqclient\_lookup** 存储过程可能会影响服务器的性能，这将随不同安装而有所不同。要找到登录名就需要对服务器上的所有当前活动连接进行扫描；因此，具有大量连接的服务器所受到的影响将会更大。此外，此信息无法保存在高速缓存中，因为它是动态信息（有时甚至为高动态）。因此，本地系统管理员需要管理这个存储过程的使用，并对任何其它使用服务器工具的客户端应用程序监控其对服务器的影响。

### 示例

显示 UserID jcazale 的 IP 地址：

```
sp_iqclient_lookup null, null, jcazale
```

Number	IPAddress	Port	UserID
11	162.66.131.36	2082	jcazale
15	164.66.131.36	1078	jcazale

显示客户端 IP 162.66.131.36 的 IP 地址：

```
sp_iqclient_lookup '162.66.131.36'
```

Number	IPAddress	Port	UserID
11	162.66.131.36	2082	jcazale
12	162.66.131.36	1078	jcaan

---

**注意：** 当用户指定了不正确的参数时，此结果为空。

---

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## **sp\_iqcolumn** 过程

显示有关数据库中的列的信息。

### 语法 1

```
sp_iqcolumn ( [ table_name ], [ table_owner ], [ table_loc ] )
```

### 语法 2

```
sp_iqcolumn [ table_name='table_name' ], [ table_owner='tableowner' ],
[table_loc='table_loc' ]
```

### 参数

- **语法 1** - 如果您指定 *table\_owner* 而不指定 *table\_name*, 则必须用 NULL 代替 *table\_name*。例如, sp\_iqcolumn NULL, DBA。
- **语法 2** - 可按任意顺序指定参数。将 '*table\_name*' 和 '*table\_owner*' 用单引号引起来。

### 特权

您必须具有系统过程的 EXECUTE 特权。

### 注释

显示有关数据库中的列的信息。如果指定 *table\_name* 参数, 则仅返回具有该名称的表的列。如果指定 *table\_owner* 参数, 则仅返回该用户拥有的表。如果同时指定 *table\_name* 和 *table\_owner* 参数, 则选择唯一表中的列 (如果该表存在)。如果指定 *table\_loc*, 则仅返回用该段类型定义的表。如果不指定任何参数, 则返回数据库中所有表的所有列。**sp\_iqcolumn** 不返回系统表的列信息。

列名	描述
table_name	表的名称。
table_owner	表的所有者。
column_name	列的名称。
domain_name	数据类型。
width	具有精度和标度的数值数据类型的精度或不具有标度的数值数据类型的存储宽度; 字符数据类型的宽度。
scale	数值数据类型的标度。
nulls	如果列可以包含 NULL 则为 "Y", 如果列不能包含 NULL 则为 "N"。
default	如果列是标识/自动增量列, 则为 'Identity/Autoincrement', 如果不是, 则为空。
cardinality	按索引进行的非重复计数 (如果已知)。
location	TEMP = IQ 临时存储库, MAIN = IQ 主存储库, SYSTEM = 目录存储库。
isPartitioned	如果列属于分区表并有一个或多个分区, 而列分区的 dbspace 与表分区的 dbspace 不同, 则为 "Y", 如果列的表没有分区或列的每个分区驻留在与表分区相同的 dbspace 中, 则为 "N"。

列名	描述
remarks	使用 <b>COMMENT</b> 语句添加的用户注释。
check	检查约束表达式。

### 另请参见

- `sp_iqconstraint` 过程 (第 409 页)
- `sp_iqdatatype` 过程 (第 415 页)
- `sp_iqevent` 过程 (第 436 页)
- `sp_iqhelp` 过程 (第 441 页)
- `sp_iqindex` 和 `sp_iqindex_alt` 过程 (第 447 页)
- `sp_iqpkeys` 过程 (第 485 页)
- `sp_iqprocparm` 过程 (第 489 页)
- `sp_iq_reset_identity` 过程 (第 499 页)
- `sp_iqtable` 过程 (第 539 页)
- `sp_iqview` 过程 (第 554 页)
- 确定数据库所使用的安全模型 (第 381 页)

### **sp\_iqcolumn** 过程示例

使用示例作为 `sp_iqcolumn` 用法的参考。

以下两种语法变体都会返回 `Departments` 表中的所有列:

```
sp_iqcolumn Departments
```

```
call sp_iqcolumn (table_name='Departments')
```

table_name	table_owner	column_name	domain_name	width	scale
Departments	GROUPO	DepartmentID	integer	4	0
		DepartmentName	char	40	0
Departments	GROUPO	DepartmentHead	integer	4	0
cardinality	location	isPartitioned	remarks	check	
5	Main	N	(NULL)	(NULL)	
0	Main	N	(NULL)	(NULL)	
0	Main	N	(NULL)	(NULL)	

以下语法变体会返回表所有者 `DBA` 拥有的所有表中的所有列。

```
sp_iqcolumn table_owner='DBA'
```

## sp\_iqcolumnmetadata 过程

返回有关一个或多个表中列索引的详细信息。

*语法*

**sp\_iqcolumnmetadata** [ *table.name* [, *owner-name* ] ]

*特权*

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种:

- ALTER ANY INDEX 系统特权
- ALTER ANY OBJECT 系统特权
- 表的 REFERENCE 权限
- 您拥有该表

*注释*

**sp\_iqcolumnmetadata** 可读取索引元数据以返回有关基表和全局临时表中列索引的详细信息。所报告的全局临时表的索引元数据用于该表的单个实例。

包括可选的 [*table.name*] 参数以生成该表的详细信息。忽略 [*table.name*] 参数以生成数据库中所有表的详细信息。

## sp\_iqcolumnuse 过程

报告负载访问的列的详细使用信息。

*语法*

**sp\_iqcolumnuse**

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

*注释*

不会报告在 SYSTEM 中创建的表的列。

**表 68. sp\_iqcolumnuse 列**

列名	描述
TableName	表名称
ColumnName	列名
所有者 (Owner)	列所有者的用户名
UID**	列唯一标识符



列名	描述
LastDT	上次访问的日期/时间
NRef	查询引用数

\*\*UID 是系统分配的用于唯一标识列实例的编号（实例是在创建对象时定义的）。

**提示：** INDEX\_ADVISOR 选项会生成消息，建议增加列索引以提高一个或多个查询的性能。

### 示例

**sp\_iqcolumnuse** 过程的输出样本：

TableName	NRef	ColumnName	Owner	UID	LastDT
orders	13	o_orderdate	DBA		20070917
orders	13	o_shippriority	DBA		20070917
lineitem	13	l_orderkey	DBA		20070917
lineitem	13	l_extendedp..	DBA		20070917
lineitem	13	l_discount	DBA		20070917
lineitem	13	l_shipdate	DBA		20070917
#tmp1	1	expression	DBA	10000000001218	20070917
#tmp1	1	expression	DBA	10000000001222	20070917
...					

**注意：** 上述示例中的较长编号为临时 ID。

### 另请参见

- sp\_iqindexadvice 过程（第 450 页）
- sp\_iqindexuse 过程（第 461 页）
- sp\_iqtableuse 过程（第 544 页）
- sp\_iqunusedcolumn 过程（第 549 页）
- sp\_iqunusedindex 过程（第 550 页）
- sp\_iqunusedtable 过程（第 551 页）
- sp\_iqworkmon 过程（第 559 页）

## **sp\_iqconnection** 过程

显示有关连接和版本的信息，包括哪些用户正在使用临时 `dbspace`、哪些用户正在使版本保持活动状态、连接在 SAP Sybase IQ 内部执行哪些操作、连接状态、数据库版本状态，等等。

### 语法

```
sp_iqconnection [ connhandle ]
```

### 适用于

Simplex 和 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- DROP CONNECTION
- MONITOR
- SERVER OPERATOR

### 注释

*connhandle* 相当于 Number 连接属性，是连接的 ID 编号。`connection_property` 系统函数返回连接 ID：

```
SELECT connection_property ( 'Number' )
```

当使用有效的 *connhandle* 的输入参数进行调用时，`sp_iqconnection` 仅对该连接返回一行。

`sp_iqconnection` 为每个活动连接都返回一行。列 ConnHandle、Name、Userid、LastReqTime、ReqType、CommLink、NodeAddr 和 LastIdle 分别是连接属性 Number、Name、Userid、LastReqTime、ReqType、CommLink、NodeAddr 和 LastIdle，并返回与系统函数 `sa_conn_info` 相同的值。其它列返回来自 SAP Sybase IQ 引擎的 SAP Sybase IQ 端的连接数据。各行按 ConnCreateTime 排序。

MPXServerName 列存储与节点间通信 (INC) 有关的信息，如下所示：

运行服务器	MPXServerName 列内容
Simplex 服务器	NULL (所有连接均为本地/用户连接)
Multiplex 协调器	<ul style="list-style-type: none"> <li>• NULL 用于本地/用户连接。</li> <li>• 包含每个 INC 连接 (根据需要或专用活动连接) 的辅助节点服务器名称 (连接源)。</li> </ul>

运行服务器	MPXServerName 列内容
Multiplex 辅助服务器	<ul style="list-style-type: none"> <li>• NULL 用于本地/用户连接。</li> <li>• 包含协调器服务器名的值（连接源）。</li> </ul>

在 Java 应用程序中，请在 "RemotePWD" 字段中指定来自 TDS 客户端的特定于 SAP Sybase IQ 的连接属性。以下示例显示如何指定 IQ 特定连接参数，其中的 **myconnection** 为 IQ 连接名：

```
p.put("RemotePWD", "", "CON=myconnection");
```

列名	描述
ConnHandle	连接的 ID 号。
名称	服务器的名称。
Userid	连接的用户 ID。
LastReqTime	指定连接的上次请求开始的时间。
ReqType	表示上次请求类型的字符串。
IQCmdType	在 SAP Sybase IQ 端执行的当前命令（如果有）。该命令类型反映在引擎的实现级别定义的命令。这些命令由事务命令、用于处理 IQ 存储库中的数据的 DDL 和 DML 命令、内部 IQ 游标命令和特殊控制命令（如 <b>OPEN</b> 和 <b>CLOSE</b> 、 <b>BACKUP DATABASE</b> 、 <b>RESTORE DATABASE</b> ）以及其它命令组成。
LastIQCmdTime	此连接上最后一个 IQ 命令在 SAP Sybase IQ 引擎 IQ 端上的启动或完成时间。
IQCursors	此连接上在 IQ 存储库中打开的游标数。
LowestIQCursorState	IQ 游标状态（如果有）。如果连接上存在多个游标，则显示的状态是所有游标中位置最靠下游标的状态，即距离完成时间最远的那个游标的状态。游标状态反映 SAP Sybase IQ 内部实现的详细信息，并可能在未来发生更改。对于此版本，游标状态包括：NONE、INITIALIZED、PARSED、DESCRIBED、COSTED、PREPARED、EXECUTED、FETCHING、END_OF_DATA、CLOSED 和 COMPLETED。就像名称所暗示的那样，游标状态在操作结束时发生更改。例如，PREPARED 状态指示游标正在执行。
IQthreads	当前分配给连接的 SAP Sybase IQ 线程数。某些线程可能已分配，但仍处于空闲状态。此列可以帮助您确定哪些连接使用了最多的资源。
TxnID	连接上当前事务的事务 ID。该事务 ID 与 .iqmsg 文件中 BeginTxn、CmtTxn 和 PostCmtTxn 消息显示的事务 ID 以及打开数据库时所记录的 Txn ID Seq 相同。

列名	描述
ConnCreateTime	连接的创建时间。
TempTableSpaceKB	此连接在处理 IQ 临时表中存储的数据时所用的 IQ 临时存储空间的字节数 (KB)。
TempWorkSpaceKB	此连接在处理诸如排序、散列和临时位图时所用的 IQ 临时存储空间的字节数 (KB)。由位图或由属于 SAP Sybase IQ 临时表索引一部分的其它对象所使用的空间将反映在 TempTableSpaceKB 中。
IQConnID	作为 .iqmsg 文件中所有消息的一部分显示的十位连接 ID。它是一个单调递增整数，在整个服务器会话内唯一。
satoiq_count	内部计数器，用于显示从 SAP Sybase IQ 引擎的 SQL Anywhere 端到 IQ 端的相交数。这在确定连接活动时有可能会用到。结果集将在行缓冲区中返回，但不会每行都增加一次 satoiq_count 或 iqtosa_count。
iqtosa_count	内部计数器，用于显示从 SAP Sybase IQ 引擎的 IQ 端到 SQL Anywhere 端的相交数。这在确定连接活动时有可能会用到。
CommLink	连接的通信链接。这是 SAP Sybase IQ 所支持的网络协议之一，如果为相同计算机连接，则为 local。
NodeAddr	客户端/服务器连接中客户端的节点。
LastIdle	请求间隔时间数。
MPXServerName	如果是 INC 连接，则 varchar(128) 值包含发起 INC 连接的 Multiplex 服务器的名称。如果不是 INC 连接，则为 NULL。
LSName	连接的逻辑服务器名。如果逻辑服务器上下文未知或不适用，则为 NULL。
INCConnName	某个用户连接的基础 INC 连接的名称。此列的数据类型为 varchar(255)。如果 <b>sp_iqconnection</b> 显示某个已挂起用户连接的 INC 连接名，则该用户连接将有一个同样挂起的关联 INC 连接。
INCConnSuspended	此列中的 "Y" 值表示某个用户连接的基础 INC 连接处于挂起状态。"N" 值则表示该连接未挂起。

示例

**sp\_iqconnection**

```

ConnHandle      Name      Userid      LastReqTime      ReqType
=====
1  'SQL_DBC_100525210'  'DBA'      '2011-03-28 09:29:24.466'  'OPEN'

      IQCmdType      LastIQCmdTime      IQCursors      LowestIQCursorState
=====
'IQUTILITYOPENCURSOR'  2011-03-28 09:29:24.0      0      'NONE'
    
```

```

IQthreads      TxnID          ConnCreateTime  TempTableSpaceKB  TempWorkSpaceKB
=====
0      3352568      2011-03-28 09:29:20.0      0      0

IQconnID  satoiq_count  iqtosa_count  CommLink  NodeAdd  LastIdle  MPXServerName
=====
34      43      2  'local'  ''      244  (NULL)

LSName      INCConnName          INCConnSuspended
=====
Finance_LS  'IQ_MFX_SERVER_P54'  'Y'

```

## sp\_iqconstraint 过程

列出使用 **CREATE TABLE** 或 **ALTER TABLE** 为指定表或列定义的参照完整性约束。

### 语法

```
sp_iqconstraint [ 'table-name', 'column-name', 'table-owner' ]
```

### 特权

您必须具有系统过程的 **EXECUTE** 特权。

### 注释

如果省略表名和列名，则报告当前已连接数据库中包括临时表在内的所有表的所有参照完整性约束。该信息包括由 **CREATE TABLE** 和/或 **ALTER TABLE** 语句定义的唯一或主键约束、参照约束和关联角色名称。

### 示例

下面是一个示例输出，其中显示所有表中候选键或外键包含所有者 bob 的 ck1 列的所有主键/外键对：

```
call sp_iqconstraint('', 'ck1', 'bob')
```

```

PTAB1 bob ASIQ_IDX_T27_HG unique ck1,ck2 selftab bob CK6FK3 Y
ASIQ_IDX_T42_HG ck1,ck2PTAB2 bob ASIQ_IDX_T27_HG unique ck1,ck2
selftab bob CK6FK4 Y
ASIQ_IDX_T206_I42_HG ck1,ck2selftab bob ASIQ_IDX_T26_HG unique
ck1,ck2 selftab bob CK3FK1 Y
ASIQ_IDX_T206_I42_HG ck1,ck2

```

显示的列为：

- 主强制表
- 表所有者
- 候选键索引
- 主键或唯一约束
- 主键列
- 外表
- 外表所有者
- 外键角色名称

- 强制状态（强制为“Y”，未受强制的为“N”）
- 外键索引
- 外键列
- 位置（“TEMP”、“MAIN”或“SYSTEM”）

#### 另请参见

- `sp_iqcolumn` 过程（第 401 页）
- `sp_iqdatatype` 过程（第 415 页）
- `sp_iqevent` 过程（第 436 页）
- `sp_iqhelp` 过程（第 441 页）
- `sp_iqindex` 和 `sp_iqindex_alt` 过程（第 447 页）
- `sp_iqpkeys` 过程（第 485 页）
- `sp_iqprocparm` 过程（第 489 页）
- `sp_iq_reset_identity` 过程（第 499 页）
- `sp_iqtable` 过程（第 539 页）
- `sp_iqview` 过程（第 554 页）
- 确定数据库所使用的安全模型（第 381 页）

## `sp_iqcontext` 过程

按照连接跟踪和显示有关当前执行的语句的信息。

#### 语法

```
sp_iqcontext [ connhandle ]
```

#### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种 系统特权：

- MANAGE ANY USER
- MONITOR

#### 注释

输入参数 *connhandle* 相当于 `Number` 连接属性，是连接的 ID 号。例如，**SELECT CONNECTION\_PROPERTY( 'NUMBER' )**。

当用有效输入参数 *connhandle* 调用时，**sp\_iqcontext** 仅返回该连接的信息。

DBA 使用 **sp\_iqcontext** 可以在任何给定时刻确定哪些语句正在系统上运行，并可以标识发出相应语句的用户和连接。通过此信息，可以使用该实用程序进行以下操作：

- 将语句文本与 **sp\_iqconnection** 中的等效行匹配，以获取有关每个连接的资源使用情况 and 事务信息

- 将语句文本与当 **-zr** 服务器选项设置为 **ALL** 或 **SQL** 时创建的 **SQL** 日志中的等效行匹配
- **SAP Sybase IQ** 可以收集连接信息时，使用连接信息将 **sp\_iqcontext** 中的语句文本与 **.iqmsg** 文件中的等效行进行匹配，后者包含查询计划
- 如果产生语句文本，则将其与 **SAP Sybase IQ** 堆栈跟踪信息 (**stktrc-yyyyymmdd-hhnnss\_#.iq**) 进行匹配
- 将此信息与可能产生的操作系统堆栈跟踪（例如 **Sun Solaris** 上的 **pstack**）进行比较

所收集的语句文本的最大大小是 **Catalog** 存储的页大小。

**表 69. sp\_iqcontext 列**

列名	描述
ConnOrCursor	CONNECTION、CURSOR 或 DQP。
ConnHandle	连接的 ID 号，或者 0（对于 DQP）。
Name	服务器的名称（领导节点名称）。
Userid	连接、游标或 DQP 工作线程的用户 ID。
numIQCursors	如果第 1 列是 CONNECTION，则是在此连接上打开的游标数。 如果第 1 列是 CURSOR，则是按顺序分配给与此连接关联的游标的编号。 如果列 1 为 DQP，则为 0。CONNECTION 也可以返回值 0。
IQthreads	当前分配给连接的 IQ 线程数。可能分配了一些线程，但是它们处于空闲状态。对于 DQP 线程，表示分配给 DQP 工作线程的线程数。
TxnID	当前事务的事务 ID。对于工作线程，表示领导节点的事务 ID。
ConnOrCurCreate-Time	创建此连接、游标或 DQP 工作线程的时间。
IQConnID	作为 .iqmsg 文件中所有消息的一部分显示的连接 ID。它是一个单调递增整数，在整个服务器会话内唯一。
IQGovernPriority	一个表示用户查询的执行队列顺序的值。1 表示高优先级，2（缺省值）表示中优先级，3 表示低优先级。如果值为 -1，则表示 IQGovernPriority 不适用于此操作。使用数据库选项 IQGOVERN_PRIORITY 设置 IQGovernPriority 值。 对于 DQP 连接，此列显示无命令。

列名	描述
CmdLine	执行的用户命令的前 4096 个字符。 对于 DQP 连接，此列显示无命令。
Attributes	正在分布的查询的唯一 ID。

### 示例

以下示例显示的内容摘录自在不含参数的情况下发出 `sp_iqcontext` 时的输出，该输出是当前所有连接产生的结果。出于空间考虑，会截断列名。

```
ConnOrCu.. ConnHandle Name UserId numIQ.. IQthr.. TxnID Conn.. IQcon..
IQGov.. Cmd.. Attributes
CONNECTION 2 sun7bar dbo 0 0 0 2010-08-04 15:15:40.0 15 No command NO
COMMAND
CONNECTION 7 sun7bar dbo 0 0 0 2010-08-04 15:16:00.0 32 No command NO
COMMAND
CONNECTION 10 sun7bar dbo 0 0 0 2010-08-04 15:16:21.0 46 No command NO
COMMAND
...
CONNECTION 229 sun7bar DBA 0 0 1250445 2010-08-05 18:28:16.0 50887 2
select server_name,
inc_state, coordinator_failover from sp_iqmpxinfo() order by server_name
...
DQP 0 dbsrv2873_node_c1DBA 0 1 10000 2010-08-05 18:28:16.0 no command no
command Query ID:
12345; Condition: c1 > 100;
DQP 0 dbsrv2873_node_c1DBA 0 1 10001 2010-08-05 18:28:16.0 no command no
command Query ID:
12346; Node #12 Join (Hash);
```

输出的第一行显示连接 2 (IQ 连接 ID 15)。此连接在服务器 `sun7bar` 上，用户是 `dbo`。发出 `sp_iqcontext` 时，此连接未执行命令。

连接 229 显示正在执行的用户命令（该命令包含的字符少于该列可以显示的最多字符数 4096）。用户命令片段之前的 2 表示这是中优先级查询。

连接句柄（本示例中的第一个连接为 2）标识 `-zr` 日志中的结果。IQ 连接 ID（本示例中的第一个连接为 15）标识 `.iqmsg` 文件中的结果。在 UNIX 系统上，可以使用 `grep` 来定位连接句柄或连接 ID 的所有实例，这便于关联所有来源中的信息。

倒数第二行 (TxnID 10000) 显示 DQP 工作线程。此工作连接正在运行两个不变的条件。

最后一行 (TxnID 10001) 显示连接正在运行散列连接。

### 另请参见

- CONNECTION\_PROPERTY 函数 [系统] (第 152 页)
- sp\_iqshowpsexec 过程 (第 507 页)
- 确定数据库所使用的安全模型 (第 381 页)



## sp\_iqcopyloginpolicy 过程

通过复制现有登录策略创建新的登录策略。

### 语法 1

```
call sp_iqcopyloginpolicy ( 'existing-policy-name' , 'new-policy-name' )
```

### 语法 2

```
sp_iqcopyloginpolicy 'existing-policy-name' , 'new-policy-name'
```

### 参数

- **existing-policy-name** - 要复制的登录策略。
- **new-policy-name** - 要创建的新登录策略的名称 (CHAR(128))。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY LOGIN POLICY 系统特权。

### 示例

从现有登录策略 *root* 中复制登录策略选项值，新建一个登录策略 *lockeduser*:

```
call sp_iqcopyloginpolicy ('root','lockeduser')
```

### 另请参见

- sp\_expireallpasswords 系统过程 (第 628 页)
- sp\_iqaddlogin 过程 (第 386 页)
- sp\_iqmodifylogin 过程 (第 469 页)
- sp\_iqpassword 过程 (第 484 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqcursorinfo 过程

显示有关当前在服务器上打开的游标的详细信息。

### 语法

```
sp_iqcursorinfo [ cursor-name ] [, conn-handle ]
```

### 参数

- **cursor-name** - 游标的名称。如果仅指定此参数，则 **sp\_iqcursorinfo** 将返回所有连接中具有指定名称的所有游标的相关信息。  
要复制的登录策略。

- **conn-handle** – 表示连接 ID 的整数。如果仅指定此参数，则 **sp\_iqcursorinfo** 将返回指定的连接中所有游标的相关信息。

可以在不指定任何参数的情况下调用 **sp\_iqcursorinfo** 过程。如果不指定任何参数，则 **sp\_iqcursorinfo** 将返回当前在服务器上打开的所有游标的相关信息。如果指定两个参数，则 **sp\_iqcursorinfo** 将报告指定连接中具有指定名称的所有游标的相关信息。

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

*注释*

如果不指定第一个参数，而指定第二个参数，则必须用 NULL 替换省略的参数。例如，**sp\_iqcursorinfo** NULL, 1。

**表 70. sp\_iqcursorinfo 用法示例**

语法	输出
<b>sp_iqcursorinfo</b>	显示有关当前在服务器上打开的所有游标的信息
<b>sp_iqcursorinfo</b> 'cursor1'	显示有关在所有连接中名为 cursor1 的所有游标的信息
<b>sp_iqcursorinfo</b> NULL, 3	显示有关连接 3 中的所有游标的信息
<b>sp_iqcursorinfo</b> 'cursor2', 4	显示有关在连接 4 中名为 cursor2 的所有游标的信息

**sp\_iqcursorinfo** 存储过程显示有关在服务器上当前打开的游标的详细信息。数据库管理员通过 **sp\_iqcursorinfo** 可只使用一个存储过程和视图统计信息（例如已更新、删除和插入多少行）来监控游标状态。

如果指定了一个或多个参数，将根据指定参数对结果进行过滤。例如，如果指定 *cursor-name*，则仅显示有关指定游标的信息。如果指定 *conn-handle*，则 **sp\_iqcursorinfo** 将仅返回指定连接中的游标的相关信息。如果没有指定任何参数，则 **sp\_iqcursorinfo** 将显示当前在服务器上打开的所有游标的相关信息。

**sp\_iqcursorinfo** 过程返回以下列中的信息：

**表 71. sp\_iqcursorinfo 列**

列名	描述
名称	游标的名称
ConnHandle	连接的 ID 号
IsUpd	Y 表示游标可更新；否则为 N
IsHold	Y 表示游标是持有游标；否则为 N

列名	描述
IQConnID	作为 .iqmsg 文件中所有消息的一部分显示的十位连接 ID。它是一个单调递增整数，在整个服务器会话内唯一。
UserID	创建和运行游标的用户的用户 ID（或用户名）
CreateTime	游标的创建时间
CurrentRow	游标在结果集中的当前位置
NumFetch	游标提取行的次数。同一行可以多次提取。
NumUpdate	游标更新行的次数（如果游标可更新）。同一行可以多次更新。
NumDelete	游标删除行的次数（如果游标可更新）。
NumInsert	游标插入行的次数（如果游标可更新）。
RWTabOwner	由游标在 RW 模式下打开的表的所有者。
RWTabName	由游标在 RW 模式下打开的表的名称。
CmdLine	用户执行的命令的前 4096 个字符

### 示例

显示有关当前在服务器上打开的所有游标的信息：

```

sp_iqcursorinfo
Name          ConnHandle      IsUpd      IsHold      IQConnID      UserID
-----
-- crsr1          1              Y          N           118           DBA
crsr2          3              N          N           118           DBA
CreateTime          CurrentRow      NumFetch      NumUpdate
-----
2009-06-26 15:24:36.000          19          100000000      200000000
2009-06-26 15:38:38.000          20000      200000000
NumDelete      NumInsert      RWTabOwner      RWTabName      CmdLine
-----
--- 20000000      3000000000          DBA           test1          call
proc1 ()
proc2 ()

```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqdatatype 过程

显示有关系统数据类型和用户定义数据类型的信息。

### 语法

```

sp_iqdatatype [ type-name ], [ type-owner ], [ type-type ]

```

## 参数

- **type-name** - 数据类型的名称。
- **type-owner** - 数据类型的创建者的名称。
- **type-type** - 数据类型的类型。允许的值如下：
  - **SYSTEM**: 仅显示系统定义的数据类型（由用户 `sys` 或 `dbo` 拥有的数据类型）的相关信息
  - **ALL**: 显示用户和系统数据类型的相关信息
  - 其它任何值: 显示有关用户数据类型的信息

可以在不指定任何参数的情况下调用 **sp\_iqdatatype** 过程。如果不指定任何参数，则缺省情况下仅显示用户定义数据类型（不归 `dbo` 或 `sys` 所有的数据类型）的相关信息。

## 特权

您必须具有系统过程的 **EXECUTE** 特权。

## 注释

如果不指定前两个参数中的任何一个参数，但指定序列中的下一个参数，则必须用 **NULL** 替换省略的参数。例如，`sp_iqdatatype NULL, NULL, SYSTEM` 和 `sp_iqdatatype NULL, user1`。

表 72. **sp\_iqdatatype** 用法示例

语法	输出
<code>sp_iqdatatype</code>	显示数据库中所有用户定义数据类型的相关信息
<code>sp_iqdatatype country_t</code>	显示名为 <code>country_t</code> 的用户定义数据类型的相关信息
<code>sp_iqdatatype non_existing_type</code>	不返回任何行，因为数据类型 <code>non_existing_type</code> 不存在
<code>sp_iqdatatype NULL, DBA</code>	显示 <code>DBA</code> 拥有的所有用户定义数据类型的相关信息
<code>sp_iqdatatype country_t, DBA</code>	显示 <code>DBA</code> 拥有的数据类型 <code>country_t</code> 的相关信息
<code>sp_iqdatatype rowid</code>	<code>rowid</code> 是系统定义的数据类型。如果没有同样名为 <code>rowid</code> 的用户定义的数据类型，则不返回任何行。（缺省情况下，仅返回用户定义的数据类型。）
<code>sp_iqdatatype rowid, SYS</code>	不返回任何行，因为数据类型 <code>rowid</code> 不是用户定义的数据类型（缺省情况下，仅返回用户定义的数据类型）

语法	输出
<code>sp_iqdatatype NULL, NULL, SYSTEM</code>	显示所有系统定义的数据类型（由 <code>dbo</code> 或 <code>sys</code> 拥有）的相关信息
<code>sp_iqdatatype rowid, NULL, SYSTEM</code>	显示系统数据类型 <code>rowid</code> 的相关信息
<code>sp_iqdatatype NULL, NULL, 'ALL'</code>	显示用户定义数据类型和系统数据类型的相关信息

**sp\_iqdatatype** 存储过程显示数据库中的系统和用户定义数据类型的相关信息。用户定义数据类型也称为域。**sp\_iqdatatype** 输出中不包括预定义域名。

如果指定了一个或多个参数，系统将根据指定参数对 **sp\_iqdatatype** 结果进行过滤。例如，如果指定 *type-name*，则仅显示有关指定数据类型的信息。如果指定 *type-owner*，则 **sp\_iqdatatype** 仅返回指定所有者拥有的数据类型的相关信息。如果未指定参数，则 **sp\_iqdatatype** 显示数据库中的所有用户定义数据类型的相关信息。

**sp\_iqdatatype** 过程返回以下列中的信息：

表 73. **sp\_iqdatatype** 列

列名	描述
<code>type_name</code>	数据类型的名称
<code>creator</code>	数据类型的所有者
<code>nulls</code>	Y 表示用户定义数据类型允许空值；N 表示数据类型不允许空值，U 表示没有为数据类型指定空值。
<code>width</code>	显示字符串列的长度、数值列的精度以及所有其它数据类型的存储字节数
标度	显示数值数据类型列的小数点后的位数，对于所有其它数据类型该值为零
“default”	数据类型的缺省值
“check”	数据类型的 CHECK 条件

#### 示例

显示用户定义数据类型 `country_t` 的相关信息：

```
sp_iqdatatype country_t
```

```

type_name      creator      nulls      width      scale      “default”      “c
heck”
country_t     DBA         U          15         0          (NULL)        (NULL)

```

#### 另请参见

- `sp_iqcolumn` 过程（第 401 页）

- sp\_iqconstraint 过程 (第 409 页)
- sp\_iqevent 过程 (第 436 页)
- sp\_iqhelp 过程 (第 441 页)
- sp\_iqindex 和 sp\_iqindex\_alt 过程 (第 447 页)
- sp\_iqpkeys 过程 (第 485 页)
- sp\_iqprocparm 过程 (第 489 页)
- sp\_iq\_reset\_identity 过程 (第 499 页)
- sp\_iqtable 过程 (第 539 页)
- sp\_iqview 过程 (第 554 页)
- 确定数据库所使用的安全模型 (第 381 页)

## **sp\_iqdbsize 过程**

显示当前数据库的大小。

### *语法*

```
sp_iqdbsize ( [ main ] )
```

### *适用于*

Simplex 和 Multiplex。

### *特权*

您必须具有系统过程的 EXECUTE 特权，以及 ALTER DATABASE 系统特权。

### *注释*

返回数据库的总大小。此外，还返回当压缩数据库（在磁盘上）时在内存中保留数据库所需的页数和 IQ 页数。

如果在 Multiplex 数据库上运行，缺省参数为返回共享 IQ 存储库大小的 main。

如果在启用 RLV 的表中不存在行的情况下运行，则 Physical Blocks、RLVLogBlocks 和 RLVLogKBytes 列将包含非零条目，而其它列包含零条目。这表示没有行级版本控制的表。

列名	描述
Database	数据库文件的路径名称。
Physical Blocks	数据库总大小（以块为单位）。 IQ 数据库由一个或多个 dbspace 组成。每个 dbspace 都有固定大小，该大小最初以 MB 为单位指定。使用此 IQ 页大小和该 IQ 页大小对应的块大小将此 MB 数量转换为块数。Physical Blocks 列反映每个 SAP Sybase IQ dbspace 大小的累计总数，以块为单位表示。

列名	描述
KBytes	数据库总大小（以千字节为单位）。此值是数据库的总块数（前一 <code>sp_iqdbsize</code> 列中的 <code>Physical Blocks</code> ）乘以块大小的结果。块大小取决于 IQ 页大小。
Pages	在内存中表示表中存储的所有数据以及这些对象的元数据所需的总 IQ 页数。此值始终大于或等于 <code>Compressed Pages</code> （下一 <code>sp_iqdbsize</code> 列）的值。
Compressed Pages	将表中的数据以及这些对象的元数据存储存储在磁盘上所需的总 IQ 页数。此值始终小于或等于 <code>Pages</code> （前一 <code>sp_iqdbsize</code> 列）的值，因为将 IQ 页从内存写入磁盘时 SAP Sybase IQ 会压缩页。 <code>sp_iqdbsize Compressed Pages</code> 列表示压缩的页数。
NBlocks	用于存储表中数据的块总大小。此值始终小于或等于 <code>sp_iqdbsize Physical Blocks</code> 值。
Catalog Blocks	用于存储表中元数据的块总大小。
RLVLogBlocks	用于存储 RLV 存储库日志信息的块数。
RLVLogKBytes	RLV 日志的总大小 (KB)。

### 示例

显示数据库 `iqdemo` 的大小信息：

```
sp_iqdbsize
Database
PhysicalBlocks KBytes Pages CompressedPages NBlocks CatalogBlocks
RLVLogBlocks RLVLogKBytes
=====
=====
/system1/sybase/IQ-16_0/demo/iqdemo.db
          1280    522    688          257    1119          18
```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqdbspace 过程

显示每个 SAP Sybase IQ `dbspace` 的详细信息。

### 语法

```
sp_iqdbspace [ dbspace-name ]
```

### 适用于

Simplex 和 Multiplex。

**特权**

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY DBSPACE 系统特权。

**注释**

利用 **sp\_iqdbspace** 中的信息确定是否必须移动数据，以及对于已移动的数据是否已释放旧版本。

列名	描述
DBSpaceName	在 <b>CREATE DBSPACE</b> 语句中指定的 dbspace 的名称。无论是指定 <b>CREATE DATABASE...CASE IGNORE</b> 还是指定 <b>CASE RESPECT</b> ，Dbspace 名称始终不区分大小写。
DBSpaceType	dbspace 的类型 (MAIN、SHARED_TEMP、TEMPORARY、RLV 或 CACHE)。
Writable	T (可写) 或 F (不可写)。
Online	T (联机) 或 F (脱机)。
Usage	所有文件当前使用的 dbspace 占整个 dbspace 的百分比。
TotalSize	dbspace 中所有文件的总大小，以 B (字节)、K (千字节)、M (兆字节)、G (千兆字节)、T (千吉字节) 或 P (千万亿字节) 为单位。
Reserve	dbspace 中可以添加到所有文件的保留空间总大小。
NumFiles	dbspace 中的文件数。
NumRWFiles	dbspace 中的读/写文件数。
Stripingon	F (关闭)。
StripeSize	如果磁盘条带化已开启，则始终为 1。
BlkTypes	用户数据和内部系统结构占用的空间。
OkToDrop	"Y" 表示可删除 dbspace；否则为 "N"。

BlkTypes 块类型标识符的值：

标识符	块类型
A	活动版本
B	备份结构
C	检查点日志
D	数据库标识



标识符	块类型
F	空闲列表
G	全局空闲列表管理器
H	空闲列表的头块
I	索引建议存储
M	Multiplex CM*
O	旧版本
R	RLV 空闲列表管理器
T	表使用
U	索引使用
N	列使用
X	在检查点处删除

\*Multiplex 提交标识块（实际 128 块）存在于所有 IQ 数据库中，即使 Simplex 数据库不使用它也不例外。

#### 示例

显示有关 dbspace 的信息：

```
sp_iqdbspace;
```

**注意：**以下示例显示 iqdemo 数据库中的对象，以便更好地阐释输出。iqdemo 包括一个名为 iq\_main 的示例用户 dbspace，您自己的数据库中可能不包括此 dbspace。

DBSpaceName	DBSpaceType	Writable
IQ_MAIN	MAIN	T
IQ__SYSTEM_MAIN	MAIN	T
IQ_SYSTEM_TEMP	TEMPORARY	T
myDas	CACHE	T

(继续)	Usage	DBSpaceName
Online		
T	55	IQ_MAIN

(继续) Online	Usage	DBSpaceName
T	21	IQ__SYSTEM_MAIN
T	1	IQ_SYSTEM_TEMP
T	1	myDas

(继续) Reserve	NumFiles	NumRWFiles
200M	1	1
50M	1	1
50M	1	1
0B	5	5

(继续) DBSpaceName	Stripingon	Stripe Size
IQ_MAIN	T	1K
IQ__SYSTEM_MAIN	F	8K
IQ_SYSTEM_TEMP	F	8K
myDas	T	1K

(继续) Blk Types	OkTo Drop
1H, 5169A, 190	N
1H, 7648F, 32D, 128M	N
1H, 64F, 32A	N
5, 192FH	Y

另请参见

- sp\_iqindexinfo 过程 (第 453 页)
- sp\_iqdbspaceinfo 过程 (第 423 页)
- sp\_iqspaceinfo 过程 (第 509 页)

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqdbspaceinfo 过程

显示在指定表中使用的每个对象和子对象的大小。这不受 RLV dbspace 支持。

### 语法

```
sp_iqdbspaceinfo [ dbspace-name ] [ , owner_name ] [ ,
object_name ] [ , object-type ]
```

### 参数

所有参数均为可选参数，并且任何参数的提供均不受其它参数值的影响。

- **dbspace\_name** - 如果已指定，则 **sp\_iqdbspaceinfo** 会为指定 **dbspace** 中具备组件的每个表显示一行。否则，该过程显示数据库中所有 **dbspace** 的信息。
- **owner\_name** - 对象的所有者。如果指定，**sp\_iqdbspaceinfo** 将仅显示包含指定所有者的那些表的输出。如果未指定，**sp\_iqdbspaceinfo** 显示数据库中所有用户的表的相关信息。
- **object\_name** - 表的名称。如果未指定，**sp\_iqdbspaceinfo** 显示数据库中所有表的相关信息。
- **object\_type** - 有效的 **table** 对象。

**sp\_iqdbspaceinfo** 存储过程支持用于解释 *dbspace\_name*、*object\_name* 和 *owner\_name* 的通配符。它以 **LIKE** 子句匹配查询内部模式的方式显示匹配指定模式的所有 **dbspace** 的信息。

### 适用于

Simplex 和 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- BACKUP DATABASE
- SERVER OPERATOR
- MANAGE ANY DBSPACE

### 注释

如果指定 RLV **dbspace**，该过程将不返回任何结果。

**sp\_iqdbspaceinfo** 向 **DBA** 显示各个 **dbspace** 中驻留的对象所占用的空间大小。**DBA** 可利用这些信息确定必须先重新定位哪些对象，然后才能删除 **dbspace**。子对象列以整数后跟后缀 **B**、**K**、**M**、**G**、**T** 或 **P**（分别表示字节、千字节、兆字节、千兆字节、千吉字节和千万亿字节）的形式显示报告的大小。

对于表，**sp\_iqdbspaceinfo** 显示所有子对象的大小信息（以整数加上后缀 **B**、**K**、**M**、**G**、**T** 或 **P** 的形式表示），按 *dbspace\_name*、*object\_name* 和 *owner\_name* 排序。

表 74. sp\_iqdbspaceinfo 列

列名	描述
dbspace_name	dbspace 的名称。
object_type	对象的类型（仅限于 <b>table</b> 或 <b>joinindex</b> ）
owner	对象所有者的名称。
object_name	dbspace 中对象的名称。
object_id	对象的全局对象 ID。
id	对象的表 ID。
columns	给定 dbspace 上的列存储空间大小。
indexes	给定 dbspace 上的索引存储空间大小。不要使用系统生成的索引（例如，唯一约束中的 <b>HG</b> 索引或 <b>FP</b> 索引）。
metadata	给定 dbspace 上元数据对象的存储空间大小。
primary_key	给定 dbspace 上主键相关对象的存储空间大小。
unique_constraint	给定 dbspace 上唯一约束相关对象的存储空间大小。
foreign_key	给定 dbspace 上外键相关对象的存储空间大小。
dbspace_online	表示 dbspace 是处于联机状态 ( <b>Y</b> ) 还是脱机状态 ( <b>N</b> )。

如果对使用 **-r** 开关（只读）启动的服务器运行 **sp\_iqdbspaceinfo**，将显示错误 Msg 13768, Level 14, State 0: SQL Anywhere Error -757:

Modifications not permitted for read-only database. 此行为是预期行为。其它存储过程（如 **sp\_iqdbspace**、**sp\_iqfile**、**sp\_iqdbspaceobjectinfo** 或 **sp\_iqobjectinfo**）则不会发生此错误。

#### 示例

**注意：** 以下示例显示 **iqdemo** 数据库中的对象，以便更好地阐释输出。**iqdemo** 包括一个名为 **iq\_main** 的示例用户 **dbspace**，您自己的数据库中可能不包括此 **dbspace**。

显示数据库所有 **dbspace** 内的所有表中的所有对象和子对象的大小：

```
sp_iqdbspaceinfo
```

dbspace_name	object_type	owner	object_name	object_id	id
columns					
iq_main	table	DBA	empl	3689	741 96K
iq_main	table	DBA	iq_dummy	3686	740 24K
iq_main	table	DBA	sale	3698	742 96K
iq_main	table	GROUPO	Contacts	3538	732

iq_main	288K		table	GROUPO	Customers	3515	731	
iq_main	240K		table	GROUPO	Departments	3632	738	72K
iq_main	408K		table	GROUPO	Employees	3641		739
iq_main	72K		table	GROUPO	FinancialCodes	3612		736
iq_main	72K		table	GROUPO	FinancialData	3621	737	96K
iq_main	3593	735	table	GROUPO	Products			
iq_main	120K		table	GROUPO	SalesOrderItems	3580		734
iq_main	144K		table	GROUPO	SalesOrders	3565		733
indexes	metadata	primary_key	unique_constraint	foreign_key	dbspace			
ace_online								
0B	1.37M	0B	0B	0B	Y			
0B	464K	0B	0B	0B	Y			
0B	1.22M	0B	0B	0B	Y			
0B	5.45M	24K	0B	48K	Y			
48K	4.63M	24K	0B	0B	Y			
0B	1.78M	24K	0B	48K	Y			
0B	8.03M	24K	0B	48K	Y			
0B	1.53M	24K	0B	0B	Y			
0B	2.19M	24K	0B	48K	Y			
192K	4.67M	24K	0B	0B	Y			
0B	2.7M	24K	0B	104K	Y			
0B	3.35M	24K	0B	144K	Y			

显示数据库指定 **dbspace** 内由指定用户拥有的所有对象和子对象的大小:

```
sp_iqdbspaceinfo iq_main, GROUPO
```

dbspace_name	object_type	owner	object_name	object_id	id
columns					
iq_main	table	GROUPO	Contacts	3538	732
iq_main	table	GROUPO	Customers	3515	731
iq_main	table	GROUPO	Departments	3632	738
iq_main	table	GROUPO	Employees	3641	739
iq_main	table	GROUPO	FinancialCodes	3612	736
iq_main	table	GROUPO	FinancialData	3621	737
iq_main	table	GROUPO	Products	3593	735
iq_main	table	GROUPO	SalesOrderItems	3580	734
iq_main	table	GROUPO	SalesOrders	3565	733
indexes	metadata	primary_key	unique_constraint	foreign_key	dbspace
ace_online					

0B	5.45M	24K	0B	48K	Y
48K	4.63M	24K	0B	0B	Y
0B	1.78M	24K	0B	48K	Y
0B	8.03M	24K	0B	48K	Y
0B	1.53M	24K	0B	0B	Y
0B	2.19M	24K	0B	48K	Y
192K	4.67M	24K	0B	0B	Y
0B	2.7M	24K	0B	104K	Y
0B	3.35M	24K	0B	144K	Y

显示数据库中指定的 **dbspace** 内由指定用户拥有的指定对象及其子对象的大小:

```
sp_iqdbspaceinfo iq_main,GROUPO,Departments
```

dbspace_name	object_type	owner	object_name	object_id	id
columns					
iq_main	table	GROUPO	Departments	3632	738 72K
indexes	metadata	primary_key	unique_constraint	foreign_key	dbspace_online
0B	1.78M	24K	0B	48K	Y

### 另请参见

- `sp_iqindexinfo` 过程 (第 453 页)
- `sp_iqdbspace` 过程 (第 419 页)
- `sp_iqspaceinfo` 过程 (第 509 页)
- 确定数据库所使用的安全模型 (第 381 页)

## `sp_iqdbspaceobjectinfo` 过程

列出给定 **dbspace** 中表类型的对象和子对象 (包括列、索引、元数据、主键、唯一约束、外键和分区)。这不受 **RLV dbspace** 支持。

### 语法

```
sp_iqdbspaceobjectinfo [ dbspace-name ] [ , owner_name ] [ , object_name ] [ , object-type ]
```

### 参数

所有参数都是可选的, 并且任何参数的提供均不受其它参数值的影响。

- **dbspace-name** - 如果已指定, `sp_iqdbspaceobjectinfo` 将仅显示指定 **dbspace** 的输出。否则, 将显示数据库中所有 **dbspace** 的信息。
- **owner-name** - 对象的所有者。如果指定, `sp_iqdbspaceobjectinfo` 将仅显示包含指定所有者的那些表的输出。如果未指定, `sp_iqdbspaceobjectinfo` 显示数据库中所有用户的表的相关信息。
- **object-name** - 表的名称。如果未指定, `sp_iqdbspaceobjectinfo` 显示数据库中所有表的相关信息。
- **object-type** - **table** 对象的有效对象类型。

**sp\_iqdbspaceobjectinfo** 存储过程支持用于解释 *dbspace\_name*、*object\_name* 和 *owner\_name* 的通配符。它以 **LIKE** 子句匹配查询内部的模式的方式显示匹配指定模式的所有 dbspace 的信息。

#### 特权

您必须具有系统过程的 EXECUTE 特权。

#### 注释

如果指定 RLV dbspace，该过程将不返回任何结果。

对于表，**sp\_iqdbspaceobjectinfo** 将为所有关联子对象显示汇总信息，按 *dbspace\_name*、*owner* 和 *object\_name* 排序。

**sp\_iqdbspaceobjectinfo** 根据输入参数值显示以下信息：

**表 75. sp\_iqdbspaceobjectinfo 列**

列名	描述
dbspace_name	dbspace 的名称。
dbspace_id	dbspace 的标识符。
object_type	表。
owner	对象所有者的名称。
object_name	dbspace 中表对象的名称。
object_id	对象的全局对象 ID。
id	对象的表 ID。
列	位于给定 dbspace 中的表列数。如果某一列或其中一个列分区位于某个 dbspace 上，则认为该列或该列分区位于该 dbspace 上。结果以 n/N 形式显示（总计 N 列的表中有 n 列位于给定 dbspace 上）。
indexes	位于给定 dbspace 上的用户定义表索引数。以 n/N 形式显示（总计 N 个表索引中有 n 个位于给定 dbspace 上）。如果是唯一约束，则不包含系统生成的索引，如 FP 索引和 HG 索引。
metadata	布尔字段 (Y/N)，表示子对象的元数据信息是否也位于此 dbspace 上。
primary_key	布尔字段 (1/0)，表示表的主键（如果有）是否也位于此 dbspace 上。
unique_constraint	位于给定 dbspace 上的唯一表约束数。以 n/N 形式显示（总计 N 个唯一表约束中有 n 个位于给定 dbspace 上）。
foreign_key	位于给定 dbspace 上的表外键数。以 n/N 形式显示（总计 N 个表外键中有 n 个位于给定 dbspace 上）。

列名	描述
partitions	位于给定 <b>dbspace</b> 上的表分区数。以 n/N 形式显示（总计 N 个表分区中有 n 个位于给定 <b>dbspace</b> 上）。

*示例*

以下示例显示 **iqdemo** 数据库中的对象，以便更好地阐释输出。**iqdemo** 包括一个名为 **iq\_main** 的示例用户 **dbspace**，您自己的数据库中可能不包括此 **dbspace**。

显示数据库中某特定 **dbspace** 的相关信息：

```
sp_iqdbspaceobjectinfo iq_main

dbspace_name  dbspace_id  object_type  owner  object_name  object_i
d_id  columns
iq_main      16387      table       DBA    emp1         3689
 741  4/4
iq_main      16387      table       DBA    iq_dummy     3686
 740  1/1
iq_main      16387      table       DBA    sale         3698
 742  4/4
iq_main      16387      table       GROUPO Contacts     3538
 732  12/12
iq_main      16387      table       GROUPO Customers    3515
 731  10/10
iq_main      16387      table       GROUPO Departments 3632
 738  3/3
iq_main      16387      table       GROUPO Employees   3641
 739  21/21
iq_main      16387      table       GROUPO FinancialCodes 3612
 736  3/3
iq_main      16387      table       GROUPO FinancialData 3621
 737  4/4
iq_main      16387      table       GROUPO Products    3593
 735  8/8
iq_main      16387      table       GROUPO SalesOrderItems3580
 734  5/5
iq_main      16387      table       GROUPO SalesOrders   3565
 733  6/6

indexes  metadata  primary_key  unique_constraint  foreign_key  part
itions
0/0      Y          0  0/0          0/0          0/0
0/0      Y          0  0/0          0/0          0/0
0/0      Y          0  0/0          0/0          0/0
0/0      Y          1  0/0          1/1          0/0
1/1      Y          1  0/0          0/0          0/0
0/0      Y          1  0/0          1/1          0/0
0/0      Y          1  0/0          1/1          0/0
0/0      Y          1  0/0          0/0          0/0
0/0      Y          1  0/0          1/1          0/0
4/4      Y          1  0/0          0/0          0/0
0/0      Y          1  0/0          2/2          0/0
0/0      Y          1  0/0          3/3          0/0
```



显示数据库中某特定 `dbspace` 中指定用户拥有的对象的相关信息：

```
sp_iqdbspaceobjectinfo iq_main,GROUPO
```

dbspace_name	dbspace_id	object_type	owner	object_name	object_id
iq_main	16387	table	GROUPO	Contacts	3538
732	2/12				
iq_main	16387	table	GROUPO	Customers	3515
731	10/10				
iq_main	16387	table	GROUPO	Departments	3632
738	3/3				
iq_main	16387	table	GROUPO	Employees	3641
739	21/21				
iq_main	16387	table	GROUPO	FinancialCodes	3612
736	3/3				
iq_main	16387	table	GROUPO	FinancialData	3621
737	4/4				
iq_main	16387	table	GROUPO	Products	3593
735	8/8				
iq_main	16387	table	GROUPO	SalesOrderItems	3580
734	5/5				
iq_main	16387	table	GROUPO	SalesOrders	3565
733	6/6				

  

indexes	metadata	primary_key	unique_constraint	foreign_key	partitions
0/0	Y	1	0/0	1/1	0/0
1/1	Y	1	0/0	0/0	0/0
0/0	Y	1	0/0	1/1	0/0
0/0	Y	1	0/0	1/1	0/0
0/0	Y	1	0/0	0/0	0/0
0/0	Y	1	0/0	1/1	0/0
4/4	Y	1	0/0	0/0	0/0
0/0	Y	1	0/0	2/2	0/0
0/0	Y	1	0/0	3/3	0/0

在本示例中，命令将 `dbspace_x` 上的所有表移动到 `dbspace_y`。

```
SELECT 'ALTER TABLE ' || owner || '.' ||
object_name || ' MOVE TO dbspace_y;'
FROM sp_iqdbspaceobjectinfo()
WHERE object_type = 'table' AND
dbspace_name = 'dbspace_x';
```

结果为以下 **ALTER TABLE** 命令：

```
ALTER TABLE DBA.dt1 MOVE TO dbspace_y;
ALTER TABLE DBA.dt2 MOVE TO dbspace_y;
ALTER TABLE DBA.dt3 MOVE TO dbspace_y;
```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqdbstatistics 过程

报告最近一次执行 `sp_iqcheckdb` 的结果。

### 语法

```
sp_iqdbstatistics
```

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 ALTER DATABASE 系统特权。

### 注释

显示在最近一次执行 `sp_iqcheckdb` 时收集的数据库统计信息。

### 示例

下面的示例显示 `sp_iqdbstatistics` 产生的输出。对于此示例，最近一次执行 `sp_iqcheckdb` 的形式是 `sp_iqcheckdb 'allocation database'` 命令。

DB Statistics	Value	Flags
=====		
DBCC Allocation Mode Report		
=====		
** DBCC Status		
DBCC Work units Dispatched	163	
DBCC Work units Completed	163	
=====		
Allocation Summary		
=====		
Blocks Total	8192	
Blocks in Current Version	4954	
Blocks in All Versions	4954	
Blocks in Use	4986	
% Blocks in Use	60	
** Blocks Leaked	32	*****
=====		
Allocation Statistics		
=====		
Blocks Created in Current TXN	382	
Blocks To Drop in Current TXN	382	
Marked Logical Blocks	8064	
Marked Physical Blocks	4954	
Marked Pages	504	
Blocks in Freelist	126553	
Imaginary Blocks	121567	
Highest PBN in Use	5432	

```

** 1st Unowned PBN |452 |*****
Total Free Blocks |3206 |
Usable Free Blocks |3125 |
% Free Space Fragmented |2 |
Max Blocks Per Page |16 |
1 Block Page Count |97 |
3 Block Page Count |153 |
4 Block Page Count |14 |
...
9 Block Hole Count |2 |
16 Block Hole Count |194 |
Database Objects Checked |1 |
B-Array Count |1 |
Blockmap Identity Count |1 |
=====|=====|
=====  

Connection Statistics | |
=====|=====|
=====

```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqdroplogin 过程

删除 SAP Sybase IQ 用户帐户。

### 语法 1

```
call sp_iqdroplogin ( 'userid' )
```

### 语法 2

```
sp_iqdroplogin 'userid'
```

### 语法 3

```
sp_iqdroplogin userid
```

### 语法 4

```
sp_iqdroplogin ( 'userid' )
```

### 参数

- **userid** - 要删除用户的 ID。

### 特权

您必须具有系统过程的 EXECUTE 特权。

### 注释

**sp\_iqdroplogin** 删除指定的用户。

*示例*

以下命令全部用于移除用户 rose:

```
sp_iqdroplogin 'rose'  
sp_iqdroplogin rose  
call sp_iqdroplogin ('rose')
```

**另请参见**

- sp\_iqaddlogin 过程 (第 386 页)
- 确定数据库所使用的安全模型 (第 381 页)

## **sp\_iqemptyfile 过程**

清空某个 dbfile 并将该 dbfile 中的对象移动到同一个 dbspace 中的另外一个可用读写 dbfile。这不适用于 RLV dbspace 中的文件。

*语法*

```
sp_iqemptyfile ( logical-file--name )
```

*特权*

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权:

- BACKUP DATABASE
- SERVER OPERATOR
- ALTER DATABASE

此外, 您必须还要至少拥有以下系统特权之一:

- INSERT ANY TABLE
- UPDATE ANY TABLE
- DELETE ANY TABLE
- ALTER ANY TABLE
- LOAD ANY TABLE
- TRUNCATE ANY TABLE
- ALTER ANY OBJECT

*注释*

**sp\_iqemptyfile** 清空某个 dbfile。dbspace 必须是只读 dbspace 才能执行 **sp\_iqemptyfile** 过程。此过程将该文件中的对象移动到同一个 dbspace 中的另外一个可用读写 dbfile。如果没有其它读写 dbfile 可用, SAP Sybase IQ 会显示一条错误消息。

---

**注意:** 在 Multiplex 环境中, 只能在协调器上运行 **sp\_iqemptyfile**。必须有一个读写 dbspace 可用才能成功执行该过程。

---

如果 dbfile 位于 RLV dbspace 中, 此错误消息将显示:

```
Cannot empty files in an rlv store dbspace.
```

示例

清空 dbfile dbfile1:

```
sp_iqemptyfile 'dbfile1'
```

另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iquestdbspaces 过程

估计给定索引总大小所需的 **dbspace** 的数量和大小。

语法

```
sp_iquestdbspaces ( db_size_in_bytes, iq_page_size,
min_#_of_bytes, max_#_of_bytes )
```

特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权:

- MANAGE ANY DBSPACE
- ALTER DATABASE

注释

**sp\_iquestdbspaces** 根据数据的唯一程度提出多项建议:

建议	描述
min	如果数据几乎没有变化，可以选择只创建建议大小为 <b>min</b> 的 <b>dbspace</b> 段。这些建议反映在最少改变数据的情况下可能的最佳数据压缩。
avg	如果数据的变化量为平均水平，则可以创建建议大小为 <b>min</b> 的 <b>dbspace</b> 段，以及建议大小为 <b>avg</b> 的其它段。
max	如果数据变化度较高（有许多唯一值），则可以创建建议大小为 <b>min</b> 、 <b>avg</b> 和 <b>max</b> 的 <b>dbspace</b> 段。
spare	如果不确定数据中唯一值的数量，可以创建建议大小为 <b>min</b> 、 <b>avg</b> 、 <b>max</b> 和 <b>spare</b> 的 <b>dbspace</b> 段。装载数据后，可以随时删除不使用的段，但是创建的段过少可能会花费一些时间。

根据数据库的大小、IQ 页大小和每个 **dbspace** 段的字节数范围，显示有关 **dbspace** 段数量和大小信息。此过程假定数据库是使用指定 IQ 页大小的缺省块大小创建的（否则，返回的估计值将不正确）。

表 76. sp\_iquestdbspaces 参数

Name	数据类型	描述
db_size_in_bytes	decimal(16)	数据库的大小（以字节为单位）。
iq_page_size	smallint	为数据库的 IQ 段定义的页大小（必须是 2 的幂且介于 65536 和 524288 之间；缺省值为 131072）。
min_#_of_bytes	int	每个 dbspace 段的最小字节数。缺省值为 20,000,000 (20MB)。
max_#_of_bytes	int	每个 dbspace 段的最大字节数。缺省值为 2,146,304,000 (2.146GB)。

## 另请参见

- 确定数据库所使用的安全模型（第 381 页）

**sp\_iquestdbspaces 过程示例**

使用示例作为 **sp\_iquestdbspaces** 用法的参考。

```
sp_iquestdbspaces 12000000000, 65536, 500000000, 2146304000
```

dbspace files	类型	Size	Msg
1	min	2146304000	
2	min	2146304000	
3	min	507392000	
4	avg	2146304000	
5	max	2053697536	
6	spare	1200001024	

此示例估计一个 12GB 数据库所需的 dbspace 段的大小和数量。如果不需要数据唯一性，SAP 建议您最少创建 3 个段（列为 **min**）以获得最佳压缩。如果数据的变化量为平均水平，则应多创建 1 个段（列为 **avg**）。如果数据变化量很大（有很多唯一值，需要大范围索引），则可能需要再多创建 1 个段（列为 **max**）。通过创建 1200001024 字节的 spare 段，可以确保初始装载成功完成。装载数据库后，可以删除任何不使用的 dbspace 段。

**将 sp\_iquestdbspaces 与其它系统存储过程一起使用**

您需要运行两个存储过程来提供 **sp\_iquestdbspaces** 所需的 *db\_size\_in\_bytes* 参数。

**sp\_iquestdbspaces** 的结果只是基于索引平均大小的估计值。实际大小取决于表中存储的数据，尤其是数据的变化量。

SAP 强烈建议您创建 spare dbspace 段，因为之后如果不使用也可以删除它们。

1. 针对您预计经常进行连接的所有表对运行 **sp\_iqestjoin**。
2. 为每对表选择其中一个建议的索引大小。
3. 对您为所有表选择的索引大小进行总计。
4. 针对所有表运行 **sp\_iqestspace**。
5. 对 **sp\_iqestspace** 返回的所有 RAW DATA 索引大小进行总计。
6. 将步骤 3 得到的总计与步骤 5 得到的总计相加，以确定索引总大小。
7. 使用步骤 6 中计算的索引总大小作为 **sp\_iqestdbspaces** 中的 *db\_size\_in\_bytes* 参数。

## sp\_iqestspace 过程

根据表中的行数估计创建索引所需的空间大小。

### 语法

```
sp_iqestspace ( table_name, #_of_rows, iq_page_size )
```

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种 系统特权：

- CREATE ANY INDEX
- ALTER ANY INDEX
- CREATE ANY OBJECT
- ALTER ANY OBJECT
- 您拥有该表

### 注释

根据基础数据库表中的行数和数据库 IQ 页大小显示数据库所需的空间大小。此过程假定数据库是使用指定 IQ 页大小的缺省块大小创建的（否则估计值将不正确）。下表列出了 **sp\_iqestspace** 参数。

表 77. **sp\_iqestspace** 的参数

名称	数据类型	描述
<i>table_name</i>	char(256)	表名
<i>#_of_rows</i>	int	表中的行数
<i>iq_page_size</i>	smallint	为数据库的 IQ 段定义的页大小（必须是 2 的乘方且介于 65536 和 524288 之间；缺省值为 131072）

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqevent 过程

显示有关系统事件和用户定义的事件的信息。

### 语法

```
sp_iqevent [ event-name ], [ event-owner ], [ event-type ]
```

### 参数

- **event-name** - 事件的名称。
- **event-owner** - 事件的所有者。
- **event-type** - 事件类型。允许的值如下：
  - **SYSTEM** - 只显示有关系统事件（由用户 `sys` 或 `dbo` 拥有的事件）的信息：
  - **ALL** - 显示有关用户事件和系统事件的信息
  - **其它任何值** - 显示有关用户事件的信息

可以在不指定任何参数的情况下调用 **sp\_iqevent** 过程。如果不指定任何参数，则缺省情况下仅显示用户事件（不归 `dbo` 或 `sys` 所有的事件）的相关信息。

### 特权

您必须具有系统过程的 **EXECUTE** 特权。

### 注释

如果不指定前两个参数中的任何一个参数，但指定序列中的下一个参数，则必须用 **NULL** 替换省略的参数。例如，`sp_iqevent NULL, NULL, SYSTEM` 和 `sp_iqevent NULL, user1`。

**表 78. sp\_iqevent 用法示例**

语法	输出
<code>sp_iqevent</code>	显示数据库中所有用户事件的相关信息
<code>sp_iqevent e1</code>	显示事件 <code>e1</code> 的相关信息
<code>sp_iqevent non_existing_event</code>	不返回任何行，因为事件 <code>non_existing_event</code> 不存在
<code>sp_iqevent NULL, DBA</code>	显示 <code>DBA</code> 拥有的所有事件的相关信息
<code>sp_iqevent e1, DBA</code>	显示 <code>DBA</code> 拥有的事件 <code>e1</code> 的相关信息
<code>sp_iqevent ev_iqbegintxn</code>	<code>ev_iqbegintxn</code> 是系统定义的事件。如果没有同样名为 <code>ev_iqbegintxn</code> 的用户定义的事件，则不返回任何行。（缺省情况下，仅返回用户定义的事件。）



语法	输出
<code>sp_iqevent ev_iqbegintxn, dbo</code>	不返回任何行，因为事件 <code>ev_iqbegintxn</code> 不是用户事件（缺省情况下，仅返回用户事件）
<code>sp_iqevent NULL, NULL, SYSTEM</code>	显示所有系统事件（由 <code>dbo</code> 或 <code>SYS</code> 拥有）的相关信息
<code>sp_iqevent ev_iqbegintxn, NULL, SYSTEM</code>	显示系统事件 <code>ev_iqbegintxn</code> 的相关信息
<code>sp_iqevent ev_iqbegintxn, dbo, ALL</code>	显示 <code>dbo</code> 拥有的系统事件 <code>ev_iqbegintxn</code> 的相关信息

**sp\_iqevent** 存储事件可显示有关数据库中事件的信息。如果指定了一个或多个参数，将根据指定参数对结果进行过滤。例如，如果指定 *event-name*，则仅显示有关指定事件的信息。如果指定 *event-owner*，则 **sp\_iqevent** 仅返回指定所有者拥有的事件的相关信息。如果未指定任何参数，**sp\_iqevent** 将显示有关数据库中所有用户事件的信息。

**sp\_iqevent** 过程返回以下列中的信息：

表 79. sp\_iqevent 列

列名	描述
<code>event_name</code>	事件的名称
<code>event_owner</code>	事件的所有者
<code>event_type</code>	对于系统事件，是 <code>SYSEVENTTYPE</code> 系统表中列出的事件类型
<code>enabled</code>	指明是否允许触发事件 (Y/N)
<code>action</code>	事件处理程序定义
<code>condition</code>	用于控制事件处理程序触发的 <b>WHERE</b> 条件
位置	允许触发事件的位置： <ul style="list-style-type: none"> <li>• C = 统一</li> <li>• R = 远程</li> <li>• A = 所有</li> </ul>
<code>remarks</code>	注释字符串

#### 示例

显示用户定义事件 `e1` 的相关信息：

```
sp_iqevent e1
```

event_name	event_owner	event_type	enabled	action
el	DBA	(NULL)	Y	(NULL)
condition	location	remarks		
(NULL)	A	(NULL)		

显示有关所有系统事件的信息:

sp_iqevent	NULL, NULL, SYSTEM			
event_name	event_owner	event_type	enabled	action
ev_iqbegintxn	dbo	IQTLVAvailable	Y	begin call
ev_iqmpxcompact	dbo	(NULL)	N	dbo.sp_iqlog... begin Declare
condition	location	remarks (NULL)	A	_Catalog... (NULL)
(NULL)	A	(NULL)		

### 另请参见

- sp\_iqcolumn 过程 (第 401 页)
- sp\_iqconstraint 过程 (第 409 页)
- sp\_iqdatatype 过程 (第 415 页)
- sp\_iqhelp 过程 (第 441 页)
- sp\_iqindex 和 sp\_iqindex\_alt 过程 (第 447 页)
- sp\_iqkeys 过程 (第 485 页)
- sp\_iqprocparm 过程 (第 489 页)
- sp\_iq\_reset\_identity 过程 (第 499 页)
- sp\_iqtable 过程 (第 539 页)
- sp\_iqview 过程 (第 554 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqfile 过程

显示有关 dbspace 中每个 dbfile 的详细信息。

### 语法

```
sp_iqfile [ dbspace-name ]
```

### 适用于

Simplex 和 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权, 以及 MANAGE ANY DBSPACE 系统特权。

### 注释

**sp\_iqfile** 显示 dbspace 中每个 dbfile 中的数据的用法、属性和类型。可以使用这些信息确定是否必须移动数据, 以及对于已移动的数据是否已释放旧版本。

列名	描述
DBSpaceName	在 <b>CREATE DBSPACE</b> 语句中指定的 <b>dbspace</b> 的名称。无论是指定 <b>CREATE DATABASE...CASE IGNORE</b> 还是指定 <b>CASE RESPECT</b> ， <b>DbSPACE</b> 名称始终不区分大小写。
DBFileName	逻辑文件名。
Path	物理文件或原始分区的位置。
SegmentType	<b>dbspace</b> 的类型 ( <b>MAIN</b> 、 <b>TEMPORARY</b> 、 <b>RLV</b> 或 <b>CACHE</b> )。
RWMode	<b>dbspace</b> 的模式：始终为读写 ( <b>RW</b> )。
Online	<b>T</b> (联机) 或 <b>F</b> (脱机)。
Usage	此文件当前使用的 <b>dbspace</b> 占整个 <b>dbspace</b> 的百分比。在 <b>Multiplex</b> 配置中针对辅助节点运行时，此列将显示 <b>NA</b> 。
DBFileSize	文件或原始分区的当前大小。对于原始分区来说，此大小值可以小于实际大小。
Reserve	<b>dbspace</b> 中可添加到此文件的保留空间。
StripeSize	如果磁盘条带化已开启，则始终为 1。
BlkTypes	用户数据和内部系统结构占用的空间。
FirstBlk	分配给文件的第一个 <b>IQ</b> 块号。
LastBlk	分配给文件的最后一个 <b>IQ</b> 块号。
OkToDrop	" <b>Y</b> " 表示可删除文件；否则为 " <b>N</b> "。

标识符	块类型
A	活动版本
B	备份结构
C	检查点日志
D	数据库标识
F	空闲列表
G	全局空闲列表管理器
H	空闲列表的标头块
I	索引建议存储

标识符	块类型
M	Multiplex CM*
O	旧版本
R	RLV 空闲列表管理器
T	表使用
U	索引使用
N	列使用
X	在检查点处删除

\*Multiplex 提交标识块（实际 128 块）存在于所有 IQ 数据库中，即使 Simplex 数据库不使用也不例外。

示例

显示有关 dbspace 中文件的信息：

```
sp_iqfile;
```

```
sp_iqfile;
DBSpaceName, DBFileName, Path, SegmentType, RWMode, Online,
Usage, DBFileSize, Reserve, StripeSize, BlkTypes, FirstBlk,
LastBlk, OkToDrop

'IQ_SYSTEM_MAIN', 'IQ_SYSTEM_MAIN', '/sun1-c1/users/smith/mpx/m/
mpx_db.iq', 'MAIN', 'RW', 'T', '21', '
2.92G', '0B', '1K', '1H', 76768F, 32D, 19A, 1850, 128M, 34B, 32C'
, 1, 384000, 'N'

'mpx_main1', 'mpx_main1', '/sun1-c1/users/smith/mpx/m/
mpx_main1.iq', 'MAIN', 'RW', 'T', '1'
, '100M', '0B', '1K', '1H', 1045440, 1058239, 'N'

'IQ_SHARED_TEMP', 'sharedfile1_bcp', '/sun1-c1/users/smith/mpx/m/
f1', 'SHARED_TEMP', 'RO', 'T', '0',
'50M', '0B', '1K', '1H', 1, 6400, 'N'

'IQ_SHARED_TEMP', 'sharedfile2_bcp', '/sun1-c1/users/smith/mpx/m/
f2', 'SHARED_TEMP', 'RO', 'T', '0',
'50M', '0B', '1K', '1H', 1045440, 1051839, 'N'

'myDAS', 'ssd_dev_1', '/dev/raw/ssd_dev_1', 'CACHE', 'RW', 'T', '2',
'20M', '0B', '1K', '1H', '64F', '1', '5120', 'N'
'myDAS', 'ssd_dev_2', '/dev/raw/ssd_dev_2', 'CACHE', 'RW', 'T', '1',
'20M', '0B', '1K', '1H', '32F', '522208', '527327', 'N'
'myDAS', 'ssd_dev_3', '/dev/raw/ssd_dev_3', 'CACHE', 'RW', 'T', '1',
'20M', '0B', '1K', '1H', '32F', '1044416', '1049535', 'N'
'myDAS', 'ssd_dev_4', '/dev/raw/ssd_dev_4', 'CACHE', 'RW', 'T', '1',
'20M', '0B', '1K', '1H', '32F', '1566624', '1571743', 'N'
```

```
'myDAS', 'ssd_dev_5', '/dev/raw/ssd_dev_5', 'CACHE', 'RW', 'T', '1',
'20M', '0B', '1K', '1H', '32F', '2088832', '2093951', 'N'

'IQ_SYSTEM_TEMP', 'IQ_SYSTEM_TEMP', '/sun1-c1/users/smithmpx/m/
mpx_db.iqtmp', 'TEMPORARY', 'RW',
'T', '1', '2.92G', '0B', '1K', '1H,64F,33A', '1,384000', 'N'
```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqhelp 过程

显示有关系统对象及数据类型和用户定义的对象及数据类型的信息。

### 语法

```
sp_iqhelp [ obj-name ], [ obj-owner ], [ obj-category ], [ obj-type ]
```

#### **obj-type**:

```
table
| view
| procedure
| event
| datatype
```

### 参数

- **obj-name** - 对象的名称。
- **obj-owner** - 对象的所有者。
- **obj-category** - 用于指定对象类别的可选参数。

列、约束和索引与表关联，不能直接对其进行查询。在查询表时，将显示与该表关联的列、索引和约束的相关信息。

如果指定的对象类别不是允许值，则会返回 “Invalid object category” 错误。

允许值如下：

- **事件** - 对象是基表
- **视图** - 对象是视图
- **过程** - 对象是存储过程或函数
- **事件** - 对象是事件
- **数据类型** - 对象是系统数据类型或用户定义数据类型
- **obj-type** - 对象的类型。允许的值如下：
  - **SYSTEM** - 只显示有关系统对象（由用户 sys 或 dbo 拥有的对象）的信息：

- **ALL** - 显示所有对象的相关信息。缺省情况下，只显示非系统对象的相关信息。如果指定的对象类型不是 **SYSTEM** 或 **ALL**，则返回 “Invalid object type” 错误。

可以在不指定任何参数的情况下调用 **sp\_iqhelp** 过程。如果不指定任何参数，**sp\_iqhelp** 将显示数据库中所有独立对象（即基表、视图、存储过程、函数、事件和数据类型）的相关信息。

*特权*

您必须具有系统过程的 **EXECUTE** 特权。

*注释*

如果前三个参数均未指定，但指定序列中的下一个参数，则必须用 **NULL** 替换省略的参数。例如，**sp\_iqhelp NULL, NULL, NULL, SYSTEM** 和 **sp\_iqhelp NULL, user1, “table”**。

将 *obj-category* 参数用单引号或双引号引起来（值为 **NULL** 时例外）。

如果 **sp\_iqhelp** 在数据库中未找到符合指定说明的对象，则会返回 “No object found for the given description” 错误。

**表 80. sp\_iqhelp 用法示例**

语法	输出
<b>sp_iqhelp</b>	显示有关数据库中所有用户定义的表、视图、过程、事件和数据类型的摘要信息
<b>sp_iqhelp t1, u1, “table”</b>	显示用户 <b>u1</b> 拥有的表 <b>t1</b> 以及与 <b>t1</b> 关联的列、索引和约束的相关信息
<b>sp_iqhelp NULL, u1, “view”</b>	显示用户 <b>u1</b> 拥有的视图 <b>v1</b> 以及与 <b>v1</b> 关联的列的相关信息
<b>sp_iqhelp sp2</b>	显示过程 <b>sp2</b> 和 <b>sp2</b> 的参数的相关信息
<b>sp_iqhelp e1</b>	显示事件 <b>e1</b> 的相关信息
<b>sp_iqhelp dt1</b>	显示数据类型 <b>dt1</b> 的相关信息
<b>sp_iqhelp NULL, NULL, NULL, SYSTEM</b>	显示有关所有系统对象（由 <b>dbo</b> 或 <b>SYS</b> 拥有）的摘要信息
<b>sp_iqhelp non_existing_obj</b>	返回错误 “Object ‘non_existing_obj’ not found”，因为对象 <b>non_existing_obj</b> 不存在

语法	输出
sp_iqhelp NULL, non_existing_user	返回错误 “User ‘non_existing_user’ not found” , 因为用户 <b>non_existing_user</b> 不存在
sp_iqhelp t1, NULL, “apple”	返回错误 “Invalid object category ‘apple’ ” , 因为 <b>“apple”</b> 不是 <i>obj-category</i> 的允许值
sp_iqhelp t1, NULL, NULL, “USER”	返回错误 “Invalid object type ‘USER’ ” , 因为 <b>“USER”</b> 不是 <i>obj-type</i> 的允许值

**sp\_iqhelp** 存储过程可显示 IQ 数据库中系统对象和用户定义对象及数据类型的相关信息。**sp\_iqhelp** 支持的对象包括表、视图、列、索引、约束、存储过程、函数、事件和数据类型。

如果指定了一个或多个参数，将根据指定参数对结果进行过滤。例如，如果指定 *obj-name*，则仅显示有关指定对象的信息。如果指定 *obj-owner*，则 **sp\_iqhelp** 仅返回指定所有者拥有的对象的相关信息。如果未指定任何参数，**sp\_iqhelp** 将显示有关数据库中所有用户定义的表、视图、过程、事件和数据类型的摘要信息。

**sp\_iqhelp** 过程可返回摘要信息或详细信息，具体取决于指定参数是与多个对象匹配还是与一个对象匹配。**sp\_iqhelp** 的输出列与存储过程 **sp\_iqtable**、**sp\_iqindex**、**sp\_iqview** 和 **sp\_iqconstraint** 显示的列类似。

如果有多个对象与指定的 **sp\_iqhelp** 参数匹配，**sp\_iqhelp** 将显示有关这些对象的摘要信息。

表 81. sp\_iqhelp 摘要信息

对象类型	显示的列
基表	table_name、table_owner、server_type、location、table_constraints、remarks
视图	view_name、view_creator、view_def、server_type、location、remarks
存储过程	proc_name、proc_creator、proc_defn、replicate、srvid、remarks
函数	proc_name、proc_creator、proc_defn、replicate、remarks
事件	event_name、event_creator、enabled、location、event_type、action、external_action、condition、remarks
系统数据类型和用户定义数据类型	type_name、creator、nulls、width、scale、default、check

如果有一个对象与指定的 `sp_iqhelp` 参数匹配, `sp_iqhelp` 将显示有关该对象的详细信息。

**表 82. sp\_iqhelp 详细信息**

对象类型	描述	列
表	显示指定基表、列、索引和约束的相关信息。	<ul style="list-style-type: none"> <li>表列: <code>table_name</code>、<code>table_owner</code>、<code>server_type</code>、<code>location</code>、<code>table_constraints</code>、<code>remarks</code></li> <li>列列: <code>column_name</code>、<code>domain_name</code>、<code>width</code>、<code>scale</code>、<code>nulls</code>、<code>default</code>、<code>check</code>、<code>pkey</code>、<code>user_type</code>、<code>cardinality</code>、<code>est_cardinality</code>、<code>remarks</code></li> <li>索引列: <code>index_name</code>、<code>column_name</code>、<code>index_type</code>、<code>unique_index</code>、<code>location</code>、<code>remarks</code></li> <li>约束列: <code>constraint_name</code> (角色)、<code>column_name</code>、<code>index_name</code>、<code>constraint_type</code>、<code>foreigntable_name</code>、<code>foreigntable_owner</code>、<code>foreigncolumn_name</code>、<code>foreignindex_name</code>、<code>location</code></li> </ul>
视图	显示有关指定视图及其列的信息	<ul style="list-style-type: none"> <li>视图列: <code>view_name</code>、<code>view_creator</code>、<code>view_def</code>、<code>server_type</code>、<code>location</code>、<code>remarks</code></li> <li>列列: <code>column_name</code>、<code>domain_name</code>、<code>width</code>、<code>scale</code>、<code>nulls</code>、<code>default</code>、<code>check</code>、<code>pkey</code>、<code>user_type</code>、<code>cardinality</code>、<code>est_cardinality</code>、<code>remarks</code></li> </ul>
存储过程	显示有关指定过程及其参数的信息	<ul style="list-style-type: none"> <li>过程列: <code>proc_name</code>、<code>proc_creator</code>、<code>proc_defn</code>、<code>replicate</code>、<code>srvid</code>、<code>remarks</code></li> <li>参数列: <code>parameter_name</code>、<code>type</code>、<code>width</code>、<code>scale</code>、<code>default</code>、<code>mode</code></li> </ul>
函数	显示有关指定函数及其参数的信息	<ul style="list-style-type: none"> <li>函数列: <code>proc_name</code>、<code>proc_creator</code>、<code>proc_defn</code>、<code>replicate</code>、<code>srvid</code>、<code>remarks</code></li> <li>参数列: <code>parameter_name</code>、<code>type</code>、<code>width</code>、<code>scale</code>、<code>default</code>、<code>mode</code></li> </ul>
事件	显示有关指定事件的信息	<ul style="list-style-type: none"> <li>事件列: <code>event_name</code>、<code>event_creator</code>、<code>enabled</code>、<code>location</code>、<code>event_type</code>、<code>action</code>、<code>external_action</code>、<code>condition</code>、<code>remarks</code></li> </ul>
数据类型	显示有关指定数据类型的信息	<ul style="list-style-type: none"> <li>数据类型列: <code>type_name</code>、<code>creator</code>、<code>nulls</code>、<code>width</code>、<code>scale</code>、<code>default</code>、<code>check</code></li> </ul>

**注意:** 系统过程的过程定义 (`proc-defn`) 在视图中被加密和隐藏。



有关各个输出列的说明，请参见相关的存储过程。例如，有关表列的说明，请参见 **sp\_iqtable** 过程。

### 示例

显示有关表 `sale` 的详细信息：

```
sp_iqhelp sale
```

```
Table_name Table_owner Server_type Location dbspace_id isPartitioned
table_constraints
=====
sale      DBA          IQ           Main      16387      N

Remarks  table_constraints
=====
          (NULL)      (NULL)

column_name domain_name width scale nulls default cardinality
=====
prod_id     integer    4      0      Y      (NULL)  0
month_num   integer    4      0      Y      (NULL)  0
rep_id      integer    4      0      Y      (NULL)  0
sales       integer    4      0      Y      (NULL)  0

  est_cardinality  isPartitioned  remarks  check
  =====
  0                N              (NULL)  (NULL)
  0                N              (NULL)  (NULL)
  0                N              (NULL)  (NULL)
  0                N              (NULL)  (NULL)

index_name          column_name  index_type  unique_index  location
=====
ASIQ_IDX_T463_C2_FP month_num    FP          N             Main
ASIQ_IDX_T463_C1_FP prod_id      FP          N             Main
ASIQ_IDX_T463_C3_FP rep_id       FP          N             Main
ASIQ_IDX_T463_C4_FP sales        FP          N             Main

  remarks
  =====
  (NULL)
  (NULL)
  (NULL)
  (NULL)
```

显示有关过程 `sp_customer_list` 的详细信息：

```
sp_iqhelp sp_customer_list
proc_name      proc_owner  proc_defn
=====
sp_customer_list  DBA        create procedure DBA.sp_customer_list()
result(id integer company_name char(35))
begin
select id company_name from Customers
end
```

replicate	srvid	remarks			
=====	=====	=====			
N	(NULL)	(NULL)			
parm_name	parm_type	parm_mode	domain_name	width	scale
=====	=====	=====	=====	=====	=====
id	result	out	integer	4	0
company_name	result	out	char	35	0
default					
=====					
(NULL)					

### 另请参见

- sp\_iqcolumn 过程 (第 401 页)
- sp\_iqconstraint 过程 (第 409 页)
- sp\_iqdatatype 过程 (第 415 页)
- sp\_iqevent 过程 (第 436 页)
- sp\_iqindex 和 sp\_iqindex\_alt 过程 (第 447 页)
- sp\_iqpkeys 过程 (第 485 页)
- sp\_iqprocparm 过程 (第 489 页)
- sp\_iq\_reset\_identity 过程 (第 499 页)
- sp\_iqtable 过程 (第 539 页)
- sp\_iqview 过程 (第 554 页)
- 确定数据库所使用的安全模型 (第 381 页)

### **sp\_iqhelp 与 Adaptive Server 的兼容性**

SAP Sybase IQ **sp\_iqhelp** 存储过程与 Adaptive Server **sp\_help** 过程相似，后者显示有关 SYSOBJECTS 系统表中列出的所有数据库对象的信息，以及系统数据类型和用户定义数据类型的相关信息。

在支持的对象类型和对象命名空间方面，SAP Sybase IQ 与 Adaptive Server 存在一些体系结构差异。在 Adaptive Server 中，所有对象（表、视图、存储过程、日志、规则、缺省值、触发器、检查约束、参照约束和临时对象）都存储在 SYSOBJECTS 系统表中，并且位于同一命名空间。SAP Sybase IQ 支持的对象（表、视图、存储过程、事件、主键以及唯一约束、检查约束和参照约束）存储在不同系统表中，并且位于不同的命名空间。例如，在 SAP Sybase IQ 中，表名可以与事件名或存储过程名相同。

由于 SAP Sybase IQ 和 Adaptive Server 存在体系结构差异，SAP Sybase IQ **sp\_iqhelp** 支持的对象类型及其语法不同于 Adaptive Server **sp\_help**；但是，这两种存储过程显示的数据库的对象的信息类型相似。

## sp\_iqindex 和 sp\_iqindex\_alt 过程

列出有关索引的信息。

### 语法 1

```
sp_iqindex ( [ table_name ], [ column_name ], [ table_owner ] )
```

### 语法 2

```
sp_iqindex [ table_name='tablename' ],  
[ column_name='columnname' ], [ table_owner='tableowner' ]
```

### 语法 3

```
sp_iqindex_alt ( [ table_name ], [ column_name ], [ table_owner ] )
```

### 语法 4

```
sp_iqindex_alt [ table_name='tablename' ],  
[ column_name='columnname' ], [ table_owner='tableowner' ]
```

### 参数

- **语法 1** - 如果不指定前两个参数中的任何一个参数，但指定序列中的下一个参数，则必须用 NULL 替换省略的参数。例如，sp\_iqindex NULL, NULL, DBA 和 sp\_iqindex Departments, NULL, DBA。
- **语法 2** - 可以按任意顺序指定参数。将其用单引号引起来。
- **语法 3 和 4** - 当存在多列索引时将生成稍有不同的输出。允许的选项与语法 1 和 2 相同。

### 特权

您必须具有系统过程的 EXECUTE 特权。

### 注释

显示有关数据库中索引的信息。指定其中一个参数将只返回该表、该列或指定用户拥有的表中的索引。如果指定多个参数，将根据指定的所有参数对结果进行过滤。如果未指定任何参数，则会返回数据库中所有表的所有索引。

**表 83. sp\_iqindex 和 sp\_iqindex\_alt 列**

列名	描述
table_name	表的名称
table_owner	表的所有者
column_name	列的名称；多列索引中可出现多个名称

列名	描述
index_type	索引类型的缩写形式 (例如 <b>HG</b> 、 <b>LF</b> )
index_name	索引的名称
unique_index	“U” 表示索引是唯一索引; 否则为 “N”
location	TEMP = IQ 临时存储, MAIN = IQ 存储, SYSTEM = Catalog 存储
remarks	使用 <b>COMMENT</b> 语句添加的用户注释

**sp\_iqindex** 始终针对每个索引生成一行。如果有多列索引, **sp\_iqindex\_alt** 将针对每个列中的每个索引生成一行。

### 另请参见

- sp\_iqcolumn 过程 (第 401 页)
- sp\_iqconstraint 过程 (第 409 页)
- sp\_iqdatatype 过程 (第 415 页)
- sp\_iqevent 过程 (第 436 页)
- sp\_iqhelp 过程 (第 441 页)
- sp\_iqpkeys 过程 (第 485 页)
- sp\_iqprocparm 过程 (第 489 页)
- sp\_iq\_reset\_identity 过程 (第 499 页)
- sp\_iqtable 过程 (第 539 页)
- sp\_iqview 过程 (第 554 页)
- 确定数据库所使用的安全模型 (第 381 页)

### **sp\_iqindex** 和 **sp\_iqindex\_alt** 过程示例

使用示例作为 **sp\_iqindex** 和 **sp\_iqindex\_alt** 用法的参考。

以下两种语法变体都会返回 DepartmentID 列中的所有索引:

```
call sp_iqindex (NULL, 'DepartmentID')
```

```
sp_iqindex column_name='DepartmentID'
```

table_name	table_owner	column_name	index_type
Departments	GROUPO	DepartmentID	FP
Departments	GROUPO	DepartmentID	HG
Employees	GROUPO	DepartmentID	FP

(继续) index_name	unique_index	位置	dbspace_id	remarks
ASIQ_IDX_T201_C1_FP	N	主	16387	(NULL)
ASIQ_IDX_T201_C1_HG	U	主	16387	(NULL)
ASIQ_IDX_T202_C5_FP	N	主	16387	(NULL)

以下两种语法变体都会返回表所有者 GROUPO 拥有的表 Departments 中的所有索引:

```
sp_iqindex Departments,NULL,GROUPO
```

```
sp_iqindex table_name='Departments',table_owner='DBA'
```

table_name	table_owner	column_name	index_type
Departments	GROUPO	DepartmentHeadID	FP
Departments	GROUPO	DepartmentID	FP
Departments	GROUPO	DepartmentID	HG
Departments	GROUPO	DepartmentName	FP

(继续) index_name	unique_index	位置	dbspace_id	remarks
ASIQ_IDX_T201_C3_FP	N	主	16387	(NULL)
ASIQ_IDX_T201_C1_FP	N	主	16387	(NULL)
ASIQ_IDX_T201_C1_HG	U	主	16387	(NULL)
ASIQ_IDX_T201_C2_FP	N	主	16387	(NULL)

**sp\_iqindex\_alt** 的以下两种语法变体都会返回包含列 City 的表 Employees 中的索引。索引 emp\_loc 是列 City 和 State 的多列索引。对于多列索引，**sp\_iqindex\_alt** 会针对每列显示一行。

```
sp_iqindex_alt Employees,City
```

```
sp_iqindex_alt table_name='Employees',  
column_name='City'
```

table_name	table_owner	column_name	index_type
Employees	GROUPO	City	FP

table_name	table_owner	column_name	index_type
Employees	GROUPO	City	HG
Employees	GROUPO	状态	HG

(继续)	unique_index	dbspace_id	remarks
index_name			
ASIQ_IDX_T452_C7_FP	N	16387	(NULL)
emp_loc	N	16387	(NULL)
emp_loc	N	16387	(NULL)

对于相同的表和列，**sp\_iqindex** 产生的输出稍有不同：

```
sp_iqindex Employees, City
```

```
sp_iqindex table_name='Employee', column_name='City'
```

table_name	table_owner	column_name	index_type
Employees	GROUPO	City	FP
Employees	GROUPO	City,State	HG

(继续)	unique_index	dbspace_id	location	remarks
index_name				
ASIQ_IDX_T452_C7_FP	N	16387	Main	(NULL)
emp_loc	N	16387	Main	(NULL)

## sp\_iqindexadvice 过程

显示存储的索引建议消息。可以选择清除建议存储。

语法

```
sp_iqindexadvice ( [ resetflag ] )
```

参数

- **resetflag** - 允许调用者清除索引建议存储。如果 *resetflag* 不为零，则在检索完最后一行后将删除所有建议。

特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种 系统特权：

- ALTER ANY INDEX
- ALTER ANY OBJECT

### 注释

允许用户使用 SQL 来查询聚合索引顾问消息。该信息有助于确定哪些索引或模式的更改将会影响大多数查询。

**INDEX\_ADVISOR** 列:

**表 84. sp\_iqindexadvice 列**

列名	描述
Advice	唯一的建议消息
NInst	消息实例数
LastDT	上次生成建议的日期/时间

### 示例

**sp\_iqindexadvice** 过程的输出样本:

**表 85. sp\_iqindexadvice 输出**

Advice	NInst	LastDT
Add a CMP index on DBA.tb (c2, c3) Predicate:(tb.c2 = tb.c3)	2073	2009-04-07 16:37:31.000
Convert HG index on DBA.tb.c4 to a unique HG	812	2009-04-06 10:01:15.000
Join Key Columns DBA.ta.c1 and DBA.tb.c1 have mismatched data types	911	2009-02-25 20:59:01.000

### 另请参见

- sp\_iqcolumnuse 过程 (第 404 页)
- sp\_iqindexuse 过程 (第 461 页)
- sp\_iqtableuse 过程 (第 544 页)
- sp\_iqunusedcolumn 过程 (第 549 页)
- sp\_iqunusedindex 过程 (第 550 页)
- sp\_iqunusedtable 过程 (第 551 页)
- sp\_iqworkmon 过程 (第 559 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqindexfragmentation 过程

报告 SAP Sybase IQ 索引中的 B 树、GArray 和位图结构占用的页空间百分比的相关信息。

对于 GArray，填充百分比计算不会将 GArray 组内的保留空间考虑在内，后者由 **GARRAY\_FILL\_FACTOR\_PERCENT** 选项控制。

### 语法

```
dbo.sp_iqindexfragmentation ( 'target ' )
```

```
target: table table-name | index index-name [...]
```

### 参数

- **table-name** - 目标 **table** *table-name* 报告指定表中的所有非缺省索引。
- **index-name** - 目标 **index** *index-name* 报告指定索引。每个 *index-name* 都是限定索引名。可以在表中指定多个索引，但必须对每个指定的索引重复 **index** 关键字。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY DBSPACE 系统特权。

### 示例

报告表 DBA.prop\_nu.prop\_nu\_a 中的唯一 **HG** 索引的内部索引碎片：

索引	IndexType	Btree_Node_pages	GARRAY_FILL_FACTOR_PERCENT
DBA.prop_nu.prop_nu_a	HG	8	25
SQLCODE:	0		
Fill Percent	B 树页数	GArray 页数	位图页数
0-10%	13	2	8
11-20%	1	8	0
21-30%	0	4	0
31-40%	3	20	0
41-50%	4	116	0
51-60%	6	4	0
61-70%	3	3	0
71-80%	4	1	0



索引	IndexType	Btree_Node_pages	GARRAY_FILL_FACTOR_PERCENT
81-90%	1	1	0
91-100%	192	276	0

**注意：** 所有百分比均截断至最接近的百分点。**HG** 索引还会显示选项 `GARRAY_FILL_FACTOR_PERCENT` 的值。使用 **B** 树的索引类型还将显示节点（非叶）页数。这些类型包括 **HG**、**LF**、**WD**、**DATE** 和 **DTTM**。

如果在执行此存储过程期间出现错误，则 `SQLCODE` 为非零值。

### 另请参见

- `sp_iqindexmetadata` 过程（第 455 页）
- `sp_iqindexinfo` 过程（第 453 页）
- `sp_iqindexsize` 过程（第 459 页）
- `sp_iqrebuildindex` 过程（第 492 页）
- `sp_iqrowdensity` 过程（第 503 页）
- 确定数据库所使用的安全模型（第 381 页）

## `sp_iqindexinfo` 过程

显示某个给定对象在每个主 `dbspace` 的每个索引中所用的块数。如果对象驻留在多个 `dbspace` 中，则 `sp_iqindexinfo` 返回在所有 `dbspace` 中使用的空间，如下例所示。

### 语法

```
sp_iqindexinfo ' { database
  | [ table table-name | index index-name ] [...] }
  [ resources resource-percent ]'
```

### 适用于

Simplex 和 Multiplex。

### 特权

您必须具有系统过程的 `EXECUTE` 特权，以及 `MANAGE ANY DBSPACE` 系统特权。

### 注释

您可以请求整个数据库的索引信息，或者指定任意数目的表参数或索引参数。如果指定了表名，`sp_iqindexinfo` 会返回表中所有索引的信息。如果指定了索引名，则仅返回关于该索引的信息。

如果指定的 `table-name` 或 `index-name` 不明确或无法找到该对象，则返回错误。

缺省情况下，在 Multiplex 数据库中，`sp_iqindexinfo` 显示辅助节点上共享 IQ 存储库的相关信息。如果指定了各个表或索引，则会自动选中要显示的存储库。

*resource-percent* 必须是大于 0 的整数。借助资源百分比，您可以通过指定要使用的总 CPU 百分比来限制 **sp\_iqindexinfo** 过程的 CPU 利用率。

**sp\_iqindexinfo** 会向 DBA 显示给定对象驻留在哪些 **dbspace** 上。DBA 可用该信息来确定哪些 **dbspace** 必须具有 **relocate** 模式才能重新分配对象。

**sp\_iqindexinfo** 的结果将依据运行该命令的事务所见到的版本来显示。不显示由其它版本使用的块。

表 86. **sp\_iqindexinfo** 列

列名	描述
Object	表或索引名
DbSPACE_name	dbSPACE 的名称
ObjSize	该 dbSPACE 中此对象的数据大小
DBSpPct	该对象所使用的 dbSPACE 百分比
MinBlk	该 dbSPACE 中此对象使用的第一个块
MaxBlk	此 dbSPACE 中该对象使用的最后一个块；用于确定必须重新分配哪些对象然后才能将 dbSPACE 调整为较小的大小

示例

显示 Departments 表中索引的相关信息：

```
sp_iqindexinfo 'table GROUPO.Departments';
```

Object	DbSPACEName	ObjSize	DBSpPct	MinBlk	MaxBlk
GROUPO.Departments	iq_main	288K	1	1,045,496.00	1,048,891.00
GROUPO.Departments.ASIQ_IDX_T779_C1_FP	iq_main	176K	1	1,047,197.00	1,047,328.00
GROUPO.Departments.ASIQ_IDX_T779_C2_FP	iq_main	160K	1	1,047,213.00	1,047,324.00
GROUPO.Departments.ASIQ_IDX_T779_C3_FP	iq_main	184K	1	1,047,229.00	1,047,317.00

Object	DbSpaceName	ObjSize	DBSpPct	MinBlk	MaxBlk
GROUPO.Departments.ASIQ_IDX_T779_C3_HG	iq_main	440K	1	1,048,421.00	1,048,796.00
GROUPO.Departments.ASIQ_IDX_T779_I4_HG	iq_main	288K	1	1,047,261.00	1,047,306.00

### 另请参见

- `sp_iqdbspace` 过程 (第 419 页)
- `sp_iqdbspaceinfo` 过程 (第 423 页)
- `sp_iqspaceinfo` 过程 (第 509 页)
- `sp_iqindexmetadata` 过程 (第 455 页)
- `sp_iqindexfragmentation` 过程 (第 452 页)
- `sp_iqindexsize` 过程 (第 459 页)
- 确定数据库所使用的安全模型 (第 381 页)

## `sp_iqindexmetadata` 过程

显示给定索引的索引元数据。

您可以选择将输出限制为仅输出某个指定表的索引，或仅输出属于某个指定所有者的索引。

### 语法

```
dbo.sp_iqindexmetadata (index-name
[ , table-name [ , owner-name ] ] )
```

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种:

- ALTER ANY INDEX 系统特权
- ALTER ANY OBJECT 系统特权
- 对表的 REFERENCES 特权
- 您拥有该表

### 注释

指定一个表名可将输出限制为仅输出属于该表的索引。指定一个所有者名可将输入限制为仅输出属于该所有者的索引。省略的参数缺省为 NULL。每个过程只能指定一个索引。

可以通过 `sp_iqindexmetadata` 获得列的用户供应商 IQ UNIQUE 值。如果存在唯一的 HG 或 LF 索引，它会报告精确的基数。如果（只）存在不唯一的 HG 索引，则报告基数为 0。

输出的第一行为索引的所有者名、表名和索引名。

输出的后续行具体取决于所指定的索引类型。

**表 87. sp\_iqindexmetadata 输出行**

索引类型	返回的元数据
<b>CMP, DATE, DTTM, TIME</b>	Type, Version
<b>FP</b>	Type、 Style、 Version、 DBType、 Maximum Width、 EstUnique、 TokenCount、 NBit、 CountSize、 DictSize、 CountLen、 MaxKeyToken、 MinKey Token、 MinCount、 MaxCount、 DistinctKey、 IQ Unique
<b>HG</b>	Type、 Version、 Maintains Exact Distinct、 Level 0 Threshold、 Force Physical Delete、 Maximum Level Count、 Tier ratio、 Auto sizing、 Average Load Size (records)、 Active Subindex count、 Cardinality Range Min - Max、 Estimated Cardinality、 Accuracy of Cardinality
<b>HNG</b>	Type, Version, BitsPerBlockmap, NumberOfBits
<b>LF</b>	Type, Version, IndexStatus, NumberOfBlockmaps, BitsPerBlockmap, Distinct Keys
<b>WD</b>	Type, Version, KeySize, Delimiters, DelimiterCount, MaxKeyWordLength, PermitEmptyWord

示例

值 1	值 2	值 3
DBA	test2	test2_c1_hg
类型	HG	
Version	3	
Maintains Exact Distinct	否	
Level 0 Threshold	3000000	
Force Physical Delete	是	
Maximum Level Count	10	
Tier ratio	30	

值 1	值 2	值 3
Auto sizing	On	
Avarage Load Size (records)	58622	
Active Subindex count	3	
Cardinality Range Min - Max	5-5	
Estimated Cardinality	5	
Accuracy of Cardinality	100	
Level:0 Main Index Total Row Count	1	
Level:0 Main Index Deleted Row Count	0	
Level:0 Main Index # of Btree Pages in Main btree	1	
Level:0 Main Index # of GArray Pages	1	
Level:0 Main Index # of Keys in Main Btree	1	
Level:0 Main Index # of Keys Probed in Btree	0	
Level:0 Main Index # of Keys Found Duplicate in Btree	0	
Level:0 Main Index # of Keys Possible Distinct in Btree	0	
Level:1 Main Index Total Row Count	3145747	
Level:1 Main Index Deleted Row Count	0	
Level:1 Main Index # of Btree Pages in Main btree	1	
Level:1 Main Index # of Btree Pages in Conjugate btree	1	
Level:1 Main Index # of GArray Pages	2	
Level:1 Main Index # of Keys in Main Btree	8	
Level:1 Main Index # of Keys in Conjugate Btree	3	
Level:1 Main Index # rows in Conjugate Btree	2949127	
Level:1 Main Index # of Keys Probed in Btree	0	
Level:1 Main Index # of Keys Found Duplicate in Btree	0	
Level:1 Main Index # of Keys Possible Distinct in Btree	0	
Level:1 Incremental Index Total Row Count	1	
Level:1 Incremental Index Deleted Row Count	0	

值 1	值 2	值 3
Level:1 Incremental Index # of Btree Pages	1	
Level:1 Incremental Index # of GArray Pages	1	
Level:1 Incremental Index #of Keys in Btree	1	
Level:1 Incremental Index # of Keys Probed in Btree	0	
Level:1 Incremental Index # of Keys Found Duplicate in Btree	0	
Level:1 Incremental Index # of Keys Possible Distinct in Btree	0	

### 另请参见

- `sp_iqindexfragmentation` 过程 (第 452 页)
- `sp_iqindexinfo` 过程 (第 453 页)
- `sp_iqindexsize` 过程 (第 459 页)
- 确定数据库所使用的安全模型 (第 381 页)

## `sp_iqindexfragmentation` 过程

标识迁移数据库中必须重建的宽列，然后才能针对这些宽列执行读写活动。

### 语法

**`sp_iqindexrebuildwidedata`** [*table.name*]

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种:

- INSERT ANY TABLE 系统特权
- 用于重建索引的针对表的 INSERT 特权
- 您拥有该表

### 注释

必须重建列宽超过 255 个字符的 CHAR、VARCHAR、BINARY 和 VARBINARY 列以及迁移到 SAP Sybase IQ 16.0 数据库中的所有 Long Varchar 列和 Long Binary 列，然后数据库引擎才能针对这些列执行读/写活动。`sp_iqindexrebuildwidedata` 标识这些列，并生成可用于 `sp_iqrebuildindex` 过程以重建列的语句的列表。

包括可选的 [*table.name*] 参数，以生成该表的宽列列表。忽略 [*table.name*] 参数，以生成用于数据库中所有表的宽列列表。

### 示例

为表 T2 生成宽列重建语句:

```
sp_iqindexrebuildwidedata T2
```

输出:

```
Owner      Table      Column      Domain      Width      IndexType      sp_iqrebuild
DBA        T2         C1          char         1020         Long varchar FP  sp_iqrebuildindex 'DBA.T2' 'column
"C1" 0';
```

另请参见

- `sp_iqrebuildindex` 过程 (第 492 页)

## `sp_iqindexsize` 过程

给出指定索引的大小。

语法

```
sp_iqindexsize [ [ owner. ] table. ] index_name
```

特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权:

- MANAGE ANY DBSPACE
- ALTER ANY INDEX
- 您拥有该表

注释

**表 88. `sp_iqindexsize` 列**

列名	描述
Username	索引所有者。
Indexname	返回其结果的索引, 包括表名。
Type	索引类型。
Info	报告其 KBytes、Pages 和 Compressed Pages 的 IQ 索引的组件。组件因素索引类型而异。例如, 缺省的 (FP) 索引包括 BARRAY (barray) 和 Bitmap (bm) 组件。Low_Fast (LF) 索引包括 B-tree (bt) 和 Bitmap (bm) 组件。
KBytes	以 KB 表示的物理对象大小。
Pages	将对象保存在内存中所需的 IQ 页数。
Compressed Pages	对象在磁盘上压缩后的 IQ 页数。

以字节或千字节为单位返回索引的总大小, Info 列则描述会报告其 KBytes、Pages 和 Compressed Pages 的 IQ 索引的组件。描述的组件因素索引类型而异。例如, 缺省的 (FP) 索引包括 BARRAY (barray) 和 Bitmap (bm) 组件。Low\_Fast (LF) 索引包括 B-tree (bt) 和 Bitmap (bm) 组件。

此外，还返回将对象保存在内存中所需的页数，以及将索引在磁盘上压缩后的 IQ 页数。

对于此过程，必须指定 *index\_name* 参数。若要将结果限制为单个表中的该索引名称，请在指定索引时将 *owner.table* 包括在内。

示例

```
sp_iqindexsize ASIQ_IDX_T780_I4_HG
```

Username	Indexname	Type	Info	Kbytes	Pages	Compressed Pages
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	Total	288	4	2
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	vdo	0	0	0
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	bt	152	2	1
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	garray	0	0	0
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	bm	136	2	1
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	barray	0	0	0
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	dpstore	0	0	0
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	largelob	0	0	0
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	txtPst	0	0	0



```
CREATE TEXT INDEX ti ON Employees( Street ) IMMEDIATE
REFRESH;sp_iqindexsize 'ti';
```

Username	Indexname	类型	信息	KBytes	Pages	Compressed Pages
GRUPO	GRUPO.Employees.ti	TEXT	总	896	12	6
GRUPO	GRUPO.Employees.ti	TEXT	vdo	0	0	0
GRUPO	GRUPO.Employees.ti	TEXT	bt	304	4	2
GRUPO	GRUPO.Employees.ti	TEXT	garray	152	2	1
GRUPO	GRUPO.Employees.ti	TEXT	bm	136	2	1
GRUPO	GRUPO.Employees.ti	TEXT	barray	152	2	1
GRUPO	GRUPO.Employees.ti	TEXT	dpstore	0	0	0
GRUPO	GRUPO.Employees.ti	TEXT	largelob	0	0	0
GRUPO	GRUPO.Employees.ti	TEXT	txtPst	304	4	2

另请参见

- sp\_iqindexmetadata 过程 (第 455 页)
- sp\_iqindexfragmentation 过程 (第 452 页)
- sp\_iqindexinfo 过程 (第 453 页)
- 确定数据库所使用的安全模型 (第 381 页)

## **sp\_iqindexuse** 过程

报告负载访问的辅助 (非 FP) 索引的详细使用信息。

语法

```
sp_iqindexuse
```

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

*注释*

负载访问的每个辅助索引都显示一行。不会显示尚未访问的索引。索引使用情况根据优化程序、约束和查询使用情况进行细分。

系统不会报告 SYSTEM 中创建的表的索引。

**表 89. sp\_iqindexuse 列**

列名	描述
IndexName	索引名
TableName	表名称
Owner	索引所有者的用户名
UID**	索引唯一标识符
Type	索引类型
LastDT	上次访问的日期/时间
NOpt	元数据/唯一性访问的次数
NQry	查询访问的次数
NConstraint	访问唯一或参照完整性检查的次数

\*\*UID 是系统分配的用于唯一标识索引实例的编号（实例是在创建对象时定义的）。

*示例*

**sp\_iqindexuse** 过程的输出样本。

IndexName	TableName	Owner	UID	Type	LastDT	NQry	NConstraint	NOpt
n_nationkey_hg	nation	DBA	29	HG	20070917			
22:08:06~	12 0 12							
n_regionkey_hg	nation	DBA	31	HG	20070917			
22:08:06~	12 0 0							
r_regionkey_hg	region	DBA	47	HG	20070917			
22:08:06~	12 0 12							
s_suppkey_hg	supplier	DBA	64	HG	20070917			
22:08:06~	12 0 12							
p_partkey_hg	part	DBA	87	HG	20070917			
22:08:06~	6 0 6							
s_suppkey_hg	supplier	DBA	64	HG	20070917			
22:08:06~	12 0 12							
...								

**另请参见**

- sp\_iqcolumnuse 过程 (第 404 页)
- sp\_iqindexadvice 过程 (第 450 页)
- sp\_iqtableuse 过程 (第 544 页)
- sp\_iqunusedcolumn 过程 (第 549 页)
- sp\_iqunusedindex 过程 (第 550 页)
- sp\_iqunusedtable 过程 (第 551 页)
- sp\_iqworkmon 过程 (第 559 页)
- 确定数据库所使用的安全模型 (第 381 页)

**sp\_iqlmconfig 过程**

控制许可管理配置，显示和设置许可证类型和状态。

*语法 1*

```
sp_iqlmconfig 'edition', { 'SE' | 'SA' | 'EE' }
```

**表 90. “edition” 参数的摘要信息**

主题	值
默认值	'EE' (企业版)
值的范围	'SE' (Small Business) 'SA' (Single Application) 'EE' (Enterprise Edition)
状态	静态

*语法 2*

```
sp_iqlmconfig 'license type', { 'CP' | 'DT' | 'SF' | 'AC' | 'BC' | 'CH' | 'DH' | 'SH' | 'AH' | 'BH' }
```

**表 91. “license type” 参数的摘要信息**

主题	值
默认值	'DT' (开发和测试)
值的范围	'AC' (OEM CPU 许可证) 'AH' (OEM CPU 许可证芯片) 'BC' (OEM 备用许可证) 'BH' (OEM 备用许可证芯片) 'CP' (CPU 许可证) 'CH' (CPU 许可证芯片) 'DH' (开发和测试许可证芯片) 'DT' (开发和测试) 'EV' (评测) 'SF' (备用 CPU 许可证) 'SH' (备用 CPU 许可证芯片)
状态	静态

*语法 3*

```
sp_iqlmconfig 'email severity', { 'ERROR' | 'WARNING' | 'INFORMATIONAL' | 'NONE' }
```

未指定电子邮件通知处于禁用状态。

**语法 4**

```
sp_iqlmconfig 'smtp host' , ' <host name>' | '
```

**表 92. 参数**

参数	描述
主机名	指定用于电子邮件通知的 SMTP 主机。

**语法 5**

```
sp_iqlmconfig 'email sender' , ' <email address>' | '
```

**表 93. 参数**

参数	描述
<email address>	指定用于电子邮件通知的发件人的电子邮件地址。

**语法 6**

```
sp_iqlmconfig 'email recipients' , ' <email recipients>' | '
```

**表 94. 参数**

参数	描述
<email recipients>	指定将向其发送电子邮件通知的用逗号分隔的电子邮件地址列表。

**语法 7**

```
sp_iqlmconfig |
```

**特权**

您必须具有系统过程的 EXECUTE 特权，以及 SERVER OPERATOR 系统特权。

**注释**

启动时，**sp\_iqlmconfig** 检查指定许可证的版本类型和许可证类型。

- 如果未找到指定许可证，则服务器处于宽限模式。
- 只有指定了非空的版本值时，指定许可证类型才有效。
- 如果不带任何参数调用 **sp\_iqlmconfig**（语法 3），它将显示上述所有信息，以及其它信息，例如：
  - 产品版本和许可证类型
  - 正在使用哪些可选许可证
  - 许可证计数

- 电子邮件通知
- 有关许可证的常规信息

### 另请参见

- 可用于服务器的属性（第 92 页）
- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqlocks 过程

针对 IQ 主存储库和 IQ 目录存储库显示数据库中锁的相关信息。

### 语法

```
sp_iqlocks ([connection,] [[owner.]table_name,] max_locks,]
[sort_order])
```

### 参数

所有参数均可用于限制结果。

参数	数据类型	描述
connection	integer	连接 ID。使用该选项时，该过程仅返回指定连接的锁的相关信息。缺省值为零，将返回所有连接的相关信息。
owner.table_name	char(128)	表名。使用该选项时，该过程仅返回指定表的锁的相关信息。缺省值为 NULL，将返回数据库中所有表的相关信息。如果未指定所有者，系统将假定过程的调用者拥有该表。
max_locks	integer	可返回其信息的锁的最大数目。缺省值为 0，将返回所有锁的信息。
sort_order	char(1)	信息返回的顺序： <ul style="list-style-type: none"> <li>• C 表示按 connection 排序（缺省值）</li> <li>• T 表示按 table_name 排序</li> </ul>

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

### 注释

显示数据库中当前锁的相关信息。根据您所指定的选项，可以将结果限制为显示单个连接或单个表的锁，或显示指定数目的锁。

sp\_iqlocks 显示以下信息，并按 sort\_order 参数中指定的顺序进行排序：

列	数据类型	描述
conn_name	VARCHAR(128)	当前连接的名称。

列	数据类型	描述
conn_id	INTEGER	具有锁的连接 ID。
user_id	CHAR(128)	与该连接 ID 相关联的用户。
table_type	CHAR(6)	表的类型。该类型为 <b>BASE</b> （对于表）、 <b>GLBTMP</b> （对于全局临时表）或 <b>MVIEW</b> （对于实例化视图）。 仅 IQ 目录存储库中的 SQL Anywhere 表支持实例化视图。
creator	VARCHAR(128)	表的所有者。
table_name	VARCHAR(128)	控制锁的表。
index_id	INTEGER	索引 ID 或 NULL。
lock_class	CHAR(8)	锁类。Schema、Row、Table 或 Position 之一。
lock_duration	CHAR(11)	锁的持续时间。Transaction、Position 或 Connection 之一。
lock_type	CHAR(9)	锁类型（这取决于锁类）。
row_identifier	UNSIGNED BIGINT	锁的启动行标识符或 NULL。
row_range	BIGINT	锁定的连续行行数。RLV 存储库中的行锁可以是单行也可以是多行。

如果 **sp\_iqlocks** 无法找到拥有表锁的用户的连接 ID 或用户名，将以 0（零）来表示连接 ID 并以 User unavailable 表示用户名。

lock\_type 列中的值取决于 lock\_class 列中的锁分类。可以返回以下值：

锁类	锁类型	注释
模式	<ul style="list-style-type: none"> <li>已共享 - 共享模式锁。</li> <li>独占 - （仅 IQ 目录存储表）独占模式锁。</li> </ul>	对于模式锁，row_identifier 和索引 ID 值均为 NULL。

锁类	锁类型	注释
行	<ul style="list-style-type: none"> <li>• <b>读取</b> - 读锁定。</li> <li>• <b>意图</b> - 意图锁。</li> <li>• <b>ReadPK</b> - 读锁定。</li> <li>• <b>写入</b> - 写锁定。</li> <li>• <b>WriteNoPK</b> - 写锁定。</li> <li>• <b>代理</b> - 代理锁定。</li> </ul>	<p>行的读锁定可以是短期锁定（在隔离级别 1 的扫描），也可以是在更高隔离级别的长期锁定。<b>lock_duration</b> 列指明读锁定是由于游标稳定性而短期锁定 (<b>Position</b>)，还是一直保持到 <b>COMMIT/ROLLBACK</b> 的长期锁定 (<b>Transaction</b>)。行锁定始终保持在 8 字节行标识符作为 <b>row_identifier</b> 列中的 64 位整数值而被报告的特定行上。</p> <p>代理锁定是行锁定的一种特殊情况。代理锁定被保持在代理条目上，这些代理条目是在参照完整性检查延迟时创建的。对于表中所创建的每个代理条目，没有唯一的代理锁定。而是一个代理锁定对应于由给定连接为给定表所创建的一组代理条目。对于与代理锁定关联的表和连接，<b>row_identifier</b> 值是唯一的。</p> <p>如果需要，现在可单独锁定一行中的键值部分和非键值部分。连接可以获得共享（读）访问行的键值部分的读锁定，这样其它连接仍可获取该行的其它非键值列的写锁定。更新某行的非键值列时不会妨碍对引用该行的外行执行插入和删除操作。</p>
表	<ul style="list-style-type: none"> <li>• <b>已共享</b> - 共享表锁。</li> <li>• <b>意图</b> - 意图更新表锁。</li> <li>• <b>独占</b> - （仅 IQ 目录存储表）独占表锁。</li> </ul>	无
位置	<ul style="list-style-type: none"> <li>• <b>幻像</b> - （仅 IQ 目录存储表）幻像锁。</li> <li>• <b>插入</b> - 插入锁。</li> </ul>	位置锁通常也被保持在特定行上，且此行的 64 位行标识符出现在结果集的 <b>row_identifier</b> 列中。然而，位置锁还可以被保持在整个扫描上（索引扫描或顺序扫描），在此情况下， <b>row_identifier</b> 列为 NULL。

**注意：**独占、幻像或反幻像锁可以放置在 IQ 目录存储表中，但不能放在 IQ 主存储库的 SAP Sybase IQ 表中。除非已在目录存储库中显式撤消表锁，否则永远无法在 SAP Sybase IQ 数据库中看到这些类型的锁。

#### 示例

此示例显示 SAP Sybase IQ 数据库中的 **sp\_iqlocks** 过程调用及其输出。该过程进行调用时使用的均为缺省选项，因此其输出按连接排序显示所有锁。

```
call sp_iqlocks()
```

conn_name	conn_id	user_id	table_type	creator	table_name		
SQL_DBC_13cd6038	3			DBA	BASE	DBA	rv_locks2
SQL_DBC_13cd6038	3			DBA	BASE	DBA	rv_locks2
SQL_DBC_13cd6038	3			DBA	BASE	DBA	rv_locks2
RVL_CONN_T775	1000000407				BASE	DBA	rv_locks2

  

index_id	lock_class	lock_duration	lock_type	row_identifier
	Schema	Transaction	Shared	
	Row	Transaction	Row	1 4
	Row	Transaction	Row	281474976710656 1
	Table	Transaction	Intent	

**另请参见**

- 确定数据库所使用的安全模型 (第 381 页)

**sp\_iqmergerlvstore 过程**

触发行级版本 (RLV) 存储库和 IQ 主存储库的合并。

*语法*

```
sp_iqmergerlvstore 'merge_type', 'table_name', [ 'table_owner' ]
```

*特权*

您必须具有系统过程的 EXECUTE 特权。

*注释*

- 合并类型可以是 BLOCKING | NON-BLOCKING。
- 执行合并后，存储过程将自动提交合并事务。

**sp\_iqmodifyadmin 过程**

将指定登录策略中的某个选项设为一个特定值。如果未指定登录策略，则在根策略上设置该选项。在 Multiplex 中，sp\_iqmodifyadmin 将采用可选参数，即 Multiplex 服务器名称。

*语法 1*

```
call sp_iqmodifyadmin ( 'policy_option_name', 'value_in', [ 'login_policy_name' ] )
```



### 语法 2

```
sp_iqmodifyadmin 'policy_option_name' ,
'value_in' , 'login_policy_name'
```

### 语法 3

```
sp_iqmodifyadmin policy_option_name, value_in, ,login_policy_name
```

### 语法 4

```
sp_iqmodifyadmin 'policy_option_name' ,
'value_in' , 'login_policy_name' , 'server_name'
```

### 参数

- **policy\_option\_name** - 要更改的登录策略选项。
- **value\_in** - 登录策略选项的新值。
- **login\_policy\_name** - 要更改其登录策略选项的策略。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY LOGIN POLICY 系统特权。

### 示例

对于 *lockeduser* 策略，将登录选项 *locked* 设置为 ON:

```
call sp_iqmodifyadmin ('locked','on','lockeduser')
```

对于 *Writer1 Multiplex* 服务器上的 *lockeduser* 策略，将登录选项 *locked* 设置为 ON:

```
call sp_iqmodifyadmin ('locked','on','lockeduser','Writer1')
```

### 另请参见

- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqmodifylogin 过程

为用户分配登录策略。

### 语法 1

```
call sp_iqmodifylogin 'userid', ['login_policy_name']
```

### 语法 2

```
sp_iqmodifylogin 'userid', ['login_policy_name']
```

### 参数

- **userid** - 存放待修改的帐户名的变量。
- **login\_policy\_name** - (可选) 指定将分配给用户的登录策略的名称。如果未指定任何登录策略名, 则向用户分配根登录策略。

### 特权

您必须具有系统过程的 EXECUTE 特权, 以及 MANAGE ANY USER 系统特权。

### 示例

将用户 joe 分配给登录策略 expired\_password:

```
sp_iqmodifylogin 'joe', 'expired_password'
```

将用户 joe 分配给根登录策略:

```
call sp_iqmodifylogin ('joe')
```

### 另请参见

- sp\_expireallpasswords 系统过程 (第 628 页)
- sp\_iqaddlogin 过程 (第 386 页)
- sp\_iqcopyloginpolicy 过程 (第 413 页)
- sp\_iqpassword 过程 (第 484 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqmpxcheckdqpconfig 过程

**sp\_iqmpxcheckdqpconfig** 是一个诊断工具, 用于检查当前连接的 DQP 配置。如果 DQP 出现故障, 运行 **sp\_iqmpxcheckdqpconfig** 来确定是否是 DQP 配置问题导致了查询分布失败。

### 语法

```
sp_iqmpxcheckdqpconfig
```

### 适用于

仅限 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权。

注释

表 95. 列说明

列名	描述
DiagMsgID	唯一标识诊断消息。
描述	描述在 DQP 配置中发现的问题的诊断消息。

表 96. 诊断消息

DiagMsgID	描述
0	使用 DQP 配置未发现问题
1	数据库为 Simplex
2	Multiplex 以单节点配置模式运行
3	逻辑服务器策略选项 dqp_enabled 设置为 0
4	临时 dqp_enabled 连接选项设置为 OFF
5	逻辑服务器上下文只有一个成员节点
6	由于当前逻辑服务器中指定的成员资格无效，协调器将不参与 DQP
7	根逻辑服务器策略中的 ALLOW_COORDINATOR_AS_MEMBER 设置为 OFF，这使得协调器在逻辑服务器中的逻辑成员资格当前无效，因此协调器不参与 DQP
8	IQ_SHARED_TEMP dbspace 中没有 dbfile
9	IQ_SHARED_TEMP dbspace 中的所有 dbfile 都为 READ ONLY
10	IQ_SHARED_TEMP dbspace 为动态脱机

示例

sp\_iqmpxcheckdqpconfig 过程的输出样本：

```
diagmsgid      description
3              Logical server policy option dqp_enabled is set to 0
5              Logical server context has only one member node
6              Coordinator does not participate in DQP since its
               named membership in the logical server is
               currently ineffective
7              Coordinator does not participate in DQP since
               its logical membership in the logical server
               is currently ineffective because
               ALLOW_COORDINATOR_AS_MEMBER option in Root
```

```
8 Logical server policy set to OFF
   There is no dbfile in IQ_SHARED_TEMP dspace
```

**另请参见**

- 确定数据库所使用的安全模型 (第 381 页)

**sp\_iqmpxdumptlvlog 过程**

返回可读格式的表版本日志的内容。

*语法*

```
sp_iqmpxdumptlvlog  
[main], [asc | desc]
```

*适用于*

仅限 Multiplex。

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE MULTIPLEX 系统特权。

*注释*

**sp\_iqmpxdumptlvlog** 返回队列内容，协调器通过该队列将 DML 和 DDL 命令传播到辅助节点。

**asc** 或 **desc** 参数指定行顺序。这些参数需要 **main** 参数。缺省选项有：

```
'main', 'asc'.
```

*示例*

显示 **sp\_iqmpxdumptlvlog** 的输出：

RowID	Contents
1	Txn CatId:196 CmtId:196 TxnId:195 Last Rec:1 UpdateTime: 2011-08-08 15:41:43.621
2	Txn CatId:243 CmtId:243 TxnId:242 Last Rec:5 UpdateTime: 2011-08-08 15:42:25.070
3	DDL: Type=34, CatID=0, IdxID=0, Object=IQ_SYSTEM_TEMP, Owner=mpx4022_w1
4	CONN: CatID=0, ConnUser=
5	SQL: ALTER DBSPACE "IQ_SYSTEM_TEMP" ADD FILE "w1_tmpl" '/dev/raw/raw25' FILE ID 16391 PREFIX 65536 FINISH 0 FIRST BLOCK
1	BLOCK COUNT 3276792 RESERVE 0 MULTIPLEX SERVER "mpx4022_w1" COMMITID 242 CREATETIME '2011-08-08 15:42:24.860'
6	Txn CatId:283 CmtId:283 TxnId:282 Last Rec:7 UpdateTime: 2011-08-08 15:42:50.827
7	RFRB TxnID: 242 CmtID:243 ServerID 0 BlkmapID: 0d00000000000000d2000a0000000000200000000000000000 000000000000000000000008003501010000000c38000000000000

```
0100000000000000000000000000RFID:01000501000000001300000
0000000000100000000000100RBID:010005010000000013000
```

## sp\_iqmpxfilestatus 过程

在协调节点上执行时，显示每个包括的辅助节点上的每个共享 dbfile 的文件状态。如果在辅助节点上执行，仅显示当前节点的文件状态。

### 语法

```
sp_iqmpxfilestatus
```

### 适用于

仅限 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE MULTIPLEX 系统特权。

### 注释

**sp\_iqmpxfilestatus** 返回：

列名	数据类型	描述
ServerID	unsigned int	Multiplex 服务器标识符，来自 SYSIQMPXINFO
DBSpaceName	char(128)	从中保留空间的 Dbspace
FileName	char(128)	dbspace 文件的逻辑文件名
FileStatus	char(2)	Dbspace 文件状态： <ul style="list-style-type: none"> <li>VALID - 文件路径和特权正确</li> <li>INVALID_PATH - 路径名不可访问</li> <li>INVALID_PERM - 文件特权不正确</li> </ul>

### 示例

显示 **sp\_iqmpxfilestatus** 的示例输出：

```
server_id,server_name,DBSpace_name,FileName,FileStatus
1,'mpx2422_m','IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','VALID'
1,'mpx2422_m','mpx_main1','mpx_main1','VALID'
1,'mpx2422_m','IQ_SHARED_TEMP','sharedfile_dba','VALID'
1,'mpx2422_m','IQ_SHARED_TEMP','sharedfile_dba1','VALID'
2,'mpx2422_w1','IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','VALID'
2,'mpx2422_w1','mpx_main1','mpx_main1','VALID'
2,'mpx2422_w1','IQ_SHARED_TEMP','sharedfile_dba','VALID'
2,'mpx2422_w1','IQ_SHARED_TEMP','sharedfile_dba1','VALID'
3,'mpx2422_r1','IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','VALID'
3,'mpx2422_r1','mpx_main1','mpx_main1','VALID'
```

```
3, 'mpx2422_r1', 'IQ_SHARED_TEMP', 'sharedfile_dba', 'VALID'
3, 'mpx2422_r1', 'IQ_SHARED_TEMP', 'sharedfile_dba1', 'VALID'
```

## **sp\_iqmpxinconnpoolinfo** 过程

如果在协调器节点上运行，则显示每一节点的 INC 连接池状态。如果在辅助节点上执行，则仅显示当前节点的 INC 连接池状态。

### 语法

```
sp_iqmpxinconnpoolinfo
```

### 适用于

仅限 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE MULTIPLEX 系统特权。

### 注释

如果过程在协调器上运行，而且辅助节点未响应或已超时，那么结果集将忽略该节点的行，原因是只有该节点正在运行时才能访问此数据。

**sp\_iqmpxinconnpoolinfo** 过程返回：

列名	数据类型	描述
server_id	unsigned int	服务器的标识符
server_name	char(128)	服务器的名称
current_pool_size	unsigned int	连接池的当前大小
idle_connection_count	unsigned int	池中的空闲连接数
connections_in_use	unsigned int	正在使用的连接数

### 示例

显示 **sp\_iqmpxinconnpoolinfo** 的示例输出：

```
server_id,server_name,current_pool_size,
idle_connection_count,connections_in_use
2, 'r2_dbsrv90210', 0, 0, 0
3, 'w3_dbsrv90210', 0, 0, 0
```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## **sp\_iqmpxinheartbeatinfo** 过程

如果在协调节点上运行，显示每一节点的 INC 心跳状态。如果在辅助节点上执行，仅显示当前节点的 INC 心跳状态。

### 语法

**sp\_iqmpxinheartbeatinfo**

### 适用于

仅限 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE MULTIPLEX 系统特权。

### 注释

**sp\_iqmpxinheartbeatinfo** 过程返回：

列名	数据类型	描述	值
server_id	unsigned int	服务器的标识符	
server_name	char(128)	服务器的名称	
last_positive_hb	TIMESTAMP	活动上次成功强制回应的日期/时间	DD:MM:YYYY: HH:MM:SS
time_not_responding	TIME	自活动上次成功强制回应的时间	HH:MM:SS
time_until_timeout	TIME	在节点未响应的情况下，声明节点脱机之前所剩余的时间。	

### 示例

**sp\_iqmpxinheartbeatinfo** 的示例输出：

```
server_id,server_name,last_positive_hb,
time_not_responding,time_until_timeout
2,'r2_dbsrv90210',2012-11-17
15:48:42.0,00:00:00,00:00:00
3,'w3_dbsrv90210',2012-11-17
15:48:42.0,00:00:00,00:00:00
```

如果所用时间超过 24 个小时，则 SAP Sybase IQ 将返回 **sp\_iqmpxinheartbeatinfo** 输出，如下所示：

```
server_id,server_name,last_positive_hb,
time_not_responding,time_until_timeout
2,'r2_mpx_cr_srv',Jan 14 2013 11:57AM,11:59PM,11:59PM
3,'w4_mpx_cr_srv',Jan 14 2013
11:57AM,11:59PM,11:59PM
```

```
(2 rows affected)
(return status = 0)
```

如果 `time_not_responding` 和 `time_until_timeout` 列中的值为 11:59PM, 则意味着时间已经超过了 24 小时的限制。

**另请参见**

- 确定数据库所使用的安全模型 (第 381 页)

**sp\_iqmpxincstatistics 过程**

显示自服务器启动以来到执行该过程这一刻为止, 节点间通信 (INC) 状态汇总统计信息的快照。

*语法*

**sp\_iqmpxincstatistics**

*适用于*

仅限 Multiplex。

*特权*

您必须具有系统过程的 EXECUTE 特权, 以及 MANAGE ANY STATISTICS 系统特权。

*注释*

返回:

**表 97. sp\_iqmpxincstatistics 列**

列名	数据类型	描述
stat_name	char(128)	INC 统计信息名称
stat_value	unsigned integer	INC 统计信息值

**表 98. 有效的 stat\_name 值**

值	描述
NumSuspendedINC	自服务器启动以来的挂起 INC 连接数
NumResumedINC	自服务器启动以来的恢复 INC 连接数
NumDroppedSuspendedINC	已经挂起的删除 INC 连接数 (仅协调器)
NumSuspendedTxnRollbackINC	因 INC 失败而回退的全局 DML 事务数 (仅写入程序)

*示例*

显示一个挂起的事务和一个恢复的事务:



```

sp_iqmpxincstatistics
stat_name          stat_value
NumSuspendedINC   1
NumResumedINC     1
NumSuspendedTXNRollBackINC 0

```

## sp\_iqmpxinfo 过程

为 Multiplex 中每个节点返回一行。可从任意 Multiplex 节点运行。

### 语法

```
sp_iqmpxinfo
```

### 适用于

仅限 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE MULTIPLEX 系统特权。

### 注释

**sp\_iqmpxinfo** 过程返回：

列名	数据类型	描述
server_id	unsigned int	显示信息的服务器的标识符
server_name	char(128)	服务器的名称
connection_info	long varchar	已设置格式的字符串，包含用于 Multiplex 服务器之间 TCP/IP 连接的连接字符串的主机/端口部分。
db_path	long varchar	完整数据库路径
role	char(16)	'coordinator'   'writer'   'reader'
status	char(8)	'included'   'excluded'
mpx_mode	char(16)	'single'   'coordinator'   'writer'   'reader'   'unknown'
inc_state	char(16)	'active'   'not responding'   'timed out'
coordinator_failover	char(128)	故障转移服务器的名称
current_version	unsigned bigint	十进制格式的版本 ID
active_versions	long varchar	以逗号分隔的十进制格式版本 ID 列表。
private_connection_info	long varchar	已设置格式的字符串，包含用于 Multiplex 服务器之间专用 TCP/IP 连接的连接字符串的主机/端口部分

列名	数据类型	描述
mipc_priv_state	char(16)	'active' - 通过专用互连与该节点建立的 MIPC 连接是活动的   'not responding' - 通过专用互连与该节点建立的 MIPC 连接无响应。
mipc_public_state	char(16)	'active' - 通过公共互连与该节点建立的 MIPC 连接是活动的。  'not responding' - 通过公共互连与该节点建立的 MIPC 连接无响应。

### 示例

**sp\_iqmpxinfo** 的示例输出:

```
server_id,server_name,connection_info,db_path,role,
status,mpx_mode,inc_state,coordinator_failover,
current_version,active_versions,private_connection_
info,mipc_priv_state,mipc_public_state

1,'my_mpx1','host=(fe80::214:4fff:fe45:be26%2):1362
0,(fd77:55d:59d9:329:214:4fff:fe45:be2
6%2):13620,10.18.41.196:13620','/system3/users
/devices/s16900269/iqmpx1/mpx1.db',
'coordinator','included','coordinator','N/A',
'my_mpx2',0,,,'active','active'

2,'IQ_mpx2','host=system3:13625',
'/system3/users/devices/s16900269
/iqmpx_2/wk0001.db','writer','included',
'writer','active','IQ_mpx20','not responding','active'

3,'IQ_mpx3','host=system3:13630/system3/users/devi
ces/s16900269/iqmpx_3/mpx1.db','reader','included',
'unknown',timed out',
'IQ_mpx20','not responding',
'not responding'
```

### 另请参见

- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqmpxsuspendedconninfo 过程

显示协调器节点上当前挂起的连接和事务的相关详细信息。

### 语法

**sp\_iqmpxsuspendedconninfo**

### 适用于

仅限 Multiplex。

*特权*

您必须具有系统过程的 EXECUTE 特权。查看自己的挂起连接不需要具有系统特权。您还必须具有要查看数据库中的所有挂起连接，需要具有以下系统特权之一：

- DROP CONNECTION
- MONITOR
- SERVER OPERATOR

*注释*

返回：

**表 99. sp\_iqmpxsuspendedconninfo 列**

列名	数据类型	描述
ConnName	char(128)	连接名
ConnHandle	unsigned integer	连接标识符
GlobalTxnID	unsigned bigint	该连接上活动事务的全局事务标识符
MPXServerName	char(128)	发起 INC 连接的 Multiplex 服务器的名称
TimeInSuspended-State	integer	处于挂起状态的连接所用的总时间（以秒为单位）
SuspendTimeout	integer	挂起超时（以秒为单位） (2*MPX_LIVENESS_TIMEOUT)

*示例***sp\_iqmpxsuspendedconninfo**

```

ConnName          ConnHandle      GlobalTxnId
=====          =
'IQ_MPX_SERVER_P54'      14              112753

MPXServerName     TimeInSuspendedState
=====          =
'HP1_12356_IQ_mpx2'      37

SuspendTimeout
=====
360

```

## sp\_iqmpxvalidate 过程

检查 Multiplex 配置中是否有不一致。

*语法*

```
call dbo.sp_iqmpxvalidate ( 'show_msgs' )
```

*适用于*

仅限 Multiplex。

*特权*

您必须具有系统过程的 EXECUTE 特权。

*注释*

对表 SYS.SYSIQDBFILE 以及其它 Multiplex 事件和存储过程执行多项检查。可在任何服务器上运行。向调用程序返回严重级结果；值如下：

值	描述
0	未检测到错误
1	非预期的动态状态
2	非致命配置错误；例如，Multiplex 操作已损害
3	致命配置问题；例如，一个或多个服务器可能未启动

如果交互调用，那么除非调用参数不是“Y”，否则还会返回已发现错误的表（如果有）。

各项错误均显示严重级别。如果没有错误，则过程将返回 No errors detected。

## sp\_iqmpxversioninfo 过程

显示此服务器的当前版本信息，包括服务器类型（写入服务器、查询服务器、单节点模式）和同步状态。

*语法*

```
sp_iqmpxversioninfo
```

*适用于*

仅限 Multiplex。

*特权*

您必须具有系统过程的 EXECUTE 特权。

注释

表 100. sp\_iqmpxversioninfo 列

列	数据类型	描述
CatalogID	unsigned bigint	此服务器上的目录版本
VersionID	unsigned bigint	此服务器上可用的最新版本
OAVID	unsigned bigint	此服务器上的最旧活动版本
ServerType	char(1)	服务器的类型：“C”（协调器）、“W”（写入服务器）或“Q”（查询服务器）
CatalogSync	char(1)	目录同步：“T”（同步）或“F”（不同步）
WCatalogID	unsigned bigint	写入服务器上的目录版本
WVersionID	unsigned bigint	写入服务器上可用的最新版本

另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqobjectinfo 过程

返回数据库对象和子对象的分区和 dbspace 分配。

语法

```
sp_iqobjectinfo [ owner_name ] [ , object_name ] [ , object-type ]
```

参数

- **owner\_name** - 对象的所有者。如果指定，**sp\_iqobjectinfo** 将仅显示包含指定所有者的那些表的输出。如果未指定，**sp\_iqobjectinfo** 显示数据库中所有用户的表的相关信息。
- **object\_name** - 表的名称。如果未指定，**sp\_iqobjectinfo** 显示数据库中所有表的相关信息。
- **object-type** - 有效的 **table** 对象类型。

如果 **object-type** 是表，则必须用引号引起。

所有参数都是可选的，并且任何参数的提供均不受其它参数值的影响。

特权

您必须具有系统过程的 EXECUTE 特权。

**注释**

将输入参数与 **sp\_iqobjectinfo** 一起使用；您可以查询 **sp\_iqobjectinfo** 的结果，如果使用输入参数，而不是在查询的 **WHERE** 子句中使用谓词，执行效果会更好。例如，将查询 A 编写为：

```
SELECT COUNT(*) FROM sp_iqobjectinfo()
WHERE owner = 'DBA'
AND object_name = 'tab_case510'
AND object_type = 'table'
AND sub_object_name is NULL
AND dbspace_name = 'iqmain7'
AND partition_name = 'P1'
```

查询 B 对查询 A 进行了重新编写，改为使用 **sp\_iqobjectinfo** 输入参数：

```
SELECT COUNT(*) FROM sp_iqobjectinfo('DBA','tab_case510','table')
WHERE sub_object_name is NULL
AND dbspace_name = 'iqmain7'
AND PARTITION_NAME = 'P1'
```

查询 B 返回结果的速度要快于查询 A。将输入参数传递到 **sp\_iqobjectinfo** 后，该过程将进行比较，然后连接系统表中的少量记录，因此与查询 A 相比，工作量较少。在查询 B 中，过程本身使用了谓词，它返回的结果集较小，因此查询中使用的谓词数较少。

**sp\_iqobjectinfo** 存储过程支持用于解释 *owner\_name*、*object\_name* 和 *object\_type* 的通配符。它以 **LIKE** 子句匹配查询内部模式的方式显示匹配指定模式的所有 **dbspace** 的信息。

返回（表类型中）特定或所有数据库对象及其子对象的所有分区和 **dbspace** 分配。子对象为列、索引、主键、唯一约束和外键。

**表 101. sp\_iqobjectinfo 列**

列名	描述
owner	对象所有者的名称。
object_name	位于 <b>dbspace</b> 上的（表类型）对象的名称。
sub_object_name	<b>dbspace</b> 中对象的名称。
object_type	对象的类型（列、索引、主键、唯一约束、外键、分区或表）。
object_id	对象的全局对象 ID。
id	对象的表 ID。
dbspace_name	对象所在的 <b>dbspace</b> 的名称。在已分区对象的特殊元行中显示字符串 "[multiple]"。[multiple] 行表示输出中存在多个描述表或列的行。
partition_name	给定对象的分区的名称。

## 示例

**注意：**以下示例显示 iqdemo 数据库中的对象，以便更好地阐释输出。iqdemo 包括一个名为 iq\_main 的示例用户 dbspace，您自己的数据库中可能不包括此 dbspace。

显示某特定用户拥有的特定数据库对象及子对象的分区和 dbspace 分配的相关信息：

```
sp_iqobjectinfo GROUPO, Departments
```

owner	object_name	sub_object_name	object_type	obj
ect_id	id			
GROUPO	Departments	(NULL)	table	3
632	738			
GROUPO	Departments	DepartmentID	column	3
633	738			
GROUPO	Departments	DepartmentName	column	3
634	738			
GROUPO	Departments	DepartmentHeadID	column	3
635	738			
GROUPO	Departments	DepartmentsKey	primary	
key	83	738		
GROUPO	Departments	FK_DepartmentHeadID_EmployeeID	foreign	
key	92	738		
dbspace_name	partition_name			
iq_main	(NULL)			
iq_main	(NULL)			
iq_main	(NULL)			
iq_main	(NULL)			
iq_main	(NULL)			
iq_main	(NULL)			

显示 *object-type table* 的某特定用户拥有的特定数据库对象及子对象的分区和 dbspace 分配的相关信息：

```
sp_iqobjectinfo DBA, sale, 'table'
```

owner	object_name	sub_object_name	object_type	object_id	id
DBA	sale	(NULL)	table	3698	742
DBA	sale	prod_id	column	3699	742
DBA	sale	month_num	column	3700	742
DBA	sale	rep_id	column	3701	742
DBA	sale	sales	column	3702	742
dbspace_name	partition_name				
iq_main	(NULL)				
iq_main	(NULL)				
iq_main	(NULL)				
iq_main	(NULL)				
iq_main	(NULL)				

## 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqpassword 过程

更改用户口令。

### 语法 1

```
call sp_iqpassword ( 'caller_password' , 'new_password' [,
' user_name' ] )
```

### 语法 2

```
sp_iqpassword 'caller_password' , 'new_password' [, 'user_name' ]
```

### 参数

- **caller\_password** - 您的口令。在您更改自己的口令时，该口令是您的旧口令。如果具有 **CHANGE PASSWORD** 系统特权的用户要更改其他用户的口令，**caller\_password** 则为执行更改的用户口令。
- **new\_password** - 用户的新口令或 *loginname* 的新口令。
- **user\_name** - 要由具有 **CHANGE PASSWORD** 系统特权的另一用户更改口令的用户的登录名。更改自己的口令时，请勿指定 **user\_name**。

### 特权

您必须具有系统过程的 **EXECUTE** 特权。设置自己的口令不需要具备其它系统特权。设置其他用户的口令需具备 **CHANGE PASSWORD** 系统特权。

### 注释

用户口令是一个标识符。任何用户均可使用 **sp\_iqpassword** 来更改自己的口令。更改任何现有用户的口令需具备 **CHANGE PASSWORD** 系统特权。

标识符的最大长度是 128 个字节。当以下任一条件成立时，标识符必须用双引号引起来或用中括号括起来：

- 标识符包含空格。
- 标识符的首字符不是字母字符（定义将在后面提供）。
- 标识符包含保留字。
- 标识符包含字母和数字以外的其它字符。  
字母字符包括字母表中的字母，以及下划线( )、at 符号(@)、井号(#)和美元符号(\$)。数据库归类序列指出了哪些字符被视为字母字符或数字字符。

### 示例

将已登录用户的口令从 **irk103** 更改为 **exP984**：

```
sp_iqpassword 'irk103', 'exP984'
```

如果已登录用户具有 **CHANGE PASSWORD** 系统特权或该用户为用户 **joe**，将用户 **joe** 的口令从 **eprr45** 更改为 **pdi032**：



```
call sp_iqpassword ('epr45', 'pdi932', 'joe')
```

### 另请参见

- `sp_expireallpasswords` 系统过程 (第 628 页)
- `sp_iqaddlogin` 过程 (第 386 页)
- `sp_iqcopyloginpolicy` 过程 (第 413 页)
- `sp_iqmodifylogin` 过程 (第 469 页)

## sp\_iqpkkeys 过程

按表、列、表所有者或为数据库中的所有 SAP Sybase IQ 表显示主键和主键约束的相关信息。

### 语法

```
sp_iqpkkeys { [ table-name ], [ column-name ], [ table-owner ] }
```

### 参数

- **table-name** - 基表或全局临时表的名称。如果指定，该过程将仅返回在指定表上定义的主键的相关信息。
- **column-name** - 列的名称。如果指定，该过程将仅返回指定列上主键的相关信息。
- **table-owner** - 表的所有者或表。如果指定，该过程将仅返回指定所有者拥有的表上主键的相关信息。

### 特权

您必须具有系统过程的 EXECUTE 特权。

### 注释

可指定一个或多个参数。如果不指定前两个参数中的任何一个参数，但指定序列中的下一个参数，则必须用 NULL 替换省略的参数。如果未指定任何参数，则会显示数据库中所有表上的所有主键的说明。如果任意指定参数无效，则输出中不会显示任何行。

**表 102. sp\_iqpkkeys 用法示例**

语法	输出
<code>sp_iqpkkeys sales</code>	显示表 sales 上定义的主键的相关信息
<code>sp_iqpkkeys sales, NULL, DBA</code>	显示 DBA 拥有的表 sales 上定义的主键的相关信息
<code>sp_iqpkkeys sales, store_id, DBA</code>	显示 DBA 拥有的表 sales 中列 store_id 上所定义主键的相关信息

语法	输出
sp_iqpkeys NULL, NULL, DBA	显示 DBA 拥有的所有表上定义的主键的相关信息

sp\_iqpkeys 存储过程显示有关数据库中基表和全局临时表的主键的下列信息：

**表 103. sp\_iqpkeys 列**

列名	描述
table_name	表的名称
table_owner	表的所有者
column_name	定义主键的列的名称
column_id	列 ID
constraint_name	主键约束的名称
constraint_id	主键约束 ID

*示例*

显示在表 sales1 的列上定义的主键：

```
sp_iqpkeys sales1
table_name table_owner column_name column_id constraint_name constraint_id
sales1      DBA         store_id      1          MA114          114
```

显示在表 sales2 的列上定义的主键：

```
sp_iqpkeys sales2
table_name table_owner column_name column_id constraint_name constraint_id
sales2     DBA         store_id,    1,2        MA115          115
           order_num
```

显示在表 sales2 的列 store\_id 上定义的主键：

```
sp_iqpkeys sales2, store_id
table_name table_owner column_name column_id constraint_name constraint_id
sales2     DBA         store_id      1          MA115          115
```

**另请参见**

- sp\_iqcolumn 过程 (第 401 页)
- sp\_iqconstraint 过程 (第 409 页)
- sp\_iqdatatype 过程 (第 415 页)

- sp\_iqevent 过程 (第 436 页)
- sp\_iqhelp 过程 (第 441 页)
- sp\_iqindex 和 sp\_iqindex\_alt 过程 (第 447 页)
- sp\_iqprocparm 过程 (第 489 页)
- sp\_iq\_reset\_identity 过程 (第 499 页)
- sp\_iqtable 过程 (第 539 页)
- sp\_iqview 过程 (第 554 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqprocedure 过程

显示与系统过程和用户定义过程有关的信息。

### 语法

```
sp_iqprocedure [ proc-name ], [ proc-owner ], [ proc-type ]
```

### 参数

- **proc-name** - 过程的名称。
- **proc-owner** - 过程的所有者。
- **proc-type** - 过程的类型。允许的值如下：
  - **SYSTEM** - 只显示有关系统过程 (由用户 SYS 或 dbo 拥有的过程) 的信息
  - **ALL** - 显示有关用户过程和系统过程的信息
  - **其它任何值** - 显示有关用户过程的信息

可以在不指定任何参数的情况下调用 **sp\_iqprocedure** 过程。如果不指定任何参数, 则缺省情况下仅显示用户定义过程 (不归 dbo 或 SYS 所有的过程) 的相关信息。

### 特权

您必须具有系统过程的 EXECUTE 特权。

### 注释

如果不指定前两个参数中的任何一个参数, 但指定序列中的下一个参数, 则必须用 NULL 替换省略的参数。例如, sp\_iqprocedure NULL, NULL, SYSTEM 和 sp\_iqprocedure NULL, user1。

**表 104. sp\_iqprocedure 用法示例**

语法	输出
sp_iqprocedure	显示数据库中不归 dbo 或 SYS 所有的所有过程的相关信息
sp_iqprocedure sp_test	显示过程 <b>sp_test</b> 的相关信息

语法	输出
sp_iqprocedure non_existing_proc	不返回任何行，因为过程 <b>non_existing_proc</b> 不存在
sp_iqprocedure NULL, DBA	显示 DBA 拥有的所有过程的相关信息
sp_iqprocedure sp_test, DBA	显示 DBA 拥有的过程 <b>sp_test</b> 的相关信息
sp_iqprocedure sp_iqtable	过程 <b>sp_iqtable</b> 不是系统过程。如果没有同样名为 <b>sp_iqtable</b> 的用户定义的过程，则不返回任何行。（缺省情况下，仅返回用户定义过程。）
sp_iqprocedure sp_iqtable, dbo	不返回任何行，因为过程 <b>sp_iqtable</b> 不是用户过程（缺省情况下，仅返回用户过程）
sp_iqprocedure NULL, NULL, SYSTEM	显示所有系统过程（由 dbo 或 SYS 拥有）的相关信息
sp_iqprocparm sp_iqtable, NULL, SYSTEM	显示系统过程 <b>sp_iqtable</b> 的相关信息
sp_iqprocedure sp_iqtable, dbo, ALL	显示 dbo 拥有的系统过程 <b>sp_iqtable</b> 的相关信息

**sp\_iqprocedure** 存储过程显示与数据库中的过程有关的信息。如果指定了一个或多个参数，将根据指定参数对结果进行过滤。例如，如果指定 *proc-name*，则仅显示有关指定过程的信息。如果指定 *proc-owner*，则 **sp\_iqprocedure** 仅返回指定所有者拥有的过程的相关信息。如果未指定任何参数，**sp\_iqprocedure** 显示数据库中所有用户定义的过程的信息。

**sp\_iqprocedure** 过程返回以下列中的信息：

表 105. sp\_iqprocedure 列

列名	描述
proc_name	过程的名称
proc_owner	过程的所有者
proc_defn	用于创建过程的命令。对于隐藏过程，将显示关键字“HIDDEN”。
replicate	如果过程是 Replication Server 安装中的主数据源，将显示 Y；否则显示 N。
srvid	如果过程在远程数据库服务器上，则指示该远程服务器。
remarks	注释字符串

示例

显示有关用户定义过程 *sp\_test* 的信息：

```

sp_iqprocedure sp_test

proc_name      proc_owner      proc_defn      replicate      srvid      r
emarks
sp_test        DBA              create procedure N      (NULL)      (NULL)
                DBA.sp_test(in n1
                integer)
                begin message 'sp_test' end

```

显示由用户 DBA 拥有的所有过程的相关信息：

```

sp_iqprocedure NULL, DBA

proc_name      proc_owner      proc_defn      replicate      srvid      r
emarks
sp_test        DBA              create procedure N      (NULL)      (NULL)
                DBA.sp_test(in n1
                integer)
                begin message 'sp_test' end
sp_dept        DBA              create procedure N      (NULL)      (NULL)
                DBA.sp_dept() begin end

```

另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqprocparm 过程

显示与存储过程参数有关的信息，包括结果集变量和 SQLSTATE/SQLCODE 错误值。

语法

```

sp_iqprocparm [ proc-name ], [ proc-owner ], [ proc-type ]

```

参数

- **proc-name** - 过程的名称。
- **proc-owner** - 过程的所有者。
- **proc-type** - 过程的类型。允许的值如下：
  - **SYSTEM** - 只显示有关系统过程（由用户 SYS 或 dbo 拥有的过程）的信息
  - **ALL** - 显示有关用户过程和系统过程的信息
  - **其它任何值** - 显示有关用户过程的信息

可以在不指定任何参数的情况下调用 **sp\_iqprocparm**。如果不指定任何参数，则显示用户定义过程（不归 dbo 或 SYS 所有的过程）的输入/输出和结果参数。

特权

您必须具有系统过程的 EXECUTE 特权。

## 注释

如果不指定前两个参数中的任何一个参数，但指定序列中的下一个参数，则必须用 NULL 替换省略的参数。例如，`sp_iqprocparm NULL, NULL, SYSTEM` 和 `sp_iqprocparm NULL, user1`。

表 106. `sp_iqprocparm` 用法示例

语法	输出
<code>sp_iqprocparm</code>	显示数据库中不归 <code>dbo</code> 或 <code>sys</code> 所有的所有过程的参数
<code>sp_iqprocparm sp_test</code>	显示过程 <code>sp_test</code> 的相关信息
<code>sp_iqprocparm non_existing_proc</code>	不返回任何行，因为过程 <code>non_existing_proc</code> 不存在
<code>sp_iqprocparm NULL, DBA</code>	显示 <code>DBA</code> 拥有的所有过程的参数
<code>sp_iqprocparm sp_test, DBA</code>	显示 <code>DBA</code> 拥有的过程 <code>sp_test</code> 的参数
<code>sp_iqprocparm sp_iqtable</code>	<code>sp_iqtable</code> 是一个系统过程。如果没有同样名为 <code>sp_iqtable</code> 的用户定义的过程，则不返回任何行。（缺省情况下，仅返回用户定义过程。）
<code>sp_iqprocparm sp_iqtable, dbo</code>	不返回任何行，因为过程 <code>sp_iqtable</code> 不是用户过程。（缺省情况下，仅返回用户过程。）
<code>sp_iqprocparm NULL, NULL, SYSTEM</code>	显示所有系统过程（由 <code>dbo</code> 或 <code>sys</code> 拥有）的参数
<code>sp_iqprocparm sp_iqtable, NULL, SYSTEM</code>	显示系统过程 <code>sp_iqtable</code> 的参数
<code>sp_iqprocparm sp_iqtable, dbo, ALL</code>	显示由 <code>dbo</code> 拥有的系统过程 <code>sp_iqtable</code> 的参数

`sp_iqprocparm` 存储过程可显示有关存储过程参数的信息，包括结果集变量和 `SQLSTATE/SQLCODE` 错误值。如果指定了一个或多个参数，将根据指定参数对结果进行过滤。例如，如果指定了 `proc-name`，将仅显示与该指定过程的参数有关的信息。如果指定 `proc-owner`，则 `sp_iqprocparm` 仅返回指定所有者拥有的过程参数的相关信息。如果未指定任何参数，`sp_iqprocparm` 显示有关数据库中所有用户定义过程的参数的信息。

`sp_iqprocparm` 过程返回以下列中的信息：

表 107. sp\_iqprocparm 列

列名	描述
proc_name	过程的名称
proc_owner	过程的所有者
parm_name	参数名
parm_type	参数类型为下列值之一： <ul style="list-style-type: none"> <li>普通参数（变量）</li> <li>结果变量：与返回结果集的过程一起使用</li> <li>SQLSTATE 错误值</li> <li>SQLCODE 错误值</li> </ul>
parm_mode	参数模式：参数为过程提供一个值；参数返回一个值；参数既提供一个值又返回一个值；参数既不提供值也不返回值。参数模式为以下模式之一： <ul style="list-style-type: none"> <li>in：参数为过程提供一个值</li> <li>out：参数返回一个值</li> <li>inout：参数既提供一个值又返回一个值</li> <li>空值：参数既不提供值也不返回值</li> </ul>
domain_name	参数的数据类型名称，如 SYSDOMAIN 系统表中所列
width	字符串参数的长度、数值参数的精度以及所有其它数据类型的存储字节数
scale	数值数据类型参数的小数点后的位数，对于所有其它数据类型该值为零
default	参数的缺省值，保留为字符串

**示例**

显示用户定义过程 sp\_test 的参数的相关信息：

```
sp_iqprocparm sp_test
```

```
proc_name proc_owner parm_name parm_type parm_mode domain_name width
h scale default
```

```
sp_test DBA ID
normal in integer 4 0 (NULL)
```

显示有关系统过程 sp\_iqshowcompression 的参数的信息：

```
sp_iqprocparm sp_iqshowcompression, dbo, system
```

```
proc_name proc_owner parm_name parm_type parm_mode
domain_name width scale default
```

```
sp_iqshowcompression dbo @owner_name normal in
```

char	128	0	(NULL)			
sp_iqshowcompression		dbo		@table_name	normal	in
char	128	0	(NULL)			
sp_iqshowcompression		dbo		@column_name	normal	in
char	128	0	(NULL)			
sp_iqshowcompression		dbo		Column	result	out
char	128	0	(NULL)			
sp_iqshowcompression		dbo		Compression	result	out
char	3	0	(NULL)			

### 另请参见

- sp\_iqcolumn 过程 (第 401 页)
- sp\_iqconstraint 过程 (第 409 页)
- sp\_iqdatatype 过程 (第 415 页)
- sp\_iqevent 过程 (第 436 页)
- sp\_iqhelp 过程 (第 441 页)
- sp\_iqindex 和 sp\_iqindex\_alt 过程 (第 447 页)
- sp\_iqkeys 过程 (第 485 页)
- sp\_iq\_reset\_identity 过程 (第 499 页)
- sp\_iqtable 过程 (第 539 页)
- sp\_iqview 过程 (第 554 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqrebuildindex 过程

重建列索引。

要重建索引来替代缺省索引 FP，请指定索引名。无论

FP\_NBIT\_IQ15\_COMPATIBILITY 设置如何，**sp\_iqrebuildindex** 行为相同。

### 语法

```
sp_iqrebuildindex (table_name, index_clause)
```

**注意：** 第三方参考文档介绍不支持的 **sp\_iqrebuildindex** 语法。请注意，在索引子句中指定表名将产生错误：

```
sp_iqrebuildindex tbl, 'column tbl.c1'
```

### 参数

- **table\_name** - 进行索引重建过程的部分限定表名或完全限定表名。如果用户既拥有该表又执行该过程，则可以使用部分限定名；否则，表名必须为完全限定名。
- **index\_clause** - 下列字符串中的一个或多个，以空格分隔：

**column** *column\_name* [*count*]

**index** *index\_name*



每个 *column\_name* 或 *index\_name* 必须引用指定表中的某列或某索引。如果多次指定 *column\_name* 或 *index\_name*，该过程将返回错误，并且不会重建任何索引。

*count* 是一个非负数，表示 **IQ UNIQUE** 值。在 **CREATE TABLE** 语句中，**IQ UNIQUE (count)** 表示给定列中所能容纳的不同值的估计数量。不同的值的数量会影响查询速度和存储要求。

**MERGEALL** 和 **RETIER** 是特定于 HG 索引操作的关键字：

```
sp_iqrebuildindex(' table name' , ' index index name [ MERGEALL |
RETIER ] ' )
```

如果从 HG 索引的某操作中忽略 **MERGEALL** 或 **RETIER**，**sp\_iqrebuildindex** 将截断并根据列数据重建整个 HG 索引。

**MERGEALL** 可合并分层 HG 索引的所有层，并且将内容移动到相应的层中：

```
sp_iqrebuildindex(' table name' , ' index index name MERGEALL ' )
```

该合并可以确保在分层 HG 索引中只含有一个活动的子索引。如果有很多删除记录（如 **sp\_iqindexmetadata** 所示），**MERGEALL** 操作可以降低分层索引的查询访问时间。只有使用 **index** 子句并且指定索引为 HG 类型时才支持 **MERGEALL** 操作。

**RETIER** 是特定于 HG 索引的关键字，可将 HG 索引的格式从不分层的 HG 更改为分层的 HG，或将分层的 HG 更改为不分层的 HG：

```
sp_iqrebuildindex(' table name' , ' index index name RETIER ' )
```

**RETIER** 切换 HG 索引的格式：

- **RETIER** 将分层 HG 索引转换为单个不分层 HG 索引。禁用分层元数据，并且只维护一个子索引。
- **RETIER** 将不分层的 HG 索引转换成分层 HG 索引，并将包含所有数据的单个子索引推送到相应层中。

**MERGEALL** 和 **RETIER** 仅在以下情况下受支持：使用 **index** 子句并且指定的索引为 HG 索引。

您必须指定关键字 **column** 和 **index**。这些关键字不区分大小写。

---

**注意：** 此过程不支持 **TEXT** 索引。要重建 **TEXT** 索引，必须首先删除再重新创建索引。

---

#### 特权

您必须具有系统过程的 **EXECUTE** 特权。必须具有以下一种：

- **INSERT ANY TABLE** 系统特权
- 用于重建索引的针对表的 **INSERT** 特权
- 您拥有该表

*注释*

如果指定列名，**sp\_iqrebuildindex** 会为该列重建缺省 FP 索引；不需要索引名。除列名之外，如果您还指定由 SAP Sybase IQ 分配的缺省 FP 索引名，**sp\_iqrebuildindex** 将返回一个错误。

在数据类型 LONG VARCHAR (CLOB) 的列中，**sp\_iqrebuildindex** 重建 WD 索引。

含有 IQ UNIQUE *n* 值的列可确定 **sp\_iqrebuildindex** 将列重建为 Flat FP 还是 NBit。如果 IQ UNIQUE *n* 值设置为 0，索引将重建为 Flat FP。如果 *n* 值大于 0 小于 2,147,483,647，索引将重建为 NBit。不含 *n* 值的 NBit 列将重建为 NBit。即使不指定计数，**sp\_iqrebuildindex** 也会将 NBit 列重建为 NBit。如果指定计数，*n* 值必须大于索引中已存在的唯一值数。

如果您使用 Flat FP 索引重建列，并且列不包含 IQ UNIQUE *n* 值，则 **sp\_iqrebuildindex** 会将索引重建为 Flat FP 直至达到 FP\_NBIT\_AUTOSIZE\_LIMIT 和 FP\_NBIT\_LOOKUP\_MB 选项中定义的限制。如果 FP\_NBIT\_ENFORCE\_LIMITS=ON 和基数超过计数，则指定平面列的 *n* 值将抛出错误。

用户可以通过 **sp\_iqrebuildindex** 缺省接口根据现有的 FP 索引重新创建整个 HG 索引。**sp\_iqrebuildindex** 重新读取 FP 索引列值并创建 HG 索引。但是，这样会保留层大小、连续装载大小等所有元数据。

*示例*

重建列 *Surname* 的缺省索引 FP:

```
sp_iqrebuildindex 'empl', 'column dept_id'
```

或者:

```
call sp_iqrebuildindex ('empl', 'column dept_id')
```

创建列 *c1* 的平面索引 FP:

```
CREATE TABLE mytable (c1 int IQ UNIQUE (0))
```

将缺省索引 Flat FP 转换为估计非重复计数为 1024 的索引 Nbit:

```
sp_iqrebuildindex 'mytable', 'column c1 1024'
```

或者:

```
call sp_iqrebuildindex ('mytable', 'column c1 1024')
```

---

**注意:** 在 HG 索引上运行 **sp\_iqrebuildindex** 时，用户将发现临时性能下降。

---

**另请参见**

- **sp\_iqindexfragmentation** 过程 (第 452 页)
- **sp\_iqrowdensity** 过程 (第 503 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqrebuildindex 过程

重建 16.0 之前列宽超过 255 个字节的 FP 索引。

### 语法

```
sp_iqrebuildindexwide [ table_name ] [, table_owner] [, level]
```

### 参数

- **table\_name** – 标识表。该参数为必需参数，但可以包括空字符串。用空字符串替代 **table\_name** 将为命令中指定的 **table owner** 重建数据库中的所有宽列表。用空字符串替代 **table\_name** 和 **table owner** 将重建数据库中的所有宽列表。
- **table\_owner** – 是表的所有者。显式 **table owner** 名是可选的；缺省值为空字符串。用空字符串替代 **table\_name** 将为命令中指定的 **table owner** 重建数据库中的所有宽列表。使用显式 **table\_name** 和空字符串作为 **table\_owner** 将为所有用户重建表。用空字符串替代 **table\_owner** 和 **table owner** 将重建数据库中的所有宽列表。
- **level** – 决定着 **sp\_iqrebuildindexwide** 重建表的程度。该参数为可选参数并且包括四个选项：
  - 级别 '1' 将为给定用户重建 16.0 之前列宽超过 255 个字节的所有列
  - 级别 '2' 将重建所有标记化 FP（即，16.0 之前的单/双/三字节 FP、可投影的单字节 FP 和双字节 FP 以及 16.0 NBit FP）、VARCHAR 或 VARBINARY 列以及 16.0 之前列宽超过 255 个字节的的所有列
  - 级别 '3' 重建 16.0 之前列宽超过 255 个字节的的所有固定 Flat FP
  - 级别 '4' 可应用级别 1、2 和 3 并可重建 16.0 之前所有宽度大于 255 字节的列、全部标记化 FP、全部 varchar 和 varbinary 列和全部 FLAT 固定 FP

忽略级别参数在级别 '1' 执行 **sp\_iqrebuildindexwide**。

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种：

- INSERT ANY TABLE 系统特权
- 用于重建索引的针对表的 INSERT 特权
- 您拥有该表

### 注释

必须重建列宽超过 255 个字符的 CHAR、VARCHAR、BINARY 和 VARBINARY 列以及迁移到 SAP Sybase IQ 16.0 数据库中的所有 LONG VARCHAR 列和 LONG BINARY 列，然后数据库引擎才能针对这些列执行读/写活动。

首次打开表进行读写访问时，SAP Sybase IQ 便隐式重建这些类型的列。而 **sp\_iqrebuildindexwide** 会将这些列重建为由 level 参数定义的状态。

**sp\_iqrebuildindexwide** 将执行结果写入 SAP Sybase IQ 消息文件 (<dbname>.iqmsg)

## 示例

- **示例 1** – 假设 vartab 表由 DBA 拥有并且运行

```
select * from sp_iqrowdensity('table user1.vartab3')
```

将返回以下结果:

```
DBA.vartab  rid  Flat style FP
DBA.vartab  lvb1 Long varbinary FP
DBA.vartab  blob2 Long binary FP
DBA.vartab  blob1 Long binary FP
DBA.vartab  vc1  Flat style FP
DBA.vartab  lvc1 Long varchar FP
DBA.vartab  clob2 Long binary FP
DBA.vartab  clob1 Long binary FP
DBA.vartab  part Flat style FP
DBA.vartab  vb1  Flat style FP
```

使用 vartab 作为 **table\_name**、DBA 作为 **table\_owner**，并以级别 '1' 运行 **sp\_iqrebuildindexwide**:

```
call sp_iqrebuildindexwide('vartab', 'DBA', 1)
```

重建列 clob1、colb2、lvc1、lvb1、blob1 和 blob2，并在 .iqmsg 中写入以下消息:

```
Index Rebuild in progress for "DBA"."vartab" column "clob1" column
"clob2" column "lvc1" column "lvb1" column "blob1" column "blob2"
```

- **示例 2** – 假设 vartab3 表由 user1 拥有并且运行

```
select * from sp_iqrowdensity('table user1.vartab3')
```

将返回以下结果:

```
user1.vartab3  rid  Flat style FP
user1.vartab3  part Flat style FP
user1.vartab3  vb1 Long varbinary FP
user1.vartab3  tk5 Two Byte FP
user1.vartab3  tk4 One Byte FP
user1.vartab3  tk3 Three Byte FP
user1.vartab3  vc1 Long varchar FP
user1.vartab3  tk2 Projectable Two Byte FP
user1.vartab3  b1  Flat style FP
user1.vartab3  c1  Long varchar FP
user1.vartab3  tk1 Projectable One Byte FP
```

使用 vartab3 作为 **table\_name**、user1 作为 **table\_owner**，并以级别 '2' 运行 **sp\_iqrebuildindexwide**:

```
sp_iqrebuildindexwide('vartab3', 'user1', 2)
```

重建列 vc1、vb1、c1、b1、tk1、tk2、tk3、tk4 和 tk5，并在 .iqmsg 中写入以下消息:

```
Index Rebuild in progress for "user1"."vartab3" column "vc1"
column "vb1" column "c1" column "b1" column "tk1" column "tk2"
column "tk3" column "tk4" column "tk5"
```

- **示例 3** – 假设 vartab3 表由 user1 拥有并且运行

```
select * from sp_iqrowdensity('table user1.vartab3')
```

将返回以下结果:

```
user1.vartab3 part Flat style FP
user1.vartab3 c1 2 Bit FP
user1.vartab3 tk4 2 Bit FP
user1.vartab3 tk5 2 Bit FP
user1.vartab3 vb1 2 Bit FP
user1.vartab3 tk3 2 Bit FP
user1.vartab3 tk2 2 Bit FP
user1.vartab3 tk1 2 Bit FP
user1.vartab3 b1 2 Bit FP
user1.vartab3 rid Flat style FP
user1.vartab3 vc1 2 Bit FP
```

使用 vartab3 作为 **table\_name**、user1 作为 **table\_owner**，并以级别 '3' 运行

**sp\_iqrebuildindexwide:**

```
call sp_iqrebuildindexwide('vartab3', 'user1', 3)
```

重建列 rid 和 part，并在 .iqmsg 中写入以下消息:

```
Index Rebuild in progress for "user1"."vartab3" column "rid"
column "part"
```

对重建表运行

```
select * from sp_iqrowdensity('table user1.vartab3')
```

将返回以下结果:

```
user1.vartab3 rid 2 Bit FP
user1.vartab3 vc1 2 Bit FP
user1.vartab3 vb1 2 Bit FP
user1.vartab3 b1 2 Bit FP
user1.vartab3 tk1 2 Bit FP
user1.vartab3 tk4 2 Bit FP
user1.vartab3 tk5 2 Bit FP
user1.vartab3 tk3 2 Bit FP
user1.vartab3 c1 2 Bit FP
user1.vartab3 part 2 Bit FP
user1.vartab3 tk2 2 Bit FP
```

**另请参见**

- sp\_iqindexfragmentation 过程 (第 458 页)
- sp\_iqrebuildindex 过程 (第 492 页)

## sp\_iqrename 过程

重命名用户创建的表、列、索引、约束（唯一、主键、外键和检查）、存储过程和函数。

### 语法

```
sp_iqrename object-name, new-name [, object-type ]
```

### 参数

- **object-name** – 用户创建的对象的原名称。

或者，可以指定 *owner-name* 为部分 *object-name* 作为 *owner-name.object-name*，其中 *owner-name* 是重命名对象所有者的名称。如果未指定 *owner-name*，则假定调用 **sp\_iqrename** 的用户为对象的所有者。只有调用 **sp\_iqrename** 的用户具有重命名对象所需的特权时，才能成功地重命名对象。

如果要重命名的对象为列、索引或约束，则必须指定与该对象相关联的表的名称。对于列、索引或约束，*object-name* 可以是 *table-name.object-name* 或 *owner-name.table-name.object-name* 形式。

- **new-name** – 对象的新名称。此名称必须符合标识符规则，并且对于要重命名的对象的类型而言必须唯一。
- **object-type** – 可选参数，用于指定要重命名的用户创建对象的类型，即对象 *object-name* 的类型。*object-type* 参数可用大写或小写字母进行指定。

有效值包括：

- **列** – 要重命名的对象是列
- **索引** – 要重命名的对象是索引
- **constraint** – 要重命名的对象是唯一约束、主键约束、检查约束或参照（外键）约束
- **procedure** – 要重命名的对象是函数
- **未指定 object-type** – 要重命名的对象是表

---

**警告！** 您必须对将要通过 **sp\_iqrename** 重命名的对象相关的对象（过程、函数以及视图）的定义进行相应地更改。**sp\_iqrename** 过程不会自动更新相关对象的定义。您必须手动更改它们。

---

### 特权

您必须具有系统过程的 EXECUTE 特权。重命名特定对象类型需要以下各种不同的特权：

- **ALTER ANY OBJECT 系统特权** – 您可以重命名任何对象。
- **ALTER ANY TABLE 系统特权** – 您可以重命名任何表、列或约束。
- **ALTER ANY INDEX 系统特权** – 您可以重命名任何索引，但不能重命名表或列。

- 对表的 **REFERENCES 特权** - 您只能重命名该表的索引。
- 表的 **ALTER 特权** - 您只能重命名表、表列和约束。
- 您拥有该表 - 您可以重命名拥有的任何对象。

您必须对运行此过程所需的对象具有独占访问权。

#### 注释

**sp\_iqrename** 存储过程重命名用户创建的表、列、索引、约束（唯一、主键、外键和检查）和函数。

如果尝试使用对于某类型对象不唯一的名称来重命名此对象，则 **sp\_iqrename** 将返回消息“项目已存在”。

**sp\_iqrename** 不支持对视图、过程、事件或数据类型进行重命名。如果指定 **event** 或 **datatype** 作为 *object-type* 参数，则 **sp\_iqrename** 将返回消息“不支持此功能。”。

您也可以使用 **ALTER INDEX** 语句和 **ALTER TABLE** 语句的 **RENAME** 子句进行重命名。

#### 示例

将用户 shweta 拥有的表 titles 重命名为 books:

```
sp_iqrename shweta.titles, books
```

将表 books 的列 id 重命名为 isbn:

```
sp_iqrename shweta.books.id, isbn, column
```

将表 books 中的索引 idindex 重命名为 isbnindex:

```
sp_iqrename books.idindex, isbnindex, index
```

将表 books 中的主键约束 prim\_id 重命名为 prim\_isbn:

```
sp_iqrename books.prim_id, prim_isbn, constraint
```

#### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iq\_reset\_identity 过程

将与指定表关联的标识/自动增量列的种子设置为指定值。

#### 语法

```
sp_iq_reset_identity (table_name, table_owner, value)
```

#### 参数

您必须指定 *table\_name*、*table owner* 和 *value*。

#### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种:

- ALTER ANY TABLE 系统特权
- ALTER ANY OBJECT 系统特权
- 表的 ALTER 特权
- 您拥有该表

### 注释

标识/自动增量列存储自动生成的编号。生成的值是传入数据的唯一标识符。这些有序的值是自动生成的，并且绝不重复使用，即使行已从表中删除也是如此。指定的种子值会替换缺省种子值，并且在数据库关闭和出故障时也保持不变。

### 另请参见

- sp\_iqcolumn 过程 (第 401 页)
- sp\_iqconstraint 过程 (第 409 页)
- sp\_iqdatatype 过程 (第 415 页)
- sp\_iqevent 过程 (第 436 页)
- sp\_iqhelp 过程 (第 441 页)
- sp\_iqindex 和 sp\_iqindex\_alt 过程 (第 447 页)
- sp\_iqpkeys 过程 (第 485 页)
- sp\_iqprocparm 过程 (第 489 页)
- sp\_iqtable 过程 (第 539 页)
- sp\_iqview 过程 (第 554 页)
- 确定数据库所使用的安全模型 (第 381 页)

### **sp\_iq\_reset\_identity 过程示例**

使用示例作为 **sp\_iq\_reset\_identity** 用法的参考。

下面的示例创建起始种子为 50 的标识列：

```
CREATE TABLE mytable(c1 INT identity)
call sp_iq_reset_identity('mytable', 'dba', 50)
```

### **sp\_iqrestoreaction 过程**

显示在给定日期之后要使数据库达到一致状态所需要的恢复操作。

### 语法

```
sp_iqrestoreaction 'timestamp'
```

### 参数

- **timestamp** - 指定过去的日期目标。

### 特权

您必须具有系统过程的 EXECUTE 特权。



**注释**

如果不能使数据库达到时间戳一致状态，`sp_iqrestoreaction` 将返回错误。否则，建议执行可将数据库返回到一致状态的恢复操作。

可以恢复数据库的公共点与恰好在指定时间戳之前备份读写文件的上次备份时间重合。备份只能是包括全部内容或读写文件。

根据备份时间，输出可能不完全是升序。如果备份存档包括多个只读 `dbfile`，则它可能包含多个行（具有相同的备份时间和备份 ID）。

如果多次备份只读 `dbfile` 或 `dbspace`，则恢复使用上次备份。相应的备份时间可以在指定时间戳之后，只要 `dbspace/dbfile` 更改 ID 与恢复的最后一个读写备份中记录的 `dbspace/dbfile` 更改 ID 相匹配。

`sp_iqrestoreaction` 返回以下内容：

**表 108. sp\_iqrestoreaction 列**

列名	描述
<code>sequence_number</code>	要执行的步骤的顺序
<code>backup_id</code>	备份事务的标识符
<code>backup_archive_list</code>	备份中存档文件的列表
<code>backup_time</code>	备份执行的时间
<code>virtual_type</code>	虚拟备份的类型：“Non-virtual”、“Decoupled”或“Encapsulated”
<code>restore_dbpace</code>	可以为空。表示要从备份存档中恢复的所有 <code>dbspace</code>
<code>restore_dbfile</code>	可以为空。表示要从备份存档中恢复的给定 <code>dbspace</code> 中的所有 <code>dbfile</code>
<code>backup_comment</code>	用户注释

**示例**

`sp_iqrestoreaction` 的输出样本：

```
sequence_number  backup_id  backup_archive_list  backup_time
14:47:40.0      1          1192                c:\\temp\\b1         2008-09-23
14:48:05.01    2          1201                c:\\temp\\b2.inc     2008-09-23
14:48:13.0     3          1208                c:\\temp\\b3.inc     2008-09-23

virtual_type     restore_dbpace  restore_dbfile  backup_comment
Nonvirtual
Nonvirtual
Nonvirtual
```

## 另请参见

- 确定数据库所使用的安全模型（第 381 页）

**sp\_iqrlvmemory 过程**

按表监控 RLV 存储内存使用情况。

## 语法

```
sp_iqrlvmemory ( [table_name [,table_owner ] ] )
```

## 特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

## 注释

版本特定数据（如版本位图和按需索引）不包含在 RLV 内存使用情况报告中。它们不会影响 RLV 内存限制，因此不在 sp\_iqrlvmemory 中报告。

如果不指定任何参数，将返回有关所有占用内存的 RLV 表的信息。可提供 *table\_name* 和 *table\_owner* 选项，将输出限定为一个表。如果不指定 *table\_owner*，则缺省视为当前用户。

sp\_iqrlvmemory 针对占用 RLV 存储内存的每个表输出一行，输出列为：

列名	描述
table_id	该行所代表的表的 ID。
fragments	此表的存储碎片数。
total	此表使用的 RLV 存储内存总计 (MB)。
data	此表的列碎片所占用的 RLV 存储内存 (MB)。
dictionary	此表的字典所占用的 RLV 存储内存 (MB)。
bitmap	用于存储表级位图的 RLV 存储内存 (MB)。

## 示例

以下示例返回用户 DBA 拥有的表 rlv\_table1 的当前 RLV 内存使用情况。

```
sp_iqrlvmemory 'rlv_table1', 'DBA'
```

输出：

	table_id	fragments	total	data	dictionary	bitmap
1	778	1	1	0	1	1
2	779	1	48	48	0	1

	table_id	fragments	total	data	dictionary	bitmap
3	785	2	1596	1584	0	12

## sp\_iqrowdensity 过程

报告在 FP 索引级的表的内部行碎片的相关信息。

### 语法

```
dbo.sp_iqrowdensity ( 'target' )
```

```
target: (table table-name | (column column-name (...))
```

### 参数

- **table-name** – 报告指定表中的所有列。
- **column-name** – 报告目标表中的指定列。可指定多个目标列，但每次都必须重复使用关键字。

您必须指定关键字 **table** 和 **column**。这些关键字不区分大小写。

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种 系统特权：

- MONITOR
- MANAGE ANY DBSPACE
- CREATE ANY INDEX
- ALTER ANY INDEX
- CREATE ANY OBJECT
- ALTER ANY OBJECT
- 您拥有该表

### 注释

sp\_iqrowdensity 测量缺省索引级别的行碎片。密度是现有表行的索引所需的最少页数与该索引实际使用的页数之比。该过程返回  $0 < density < 1$  数值形式的密度。例如，如果最少需要 8 页存储空间的索引占用了 10 页，则其密度为 0.8。

报告的密度不指示可以通过重新创建或重组缺省索引来释放的磁盘页数。

该过程显示列的行密度的相关信息，但并未建议进一步的操作。您必须决定是否重新创建、重组或重新生成索引。

sp\_iqrowdensity IndexType 列通常返回对列进行编码所需的最多位数。

和先前版本的 FP(1)、FP(2)、FP(3) 字典压缩不同，先前版本对每页均使用相同的位数，而 NBit 会对每页动态编码。sp\_iqrowdensity 通常返回所有页中使用的最多位数。

示例

报告表 *SalesOrders* 中列 *ID* 的行密度:

```
sp_iqrowdensity('column groupo.SalesOrders.ID')
```

Tablename	ColumnName	IndexType	Density
GROUPO.SalesOrders	ID	NBit FP	1.0

另请参见

- `sp_iqindexfragmentation` 过程 (第 452 页)
- `sp_iqrebuildindex` 过程 (第 492 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqsetcompression 过程

设置数据类型为 LONG BINARY (BLOB) 和 LONG VARCHAR (CLOB) 的列中的数据压缩。

语法

```
sp_iqsetcompression ( owner, table, column, on_off_flag )
```

权限

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权:

- ALTER ANY TABLE
- ALTER ANY OBJECT

注释

`sp_iqsetcompression` 用于控制数据类型为 LONG BINARY (BLOB) 和 LONG VARCHAR (CLOB) 的列的压缩。压缩设置仅适用于基表。

`sp_iqsetcompression` 的副作用是更改压缩设置之后将会发生 COMMIT。

表 109. `sp_iqsetcompression` 的参数

名称	描述
<i>owner</i>	要设置压缩的表的所有者
<i>table</i>	要设置压缩的表
<i>column</i>	要设置压缩的列
<i>on_off_flag</i>	压缩设置: ON 表示启用压缩, OFF 表示禁用压缩

*示例*

假定以下表定义:

```
CREATE TABLE USR.pixTable (picID INT NOT NULL,
picJPG LONG BINARY NOT NULL);
```

要禁用对 LOB 列 picJPG 的压缩, 请调用 **sp\_iqsetcompression**:

```
CALL sp_iqsetcompression('USR', 'pixTable', 'picJPG',
'OFF');
```

此命令不返回任何行。

*另请参见*

- 确定数据库所使用的安全模型 (第 381 页)

**sp\_iqsharedtempdistrib 过程**

显示当前共享的临时空间使用情况分布。如果从协调器运行, 则 **sp\_iqsharedtempdistrib** 会显示所有节点的共享临时空间分布。如果从辅助节点运行, 仅显示该节点的共享临时空间使用情况。

**Multiplex** 中为每个节点保留共享临时空间, 可按需使用。为某一节点保留的空间位于某一分配单元。基于动态空间需求, 节点可以保留多个分配单元。节点需要更多空间时, 可租用多个分配单元; 不再需要时, 可将空间归还全局缓冲池。空间使用情况缩减、租用期结束或服务器关闭时, 分配单元将过期。

*语法*

```
sp_iqsharedtempdistrib
```

*适用于*

仅限 **Multiplex**。

*特权*

您必须具有系统过程的 EXECUTE 特权, 以及 MANAGE ANY DBSPACE 系统特权。

*注释*

列	数据类型	描述
ServerID	unsigned bigint	Multiplex 服务器的服务器 ID, 来自 SYSIQMPXINFO。
DBSpaceName	char(128)	保留了空间的 dbspace 的名称。

列	数据类型	描述
UnitType	char(10)	分配单元的类型。有效值包括： <ul style="list-style-type: none"> <li>• 活动 - 当前保留，节点正在使用。</li> <li>• 已到期 - 为节点保留，但正转换回全局空间池。</li> <li>• 隔离 - 为节点保留，但由于节点故障已被隔离。</li> </ul>
VersionID	unsigned bigint	单元的版本 ID。对于活动单元，为单元为节点保留时的版本。对于已到期单元，为单元到期时的版本。对于隔离单元，为单元被隔离时的版本。
NBlocks	unsigned bigint	单元中未完成块的数量。

另请参见

- 确定数据库所使用的安全模型（第 381 页）

### sp\_iqshowcompression 过程

显示数据类型为 LONG BINARY (BLOB) 和 LONG VARCHAR (CLOB) 的列的压缩设置。

语法

```
sp_iqshowcompression ( owner, table, column )
```

特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- ALTER ANY TABLE
- ALTER ANY OBJECT

注释

返回列名和压缩设置。压缩设置值为“ON”（启用压缩）和“OFF”（禁用压缩）。

表 110. sp\_iqshowcompression 的参数

名称	描述
<i>owner</i>	要设置压缩的表的所有者
<i>table</i>	要设置压缩的表
<i>column</i>	要设置压缩的列

示例

假定以下表定义：

```
CREATE TABLE USR.pixTable (picID INT NOT NULL,
picJPG LONG BINARY NOT NULL);
```

要检查 `pixTable` 表中列的压缩状态，请调用 `sp_iqshowcompression`：

```
CALL sp_iqshowcompression('USR', 'pixTable',
'picJPG') ;
```

此命令返回一行：

```
'picJPG', 'ON'
```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqshowpsex 过程

显示数据库选项设置的相关信息，这些设置控制任务优先级和连接的资源使用情况。

### 语法

```
sp_iqshowpsex [ connection-id ]
```

### 参数

- **connection-id** - 表示连接 ID 的整数。

如果指定 `connection-id`，`sp_iqshowpsex` 只返回指定连接相关的信息。如果未指定 `connection-id`，`sp_iqshowpsex` 会返回所有连接的相关信息。

如果指定的 `connection-id` 不存在，`sp_iqshowpsex` 不会返回任何行。

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种 系统特权：

- DROP CONNECTION
- MONITOR
- SERVER OPERATOR

### 注释

`sp_iqshowpsex` 存储过程显示用于控制任务优先级和连接的资源使用情况的数据库选项设置的相关信息，这些信息在数据库管理员进行性能调优时很有用。

表 111. `sp_iqshowpsex` 列

列名	描述
connectionid	连接 ID

列名	描述
应用程序	打开连接的客户端应用程序的相关信息。包括 <b>AppInfo</b> 连接属性信息： <b>HOST</b> : 客户端计算机的主机名； <b>EXE</b> : 客户端可执行文件的名称（仅限 Windows）； <b>APPINFO</b> : 客户端连接字符串中的 APPINFO（如果已指定）。
userid	打开连接的用户的登录名
iqgovern_priority	数据库选项 IQGOVERN_PRIORITY 的值，用于为 -iqgovern 队列中等待的每个查询分配优先级。缺省情况下，该选项的值为 2 (MEDIUM)。值 1、2 和 3 分别显示为 HIGH、MEDIUM 和 LOW。
max_query_time	数据库选项 MAX_QUERY_TIME 的值，用于设置限制，以便优化程序可以禁止过长的查询。缺省情况下，该选项处于禁用状态且值为 0。
query_row_limit	数据库选项 QUERY_ROWS_RETURNED_LIMIT 的值，用于设置行阈值，以便根据估计的结果集大小来拒绝查询。缺省值为 0，表示无限制。
query_temp_space_limit	数据库选项 QUERY_TEMP_SPACE_LIMIT 的值（以 MB 为单位），用于约束用户查询对临时 IQ dbspace 的使用。缺省值为 2,000MB。
max_cursors	数据库选项 MAX_CURSOR_COUNT 的值，用于指定资源调控器，以便限制连接每次可以使用的最大游标数。缺省值为 50。如果值为 0，则表示不受限制。
max_statements	数据库选项 MAX_STATEMENT_COUNT 的值，用于指定资源调控器，以便限制连接每次可以使用的最大预准备语句数。缺省值为 100。如果值为 0，则表示不受限制。

**注意：** **AppInfo** 属性在 Open Client 或 jConnect 应用程序（例如 Interactive SQL）中可能无法使用。如果 **AppInfo** 属性不可用，application 列将为空。

*示例*

显示数据库选项设置的相关信息，这些设置可控制任务优先级和 ID 为 2 的连接的资源使用情况：

```
sp_iqshowpsexec 2

connectionid      application
                2      HOST=GOODGUY-XP;EXE=C:\\Program Files\\Sybase\\
                    IQ-16_0\\bin32\\dbisqlg.exe;

userid      iqgovern_priority      max_query_time      query_row_limit
DBA          MEDIUM                    0                    0

query_temp_space_limit      max_statements      max_cursors
                2000                    50                    100
```



**另请参见**

- CONNECTION\_PROPERTY 函数 [系统] (第 152 页)
- sp\_iqcontext 过程 (第 410 页)
- 确定数据库所使用的安全模型 (第 381 页)

**sp\_iqspaceinfo 过程**

显示当前数据库中每个对象使用的块数和对象所在的 **dbspace** 的名称。

*语法*

```
sp_iqspaceinfo [ 'main
| [table table-name | index index-name] [...] ' ]
```

*适用于*

Simplex 和 Multiplex。

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY DBSPACE 系统特权。

*注释*

针对当前数据库显示对象名、每个对象使用的块数和 **dbspace** 的名称。sp\_iqspaceinfo 不需要任何参数。

sp\_iqspaceinfo 返回的信息对于管理 **dbspace** 很有帮助。

如果在 Multiplex 数据库上运行，缺省参数为返回共享 IQ 存储库大小的 main。

如果不提供任何参数，则必须至少拥有一个用户创建对象（如表）来接收结果。

*示例*

该输出来自 iqdemo 数据库中运行的 sp\_iqspaceinfo 存储过程。本例中删除了一些表和索引的输出。

Name	NBlocks	dbspace_name
Contacts	19	IQ_SYSTEM_MAIN
SalesOrderItems.DBA.ASIQ_IDX_T205_C5_FP	56	IQ_SYSTEM_MAIN
Contacts.DBA.ASIQ_IDX_T206_C10_FP	55	IQ_SYSTEM_MAIN
Contacts.DBA.ASIQ_IDX_T206_C1_FP	61	IQ_SYSTEM_MAIN
...		
Contacts.DBA.ASIQ_IDX_T206_C9_FP	55	IQ_SYSTEM_MAIN
Contacts.DBA.ASIQ_IDX_T206_I11_HG	19	IQ_SYSTEM_MAIN
Customers	20	IQ_SYSTEM_MAIN
Customers.DBA.ASIQ_IDX_T207_C1_FP	61	IQ_SYSTEM_MAIN
Customers.DBA.ASIQ_IDX_T207_C2_FP	55	IQ_SYSTEM_MAIN
...		
Customers.DBA.ASIQ_IDX_T207_I10_HG	19	IQ_SYSTEM_MAIN
...		

**另请参见**

- `sp_iqindexinfo` 过程 (第 453 页)
- `sp_iqdbspace` 过程 (第 419 页)
- `sp_iqdbspaceinfo` 过程 (第 423 页)
- 确定数据库所使用的安全模型 (第 381 页)

## **sp\_iqspaceused 过程**

显示 IQ 存储库、IQ 临时存储库、RLV 存储库以及 IQ 全局和局部共享临时存储库中可用空间和已用空间的相关信息。

*语法*

```
sp_iqspaceused(out mainKB           unsigned bigint,  
               out mainKBUsed       unsigned bigint,  
               out tempKB           unsigned bigint,  
               out tempKBUsed       unsigned bigint,  
               out shTempTotalKB     unsigned bigint,  
               out shTempTotalKBUsed unsigned bigint,  
               out shTempLocalKB     unsigned bigint,  
               out shTempLocalKBUsed unsigned bigint,  
               out rlvLogKB          unsigned bigint,  
               out rlvLogKBUsed      unsigned bigint)
```

*适用于*

Simplex 和 Multiplex。

*特权*

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- ALTER DATABASE
- MANAGE ANY DBSPACE
- MONITOR

*注释*

**sp\_iqspaceused** 以 `unsigned bigint` 输出参数形式返回多个值。该系统存储过程可由用户定义的存储过程调用，以确定正在使用的主存储空间、临时存储空间和 RLV 存储空间的大小。

**sp\_iqspaceused** 返回 **sp\_iqstatus** 提供的信息的部分内容，但允许用户以 SQL 变量返回信息以在计算中使用。

如果在 Multiplex 数据库上运行，该过程将应用于在其上运行该过程的服务器。另外还返回 `IQ_SHARED_TEMP` 的已用空间。

列名	描述
mainKB	IQ 主存储空间的总大小 (KB)。
mainKBUsed	数据库使用的 IQ 主存储空间的大小 (KB)。辅助 Multiplex 节点返回 '(Null)'。
tempKB	IQ 临时存储空间的总大小 (KB)。
tempKBUsed	数据库使用的 IQ 临时存储空间的总大小 (KB)。
shTempTotalKB	IQ 全局共享临时存储空间的总大小 (KB)。
shTempLocalKB	IQ 局部共享临时存储空间的总大小 (KB)。
shTempLocalKBUsed	数据库使用的 IQ 局部共享临时存储空间的大小 (KB)。
rlvLogKB	RLV 存储空间的总大小 (KB)。
rlvLogKBUsed	数据库使用的 RLV 存储空间的大小 (KB)。

### 示例

**sp\_iqspaceused** 需要 7 个输出参数。创建一个用于声明 7 个输出参数的用户定义存储过程 **myspace**，然后调用 **sp\_iqspaceused**：

```

create or replace procedure dbo.myspace()
begin
    declare mt unsigned bigint;
    declare mu unsigned bigint;
    declare tt unsigned bigint;
    declare tu unsigned bigint;
    declare gt unsigned bigint;
    declare gu unsigned bigint;
    declare lt unsigned bigint;
    declare lu unsigned bigint;
    declare tt_t unsigned bigint;
    declare mt_t unsigned bigint;
    declare gt_t unsigned bigint;
    declare lt_t unsigned bigint;
    call sp_iqspaceused(mt,mu,tt,tu,gt,gu,lt,lu);
    if (tt = 0) then
        set tt_t = 0;
    else
        set tt_t = tu*100/tt;
    end if;
    if (mt = 0) then
        set mt_t = 0;
    else
        set mt_t = mu*100/mt;
    end if;
    if (gt = 0) then
        set gt_t = 0;
    else

```

```

        set gt_t = gu*100/gt;
    end if;
    if (lt = 0) then
        set lt_t = 0;
    else
        set lt_t = lu*100/lt;
    end if;
    select cast(mt/1024 as unsigned bigint) as mainMB,
           cast(mu/1024 as unsigned bigint) as mainusedMB, mt_t as
mainPerCent,
           cast(tt/1024 as unsigned bigint) as tempMB,
           cast(tu/1024 as unsigned bigint) as tempusedMB, tt_t as
tempPerCent,
           cast(gt/1024 as unsigned bigint) as shTempTotalKB,
           cast(gu/1024 as unsigned bigint) as shTempTotalKBUsed, gt_t
as globalshTempPerCent,
           cast(lt/1024 as unsigned bigint) as shTempLocalMB,
           cast(lu/1024 as unsigned bigint) as shTempLocalKBUsed, lt_t
as localshTempPerCent;
end

```

要显示 **sp\_iqspaceused** 的输出，请执行 **myspace**：

```
myspace
```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqstatistics 过程

返回每项可用统计或指定统计的序列号、名称、说明、值和单位指示符。

### 语法

```
sp_iqstatistics [ stat_name ]
```

### 参数

- **stat\_name** - （可选）VARCHAR 参数指定统计的名称。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY STATISTICS 系统特权。

### 注释

如果提供 **stat\_name**，**sp\_iqstatistics** 将为给定统计返回一行，如果名称无效，则不返回行。在不指定任何参数的情况下调用时，**sp\_iqstatistics** 将返回所有统计。

## 结果集

列名	数据类型	描述
stat_num	UNSIGNED INTEGER	统计的序列号
stat_name	VARCHAR(255)	统计的名称
stat_desc	VARCHAR(255)	统计的说明
stat_value	LONG VARCHAR	统计的值
stat_unit	VARCHAR(128)	单位指示符

可能返回以下统计：

stat_num	stat_name	stat_desc	stat_unit
0	CpuTotalTime	自上次服务器启动以来 SAP Sybase IQ 服务器使用的总 CPU 时间（以秒为单位）	Second
1	CpuUserTime	自上次服务器启动以来 SAP Sybase IQ 服务器使用的 CPU 用户时间（以秒为单位）	Second
2	CpuSystemTime	自上次服务器启动以来 SAP Sybase IQ 服务器使用的 CPU 系统时间（以秒为单位）	Second
3	ThreadsFree	SAP Sybase IQ 的可用线程数	N/A
4	ThreadsInUse	SAP Sybase IQ 正在使用的线程数	N/A
5	MemoryAllocated	分配的内存（以 MB 为单位）	MB
6	MemoryMaxAllocated	分配的最大内存（以 MB 为单位）	MB
7	MainCacheCurrentSize	主高速缓存的当前大小（以 MB 为单位）	MB
8	MainCacheFinds	主高速缓存查找请求总数	N/A
9	MainCacheHits	主高速缓存总命中次数	N/A
10	MainCachePagesPinned	已占用的主高速缓存页数	Page
11	MainCachePagesPinnedPercentage	已占用的主高速缓存页百分比	%

stat_num	stat_name	stat_desc	stat_unit
12	MainCachePagesDirtyPercentage	主高速缓存脏页百分比	%
13	MainCachePagesInUsePercentage	使用中的主高速缓存页百分比	%
14	TempCacheCurrentSize	临时高速缓存的当前大小 (以 MB 为单位)	MB
15	TempCacheFinds	临时高速缓存查寻请求总数	N/A
16	TempCacheHits	临时高速缓存总命中次数	N/A
17	TempCachePagesPinned	已占用的临时高速缓存页数	Page
18	TempCachePagesPinnedPercentage	已占用的临时高速缓存页百分比	%
19	TempCachePagesDirtyPercentage	临时高速缓存脏页百分比	%
20	TempCachePagesInUsePercentage	使用中的临时高速缓存页百分比	%
21	MainStoreDiskReads	从主存储库中读取到的空间大小 (KB)	KB
22	MainStoreDiskWrites	写入主存储库的空间大小 (KB)	KB
23	TempStoreDiskReads	从主存储库中读取到的空间大小 (KB)	KB
24	TempStoreDiskWrites	写入主存储库的空间大小 (KB)	KB
25	ConnectionsTotalConnections	服务器启动后的连接总数	N/A
26	ConnectionsTotalDisconnections	服务器启动后断开连接的总数	N/A
27	ConnectionsActive	活动连接数	N/A
28	OperationsWaiting	等待 SAP Sybase IQ 资源调控器的操作数	N/A
29	OperationsActive	SAP Sybase IQ 资源调控器允许的活动并发操作数	N/A
30	OperationsActiveLoadTableStatements	活动 LOAD TABLE 语句数	N/A

**示例**

显示单个统计的总 CPU 时间：

```
sp_iqstatistics 'CpuTotalTime'
```

显示 MainCache% 的所有统计：

```
SELECT * from sp_iqstatistics() WHERE stat_name LIKE 'MainCache%'
```

stat_num	stat_name	stat_desc	stat_value	stat_unit
7	MainCacheCurrentSize	高速缓存的当前大小（以 MB 为单位）	64	mb
8	MainCacheFinds	高速缓存 dbspace 查寻请求总数	95303	
9	MainCacheHits	高速缓存 dbspace 总命中次数	95283	
10	MainCachePagesPinned	已占用的高速缓存 dbspace 页数	0	page
11	MainCachePagesPinned-Percentage	已占用的高速缓存 dbspace 页的百分比	0	%
12	MainCachePagesDirty-Percentage	高速缓存 dbspace 脏页百分比	0.39	%
13	MainCachePagesInUse-Percentage	使用中的高速缓存 dbspace 页百分比	4.44	%

**另请参见**

- 确定数据库所使用的安全模型（第 381 页）

**sp\_iqstatus 过程**

显示有关当前数据库的各种 SAP Sybase IQ 状态信息。

**语法**

```
sp_iqstatus
```

**适用于**

Simplex 和 Multiplex。

**特权**

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- ALTER DATABASE
- MANAGE ANY DBSPACE
- MONITOR
- SERVER OPERATOR

*注释*

显示有关当前数据库的状态信息，包括数据库名称、创建日期、页大小、dbspace 的段数、块使用情况、缓冲区使用情况、I/O、备份信息等。

**sp\_iqstatus** 显示主存储库和临时存储库的空间不足状态。如果存储库出现空间不足状态，**sp\_iqstatus** 会在该存储库的空间不足状态显示值中显示 Y。

可以使用 **sp\_iqstatus** 监控行级版本控制 (RLV) 存储库使用的内存。**RLV memory limit** 行显示了 `-iqrlymem` 服务器选项或 `sa_server_option rlv_memory_mb` 指定的内存限制。RLV memory used 行可显示 RLV 存储库使用的内存量。

可以使用 **sp\_iqstatus** 监控高速缓存 dbspace 中直连存储设备使用的内存：

测量	描述
高速缓存 Dbspace 文件数	数据库中高速缓存 dbspace dbfile 的数量。
高速缓存 Dbspace 块	标识高速缓存 dbspace 块和相应的存储设备 dbfile 名称。
已用高速缓存 Dbspace IQ 块	已用 IQ 块数与 IQ 块总数进行比较。也可以按百分比显示使用情况。如果此百分比偏高，则考虑添加额外存储。

**sp\_iqspaceused** 返回与 **sp\_iqstatus** 提供的信息相同的部分内容，但允许用户以 SQL 变量返回信息以在计算中使用。

要显示可通过删除连接回收的空间，请使用 **sp\_iqstatus** 并将返回的两行的结果相加：

```
(DBA)> select * from sp_iqstatus() where name like '%Versions:%'
Execution time: 6.25 seconds
Name          Value
-----
Other Versions: 2 = 1968Mb
Active Txn Versions: 1 = C:2175Mb/D:2850Mb

(First 2 rows)
```

上面的输出样本显示一个活动写入事务已创建 2175MB 的数据，并销毁 2850MB 的数据。事务中使用的且尚未释放的总数据为 4818MB，或 1968MB + 2850MB = 4818MB。

**sp\_iqstatus** 忽略将在下一个检查点释放的块。但是，这些块将以 X 类型出现在 **sp\_iqdbspace** 输出中。

在 Multiplex 中，该过程还列出有关共享 IQ 存储库和 IQ 临时存储库的信息。如果 **sp\_iqstatus** 显示 Multiplex 服务器上正在使用的主块的百分比较高，请运行 **sp\_iqversionuse** 查明正在使用哪些版本以及释放这些版本可恢复的空间大小。



## 示例

**注意：**此示例包括一个名为 `iq_main` 的用户 `dbspace` 示例，您自己的数据库中可能不存在该示例。

以下输出来自 `sp_iqstatus` 存储过程：

```
Sybase IQ (TM) Copyright (c) 1992-2013 by SAP AG or an SAP affiliate
company. All rights reserved.
Version: 16.0.0.562/130821/P/Mainline/Sun_x64/OS 5.10/64bit/
2013-08-21 06:15:41
Time Now: 2013-08-21 06:27:14.150
Build Time: 2013-08-21 06:15:41
File Format: 23 on 03/18/1999
Server mode: IQ Server
Catalog Format: 2
Stored Procedure Revision: 1
Page Size: 65536/4096blks/16bpp
Number of Main DB Files: 2
Main Store Out Of Space: N
Number of Cache Dbspace Files: 5
Number of Shared Temp DB Files: 0
Shared Temp Store Out Of Space: N
Number of Local Temp DB Files: 1
Local Temp Store Out Of Space: N
DB Blocks: 1-25600 IQ_SYSTEM_MAIN
DB Blocks: 522208-547807 MainUser
Cache Dbspace Blocks: 1-5120 ssd_dev_1
Cache Dbspace Blocks: 522208-527327 ssd_dev_2
Cache Dbspace Blocks: 1044416-1049535 ssd_dev_3
Cache Dbspace Blocks: 1566624-1571743 ssd_dev_4
Cache Dbspace Blocks: 2088832-2093951 ssd_dev_5
Local Temp Blocks: 1-25600 IQ_SYSTEM_TEMP
Create Time: 2013-08-21 06:27:05.444
Update Time: 2013-08-21 06:27:14.035
Main IQ Buffers: 1588, 100Mb
Temporary IQ Buffers: 1588, 100Mb
Main IQ Blocks Used: 5250 of 38400, 13%=20Mb, Max Block#: 5313
Cache Dbspace IQ Blocks Used: 197 of 25600, 0%=0Mb, Max Block#: 0
Shared Temporary IQ Blocks Used: 0 of 0, 0%=0Mb, Max Block#: 0
Local Temporary IQ Blocks Used: 65 of 12800, 0%=0Mb, Max Block#: 0
Main Reserved Blocks Available: 12800 of 12800, 100%=50Mb
Shared Temporary Reserved Blocks Available: 0 of 0, 0%=0Mb
Local Temporary Reserved Blocks Available: 12800 of 12800, 100%=50Mb
IQ Dynamic Memory: Current: 292mb, Max: 308mb
Main IQ Buffers: Used: 18, Locked: 0
Temporary IQ Buffers: Used: 4, Locked: 0
Main IQ I/O: I: L459/P9 O: C21/D33/P22 D:1 C:100.0
Temporary IQ I/O: I: L320/P0 O: C54/D59/P8 D:50 C:100.0
Other Versions: 0 = 0Mb
Active Txn Versions: 0 = C:0Mb/D:0Mb
Last Full Backup ID: 0
Last Full Backup Time:
Last Backup ID: 0
```

```
Last Backup Type: None
Last Backup Time:
DB Updated: 0
Blocks in next ISF Backup: 0 Blocks: =0Mb
Blocks in next ISI Backup: 0 Blocks: =0Mb
IQ large memory space: 2048Mb
IQ large memory flexible percentage: 50
IQ large memory flexible used: 0Mb
IQ large memory inflexible percentage: 90
IQ large memory inflexible used: 0Mb
IQ large memory anti-starvation percentage: 50
DB File Encryption Status: OFF
RLV memory limit: 2048Mb
RLV memory used: 0Mb
```

下面是了解 Main IQ I/O 和 Temporary IQ I/O 输出代码的关键:

- I: 输入
- L: 读取的逻辑页 (“查找”)
- P: 读取的物理页
- O: 输出
- C: 创建的页
- D: 脏页
- P: 物理写入
- D: 损坏的页
- C: 压缩率

#### 另请参见

- `sp_iqtransaction` 过程 (第 545 页)
- `sp_iqversionuse` 过程 (第 552 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqsysmon 过程

监控 SAP Sybase IQ 的多个组件, 其中包括管理缓冲区高速缓存、内存、线程、锁、I/O 功能和 CPU 利用率。

#### 批处理模式语法

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor [, 'section(s)']
or
sp_iqsysmon 'time-period' [, 'section(s)']
```

#### 文件模式语法

```
sp_iqsysmon start_monitor, 'filemode' [, 'monitor-options']
sp_iqsysmon stop_monitor
```

### 批处理模式参数

- **start\_monitor** - 开始监控。
- **stop\_monitor** - 停止监控并显示报告。
- **time-period** - 监控时间段，采用 HH:MM:SS 格式。
- **section(s)** - 要由 **sp\_iqsysmon** 显示的一个或多个部分的缩写。

请参见“注释（第 0 页）”部分了解有关缩写的完整列表的信息。

如果指定多个部分，请使用空格分隔各个部分缩写，并将列表括在单引号或双引号中。缺省情况下显示所有部分。

对于与 IQ 主存储库相关的部分，可通过分别在部分缩写前面添加前缀 'm' 或 't' 来指定主存储库或临时存储库。如果不添加前缀，则会监控这两个存储库。例如，如果指定 'mbufman'，则仅监控 IQ 主存储库缓冲区管理器。如果指定 'mbufman tbufman' 或 'bufman'，则会同时监控主存储库缓冲区管理器和临时存储库缓冲区管理器。

---

**注意：** **sp\_iqsysmon** 当前不支持 SAP Sybase IQ 组件磁盘 I/O 和锁管理器。

---

### 文件模式参数

- **start\_monitor** - 开始监控。
- **stop\_monitor** - 停止监控并将剩余输出写入日志文件。
- **filemode** - 指定 **sp\_iqsysmon** 在文件模式下运行。在文件模式下，将在监控周期的每个间隔显示统计信息样本。缺省情况下，输出写入到名为 *dbname.connid-icqmon* 的日志文件中。使用 **file\_suffix** 选项可更改输出文件的后缀。有关 **file\_suffix** 选项的说明，请参见 *monitor\_options* 参数。
- **monitor\_options** - *monitor\_options* 字符串可以包含一个或多个选项：
  - **-interval seconds** - 指定报告间隔（秒）。每一间隔后，将监控器示例统计信息输出到日志文件中。如果未指定 **-interval** 选项，则缺省设置为每 60 秒输出一次。最小报告间隔为 2 秒。如果为该选项指定的间隔无效或小于 2 秒，则会将间隔设置为 2 秒。

初次显示时，显示自服务器启动以来的计数器信息。后续显示则显示与先前显示的不同之处。在运行有关性能问题的查询期间或在通常会出现性能问题的那一天中的某个时间，以 60 秒的缺省间隔运行监控器通常可以获得有用的结果。非常短的间隔可能不会提供有意义的结果。间隔应与作业时间成比例；60 秒通常已足够。

- **-file\_suffix suffix** - 创建一个名为 *dbname.connid-suffix* 的监控器输出文件。如果未指定 **-file\_suffix** 选项，则后缀缺省为 *icqmon*。如果指定 **-file\_suffix** 选项，但是未提供后缀或提供空字符串作为后缀，则不使用后缀。

- **-append 或 -truncate** - 分别指示 **sp\_iqsysmon** 向现有输出文件附加内容或截断现有输出文件。截断是缺省设置。如果同时指定这两个选项，则在字符串中较晚指定的那个选项有效。
- **-section section(s)** - 指定要写入监控器日志文件的一个或多个部分的缩写。

请参见“注释（第 0 页）”部分了解有关缩写的完整列表的信息。

缺省情况下写入所有部分。文件模式下在部分列表中指定的缩写与批处理模式下使用的缩写相同。如果指定多个部分，则必须以空格分隔各个部分的缩写。

如果指定 **-section** 选项时没有指定任何部分，则不会对任何部分进行监控。无效部分缩写将被忽略，并向 IQ 消息文件中写入一条警告。

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

*注释*

要报告的报告部分或 IQ 组件	要输入的缩写
缓冲区分配	(主) - mbufalloc (临时) - tbufalloc
缓冲区管理器	(主) - mbufman (临时) - tbufman
缓冲池	(主) - mbufpool (临时) - tbufpool
目录统计信息	目录
CPU 利用率	cpu
空闲列表管理	(主) - mfreelist (临时) - tfreelist
内存管理	memory
预取管理	(主) - mprefetch (临时) - tprefetch
IQ RLV 内存存储库统计信息	rlv
大容量内存分配器 (LMA) 统计信息	lma
服务器上下文统计信息	server
线程管理	threads

要报告的报告部分或 IQ 组件	要输入的缩写
事务管理	txn

**sp\_iqsysmon** 存储过程监控 SAP Sybase IQ 的多个组件，其中包括管理缓冲区高速缓存、内存、线程、锁、I/O 功能和 CPU 利用率。

**sp\_iqsysmon** 过程支持两种监控模式：

- **批处理模式** - **sp\_iqsysmon** 收集在监控器启动到停止期间或 *time-period* 参数指定的期间内监控器的统计信息。监控期间结束时，**sp\_iqsysmon** 显示合并统计信息的列表。

批处理模式下的 **sp\_iqsysmon** 与 SAP Adaptive Server® Enterprise 过程 **sp\_sysmon** 相似。

- **文件模式** - **sp\_iqsysmon** 将监控器启动到停止期间的每个时间间隔的示例统计信息写入日志文件。

在文件模式下初次显示时，将显示自服务器启动以来的计数器信息。后续显示则显示与先前显示的不同之处。

文件模式下的 **sp\_iqsysmon** 与 IQ UTILITIES 命令 **START MONITOR** 和 **STOP MONITOR** 接口相似。

#### 批处理模式语法示例

示例 1:

在批处理模式下启动监控器，并显示主存储库和临时存储库的所有部分：

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor
```

示例 2:

在批处理模式下启动监控器，并显示主存储库的缓冲区管理器和缓冲池统计信息。

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor 'mbufman mbufpool'
```

示例 3:

在 10 分钟后输出监控信息：

```
sp_iqsysmon '00:10:00'
```

示例 4:

5 分钟之后，仅输出 **sp\_iqsysmon** 报告的内存管理器部分：

```
sp_iqsysmon '00:05:00', memory
```

示例 5:

启动监控器，执行两个过程和一个查询，停止监控器，然后仅输出报告的缓冲区管理器部分：

```
sp_iqsysmon start_monitor
  go
  execute proc1
  go
  execute proc2
  go
  select sum(total_sales) from titles
  go
  sp_iqsysmon stop_monitor, bufman
  go
```

示例 6:

2 分钟后仅输出报告的主缓冲区管理器和主缓冲池部分：

```
sp_iqsysmon '00:02:00', 'mbufman mbufpool'
```

示例 7:

1 小时后仅输出报告的 **RLV** 部分：

```
sp_iqsysmon '01:00:00', 'rlv'
```

示例 8:

5 秒后仅输出报告的 **LMA** 部分：

```
sp_iqsysmon '00:00:05', 'lma'
```

示例 9:

在批处理模式下运行监控器 10 秒钟，并在该时间段结束时显示合并统计信息：

```
sp_iqsysmon '00:00:10', 'mbufpool memory'
```

#### 文件模式语法示例

示例 1:

在监控器启动到停止期间，每 2 秒截断信息一次并将其写入日志文件中：

```
sp_iqsysmon start_monitor, 'filemode', '-interval 2'
.
.
.
  sp_iqsysmon stop_monitor
```

示例 2:

仅将主缓冲区管理器和内存管理器部分的输出附加到名为 dbname.connid-testmon 的 **ASCII** 文件。对于数据库 iqdemo，将结果写入文件 iqdemo.2-testmon 中：

```
sp_iqsysmon start_monitor, 'filemode',
  '-file_suffix testmon -append -section mbufman memory'
.
```

```

.
.
sp_iqsysmon stop_monitor

```

示例 3:

仅输出报告的 RLV 和 LMA 部分:

```

sp_iqsysmon start_monitor,'filemode','-section rlv lma'
sp_iqsysmon stop_monitor

```

示例 4:

在文件模式下启动监控器，并将主缓冲池和内存管理器的统计信息写入日志文件（每 5 秒写入一次）：

```

sp_iqsysmon start_monitor, 'filemode', '-interval 5 -section
mbufpool memory'
sp_iqsysmon stop_monitor

```

**另请参见**

- 确定数据库所使用的安全模型（第 381 页）

### **sp\_iqsysmon 过程示例**

sp\_iqsysmon 输出示例。

示例 1:

在 20 分钟后显示缓冲区分配（主存储和临时存储）的输出。

```

sp_iqsysmon '00:20:00', 'mbufalloc tbufalloc'

```

```

=====
Buffer Allocator (Main)"
=====

STATS-NAME                VALUE
NActiveCommands           2
BufAllocMaxBufs           2275 ( 81.6% )
BufAllocAvailBufs         2115 ( 93.0% )
BufAllocReserved          160 ( 7.0% )
BufAllocAvailPF           750 ( 33.0% )
BufAllocSlots             100
BufAllocNPinUsers         0
BufAllocNPFUsers          2
BufAllocNPostedUsrs       0
BufAllocNUnpostUsrs       0
BufAllocPinQuota          0
BufAllocNPostEst          0
BufAllocNUnPostEst        0
BufAllocMutexLocks        0
BufAllocMutexWaits        0 ( 0.0% )

STATS-NAME                VALUE
NActiveCommands           2
BufAllocMaxBufs           2275 ( 81.6% )

```

# 系统过程

```

BufAllocAvailBufs      2115 ( 93.0% )
BufAllocReserved       160 ( 7.0% )
BufAllocAvailPF        750 ( 33.0% )
BufAllocSlots          100
BufAllocNPinUsers      0
BufAllocNPFUsers       2
BufAllocNPostedUsrs    0
BufAllocNUnpostUsrs    0
BufAllocPinQuota       0
BufAllocNPostEst       0
BufAllocNUnPostEst     0
BufAllocMutexLocks     0
BufAllocMutexWaits     0 ( 0.0% )
    
```

STATS-NAME	FP	GARRAY	TOTAL	UNKNWN	HASH	CSORT	ROW	
ROWCOL			LOB	BTREE	BM	BV	STORE	TEST
NumClients			2	0	0		2	
0	0	0	0	0	0	0	0	0
PinUserQuota			0	0	0		0	
0	0	0	0	0	0	0	0	0
PrefetchUserQuota			160	0	0		160	
0	0	0	0	0	0	0	0	0
PinUserRegisters			2	2	0		0	
0	0	0	0	0	0	0	0	0
PfUserRegisters			4697	0	0		382	
2621	377	182	0	2	0	0	0	0
ClientCountOfPinner			0	1	3		10	
33	66	100	333	666	1000	3333	6666	10000
Unknown			0	0	0		0	
0	0	0	0	0	0	0	0	0
Hash			0	0	0		0	
0	0	0	0	0	0	0	0	0
Sort			0	0	0		0	
0	0	0	0	0	0	0	0	0
Row			2	0	0		0	
0	0	0	0	0	0	0	0	0
RowColumn			0	0	0		0	
0	0	0	0	0	0	0	0	0
FP			0	0	0		0	
0	0	0	0	0	0	0	0	0
Garray			0	0	0		0	
0	0	0	0	0	0	0	0	0
LOB			0	0	0		0	
0	0	0	0	0	0	0	0	0
BTree			0	0	0		0	
0	0	0	0	0	0	0	0	0
BM			0	0	0		0	
0	0	0	0	0	0	0	0	0
BV			0	0	0		0	
0	0	0	0	0	0	0	0	0
Store			0	0	0		0	
0	0	0	0	0	0	0	0	0
Test			0	0	0		0	
0	0	0	0	0	0	0	0	0



DBCC				0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
Unknown				0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
Unknown				0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
Run				0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
QCPRun				0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
TextDoc				0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
Unknown				0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
Unknown				0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
VDO				0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
Load				Pass	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
STATS-NAME (cont'd)				DBCC	BLKMAP	IQUTIL					
NumClients				0	0	0			0	0	
0	0	0	0	0	0	0			0	0	
PinUserQuota				0	0	0			0	0	
0	0	0	0	0	0	0			0	0	
PrefetchUserQuota				0	0	0			0	0	
0	0	0	0	0	0	0			0	0	
PinUserRegisters				0	0	0			0	0	
0	0	0	0	0	0	0			0	0	
PfUserRegisters				0	0	0			0	0	
0	0	0	1133	0	0	0			0	0	
ClientCountOfPinners				33333	66666	100000			4294967295		
Unknown				0	0	0			0	0	
Hash				0	0	0			0	0	
Sort				0	0	0			0	0	
Row				0	0	0			0	0	
RowColumn				0	0	0			0	0	
FP				0	0	0			0	0	
Garray				0	0	0			0	0	
LOB				0	0	0			0	0	
BTree				0	0	0			0	0	
BM				0	0	0			0	0	
BV				0	0	0			0	0	
Store				0	0	0			0	0	
Test				0	0	0			0	0	
DBCC				0	0	0			0	0	
Unknown				0	0	0			0	0	
Unknown				0	0	0			0	0	
Run				0	0	0			0	0	
QCPRun				0	0	0			0	0	
TextDoc				0	0	0			0	0	
Unknown				0	0	0			0	0	
Unknown				0	0	0			0	0	
VDO				0	0	0			0	0	

# 系统过程

Load	0	0	0	0	0	0	0	0	
=====									
Buffer Allocator (Temporary)									
=====									
STATS-NAME	VALUE								
NActiveCommands	2								
BufAllocMaxBufs	2275 ( 81.6% )								
BufAllocAvailBufs	2263 ( 99.5% )								
BufAllocReserved	12 ( 0.5% )								
BufAllocAvailPF	908 ( 39.9% )								
BufAllocSlots	100								
BufAllocNPinUsers	2								
BufAllocNPFUsers	2								
BufAllocNPostedUsrs	0								
BufAllocNUnpostUsrs	0								
BufAllocPinQuota	175								
BufAllocNPostEst	2								
BufAllocNUnPostEst	2								
BufAllocMutexLocks	0								
BufAllocMutexWaits	0 ( 0.0% )								
STATS-NAME	FP		GARRAY	TOTAL	UNKNWN	HASH	CSORT	ROW	
ROWCOL	FP	GARRAY	LOB	BTREE	BM	BV	STORE	TEST	
NumClients	0	0	4	0	0	0	4	0	
0	0	0	0	0	0	0	0	0	
PinUserQuota	0	0	10	0	0	0	10	0	
0	0	0	0	0	0	0	0	0	
PrefetchUserQuota	0	0	2	0	0	0	2	0	
0	0	0	0	0	0	0	0	0	
PinUserRegisters	0	0	668	0	300	0	247	0	
0	0	0	0	0	0	0	0	0	
PfUserRegisters	0	0	675	0	0	0	295	0	
0	0	0	0	0	0	0	1	0	
ClientCountOfPinner	33	66	100	333	666	1000	3333	6666	10000
Unknown	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Hash	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Sort	0	0	2	0	1	0	0	1	0
0	0	0	0	0	0	0	0	0	0
Row	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
RowColumn	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
FP	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Garray	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
LOB	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
BTree	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	
BM				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
BV				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
Store				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
Test				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
DBCC				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
Unknown				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
Unknown				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
Run				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
QCPRun				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
TextDoc				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
Unknown				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
Unknown				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
VDO				0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
Load			Pass		2	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	
STATS-NAME (cont'd)				DBCC	BLKMAP	IQUTIL						
NumClients				0	0	0	0	0				
0	0	0	0	0	0	0	0	0	0			
PinUserQuota				0	0	0	0	0				
0	0	0	0	0	0	0	0	0	0			
PrefetchUserQuota				0	0	0	0	0				
0	0	0	0	0	0	0	0	0	0			
PinUserRegisters				0	0	0	0	110	2			
0	0	0	0	9	0	0	0	0	0			
PfUserRegisters				0	0	0	0	378	0			
0	0	1	0	0	0	0	0	0	0			
ClientCountOfPinner				33333	66666	100000	4294967295					
Unknown				0	0	0	0					
Hash				0	0	0	0					
Sort				0	0	0	0					
Row				0	0	0	0					
RowColumn				0	0	0	0					
FP				0	0	0	0					
Garray				0	0	0	0					
LOB				0	0	0	0					
BTree				0	0	0	0					
BM				0	0	0	0					
BV				0	0	0	0					
Store				0	0	0	0					
Test				0	0	0	0					

## 系统过程

DBCC	0	0	0	0	0
Unknown	0	0	0	0	0
Unknown	0	0	0	0	0
Run	0	0	0	0	0
QCPRun	0	0	0	0	0
TextDoc	0	0	0	0	0
Unknown	0	0	0	0	0
Unknown	0	0	0	0	0
VDO	0	0	0	0	0
Load	0	0	0	0	0
0					

### 示例 2:

在 20 分钟后显示缓冲区管理器（主存储和临时存储）的输出。

```
sp_iqsysmon '00:20:00', 'mbufman tbufman'
```

```
=====
Buffer Manager (Main)
=====
```

STATS-NAME				TOTAL	NONE	TXTPOS	TXTDOC	CMPACT
BTREEV	BTREEF	BV	VDO	DBEXT	DBID	SORT	STORE	GARRAY
Finds			80137	0	0	0	0	9046
3307	0	20829	0	0	0	0	275	
Hits			80090	0	0	0	0	9015
3291	0	20829	0	0	0	0	275	
Hit%			99.9	0	0	0	0	99.7
99.5	0	100	0	0	0	0	100	
FalseMiss			26469	0	0	0	0	0
63	40	0	1097	0	0	0	0	0
UnOwnRR			48	0	0	0	0	31
16	0	1	0	0	0	0	0	
Cloned			0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
Creates			1557	0	0	0	0	60
179	0	256	0	0	0	0	58	
Destroys			546	0	0	0	0	0
12	21	0	6	0	0	0	0	29
Dirtyes			7554	0	0	0	0	1578
585	0	0	0	0	0	0	0	
RealDirtyes			2254	0	0	0	0	0
117	180	0	542	0	0	0	0	58
PrefetchReqs			80	0	0	0	0	0
0	0	74	0	0	0	0	0	0
PrefetchNotInMem			1	0	0	0	0	0
0	0	0	1	0	0	0	0	0
PrefetchInMem			1466	0	0	0	0	0
0	0	0	1466	0	0	0	0	0
Reads			48	0	0	0	0	31
16	0	1	0	0	0	0	0	
PReadBlks			114	0	0	0	0	0
80	32	0	2	0	0	0	0	0
PReadKB			0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0
ReReads				0		0		0	0	0
0	0	0	0	0	0	0	0	0	0	0
Writes				2002		0		0		104
163	0	538		0		0		0		29
PWriteBlks				6506		0		0		210
326	0	1115		0		0		0		58
PWriteKB				0		0		0		0
0	0	0	0	0	0	0	0	0	0	0
GrabbedDirty				0		0		0		0
0	0	0	0	0	0	0	0	0	0	0
ReadRemoteRpc				0		0		0		0
0	0	0	0	0	0	0	0	0	0	0
ReadRemotePhyIO				0		0		0		0
0	0	0	0	0	0	0	0	0	0	0
STATS-NAME (cont'd)			BARRAY	BLKMAP	HASH	CKPT	BM			
TEST	CMID	RIDCA	LOB	LVCRID	FILE	RIDMAP	RVLOG			
Finds			2681	8329	0	0	35670			
0	0	0	0	0	0	0	0			
Hits			2681	8329	0	0	35670			
0	0	0	0	0	0	0	0			
Hit%			100	100	0	0	100			
0	0	0	0	0	0	0	0			
FalseMiss			84	8329	0	0	16856			
0	0	0	0	0	0	0	0			
UnOwnRR			0	0	0	0	0			
0	0	0	0	0	0	0	0			
Cloned			0	0	0	0	0			
0	0	0	0	0	0	0	0			
Creates			108	358	0	0	538			
0	0	0	0	0	0	0	0			
Destroys			0	126	0	0	59			
0	0	0	0	0	0	0	0			
Dirtyies			512	235	0	0	4644			
0	0	0	0	0	0	0	0			
RealDirtyies			128	593	0	0	636			
0	0	0	0	0	0	0	0			
PrefetchReqs			6	0	0	0	0			
0	0	0	0	0	0	0	0			
PrefetchNotInMem			0	0	0	0	0			
0	0	0	0	0	0	0	0			
PrefetchInMem			0	0	0	0	0			
0	0	0	0	0	0	0	0			
Reads			0	0	0	0	0			
0	0	0	0	0	0	0	0			
PReadBlks			0	0	0	0	0			
0	0	0	0	0	0	0	0			
PReadKB			0	0	0	0	0			
0	0	0	0	0	0	0	0			
ReReads			0	0	0	0	0			
0	0	0	0	0	0	0	0			
Writes			128	466	0	0	574			
0	0	0	0	0	0	0	0			
PWriteBlks			239	3728	0	0	830			
0	0	0	0	0	0	0	0			

# 系统过程

PWriteKB	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
GrabbedDirty	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
ReadRemoteRpc	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
ReadRemotePhyIO	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
STATS-NAME VALUE									
BusyWaits	98								
LRUNumLocks	401784								
LRUNumSpinsWoTO	0 0%								
LRUNumSpinLoops	4315								
LRUNumTimeOuts	4315 -1.10%								
BmapHTNumLocks	0								
BmapHTNumWaits	0 0%								
CacheTeamTimesWoken	182								
CacheTeamNumAsleep	10								
BmapHTMaxEntries	4096								
BmapHTNEntries	27								
BmapHTNInserts	31954								
BmapHTNCollisn	203								
BmapHTNFinds	51419								
BmapHTNHits	19576								
BmapHTNHits1	19550								
BmapHTNHits2	26								
BmapHTNClears	31933								
BmapHTNLChain	1								
BmapHTNRehash	0								
BlockmapMutexsNLocks	0								
BlockmapMutexsNWaits	0								
BlockmapUID	3659								
BlockmapUIDnallocs	3652								
BlockmapRegEver	31851								
BlockmapRegisters	31844								
BufHTNBuckets	4608								
BufHTNEntries	1208								
BufHTNw2orMore	158								
BufHTMaxBucketSize	19								
BufHTNFoiledOps	0								
IONumLocks	0								
IONumWaits	0 0%								
=====									
Buffer Manager (Temporary)									
=====									
STATS-NAME TOTAL NONE TXTPOS TXTDOC CMPACT									
BTREEV	BTREEF	BV	VDO	DBEXT	DBID	SORT	STORE	GARRAY	
Finds			31656	0	0	0	0	0	
0	0	0	0	0	1022	0	0	0	
Hits			31655	0	0	0	0	0	
0	0	0	0	0	1022	0	0	0	
Hit%			100	0	0	0	0	0	
0	0	0	0	0	100	0	0	0	

FalseMiss				23898	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
UnOwnRR				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Cloned				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Creates				5682	0	0	0	0	0	0
0	0	0	0	0	0	1048	716	0	0	0
Destroys				5670	0	0	0	0	0	0
0	0	0	0	0	0	821	17	0	0	0
Dirtyes				6702	0	0	0	0	0	0
0	0	0	0	0	0	379	0	0	0	0
RealDirtyes				5692	0	0	0	0	0	0
0	0	0	0	0	0	1048	716	0	0	0
PrefetchReqs				1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PrefetchNotInMem				1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PrefetchInMem				446	0	0	0	0	0	0
0	0	0	0	0	0	446	0	0	0	0
Reads				2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PReadBlks				4096	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PReadKB				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
ReReads				2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Writes				10	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PWriteBlks				80	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PWriteKB				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
GrabbedDirty				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
ReadRemoteRpc				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
ReadRemotePhyIO				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
STATS-NAME (cont'd)				BARRAY	BLKMAP	HASH	CKPT	BM		
TEST	CMID	RIDCA	LOB	LVCRID	FILE	RIDMAP	RVLOG			
Finds				0	8569	124	0	21939		
0	0	0	0	0	2	0	0	0		
Hits				0	8569	124	0	21939		
0	0	0	0	0	1	0	0	0		
Hit%				0	100	100	0	100		
0	0	0	0	0	50	0	0	0		
FalseMiss				0	8569	0	0	15328		
0	0	0	0	0	1	0	0	0		
UnOwnRR				0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
Cloned				0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
Creates				0	1440	777	0	1041		

# 系统过程

0	0	0	0	0	0	660	0	0
Destroys	0	0	0	0	1434	777	0	123
0	0	0	0	0	0	660	0	0
Dirtyies	0	0	0	0	0	0	0	6323
0	0	0	0	0	0	0	0	0
RealDirtyies	0	0	0	0	1440	777	0	1051
0	0	0	0	0	0	660	0	0
PrefetchReqs	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0
PrefetchNotInMem	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0
PrefetchInMem	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
Reads	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0	0
PReadBlks	0	0	0	0	0	0	0	0
0	0	0	0	0	4096	0	0	0
PReadKB	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
ReReads	0	0	0	0	0	0	0	0
0	0	0	0	0	2	0	0	0
Writes	0	0	0	0	0	0	0	10
0	0	0	0	0	0	0	0	0
PWriteBlks	0	0	0	0	0	0	0	80
0	0	0	0	0	0	0	0	0
PWriteKB	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
GrabbedDirty	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
ReadRemoteRpc	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
ReadRemotePhyIO	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

STATS-NAME	VALUE
BusyWaits	0
LRUNumLocks	136253
LRUNumSpinsWoTO	0 0%
LRUNumSpinLoops	2780
LRUNumTimeOuts	2780 -0.02%
BmapHTNumLocks	0
BmapHTNumWaits	0 0%
CacheTeamTimesWoken	1
CacheTeamNumAsleep	10
BmapHTMaxEntries	4096
BmapHTNEntries	17
BmapHTNInserts	2334
BmapHTNCollisn	0
BmapHTNFinds	183
BmapHTNHits	0
BmapHTNHits1	0
BmapHTNHits2	0
BmapHTNClears	2327
BmapHTNLChain	0
BmapHTNRehash	0
BlockmapMutexsNLocks	0



BlockmapMutexsNwaits	0	
BlockmapUID	2380	
BlockmapUIDnallocs	2335	
BlockmapRegEver	2344	
BlockmapRegisters	2334	
BufHTNBuckets	4608	
BufHTNEntries	24	
BufHTNw2orMore	0	
BufHTMaxBucketSize	3	
BufHTNFoiledOps	0	
IONumLocks	0	
IONumWaits	0	0%

示例 3:

在 20 分钟后显示缓冲池（主存储和临时存储）的输出。

```
sp_iqsysmon '00:20:00', 'mbufpool tbufpool'

=====
Buffer Pool (Main)
=====

STATS-NAME          TOTAL      NONE  TXTPOS  TXTDOC  CMPACT
BTREEV  BTREEF      BV      VDO  DBEXT    DBID    SORT  STORE  GARRAY
MovedToMRU          68731      0      0      0      0      0      0      9094
2767      0      21083      0      0      0      0      303
MovedToWash          0      0      0      0      0      0      0      0
0      0      0      0      0      0      0      0      0
RemovedFromLRU      67564      0      0      0      0      0      0      0
9020      2597      0      20830      0      0      0      0      274
RemovedFromWash     11457      0      0      0      0      0      0      0
1559      356      0      2189      0      0      0      0      68
RemovedInScanMode      0      0      0      0      0      0      0      0
0      0      0      0      0      0      0      0      0
MovedToPSList          0      0      0      0      0      0      0      0
0      0      0      0      0      0      0      0      0
RemovedFromPSList      0      0      0      0      0      0      0      0
0      0      0      0      0      0      0      0      0

STATS-NAME (cont'd)  BARRAY  BLKMAP  HASH  CKPT  BM
TEST  CMID  RIDCA  LOB  LVCRID  FILE  RIDMAP  RVLOG
MovedToMRU          2169  8561      0      0      0      24754
0      0      0      0      0      0      0
MovedToWash          0      0      0      0      0      0
0      0      0      0      0      0      0
RemovedFromLRU      2065  8330      0      0      0      24448
0      0      0      0      0      0      0
RemovedFromWash      233  1437      0      0      0      5615
0      0      0      0      0      0      0
RemovedInScanMode      0      0      0      0      0      0
0      0      0      0      0      0      0
MovedToPSList          0      0      0      0      0      0
0      0      0      0      0      0      0
RemovedFromPSList      0      0      0      0      0      0
0      0      0      0      0      0      0
```

# 系统过程

STATS-NAME	VALUE
Pages	2787
InUse	1208 ( 43.3% )
Dirty	11 ( 0.4% )
Pinned	19 ( 0.7% )
Flushes	0
FlushedBufferCount	0
GetPageFrame	1605
GetPageFrameFailure	0
GotEmptyFrame	1605
Washed	0
TimesSweepersWoken	0
PriorityWashed	0
NPrioritySweepersWoken	0
washTeamSize	10
WashMaxSize	455 ( 16.3% )
washNBuffers	455 ( 16.3% )
washNDirtyBuffers	0 ( 0.0% )
washSignalThreshold	46 ( 1.7% )
washNActiveSweepers	0
NPriorityWashBuffers	0
NActivePrioritySweepers	0
washIntensity	0
FlushAndEmpties	0
EmptiedBufferCount	0
EmptiedSkippedCount	0
EmptiedWriteCount	0
EmptiedErrorCount	0
nAffinityTotal	0 ( 0.0% )
nAffinityArea	0 ( 0.0% )

=====  
 Buffer Pool (Temporary)  
 =====

STATS-NAME				TOTAL	NONE	TXTPOS	TXTDOC	CMPACT
BTREEV	BTREEF	BV	VDO	DBEXT	DBID	SORT	STORE	GARRAY
MovedToMRU			30514	0	0	0	0	0
0	0	0	0	0	0	1218	696	0
MovedToWash			258	0	0	0	0	0
0	0	0	0	0	0	0	256	0
RemovedFromLRU			30506	0	0	0	0	0
0	0	0	0	0	0	1218	694	0
RemovedFromWash			30503	0	0	0	0	0
0	0	0	0	0	0	1218	694	0
RemovedInScanMode			0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
MovedToPSList			0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
RemovedFromPSList			0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

  

STATS-NAME (cont'd)				BARRAY	BLKMAP	HASH	CKPT	BM
TEST	CMID	RIDCA	LOB	LVCRID	FILE	RIDMAP	RVLOG	
MovedToMRU			0	8575	124	0	19898	

```

0      0      0      0      0      3      0      0      0
MovedToWash
0      0      0      0      0      2      0      0      0
RemovedFromLRU
0      0      0      0      0      8569    124     0    19898
RemovedFromWash
0      0      0      0      0      8569    124     0    19898
RemovedInScanMode
0      0      0      0      0      0      0      0      0
MovedToPSList
0      0      0      0      0      0      0      0      0
RemovedFromPSList
0      0      0      0      0      0      0      0      0

```

```

STATS-NAME          VALUE
Pages                2787
InUse                24 ( 0.9% )
Dirty                17 ( 0.6% )
Pinned                4 ( 0.1% )
Flushes              0
FlushedBufferCount  0
GetPageFrame         5684
GetPageFrameFailure  0
GotEmptyFrame        5684
Washed               0
TimesSweepersWoken  0
PriorityWashed        0
NPrioritySweepersWoken 0
washTeamSize         10
WashMaxSize          455 ( 16.3% )
washNBuffers         20 ( 0.7% )
washNDirtyBuffers    13 ( 0.5% )
washSignalThreshold  46 ( 1.7% )
washNActiveSweepers  0
NPriorityWashBuffers  0
NActivePrioritySweepers 0
washIntensity        0
FlushAndEmpties     0
EmptiedBufferCount  0
EmptiedSkippedCount  0
EmptiedWriteCount    0
EmptiedErrorCount    0
nAffinityTotal       0 ( 0.0% )
nAffinityArea        0 ( 0.0% )

```

#### 示例 4:

在 20 分钟后显示预取管理器（主存储和临时存储）的输出。

```
sp_iqsysmon '00:20:00', 'mprefetch tprefetch'
```

```

=====
Prefetch Manager (Main)
=====

```

```
STATS-NAME          VALUE
```

## 系统过程

```
PFMgrNThreads          10
PFMgrNSubmitted        81
PFMgrNDropped          0
PFMgrNValid            0
PFMgrNRead             1
PFMgrNReading          0
PFMgrCondVar           Locks 0 Lock-Waits 0 ( 0.0% ) Signals 0
Broadcasts 2 Waits 2

=====
Prefetch Manager (Temporary)
=====

STATS-NAME              VALUE
PFMgrNThreads          10
PFMgrNSubmitted        1
PFMgrNDropped          0
PFMgrNValid            0
PFMgrNRead             1
PFMgrNReading          0
PFMgrCondVar           Locks 0 Lock-Waits 0 ( 0.0% ) Signals 0
Broadcasts 2 Waits 2
```

示例 5:

在 20 分钟后显示 IQ 存储库空闲列表（主存储和临时存储）的输出。

```
sp_iqsysmon '00:20:00', 'mfreelist tfreelist'
```

```
=====
IQ Store (Main) Free List
=====
```

```
STATS-NAME              VALUE
FLBitCount              74036
FLIsOutOfSpace          NO
FLMutexLocks            0
FLMutexWaits            0 ( 0.0% )
```

```
=====
IQ Store (Temporary) Free List
=====
```

```
STATS-NAME              VALUE
FLBitCount              4784
FLIsOutOfSpace          NO
FLMutexLocks            0
FLMutexWaits            0 ( 0.0% )
```

示例 6:

在 20 分钟后显示内存管理器、线程管理器、CPU 利用率以及事务管理器的输出。

```
sp_iqsysmon '00:20:00', 'memory threads cpu txn'
```

```
=====
Memory Manager
```

```

=====
STATS-NAME                               VALUE
MemAllocated                             67599536 ( 66015 KB )
MemAllocatedMax                           160044816 ( 156293 KB )
MemAllocatedEver                           1009672456 ( 986008 KB )
MemNAllocated                               77309
MemNAllocatedEver                           914028
MemNTimesLocked                             0
MemNTimesWaited                             0 ( 0.0 %)
=====

Thread Manager
=====

STATS-NAME                               VALUE
ThrNumOfCpus                               4
ThreadLimit                                 99
ThrNumThreads                               98 ( 99.0 %)
ThrReserved                                 15 ( 15.2 %)
ThrNumFree                                  55 ( 55.6 %)
NumThrUsed                                  44 ( 44.4 %)
UsedPerActiveCmd                            22
ThrNTeamsInUse                              5
ThrMaxTeams                                 7
NumTeamsAlloc                               238
TeamThrAlloc                                421
SingleThrAlloc                              492
ThrMutexLocks                               0
ThrMutexWaits                               0 ( 0.0 %)
=====

CPU time statistics
=====

STATS-NAME                               VALUE
Elapsed Seconds                            59.65 ( 25.0 %)
CPU User Seconds                            37.79 ( 15.8 %)
CPU Sys Seconds                             1.89 ( 0.8 %)
CPU Total Seconds                           39.68 ( 16.6 %)
=====

Transaction Manager
=====

STATS-NAME                               VALUE
TxnMgrNPending                             0
TxnMgrNBlocked                             2
TxnMgrNWaiting                             0
TxnMgrPCcondvar                            Locks    0      Lock-Wait 0 ( 0.0 %)
Signals 0 Broadcasts 2 Waits 2
TxnMgrTxnIDseq                              407
TxnMgrtxncblock                             Locks    0      Lock-Wait 0 ( 0.0 %)
TxnMgrVersionID                             0
TxnMgrOAVI                                  0

```

```
TxnMgrVersionLock          Locks      0      Lock-Wait 0 ( 0.0 %)
Signals 0 Broadcasts 0 Waits 0
```

示例 7:

在 20 分钟后显示服务器上下文和目录统计信息的输出。

```
sp_iqsysmon '00:20:00', 'context catalog'

=====
Context Server statistics
=====

STATS-NAME                VALUE
StCntxNumConns            1
StCntxNResource           16
StCntxNOrigResource       18
StCntxNWaiting            0
StCntxNWaited             0
StCntxNAdmitted          1116
StCntxLock                Locks  0 Lock-Waits 0 ( 0.0 %)
StCntxCondVar             Locks  0 Lock-Waits 0 ( 0.0 %)

=====
Catalog, DB Log, and Repository statistics
=====

STATS-NAME                VALUE
CatalogLock              RdLocks 0   RdWaits 0 ( 0.0 %) RdTryFails
0 WrLocks 30037 WrWaits 0 ( 0.0 %) WrTryFail 0
DbLogMLock               Locks  0 Lock-Waits 0 ( 0.0 %)
DbLogSLock               Locks  0 Lock-Waits 0 ( 0.0 %)
RepositoryNList          0
RepositoryLock           Locks  1 SpinsWoTO 0 ( 0.0 %) Spins
0 TimeOuts 0 ( 0.0 %)
```

示例 8:

在 20 分钟后显示 IQ RLV 内存存储库和大容量内存分配器 (LMA) 统计信息的输出。

```
sp_iqsysmon '00:20:00', 'rlv lma'

=====
IQ In-Memory Store
=====

STATS-NAME                VALUE
RLV Memory Limit         2048 MB
RLV Memory Used          0 MB
RLV Chunks Used          0

=====
Large Memory Allocator
=====

STATS-NAME                VALUE
Large Memory Space       2048 MB
```

```

Large Memory Max Fle      512 MB
Large Memory Num Fle      0
Large Memory Flexibl      0.5
Large Memory Flexibl      0 MB
Large Memory Inflexi      0.9
Large Memory Inflexi      0 MB
Large Memory Anti-St      0.5
Large Memory Num Con      0

```

## sp\_iqtable 过程

显示数据库中的表的相关信息。

### 语法 1

```
sp_iqtable ( [ table_name ], [table_owner ], [table_type ] )
```

#### table\_type :

```

TEMP
| VIEW
| ALL
| any_other_value

```

### 语法 2

```
sp_iqtable [table_name='tablename'],
[table_owner='tableowner' ], [table_type='tabletype' ]
```

### 参数

- **table\_type** -
  - **TEMP** - 全局临时表
  - **VIEW** - 视图
  - **ALL** - IQ 表、全局临时表和视图
  - **any\_other\_value** - IQ 表

### 特权

您必须具有系统过程的 EXECUTE 特权。

### 注释

就语法 1 而言，如果不指定前两个参数中的任何一个参数，但指定序列中的下一个参数，则必须用 NULL 替换省略的参数。例如，sp\_iqtable NULL,NULL,TEMP 和 sp\_iqtable NULL,dbo,SYSTEM。

---

**注意：**在语法 1 中，*table\_type* 值 ALL 和 VIEW 必须用单引号引起来。

---

就语法 2 而言，可按任意顺序指定参数。将其用单引号引起来。

如果指定一个参数，则将仅返回与该参数匹配的表。如果指定多个参数，将根据指定的所有参数对结果进行过滤。如果不指定任何参数，则将返回数据库中的所有 SAP Sybase IQ 表。无法返回局部临时表的名称。

列名	描述
table_name	表的名称。
table_type	<ul style="list-style-type: none"> <li>• <b>BASE</b> - 基表。</li> <li>• <b>MAT VIEW</b> - 实例化视图。(仅 SA 表)</li> <li>• <b>GBL TEMP</b> - 全局临时表。</li> <li>• <b>PARTITION</b> - 表分区 (该表仅供内部使用, SAP Sybase IQ 用户无法使用)。</li> <li>• <b>VIEW</b> - 视图。</li> </ul>
table_owner	表的所有者
server_type	<p>IQ - 在 IQ 存储中创建的对象。</p> <p>SA - 在 SA 存储中创建的对象。</p> <p>所有视图都是在 SA 存储中创建的。</p>
location	<p>TEMP - IQ 临时存储。</p> <p>MAIN - IQ 存储。</p> <p>SYSTEM - 目录存储。</p>
dbspace_id	标识 dbspace 的编号。
isPartitioned	如果列属于分区表并有一个或多个分区, 而列分区的 dbspace 与表分区的 dbspace 不同, 则为 "Y", 如果列的表没有分区或列的每个分区驻留在与表分区相同的 dbspace 中, 则为 "N"。
remarks	使用 <b>COMMENT</b> 语句添加的用户注释。
table_constraints	对表的约束。
PartitionType	<p>如果进行了分区, 则表示分区类型。</p> <ul style="list-style-type: none"> <li>• 散列范围</li> <li>• 范围</li> <li>• 散列</li> <li>• 无</li> </ul>
isRLV	表示表是否启用了 RLV。

另请参见

- sp\_iqcolumn 过程 (第 401 页)
- sp\_iqconstraint 过程 (第 409 页)
- sp\_iqdatatype 过程 (第 415 页)



- sp\_iqevent 过程 (第 436 页)
- sp\_iqhelp 过程 (第 441 页)
- sp\_iqindex 和 sp\_iqindex\_alt 过程 (第 447 页)
- sp\_iqpkeys 过程 (第 485 页)
- sp\_iqprocparm 过程 (第 489 页)
- sp\_iq\_reset\_identity 过程 (第 499 页)
- sp\_iqview 过程 (第 554 页)
- 确定数据库所使用的安全模型 (第 381 页)

### **sp\_iqtable 过程示例**

#### **sp\_iqtable 输出示例。**

以下两种语法变体都返回表 Departments 的相关信息：

```
sp_iqtable ('Departments')
```

```
sp_iqtable table_name='Departments'
```

Table_name	Table_type	Table_owner
Departments	BASE	GROUPO

Server_type	Location	dbspace_id
IQ	主	16387

isPartitioned	注释	table_constraints
N	包含体育用品公司中各部门的名称及部门领导名称	(NULL)

PartitionType	isRlv
无	F

以下两种语法变体都会返回表所有者 GROUPO 拥有的所有表：

```
sp_iqtable NULL, GROUPO
sp_iqtable table_owner='GROUPO'
```

Table_name	Table_type	Table_owner	Server_type	Location
Contacts	BASE	GROUPO	IQ	主
Customers	BASE	GROUPO	IQ	主
Departments	BASE	GROUPO	IQ	主

Table_name	Table_type	Table_owner	Server_type	Location
Employees	BASE	GROUPO	IQ	主
FinancialCodes	BASE	GROUPO	IQ	主
FinancialData	BASE	GROUPO	IQ	主
Products	BASE	GROUPO	IQ	主
SalesOrders	BASE	GROUPO	IQ	主
SalesOrderItems	BASE	GROUPO	IQ	主

dbspace_id	isPartitioned	注释	table_constraints
16387	N	公司希望保留其联系信息的所有人员的姓名、地址和电话号码	(NULL)
16387	N	体育用品公司的客户	(NULL)
16387	N	包含体育用品公司中各部门的名称及部门领导名称	(NULL)
16387	N	包含姓名、薪水、雇用日期和生日等信息	(NULL)
16387	N	体育用品公司的收入和支出类型	(NULL)
16387	N	体育用品公司的收入和支出	(NULL)
16387	N	体育用品公司销售的产品	(NULL)
16387	N	构成销售订单的各项	(NULL)
16387	N	客户向体育用品公司提交的销售订单	(NULL)

PartitionType	isRlvd
无	F
无	F
无	F
无	F
无	F
无	F
无	F

PartitionType	isRlvd
无	F
无	F

## sp\_iqtablesize 过程

返回指定表的大小。

### 语法

```
sp_iqtablesize ( table_owner.table_name )
```

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- MANAGE ANY DBSPACE
- ALTER ANY TABLE
- 您拥有该表

### 注释

以 KBytes 和 NBlocks (IQ 块数) 为单位返回表的总大小。此外, 还返回将表保存在内存中所需的页数, 以及压缩表时 (在磁盘上) 压缩的 IQ 页数。对于此过程, 必须指定 *table\_name* 参数。如果您是 *table\_name* 的所有者, 则不必指定 *table\_owner* 参数。

列名	描述
Ownername	所有者的名称
Tablename	表名
Columns	表中的列数
KBytes	表的物理大小 (KB)
Pages	将表保存在内存中所需的 IQ 页数
CompressedPages	(在磁盘上) 压缩表时, 压缩的 IQ 页数
NBlocks	IQ 块数
RlvLogPages	将 RLV 表日志信息存储在磁盘上所需的 IQ 页数
RlvLogKBytes	RLV 表日志的大小 (KB)。

Pages 是表的总 IQ 页数。页的测量单位是 IQ 页大小。所有内存缓冲区 (IQ 缓冲区高速缓存中的缓冲区) 的大小都相同。

磁盘上的 IQ 页经过压缩。磁盘上的每个 IQ 页使用 1 到 16 个块。如果 IQ 页大小为 128KB，则 IQ 块大小为 8KB。在这种情况下，磁盘上的单个页可能为 8、16、24、32、40、48、56、64、72、80、88、96、104、112、120 或 128KB。

用 KBytes 值除以页大小，即可得到磁盘上页的平均大小。

**注意：** SAP Sybase IQ 始终读写整个页而非块。例如，如果将某页压缩到 88K，则 IQ 会在一个 I/O 中读取并写入 88K。一般页按因数 2 到 3 进行压缩。

NBlocks 是 Kbytes 除以 IQ 块大小得到的数值。

CompressedPages 是压缩的页数。例如，如果 Pages 是 1000，CompressedPages 是 992，这表示压缩了 1000 页中的 992 页。CompressedPages 除以 Pages 的结果通常接近 100%，因为大部分页都会压缩。空白页不会压缩，因为 SAP Sybase IQ 不写入空白页。无论 IQ 页是否写满，这些页都会正常压缩。

示例

```
call sp_iqtablesize ('dba.t1')
```

Ownername	Tablename	Columns
DBA	t1	3

(继续)	Pages	CompressedPages
KBytes		
192	5	4

(继续)	RivLogPages	RivLogBytes
NBlocs		
24	96	12288

另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sp\_iqtableuse 过程

报告负载访问的表的详细使用信息。

语法

```
sp_iqtableuse
```

特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

*注释*

不会报告在 **SYSTEM** 中创建的表。

**表 112. sp\_iqtableuse 列**

列名	描述
TableName	表名称
所有者 (Owner)	表所有者的用户名
UID**	表唯一标识符
LastDT	上次访问的日期/时间
NRef	查询引用数

\*\*UID 是系统分配的用于唯一标识表实例的编号（这里的实例是在创建对象时定义的）。

**另请参见**

- sp\_iqcolumnuse 过程（第 404 页）
- sp\_iqindexadvice 过程（第 450 页）
- sp\_iqindexuse 过程（第 461 页）
- sp\_iqunusedcolumn 过程（第 549 页）
- sp\_iqunusedindex 过程（第 550 页）
- sp\_iqunusedtable 过程（第 551 页）
- sp\_iqworkmon 过程（第 559 页）
- 确定数据库所使用的安全模型（第 381 页）

**sp\_iqtransaction 过程**

显示事务和版本的相关信息。

*语法*

**sp\_iqtransaction**

*适用于*

Simplex 和 Multiplex。

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

**注释**

**sp\_iqtransaction** 将针对 SAP Sybase IQ 事务管理器中的每个事务控制块返回一行信息。列 Name、Userid 和 ConnHandle 分别是连接属性 **Name**、**Userid** 和 **Number**。行按 TxnID 排序。

**sp\_iqtransaction** 输出只包含具有进行中事务的那些连接。要包括所有连接，请使用 **sp\_iqconnection**。

**注意：** 尽管您可以使用 **sp\_iqtransaction** 来标识阻止其他用户向表中写入内容的用户，但最好选择使用 **sp\_iqlocks** 来实现此目的。

列名	描述
名称	应用程序的名称。
Userid	连接的用户 ID。
TxnID	此事务控制块的事务 ID。该事务 ID 在 <b>begin transaction</b> 时分配。它将出现在 .iqmsg 文件的 BeginTxn、CmtTxn 和 PostCmtTxn 消息中，并与打开数据库时所记录的 Txn ID Seq 相同。
CmtID	提交事务时事务管理器分配的 ID。对于活动事务，CmtID 为零。
VersionID	对于 Simplex 和 Multiplex 节点，值 0 表示事务未实现版本化，且尚未分配 VersionID。  对于 Multiplex 协调器，在事务建立表锁后分配 VersionID。Multiplex 辅助服务器自协调器接收 VersionID。VersionID 由 SAP Sybase IQ 内存中目录和 IQ 事务管理器在内部使用，以对 Multiplex 数据库中的所有节点唯一标识数据库版本。
状态	事务控制块的状态。此变量反映 SAP Sybase IQ 内部实现的详细信息，并可能在未来发生更改。当前，事务状态为 NONE、ACTIVE、ROLLING_BACK、ROLLED_BACK、COMMITTING、COMMITTED 和 APPLIED。  NONE、ROLLING_BACK、ROLLED_BACK、COMMITTING 和 APPLIED 是瞬时状态，寿命很短。  ACTIVE 指示事务处于活动状态。  COMMITTED 指示事务已完成并正在等待 APPLIED，此时，将对对于任何事务都不可见的版本执行垃圾回收。  一旦事务状态变为 ROLLED_BACK、COMMITTED 或 APPLIED，除了打开游标所持有的锁，该事务将不再拥有任何其它锁。
ConnHandle	连接的 ID 号。

列名	描述
IQConnID	十位连接 ID，是 .iqmsg 文件中所有消息的一部分。它是一个单调递增整数，在整个服务器会话内唯一。
MainTableKBCr	此事务创建的 IQ 存储空间的大小（以 KB 为单位）。
MainTableKBDr	此事务删除的 IQ 存储空间的大小（以 KB 为单位），但由于该空间在其它数据库版本或此事务的其它保存点仍可见，因此该空间仍位于磁盘存储中。
TempTableKBCr	此事务创建的用于存储 IQ 临时表数据的 IQ 临时存储空间的大小（以 KB 为单位）。
TempTableKBDr	此事务删除的 IQ 临时表空间的大小（以 KB 为单位），但由于该空间对 IQ 游标仍可见，或者归此事务的其它保存点所有，因此该空间仍位于磁盘的 IQ 临时存储库中。
TempWorkSpaceKB	<p>对于 ACTIVE 事务，该事务此时使用的工作空间的快照，如排序、散列和临时位图。该数值因您运行 <b>sp_iqtransaction</b> 的时间而异。例如，查询引擎可能会在临时高速缓存中创建 60MB，但很快释放其中的大部分空间，即使查询处理仍在继续也是如此。如果在查询完成后运行 <b>sp_iqtransaction</b>，该列显示要小得多的数值。当事务不再处于活动状态时，该列为零。</p> <p>对于 ACTIVE 事务，该列与 <b>sp_iqconnection</b> 的 TempWorkSpaceKB 列相同。</p>
TxnCreateTime	事务的开始时间。一旦建立活动连接或者前一个事务执行提交或回退操作，所有 SAP Sybase IQ 事务将立即隐式开始。
CursorCount	引用该事务控制块的打开的 SAP Sybase IQ 游标数。如果事务处于 ACTIVE 状态，则它指示事务中创建的打开的游标数。如果事务处于 COMMITTED 状态，则它指示引用该事务控制块拥有的数据库版本的 HOLD 游标数。
SpCount	事务控制块中存在的保存点结构数。保存点可以隐式创建和释放。因此，此数值不指示事务中用户创建的保存点的数量。
SpNumber	事务的活动保存点数。这是实现详细信息，因此可能不能反映用户创建的保存点。
MPXServerName	指示活动事务是否来自节点间通信 (INC) 连接。如果来自 INC 连接，则值是发起事务的 Multiplex 服务器的名称。如果不是来自 INC 连接，则为 NULL。如果事务不是活动的，则始终为 NULL。
GlobalTxnID	与当前事务关联的全局事务 ID，如果没有则为 0（零）。

列名	描述
VersioningType	事务的快照版本控制类型；表级（缺省值）或行级。行级快照版本控制 (RLV) 仅适用于启用 RLV 的表。一旦启动某个事务，该值便不能进行更改。
Blocking	指示连接阻塞是已启用 (True) 还是已禁用 (False)。可使用 <b>BLOCKING</b> 数据库选项设置连接阻塞。如果为 true，事务阻塞，即事务先等待解除冲突锁，然后尝试重试锁定请求。
BlockingTimeout	指示事务等待清除锁定冲突的时间（毫秒）。可使用 <b>BLOCKING_TIMEOUT</b> 数据库选项设置超时阈值。值 0（缺省值）表示事务无限期待。

示例

示例 `sp_iqtransaction` 输出：

```

Name      Userid  TxnID  CmtID  VersionID  State  ConnHandle  IQConnID
=====  =====  =====  =====  =====  =====  =====  =====
red2      DBA     10058  10700  10058     Active  419740283  14

MainTableKBCr      MainTableKBDr      TempTableKBCr  TempTableKBDr
=====  =====  =====  =====
0          0          65824          0

TempWorkSpaceKB  TxnCreateTime          CursorCount  SpCount
SpNumber
=====
0          2013-03-26 13:17:27.612          1          3          2

MPXServerName  GlobalTxnID  VersioningType  Blocking
BlockingTimeout
=====
0          (NULL)      0          Row-level      True
    
```

另请参见

- `sp_iqstatus` 过程（第 515 页）
- `sp_iqversionuse` 过程（第 552 页）
- 确定数据库所使用的安全模型（第 381 页）



## sp\_iqunusedcolumn 过程

报告负载未引用的 IQ 列。

### 语法

```
sp_iqunusedcolumn
```

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

### 注释

不会报告在 SYSTEM 中创建的表和本地临时表的列。

**表 113. sp\_iqunusedcolumn 列**

列名	描述
TableName	表名称
ColumnName	列名
所有者 (Owner)	列所有者的用户名

### 示例

**sp\_iqunusedcolumn** 过程的输出样本：

TableName	ColumnName	Owner
SalesOrders	ID	GROUP0
SalesOrders	CustomerID	GROUP0
SalesOrders	OrderDate	GROUP0
SalesOrders	FinancialCode	GROUP0
SalesOrders	Region	GROUP0
SalesOrders	SalesRepresentative	GROUP0
SalesOrderItems	ID	GROUP0
SalesOrderItems	LineID	GROUP0
SalesOrderItems	ProductID	GROUP0
SalesOrderItems	Quantity	GROUP0
SalesOrderItems	ShipDate	GROUP0
Contacts	ID	GROUP0
Contacts	Surname	GROUP0
Contacts	GivenName	GROUP0 ...

### 另请参见

- [sp\\_iqcolumnuse](#) 过程 (第 404 页)
- [sp\\_iqindexadvice](#) 过程 (第 450 页)

- sp\_iqindexuse 过程 (第 461 页)
- sp\_iqtableuse 过程 (第 544 页)
- sp\_iqunusedindex 过程 (第 550 页)
- sp\_iqunusedtable 过程 (第 551 页)
- sp\_iqworkmon 过程 (第 559 页)
- 确定数据库所使用的安全模型 (第 381 页)

## sp\_iqunusedindex 过程

报告负载未引用的 IQ 辅助 (非 FP) 索引。

### 语法

**sp\_iqunusedindex**

### 特权

您必须具有系统过程的 EXECUTE 特权, 以及 MONITOR 系统特权。

### 注释

不会报告在 SYSTEM 中创建的表或本地临时表的索引。

**表 114. sp\_iqunusedindex 列**

列名	描述
IndexName	索引名
TableName	表名称
所有者 (Owner)	索引所有者的用户名
IndexType	索引类型

### 示例

**sp\_iqunusedindex** 过程的输出样本:

IndexName	TableName	Owner	IndexType
ASIQ_IDX_T450_I7_HG	SalesOrders	GROUP0	HG
ASIQ_IDX_T450_C6_HG	SalesOrders	GROUP0	HG
ASIQ_IDX_T450_C4_HG	SalesOrders	GROUP0	HG
ASIQ_IDX_T450_C2_HG	SalesOrders	GROUP0	HG
ASIQ_IDX_T451_I6_HG	SalesOrderItems	GROUP0	HG
ASIQ_IDX_T451_C3_HG	SalesOrderItems	GROUP0	HG
ASIQ_IDX_T451_C1_HG	SalesOrderItems	GROUP0	HG
ASIQ_IDX_T452_I11_HG	Contacts	GROUP0	HG
ASIQ_IDX_T453_I10_HG	Contacts	GROUP0	HG
ASIQ_IDX_T454_I4_HG	FinancialCodes	GROUP0	HG

ASIQ_IDX_T455_I5_HG	FinancialData	GROUPO	HG
ASIQ_IDX_T455_C3_HG	FinancialData	GROUPO	HG
ASIQ_IDX_T456_I8_HG	Products	GROUPO	HG
ASIQ_IDX_T457_I4_HG	Departments	GROUPO	HG
ASIQ_IDX_T457_C3_HG	Departments	GROUPO	HG
ASIQ_IDX_T458_I21_HG	Departments	GROUPO	HG
ASIQ_IDX_T458_C5_HG	Departments	GROUPO	HG

**另请参见**

- `sp_iqcolumnuse` 过程 (第 404 页)
- `sp_iqindexadvice` 过程 (第 450 页)
- `sp_iqindexuse` 过程 (第 461 页)
- `sp_iqtableuse` 过程 (第 544 页)
- `sp_iqunusedcolumn` 过程 (第 549 页)
- `sp_iqunusedtable` 过程 (第 551 页)
- `sp_iqworkmon` 过程 (第 559 页)
- 确定数据库所使用的安全模型 (第 381 页)

**sp\_iqunusedtable 过程**

报告负载未引用的 IQ 表。

*语法*

```
sp_iqunusedtable
```

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

*注释*

不会报告在 SYSTEM 中创建的表和本地临时表。

**表 115. sp\_iqunusedtable 列**

列名	描述
TableName	表名称
所有者 (Owner)	表所有者的用户名

*示例*

下表说明 `sp_iqunusedtable` 过程的输出样本。

TableName	Owner	FinancialCodes	GROUPO
Contacts	GROUPO	Employees	GROUPO
emp1	DBA	SalesOrders	GROUPO

FinancialData	GROUPO Departments	GROUPO
SalesOrderItems	GROUPO Products	GROUP
iq_dummy	DBA Customers	GROUPO
sale	DBA	

### 另请参见

- `sp_iqcolumnuse` 过程 (第 404 页)
- `sp_iqindexadvice` 过程 (第 450 页)
- `sp_iqindexuse` 过程 (第 461 页)
- `sp_iqtableuse` 过程 (第 544 页)
- `sp_iqunusedcolumn` 过程 (第 549 页)
- `sp_iqunusedindex` 过程 (第 550 页)
- `sp_iqworkmon` 过程 (第 559 页)
- 确定数据库所使用的安全模型 (第 381 页)

## `sp_iqversionuse` 过程

显示 IQ 主存储库的版本使用情况。

### 语法

**sp\_iqversionuse**

### 适用于

Simplex 和 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

### 注释

**sp\_iqversionuse** 系统存储过程可帮助解决由于使用多个表版本导致数据库使用存储空间过多的情况。

如果出现空间不足状况，或者 **sp\_iqstatus** 显示 Multiplex 服务器上正在使用较高百分比的主块时，请运行 **sp\_iqversionuse** 查明正在使用的版本以及通过释放该版本可以恢复的空间量。

该过程为版本的每位用户生成一行。首先在协调器上运行 **sp\_iqversionuse** 以确定应释放哪些版本，以及当版本不再使用时要释放的空间大小（以 KB 为单位）。连接至协调器的用户的连接 ID 显示在 IQConn 列中。由辅助服务器引致的版本使用显示为连接 ID 为 0 的辅助服务器名。

空间大小用一个范围来表示，因为实际大小通常取决于释放的其它版本。释放的实际空间大小可以是 MinKBRelease 和 MaxKBRelease 之间的任意一个值。最旧版本的 MinKBRelease 总是等于 MaxKBRelease。

WasReported 列用于 Multiplex 设置中。WasReported 指示版本使用情况信息是否已从辅助服务器发送至协调器。新版本在协调器上的 WasReported 最初为 0。数据库服务器将版本使用情况信息复制回协调器后，WasReported 便更改为 1。

在 Multiplex 辅助服务器上运行 `sp_iqversionuse`，以确定指向辅助服务器的各个连接。来自其它服务器的用户不会显示在辅助服务器上。

表 116. `sp_iqversionuse` 列

列名	描述
VersionID	在 Simplex 数据库中，VersionID 显示为零。对于 Multiplex 协调器，VersionID 与活动事务的 TxnID 相同，VersionID 与已提交事务的 CmtID 相同。在 Multiplex 辅助服务器中，VersionID 是在 Multiplex 协调器上创建数据库版本的事务的 CmtID。VersionID 由 SAP Sybase IQ 内存目录和 SAP Sybase IQ 事务管理器在内部使用，以对 Multiplex 数据库中的所有节点唯一标识数据库版本。
Server	该版本的用户所连接的服务器。
IQConnID	使用该版本的连接 ID。
WasReported	指示服务器是否已接收到该版本的使用情况信息。
MinKBRelease	该版本不再使用时返回的最小空间量。
MaxKBRelease	该版本不再使用时返回的最大空间量。

### 示例

`sp_iqversionuse` 系统过程的样本输出：

```
VersionID Server          IQConnID WasReported
=====  =====
          0 ab2ab_iqdemo          9          0

MinKBRelease  MaxKBRelease
=====
          0          0
```

以下示例显示 Multiplex 输出。在协调器 (*mpxw*) 上，连接 108 正在使用最旧版本 42648。提交或回退连接 108 上的事务可释放 7.9 MB 的空间。根据协调器的输出，辅助服务器 (*mpxq*) 正在使用版本 42686。根据辅助服务器的输出，实际连接为连接 31。释放版本 42686 所返回的实际空间大小取决于是否先释放 42648。

在协调器上，版本 42715 和 42728 的 WasReported 为 0，因为这些都是尚未复制的新版本。由于版本 42728 没有显示在辅助服务器的输出中，因此它尚未由辅助服务器使用。

`sp_iqversionuse` 在协调器 *mpxw* 上执行时，返回输出：

```
call dbo.sp_iqversionuse
```

VersionID	Server	IQConn	WasReported	MinKBRelease	MaxKBRelease
42648	'mpxw'	108	1	7920	7920
42686	'mpxq'	0	1	7920	304
42702	'mpxq'	0	1	0	688
42715	'mpxq'	0	0	0	688
42728	'mpxq'	0	0	0	688

**sp\_iqversionuse** 在协调器 *mpxq* 上执行时，返回输出：

```
call dbo.sp_iqversionuse
```

VersionID	Server	IQConn	WasReported	MinKBRelease	MaxKBRelease
42686	'mpxq'	31	1	0	0
42715	'mpxq'	00	1	0	0

另请参见

- **sp\_iqstatus** 过程 (第 515 页)
- **sp\_iqtransaction** 过程 (第 545 页)
- 确定数据库所使用的安全模型 (第 381 页)

## **sp\_iqview** 过程

显示数据库中视图的相关信息。

语法 1

```
sp_iqview ([view_name],[view_owner],[view_type])
```

**view\_type** :

```
SYSTEM
| ALL
| any_other_value
```

语法 2

```
sp_iqview [view_name='viewname'],
[view_owner='viewowner' ],[view_type='viewtype' ]
```

参数

- **view\_type** -
  - **SYSTEM** - 系统视图

- **ALL** - 用户和系统视图
- **any\_other\_value** - 用户视图

#### 特权

您必须具有系统过程的 EXECUTE 特权。

#### 注释

对于语法 1, **sp\_iqview NULL,NULL,SYSTEM**, 如果不指定前两个参数中的任何一个, 但指定序列中的下一个参数, 则必须用 **NULL** 替换省略的参数。

例如: `sp_iqview NULL,NULL,SYSTEM` 和 `sp_iqview deptview,NULL,'ALL'`。

---

**注意:** 在语法 1 中, *view\_type* 值 **ALL** 必须用单引号引起来。

---

对于语法 2, 参数可以按任何顺序进行指定, 并用单引号引起。

如果指定参数之一, 则仅返回具有指定视图名的视图或指定用户拥有的视图。如果指定多个参数, 将根据指定的所有参数对结果进行过滤。如果不指定参数, 将返回数据库中的所有用户视图。

**表 117. sp\_iqview 列**

列名	描述
view_name	视图的名称
view_owner	视图的所有者
view_def	在 <b>CREATE VIEW</b> 语句中指定的视图定义
remarks	使用 <b>COMMENT</b> 语句添加的用户注释

**sp\_iqview** 将返回大于 32K 字符而没有截断的视图定义。

#### 另请参见

- `sp_iqcolumn` 过程 (第 401 页)
- `sp_iqconstraint` 过程 (第 409 页)
- `sp_iqdatatype` 过程 (第 415 页)
- `sp_iqevent` 过程 (第 436 页)
- `sp_iqhelp` 过程 (第 441 页)
- `sp_iqindex` 和 `sp_iqindex_alt` 过程 (第 447 页)
- `sp_iqpkeys` 过程 (第 485 页)
- `sp_iqprocparm` 过程 (第 489 页)
- `sp_iq_reset_identity` 过程 (第 499 页)
- `sp_iqtable` 过程 (第 539 页)

- 确定数据库所使用的安全模型（第 381 页）

### sp\_iqview 过程示例

使用示例作为 **sp\_iqview** 用法的参考。

以下两种不同形式的语法都会返回视图 deptview 的相关信息：

```
call sp_iqview('ViewSalesOrders')
```

```
sp_iqview view_name='ViewSalesOrders'
```

以下两种不同形式的语法都会返回视图所有者 GROUPO 拥有的所有视图：

```
sp_iqview NULL,GROUPO
```

```
sp_iqview view_owner='GROUPO'
```

view_name	view_owner	view_def	remarks
ViewSalesOrders	GROUPO	Create views GROUPO , ViewSalesOrders( ID, LineID, ProductID, Quantity, OrderDate, Ship-Date, Region, SalesRepresentativeName	(NULL)

### sp\_iqwho 过程

显示所有当前用户和连接的相关信息，或显示特定用户或连接的相关信息。

语法

```
sp_iqwho [ { connhandle | user-name } [, arg-type ] ]
```

参数

- **connhandle** - 表示连接 ID 的整数。如果已指定该参数，则 **sp\_iqwho** 仅返回指定连接的相关信息。如果指定的连接未打开，则输出中不显示任何行。
- **user-name** - 表示用户登录名的 char(255) 参数。如果指定此参数，则 **sp\_iqwho** 将仅返回指定用户的相关信息。如果指定的用户尚未打开任何连接，则输出中不显示任何行。如果在数据库中不存在指定的用户名，则 **sp\_iqwho** 会返回错误消息 “User user-name does not exist”
- **arg-type** - 是可选的，并且只有在指定第一个参数以后才能指定该参数。*arg-type* 的值只能是 “user”。如果将 *arg-type* 的值指定为 “user”，则 **sp\_iqwho** 会将第一个参数解释为用户名，即使第一个参数为数字。如果为 *arg-type* 指定 “user” 以外的任何值，则 **sp\_iqwho** 将返回错误 “无效参数”。

请用双引号将 *arg-type* 值引起来。

特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- DROP CONNECTION



- MONITOR
- SERVER OPERATOR

*注释*

**sp\_iqwho** 存储过程显示所有当前用户和连接的相关信息，或显示特定用户或连接的相关信息。

列名	描述
ConnHandle	SA 连接句柄
IQConnID	SAP Sybase IQ 特定连接 ID
Userid	打开连接“ConnHandle”的用户的姓名
BlockedOn	在其上阻塞特定连接的连接；如果在任何连接上均未进行阻塞则为 0
BlockUserid	阻塞连接的所有者；如果没有阻塞连接则为 NULL
ReqType	通过连接进行的请求的类型；如果未发出任何命令则为 DO_NOTHING
IQCmdType	连接所发出的 SAP Sybase IQ 命令的类型；如果未发出任何命令则为 NONE
QIdle	自上个 SAP Sybase IQ 命令通过连接发出以来所经历的时间（以秒为单位）；如果没有上个 SAP Sybase IQ 命令，则显示自 "01-01-2000" 起所经历的时间
SAIdle	自通过连接发出上个 SA 请求以来所经过的时间（以秒为单位）；如果之前没有 SA 命令，则显示自 2000 年 1 月 1 日起所经过的时间。
Q Cursors	连接中的活动游标数；如果没有游标则为 0
QThreads	连接的线程数。连接打开后至少会立即启动一个线程，因此 QThreads 的最小值为 1。
TempTableSpaceKB	临时表空间的大小（以 KB 为单位）；如果没有使用临时表空间则为 0
TempWorkSpaceKB	临时工作空间的大小（以 KB 为单位）；如果没有使用临时工作空间则为 0

表 118. sp\_who 和 sp\_iqwho 列的对应关系

sp_who 列	sp_iqwho 列
fid	块所属的系列；已忽略，因为不适用于 SAP Sybase IQ
spid	ConnHandle、IQConnID
status	QIdle、SAIdle

sp_who 列	sp_iqwho 列
loginame	Userid
origname	用户别名; 已忽略, 因为不适用于 SAP Sybase IQ
hostname	运行服务器的主机的名称; 当前不支持
blk_spid	BlockedOn
dbname	已忽略, 因为对于 SAP Sybase IQ, 有一个服务器和一个数据库, 并且对于每个连接它们都相同
cmd	ReqType、IQCmdType
block_xloid	BlockUserid

如果未指定任何参数, 则 **sp\_iqwho** 将显示所有当前活动连接和用户的相关信息。

第一个 **sp\_iqwho** 参数可以指定为连接句柄或用户名。参数 *connhandle* 和 *user-name* 都是排它参数和可选参数。一次只能指定这些参数中的一个。缺省情况下, 如果第一个参数是数字, 则假定此参数为连接句柄。如果第一个参数不是数字, 则假定此参数为用户名。

SAP Sybase IQ 允许数字形式的用户名。*arg-type* 参数指示 **sp\_iqwho** 将第一个参数中的数值解释为用户名。例如:

```
sp_iqwho 1, "user"
```

指定 *arg-type* "user" 时, **sp\_iqwho** 将第一个参数解释为用户名, 而不是连接 ID。如果名为 1 的用户存在于数据库中, 则 **sp\_iqwho** 将显示由用户 1 打开的连接相关的信息。

语法	输出
sp_iqwho	显示所有活动连接
sp_iqwho 3	显示连接 3 的相关信息
sp_iqwho "DBA"	显示由用户 DBA 打开的连接
sp_iqwho 3, "user"	将 3 解释为用户名, 并显示由用户 3 打开的连接。如果用户 3 不存在, 则返回错误 "User 3 does not exist"
sp_iqwho non-existing-user	返回错误 "User non-existing-user does not exist"
sp_iqwho 3, "xyz"	返回错误 "Invalid parameter: xyz"

### sp\_iqwho 过程示例

使用示例作为 **sp\_iqwho** 用法的参考。

显示所有活动连接:

ConnHandle	IQConnID	Userid	ReqType	IQCmdType	Blocked
On	BlockUserid	IQCursors			
12	118	DBA	CURSOR_OPEN	IQUTILITYOPENCURSOR	0
(NULL)	0				
13	119	shweta	DO_NOTHING	NONE	0
(NULL)	0				
IQThreads	IQIdle	SAIdle	TempTableSpaceKB	TempWorkSpaceKB	
1	1	0	0	0	
1	16238757	470	0	0	

### sp\_iqwho 与 Adaptive Server 的兼容性

SAP Sybase IQ **sp\_iqwho** 存储过程将合并由 Adaptive Server **sp\_who** 过程显示的 SAP Sybase IQ 等效列。

一些 Adaptive Server 列会被忽略，因为它们并不适用于 SAP Sybase IQ。

### sp\_iqworkmon 过程

控制对负载监控器使用信息的收集，并报告监控收集状态。**sp\_iqworkmon** 将收集所有 SQL 语句信息。

语法

```
sp_iqworkmon [ 'action' ] [ , 'mode' ]
```

```
action = 'start' , 'stop' , 'status' , 'reset'
```

```
mode = 'index' , 'table' , 'column' , 'all'
```

例如:

```
sp_iqworkmon 'start' , 'all'
```

如果指定了一个参数，则该参数只能为 *action*。例如:

```
sp_iqworkmon 'stop'
```

特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

注释

参数	描述
action	<p>指定要应用的控制操作。<i>start</i> 值针对指定模式立即启动监控。<i>stop</i> 值立即停止监控。<i>status</i> 值 (缺省值) 显示当前状态, 而不会更改状态。</p> <p>系统始终会保留统计信息, 除非使用 <i>reset</i> 参数清除它们, 或重新启动服务器。统计信息集合不会在服务器重新启动后自动恢复, 需要使用 <i>start</i> 重新启动该集合。</p>
mode	<p>指定要控制的监控类型。<b>INDEX</b>、<b>TABLE</b> 和 <b>COLUMN</b> 关键字分别单独控制对索引使用情况、表使用情况和列使用情况的监控。缺省的 <b>ALL</b> 关键字同时控制对所有使用情况监控功能的监控。</p>

执行 **sp\_iqworkmon** 时总存在一个结果集。如果指定特定模式 (如索引), 则只显示该模式所对应的行。

列名	描述
MonMode	表、索引或列
状态	已启动或已停止
Rowcount	当前收集的行数

示例

**sp\_iqworkmon** 过程的输出样本:

```

MonMode      Status      Rowcount index      started      15
table        started    10 column      started      31
    
```

另请参见

- [sp\\_iqcolumnuse 过程 \(第 404 页\)](#)
- [sp\\_iqindexadvice 过程 \(第 450 页\)](#)
- [sp\\_iqindexuse 过程 \(第 461 页\)](#)
- [sp\\_iqtableuse 过程 \(第 544 页\)](#)
- [sp\\_iqunusedcolumn 过程 \(第 549 页\)](#)
- [sp\\_iqunusedindex 过程 \(第 550 页\)](#)
- [sp\\_iqunusedtable 过程 \(第 551 页\)](#)

## 按字母顺序排列的目录存储过程的列表

Catalog 存储存储过程以表格形式返回显示数据库服务器、数据库和连接属性的结果集。

这些过程由 **dbo** 用户 ID 拥有。**PUBLIC** 角色对它们拥有 **EXECUTE** 权限。

## sa\_ansi\_standard\_packages 系统过程

返回有关 SQL 语句中使用的非核心 SQL 扩充的信息。

### 语法

```
sa_ansi_standard_packages (
  standard
, statement
)
```

### 参数

- **standard** - 使用此 LONG VARCHAR 参数指定用于核心扩展的标准。SQL:1999 或 SQL:2003 其中之一。
- **statement** - 使用此 LONG VARCHAR 参数指定用于评估的 SQL 语句。

### 结果集

列名	数据类型	说明
package_id	VARCHAR(10)	功能标识符。
package_name	LONG VARCHAR	功能名称。

### 注释

如果存在用于语句的非核心扩展，则结果集为空。

### 特权

无

### 副作用

无

## sa\_audit\_string 系统过程

将字符串添加到事务日志中。

### 语法

```
sa_audit_string( string )
```

### 参数

- **string** - 要添加到事务日志中的 VARCHAR(128) 字符串。

### 注释

如果审计功能是打开的，此系统过程会向存储在事务日志中的审计信息添加一条注释。字符串的最大长度可以是 128 个字符。

*特权*

您必须具有 **MANAGE AUDITING** 系统特权。

*副作用*

无

**示例**

以下示例使用 `sa_audit_string` 向事务日志添加一条注释：

```
CALL sa_audit_string( 'Auditing test' );
```

**另请参见**

- 确定数据库所使用的安全模型（第 381 页）

## **sa\_char\_terms** 系统过程

将 CHAR 字符串拆分为多个术语，并以行返回每个术语及其位置。

*语法*

```
sa_char_terms( 'char-string' [, 'text-config-name'
[, 'owner' ] ] )
```

*参数*

- **char-string** - 所分析的 CHAR 字符串。
- **text-config-name** - 处理字符串时应用的文本配置对象。缺省值为 'default\_char'。
- **owner** - 指定的文本配置对象的所有者。缺省值为 DBA。

*特权*

无

*注释*

可以使用 **sa\_char\_terms** 来查明在应用文本配置对象的设置时如何解释字符串。这可以帮助您了解哪些术语将在编制索引期间被删除或者将从查询字符串中被删除。

*示例*

返回 CHAR 字符串中的术语 'the quick brown fox jumped over the fence':

```
CALL sa_char_terms
( 'the quick brown fox jumped over the fence' );
```

**表 119. CHAR 字符串解释**

术语	位置
the	1

术语	位置
quick	2
brown	3
fox	4
jumped	5
over	6
the	7
fence	8

## sa\_checkpoint\_execute 系统过程

允许在检查点操作期间执行 shell 命令。

### 语法

```
sa_checkpoint_execute 'shell_commands'
```

### 参数

- **shell\_commands** - 需要在系统 shell 中执行一个或多个用户命令。shell 命令特定于系统 shell。命令以分号 (;) 分隔。

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种 系统特权：

- CHECKPOINT
- MANAGE ANY MIRROR SERVER

### 注释

当数据库处于抑制状态时，允许用户执行 shell 命令，以在检查点操作期间复制正在运行的数据库。复制的数据库可以启动并进行常规恢复，类似于发生系统故障后所进行的恢复。

**sa\_checkpoint\_execute** 启动检查点，然后在检查点操作期间执行系统 shell，将用户命令传递至 shell。然后，服务器等候 shell 完成，创建一个在其中复制数据库文件的任意大小时间窗口。由于在执行检查点期间大部分数据库活动都会停止，因此 shell 命令的持续时间应限制在可接受的用户响应时间内。

如果 shell 命令返回非零状态，**sa\_checkpoint\_execute** 会返回错误。

请勿与交互式命令一起使用 **sa\_checkpoint\_execute**，因为服务器必须等待交互式命令注销。对于可能会变成交互式命令的所有 shell 命令，请提供覆盖标志以禁用提示；换句话说，**COPY**、**MOVE** 和 **DELETE** 命令可能会提示进行确认。

**sa\_checkpoint\_execute** 的用途是与磁盘镜像一起使用以拆分镜像设备。

当使用 `sa_checkpoint_execute` 将 `iqdemo.*` 文件复制到其它目录时，系统将复制除 `.db` 和 `.log` 文件以外的所有文件。返回错误 -910。

该错误并不属于产品缺陷，而是 Windows 限制；Windows 复制命令不能在数据库打开 Catalog 文件时复制这些 Catalog 文件。

*示例*

假定已创建名为 `backup` 的子目录，以下语句将发出检查点，将所有 **iqdemo** 数据库文件复制到备份子目录，然后完成检查点：

```
sa_checkpoint_execute 'cp iqdemo.* backup/'
```

**另请参见**

- 确定数据库所使用的安全模型（第 381 页）

**sa\_conn\_activity 系统过程**

为每个与服务器上指定数据库的连接返回一个最近准备的 SQL 语句。

*语法*

```
sa_conn_activity( [ connidparm ] )
```

*参数*

- **connidparm** - 使用此可选的 INTEGER 参数指定连接 ID 号。缺省值为 NULL。

*结果集*

列名	数据类型	说明
Number	INTEGER	返回当前连接的连接 ID（数字）。
Name	VARCHAR(255)	返回当前连接的名称。 临时连接名在连接名前面附有 INT：
Userid	VARCHAR(255)	返回连接的用户 ID。
DBNumber	INTEGER	返回数据库的 ID 编号。
LastReqTime	VARCHAR(255)	返回指定连接的最后一个请求开始的时间。此属性可以为内部连接（如事件）返回空字符串。
LastStatement	LONG VARCHAR	返回最近为当前连接准备的 SQL 语句。

*注释*

如果 **connidparm** 小于零，则返回当前连接的信息。如果 **connidparm** 未提供或为 NULL，则为数据库服务器上运行的所有数据库的所有连接返回信息。



`sa_conn_activity` 系统过程将返回一个结果集，该结果集由连接的最近准备 SQL 语句组成。在调用 `sa_conn_activity` 之前，必须为数据库服务器启用语句记录。要达到此目的，可在启动数据库服务器时指定 `-zl` 选项，也可执行以下内容：

```
CALL sa_server_option('RememberLastStatement','ON');
```

当数据库服务器忙而您想获取为每个连接预准备的最近 SQL 语句的信息时，此过程很有用。此功能可以替代请求记录。

#### 特权

为当前连接 ID 执行此系统过程不需要任何特权。要为其它连接执行此系统过程，您必须具有 `SERVER OPERATOR`、`MONITOR` 或 `DROP CONNECTION` 系统特权。

#### 副作用

无

#### 示例

以下示例使用 `sa_conn_activity` 系统过程显示为每个连接最近准备的 SQL 语句。

```
CALL sa_conn_activity( );
```

Number	Name	Userid	DBNumber	...
1,949	SQL_DBC_117acc40	DBA	0	...
1,948	setup	User1	0	...
...	...	...	...	...

#### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sa\_conn\_info 系统过程

报告连接属性信息。

#### 语法

```
sa_conn_info( [ connidparm ] )
```

#### 参数

- *connidparm* - 此可选 INTEGER 参数指定连接 ID 号。缺省值为 NULL。

#### 结果集

列名	数据类型	说明
Number	INTEGER	返回当前连接的连接 ID（数字）。

列名	数据类型	说明
Name	VARCHAR(255)	返回当前连接的连接 ID (数字)。 临时连接名在连接名前面附有 INT:。
Userid	VARCHAR(255)	返回连接的用户 ID。
DBNumber	INTEGER	返回数据库的 ID 编号。
LastReqTime	VARCHAR(255)	返回指定连接的最后一个请求开始的时间。此属性可以为内部连接 (如事件) 返回空字符串。
ReqType	VARCHAR(255)	返回上次请求的类型。如果连接已由连接池进行了高速缓存, 则其 ReqType 值为 CONNECT_POOL_CACHE。
CommLink	VARCHAR(255)	返回连接的通信链接。这是 SAP Sybase IQ 所支持的网络协议之一, 或者 local, 如果为相同计算机连接。
NodeAddr	VARCHAR(255)	返回客户端/服务器连接中客户端的地址。
ClientPort	INTEGER	返回客户端的 TCP/IP 端口号; 或者如果连接不是 TCP/IP 连接, 则返回 0。
ServerPort	INTEGER	返回数据库服务器的 TCP/IP 端口号或 0。
BlockedOn	INTEGER	如果当前连接未被阻塞, 则返回零; 如果当前连接被阻塞, 则返回由于锁定冲突而阻塞连接的连接号。
LockRowID	UNSIGNED BIGINT	返回锁定行的标识符。 如果连接并未等待与行相关联的锁 (即它未等待锁或正在等待不与行相关联的锁), 则 LockRowID 为 NULL。
LockIndexID	INTEGER	返回锁定索引的标识符。 如果锁与 LockTable 中的表上的所有索引相关联, 则 LockIndexID 为 -1。如果连接并未等待与索引相关联的锁 (即它未等待锁或正在等待不与索引相关联的锁), 则 LockRowID 为 NULL。
LockTables	VARCHAR(255)	如果连接正在等待锁, 则返回与锁相关联的表名称。否则, LockTable 返回一个空字符串。
UncommitOps	INTEGER	返回未提交的操作数。
ParentConnection	INTEGER	返回已创建临时连接以执行数据库操作 (如执行备份或创建数据库) 的连接的连接 ID。对于其它类型的连接, 此属性返回 NULL。

### 注释

如果 *connidparm* 小于零，则返回由当前连接的连接属性组成的结果集。如果 *connidparm* 未提供或为 NULL，则为数据库服务器上运行的所有数据库的所有连接返回连接属性。

当发生阻塞时，使用此过程所返回的 **BlockedOn** 值可以检查哪些用户被阻塞以及是谁阻塞的。**sa\_locks** 系统过程可用于显示阻塞的连接所持有的锁。

有关基于任意这些属性的详细信息，可执行与以下类似的内容：

```
SELECT *, DB_NAME( DBNumber ),
        CONNECTION_PROPERTY( 'LastStatement', Number )
FROM sa_conn_info( );
```

**LockRowID** 的值可用于在 **sa\_locks** 过程的输出中查找锁。

**LockIndexID** 的值可用于在 **sa\_locks** 过程的输出中查找锁。此外，**LockIndexID** 的值对应于 **ISYSIDX** 系统表的主键，可使用 **SYSIDX** 系统视图查看此系统表。

每个锁都具有关联的表，因此，**LockTable** 的值可用于明确地确定某连接是否正在等待锁。

### 特权

为当前连接 **ID** 执行此系统过程不需要任何特权。要为其它连接执行此系统过程，您必须具有 **SERVER OPERATOR**、**MONITOR** 或 **DROP CONNECTION** 系统特权。

### 副作用

无

### 示例

以下示例使用 **sa\_conn\_info** 系统过程为连接到服务器的所有连接返回一个汇总连接属性的结果集。

```
CALL sa_conn_info( );
```

Number	Name	Userid	DBNumber	...
79	SQL_DBC_10dcf810	DBA	0	...
46	setup	User1	0	...
...	...	...	...	...

以下示例使用 **sa\_conn\_info** 系统过程返回结果集，其中显示是哪个连接创建了临时连接。

```
SELECT Number, Name, ParentConnection FROM sa_conn_info();
```

**Connection 8** 创建了执行 **CREATE DATABASE** 语句的临时连接。

```
Number      Name      ParentConnection
-----
```

```
1000000048 INT: CreateDB      8
9          SQL_DBC_14675af8 (NULL)
8          SQL_DBA_152d5ac0 (NULL)
```

**另请参见**

- 确定数据库所使用的安全模型 (第 381 页)

## sa\_conn\_list 系统过程

返回包含连接 ID 的结果集。

*语法*

```
sa_conn_list ([ connidparm ] [ ,dbidparm])
```

*参数*

- **connidparm** - 此可选 INTEGER 参数用于指定连接 ID 号。
- **dbidparm** - 此可选 INTEGER 参数用于指定数据库 ID 号。

*特权*

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种 系统特权:

- SERVER OPERATOR
- MONITOR
- DROP CONNECTION

*注释*

列名	数据类型	描述
数字	INTEGER	连接 ID 号。

如果 *connidparm* 大于零, 则返回所提供的连接的信息。如果 *connidparm* 小于零, 则返回当前连接的信息。如果 *connidparm* 和 *dbidparm* 未提供或为 NULL, 则将返回数据库服务器上运行的所有数据库的所有连接的连接 ID。

如果 *connidparm* 为 NULL, 并且 *dbidparm* 大于或等于零, 则只返回该数据库的连接 ID。如果 *connidparm* 为 NULL 且 *dbidparm* 小于零, 则只返回当前数据库的连接 ID。

*示例*

以下示例使用 **sa\_conn\_list** 系统过程显示连接 ID 列表。

```
CALL sa_conn_list( );
```

编号
1,949

编号
1,948
...

另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sa\_conn\_properties 系统过程

报告连接属性信息。

语法

```
sa_conn_properties( [ connidparm ] )
```

参数

- **connidparm** - 使用此可选的 INTEGER 参数指定连接 ID 号。缺省值为 NULL。

结果集

列名	数据类型	说明
Number	INTEGER	返回当前连接的连接 ID（数字）。
PropNum	INTEGER	返回连接属性编号。
PropName	VARCHAR(255)	返回连接属性名称。
PropDescription	VARCHAR(255)	返回连接属性说明。
值	LONG VARCHAR	返回连接属性值。

注释

以 Number 的形式返回连接 ID，并返回每个可用连接属性的 PropNum、PropName、PropDescription 和 Value。返回所有连接属性、与连接相关的数据库选项设置以及与连接相关的统计信息的值。同时返回具有 NULL 值的有效属性。

如果 *connidparm* 小于零，则返回当前连接的属性值。如果 *connidparm* 未提供或为 NULL，则返回当前数据库的所有连接的属性值。

特权

为当前连接 ID 执行此系统过程不需要任何特权。要为其它连接执行此系统过程，您必须具有 SERVER OPERATOR、MONITOR 或 DROP CONNECTION 系统特权。

副作用

无

### 示例

以下示例使用 `sa_conn_properties` 系统过程为所有连接返回一个汇总连接属性信息的结果集。

```
CALL sa_conn_properties ( );
```

Number	PropNum	PropName	...
79	37	ClientStmtCacheHits	...
79	38	ClientStmtCacheMisses	...
...	...	...	...

以下示例使用 `sa_conn_properties` 系统过程返回所有连接的列表，并按 CPU 时间降序排序\*：

```
SELECT Number AS connection_number,
       CONNECTION_PROPERTY ( 'Name', Number ) AS connection_name,
       CONNECTION_PROPERTY ( 'Userid', Number ) AS user_id,
       CAST ( Value AS NUMERIC ( 30, 2 ) ) AS approx_cpu_time
FROM sa_conn_properties ( )
WHERE PropName = 'ApproximateCPUTime'
ORDER BY approx_cpu_time DESC;
```

\*经 Breck Carter 允许使用的示例，引自 RisingRoad Professional Services (<http://www.risingroad.com>)。

### 另请参见

- 确定数据库所使用的安全模型 (第 381 页)

## sa\_db\_info 系统过程

报告数据库属性信息。

### 语法

```
sa_db_info ( [ dbidparm ] )
```

### 参数

- **dbidparm** – 使用此可选的 INTEGER 参数指定数据库 ID 号。缺省值为 NULL。

### 结果集

列名	数据类型	说明
Number	INTEGER	返回当前连接的连接 ID (数字)。
Alias	VARCHAR(255)	返回数据库名。

列名	数据类型	说明
文件	VARCHAR(255)	返回数据库根文件的文件名，包括路径。
ConnCount	INTEGER	返回数据库的连接数。属性值不包括用于内部操作的连接，但包含用于事件和外部环境支持的连接。
PageSize	INTEGER	返回数据库的页大小（以字节为单位）。
LogName	VARCHAR(255)	返回事务日志的文件名，包括路径。

### 注释

如果指定数据库 ID，则 `sa_db_info` 返回单个行，其中包含指定数据库的 `Number`、`Alias`、`File`、`ConnCount`、`PageSize` 和 `LogName`。

如果 `dbidparm` 大于零，则返回所提供的数据库的属性。如果 `dbidparm` 小于零，则返回当前数据库的属性。如果 `dbidparm` 未提供或为 `NULL`，则返回数据库服务器上运行的所有数据库的属性。

### 特权

为当前数据库执行此系统过程不需要任何特权。要为其它数据库执行此系统过程，您必须具有 `SERVER OPERATOR` 或 `MONITOR` 系统特权。

### 副作用

无

### 示例

以下语句为服务器上运行的每个数据库返回一行：

```
CALL sa_db_info( );
```

属性	值
Number	0
Alias	iqdemo
File	C:\ProgramData\SybaseIQ\demo \iqdemo.db
ConnCount	3
PageSize	4096
LogName	C:\ProgramData\SybaseIQ\demo \iqdemo.log

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sa\_db\_properties 系统过程

报告数据库属性信息。

### 语法

```
sa_db_properties ( [ dbidparm ] )
```

### 参数

- **dbidparm** – 使用此可选的 INTEGER 参数指定数据库 ID 号。缺省值为 NULL。

### 结果集

列名	数据类型	说明
Number	INTEGER	数据库 ID 号。
PropNum	INTEGER	数据库属性编号。
PropName	VARCHAR(255)	数据库属性名称。
PropDescription	VARCHAR(255)	数据库属性说明。
Value	LONG VARCHAR	数据库属性值。

### 注释

如果指定数据库 ID，则 sa\_db\_properties 系统过程返回数据库 ID 号和每个可用数据库属性的 PropNum、PropName、PropDescription 和 Value。返回所有数据库属性和与数据库相关的统计信息的值。同时返回具有 NULL 值的有效属性。

如果 dbidparm 大于零，则返回所提供的数据库的数据库属性。如果 dbidparm 小于零，则返回当前数据库的数据库属性。如果 dbidparm 未提供或为 NULL，则返回数据库服务器上运行的所有数据库的数据库属性。

### 特权

为当前数据库执行此系统过程不需要任何特权。要为其它数据库执行此系统过程，您必须具有 SERVER OPERATOR 或 MONITOR 系统特权。

### 副作用

无

### 示例

如果调用者具有 SERVER OPERATOR 或 MONITOR 系统特权，则以下示例使用 sa\_db\_properties 系统过程返回一个汇总了所有数据库的数据库属性的结果集。否则，将返回当前数据库的数据库属性。

```
CALL sa_db_properties ( );
```



Number	PropNum	PropName	...
0	0	ConnCount	...
0	1	IdleCheck	...
0	2	IdleWrite	...
...	...	...	...

以下示例使用 `sa_db_properties` 系统过程返回一个汇总了另一个数据库的数据库属性的结果集。

```
CALL sa_db_properties( 1 );
```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sa\_describe\_shapefile 系统过程

描述包含在 ESRI shapefile 中的列的名称和类型。此系统功能需与空间数据功能一起使用。

### 语法

```
sa_describe_shapefile ( shp_filename , srid [, encoding ] )
```

### 参数

- **shp\_filename** - 标识 ESRI shapefile 位置的 `VARCHAR(512)` 参数。文件名必须包含扩展名 `.shp`，并且在同一目录中还必须有与相同基本名相关联的 `.dbf` 文件。路径相对于数据库服务器，而不是客户端应用程序。
- **srid** - 标识 shapefile 中几何的 `SRID` 的 `INTEGER` 参数。指定 `NULL` 以表明该列可以存储多个 `SRID`。指定 `NULL` 限制可以在几何值上执行的操作。
- **encoding** - 可选的 `VARCHAR(50)` 参数，用于标识在读取 shapefile 时要使用的编码。缺省值为 `NULL`。如果编码为 `NULL`，则使用 `ISO-8859-1` 字符集。

### 特权

您必须具有系统过程的 `EXECUTE` 特权。此外：

- 如果 `-gl` 数据库选项设置为 `DBA`，您必须具有以下系统特权之一：
  - `ALTER ANY TABLE`
  - `ALTER ANY OBJECT`
  - `LOAD ANY TABLE`
  - `READ FILE`
- 如果 `-gl` 数据库选项设置为 `ALL`，则不需要任何其它系统特权。
- 如果 `-gl` 数据库选项设置为 `NONE`，您必须具有 `READ FILE` 系统特权。

## 注释

列名	数据类型	描述
column_number	INTEGER	此行描述的列的顺序位置，从 1 开始。
name	VARCHAR(128)	列的名称。
domain_name_with_size	VARCHAR(160)	数据类型名称，包括大小和精度（如 CREATE TABLE 或 CAST 函数中所用）。

`sa_describe_shapefile` 系统过程用于说明 ESRI shapefile 中的列的名称和类型。此信息可用于创建一个表，以便使用 `LOAD TABLE` 或 `INPUT` 语句从 shapefile 装载数据。或者，也可以通过为 `OPENSTRING...FORMAT SHAPEFILE` 指定 `WITH` 子句来使用该系统过程读取 shapefile。

## 示例

以下示例将显示一个字符串，该字符串用于创建存储 shapefile 数据的表：

```
BEGIN
  DECLARE create_cmd LONG VARCHAR;
  SELECT 'create table if not exists esri_load( record_number int
primary key, ' ||
        (SELECT list( name || ' ' || domain_name_with_size, ', '
ORDER BY column_number )
FROM sa_describe_shapefile( 'c:\\esri\\tgr36069trt00.shp',
1000004326 )
WHERE column_number > 1 ) || ' )'
INTO create_cmd;
SELECT create_cmd;
EXECUTE IMMEDIATE create_cmd;
END
```

可使用以下语句将 shapefile 数据装载到表中（条件是您具有 `LOAD ANY TABLE` 系统特权，并且 `-gl` 数据库选项未设置为 `NONE`）：

```
LOAD TABLE esri_load
USING FILE 'c:\\esri\\tgr36069trt00.shp'
FORMAT SHAPEFILE;
```

**sa\_dependent\_views 系统过程**

返回给定表或视图的所有相关视图的列表。

## 语法

```
sa_dependent_views(
[ tbl_name
[, owner_name ] ]
)
```

### 参数

- **tbl\_name** - 使用此可选的 CHAR(128) 参数指定表或视图的名称。缺省值为 NULL。
- **owner\_name** - 使用此可选的 CHAR(128) 参数指定 **tbl\_name** 的所有者。缺省值为 NULL。

### 结果集

列名	数据类型	说明
table_id	UNSIGNED INTEGER	表或视图的对象 ID。
dep_view_id	UNSIGNED INTEGER	相关视图的对象 ID。

### 注释

使用此过程获取表及其相关视图的 ID 列表。

如果现有的表均不满足为表和所有者名称指定的条件，则不会生成错误。以下条件同样适用：

- 如果 **owner** 和 **tbl\_name** 均为 NULL，则返回所有具有相关视图的表的相关信息。
- 如果 **tbl\_name** 为 NULL，但指定了 **owner**，则返回指定所有者拥有的所有表的相关信息。
- 如果指定了 **tbl\_name**，但 **owner** 为 NULL，则返回具有指定名称的所有表的相关信息。

### 特权

无

### 副作用

无

### 示例

在此示例中，**sa\_dependent\_views** 系统过程用来获取 **SalesOrders** 表的相关视图的 ID 列表。该过程为 **SalesOrders** 返回 **table\_id**，为相关视图 (**ViewSalesOrders**) 返回 **dep\_view\_id**。

```
CALL sa_dependent_views( 'SalesOrders' );
```

在以下示例中，在 **SELECT** 语句中使用 **sa\_dependent\_views** 系统过程以获取 **SalesOrders** 表的相关视图的名称列表。该过程返回 **ViewSalesOrders** 视图。

```
SELECT t.table_name FROM SYSTAB t,
sa_dependent_views( 'SalesOrders' ) v
WHERE t.table_id = v.dep_view_id;
```

## sa\_disable\_auditing\_type 系统过程

禁用指定事件的审计。

### 语法

```
sa_disable_auditing_type( types )
```

### 参数

- **类型** - 使用此 VARCHAR(128) 参数指定包含以下一个或多个值的以逗号分隔的字符串：
  - **all** - 禁用所有类型的审计。
  - **connect** - 禁用成功和失败的连接尝试的审计。
  - **connectFailed** - 禁用失败的连接尝试的审计。
  - **DDL** - 禁用 DDL 语句的审计。
  - **选项** - 禁用公共选项的审计。
  - **permission** - 禁用权限检查、用户检查和 SETUSER 语句的审计。
  - **permissionDenied** - 禁用失败的权限和用户检查审计。
  - **触发器** - 禁用响应触发事件的审计。

### 注释

可使用 sa\_disable\_auditing\_type 系统过程来禁用一个或多个类别信息的审计。

将此选项设置为 all 则禁用所有审计。也可以通过将 PUBLIC.auditing 选项设置为 Off 来禁用审计。

### 特权

您必须具有 SET ANY SECURITY OPTION 系统特权。

### 副作用

无

### 示例

禁用所有审计：

```
CALL sa_disable_auditing_type( 'all' );
```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sa\_disk\_free\_space 系统过程

报告事务日志、事务日志镜像和/或临时文件的可用空间信息。

### 语法

```
sa_disk_free_space ( [ p_dbpace_name ] )
```

### 参数

- **p\_dbpace\_name** - 使用此 VARCHAR(128) 参数指定事务日志文件、事务日志镜像文件或临时文件。缺省值为 NULL。

指定 **SYSTEM** 以获取有关主数据库文件的信息，指定 **TEMPORARY** 或 **TEMP** 以获取有关临时文件的信息，指定 **TRANSLOG** 以获取有关事务日志的信息，或指定 **TRANSLOGMIRROR** 以获取有关事务日志镜像的信息。

### 结果集

列名	数据类型	说明
dbspace_name	VARCHAR(128)	这可以是事务日志文件、事务日志镜像文件或临时文件。
free_space	UNSIGNED BIGINT	卷上的空闲字节数。
total_space	UNSIGNED BIGINT	驱动器的可用磁盘空间总量。

### 注释

如果未指定 **p\_dbpace\_name** 参数或该参数为 NULL，那么结果集中如果存在事务日志、事务日志镜像和临时文件，则每个文件占用一行。如果指定了 **p\_dbpace\_name**，则正好返回一行或零行（如果指定了 **log** 或 **mirror** 但是没有日志文件或镜像文件，则返回零行）。

### 特权

您必须具有 **MANAGE ANY DBSPACE** 系统特权。

### 副作用

无

### 示例

以下示例使用 **sa\_disk\_free\_space** 系统过程返回一个包含可用空间信息的结果集。

```
CALL sa_disk_free_space ( );
```

dbspace_name	free_space	total_space
system	10952101888	21410402304

dbspace_name	free_space	total_space
translog	10952101888	21410402304
temporary	10952101888	21410402304

另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sa\_enable\_auditing\_type 系统过程

启用审计并指定需要审计的事件。

语法

```
sa_enable_auditing_type( types )
```

参数

- **类型** - 使用此 VARCHAR(128) 参数指定包含以下一个或多个值的以逗号分隔的字符串：
  - **all** - 启用所有类型的审计。
  - **connect** - 启用成功和失败的连接尝试的审计。
  - **connectFailed** - 启用失败的连接尝试的审计。
  - **DDL** - 启用 DDL 语句审计。
  - **options** - 启用公共选项的审计。
  - **permission** - 启用权限检查、用户检查和 SETUSER 语句的审计。
  - **permissionDenied** - 启用失败的权限和用户检查审计。
  - **triggers** - 启用触发事件之后的审计。

注释

sa\_enable\_auditing\_type 与 PUBLIC.auditing 选项一同使用可启用对特定类型信息的审计。

如果将 PUBLIC.auditing 选项设置为 On，并且不指定要审计的信息的类型，则会采用缺省设置 (all)。在这种情况下，将记录所有类型的审计信息。

如果将 PUBLIC.auditing 选项设置为 On，并使用 sa\_disable\_auditing\_type 禁用所有类型的审计，则不记录任何审计信息。要想重新建立审计，必须使用 sa\_enable\_auditing\_type 指定想要审计的信息的类型。

如果将 PUBLIC.auditing 选项设置为 Off，则无论 sa\_enable\_auditing\_type 的设置如何，都不记录任何审计信息。

特权

您必须具有 SET ANY SECURITY OPTION 系统特权。

*副作用*

无

### 示例

若仅启用选项审计：

```
CALL sa_enable_auditing_type( 'options' );
```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## **sa\_eng\_properties** 系统过程

报告数据库服务器属性信息。

*语法*

```
sa_eng_properties( )
```

*结果集*

列名	数据类型	说明
PropNum	INTEGER	数据库服务器属性编号。
PropName	VARCHAR(255)	数据库服务器属性名称。
PropDescription	VARCHAR(255)	数据库服务器属性说明。
Value	LONG VARCHAR	数据库服务器属性值。

*注释*

返回每个可用服务器属性的 PropNum、PropName、PropDescription 和 Value。返回所有数据库服务器属性以及与数据库服务器相关的统计信息的值。

*特权*

无

*副作用*

无

### 示例

以下语句返回可用服务器属性的集合

```
CALL sa_eng_properties( );
```

PropNum	PropName	...
1	IdleWrite	...
2	IdleChkPt	...
...	...	...

## sa\_external\_library\_unload 系统过程

卸载外部库。

### 语法

```
sa_external_library_unload ( [ 'external-library' ] )
```

### 参数

- **external-library** - 此可选 LONG VARCHAR 参数用于指定要卸载的库的名称。如果未指定任何库，则将卸载所有未使用的外部库。

### 特权

您必须具有系统过程的 EXECUTE 特权。您还必须具有 MANAGE ANY EXTERNAL OBJECT 系统特权。

### 注释

如果指定了外部库，但它在使用中或还未装载，则会返回错误。如果未指定任何参数，则在找不到任何已装载的外部库时，会返回错误。

### 示例

卸载名为 myextlib.dll 的外部库：

```
CALL sa_external_library_unload( 'myextlib.dll' );
```

卸载当前未使用的所有库：

```
CALL sa_external_library_unload();
```

### 另请参见

- 确定数据库所使用的安全模型（第 381 页）

## sa\_flush\_cache 系统过程

清空数据库服务器高速缓存中的当前数据库的所有页。

### 语法

```
sa_flush_cache( )
```



*注释*

数据库管理员可以使用此过程来清空数据库服务器高速缓存中的当前数据库内容。这在测量性能以确保得到可重复的结果时很有用。

*特权*

您必须具有 **SERVER OPERATOR** 系统特权。

*副作用*

无

**示例**

以下示例清空数据库服务器高速缓存中当前数据库的所有页。

```
CALL sa_flush_cache( );
```

**另请参见**

- 确定数据库所使用的安全模型（第 381 页）

**sa\_get\_user\_status 系统过程**

用于确定用户的当前状态。

*语法*

```
sa_get_user_status( )
```

*结果集*

列名	数据类型	说明
user_id	UNSIGNED INTEGER	标识用户的唯一编号。
user_name	CHAR(128)	用户的名称。
connections	INTEGER	该用户当前建立的连接数。
failed_logins	UNSIGNED INTEGER	用户进行的登录失败重试次数。
last_login_time	TIMESTAMP	用户上次登录时的本地时间。
locked	TINYINT	指示用户帐户是否锁定。
reason_locked	LONG VARCHAR	帐户被锁定的原因。

列名	数据类型	说明
user_dn	CHAR(1024)	正在连接到 LDAP 服务器的用户 ID 的可分辨名称 (Distinguished Name, 简称 DN)。
user_dn_cached_at	TIMESTAMP	存储 DN 时的本地时间。
password_change_state	BIT	用于指示是否正在对双重口令进行更改的值 (0 = 否, 1 = 是)。缺省值为 0。
password_change_first_user	UNSIGNED INTEGER	如果用户设置了双重口令的第一部分, 则为该用户的 user_id; 否则为 NULL。
password_change_second_user	UNSIGNED INTEGER	如果用户设置了双重口令的第二部分, 则为该用户的 user_id; 否则为 NULL。
user_dn	CHAR(1024)	用户的可分辨名称 (DN)。
user_dn_cached_at	TIMESTAMP	找到可分辨名称时的日期和时间。

*注释*

此过程返回一个显示用户当前状态的结果集。除了基本用户信息外, 该过程还包括两列, 分别指示用户是否被锁定以及锁定原因。用户可能由于以下几种原因而被锁定: 由于策略、口令到期或失败重试次数过多而被锁定。

如果使用 LDAP 用户验证对用户进行验证, 则输出将包括该用户的可分辨名称以及该可分辨名称被找到之时的日期和时间。

*特权*

您可以查看有关您自己的信息, 不需要任何特权。要查看有关其他用户的信息, 则必须具有 **MANAGE ANY USER** 系统特权。

*副作用*

无

*示例*

以下示例使用 `sa_get_user_status` 系统过程返回数据库用户的状态。

```
CALL sa_get_user_status;
```

*另请参见*

- 确定数据库所使用的安全模型 (第 381 页)

**sa\_get\_ldapsrver\_status 系统过程**

确定 LDAP 服务器配置对象的当前状态。

*语法*

```
sa_get_ldapsrver_status()
```

**特权**

您必须具有系统过程的 EXECUTE 特权。

**注释**

列名	数据类型	说明
ldsrv_id	UNSIGNED BIGINT	LDAP 服务器配置对象的唯一标识符，同时也是主键，由登录策略用来引用 LDAP 服务器。
ldsrv_name	CHAR(128)	指派给 LDAP 服务器配置对象的名称。
ldsrv_state	CHAR(9)	LDAP 服务器的只读状态： 1 - RESET 2 - READY 3 - ACTIVE 4 - FAILED 5 - SUSPENDED 数字值存储在系统表中；相应的文本值在系统视图中显示。
ldsrv_last_state_change	TIMESTAMP	表示上一次更改状态的时间。无论 LDAP 服务器的本地时区是多少，该值都以协调通用时间 (UTC) 形式存储。

在执行检查点操作并且内存中的内容写入磁盘目录之前，查看 SYSLDAPSERVER 列的值。在对 LDAP 服务器对象执行检查点操作期间，因存在导致 LDAP 服务器对象状态发生更改的事件（如由于访问 LDAP 目录服务器失败而导致连接失败），目录列 ldsrv\_state 和 ldsrv\_last\_state\_change 的更新以异步方式执行。LDAP 服务器对象的状态反映了 LDAP 目录服务器的状态。

**sa\_list\_external\_library 系统过程**

列出服务器上当前装载的外部库。

**语法**

```
sa_list_external_library ( )
```

**特权**

您必须具有系统过程的 EXECUTE 特权。您还必须具有 MANAGE ANY EXTERNAL OBJECT 系统特权。

**注释**

返回引擎中装载的外部库的列表以及各个库的引用计数。

引用计数是引擎中库的实例数。可以通过执行过程 `sa_external_library_unload` 来卸载外部库（只要其引用计数是 0）。

*示例*

列出外部库及其引用计数：

```
CALL sa_list_external_library()
```

**另请参见**

- 确定数据库所使用的安全模型（第 381 页）

## **sa\_locks 系统过程**

显示数据库中的所有锁。

*语法*

```
sa_locks (
    [ connection
    [, creator
    [, table_name
    [, max_locks ] ] ] )
```

*参数*

- **connection** - 使用此 INTEGER 参数指定连接 ID 号。此过程仅返回指定连接的锁信息。缺省值为 0（或 NULL），此情况下返回有关所有连接的信息。
- **creator** - 使用此 CHAR(128) 参数指定用户 ID。该过程只返回有关指定用户所拥有的表的信息。creator 参数的缺省值为 NULL。当此参数设置为 NULL 时，sa\_locks 返回以下信息：
  - 如果未指定 table\_name 参数，则返回数据库中所有表的锁定信息
  - 如果指定了 table\_name 参数，则返回当前用户创建的具有指定名称的表的锁定信息
- **table\_name** - 使用此 CHAR(128) 参数指定表名。此过程仅返回指定表的信息。缺省值为 NULL，此情况下返回所有表的信息。
- **max\_locks** - 使用此 INTEGER 参数指定为其返回信息的锁的最大数目。缺省值为 1000。值 -1 表示返回所有锁信息

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

*注释*

列名	数据类型	描述
conn_name	VARCHAR(128)	当前连接的名称。

列名	数据类型	描述
conn_id	INTEGER	连接 ID 号。
user_id	CHAR(128)	连接的用户 ID。
table_type	CHAR(6)	表的类型。该类型为 <b>BASE</b> （对于表）、 <b>GLBTMP</b> （对于全局临时表）或 <b>MVIEW</b> （对于实例化视图）。
creator	VARCHAR(128)	表的所有者。
table_name	VARCHAR(128)	控制锁的表。
index_id	INTEGER	索引 ID 或 NULL。
lock_class	CHAR(8)	锁类。Schema、Row、Table 或 Position 之一。
lock_duration	CHAR(11)	锁的持续时间。Transaction、Position 或 Connection 之一。
lock_type	CHAR(9)	锁类型（这取决于锁类）。
row_identifier	UNSIGNED BIGINT	行的标识符。此为 8 字节行标识符或为 NULL。

sa\_locks 过程返回的结果集包含数据库中所有锁的信息。lock\_type 列中的值取决于 lock\_class 列中的锁分类。可以返回以下值：

锁类	锁类型	注释
模式	<ul style="list-style-type: none"> <li>已共享 - 共享模式锁。</li> <li>独占 - （仅 IQ 目录存储表）独占模式锁。</li> </ul>	对于模式锁，row_identifier 和索引 ID 值均为 NULL。

锁类	锁类型	注释
行	<ul style="list-style-type: none"> <li>• <b>读取</b> - 读锁定。</li> <li>• <b>意图</b> - 意图锁。</li> <li>• <b>ReadPK</b> - 读锁定。</li> <li>• <b>写入</b> - 写锁定。</li> <li>• <b>WriteNoPK</b> - 写锁定。</li> <li>• <b>代理</b> - 代理锁定。</li> </ul>	<p>行的读锁定可以是短期锁定（在隔离级别 1 的扫描），也可以是在更高隔离级别的长期锁定。<b>lock_duration</b> 列指明读锁定是由于游标稳定性而短期锁定 (Position)，还是一直保持到 COMMIT/ROLLBACK 的长期锁定 (Transaction)。行锁定始终保持在 8 字节行标识符作为 <b>row_identifier</b> 列中的 64 位整数值而被报告的特定行上。</p> <p>代理锁定是行锁定的一种特殊情况。代理锁定被保持在代理条目上，这些代理条目是在参照完整性检查延迟时创建的。对于表中所创建的每个代理条目，没有唯一的代理锁定。而是一个代理锁定对应于由给定连接为给定表所创建的一组代理条目。对于与代理锁定关联的表和连接，<b>row_identifier</b> 值是唯一的。</p> <p>如果需要，现在可单独锁定一行中的键值部分和非键值部分。连接可以获得共享（读）访问行的键值部分的读锁定，这样其它连接仍可获取该行的其它非键值列的写锁定。更新某行的非键值列时不会妨碍对引用该行的外行执行插入和删除操作。</p>
表	<ul style="list-style-type: none"> <li>• <b>已共享</b> - 共享表锁。</li> <li>• <b>意图</b> - 意图更新表锁。</li> <li>• <b>独占</b> - （仅 IQ 目录存储表）独占表锁。</li> </ul>	无
位置	<ul style="list-style-type: none"> <li>• <b>幻像</b> - （仅 IQ 目录存储表）幻像锁。</li> <li>• <b>插入</b> - 插入锁。</li> </ul>	位置锁通常也被保持在特定行上，且此行的 64 位行标识符出现在结果集的 <b>row_identifier</b> 列中。然而，位置锁还可以被保持在整个扫描上（索引扫描或顺序扫描），在此情况下， <b>row_identifier</b> 列为 NULL。

位置锁可以与顺序表扫描或索引扫描相关联。**index\_id** 列指明位置锁是否与顺序扫描相关联。如果由于顺序扫描而保持位置锁，则 **index\_id** 列为 NULL。如果由于特定的索引扫描而保持位置锁，则此索引的索引标识符在 **index\_id** 列中列出。该索引标识符对应于 **ISYSIDX** 系统表的主键，可使用 **SYSIDX** 视图查看此系统表。如果由于全部索引的扫描而保持位置锁，则索引 ID 值为 -1。

### 示例

可以执行以下查询来标识锁。

```
CALL sa_locks( );
```

## sa\_make\_object 系统过程

请确保在执行 **ALTER** 语句前存在对象的框架实例。

### 语法

```
sa_make_object(
  objtype
  , objname
  [, owner
  [, tabname ] ]
)
```

```
objtype:
'procedure'
| 'function'
| 'view'
| 'trigger'
| 'service'
| 'event'
```

### 参数

- **objtype** - 使用此 CHAR(30) 参数指定所创建对象的类型。如果 objtype 为 'trigger'，此参数指定要在其上创建触发器的表的所有者。
- **objname** - 使用此 CHAR(128) 参数指定要创建的对象名称。
- **owner** - 使用此可选的 CHAR(128) 参数指定要创建的对象的所有者。缺省值为 NULL。
- **tablename** - 仅在 objtype 为 'trigger' 时才需要此 CHAR(128) 参数，在此情况下，该参数用于指定要在其上创建触发器的表的名称。缺省值为 NULL。

### 注释

在那些需要重复运行以创建或修改数据库模式的脚本中，此过程很有用。这类脚本中的一个共同问题是：当这些脚本在第一次运行时，必须执行一个 **CREATE** 语句，以后运行时必须执行 **ALTER** 语句。使用此过程，就不必通过查询系统视图来查看对象是否存在。

对于过程、函数、视图和触发器，现在可以使用 **OR REPLACE** 子句代替此系统过程。

若要使用此过程，请在此过程后使用一个包含整个对象定义的 **ALTER** 语句。

### 特权

您必须具有所需的特权，如下所示：

- 归调用者所有的过程或函数 - CREATE PROCEDURE、CREATE ANY PROCEDURE 或 CREATE ANY OBJECT 系统特权
- 归其他用户所有的过程或函数 - CREATE ANY PROCEDURE 或 CREATE ANY OBJECT 系统特权
- 服务 - MANAGE ANY WEB SERVICE 系统特权
- 事件 - MANAGE ANY EVENT 或 CREATE ANY OBJECT 系统特权
- 归调用者所有的视图 - CREATE VIEW、CREATE ANY VIEW 或 CREATE ANY OBJECT 系统特权
- 归其他用户所有的视图 - CREATE ANY VIEW 或 CREATE ANY OBJECT 系统特权
- 触发器 - 如果触发器在归您所有的表中，那么您必须具有 CREATE ANY TRIGGER 或 CREATE ANY OBJECT 系统特权。

如果触发器在归其他用户所有的表中，那么您必须具有 CREATE ANY TRIGGER 或 CREATE ANY OBJECT 系统特权。此外，您必须具有以下特权之一：

- ALTER ANY TABLE 特权
- ALTER ANY OBJECT 系统特权
- 对要在其上创建触发器的表的 ALTER 权限。

#### 副作用

自动提交

#### 示例

以下语句确保创建框架过程定义，定义该过程，并授予对该过程的特权。可以对数据库重复运行包含这些指令的脚本文件而不会出现任何错误。

```
CALL sa_make_object( 'procedure', 'myproc' );
ALTER PROCEDURE myproc( in p1 INT, in p2 CHAR(30) )
BEGIN
    // ...
END;
GRANT EXECUTE ON myproc TO public;
```

以下示例使用 sa\_make\_object 系统过程添加一个框架 Web 服务。

```
CALL sa_make_object( 'service', 'my_web_service' );
```

## sa\_nchar\_terms 系统过程

将 NCHAR 字符串分解为多个术语，并以行返回每个术语及其位置。

#### 语法

```
sa_nchar_terms( 'char-string' [ , 'text-config-  
name' [ , 'owner' ] ] )
```



### 参数

- **char-string** - 所分析的 NCHAR 字符串。
- **text-config-name** - 处理字符串时应用的文本配置对象。缺省值为 'default\_nchar'。
- **owner** - 指定的文本配置对象的所有者。缺省值为 DBA。

### 特权

您必须具有系统过程的 EXECUTE 特权。

### 注释

可以使用 **sa\_nchar\_terms** 来了解在应用文本配置对象的设置时如何解释字符串。如果想知道索引过程中哪些术语会被删除或者会从查询字符串中删除哪些术语，这一点很有用。

**sa\_nchar\_terms** 的语法与 **sa\_char\_terms** 系统过程的语法类似。

---

**注意：** 仅 IN SYSTEM 表支持 NCHAR 数据类型。

---

## sa\_performance\_diagnostics 系统过程

返回所有连接的请求计时信息的汇总（前提是数据库服务器启用了请求计时记录）。

### 语法

```
sa_performance_diagnostics ( )
```

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

### 注释

列名	数据类型	描述
Number	INTEGER	返回当前连接的连接 ID（数字）。

列名	数据类型	描述
Name	VARCHAR(255)	<p>返回当前连接的名称。</p> <p>可使用 ConnectionName (CON) 连接参数来指定连接的名称。</p> <p>以下名称用于数据库服务器建立的临时连接：</p> <ul style="list-style-type: none"> <li>• INT:ApplyRecovery</li> <li>• INT:BackupDB</li> <li>• INT:Checkpoint</li> <li>• INT:Cleaner</li> <li>• INT:CloseDB</li> <li>• INT&gt;CreateDB</li> <li>• INT&gt;CreateMirror</li> <li>• INT:DelayedCommit</li> <li>• INT:DiagRcvr</li> <li>• INT:DropDB</li> <li>• INT:EncryptDB</li> <li>• INT:Exchange</li> <li>• INT:FlushMirrorLog</li> <li>• INT:FlushStats</li> <li>• INT:HTTPReq</li> <li>• INT:PromoteMirror</li> <li>• INT:PurgeSnapshot</li> <li>• INT:ReconnectMirror</li> <li>• INT:RecoverMirror</li> <li>• INT:RedoCheckpoint</li> <li>• INT:RefreshIndex</li> <li>• INT:ReloadTrigger</li> <li>• INT:RenameMirror</li> <li>• INT:RestoreDB</li> <li>• INT:StartDB</li> <li>• INT:VSS</li> </ul>
Userid	VARCHAR(255)	返回连接的用户 ID。
DBNumber	INTEGER	返回数据库的 ID 编号。
LoginTime	TIMESTAMP	返回建立连接的日期和时间。

列名	数据类型	描述
TransactionStartTime	TIMESTAMP	返回一个字符串，其中包含在 COMMIT 或 ROLLBACK 之后首次修改数据库的时间；如果自上次执行 COMMIT 或 ROLLBACK 以来未对数据库进行过任何修改，则返回一个空字符串。
LastReqTime	TIMESTAMP	返回指定连接的最后一个请求开始的时间。此属性可以为内部连接（如事件）返回空字符串。
ReqType	VARCHAR(255)	返回上次请求的类型。如果连接已由连接池进行了高速缓存，则其 ReqType 值为 CONNECT_POOL_CACHE。
ReqStatus	VARCHAR(255)	<p>返回请求状态。它可以是以下各值之一：</p> <ul style="list-style-type: none"> <li>• <b>Idle</b> - 连接当前未处理请求。</li> <li>• <b>Unscheduled</b> - 连接有工作要做并且正在等待可用的数据库服务器工作线程。</li> <li>• <b>BlockedIO</b> - 连接被阻塞，正在等待 I/O。</li> <li>• <b>BlockedContention</b> - 连接被阻塞，正在等待访问共享数据库服务器数据结构。</li> <li>• <b>BlockedLock</b> - 连接被阻塞，正在等待锁定的对象。</li> <li>• <b>Executing</b> - 连接正在执行请求。</li> </ul> <p>仅当已使用 -zt 服务器选项为数据库服务器开启了请求计时信息记录功能时，才会返回标有星号 (*) 的值。如果未记录请求计时信息（缺省设置），则会将这些值报告为 Executing。</p>
ReqTimeUnscheduled	DOUBLE	返回未调度的时间长度；如果未指定 -zt 选项，则返回 NULL。
ReqTimeActive	DOUBLE	返回处理请求所用的时间长度（以秒为单位）；如果未指定 -zt 选项，则返回 NULL。
ReqTimeBlockIO	DOUBLE	返回等待 I/O 完成所用的时间长度（以秒为单位）；如果未指定 -zt 选项，则返回 NULL。
ReqTimeBlockLock	DOUBLE	返回等待锁所用的时间长度（以秒为单位）；如果未指定 -zt 选项，则返回 NULL。
ReqTimeBlockContention	DOUBLE	返回等待原子访问所用的时间长度（以秒为单位）；如果将 RequestTiming 服务器属性设置为 Off，则返回 NULL。

列名	数据类型	描述
ReqCountUnscheduled	INTEGER	返回连接等待调度的次数；如果未指定 -zt 选项，则返回 NULL。
ReqCountActive	INTEGER	返回已处理的请求数；如果将 RequestTiming 服务器属性设置为 Off，则返回 NULL。
ReqCountBlockIO	INTEGER	返回连接等待 I/O 完成的次数；如果未指定 -zt 选项，则返回 NULL。
ReqCountBlockLock	INTEGER	返回连接等待锁的次数；如果未指定 -zt 选项，则返回 NULL。
ReqCountBlockContention	INTEGER	返回连接等待原子访问的次数；如果未指定 -zt 选项，则返回 NULL。
LastIdle	INTEGER	返回请求间隔时间。
BlockedOn	INTEGER	如果当前连接未被阻塞，则返回零；如果当前连接被阻塞，则返回由于锁定冲突而阻塞连接的连接号。
UncommitOp	INTEGER	返回未提交的操作数。
CurrentProcedure	VARCHAR(255)	返回连接当前正在执行的过程的名称。如果该连接正在执行嵌套过程调用，则该名称为当前过程的名称。如果未在执行任何过程，则返回一个空字符串。
EventName	VARCHAR(255)	如果连接正在运行事件处理程序，则返回相关事件的名称。否则，返回一个空字符串。
CurrentLineNumber	INTEGER	返回连接正在执行的过程或复合语句的当前行号。可使用 CurrentProcedure 属性标识该过程。如果该行是来自客户端的复合语句的一部分，则返回一个空字符串。

列名	数据类型	描述
LastStatement	LONG VARCHAR	<p>返回最近为当前连接准备的 SQL 语句。</p> <p>当准备了语句时，将会设置 LastStatement 值；当删除了语句时，将会清除该值。每个连接仅会记忆一个语句字符串。</p> <p>如果 sa_conn_activity 对某个连接报告了一个非空值，它很可能为该连接当前执行的语句。假如该语句已完成，则很有可能已将其删除并清除了属性值。如果某个应用程序准备了多条语句并保有其语句句柄，则 LastStatement 值不反映连接当前进行的操作。</p> <p>如果启用客户端语句高速缓存并且重用了某个高速缓存的语句，则此属性返回空字符串。</p>
LastPlanText	LONG VARCHAR	<p>返回在连接上执行的最后一个查询的长文本计划。通过设置 sa_server_option 系统过程的 RememberLastPlan 选项或使用 -zp 服务器选项控制对最后一个计划的记忆情况。</p>
AppInfo	LONG VARCHAR	<p>返回有关建立连接的客户端的信息。对于 HTTP 连接，这包括有关浏览器的信息。对于使用较旧版本 jConnect 或 Sybase Open Client 的连接，信息可能不完整。</p> <p>API 值可为 DBLIB、ODBC、OLEDB、ADO.NET、iAnywhereJDBC、PHP、PerlDBD 或 DBEXPRESS。</p>
LockCount	INTEGER	返回连接所持有的锁的个数。
SnapshotCount	INTEGER	返回与连接相关联的快照数。

**sa\_performance\_diagnostics** 系统过程返回一个包含一组请求计时属性和统计信息的结果集（前提是已通知服务器收集这些信息）。在调用 **sa\_performance\_diagnostics** 之前，必须在数据库服务器上打开请求计时信息的记录。要达到此目的，可在启动数据库服务器时指定 -zt 选项，也可执行以下内容：

```
CALL sa_server_option( 'RequestTiming','ON' );
```

#### 示例

可以执行以下查询以标识花费长时间等待数据库服务器请求完成的连接。

```
SELECT Number, Name,
       CAST( DATEDIFF( second, LoginTime, CURRENT_TIMESTAMP ) AS
       DOUBLE ) AS T,
       IF T <> 0 THEN (ReqTimeActive / T) ELSE NULL ENDIF AS
PercentActive
FROM sa_performance_diagnostics()
```

```
WHERE T > 0 AND PercentActive > 10.0
ORDER BY PercentActive DESC;
```

以下示例查找当前正在执行且执行时间已超过 60 秒的所有请求：

```
SELECT Number, Name,
       CAST( DATEDIFF( second, LastReqTime, CURRENT_TIMESTAMP ) AS
       DOUBLE ) AS ReqTime
FROM sa_performance_diagnostics()
WHERE ReqStatus <> 'IDLE' AND ReqTime > 60.0
ORDER BY ReqTime DESC;
```

## sa\_procedure\_profile\_summary 系统过程

报告已经在数据库中执行的所有过程、函数、事件或触发器的执行时间的汇总信息。

### 语法

```
sa_procedure_profile_summary (
    [ filename
    [, save_to_file ] ] )
```

### 参数

- **filename** - 使用此可选的 **LONG VARCHAR** 参数指定应将分析信息保存到的文件，或应从其装载分析信息的文件。缺省值为 **NULL**。有关保存和装载分析信息的详细信息，请参见以下的注释部分。
- **save-to-file** - 使用此可选的 **INTEGER** 参数指定是将汇总信息保存到文件，还是从先前保存的文件装载汇总信息。缺省值为 **0**。

### 结果集

列名	数据类型	描述
object_type	CHAR(1)	对象的类型。有关可能的对象类型的列表，请参见以下的注释部分。
object_name	CHAR(128)	存储过程、函数、事件或触发器的名称。
owner_name	CHAR(128)	对象的所有者。
table_name	CHAR(128)	与触发器关联的表（对于其它对象类型，该值为 <b>NULL</b> ）。
executions	UNSIGNED INTEGER	每个过程的执行次数。
millisecs	UNSIGNED INTEGER	过程的执行时间，以毫秒计。
foreign_owner	CHAR(128)	拥有系统触发器外表的数据库用户。

列名	数据类型	描述
foreign_table	CHAR(128)	系统触发器外表的名称。

#### 注释

此过程可用于：

- **返回当前汇总信息** - 要实现此目的，可以只调用该过程，而不指定任何参数。
- **将当前汇总信息保存到文件** - 要实现此目的，必须包括 *filename* 参数并将 *save\_to\_file* 参数指定为 1。
- **从文件装载存储的汇总信息** - 要实现此目的，必须包括 *filename* 参数并将 *save\_to\_file* 参数指定为 0。当以此方式使用该过程时，创建装载文件的数据库必须是正从其运行该过程的同一数据库；否则，结果可能不可用。

由于此过程返回有关存储过程、函数、事件和触发器的使用频率和效率的信息，因此您可以使用此信息来调整较慢的过程以提高数据库性能。

必须先启用分析，之后才能分析数据库。

结果集的 *object\_type* 列可以是：

- **P** - 存储过程
- **F** - 函数
- **E** - event
- **T** - trigger
- **C** - ON UPDATE 系统触发器
- **D** - ON DELETE 系统触发器

如果想得到每个执行的逐行详细信息（而不是汇总信息），请改用 **sa\_procedure\_profile** 过程。

#### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 或 MANAGE PROFILING 系统特权。

您还必须具有以下特权：

- SELECT ANY TABLE（当 *filename* 不为 NULL 且 *save\_to\_file* 为 1 时）
- LOAD ANY TABLE（当 *filename* 不为 NULL 且 *save\_to\_file* 为 0 时）

#### 示例

以下语句返回已经在数据库中执行的所有过程、函数、事件或触发器的执行时间：

```
CALL sa_procedure_profile_summary( );
```

以下语句返回与以上示例相同的汇总信息，并将其保存到名为 *summaryinfo.txt* 的文件：

```
CALL sa_procedure_profile_summary( 'summaryinfo.txt', 1 );
```

以下任一语句均可用来从名为 `summaryinfo.txt` 的文件装载存储的汇总信息：

```
CALL sa_procedure_profile_summary( 'summaryinfo'.txt, 0 );
CALL sa_procedure_profile_summary( 'summaryinfo.txt' );
```

## sa\_procedure\_profile 系统过程

报告已经在数据库中执行的过程、函数、事件或触发器中每一行的执行时间信息。

### 语法

```
sa_procedure_profile (
    [ filename
    [, save_to_file ] ] )
```

### 参数

- **filename** - 使用此可选的 **LONG VARCHAR** 参数指定应将分析信息保存到的文件，或应从其装载分析信息的文件。缺省值为 **NULL**。有关保存和装载分析信息的详细信息，请参见以下的注释部分。
- **save\_to\_file** - 使用此可选的 **INTEGER** 参数指定是将分析信息保存到文件，还是从先前存储的文件装载分析信息。缺省值为 **0**。

### 结果集

列名	数据类型	描述
object_type	CHAR(1)	对象的类型。有关可能的对象类型的列表，请参见以下的注释部分。
object_name	CHAR(128)	存储过程、函数、事件或触发器的名称。如果 <b>object_type</b> 为 <b>C</b> 或 <b>D</b> ，则这是为其定义了系统触发器的外键的名称。
owner_name	CHAR(128)	对象的所有者。
table_name	CHAR(128)	与触发器关联的表（对于其它对象类型，该值为 <b>NULL</b> ）。
line_num	UNSIGNED INTEGER	过程中的行号。
executions	UNSIGNED INTEGER	该行的执行次数。
millisecs	UNSIGNED INTEGER	该行的执行时间（以毫秒计）。
percentage	DOUBLE	特定行所需要的总执行时间的百分比。
foreign_owner	CHAR(128)	拥有系统触发器外表的数据库用户。



列名	数据类型	描述
foreign_table	CHAR(128)	系统触发器外表的名称。

*注释*

此过程可用于：

- **返回详细的过程分析信息** – 要实现此目的，可以只调用该过程，而不指定任何参数。
- **将详细的过程分析信息保存到文件** – 要实现此目的，必须包括 *filename* 参数并将 *save\_to\_file* 参数指定为 1。
- **从先前保存的文件装载详细的过程分析信息** – 要实现此目的，必须包括 *filename* 参数并将 *save\_to\_file* 参数指定为 0。当以此方式使用该过程时，创建装载文件的数据库必须是正从其运行该过程的同一数据库；否则，结果可能不可用。

结果集中包含过程、触发器、函数和事件中每一行的执行时间信息，以及这些行使用的时间占总过程执行时间的百分比信息，因此可以使用此分析信息来调整那些可能会降低性能的较慢过程。

必须先启用分析，之后才能分析数据库。

结果集的 *object\_type* 列可以是：

- **P** – 存储过程
- **F** – 函数
- **E** – event
- **T** – trigger
- **C** – ON UPDATE 系统触发器
- **D** – ON DELETE 系统触发器

如果想得到汇总信息（而不是每个执行的逐行详细信息），请改用 **sa\_procedure\_profile\_summary procedure** 过程。

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 或 MANAGE PROFILING 系统特权。

您还必须具有以下特权：

- SELECT ANY TABLE（当 *filename* 不为 NULL 且 *save\_to\_file* 为 1 时）
- LOAD ANY TABLE（当 *filename* 不为 NULL 且 *save\_to\_file* 为 0 时）

*示例*

以下语句返回已经在数据库中执行的每个过程、函数、事件或触发器中每一行的执行时间：

```
CALL sa_procedure_profile( );
```

以下语句返回与以上示例相同的详细过程分析信息，并将其保存到名为 detailedinfo.txt 的文件：

```
CALL sa_procedure_profile( 'detailedinfo.txt', 1 );
```

以下任一语句均可用来从名为 detailedinfo.txt 的文件装载详细过程分析信息：

```
CALL sa_procedure_profile( 'detailedinfo.txt', 0 );
CALL sa_procedure_profile( 'detailedinfo.txt' );
```

## **sa\_report\_deadlocks** 系统过程

从数据库服务器创建的内部缓冲区中检索有关死锁的信息。

*语法*

**sa\_report\_deadlocks()**

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

*注释*

列名	数据类型	描述
snapshotId	BIGINT	死锁实例（与特定死锁相关的所有行都具有相同的 ID）。
snapshotAt	TIMESTAMP	发生死锁的时间。
waiter	INT	正等待的连接的连接句柄。
who	VARCHAR(128)	与正等待的连接关联的用户 ID。
what	LONG VARCHAR	正等待的连接所执行的命令。 仅当通过在数据库服务器命令行中指定 -zl 选项打开捕获最新准备的 SQL 语句的功能时，此信息才可用。
object_id	UNSIGNED BIGINT	包含该行的表的对象 ID。
record_id	BIGINT	系统表的行 ID。
owner	INT	拥有正等待的锁的连接的连接句柄。
is_victim	BIT	标识回退的事务。
rollback_operation_count	UNSIGNED INT	未提交操作的数量，如果事务回退，这些操作有可能丢失。
iq_rid	UNSIGNED BIGINT	启用 IQ RLV 的表的行 ID。

列名	数据类型	描述
iq_txn_id	UNSIGNED BIGINT	关联行的事务 ID。

当 `log_deadlocks` 选项设置为 `On` 时，数据库服务器在内部缓冲区中记录有关死锁的信息。可以使用 `sa_report_deadlocks` 系统过程查看日志中的信息。

*副作用*

无。

## **sa\_rowgenerator** 系统过程

返回包含指定起始值和结束值之间的行的结果集。

*语法*

```
sa_rowgenerator (
  [ rstart
  [, rend
  [, rstep ] ] ]
)
```

*参数*

- **rstart** - 使用此可选的 `INTEGER` 参数指定起始值。缺省值是 0。
- **rend** - 使用此可选的 `INTEGER` 参数指定大于或等于 **rstart** 的结束值。缺省值是 100。
- **rstep** - 使用此可选的 `INTEGER` 参数指定序列值的增量。缺省值是 1。

*结果集*

列名	数据类型	说明
row_num	INTEGER	序列号。

*注释*

此 `sa_rowgenerator` 过程可用在查询的 `FROM` 子句中，以生成一个数字序列。此过程可以作为使用 `RowGenerator` 系统表的替代方法。可以为类似如下的任务使用 `sa_rowgenerator`：

- 在结果集中为一定数目的行生成测试数据。
- 生成一个结果集，其中包含为每个范围中的值生成的行。例如，可以为一个月的每一天生成一行，也可以生成邮政编码范围。
- 生成一个查询，其结果集具有指定数目的行。这对于测试查询性能可能有用。

如果不指定正确的起始值、结束值和非零正数步骤值，则不会返回任何行。

您可以使用以下语句模拟 `RowGenerator` 表的行为：

```
SELECT row_num FROM sa_rowgenerator( 1, 255 );
```

特权  
无

副作用  
无

### 示例

以下查询返回的结果集包含当前月的每一天的一行记录。

```
SELECT DATEADD( day, row_num-1,
               YMD( DATEPART( year, CURRENT DATE ),
                   DATEPART( month, CURRENT DATE ), 1 ) )
       AS day_of_month
FROM sa_rowgenerator( 1, 31, 1 )
WHERE DATEPART( month, day_of_month ) = DATEPART( month, CURRENT
DATE )
ORDER BY row_num;
```

以下查询显示有多少雇员生活在邮政编码范围为 (0-9999)、(10000-19999)、...、及 (90000-99999) 的地方。其中一些范围不含雇员，这时会出现警告。

`sa_rowgenerator` 过程可以用来生成这些范围，即使该邮政编码范围没有任何雇员。

```
SELECT row_num AS r1, row_num+9999 AS r2, COUNT( PostalCode ) AS
       zips_in_range
FROM sa_rowgenerator( 0, 99999, 10000 ) D LEFT JOIN Employees
       ON PostalCode BETWEEN r1 AND r2
GROUP BY r1, r2
ORDER BY 1;
```

以下示例生成 10 行数据并将其插入到 `NewEmployees` 表中：

```
INSERT INTO NewEmployees ( ID, Salary, Name )
       SELECT row_num, CAST( RAND() * 1000 AS INTEGER ), 'Mary'
FROM sa_rowgenerator( 1, 10 );
```

以下示例使用 `sa_rowgenerator` 系统过程来创建包含所有整数的视图。此示例中的值 2147483647 表示所支持的最大有符号整数。

```
CREATE VIEW Integers AS
       SELECT row_num AS n
FROM sa_rowgenerator( 0, 2147483647, 1 );
```

以下示例使用 `sa_rowgenerator` 系统过程来创建包含日期从 0001-01-01 到 9999-12-31 的视图。此示例中的值 3652058 表示 0001-01-01 和 9999-12-31 之间的天数，其中 0001-01-01 和 9999-12-31 分别为 SAP Sybase IQ 中所支持的最早和最晚的日期。

```
CREATE VIEW Dates AS
       SELECT DATEADD( day, row_num, '0001-01-01' ) AS d
FROM sa_rowgenerator( 0, 3652058, 1 );
```

以下查询将返回 1900 至 2058 年之间包含 54 周的所有年份。

```
SELECT DATEADD ( day, row_num, '1900-01-01' ) AS d, DATEPART ( week,
d ) w
FROM sa_rowgenerator ( 0, 63919, 1 )
WHERE w = 54;
```

## sa\_server\_option 系统过程

当服务器正在运行时替换服务器选项。

### 语法

```
sa_server_option( opt , val )
```

### 参数

- **opt** - 使用此 CHAR(128) 参数指定服务器选项名称。
- **val** - 使用此 CHAR(128) 参数为服务器选项指定新值。

### 特权

您必须具有系统过程的 EXECUTE 特权。您还必须具有 以下选项与应用程序分析或请求记录相关，要使用这些选项，您必须具有 MANAGE PROFILING 系统特权：

- ProcedureProfiling
- ProfileFilterConn
- ProfileFilterUser
- RequestFilterConn
- RequestFilterDB
- RequestLogFile
- RequestLogging
- RequestLogMaxSize
- RequestLogNumFiles

### 注释

选项名称	值	其它信息
AutoMultiProgrammingLevel	YES、NO	缺省值为 YES。 设置为 YES 时，数据库服务器会自动调整其进程并发水平，该属性控制可以同时处于活动状态的最大任务数。如果通过将该选项设置为 NO 来选择手动控制进程并发水平，则还可以设置进程并发水平的初始值、最小值和最大值。
AutoMultiProgrammingLevelStatistics	YES、NO	缺省值为 NO。 设置为 YES 时，进程并发水平自动调整的统计信息将显示在数据库服务器消息日志中。

选项名称	值	其它信息
CacheSizingStatistics	YES、NO	缺省值为 NO。 当设置为 YES 时，每当高速缓存大小发生变化时，数据库服务器消息窗口中就会显示高速缓存信息。
CollectStatistics	YES、NO	缺省值为 YES。 当设置为 YES 时，数据库服务器收集性能监控器统计信息。
ConnsDisabled	YES、NO	缺省值为 NO。 当设置为 YES 时，不允许与数据库服务器上的任何数据库建立任何其它连接。
ConnsDisabled-ForDB	YES、NO	缺省值为 NO。 当设置为 YES 时，不允许与当前数据库建立任何其它连接。
ConsoleLogFile	filename	用来记录数据库服务器消息日志信息的文件的名称。指定一个空字符串会停止将信息记录到文件。由于此值为 SQL 字符串，所以路径中的所有反斜线字符均需写两遍。
ConsoleLogMaxSize	file-size (以字节为单位)	用来记录数据库服务器消息日志信息的文件的最大大小 (以字节为单位)。当数据库服务器消息日志文件达到该属性或 -on 服务器选项指定的大小后，通过附加扩展名 .old 重命名该文件 (如果已经存在具有相同名称的文件，则将其替换)。然后，重新启动该数据库服务器消息日志文件。
CurrentMultiProgrammingLevel	整数	缺省值为 20。 设置数据库服务器的进程并发水平。
DatabaseCleaner	ON、OFF	缺省值为 ON。 除非有技术支持部门的建议，否则不要更改此选项的设置。

选项名称	值	其它信息
DeadlockLogging	ON、OFF、 RESET、CLEAR	<p>缺省值为 OFF。</p> <p>控制死锁记录。同时支持值 <code>deadlock_logging</code>。支持以下值：</p> <ul style="list-style-type: none"> <li>• <b>ON</b> - 启用死锁记录。</li> <li>• <b>OFF</b> - 禁用死锁记录，但使死锁数据可供查看。</li> <li>• <b>RESET</b> - 清除记录的死锁数据（如果存在的话），然后启用死锁记录。</li> <li>• <b>CLEAR</b> - 清除记录的死锁数据（如果存在的话），然后禁用死锁记录。</li> </ul> <p>启用死锁记录后，可以使用 <code>sa_report_deadlocks</code> 系统过程从数据库中检索死锁信息。</p>
DebuggingInformation	YES、NO	<p>缺省值为 NO。</p> <p>显示诊断消息和其它消息以用于疑难解答。消息显示在数据库服务器消息窗口中。</p>
DiskSandbox	ON、OFF	<p>缺省值为 OFF。</p> <p>为在数据库服务器上启动的所有没有显式磁盘沙箱设置的数据库设置缺省磁盘沙箱设置。使用 <code>sa_server_option</code> 系统过程更改磁盘沙箱设置并不会影响数据库服务器上已运行的数据库。要使用 <code>sa_server_option</code> 系统过程更改磁盘沙箱设置，必须为 <code>manage_disk_sandbox</code> 安全功能提供安全功能密钥。</p>
DropBadStatistics	YES、NO	<p>缺省值为 YES。</p> <p>允许通过自动统计信息管理删除返回数据库错误估计的统计信息。</p>
DropUnusedStatistics	YES、NO	<p>缺省值为 YES。</p> <p>允许通过自动统计信息管理删除数据库中连续 90 天尚未使用的统计信息。</p>
IdleTimeout	整数（以分钟为单位）	<p>缺省值为 240。</p> <p>断开未在指定分钟内提交请求的 TCP/IP 连接。这可防止不活动的连接无限期地持有锁。</p>

选项名称	值	其它信息
IPAddressMonitor-Period	整数（以秒为单位）	最小值为 10，缺省值为 0。对于便携式设备，缺省值为 120。 设置检查新 IP 地址的时间（以秒为单位）。
LivenessTimeout	整数（以秒为单位）	缺省值为 120。 定期通过客户端/服务器 TCP/IP 网络发送活动包，用以确认连接的完整性。如果网络服务器运行了一个 LivenessTimeout 周期但没有检测到活动包，则切断通信。
MaxMultiProgrammingLevel	整数	缺省值是 CurrentMultiProgrammingLevel 值的四倍。 设置数据库服务器进程并发水平的最大值。
MessageCategoryLimit	整数	缺省值为 400。 设置可使用 sa_server_messages 系统过程检索的各个严重级和类别的最小消息数。
MinMultiProgrammingLevel	整数	缺省值是 -gtc 服务器选项值和计算机上逻辑 CPU 数量这二者中的最小值。
OptionWatchAction	MESSAGE、ERROR	缺省值为 MESSAGE。 指定数据库服务器在尝试设置列表中的选项时所采取的操作。如果 OptionWatchAction 设置为 MESSAGE，并且设置了一个由 OptionWatchList 指定的选项，数据库服务器消息窗口中将显示一条消息，指示正在设置的选项位于选项监视列表中。如果 OptionWatchAction 设置为 ERROR，将返回一条错误，指示由于该选项位于选项监视列表中，从而无法设置该选项。  可通过执行以下语句来查看此属性的当前设置： <pre>SELECT DB_PROPERTY( 'OptionWatchAction' );</pre>



选项名称	值	其它信息
OptionWatchList	以逗号分隔的数据库选项列表	<p>指定一个以逗号分隔的数据库选项列表，在设置其中的选项时数据库服务器会向您发出通知，或返回错误。字符串的长度不应超过 128 个字节。缺省情况下它是一个空字符串。例如，以下命令将 <code>automatic_timestamp</code>、<code>float_as_double</code> 和 <code>tsql_hex_constant</code> 选项添加到监视选项列表中。</p> <pre>CALL sa_server_option( 'OptionWatchList','automatic_timestamp, float_as_double,tsql_hex_constant' );</pre> <p>可通过执行以下语句来查看此属性的当前设置：</p> <pre>SELECT DB_PROPERTY( 'OptionWatchList' );</pre>
ProcedureProfiling	YES、NO、RESET、CLEAR	缺省值为 NO。
ProfileFilterConn	connection-id	指示数据库服务器捕获指定连接 ID 的分析信息，而不阻止其它连接使用该数据库。启用连接过滤后， <code>SELECT PROPERTY('ProfileFilterConn')</code> 返回的值是被监控连接的连接 ID。如果未指定 ID 或禁用连接过滤，则返回的值为 -1。
ProcessorAffinity	以逗号分隔的处理器编号和/或范围列表。缺省设置是使用所有处理器或 <code>-gta</code> 选项的设置。	<p>指示数据库服务器在 Windows 或 Linux 上使用哪些逻辑处理器。指定以逗号分隔的处理器编号和/或范围列表。如果省略范围下限，则假定为零。如果省略范围上限，则假定为操作系统已知的最高 CPU。 <code>sa_cpu_topology</code> 系统过程返回的 <code>in_use</code> 列中包含数据库服务器的当前处理器关联，<code>in_use</code> 列还指示数据库服务器是否正在使用处理器。另外，也可以查询 <code>ProcessorAffinity</code> 数据库服务器属性的值。</p> <p>在以下情况下数据库服务器可能不会使用所有指定的逻辑处理器：</p> <ul style="list-style-type: none"> <li>如果一个或多个指定的逻辑处理器不存在或处于脱机状态。</li> <li>如果许可证不允许。</li> </ul> <p>如果指定的处理器 ID 无效，则 <code>sa_server_option</code> 返回一条错误。</p>
ProfileFilterUser	user-id	指示数据库服务器捕获特定用户 ID 的分析信息。

选项名称	值	其它信息
QuittingTime	有效的日期和时间	指示数据库服务器在指定的时间关闭。
RememberLastPlan	YES、NO	<p>缺省值为 NO。</p> <p>指示数据库服务器捕获在连接上执行的最后一个查询的详细文本计划。此设置也由 <code>-zp</code> 服务器选项控制。启用 <code>RememberLastPlan</code> 时，通过查询 <code>LastPlanText</code> 连接属性的值来获取在连接上执行的最后一个查询计划的文本表示：</p> <pre>SELECT CONNECTION_PROPERTY( 'Last-PlanText' );</pre>
RememberLastStatement	YES、NO	<p>缺省值为 NO。</p> <p>指示数据库服务器为在服务器上运行的每个数据库捕获最近预准备 SQL 语句。对于存储过程调用，仅显示最外面的过程调用，不显示过程中的语句。当启用 <code>RememberLastStatement</code> 时，可以通过查询 <code>LastStatement</code> 连接属性的值为连接获取 <code>LastStatement</code> 的当前值：</p> <pre>SELECT CONNECTION_PROPERTY( 'Last-Statement' );</pre> <p>如果启用客户端语句高速缓存并且重用了某个高速缓存的语句，则此属性返回空字符串。当启用 <code>RememberLastStatement</code> 时，下面的语句为指定连接返回最新准备的语句：</p> <pre>SELECT CONNECTION_PROPERTY( 'Last-Statement', connection-id );</pre> <p><code>sa_conn_activity</code> 系统过程为所有连接都返回此相同的信息。</p> <hr/> <p><b>注意：</b> 当指定 <code>-zl</code> 或启用 <code>RememberLastStatement</code> 服务器设置时，任何用户都可以调用 <code>sa_conn_activity</code> 系统过程或获取 <code>LastStatement</code> 连接属性的值，以找到为任何其他用户最新准备的 SQL 语句。应慎用此选项，并在不需要时将其关闭。</p>

选项名称	值	其它信息
RequestFilterConn	connection-id、-1	<p>对请求记录信息进行过滤以便只记录特定连接的信息。在对具有多个活动连接或多个数据库的数据库服务器进行监控时，此过滤功能可减小请求日志文件的大小。可通过执行以下语句来获得连接 ID:</p> <pre>CALL sa_conn_info( );</pre> <p>获得连接 ID 后，即可执行以下语句来记录特定连接:</p> <pre>CALL sa_server_option( 'RequestFilterConn', connection-id );</pre> <p>在显式地重置过滤或关闭数据库服务器之前，过滤将一直有效。要重置过滤，请使用以下语句:</p> <pre>CALL sa_server_option( 'RequestFilterConn', -1 );</pre>
RequestFilterDB	database-id、-1	<p>对请求记录信息进行过滤以便只记录特定数据库的信息。在对具有多个数据库的服务器进行监控时，此过滤功能可减小请求日志文件的大小。在连接到所需的数据库后，可通过执行以下语句来获得数据库 ID:</p> <pre>SELECT CONNECTION_PROPERTY( 'DBNumber' );</pre> <p>要仅记录特定数据库的信息，请执行以下语句:</p> <pre>CALL sa_server_option( 'RequestFilterDB', database-id );</pre> <p>在显式地重置过滤或关闭数据库服务器之前，过滤将一直有效。要重置过滤，请使用以下语句:</p> <pre>CALL sa_server_option( 'RequestFilterDB', -1 );</pre>

选项名称	值	其它信息
RequestLogFile	filename	<p>用来记录请求信息的文件名。指定一个空字符串会停止将请求信息记录到请求日志文件。如果启用请求记录，但请求日志文件未指定或设置为空字符串，则服务器会将请求记录到数据库服务器消息窗口中。由于此值为 <b>SQL</b> 字符串，所以路径中的所有反斜线字符均需写两遍。</p> <p>启用客户端语句高速缓存时，如果要使用 <b>tracetime.pl</b> Perl 脚本对请求日志进行分析，应将 <b>max_client_statements_cached</b> 选项设置为 0，以便在捕获请求日志时禁用客户端语句缓存。</p>

选项名称	值	其它信息
RequestLogging	SQL、 HOSTVARS、 PLAN、 PROCEDURES、 TRIGGERS、 OTHER、 BLOCKS、 REPLACE、 ALL、YES、 NONE、NO	<p>缺省值为 NONE。</p> <p>此调用启用对发送到数据库服务器的各 SQL 语句的记录，以便结合数据库服务器选项 <code>-zr</code> 和 <code>-zo</code> 一起用来进行疑难解答。值可以是以下项的组合，这些项由加号 (+) 或逗号分隔：</p> <ul style="list-style-type: none"> <li>• <b>PLAN</b> - 启用执行计划的记录（简单形式）。如果启用过程 (PROCEDURES) 的记录，还将记录过程的执行计划。</li> <li>• <b>HOSTVARS</b> - 启用主机变量值的记录。如果指定 HOSTVARS，还将记录针对 SQL 列出的信息。</li> <li>• <b>PROCEDURES</b> - 启用在过程中执行的语句的记录。</li> <li>• <b>TRIGGERS</b> - 启用在触发器中执行的语句的记录。</li> <li>• <b>OTHER</b> - 启用 SQL 不包含的其它请求类型（如 FETCH 和 PREFETCH）的记录。然而，如果您指定 OTHER 但未指定 SQL，则其等效于指定 SQL+OTHER。包括 OTHER 可导致日志文件迅速增大，并可能对服务器性能造成负面影响。</li> <li>• <b>BLOCKS</b> - 启用显示何时在另一个连接上阻塞和解除阻塞某个连接的详细信息的记录。</li> <li>• <b>REPLACE</b> - 在记录开始时，使用同名的新（空）请求日志来替换现有的请求日志。否则，将打开现有的请求日志，并将新条目附加到文件的结尾。</li> <li>• <b>ALL</b> - 记录所有支持的信息。此值等效于指定 SQL+PLAN+HOSTVARS+PROCEDURES+TRIGGERS+OTHER+BLOCKS。此设置可导致日志文件迅速增大，会对服务器的性能产生负面影响。</li> <li>• <b>NO 或 NONE</b> - 禁用记录请求日志。</li> </ul> <p>可通过执行以下语句来查看此属性的当前设置：</p> <pre>SELECT PROPERTY( 'RequestLogging' );</pre>

选项名称	值	其它信息
RequestLogMaxSize	file-size (以字节为单位)	用来记录请求记录信息的文件的最大大小 (以字节为单位)。如果指定为 0, 则对请求记录文件而言没有最大大小, 且该文件也决不会被重命名。这是缺省值。当请求日志文件达到 sa_server_option 系统过程或 -zs 服务器选项指定的大小后, 会用附加的扩展名 .old 对该文件进行重命名 (如果已经存在这样一个文件, 则替换具有相同名称的现有文件)。然后, 重新启动请求日志文件。
RequestLogNum-Files	整数	要保留的请求日志文件副本的数目。如果请求记录在较长一段时间内处于启用状态, 则请求日志文件可能会变得很大。-zn 选项可用于指定要保留的请求日志文件副本的数量。
RequestTiming	YES、NO	<p>缺省值为 NO。</p> <p>指示数据库服务器维护每个新连接的计时信息。缺省情况下, 此功能关闭。此功能打开时, 数据库服务器会为所有新连接维护累计计时器, 该计时器表明连接在服务器的各个状态下所花费的具体时间。此更改仅对新连接有效, 并且一直持续到每个连接结束。您可以使用 sa_performance_diagnostics 系统过程以获得此计时信息的汇总, 或者通过检查以下连接属性来检索各个值:</p> <ul style="list-style-type: none"> <li>• ReqCountUnscheduled</li> <li>• ReqTimeUnscheduled</li> <li>• ReqCountActive</li> <li>• ReqTimeActive</li> <li>• ReqCountBlockIO</li> <li>• ReqTimeBlockIO</li> <li>• ReqCountBlockLock</li> <li>• ReqTimeBlockLock</li> <li>• ReqCountBlockContention</li> <li>• ReqTimeBlockContention</li> </ul> <p>当 RequestTiming 服务器属性设置为 on 时, 为每个请求维护附加计数器会产生少量开销。</p>

选项名称	值	其它信息
rlv_auto_merge	ON、OFF	<p>缺省值为 ON。</p> <p>针对启用行级版本控制的表，启用或禁用 RLV 存储库与 IQ 主存储库的自动合并。</p> <p>如果 rlv_auto_merge 为 OFF，则 RLV 和 IQ 主存储库不会自动合并。这表示您将负责手动合并数据，这样在达到 rlv_memory_mb 上限之前，RLV 存储库能够与 IQ 主存储库保持同步。</p>
rlv_memory_mb	<p>最小值为 1 MB。</p> <p>最大值为 2048。</p> <p>任何其它值均会将内存量设置为 2048 MB。</p>	<p>指定为行级版本控制保留的最大内存量（RLV 存储库，以 MB 为单位）。缺省值为 2048 MB。如果如果此值超过系统虚拟内存限制的 2/3，服务器将生成错误。</p>
SecureFeatures	feature-list	<p>用于管理已经运行的数据库服务器的安全功能。feature-list 是以逗号分隔的功能名称或功能集的列表。通过向列表中添加功能可以限制功能的可用性。要从安全功能列表中删除项目，请在安全功能名称前指定一个减号 (-)。</p> <p>要调用 sa_server_option('SecureFeatures',...)，必须在相应连接上启用 ManageFeatures 安全功能。通过 -sf 密钥（系统安全功能密钥）启用 ManageFeatures 以及所有其它功能。因此，如果已使用系统安全功能密钥，则更改 SecureFeatures 的设置将不会对相关连接产生任何影响。但是，如果使用了其它密钥（例如，通过 create_secure_feature_key 系统过程创建的密钥），则更改可能会立即影响到相关连接，具体取决于该密钥中包含的其它功能。</p> <p>对于数据库服务器，为了允许或禁止对功能进行访问而所做的更改都会立即生效。执行 sa_server_option 系统过程的连接是否会受到影响，取决于该连接所使用的安全功能密钥以及此密钥是否允许该连接访问指定的功能。</p> <p>例如，可使用以下语法来保护两个功能的安全：</p> <pre>CALL sa_server_option('SecureFeatures', 'CONSOLE_LOG,WEBCLIENT_LOG');</pre> <p>执行上述语句后，安全功能列表将根据更改内容进行设置。</p>

选项名称	值	其它信息
StatisticsCleaner	ON、OFF	缺省值为 ON。 统计信息清理程序通过对表执行扫描修复提供错误估计的统计信息。缺省情况下，统计信息清理程序在后台运行，且对性能影响很小。 关闭统计信息清理程序不会禁用统计信息调控器，但关闭统计信息清理程序时，只有在运行查询时才能创建或修复统计信息。
WebClientLogFile	filename	Web 服务客户端日志文件的名称。每次使用 <code>-zoc</code> 服务器选项或 <code>WebClientLogFile</code> 属性设置或重置文件名时，Web 服务客户端日志文件都将被截断。由于此值为字符串，所以路径中的所有反斜线字符均需写两遍。
WebClientLogging	ON、OFF	缺省值为 OFF。 此选项可启用或禁用对 Web 服务客户端的记录。记录的信息包括 HTTP 请求和响应数据。指定 ON 将开始向 Web 服务客户端日志文件记录信息，而指定 OFF 可停止向该文件记录信息。

*副作用*

无。

*示例*

以下语句的作用是，每当高速缓存大小发生变化，数据库服务器消息窗口中便会显示高速缓存信息：

```
CALL sa_server_option( 'CacheSizingStatistics', 'YES' );
```

以下语句禁止与当前数据库建立新连接：

```
CALL sa_server_option( 'ConnsDisabledForDB', 'YES' );
```

以下语句启用所有 SQL 语句、过程调用、计划、阻塞和取消阻塞事件的记录，并启动一个新请求日志：

```
CALL sa_server_option( 'RequestLogging', 'SQL+PROCEDURES+BLOCKS+PLAN+REPLACE' );
```

**另请参见**

- 确定数据库所使用的安全模型（第 381 页）



## sa\_set\_http\_header 系统过程

允许一个 Web 服务以设置 HTTP 响应标头。

### 语法

```
sa_set_http_header(
fldname
, val
)
```

### 参数

- **fldname** - 使用此 CHAR(128) 参数指定一个字符串，该字符串包含其中一个 HTTP 标头字段的名称。
- **val** - 使用此 LONG VARCHAR 参数指定应为指定参数设置的值。将响应标头设置为 NULL，有效地删除该标头。

### 注释

设置特殊标头字段 @HttpStatus 将会设置请求所返回的状态码。状态码也称为响应码。例如，以下脚本将状态码设置为 [404 未找到]：

```
CALL sa_set_http_header( '@HttpStatus', '404' );
```

可以通过指定三位数的状态代码（使用可选的以冒号分隔的文本消息）创建用户定义的状态消息。例如，以下脚本使用消息 "999 用户代码" 输出状态代码：

```
CALL sa_set_http_header( '@HttpStatus', '999:User Code' );
```

**注意：**当使用 LogOptions 协议选项登录时，用户定义的状态文本消息不会转换成数据库字符集。

错误消息主体将会被自动插入。只能使用有效的 HTTP 错误码。将状态设置为无效代码时会导致 SQL 错误。

sa\_set\_http\_header 过程在调用时始终会覆盖标头字段现有的标头值。

可以删除由数据库服务器自动生成的响应标头。例如，下面的命令删除 Expires 响应标头：

```
CALL sa_set_http_header( 'Expires', NULL );
```

### 特权

无

### 副作用

无

### 示例

以下示例将 Content-Type 标头字段设置为 text/html。

```
CALL sa_set_http_header( 'Content-Type', 'text/html' );
```

## sa\_set\_http\_option 系统过程

允许一个 Web 服务来为进程控制设置一个 HTTP 选项。

### 语法

```
sa_set_http_option(
  optname
  , val
)
```

### 参数

- **optname** – 使用此 CHAR(128) 参数指定一个字符串，该字符串包含其中一个 HTTP 选项的名称。

所支持的选项为：

- **CharsetConversion** – 使用此选项控制是否自动将结果集从数据库的字符集编码转换为客户端的字符集编码。其值只允许为 ON 和 OFF。缺省值为 ON。
- **AcceptCharset** – 使用此选项指定用于响应字符集编码的 Web 服务器的首选选项。可以按首选选项的顺序指定一个或多个字符集编码。此选项的语法符合 RFC2616 超文本传输协议 (Hypertext Transfer Protocol) 中用于 HTTP Accept-Charset 请求标头字段说明的语法。

HTTP 客户端（例如 Web 浏览器）可以提供 Accept-Charset 请求标头，该标头指定按首选选项排序的字符集编码的列表。可以选择为每个编码提供一个相关联的质量值 ( $q=qvalue$ )，该值代表针对该编码的客户端首选选项。缺省情况下，质量值为 1 ( $q=1$ )。下面是一个示例：

```
Accept-Charset: iso-8859-5, utf-8;q=0.8
```

AcceptCharset HTTP 选项值中的加号 (+) 可用作表示当前数据库字符集编码的快捷方式。加号也表示如果客户端还指定了其列表中的编码，则无论客户端分配什么样的质量值，该数据库字符集编码都具有优先级。

AcceptCharset HTTP 选项中的星号 (\*) 可用于表示当客户端和服务没有交叉列表时，只要服务器支持，Web 服务就应该使用客户端首选的字符集编码。

发送响应时，会使用客户端和 Web 服务都首选的第一个字符集编码。客户端的首选选项顺序具有更高的优先级。如果不存在互斥的编码首选选项，则使用 Web 服务首选的编码（除非 Web 服务列表中出现星号 (\*)，在这种情况下将使用客户端首选的编码）。

如果不使用 AcceptCharset HTTP 选项，则使用由客户端指定且服务器支持的首选字符集编码。如果客户端指定的编码都不受支持（或者客户端未发送 Accept-Charset 请求标头），则使用数据库字符集编码。

如果客户端未发送 Accept-Charset 标头，则会执行下列操作之一：

- 如果未指定 `AcceptCharset HTTP` 选项，则 Web 服务器将使用数据库字符集编码。
- 如果已指定 `AcceptCharset HTTP` 选项，则 Web 服务器将使用其首选的字符集编码。

如果客户端发送 `Accept-Charset` 标头，则会执行下列操作之一：

- 如果未指定 `AcceptCharset HTTP` 选项，则 Web 服务器将尝试使用客户端的某个首选字符集编码，从首选的编码开始。如果 Web 服务器不支持客户端的任何首选编码，则会使用数据库字符集编码。
- 如果已指定 `AcceptCharset HTTP` 选项，则 Web 服务器将尝试使用双方都列出的第一个首选字符集编码，从客户端首选的编码开始。例如，如果客户端发送按首选项的顺序排列的 `Accept-Charset` 标头列表编码 (`iso-a`、`iso-b` 和 `iso-c`)，而 Web 服务器首选 `iso-b`，然后是 `iso-a`，最后是 `iso-c`，则会选择 `iso-a`。

```
Web client: iso-a, iso-b, iso-c
Web server: iso-b, iso-a, iso-c
```

如果两个列表的交叉部分为空，则使用 Web 服务器的第一个首选字符集。在以下示例中，将使用编码 `iso-d`。

```
Web client: iso-a, iso-b, iso-c
Web server: iso-d, iso-e, iso-f
```

如果 `AcceptCharset HTTP` 选项中包含星号 (\*)，则重点在于客户端选择的编码，最终会使用 `iso-a`。实质上，使用星号即保证两个列表的交叉部分不会为空。

最理想的情况是客户端和 Web 服务都使用数据库字符集编码，因为这样就无需字符集转换，并且可提高 Web 服务器的响应时间。

如果 `CharsetConversion` 选项设置为 `OFF`，则不会执行 `AcceptCharset` 处理。

- **SessionID** - 使用此选项创建、删除或重命名 HTTP 会话。如果 Web 服务设置该选项以创建 HTTP 会话，但是会话在服务器重新启动后未予保留，则数据库连接仍然存在。如果已经在会话上下文内，该调用将为会话重命名新的会话 ID。当使用 `NULL` 值调用时，会在 Web 服务终止时删除该会话。

生成的会话密钥的长度不得超过 128 个字符，并且如果装载了多个数据库，则该密钥在数据库之间必须是唯一的。

- **SessionTimeout** - 使用此选项指定非活动期间 HTTP 会话持续的时间（以单位为分钟）。当 HTTP 请求使用给定会话时，会重置该超时期限。超过 `SessionTimeout` 时会自动删除会话。
- **val** - 使用此 `LONG VARCHAR` 参数指定应为指定选项设置的值。

#### 注释

在处理 Web 服务的语句或过程中使用此过程设置选项。

如果在通过 web 服务调用的过程中调用 `sa_set_http_option`，并且此选项无效或选项值无效，则将返回错误。

特权

无

副作用

无

## 示例

以下示例演示使用 `sa_set_http_option` 指示 Web 服务的数据库字符集编码的首选项。指定 UTF-8 编码作为第二选择。星号 (\*) 指示 Web 服务愿意使用客户端首选的字符集编码，只要 Web 服务器支持该编码即可。

```
CALL sa_set_http_option( 'AcceptCharset', '+,UTF-8,*');
```

以下示例演示使用 `sa_set_http_option` 正确识别 Web 服务所使用的字符编码。在此示例中，Web 服务器连接到 1251CYR 数据库，并准备好向任何 Web 浏览器提供包含西里尔字母表的 HTML 文档。

```
CREATE OR REPLACE PROCEDURE cyrillic_html ()
RESULT (html_doc XML)
BEGIN
  DECLARE pos INT;
  DECLARE charset VARCHAR(30);
  CALL sa_set_http_option( 'AcceptCharset', 'iso-8859-5, utf-8' );
  SET charset = CONNECTION_PROPERTY( 'CharSet' );
  -- Change any IANA labels like ISO_8859-5:1988
  -- to ISO_8859-5 for Firefox.
  SET pos = LOCATE( charset, ':' );
  IF pos > 0 THEN
    SET charset = LEFT( charset, pos - 1 );
  END IF;
  CALL sa_set_http_header( 'Content-Type', 'text/html; charset=' ||
    charset );
  SELECT '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">' ||
    XMLCONCAT(
      XMLELEMENT('HTML',
        XMLELEMENT('HEAD',
          XMLELEMENT('TITLE', 'Cyrillic characters')
        ),
        XMLELEMENT('BODY',
          XMLELEMENT('H1', 'First 5 lowercase Russian letters'),
          XMLELEMENT('P', UNISTR('\u0430\u0431\u0432\u0433\u0434'))
        )
      )
    );
END;

CREATE SERVICE cyrillic
TYPE 'RAW'
AUTHORIZATION OFF
USER DBA
AS CALL cyrillic_html();
```

要展示建立要使用的正确字符集编码的过程，可以考虑以下由 Web 浏览器（例如 Firefox）提供给 Web 服务的 Accept-Charset 标头。它表明浏览器首选 ISO-8859-1 和 UTF-8 编码，但可接受其它编码。

```
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
```

Web 服务不会接受 ISO-8859-1 字符集编码，因为要传输的页面包含西里尔语字符。如对 sa\_set\_http\_option 的调用所示，Web 服务首选 ISO-8859-5 或 UTF-8 编码。在此示例中选择 UTF-8 编码，因为双方都接受该编码。数据库连接属性 CharSet 指示 Web 服务已选择哪种编码。sa\_set\_http\_header 过程用于将 HTML 文档的编码指示给 Web 浏览器。

```
Content-Type: text/html; charset=UTF-8
```

如果 Web 浏览器不指定 Accept-Charset，则 Web 服务缺省使用其首选项 ISO-8859-5。sa\_set\_http\_header 过程用于指示 HTML 文档的编码。

```
Content-Type: text/html; charset=ISO_8859-5
```

以下示例设置唯一的 HTTP 会话标识符：

```
BEGIN
  DECLARE sessionid VARCHAR(30);
  DECLARE tm TIMESTAMP;
  SET tm = NOW(*);
  SET sessionid = 'MySessions ' ||
    CONVERT( VARCHAR, SECONDS(tm)*1000 + DATEPART(millisecond,tm));
  SELECT sessionid;
  CALL sa_set_http_option('SessionID', sessionid);
END;
```

以下示例将 HTTP 会话的超时时间设置为 5 分钟：

```
CALL sa_set_http_option('SessionTimeout', '5');
```

## sa\_table\_page\_usage 系统过程

报告有关数据库表的页面使用情况的信息。

语法

```
sa_table_page_usage( )
```

结果集

列名	数据类型	说明
TableId	UNSIGNED INTEGER	表 ID。
TablePages	INTEGER	表使用的表页的数目。
PctUsedT	INTEGER	已使用表页空间的百分比。
IndexPages	INTEGER	表使用的索引页的数目。

列名	数据类型	说明
PctUsedI	INTEGER	已使用索引页空间的百分比。
PctOfFile	INTEGER	表占用的全部数据库文件的百分比。
TableName	CHAR(128)	表名。

*注释*

结果包括由信息实用程序提供的同样信息。将 `progress_messages` 数据库选项设置为 `Raw` 或 `Formatted` 时，进度消息将在 `sa_table_page_usage` 系统过程运行期间从数据库服务器发送至客户端。

*特权*

必须拥有 `MANAGE ANY DBSPACE` 系统特权。

*副作用*

无

*示例*

以下示例获取 `SalesOrderItems` 表的页面使用信息。

```
SELECT * FROM sa_table_page_usage( )
WHERE TableName = 'SalesOrderItems';
```

*另请参见*

- 确定数据库所使用的安全模型（第 381 页）

**sa\_text\_index\_stats 系统过程**

返回有关数据库中 `TEXT` 索引的统计信息。

*语法*

```
sa_text_index_stats( )
```

*特权*

您必须具有系统过程的 `EXECUTE` 特权。必须具有以下一种系统特权：

- `MANAGE ANY STATISTICS`
- `CREATE ANY INDEX`
- `ALTER ANY INDEX`
- `DROP ANY INDEX`
- `CREATE ANY OBJECT`
- `ALTER ANY OBJECT`
- `DROP ANY OBJECT`

*注释*

使用 `sa_text_index_stats` 可以查看数据库中的每个 **TEXT** 索引的统计信息。

**表 120. sa\_text\_index\_stats 返回的 TEXT 索引的统计信息**

列名	类型	描述
owner_id	UNSIGNED INT	表所有者的 ID
table_id	UNSIGNED INT	表 ID
index_id	UNSIGNED INT	TEXT 索引的 ID
text_config_id	UNSIGNED BIGINT	TEXT 索引引用的文本配置的 ID
owner_name	CHAR(128)	所有者的名称
table_name	CHAR(128)	表的名称
index_name	CHAR(128)	TEXT 索引的名称
text_config_name	CHAR(128)	文本配置对象的名称
doc_count	UNSIGNED BIGINT	TEXT 索引中的索引列值总数
doc_length	UNSIGNED BIGINT	TEXT 索引中数据的总长度
pending_length	UNSIGNED BIGINT	待执行更改的总长度
deleted_length	UNSIGNED BIGINT	待执行删除的总长度
last_refresh	TIMESTAMP	最后一次刷新的日期和时间

对于 IMMEDIATE REFRESH TEXT 索引, pending\_length、deleted\_length 和 last\_refresh 的值为 NULL。

*示例*

返回数据库中每个 **TEXT** 索引的统计信息:

```
CALL sa_text_index_stats( );
```

**另请参见**

- 确定数据库所使用的安全模型 (第 381 页)

**sa\_text\_index\_vocab 系统过程**

列出所有在 **TEXT** 索引中出现的术语以及每个术语在其中出现的索引值的总数。

*语法*

```
sa_text_index_vocab (
    'text-index-name',
```

```
'table-name',
'table-owner'
)
```

参数

- **text-index-name** - 此 CHAR(128) 参数用于指定 **TEXT** 索引的名称。
- **table-name** - 此 CHAR(128) 参数用于指定构建 **TEXT** 索引时所基于的表的名称。
- **table-owner** - 此 CHAR(128) 参数用于指定表的所有者。

特权

您必须具有系统过程的 EXECUTE 特权。 必须具有以下一种:

- SELECT ANY TABLE 系统特权
- 对索引表的 SELECT 特权

注释

**sa\_text\_index\_vocab** 返回 **TEXT** 索引中出现的所有术语，以及显示每个术语的带索引值的总数（如果术语在某些带索引值中出现多次，则该值小于总出现次数）。

参数值不能是宿主变量或表达式。参数 *text-index-name*、*table-name* 和 *table-owner* 必须是约束或变量。

示例

执行 **sa\_text\_index\_vocab** 以返回 GROUP0 拥有的表 Customers 上 **TEXT** 索引 MyTextIndex 中出现的所有术语:

```
sa_text_index_vocab
('MyTextIndex', 'Customers', 'GROUP0');
```

表 121. 索引中的术语

术语	次数
a	1
Able	1
Acres	1
Active	5
Advertising	1
Again	1
...	...

另请参见

- 确定数据库所使用的安全模型（第 381 页）



## sa\_validate 系统过程

针对整个数据库或部分数据库执行校验和校验。

### 语法

```
sa_validate(
  [ tbl_name
  [, owner_name ] ]
)
```

### 参数

- **tbl\_name** - 使用此可选的 CHAR(128) 参数指定要校验的表或实例化视图的名称。缺省值为 NULL。
- **owner\_name** - 使用此可选的 CHAR(128) 参数指定所有者。单独指定时，将校验所有者拥有的所有表和实例化视图。缺省值为 NULL。

### 特权

必须具有 VALIDATE ANY OBJECT 系统特权。

### 副作用

无

### 注释

指定的参数	校验类型
无	对数据库中的所有表、实例化视图和索引进行校验。数据库本身也会进行检验，包括校验和校验。
<i>tbl_name</i>	对当前用户拥有的指定表或实例化视图及其所有索引进行校验。
<i>owner_name</i>	对指定用户拥有的所有表、实例化视图和索引进行校验。
<i>tbl_name</i> 和 <i>owner_name</i>	对指定用户拥有的指定表或实例化视图及其所有索引进行校验。

此过程返回一列，其名称为 **Messages**。校验期间返回的错误出现在列中。如果校验成功且未出现错误，则该列包含 [未检测到错误]。

### 警告！小心

应在没有任何连接对数据库进行更改时对表或整个数据库进行校验；否则，可能会报告错误，指出某种形式的数据库损坏，而实际上并没有任何损坏。

### 示例

以下语句对用户 DBA 拥有的表和实例化视图执行校验：

```
CALL sa_validate( owner_name = 'DBA' );
```

### 另请参见

- 确定数据库所使用的安全模型 (第 381 页)

## sa\_verify\_password 系统过程

校验当前用户的口令。

### 语法

```
sa_verify_password( curr_pswd )
```

### 参数

- *curr\_pswd* - 使用此 CHAR(128) 参数指定当前数据库用户的口令。

### 返回值

此函数返回 INTEGER 值。

### 注释

此过程由 `sp_password` 使用。如果口令匹配，则返回 0，而且不会发生错误。如果口令不匹配，则诊断为一条错误。如果口令不匹配，则不会终止连接。

### 特权

无

### 副作用

无

### 示例

以下示例在当前用户为 DBA 或 User1 时尝试校验当前连接的口令。如果当前口令不匹配，则发生错误。

```
IF USER_NAME() = 'DBA' THEN  
    SELECT sa_verify_password( 'sql' );  
ELSEIF USER_NAME() = 'User1' THEN  
    SELECT sa_verify_password( 'user' );  
END IF;
```

## sp\_alter\_secure\_feature\_key 系统过程

通过修改验证密钥和/或功能列表更改以前定义的安全功能密钥。

### 语法

```
sp_alter_secure_feature_key (  
    name,  
    auth_key,  
    features )
```

### 参数

- **name** - 要变更的安全功能密钥的 VARCHAR (128) 名称。必须已经存在具有给定名称的密钥。
- **auth\_key** - 安全功能密钥的 CHAR (128) 验证密钥。验证密钥必须为至少六个字符的非空字符串，或者为 NULL，NULL 表示不更改现有的验证密钥。
- **features** - 安全功能的 LONG VARCHAR 列表，以逗号分隔，可由密钥启用。feature\_list 可为 NULL，表示不更改现有 feature\_list。

### 特权

您必须具有系统过程的 EXECUTE 特权。此外，您必须是数据库服务器的所有者，并且已为连接启用 manage\_keys 功能。

### 注释

此过程允许您变更现有安全功能密钥的验证密钥或功能列表。

## sp\_auth\_sys\_role\_info 系统过程

生成有关将权限映射到对应系统角色和角色 ID 的报告。该过程将对每个权限返回一行。

### 语法

```
sp_auth_sys_role_info()
```

### 特权

您必须具有系统过程的 EXECUTE 特权。

### 注释

列名	数据类型	描述
auth	varchar(20)	权限的名称。
role_name	char(128)	对应的系统角色的名称。
role_id	unsigned int	系统角色的 ID。

## sp\_create\_secure\_feature\_key 系统过程

创建新的安全功能密钥。

### 语法

```
sp_create_secure_feature_key (
    name,
    auth_key,
    features )
```

### 参数

- **name** - 新安全功能密钥的 `VARCHAR (128)` 名称。此参数不能为 `NULL` 或空字符串。
- **auth\_key** - 安全功能密钥的 `CHAR (128)` 验证密钥。验证密钥必须为至少六个字符的非空字符串。
- **features** - 安全功能的 `LONG VARCHAR` 列表，以逗号分隔，可由新密钥启用。在功能之前指定 "-" 表示该功能在设置安全功能密钥时不重新启用。

### 特权

您必须具有系统过程的 `EXECUTE` 特权。此外，您必须是数据库服务器的所有者，并且已为连接启用 `manage_keys` 功能。

### 注释

此过程创建的新安全功能密钥可授予任何用户。系统安全功能密钥使用 `-sk` 数据库服务器选项进行创建。

## sp\_displayroles 系统过程

显示授予用户定义角色或用户的所有角色，或显示整个角色层次树。

### 语法

```
sp_displayroles (
  [ user_role_name ],
  [ display_mode ],
  [ grant_type ] )
```

### 参数

- **user\_role\_name** - 有效值包括：
  - 有效的系统特权名称或系统特权角色名称
  - 有效的用户定义角色名称
  - 有效的用户名在缺省情况下，如果未指定参数，则使用当前登录用户。
- **display\_mode** - 有效值包括：
  - **EXPAND\_UP** - 显示授予输入角色或系统特权的所有角色；即父级的角色层次树。
  - **EXPAND\_DOWN** - 显示授予输入角色或用户的所有角色或系统特权；即子级的角色层次树。如果未指定参数（缺省值），则仅显示直接授予的角色或系统特权。
- **grant\_type** - 有效值包括：

- **ALL** - 显示授予的所有角色或系统特权。
- **NO\_ADMIN** - 显示使用 **WITH NO ADMIN OPTION** 或 **WITH ADMIN OPTION** 子句授予的所有角色或系统特权。
- **ADMIN** - 显示使用 **WITH ADMIN OPTION** 或 **WITH ADMIN ONLY OPTION** 子句授予的所有角色或系统特权。

如果未指定参数，则使用 “**ALL**”。

### 特权

您必须具有系统过程的 **EXECUTE** 特权。要对其他用户执行此过程，您必须具有 **MANAGE ROLES** 系统特权。要对角色或系统特权执行此过程，您必须是角色管理员或对系统特权具有管理权限。

### 注释

列名	数据类型	说明
role_name	char(128)	列出角色/系统特权名称。
parent_role_name	char(128)	列出父项的角色名称。
grant_type	char(10)	列出授予类型。
role_level	smallint	对于 <b>Expand_down</b> 模式，1 表示直接授予的角色，2 表示其下面的下一个层次级别，依此类推。对于 <b>Expand_up</b> 模式，0 表示为指定角色授予的角色，-1 表示其上面的下一个层次级别，依此类推。

使用系统特权名称作为名称时，其结果将显示系统特权名称而不是系统特权角色名称。

使用 **Expand\_down** 模式时，级别 1（直接授予的角色）的 **parent\_role\_name** 为 **NULL**。如果未指定模式（缺省设置），则 **role\_level** 为 1，**parent\_role\_name** 为 **NULL**，因为只显示直接授予的角色。

如果使用用户名作为名称，且模式为 **expand\_up**，则不返回任何结果，因为用户位于任意角色层次中的顶级。同样，如果使用不可变系统特权名称作为名称，且模式为 **expand\_down**，也不会返回任何结果，因为不可变系统特权位于任意角色层次中的底层。

使用缺省模式时，**parent\_role\_name** 列为 **NULL** 且 **role\_level** 为 1。

### 示例

该示例假设已执行下面的 **GRANT** 语句：

```
GRANT SERVER OPERATOR TO r4;
GRANT BACKUP DATABASE TO r3 WITH ADMIN OPTION;
GRANT DROP CONNECTION TO r3 WITH ADMIN ONLY OPTION;
GRANT MONITOR TO r2;GRANT CHECKPOINT TO r1;
```

```
GRANT ROLE r2 TO r1 WITH ADMIN OPTION;
GRANT ROLE r3 TO r2 WITH NO ADMIN OPTION;
GRANT ROLE r4 TO r3 WITH ADMIN ONLY OPTION;
GRANT ROLE r1 TO user1;
GRANT ROLE r1 TO r7;
GRANT ROLE r7 TO user2 WITH ADMIN OPTION;
GRANT BACKUP DATABASE TO user2 WITH ADMIN ONLY OPTION;
```

sp\_displayroles( 'user2', 'expand\_down', 'ALL' ) 产生的输出如下所示:

role_name	parent_role_name	grant_type	role_level
r7	NULL	ADMIN	1
PUBLIC	NULL	NO ADMIN	1
BACKUP DATABASE	NULL	ADMIN ONLY	1
dbo	PUBLIC	NO ADMIN	2
r1	r7	NO ADMIN	2
r2	r1	ADMIN	3
CHECKPOINT	r1	NO ADMIN	3
r3	r2	NO ADMIN	4
MONITOR	r2	NO ADMIN	4
r4	r3	ADMIN ONLY	5
BACKUP DATABASE	r3	ADMIN	5
DROP CONNECTION	r3	ADMIN ONLY	5

sp\_displayroles( 'user2', 'expand\_down', 'NO\_ADMIN' ) 产生的输出如下所示:

role_name	parent_role_name	grant_type	role_level
r7	NULL	ADMIN	1
PUBLIC	NULL	NO ADMIN	1
dbo	PUBLIC	NO ADMIN	2
r1	r7	NO ADMIN	2
r2	r1	ADMIN	3
CHECKPOINT	r1	NO ADMIN	3

role_name	parent_role_name	grant_type	role_level
r3	r2	NO ADMIN	4
MONITOR	r2	NO ADMIN	4
BACKUP DATABASE	r3	ADMIN	5

`sp_displayroles( 'r3', 'expand_up', 'NO_ADMIN' )` 产生的输出如下所示:

role_name	parent_role_name	grant_type	role_level
r1	r7	NO ADMIN	-2
r2	r1	ADMIN	-1
r3	r2	NO ADMIN	0

`sp_displayroles( 'r1', 'NO_ADMIN', 'expand_up' )` 产生的输出如下所示:

role_name	parent_role_name	grant_type	role_level
r1	r7	NO ADMIN	0

## sp\_drop\_secure\_feature\_key 系统过程

删除安全功能密钥。

### 语法

```
sp_drop_secure_feature_key ( name )
```

### 参数

- **name** – 要删除的安全功能密钥的 VARCHAR (128) 名称。

### 特权

您必须具有系统过程的 EXECUTE 特权。此外，您必须是数据库服务器的所有者，并且已为连接启用 `manage_keys` 功能。

### 注释

如果指定的密钥不存在，则返回一条错误。如果指定的密钥存在，只要它不是允许管理安全功能和安全功能密钥的最后一个安全功能密钥，就会将其删除。例如，不能删除系统安全功能密钥，除非有另一个密钥已启用 `manage_features` 和 `manage_keys` 安全功能。

## sp\_expireallpasswords 系统过程

让所有用户口令立即到期。

### 语法 1

```
call sp_expireallpasswords
```

### 语法 2

```
sp_expireallpasswords
```

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY USER 系统特权。

### 另请参见

- sp\_iqaddlogin 过程 (第 386 页)
- sp\_iqcopyloginpolicy 过程 (第 413 页)
- sp\_iqmodifylogin 过程 (第 469 页)
- sp\_iqpassword 过程 (第 484 页)

## sp\_list\_secure\_feature\_keys 系统过程

返回有关目录内容的信息。

### 语法

```
sp_list_secure_feature_keys ( )
```

### 特权

您必须具有系统过程的 EXECUTE 特权。此外，您必须是数据库服务器的所有者，并且已为连接启用 manage\_keys 功能。

### 注释

列名	数据类型	说明
name	VARCHAR(128)	安全功能密钥的名称。
features	LONG VARCHAR	通过安全功能密钥启用的安全功能。

此过程返回现有安全功能密钥的名称，以及可通过每个密钥启用的安全功能集。

如果用户已启用 manage\_features 和 manage\_keys 安全功能，则该过程将返回所有安全功能密钥的列表。

如果用户仅启用了 manage\_keys 安全功能，则该过程返回功能或功能子集与当前用户所启用功能相同的密钥。



## sp\_login\_environment 系统过程

设置用户登录时的连接选项。

### 语法

```
sp_login_environment ( )
```

### 注释

sp\_login\_environment 是由 login\_procedure 数据库选项调用的缺省过程。

建议不要编辑此过程。相反，如果要改变登录环境，请将 login\_procedure 选项设置为指向另一个过程。

### 特权

无

### 副作用

无

## sp\_objectpermission 系统过程

生成有关授予指定角色或用户名的对象特权或者授予指定对象或 dbspace 的对象特权的报告。

### 语法

```
sp_objectpermission ( [object_name], [object_owner], [object_type] )
```

### 参数

- **object\_name** - 对象、dbspace、用户或角色的名称。如果未指定，则报告当前用户的对象特权。缺省值为 NULL。
- **object\_owner** - 指定对象名称的对象所有者的名称。显示由指定对象所有者拥有的指定对象的对象特权。必须指定该参数以获取由另一用户或角色所拥有的对象的对象特权。缺省值为 NULL。
- **object\_type** - 有效值包括：
  - TABLE\*
  - VIEW
  - MATERIALIZED VIEW
  - SEQUENCE
  - PROCEDURE
  - FUNCTION
  - DBSPACE
  - USER

---

**注意:** \*同时显示列级别的对象特权。

---

如果未指定任何值，则返回所有对象类型的特权。缺省值为 NULL。

*特权*

您必须具有系统过程的 EXECUTE 特权。任何用户都可以执行 **sp\_objectpermission** 以获取授予自己的所有对象特权。对象所有者也可以执行此过程来获取其自身拥有的对象的对象特权。要获取以下对象特权，还需具备其它系统特权：

- **授予其它用户或授权其它用户拥有的对象的对象特权** - 您还要必须具有 **MANAGE ANY OBJECT PRIVILEGE** 系统特权
- **授予角色拥有的对象或授予角色的对象特权** - 您还要必须具有 **MANAGE ANY OBJECT PRIVILEGE** 系统特权或成为角色的角色管理员
- **dbspace 的对象特权** - 您必须具有 **MANAGE ANY DBSPACE** 系统特权

*注释*

列名	数据类型	说明
授予者	char(128)	授予者的用户 ID
grantee	char(128)	被授予者的用户 ID
object_name	char(128)	对象的名称
owner	char(128)	对象所有者的名称
object_type	char(20)	对象的类型
column_name	char(128)	列的名称
permission	char(20)	特权名称
grantable	char(1)	特权是否可授予

所有参数均为可选，并可生成以下报告：

- 如果输入为对象（表、视图、过程、函数、序列等），该过程将列出对该对象具有不同对象特权的所有角色和用户。
- 如果输入为角色或用户，该过程将列出授予该角色或输入的所有对象特权。通过执行 **sp\_objectpermission** 来显示用户或角色的对象特权时，也将显示通过角色授予继承而来的对象特权。
- 如果输入为 **dbspace** 名称，该过程将列出对指定 **dbspace** 具有 **CREATE** 特权的所有用户或角色。
- 缺省情况下，对象类型为 **NULL**，并将显示与指定对象名称匹配的所有现有对象类型的对象特权。

*示例*

执行下面的 **GRANT** 语句：

```

GRANT SERVER OPERATOR TO r4;
GRANT BACKUP DATABASE TO r3 WITH ADMIN OPTION;
GRANT DROP CONNECTION TO r3 WITH ADMIN ONLY OPTION;
GRANT MONITOR TO r2;GRANT CHECKPOINT TO r1;
GRANT ROLE r2 TO r1 WITH ADMIN OPTION;
GRANT ROLE r3 TO r2 WITH NO ADMIN OPTION;
GRANT ROLE r4 TO r3 WITH ADMIN ONLY OPTION;

```

假设具有以下对象特权:

- r5 在数据库中拥有一个名为 test\_tab 的表和一个名为 test\_proc 的过程。
- 对 r5 拥有管理权限的 u5 授予以下特权:
  - GRANT SELECT ON r5.test\_tab TO r2 WITH GRANT OPTION;
  - GRANT SELECT (c1), UPDATE (c1) ON r5.test\_tab TO r6 WITH GRANT OPTION;
  - GRANT EXECUTE ON r5.test\_proc TO r3;
- 对 r6 拥有管理权限的 u6 授予以下特权:
  - GRANT SELECT (c1), REFERENCES (c1) ON r5.test\_tab TO r3;

如果执行 sp\_objectpermission( 'r1' ), 输出将如下所示:

表 122. 示例 sp\_objectpermission( 'r1' ) 输出

授予者	grantee	object_name
u5	r2	test_tab
u6	r3	test_tab
u6	r3	test_tab
u6	r3	test_proc

(继续)	object_type	授予者
owner		
r5	TABLE	u5
r5	COLUMN	u6
r5	COLUMN	u6
r5	PROCEDURE	u6

(继续)	column_name	privilege
可授予		
Y	NULL	SELECT

(继续) 可授予	column_name	privilege
N	c1	SELECT
Y	c1	REFERENCES
N	NULL	EXECUTE

如果执行 `sp_objectpermission( 'test_tab' , 'r5' , 'table' )` , 输出将如下所示:

表 123. 示例 `sp_objectpermission( 'test_tab' , 'r5' , 'table' )` 输出

授予者	grantee	object_name
u5	r2	test_tab
u5	r6	test_tab
u5	r6	test_tab
u6	r3	test_tab
u6	r3	test_tab

(继续) owner	object_type	授予者
r5	TABLE	u5
r5	COLUMN	u5
r5	COLUMN	u5
r5	COLUMN	u6
r5	COLUMN	u6

(继续) column_name	privilege	可授予
NULL	SELECT	Y
c1	SELECT	Y
c1	UPDATE	Y

(继续) column_name	privilege	可授予
c1	SELECT	N
c1	REFERENCES	N

## sp\_proc\_priv 系统过程

生成有关运行存储过程及通过该过程的特权检查至少应具备的系统特权的报告。

### 语法

```
sp_proc_priv ( [proc_name] )
```

### 特权

您必须具有系统过程的 EXECUTE 特权。

### 注释

列名	数据类型	描述
proc_name	char(128)	存储过程的名称。
privilege	long varchar	通过特权检查所需的特权。

如果针对某个存储过程显示了多个以逗号分隔的系统特权，这表示具备其中任何一个特权都能够执行该存储过程。如果针对某个存储过程显示了多行，那么执行该存储过程需具备每行中的一个系统特权。

此过程针对某个存储过程仅列出了那些始终会通过该存储过程的特权检查的系统特权。可能有其它系统特权在给定条件下能够通过执行该过程所需的特权检查，但此过程并未列出这些系统特权。

### 示例

如果在调用 `sp_proc_priv` 时未指定任何参数，该过程将显示所有存储过程以及执行每个存储过程所必需的系统特权。执行时无需任何系统特权的存储过程将不会显示。

如果执行 `sp_proc_priv ()`，输出将如下所示：

proc_name	特权
sp_iqrowdensity	MONITOR、MANAGE ANY DBSPACE、CREATE ANY INDEX、ALTER ANY INDEX、CREATE ANY OBJECT、ALTER ANY OBJECT
sp_iqworkmon	MONITOR
sp_iqindexsize	MANAGE ANY DBSPACE、ALTER ANY INDEX、ALTER ANY OBJECT

proc_name	特权
sp_addlogin	MANAGE ANY USER
sp_iqemptyfile	BACKUP DATABASE、 SERVER OPERATOR、 ALTER DATABASE
sp_iqemptyfile	INSERT ANY TABLE、 UPDATE ANY TABLE、 DELETE ANY TABLE、 ALTER ANY TABLE、 LOAD ANY TABLE、 TRUNCATE ANY TABLE、 ALTER ANY OBJECT
...	...

如果在调用 **sp\_proc\_priv** 时指定了一个过程名称参数，将返回执行该过程所必需的系统特权。如果不需要任何系统特权，将针对该过程列出“不需要特权”。

proc_name	特权
sp_iqindexsize	MANAGE ANY DBSPACE、 ALTER ANY INDEX

如果过程不存在，将显示一条错误消息。

## sp\_remote\_columns 系统过程

生成远程表中的列的列表以及它们的数据类型的描述。

要使用此系统过程，必须用 **CREATE SERVER** 语句对服务器进行定义。

### 语法

```
sp_remote_columns (
  @server_name
  , @table_name
  [, @table_owner
  [, @table_qualifier ] ]
)
```

### 参数

- **@server\_name** - 使用此 CHAR(128) 参数指定包含 **CREATE SERVER** 语句所指定的服务器名的字符串。
- **@table\_name** - 使用此 CHAR(128) 参数指定远程表的名称。
- **@table\_owner** - 使用此可选的 CHAR(128) 参数指定 **table\_name** 的所有者。缺省值为 '%'。
- **@table\_qualifier** - 使用此可选的 CHAR(128) 参数指定 **table\_name** 所在的数据库的名称。缺省值为 '%'。

## 结果集

列名	数据类型	说明
database	CHAR(128)	数据库名称。
owner	CHAR(128)	数据库所有者名称。
table_name	CHAR(128)	表名。
column_name	CHAR(128)	列的名称。
domain_id	SMALLINT	指明列的数据类型的 INTEGER。
width	INTEGER	此列的含义取决于数据类型。对于字符类型，width 代表字符数。
scale	SMALLINT	此列的含义取决于数据类型。对于 NUMERIC 数据类型，scale 是小数点后的位数。
nullable	SMALLINT	如果允许列值为 NULL，则此值为 1。否则值为 0。
base_type_str	CHAR(4096)	表示列的物理类型的加批注类型字符串。

## 注释

如果您输入 **CREATE EXISTING TABLE** 语句并且指定列列表，则对于获取远程表上可用列的列表可能会有帮助。**sp\_remote\_columns** 生成远程表上的列的列表以及它们的数据类型的描述。如果指定了数据库，则必须指定所有者或提供 NULL 值。

## 特权

无

## 副作用

无

## 标准和兼容性

- **Sybase** - 受 Open Client/Open Server 支持。

## 示例

以下示例返回远程 SAP Sybase IQ 数据库服务器 RemoteSA 上 ULProduct 表中列的相关信息。表所有者为 DBA。

```
CALL sp_remote_columns( 'RemoteSA', 'ULProduct', 'DBA', null );
```

以下示例返回在使用远程服务器 RemoteASE 的 Adaptive Server Enterprise 数据库 Production 中，SYOBJECTS 表中列的相关信息。未指定表所有者。

```
CALL sp_remote_columns( 'RemoteASE', 'sysobjects', null, 'Production' );
```

以下示例返回在使用远程服务器 MyAccessDB 的 Microsoft Access 数据库内，c:\users\me\documents\MyAccessDB.accdb 中的 Customers 表中列的相关信息。Access 数据库没有表所有者，因此指定 NULL。

```
CALL sp_remote_columns( 'MyAccessDB', 'Customers', null, 'c:\\users\\me\\documents\\MyAccessDB.accdb' );
```

## sp\_remote\_exported\_keys 系统过程

提供其外键属于指定主表的表信息。

要使用此系统过程，必须用 CREATE SERVER 语句对服务器进行定义。

### 语法

```
sp_remote_exported_keys (
  @server_name
  , @table_name
  [, @table_owner
  [, @table_qualifier ] ]
)
```

### 参数

- **@server\_name** - 使用此 CHAR(128) 参数指定主表所在的服务器。
- **@table\_name** - 使用此 CHAR(128) 参数指定包含主键的表。
- **@table\_owner** - 使用此可选的 CHAR(128) 参数指定主表的所有者。缺省值为 '%'。
- **@table\_qualifier** - 使用此可选的 CHAR(128) 参数指定包含主表的数据库。缺省值为 '%'。

### 结果集

列名	数据类型	说明
pk_database	CHAR(128)	包含主键表的数据库。
pk_owner	CHAR(128)	主键表的所有者。
pk_table	CHAR(128)	主键表。
pk_column	CHAR(128)	主键列的名称。
fk_database	CHAR(128)	包含外键表的数据库。
fk_owner	CHAR(128)	外键表的所有者。
fk_table	CHAR(128)	外键表。
fk_column	CHAR(128)	外键列的名称。



列名	数据类型	说明
key_seq	SMALLINT	键序列号。
fk_name	CHAR(128)	外键名称。
pk_name	CHAR(128)	主键名称。

### 注释

此过程提供外键属于特定主表的远程表的信息。sp\_remote\_exported\_keys 系统过程的结果集包含数据库、所有者、表、列、主键和外键的名称以及外键列的外键序列。由于基础 ODBC 和 JDBC 调用的缘故，结果集可能不同，但总是返回外键的表信息和列信息。

### 特权

无

### 副作用

无

### 示例

本示例返回远程服务器 RemoteSA 上 ULEmployee 表中外键关系的相关信息：

```
CALL sp_remote_exported_keys( 'RemoteSA', 'ULEmployee', 'DBA' );
```

## sp\_remote\_imported\_keys 系统过程

提供有关远程表的信息，这些表包含对应于指定外键的主键。

要使用此系统过程，必须用 CREATE SERVER 语句对服务器进行定义。

### 语法

```
sp_remote_imported_keys (
  @server_name
  , @table_name
  [, @table_owner
  [, @table_qualifier ] ]
)
```

### 参数

- **@server\_name** - 使用此 CHAR(128) 参数指定外键表所在的服务器。此参数需要值。
- **@table\_name** - 使用此 CHAR(128) 参数指定包含外键的表。此参数需要值。
- **@table\_owner** - 使用此可选的 CHAR(128) 参数指定外键表的所有者。缺省值为 '%'。

- **@table\_qualifier** – 使用此可选的 CHAR(128) 参数指定包含外键表的数据库。缺省值为 '%'。

### 结果集

列名	数据类型	说明
pk_database	CHAR(128)	包含主键表的数据库。
pk_owner	CHAR(128)	主键表的所有者。
pk_table	CHAR(128)	主键表。
pk_column	CHAR(128)	主键列的名称。
fk_database	CHAR(128)	包含外键表的数据库。
fk_owner	CHAR(128)	外键表的所有者。
fk_table	CHAR(128)	外键表。
fk_column	CHAR(128)	外键列的名称。
key_seq	SMALLINT	键序列号。
fk_name	CHAR(128)	外键名称。
pk_name	CHAR(128)	主键名称。

### 注释

外键引用包含相应主键的单独表中的一行。使用此过程可以获取其主键对应于特定外表的远程表的列表。sp\_remote\_imported\_keys 结果集包含数据库、所有者、表、列、主键和外键的名称以及外键列的外键序列。由于基础 ODBC 和 JDBC 调用的缘故，结果集可能不同，但总是返回主键的表信息和列信息。

### 特权

无

### 副作用

无

### 示例

以下示例返回主键与远程服务器 RemoteSA 上的 ULOrder 表中的外键相对应的表：

```
CALL sp_remote_imported_keys( 'RemoteSA', 'ULOrder', 'DBA' );
```

## **sp\_remote\_primary\_keys** 系统过程

使用远程数据访问提供远程表的主键信息。

### 语法

```
sp_remote_primary_keys (
    @server_name
    , @table_name
    [, @table_owner
    [, @table_qualifier ] ]
)
```

### 参数

- **@server\_name** - 使用此 CHAR(128) 参数指定远程服务器名。
- **@table\_name** - 使用此 CHAR(128) 参数指定远程表的名称。
- **@table\_owner** - 使用此可选的 CHAR(128) 参数指定远程表的所有者。缺省值为 '%'。
- **@table\_qualifier** - 使用此可选的 CHAR(128) 参数指定远程数据库的名称。缺省值为 '%'。

### 结果集

列名	数据类型	说明
database	CHAR(128)	远程数据库的名称。
owner	CHAR(128)	表的所有者。
table_name	CHAR(128)	表的名称。
column_name	CHAR(128)	主键列的名称。
key_seq	SMALLINT	主键序列号。
pk_name	CHAR(128)	主键名称。

### 注释

此系统过程使用远程数据访问提供远程表的主键信息。

由于基础 ODBC 调用的差异，返回的信息在目录/数据库值方面会稍有不同（这取决于为服务器所指定的远程数据访问类）。

### 特权

无

### 标准和兼容性

- **Sybase** - 受 Open Client/Open Server 支持。

### 副作用

无

### 示例

以下示例返回 SAP Sybase IQ 远程服务器 RemoteSA 中由 DBA 所有的表中主键的相关信息。

```
CALL sp_remote_primary_keys( 'RemoteSA', null, 'DBA' );
```

获取 production 数据库（在名为 RemoteASE 的 Adaptive Server Enterprise 服务器中）中由 Fred 拥有的所有表中主键的列表：

```
CALL sp_remote_primary_keys( 'RemoteASE', null, 'Fred',  
'production' );
```

## sp\_remote\_tables 系统过程

返回服务器上表的列表。

### 语法

```
sp_remote_tables(  
  @server_name  
  [, @table_name  
  [, @table_owner  
  [, @table_qualifier  
  [, @with_table_type ] ] ]  
)
```

### 参数

- **@server\_name** - 使用此 CHAR(128) 参数指定远程服务器名。
- **@table\_name** - 使用此可选 CHAR(128) 参数指定远程表的名称。缺省值为 '%'。
- **@table\_owner** - 使用此可选的 CHAR(128) 参数指定远程表的所有者。缺省值为 '%'。
- **@table\_qualifier** - 使用此可选 CHAR(128) 参数指定 *table\_name* 所在的数据库。缺省值为 '%'。
- **@with\_table\_type** - 使用此可选 BIT 参数指定是否包含远程表类型。缺省值为 0。如果希望结果集包含列出表类型的列，则指定 1；如果不希望包含，则指定 0。

## 结果集

列名	数据类型	说明
database	CHAR(128)	远程数据库的名称。
owner	CHAR(128)	表所有者的名称。
table_name	CHAR(128)	表的名称。
table_type	CHAR(128)	指定表的类型。此值取决于远程服务器的类型。例如，可能的值有 TABLE、VIEW、SYS 和 GBL TEMP。

## 注释

要使用此系统过程，必须用 **CREATE SERVER** 语句对服务器进行定义。

当您配置数据库服务器以获取特定服务器上可用的远程表的列表时，此过程可能会有帮助。此过程返回服务器上表的列表。

此过程接受五个参数。如果给定了表、所有者或数据库名，则表的列表仅包含与这些参数匹配的表。

## 特权

无

## 副作用

无

## 标准和兼容性

- **Sybase** - 受 Open Client/Open Server 支持。

## 示例

以下示例返回 SAP Sybase IQ 远程服务器 RemoteSA 中由 DBA 所有的表的相关信息。

```
CALL sp_remote_tables( 'RemoteSA', null, 'DBA' );
```

获取 production 数据库（在名为 RemoteASE 的 Adaptive Server Enterprise 服务器中）中由 Fred 所有的所有表的列表：

```
CALL sp_remote_tables( 'RemoteASE', null, 'Fred', 'production' );
```

要获取名为 RemoteExcel 的服务器引用的 ODBC 数据源中可用的所有 Microsoft Excel 工作表的列表：

```
CALL sp_remote_tables( 'RemoteExcel' );
```

**sp\_servercaps 系统过程**

显示远程服务器的容量信息。

*语法*

```
sp_servercaps ( @server_name )
```

*参数*

- **@server\_name** - 使用此 CHAR(128) 参数指定使用 CREATE SERVER 语句定义的服务器。@server\_name 是 CREATE SERVER 语句中使用的同一服务器名。

*结果*

列名	数据类型	说明
capid	INTEGER	功能标识符。
capname	CHAR(128)	功能的名称。
capvalue	CHAR(128)	功能的设置，通常为 T（真）或 F（假）。

*注释*

要使用此系统过程，必须用 CREATE SERVER 语句对服务器进行定义。

此过程显示有关远程服务器的容量的信息。此容量信息用于确定可以向远程服务器转发多少 SQL 语句。直到连接到第一台远程服务器后，才会填充列出服务器功能的 ISYSCAPABILITY 系统表。

*标准和兼容性*

- **Sybase** - 受 Open Client/Open Server 支持。

*特权*

无

*副作用*

无

**示例**

显示有关远程服务器 RemoteSA 的信息：

```
CALL sp_servercaps ( 'RemoteSA' );
```

## sp\_sys\_priv\_role\_info 系统过程

生成系统特权到相应系统角色的映射报告。为每个系统特权返回单独的行。

语法

**sp\_sys\_priv\_role\_info()**

特权

您必须具有系统过程的 EXECUTE 特权。

注释

列名	数据类型	说明
sys_priv_name	char(128)	系统特权的名称。
sys_priv_role_name	char(128)	与系统特权对应的角色名称。
sys_priv_id	unsigned int	系统特权的 ID。

## sp\_tsql\_environment 系统过程

设置用户从 jConnect 或 Open Client 应用程序连接时的连接选项。

语法

```
sp_tsql_environment( )
```

注释

sp\_login\_environment 过程是由 login\_procedure 数据库选项指定的缺省过程。对于每个新连接，均调用由 login\_procedure 指定的过程。如果连接使用的是 TDS 通信协议（即，如果是 Open Client 或 jConnect 连接），则 sp\_login\_environment 接着调用 sp\_tsql\_environment。

此过程设置数据库选项，使它们与缺省 Adaptive Server Enterprise 行为兼容。

要更改缺省行为，创建新过程，然后将 login\_procedure 选项更改为指向这些新过程。

下面列出由 sp\_tsql\_environment 过程设置的选项：

```
if db_property( 'IQStore' ) = 'Off' then
  -- SAP Sybase IQ datastore
  SET TEMPORARY OPTION close_on_endtrans='OFF';
end if;
SET TEMPORARY OPTION ansinull='OFF';
SET TEMPORARY OPTION tsql_variables='ON';
SET TEMPORARY OPTION ansi_blanks='ON';
SET TEMPORARY OPTION chained='OFF';
SET TEMPORARY OPTION quoted_identifier='OFF';
SET TEMPORARY OPTION allow_nulls_by_default='OFF';
SET TEMPORARY OPTION on_tsql_error='CONTINUE';
```

```
SET TEMPORARY OPTION isolation_level='1';
SET TEMPORARY OPTION date_format='YYYY-MM-DD';
SET TEMPORARY OPTION timestamp_format='YYYY-MM-DD HH:NN:SS.SSS';
SET TEMPORARY OPTION time_format='HH:NN:SS.SSS';
SET TEMPORARY OPTION date_order='MDY';
SET TEMPORARY OPTION escape_character='OFF';
```

*特权*  
无

*副作用*  
无

### 示例

以下示例调用 `sp_tsql_environment` 过程：

```
CALL sp_tsql_environment();
```

## sp\_use\_secure\_feature\_key 系统过程

启用现有安全功能密钥。

*语法*

```
sp_use_secure_feature_key ( name, sfkey)
```

*参数*

- **name** – 要启用的安全功能密钥的 VARCHAR (128) 名称。
- **sfkey** – 要启用的安全功能密钥的 CHAR (128) 验证密钥。验证密钥必须至少六个字符。

*特权*

您必须具有系统过程的 EXECUTE 特权。

*注释*

该过程将启用由指定安全功能密钥启动的安全功能密钥。

## Adaptive Server 系统过程和目录过程

Adaptive Server 提供了执行许多管理功能和获取系统信息的系统和目录过程。SAP Sybase IQ 支持其中的某些过程：

系统过程是用于从系统中获取报告和更新系统表的内部存储过程。目录存储过程用于在表形式的系统表中检索信息。

**注意：**虽然这些过程执行的功能与它们在 Adaptive Server 中执行的功能相同，但是它们并不完全相同。如果您的先前存在的脚本使用这些过程，您可能需要检查一下这些过程。要查看存储过程的文本，请运行：



```
sp_helptext 'owner.procedure_name'
```

对于所有由 SAP Sybase 提供的系统存储过程而言，所有者为 dbo。要查看由不同用户拥有的同名存储过程的文本，您必须指定用户，例如：

```
sp_helptext 'myname.myprocedure'
```

## Adaptive Server 系统过程

SAP Sybase IQ 中提供的 Adaptive Server 系统过程：

表 124. SAP Sybase IQ 中提供的 ASE 系统过程

系统过程	描述	权限
<b>sp_addlogin</b> <i>userid, password[, defdb [, deflanguage [, fullname]]]</i>	将新用户帐户添加到数据库中	需要 MANAGE ANY USER 系统特权。
<b>sp_addmessage</b> <i>message-number, message_text [, language]</i>	将用户定义的消息添加到 SYSUSERMESSAGES，以供存储过程 PRINT 和 RAISERROR 调用使用	需要 CREATE MESSAGE 或 CREATE ANY OBJECT 系统特权。
<b>sp_addtype</b> <i>typename, datatype, [, "identity"   nulltype]</i>	创建用户定义的数据类型。SAP Sybase IQ 不支持 IDENTITY 列。	需要 CREATE DATATYPE 或 CREATE ANY OBJECT 系统特权。
<b>sp_adduser</b> <i>userid [, name_in_db [, grpname]]</i>	将新用户添加到数据库中	创建新用户需要 MANAGE ANY USER 系统特权。创建新用户并将用户添加到特定角色需要 MANAGE ANY USER 和 MANAGE ROLES 系统特权。
<b>sp_dboption</b> <i>[dbname, optionname, {true   false}]</i>	显示或更改数据库选项	不需要。
<b>sp_droplogin</b> <i>userid</i>	从数据库中删除用户	需要 MANAGE ANY LOGIN POLICY 系统特权。
<b>sp_dropmessage</b> <i>message-number [, language]</i>	删除用户定义的消息	需要 DROP MESSAGE 系统特权。
<b>sp_droptype</b> <i>typename</i>	删除用户定义数据类型	需要 DROP DATATYPE 系统特权。
<b>sp_dropuser</b> <i>userid</i>	从数据库中删除用户	需要 MANAGE ANY USER 系统特权。

系统过程	描述	权限
<b>sp_getmessage</b> <i>message-num, @msg-var output [, language]</i>	从 SYSUSERMESSAGES 中检索存储消息字符串以供 <b>PRINT</b> 和 <b>RAISERROR</b> 语句使用。	不需要。
<b>sp_helptext</b> 'owner.object-name'	显示系统过程或视图的文本	不需要。
<b>sp_password</b> <i>caller_passwd, new_passwd [, userid]</i>	添加或更改用户 ID 的口令	更改自己的口令不需要具有系统特权。但是，更改其他用户的口令需要具有 <b>CHANGE PASSWORD</b> 系统特权。

**注意：**类似 **sp\_dropuser** 与 Adaptive Server 的存储过程存在最小兼容性。如果您习惯使用 Adaptive Server，在使用 Interactive SQL 过程前，需将其文本与 SAP Sybase IQ 过程比较。要进行比较，请使用命令：

```
sp_helptext 'owner.procedure_name'
```

对于 Sybase 提供的系统存储过程，所有者始终为 dbo。要查看由不同用户拥有的同名存储过程的文本，您必须指定用户，例如：

```
sp_helptext 'myname.myprocedure'
```

另请参见

- 用户、组/角色以及权限（第 768 页）

## Adaptive Server 目录过程

SAP Sybase IQ 可实现除 **sp\_column\_privileges** 过程以外的大部分 Adaptive Server 目录过程。

对于其中某些 Adaptive Server 目录过程，SAP Sybase IQ 的定制存储过程类似。

**表 125. ASE 目录过程在 SAP Sybase IQ 中实现**

ASE 目录存储过程	描述	IQ 过程
<b>sp_columns</b> <i>table-name [, table-owner ] [, table-qualifier] [, column-name ]</i>	返回指定列的数据类型	
<b>sp_fkeys</b> <i>pktable_name [, pktable-owner ] [, pktable-qualifier ] [, fktable-name ] [, fktable_owner ] [, fktable-qualifier ]</i>	返回指定表的外键信息	
<b>sp_pkeys</b> <i>table-name [, table_owner ] [, table_qualifier ]</i>	返回单个表的主键信息	<b>sp_iqpkeys</b>

ASE 目录存储过程	描述	IQ 过程
<b>sp_special_columns</b> <i>table_name [, table-owner ][, table-qualifier ][, col-type ]</i>	返回唯一标识表中某一行的最佳列集	
<b>sp_sproc_columns</b> <i>proc-name [, proc_owner ][, proc-qualifier ][, column-name ]</i>	返回有关存储过程的输入和返回参数的信息	<b>sp_iqprocparm</b>
<b>sp_stored_procedures</b> <i>[ sp-name ][, sp-owner ][, sp-qualifier ]</i>	返回有关一个或多个存储过程的信息	<b>sp_iqprocedure</b>
<b>sp_tables</b> <i>table-name [, table-owner ][, table-qualifier ][, table-type ]</i>	返回可出现在 <b>FROM</b> 子句中的对象列表	

不支持以下 Adaptive Server 目录过程：

- sp\_column\_privileges
- sp\_databases
- sp\_datatype\_info
- sp\_server\_info



# 系统表和视图

SAP Sybase IQ 支持系统表、系统视图、统一视图、兼容性视图以及 Adaptive Server T-SQL 兼容性视图。

## 系统表

会在许多系统表中对每个 SAP Sybase IQ 数据库结构进行描述。系统表供内部使用。

DUMMY 系统表是唯一允许直接访问的系统表。对于所有其它系统表，可通过相应视图访问它们的基础数据。

表 126. 系统表列表

系统表	仅在内部使用?
DUMMY	否
ISYSARTICLE	是
ISYSARTICLECOL	是
ISYSATTRIBUTE	是
ISYSATTRIBUTENAME	是
ISYSCAPABILITY	是
ISYSCHECK	是
ISYSCOLPERM	是
ISYSCOLSTAT	是
ISYSCONSTRAINT	是
ISYSDBFILE	是
ISYSDBSPACE	是
ISYSDBSPACEPERM	是
ISYSDEPENDENCY	是
ISYSDOMAIN	是
ISYSEVENT	是
ISYSEXTERNENV	是
ISYSEXTERNENVOBJECT	是

系统表	仅在内部使用?
ISYSEXTERNLOGIN	是
ISYSFKEY	是
ISYSGROUP	是
ISYSHISTORY	是
ISYSIDX	是
ISYSIDXCOL	是
ISYSIQBACKUPHISTORY	是
ISYSIQBACKUPHISTORYDETAIL	是
ISYSIQDBFILE	是
ISYSIQDBSPACE	是
ISYSIQIDX	是
ISYSIQINFO	是
ISYSIQLOGICALSERVER	是
ISYSIQLOGINPOLICYLSINFO	是
ISYSIQLSLOGINPOLICYOPTION	是
ISYSIQLSMEMBER	是
ISYSIQLSPOLICY	是
ISYSIQLSPOLICYOPTION	是
ISYSIQMPXSERVER	是
ISYSIQMPXSERVERAGENT	是
ISYSIQPARTITIONCOLUMN	是
ISYSIQTAB	是
ISYSIQTABCOL	是
ISYSJAR	是
ISYSJARCOMPONENT	是
ISYSJAVACLASS	是
ISYSLOGINMAP	是
ISYSLOGINPOLICY	是

系统表	仅在内部使用?
ISYSLOGINPOLICYOPTION	是
ISYSMVOPTION	是
ISYSMVOPTIONNAME	是
ISYSOBJECT	是
ISYSOPTION	是
ISYSOPTSTAT	是
ISYSPARTITION	是
ISYSPARTITIONKEY	是
ISYSPARTITIONSCHEME	是
ISYSPHYSIDX	是
ISYSPROCEDURE	是
ISYSROCPARM	是
ISYSROCPERM	是
ISYSROXYTAB	是
ISYSPUBLICATION	是
ISYSREMARK	是
ISYSREMOTEOPTION	是
ISYSREMOTEOPTIONTYPE	是
ISYSREMOTETYPE	是
ISYSREMOTEUSER	是
ISYSSCHEDULE	是
ISYSSERVER	是
ISYSSOURCE	是
ISYSSQLSERVERTYPE	是
ISYSSUBPARTITIONKEY	是
ISYSSUBSCRIPTION	是
ISYSSYNC	是
ISYSSYNCPROFILE	是

系统表	仅在内部使用?
ISYSSYNCSRIPT	是
ISYSTAB	是
ISYSTABCOL	是
ISYSTABLEPERM	是
ISYSTEXTCONFIG	是
ISYSTEXTIDX	是
ISYSTEXTIDXTAB	是
ISYSTRIGGER	是
ISYSTYPEMAP	是
ISYSUSER	是
ISYSUSERAUTHORITY	是
ISYSUSERMESSAGE	是
ISYSUSERTYPE	是
ISYSVIEW	是
ISYSWEBSERVICE	是

## **SYS.DUMMY 表与 IQ\_DUMMY 表**

DUMMY 系统表作为一个始终只有一行的表提供。

它有利于从数据库中抽取信息，如，以下示例用于从数据库中获取当前用户 ID 和当前日期。

```
SELECT USER, today(*) FROM SYS.DUMMY
```

由 SQL Anywhere（目录存储库）而不是 SAP Sybase IQ 运行使用 DUMMY 表的查询。可以在 SAP Sybase IQ 数据库中创建虚拟表。例如：

```
CREATE TABLE iq_dummy (dummy_col INT NOT NULL);
```

并显式使用此表：

```
SELECT NOW() FROM iq_dummy;
```

### **DUMMY 系统表**

列名	列类型	列约束	表约束
dummy_col	INTEGER	NOT NULL	



DUMMY 表是以只读表的形式提供的，并且始终只有一行。它有利于从数据库中抽取信息，如，以下示例用于从数据库中获取当前用户 ID 和当前日期。

```
SELECT USER, today(*) FROM IQ.DUMMY;
```

在 FROM 子句中使用 IQ.DUMMY 是可选的。If no table is specified in the FROM clause, the table is assumed to be IQ.DUMMY.可按如下方法编写这一查询：

```
SELECT USER, today(*);
```

**dummy\_col** - 此列未被使用。它之所以存在，是因为如果没有任何列就无法创建表。

从 IQ.DUMMY 表中读取的开销要比从用户创建的类似表中读取的开销小，这是因为在 IQ.DUMMY 的表页中没有内部锁。

访问计划不是通过扫描 IQ.DUMMY 表构建的。而是用行构造函数算法来代替对 IQ.DUMMY 的引用，该算法会虚拟化表引用。这样便避免了与 IQ.DUMMY 相关的争用问题。DUMMY 仍旧会以表和/或相关名称的形式出现在简单计划、详细计划及图形式计划中。

## 系统视图

---

使用系统视图可查看系统表的内容。

我们提供了若干预定义的系统视图，它们以可读格式显示系统表中的信息。

系统视图描述中含有系统视图的定义。有些定义很复杂，但您可直接使用这些视图而无需理解其定义。

### 统一视图

统一视图以用户较频繁需要的形式提供数据。

例如，统一视图经常提供通常所需的连接。统一视图与系统视图不同，因为前者不仅仅是基础系统表中原始数据的简单视图。例如，系统视图中的许多列是无法识别的 ID 值，而在统一视图中，这些列是可读名称。

### 兼容性视图

不建议使用用于与较早版本 SQL Anywhere 和 SAP Sybase IQ 兼容的兼容性视图。

请尽可能使用系统视图和统一视图而不使用兼容性视图，因为 SAP Sybase IQ 的未来版本可能不支持兼容性视图。

### 按字母顺排序的系统视图列表

系统表被隐藏；但是每个表都有一个系统视图。为确保与以后版本的 IQ 主存储库兼容，应确保您的应用程序使用系统视图而不是可能会更改的基础系统表。

**ASE T-SQL 兼容性视图**

SAP Sybase IQ 可提供一组由特殊用户 DBO 拥有的视图，这些视图对应于 Adaptive Server 系统表和视图。

另请参见

- Transact-SQL 兼容性视图（第 750 页）

**ST\_GEOMETRY\_COLUMNS 统一视图**

ST\_GEOMETRY\_COLUMNS 系统视图中的每一行描述了一个在数据库中定义的空间列。

**注意：** 空间数据、空间参照系和空间测量单位只能在目录存储库中使用。

列名	数据类型	描述
table_catalog	VARCHAR(128)	供内部使用。
table_schema	CHAR(128)	包含空间列的表所属模式的名称。其等同于表所有者。
table_name	CHAR(128)	包含空间列的表的名称。
column_name	CHAR(128)	空间列的名称。
srs_name	CHAR(128)	与空间列关联的 SRS 的名称。如果 SRS 与列不关联，则 srs_name 为 NULL。
srs_id	INTEGER	与空间列相关联的 SRS 的 SRID。
table_id	UNSIGNED INT	包含列的表的数字标识符。
column_id	UNSIGNED INT	列的数字标识符。
geometry_type_name	VARCHAR(32767)	列中包含的几何的空间数据类型（例如 ST_Point、ST_Geometry 等）。

**ST\_SPATIAL\_REFERENCE\_SYSTEMS 统一视图**

ST\_SPATIAL\_REFERENCE\_SYSTEMS 系统视图中的每一行都描述了一个在数据库中定义的 SRS。此视图提供的信息量略微不同于 SYSSPATIALREFERENCINGSYSTEM 系统视图。

**注意：** 空间数据、空间参照系和空间测量单位只能在目录存储库中使用。

列名	数据类型	描述
object_id	UNSIGNED BIGINT	仅供系统使用。

列名	数据类型	描述
owner	UNSIGNED INT	SRS 的所有者。
srs_name	CHAR(128)	SRS 的名称。
srs_id	INTEGER	空间参照系的数字标识符 (SRID)。
srs_type	CHAR(11)	按 SQL/MM 标准定义的 SRS 类型。值可以是以下各项之一： <ul style="list-style-type: none"> <li>• <b>GEOGRAPHIC</b> - 此类型是针对于拥有纬度，经度（和高程）轴的基于地理配准坐标系的 SRS。这些 SRS 是 PLANAR 或 ROUND EARTH 类型。</li> <li>• <b>PROJECTED</b> - 此类型是针对于没有纬度和经度轴的基于地理配准坐标系的 SRS。这些 SRS 是 PLANAR 类型。</li> <li>• <b>ENGINEERING</b> - 此类型是针对于基于非地理配准的坐标系的 SRS。这些 SRS 是 PLANAR 类型。</li> <li>• <b>GEOCENTRIC</b> - 不支持。</li> <li>• <b>COMPOUND</b> - 不支持。</li> <li>• <b>VERTICAL</b> - 不支持。</li> </ul> 如果 srs_type 为空，则类型未指定。
round_earth	CHAR(1)	SRS 类型是 ROUND EARTH (Y) 还是 PLANAR (N)。
axis_order	CHAR(12)	描述数据库服务器如何相对于纬度和经度来解释点（例如当使用 ST_Lat 和 ST_Long 方法时）。对于非地理空间参照系，轴顺序为 x/y/z/m。对于地理空间参照系，缺省轴顺序为经度/纬度/z/m，同样支持纬度/经度/z/m 顺序。
snap_to_grid	DOUBLE	定义执行计算时使用的网格大小。
tolerance	DOUBLE	定义比较点时使用的精度。
semi_major_axis	DOUBLE	对于 ROUND EARTH SRS 的从椭球体中心到赤道的距离。
semi_minor_axis	DOUBLE	对于 ROUND EARTH SRS 的从椭球体中心到两极的距离。

列名	数据类型	描述
inv_flattening	DOUBLE	用于 ROUND EARTH SRS 中椭圆体的反转展平。这是由以下等式创建的比率: $1/f = (\text{semi-major-axis}) / (\text{semi-major-axis} - \text{semi-minor-axis})$
min_x	DOUBLE	坐标允许的 x 的最小值。
max_x	DOUBLE	坐标允许的 x 的最大值。
min_y	DOUBLE	坐标允许的 y 的最小值。
max_y	DOUBLE	坐标允许的 y 的最大值。
min_z	DOUBLE	坐标允许的 z 的最小值。
max_z	DOUBLE	坐标允许的 z 的最大值。
min_m	DOUBLE	坐标允许的 m 的最小值。
max_m	DOUBLE	坐标允许的 m 的最大值。
min_lat	DOUBLE	坐标允许的最小纬度值。
max_lat	DOUBLE	坐标允许的最大纬度值。
min_long	DOUBLE	坐标允许的最小经度值。
max_long	DOUBLE	坐标允许的最大经度值。
organization	LONG VARCHAR	创建空间参照系使用的坐标系的组织的名称。
organization_coord- sys_id	INTEGER	创建坐标系的组织为坐标系提供的 ID。
linear_unit_of_meas- ure	CHAR(128)	SRS 使用的线性测量单位。
angular_unit_of_ measure	CHAR(128)	SRS 使用的角度测量单位。
polygon_format	LONG VARCHAR	多边形内的圆环方向。CounterClockwise、ClockWise 或 EvenOdd 格式之一。
storage_format	LONG VARCHAR	数据是以规范化格式 (内部)、非规范化格式 (原始) 还是两种格式皆有 (混合) 的形式来存储。
definition	LONG VARCHAR	其它定义设置。

列名	数据类型	描述
transform_definition	LONG VARCHAR	当从此 SRS 到另一 SRS 转换数据时使用的转换定义设置。
description	LONG VARCHAR	SRS 的说明。

### **ST\_UNITS\_OF\_MEASURE 统一视图**

ST\_UNITS\_OF\_MEASURE 系统视图中的每一行都描述了一个在数据库中定义的测量单位。与 SYSUNITOFMEASURE 系统视图相比，此视图将提供更多信息。

**注意：** 空间数据、空间参照系和空间测量单位只能在目录存储库中使用。

列名	数据类型	描述
object_id	UNSIGNED BIGINT	仅供系统使用。
owner	UNSIGNED INT	测量单位的所有者。
unit_name	CHAR(128)	测量单位的名称。
unit_type	CHAR(7)	角度或线性。
conversion_factor	DOUBLE	测量单位的转换系数。
description	LONG VARCHAR	测量单位的说明。

### **SYSARTICLE 系统视图**

SYSARTICLE 系统视图的每一行都描述了发布中的一个项目。该视图的基础系统表为 ISYSARTICLE。

列名	数据类型	说明
publication_id	UNSIGNED INT	该项目所属的发布。
table_id	UNSIGNED INT	每个项目都由来自单个表的列和行组成。该列包含该表的表 ID。
where_expr	LONG VARCHAR	对于包含 WHERE 子句定义的行的子集的项目，该列包含搜索条件。
subscribe_by_expr	LONG VARCHAR	对于包含 SUBSCRIBE BY 表达式定义的行的子集的项目，该列包含表达式。
query	CHAR(1)	为数据库服务器指示项目类型信息。
alias	VARCHAR(256)	项目的别名。
schema_change_active	BIT	如果表和发布是同步模式更改过程中的一部分，则为 1。

*基础系统表上的约束*

```
PRIMARY KEY (publication_id, table_id)
FOREIGN KEY (publication_id) REFERENCES SYS.ISYSPUBLICATION
(publication_id)
FOREIGN KEY (table_id) REFERENCES SYS.ISYSTAB (table_id)
```

**另请参见**

- SYSARTICLECOL 系统视图 (第 658 页)
- SYSARTICLECOLS 统一视图 (第 658 页)

**SYSARTICLECOL 系统视图**

SYSARTICLECOL 系统视图的每一行都标识项目中的一列。该视图的基础系统表为 ISYSARTICLECOL。

列名	数据类型	说明
publication_id	UNSIGNED INT	该列所属发布的唯一标识符。
table_id	UNSIGNED INT	该列所属的表。
column_id	UNSIGNED INT	列标识符, 来自 SYSTABCOL 系统视图。

*基础系统表上的约束*

```
PRIMARY KEY (publication_id, table_id, column_id)
FOREIGN KEY (publication_id, table_id) REFERENCES SYS.ISYSARTICLE
(publication_id, table_id)
FOREIGN KEY (table_id, column_id) REFERENCES SYS.ISYSTABCOL
(table_id, column_id)
```

**SYSARTICLECOLS 统一视图**

SYSARTICLECOLS 视图中的每一行都标识项目中的一列。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息, 请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSARTICLECOLS"
as select p.publication_name,t.table_name,c.column_name
from SYS.ISYSARTICLECOL as ac
join SYS.ISYSPUBLICATION as p on p.publication_id =
ac.publication_id
join SYS.ISYSTAB as t on t.table_id = ac.table_id
join SYS.ISYSTABCOL as c on c.table_id = ac.table_id
and c.column_id = ac.column_id
```

### **SYSARTICLES 统一视图**

SYSARTICLES 视图中的每一行都描述发布中的一个项目。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSARTICLES"
  as select u1.user_name as publication_owner,p.publication_name,
    u2.user_name as table_owner,t.table_name,
    a.where_expr,a.subscribe_by_expr,a.alias
  from SYS.ISYSARTICLE as a
    join SYS.ISYSPUBLICATION as p on(a.publication_id =
p.publication_id)
    join SYS.ISYSTAB as t on(a.table_id = t.table_id)
    join SYS.ISYSUSER as u1 on(p.creator = u1.user_id)
    join SYS.ISYSUSER as u2 on(t.creator = u2.user_id)
```

### **SYSCAPABILITIES 统一视图**

SYSCAPABILITIES 视图的每一行都为远程数据库服务器指定功能的状态。该视图从 ISYSCAPABILITY 和 ISYSCAPABILITYNAME 系统表中获取数据。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSCAPABILITIES"
  as select
  ISYSCAPABILITY.capid,ISYSCAPABILITY.srvid,property('RemoteCapabilit
y',ISYSCAPABILITY.capid) as capname,ISYSCAPABILITY.capvalue
  from SYS.ISYSCAPABILITY
```

### **SYSCAPABILITY 系统视图**

SYSCAPABILITY 系统视图的每一行都指定远程数据库服务器上功能的状态。该视图的基础系统表为 ISYSCAPABILITY。

列名	数据类型	说明
capid	INTEGER	功能的 ID，如 SYSCAPABILITYNAME 系统视图中所列。
srvid	UNSIGNED INT	在其中应用了该功能的服务器，如 SYSSERVER 系统视图中所列。
capvalue	CHAR(128)	功能的值。

#### *基础系统表上的约束*

```
PRIMARY KEY (capid, srvid)
```

```
FOREIGN KEY (srvid) REFERENCES SYS.ISYSSERVER (srvid)
```

**SYSCAPABILITYNAME 系统视图**

SYSCAPABILITYNAME 系统视图中的每一行都为 SYSCAPABILITY 系统视图中每个功能 ID 提供名称。

列名	数据类型	说明
capid	INTEGER	唯一标识功能的编号。
capname	VARCHAR(32000)	功能的名称。

**注释**

SYSCAPABILITYNAME 系统视图是使用 sa\_rowgenerator 和以下服务器属性的组合进行定义：

RemoteCapability  
MaxRemoteCapability

**SYSCATALOG 统一视图**

SYSCATALOG 视图中的每一行都描述一个系统表。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSCATALOG"( creator,
    tname,dbspacename,tabletype,ncols,primary_key,"check",
    remarks )
as select u.user_name,tab.table_name,dbs.dbSPACE_name,
    if tab.table_type_str = 'BASE' then 'TABLE' else
tab.table_type_str endif,
    (select count() from SYS.ISYSTABCOL
    where ISYSTABCOL.table_id = tab.table_id),
    if ix.index_id is null then 'N' else 'Y' endif,
    null,
    rmk.remarks
from SYS.SYSTAB as tab
    join SYS.ISYSDBSPACE as dbs on(tab.dbSPACE_id = dbs.dbSPACE_id)
    join SYS.ISYSUSER as u on u.user_id = tab.creator
    left outer join SYS.ISYSIDX as ix on(tab.table_id = ix.table_id
and ix.index_id = 0)
    left outer join SYS.ISYSREMARK as rmk on(tab.object_id =
rmk.object_id)
```

**SYSCERTIFICATE 系统视图**

SYSCERTIFICATE 系统视图的每一行都存储了一个文本 PEM 格式的证书。该视图的基础系统表为 ISYSCERTIFICATE。

列名	数据类型	描述
object_id	UNSIGNED BIGINT	证书的 ID。



列名	数据类型	描述
cert_name	CHAR(128)	证书名。
contents	LONG BINARY	压缩格式的证书内容。
update_time	TIMESTAMP	上次创建或替换的当地日期和时间。
update_time_utc	TIMESTAMP WITH TIME ZONE	上次创建或替换的 UTC 日期和时间。

### 基础系统表上的约束

PRIMARY KEY (object\_id)

UNIQUE INDEX (cert\_name)

### **SYSCHECK** 系统视图

SYSCHECK 系统视图中的每一行为表中一个指定的检查约束提供定义。该视图的基础系统表为 ISYSCHECK。

列名	数据类型	说明
check_id	UNSIGNED INT	唯一标识数据库中的约束的编号。
check_defn	LONG VARCHAR	CHECK 表达式。

### 基础系统表上的约束

PRIMARY KEY (check\_id)

FOREIGN KEY (check\_id) REFERENCES SYS.ISYSCONSTRAINT (constraint\_id)

### **SYSCOLAUTH** 统一视图

SYSCOLAUTH 视图的每一行都描述针对列授予的一组权限 (UPDATE、SELECT 或 REFERENCES)。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW
"SYS"."SYSCOLAUTH" ( grantor,grantee,creator,tname,colname,
  privilege_type,is_grantable )
as select u1.user_name,u2.user_name,u3.user_name,tab.table_name,
  col.column_name,cp.privilege_type,cp.is_grantable
  from SYS.ISYSCOLPERM as cp
    join SYS.ISYSUSER as u1 on u1.user_id = cp.grantor
    join SYS.ISYSUSER as u2 on u2.user_id = cp.grantee
    join SYS.ISYSTAB as tab on tab.table_id = cp.table_id
    join SYS.ISYSUSER as u3 on u3.user_id = tab.creator
    join SYS.ISYSTABCOL as col on col.table_id = cp.table_id
    and col.column_id = cp.column_id
```

**SYSCOLLATION 兼容性视图 (不建议使用)**

**SYSCOLLATION** 兼容性视图包含数据库的归类序列信息。它可通过内置函数来获得且不是保存在目录中。以下是该视图的定义：

```
ALTER VIEW "SYS"."SYSCOLLATION"
  as select 1 as collation_id,
         DB_PROPERTY('Collation') as collation_label,
         DB_EXTENDED_PROPERTY('Collation','Description') as
collation_name,
         cast(DB_EXTENDED_PROPERTY('Collation','LegacyData') as
binary(1280)) as collation_order
```

**SYSCOLLATIONMAPPINGS 兼容性视图 (不建议使用)**

**SYSCOLLATIONMAPPINGS** 兼容性视图只有一行，其中包含数据库归类映射。它可通过内置函数来获得且不是保存在目录中。以下是该视图的定义：

```
ALTER VIEW "SYS"."SYSCOLLATIONMAPPINGS"
  as select DB_PROPERTY('Collation') as collation_label,
         DB_EXTENDED_PROPERTY('Collation','Description') as
collation_name,
         DB_PROPERTY('Charset') as cs_label,
         DB_EXTENDED_PROPERTY('Collation','ASESensitiveSortOrder') as
so_case_label,
         DB_EXTENDED_PROPERTY('Collation','ASEInsensitiveSortOrder') as
so_caseless_label,
         DB_EXTENDED_PROPERTY('Charset','java') as jdk_label
```

**SYSCOLPERM 系统视图**

**GRANT** 语句可以将 **UPDATE**、**SELECT** 或 **REFERENCES** 特权赋予表中的单个列。每个具有 **UPDATE**、**SELECT** 或 **REFERENCES** 特权的列都记录在 **SYSCOLPERM** 系统视图的一行中。该视图的基础系统表为 **ISYSCOLPERM**。

列名	数据类型	说明
table_id	UNSIGNED INT	包含该列的表的表号。
grantee	UNSIGNED INT	被赋予该列特权的用户 ID 的用户号。如果被授予者是 <b>PUBLIC</b> 角色，那么所有用户 ID 都将被赋予特权。
grantor	UNSIGNED INT	授予特权的用户 ID 的用户号。
column_id	UNSIGNED INT	此列编号与 table_id 一同标识被授予特权的列。
privilege_type	SMALLINT	此列中的编号指示列特权的种类（16=REFERENCES、1=SELECT 或 8=UPDATE）。
is_grantable	CHAR(1)	表示是否将 <b>WITH GRANT OPTION</b> 授予了该列特权。

*基础系统表上的约束*

```
PRIMARY KEY (table_id, grantee, grantor, column_id, privilege_type)
```

```
FOREIGN KEY (table_id, column_id) REFERENCES SYS.ISYSTABCOL
(table_id, column_id)
```

```
FOREIGN KEY (grantor) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (grantee) REFERENCES SYS.ISYSUSER (user_id)
```

### **SYSVOLSTAT** 系统视图

**SYSVOLSTAT** 系统视图包含由优化程序使用的列统计数据，其中包括直方图。使用 `sa_get_histogram` 存储过程或直方图实用程序可以最有效地检索该视图的内容。该视图的基础系统表为 **ISYSVOLSTAT**。

列名	数据类型	说明
table_id	UNSIGNED INT	唯一标识该列所属的表或实例化视图的编号。
column_id	UNSIGNED INT	与 table_id 一同唯一标识该列的编号。
format_id	SMALLINT	仅供系统使用。
update_time	TIMESTAMP	上次更新列统计信息的本地时间。
density	FLOAT	估计的列的单值平均选择性，不计算存储在行中的大的单值选择性。
max_steps	SMALLINT	仅供系统使用。
actual_steps	SMALLINT	仅供系统使用。
step_values	LONG BINARY	仅供系统使用。
frequencies	LONG BINARY	仅供系统使用。
update_time_utc	TIMESTAMP WITH TIME ZONE	上次更新列统计信息的 UTC 时间。

#### 基础系统表上的约束

```
PRIMARY KEY (table_id, column_id)
```

```
FOREIGN KEY (table_id, column_id) REFERENCES SYS.ISYSTABCOL
(table_id, column_id)
```

### **SYSVOLSTATS** 统一视图

**SYSVOLSTATS** 视图中包含有以直方图的形式存储并由优化程序使用的列统计信息。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSVOLSTATS" AS SELECT u.user_name, t.table_name,
c.column_name, s.format_id,
dateadd(mi, PROPERTY('TimeZoneAdjustment'), s.update_time) as
update_time, s.density, s.max_steps, s.actual_steps,
s.step_values, s.frequencies, TODATETIMEOFFSET(s.update_time,
0) as update_time_utc
FROM SYS.ISYSVOLSTAT s
```

```
JOIN SYS.ISYSTABCOL c on (s.table_id = c.table_id and s.column_id =
c.column_id)
JOIN SYS.ISYSTAB t on (t.table_id = c.table_id)
JOIN SYS.ISYSUSER u on (u.user_id = t.creator)
```

### **SYSCOLUMN 兼容性视图 (不建议使用)**

提供 SYSCOLUMN 视图的目的是为了与提供 SYSCOLUMN 系统表的旧版本软件兼容。但以前的 SYSCOLUMN 表已被 ISYSTABCOL 系统表所取代，并且您应当使用与该系统表对应的 ISYSTABCOL 系统视图。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSCOLUMN"
as select b.table_id,
       b.column_id,
       if c.sequence is null then 'N' else 'Y' endif as pkey,
       b.domain_id,
       b.nulls,
       b.width,
       b.scale,
       b.object_id,
       b.max_identity,
       b.column_name,
       r.remarks,
       b."default",
       b.user_type,
       b.column_type
from SYS.SYSTABCOL as b
     left outer join SYS.ISYSREMARK as r on(b.object_id =
r.object_id)
     left outer join SYS.ISYSIDXCOL as c on(b.table_id = c.table_id
and b.column_id = c.column_id and c.index_id = 0)
```

### **SYSCOLUMNS 统一视图**

SYSCOLUMNS 视图的每一行都描述目录中每个表的一个列和视图。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW
"SYS"."SYSCOLUMNS"( creator, cname, tname, coltype, nulls, length,
  syslength, in_primary_key, colno, default_value,
  column_kind, remarks )
as select
u.user_name, col.column_name, tab.table_name, dom.domain_name,
  col.nulls, col.width, col.scale, if ixcol.sequence is null then 'N'
else 'Y' endif, col.column_id,
  col."default", col.column_type, rmk.remarks
from SYS.SYSTABCOL as col
     left outer join SYS.ISYSIDXCOL as ixcol on(col.table_id =
ixcol.table_id and col.column_id = ixcol.column_id and
ixcol.index_id = 0)
     join SYS.ISYSTAB as tab on(tab.table_id = col.table_id)
```

```

join SYS.ISYSDOMAIN as dom on (dom.domain_id = col.domain_id)
join SYS.ISYSUSER as u on u.user_id = tab.creator
left outer join SYS.ISYSREMARK as rmk on (col.object_id =
rmk.object_id)

```

### **SYSCOLUMNS ASE 兼容性视图**

此视图由用户 DBO 所有。每个表和视图中的每一列以及过程中的每个参数在 syscolumns 中都有对应的一行。

#### 另请参见

- 每个 Adaptive Server 数据库中的表 (第 751 页)
- SYSCOMMENTS ASE 兼容性视图 (第 665 页)
- SYSINDEXES ASE 兼容性视图 (第 681 页)
- SYSIQOBJECTS ASE 兼容性视图 (第 691 页)
- SYSIQVINDEXT ASE 兼容性视图 (第 695 页)
- SYSOBJECTS ASE 兼容性视图 (第 702 页)
- SYSTYPES ASE 兼容性视图 (第 741 页)
- SYSUSERS ASE 兼容性视图 (第 747 页)

### **SYSCOMMENTS ASE 兼容性视图**

每个视图、规则、缺省值、触发器、表约束、分区、过程、计算列、基于函数的索引键和其它形式的编译对象在 syscomments 中都有对应的条目。

此视图由用户 DBO 所有。

文本列包含原始定义语句。如果文本列的长度超过 255 个字节，条目将跨越多行。每个对象最多可占据 65,025 行。

#### 另请参见

- 每个 Adaptive Server 数据库中的表 (第 751 页)
- SYSCOLUMNS ASE 兼容性视图 (第 665 页)
- SYSINDEXES ASE 兼容性视图 (第 681 页)
- SYSIQOBJECTS ASE 兼容性视图 (第 691 页)
- SYSIQVINDEXT ASE 兼容性视图 (第 695 页)
- SYSOBJECTS ASE 兼容性视图 (第 702 页)
- SYSTYPES ASE 兼容性视图 (第 741 页)
- SYSUSERS ASE 兼容性视图 (第 747 页)

**SYSCONSTRAINT 系统视图**

SYSCONSTRAINT 系统视图的每一行描述数据库中的一个命名约束。该视图的基础系统表为 ISYSCONSTRAINT。

列名	数据类型	说明
constraint_id	UNSIGNED INT	约束的唯一 ID。
constraint_type	CHAR(1)	约束类型： <ul style="list-style-type: none"> <li>• <b>C</b> - 列检查约束</li> <li>• <b>T</b> - 表约束</li> <li>• <b>P</b> - 主键</li> <li>• <b>F</b> - 外键</li> <li>• <b>U</b> - 唯一约束</li> </ul>
ref_object_id	UNSIGNED BIGINT	要应用约束的列、表或索引的对象 ID。
table_object_id	UNSIGNED BIGINT	应用约束的表的对象 ID。
constraint_name	CHAR(128)	约束名称。

*基础系统表上的约束*

```
PRIMARY KEY (constraint_id)
```

```
FOREIGN KEY (ref_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
FOREIGN KEY (table_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
UNIQUE CONSTRAINT (table_object_id, constraint_name)
```

**SYSDBFIL 系统视图**

SYSDBFIL 系统视图中的每一行都描述一个 dbspace 文件。该视图的基础系统表为 ISYSDBFIL。

列名	数据类型	说明
dbfile_id	SMALLINT	仅供内部使用。
dbspace_id	SMALLINT	数据库中的每个 dbspace 文件都有一个指定的唯一编号。system dbspace 包含全部系统对象，且其 dbspace_id 为 0。
dbfile_name	CHAR(128)	dbspace 的文件名。
file_name	LONG VARCHAR	dbspace 的唯一名称。该名称用在 CREATE TABLE 命令中。
lob_map	LONG VARBIT	仅供内部使用。

*基础系统表上的约束*

PRIMARY KEY (dbfile\_id)

FOREIGN KEY (dbspace\_id) REFERENCES SYS.ISYSDBSPACE (dbspace\_id)

UNIQUE index (file\_name)

**SYSDBSpace 系统视图**

SYSDBSpace 系统视图中的每一行都描述了一个 dbspace 文件。该视图的基础系统表为 ISYSDBSpace。

列名	数据类型	说明
dbspace_id	SMALLINT	标识 dbspace 的唯一编号。system dbspace 包含全部系统对象，且其 dbspace_id 为 0。
object_id	UNSIGNED BIGINT	dbspace 的对象 ID。
dbspace_name	CHAR(128)	dbspace 的唯一名称。该名称用在 CREATE TABLE 命令中。
store_type	TINYINT	仅供内部使用。

*基础系统表上的约束*

PRIMARY KEY (dbspace\_id)

FOREIGN KEY (object\_id) REFERENCES SYS.ISYSOBJECT (object\_id) MATCH  
UNIQUE FULL**SYSDBSpacePERM 系统视图**

SYSDBSpacePERM 系统视图中的每一行都描述了 dbspace 文件的特权。该视图的基础系统表为 ISYSDBSpacePERM。

列名	数据类型	说明
dbspace_id	SMALLINT	标识 dbspace 的唯一编号。system dbspace 包含全部系统对象，且其 dbspace_id 为 0。
grantee	UNSIGNED INT	获得特权的用户的用户 ID。
privilege_type	SMALLINT	授予被授予者的特权。例如，CREATE 授予被授予者在 dbspace 上创建对象的特权。

*基础系统表上的约束*

FOREIGN KEY (dbspace\_id) REFERENCES SYS.ISYSDBSPACE (dbspace\_id)

FOREIGN KEY (grantee) REFERENCES SYS.ISYSUSER (user\_id)

**SYSDEPENDENCY 系统视图**

**SYSDEPENDENCY** 系统视图中的各行用来描述两个数据库对象间的依赖性。该视图的基础系统表为 **ISYSDEPENDENCY**。

当一个数据库对象引用其定义中的另一个对象时，这两个数据库对象之间就存在依赖性。例如，如果视图的查询说明引用一个表，则该视图依赖于该表。数据库服务器将跟踪视图对表、视图、实例化视图和列的依赖性。

列名	数据类型	说明
ref_object_id	UNSIGNED BIGINT	被引用对象的对象 ID。
dep_object_id	UNSIGNED BIGINT	引用对象的 ID。

*基础系统表上的约束*

```
PRIMARY KEY (ref_object_id, dep_object_id)
```

```
FOREIGN KEY (ref_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
FOREIGN KEY (dep_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

**SYSDOMAIN 系统视图**

**SYSDOMAIN** 系统视图记录有关内置数据类型（也称为域）的信息。在正常操作期间，该视图的内容不发生更改。该视图的基础系统表为 **ISYSDOMAIN**。

列名	数据类型	说明
domain_id	SMALLINT	指派给每种数据类型的唯一编号。无法更改这些编号。
domain_name	CHAR(128)	通常在 <b>CREATE TABLE</b> 命令中使用的数据类型（如 <b>CHAR</b> 或 <b>INTEGER</b> ）的名称。
type_id	SMALLINT	ODBC 数据类型。该值与 Transact-SQL 兼容性 <b>dbo.SYSTYPES</b> 表中的 <b>data_type</b> 的值相对应。
"precision"	SMALLINT	使用该数据类型可以存储的有效位数。对于非数字数据类型，列值为 <b>NULL</b> 。

*基础系统表上的约束*

```
PRIMARY KEY (domain_id)
```

**SYSEVENT 系统视图**

**SYSEVENT** 系统视图中的每一行都描述一个使用 **CREATE EVENT** 创建的事件。该视图的基础系统表为 **ISYSEVENT**。

列名	数据类型	说明
event_id	UNSIGNED INT	指派给每个事件的唯一编号。



列名	数据类型	说明
object_id	UNSIGNED BIGINT	事件的内部 ID，在数据库中唯一标识事件。
creator	UNSIGNED INT	事件所有者的用户号。通过查看 SYSUSER 系统视图中可以找到的用户的名称。
event_name	VARCHAR(128)	事件的名称。
enabled	CHAR(1)	指出是否允许触发事件。
location	CHAR(1)	将要触发事件的位置：  Y = 指定 AT ALL 子句和 FOR PRIMARY 子句 E = 指定 AT CONSOLIDATED 子句和 FOR PRIMARY 子句 T = 指定 AT REMOTE 子句和 FOR PRIMARY 子句 P = (未指定 AT 子句) 指定 FOR PRIMARY 子句 B = 指定 AT ALL 子句和 FOR ALL 子句 D = 指定 AT CONSOLIDATED 子句和 FOR ALL 子句 S = 指定 AT REMOTE 子句和 FOR ALL 子句 M = (未指定 AT 子句) 指定 FOR ALL 子句 C = AT CONSOLIDATED (未指定 FOR 子句) R = AT REMOTE (未指定 FOR 子句) A = AT ALL 子句 (未指定 FOR 子句)
event_type_id	UNSIGNED INT	对于系统事件，指 SYSEVENTTYPE 中所列的事件类型。
action	LONG VARCHAR	事件处理程序定义。模糊值指示隐藏事件。
external_action	LONG VARCHAR	仅供系统使用。
condition	LONG VARCHAR	用于控制事件处理程序触发的条件。
remarks	LONG VARCHAR	事件的注释；该列来自 ISYSREMARK。
source	LONG VARCHAR	事件的原始来源；该列来自 ISYSSOURCE。

### 基础系统表上的约束

```
PRIMARY KEY (event_id)
```

```
FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
UNIQUE INDEX (event_name)
```

**SYSEVENTTYPE 系统视图**

SYSEVENTTYPE 系统视图定义可由 CREATE EVENT 引用的系统事件类型。

列名	数据类型	说明
event_type_id	INT	指派给每种事件类型的唯一编号。
name	VARCHAR(32000)	系统事件类型的名称。
description	VARCHAR(32000)	系统事件类型的说明。

*注释*

SYSEVENTTYPE 系统视图是使用 sa\_rowgenerator 和以下服务器属性的组合进行定义：

EventTypeName  
EventTypeDef  
MaxEventType

**SYSEXTERNENV 系统视图**

支持很多外部运行时环境，包括以 C/C++ 编写的嵌入式 SQL 和 ODBC 应用程序，以及以 Java、Perl、PHP 或 C# 和 Visual Basic 等基于 Microsoft .NET Framework 公共语言运行库 (CLR) 的语言编写的应用程序。

SYSEXTERNENV 系统视图中的各行描述了标识和启动每个外部环境所需的信息。该视图的基础系统表为 ISYSEXTERNENV。

列名	数据类型	说明
object_id	UNSIGNED BIGINT	外部环境的唯一标识符。
name	CHAR(128)	此列标识外部环境或语言的名称。即 java、perl、php、clr、c_esql32、c_esql64、c_odbc32 或 c_odbc64。
scope	CHAR(1)	<p>此列为 C 或 D，分别代表 CONNECTION (连接) 或 DATABASE (数据库)。scope 列用来标识外部环境是与连接一一对应还是与数据库一一对应。</p> <p>对于与连接一一对应的外部环境 (如 PERL、PHP、C_ESQL32、C_ESQL64、C_ODBC32 和 C_ODBC64)，针对使用外部环境的每个连接会有一个外部环境实例。如果是与连接一一对应，则连接终止时外部环境即终止。</p> <p>对于与数据库一一对应的外部环境 (如 JAVA 和 CLR)，针对使用外部环境的每个数据库会有一个外部环境实例。数据库停止时，与数据库一一对应的外部环境将终止。</p>

列名	数据类型	说明
support_result_sets	CHAR(1)	此列标识那些可以返回结果集的外部环境。除了 PERL 和 PHP，所有外部环境都可以返回结果集。
location	LONG VARCHAR	此列标识数据库服务器计算机上可以找到外部环境可执行文件/二进制文件的位置。它包括可执行文件名/二进制文件名。此路径可以是完全限定路径，也可以是相对路径。如果是相对路径，则可执行文件/二进制文件必须位于数据库服务器可以找到的位置。
options	LONG VARCHAR	此列标识命令行上启动与外部环境相关联的可执行文件所需的任何选项。您不能修改此列。
user_id	UNSIGNED INT	最初启动外部环境时，必须建立返回到数据库的连接，以为外部环境的使用做好准备。缺省情况下，此连接会使用该 DBA 用户 ID 建立，但如果数据库管理员倾向于令外部环境使用另一个具有 MANAGE ANY EXTERNAL OBJECT 系统特权的用户 ID，user_id 列就会指示该不同的用户 ID。通常，此列为 NULL，数据库服务器会缺省使用这里的 DBA 用户 ID。

### 基础系统表上的约束

```
PRIMARY KEY (object_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)
```

```
UNIQUE INDEX (name)
```

### **SYSEXTERNENVOBJECT** 系统视图

支持很多外部运行时环境，包括以 C/C++ 编写的嵌入式 SQL 和 ODBC 应用程序，以及以 Java、Perl、PHP 或 C# 和 Visual Basic 等基于 Microsoft .NET Framework 公共语言运行库 (CLR) 的语言编写的应用程序。

SYSEXTERNENVOBJECT 系统视图中的每一行都描述已安装的外部对象。该视图的基础系统表为 ISYSEXTERNENVOBJECT。

列名	数据类型	说明
object_id	UNSIGNED BIGINT	外部对象的唯一标识符。
extenv_id	UNSIGNED BIGINT	外部环境的唯一标识符 (SYSEXTERNENV.object_id)。
owner	UNSIGNED INT	此列标识外部对象的创建者/所有者。
name	LONG VARCHAR	此列标识在 INSTALL EXTERNAL OBJECT 语句中指定的外部对象的名称。

列名	数据类型	说明
contents	LONG BINARY	外部对象的内容。
update_time	TIMESTAMP	此列标识上次修改（或安装）对象的时间。
update_time_utc	TIMESTAMP WITH TIME ZONE	此列标识上次修改（或安装）对象的 UTC 时间。

*基础系统表上的约束*

PRIMARY KEY (object\_id)

FOREIGN KEY (object\_id) REFERENCES SYS.ISYSOBJECT (object\_id) MATCH UNIQUE FULL

FOREIGN KEY (extenv\_id) REFERENCES SYS.ISYSEXTERNENV (object\_id)

FOREIGN KEY (owner) REFERENCES SYS.ISYSUSER (user\_id)

UNIQUE INDEX (name)

**SYSEXTERNLOGIN 系统视图**

SYSEXTERNLOGIN 系统视图的每行都描述一个用于远程数据访问的外部登录。该视图的基础系统表为 ISYSEXTERNLOGIN。

**注意：** 以前版本的目录中包含一个 SYSEXTERNLOGINS 系统表。该表已重命名为 ISYSEXTERNLOGIN（无 'S'），并且是本视图的基础表。

列名	数据类型	说明
user_id	UNSIGNED INT	本地数据库上的用户 ID。
srvid	UNSIGNED INT	SYSSERVER 系统视图中所列的远程服务器。
remote_login	VARCHAR(128)	用户登录远程服务器的登录名。
remote_password	VARBINARY(128)	用户登录远程服务器的口令。

*基础系统表上的约束*

PRIMARY KEY (user\_id, srvid)

FOREIGN KEY (user\_id) REFERENCES SYS.ISYSUSER (user\_id)

FOREIGN KEY (srvid) REFERENCES SYS.ISYSSERVER (srvid)

**SYSFILE 兼容性视图 (不建议使用)**

**SYSFILE** 系统视图中的每一行都描述数据库的一个 **dbspace**。每个数据库都由一个或多个 **dbspace** 组成；每个 **dbspace** 都对应一个操作系统文件。

自动为主数据库文件、临时文件、事务日志文件和事务日志镜像文件创建 **dbspace**。有关事务日志和事务日志镜像 **dbspace** 的信息未出现在 **SYSFILE** 系统视图中。

```
ALTER VIEW "SYS"."SYSFILE"
  as select b.dbfile_id as file_id,
    if b.dbSPACE_id = 0 and b.dbfile_id = 0 then
      db_property('File')
    else
      if b.dbSPACE_id = 15 and b.dbfile_id = 15 then
        db_property('TempFileName')
      else
        b.file_name
      endif
    endif as file_name,
  a.dbSPACE_name,
  a.store_type,
  b.lob_map,
  b.dbSPACE_id
from SYS.ISYSDBSPACE as a
  join SYS.ISYSDBFILE as b on(a.dbSPACE_id = b.dbSPACE_id)
```

**SYSFKCOL 兼容性视图 (不建议使用)**

**SYSFKCOL** 的每一行都描述关系中外表的外列与主表的主列之间的关联。不建议使用该视图；请改用 **SYSIDX** 和 **SYSIDXCOL** 系统视图。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSFKCOL"
  as select a.table_id as foreign_table_id,
    a.index_id as foreign_key_id,
    a.column_id as foreign_column_id,
    a.primary_column_id
  from SYS.ISYSIDXCOL as a
    ,SYS.ISYSIDX as b
  where a.table_id = b.table_id
    and a.index_id = b.index_id
    and b.index_category = 2
```

**SYSFKEY 系统视图**

**SYSFKEY** 系统视图的每行描述系统中的一个外键约束。该视图的基础系统表为 **ISYSFKEY**。

列名	数据类型	说明
foreign_table_id	UNSIGNED INT	外表的表号。
foreign_index_id	UNSIGNED INT	外键的索引号。

列名	数据类型	说明
primary_table_id	UNSIGNED INT	主表的表号。
primary_index_id	UNSIGNED INT	主键的索引号。
match_type	TINYINT	约束的匹配类型。匹配类型包括： <ul style="list-style-type: none"> <li>• 0 - 使用缺省匹配</li> <li>• 1 - SIMPLE</li> <li>• 2 - FULL</li> <li>• 129 - SIMPLE UNIQUE</li> <li>• 130 - FULL UNIQUE</li> </ul>
check_on_commit	CHAR(1)	指出 INSERT 和 UPDATE 语句是否应等到 COMMIT 时才检查外键是否仍然有效。
nulls	CHAR(1)	指出是否允许外键中的列包含 NULL 值。该设置与外键中所包含列的 nulls 设置无关。

### 基础系统表上的约束

```
PRIMARY KEY (foreign_table_id, foreign_index_id)
```

```
FOREIGN KEY (foreign_table_id, foreign_index_id) REFERENCES  
SYS.ISYSIDX (table_id, index_id)
```

```
FOREIGN KEY (primary_table_id, primary_index_id) REFERENCES  
SYS.ISYSIDX (table_id, index_id)
```

### **SYSPFOREIGNKEY 兼容性视图 (不建议使用)**

提供 SYSPFOREIGNKEY 视图的目的是为了与提供 SYSPFOREIGNKEY 系统表的旧版本软件兼容。但以前的 SYSPFOREIGNKEY 系统表已被 ISYSFKKEY 系统表所取代，并且您应当使用与该系统表对应的 ISYSFKKEY 系统视图。

外键是两个表（外表和主表）之间的关系。每个外键都由 SYSPFOREIGNKEY 中的一行和 SYSPFKCOL 的一行或多行定义。SYSPFOREIGNKEY 包含有关外键的一般信息，而 SYSPFKCOL 标识外键中的列并将外键中每个列与主表的主键中的一列相关联。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSPFOREIGNKEY"  
as select b.foreign_table_id,  
b.foreign_index_id as foreign_key_id,  
a.object_id,  
b.primary_table_id,  
p.root,  
b.check_on_commit,  
b.nulls,  
a.index_name as role,  
r.remarks,
```

```

    b.primary_index_id,
    a.not_enforced as fk_not_enforced,
    10 as hash_limit
  from(SYS.ISYSIDX as a left outer join SYS.ISYSPHYSIDX as p
on(a.table_id = p.table_id and a.phys_index_id = p.phys_index_id))
    left outer join SYS.ISYSREMARK as r on(a.object_id =
r.object_id)
    ,SYS.ISYSFKEY as b
  where a.table_id = b.foreign_table_id
    and a.index_id = b.foreign_index_id

```

### **SYSFOREIGNKEYS 统一视图**

**SYSFOREIGNKEYS** 视图的每一行都描述了目录中每个表的一个外键。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```

ALTER VIEW "SYS"."SYSFOREIGNKEYS"( foreign_creator,
  foreign_tname,
  primary_creator,primary_tname,role,columns )
as select fk_up.user_name,fk_tab.table_name,pk_up.user_name,
  pk_tab.table_name,ix.index_name,
  (select list(string(fk_col.column_name,' IS ',
  pk_col.column_name)
  order by fkc.table_id,fkc.index_id,fkc."sequence")
  from SYS.ISYSIDXCOL as fkc
    join SYS.ISYSTABCOL as fk_col on(
  fkc.table_id = fk_col.table_id
  and fkc.column_id = fk_col.column_id)
    ,SYS.ISYSTABCOL as pk_col
  where fkc.table_id = fk.foreign_table_id
  and fkc.index_id = fk.foreign_index_id
  and pk_col.table_id = fk.primary_table_id
  and pk_col.column_id = fkc.primary_column_id)
  from SYS.ISYSFKEY as fk
    join SYS.ISYSTAB as fk_tab on fk_tab.table_id =
fk.foreign_table_id
    join SYS.ISYSUSER as fk_up on fk_up.user_id = fk_tab.creator
    join SYS.ISYSTAB as pk_tab on pk_tab.table_id =
fk.primary_table_id
    join SYS.ISYSUSER as pk_up on pk_up.user_id = pk_tab.creator
    join SYS.ISYSIDX as ix on ix.table_id = fk.foreign_table_id and
ix.index_id = fk.foreign_index_id

```

### **SYSGROUP 兼容性视图**

对于每组的每一个成员，在 **SYSGROUP** 系统视图中都有与之对应的一行。该视图介绍了组与成员之间的多对多关系。一个组可以有許多成员，而一个用户也可以是许多组的成员。

列名	数据类型	说明
group_id	UNSIGNED INT	组的用户号。
group_member	UNSIGNED INT	成员的用户号。

### 基础系统表上的约束

```
PRIMARY KEY (group_id, group_member)
```

```
FOREIGN KEY group_id (group_id) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY group_member (group_member) REFERENCES SYS.ISYSUSER  
(user_id)
```

### **SYSGROUPS 兼容性视图**

对于每组的每一个成员，在 **SYSGROUPS** 视图中都有与之对应的一行。该视图介绍了组与成员之间的多对多关系。一个组可以有許多成员，而一个用户也可以是许多组的成员。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSGROUPS"( group_name,  
    member_name )  
    as select g.user_name,u.user_name  
        from SYS.ISYSROLEGRANT,SYS.ISYSUSER as g,SYS.ISYSUSER as u  
        where ISYSROLEGRANT.role_id = g.user_id and ISYSROLEGRANT.grantee  
= u.user_id and(  
    u.user_name in( 'SYS_SPATIAL_ADMIN_ROLE' )  
    or u.user_id <= 2147483648) and(  
    g.user_type = (0x02|0x04|0x08)  
    or g.user_name in( 'SYS','PUBLIC','dbo','diagnostics',  
    'rs_systabgroup','SA_DEBUG','SYS_SPATIAL_ADMIN_ROLE' ) )
```



**SYSHISTORY 系统视图**

SYSHISTORY 系统视图的每行记录对数据库的一个系统操作，例如数据库启动、数据库校准等。该视图的基础系统表为 ISYSHISTORY。

列名	数据类型	说明
operation	CHAR(128)	对数据库文件执行的操作类型。操作必须是以下值之一： <ul style="list-style-type: none"> <li>• <b>INIT</b> - 有关何时创建数据库的信息。</li> <li>• <b>UPGRADE</b> - 有关何时升级数据库的信息。</li> <li>• <b>START</b> - 有关何时在特定操作系统下使用特定数据库服务器版本启动数据库的信息。</li> <li>• <b>LAST_START</b> - 有关最近一次启动数据库服务器的时间的信息。启动数据库时，如果所使用的数据库服务器版本和/或所在的操作系统与当前存储在 <b>LAST_START</b> 行中的值不同，<b>LAST_START</b> 操作将会被转换成 <b>START</b> 操作。</li> <li>• <b>DTT</b> - 有关倒数第二次对 <b>dbspace</b> 执行的磁盘传送时间 (Disk Transfer Time, 简称 <b>DTT</b>) 校准操作的信息。即有关倒数第二次执行 <b>ALTER DATABASE CALIBRATE</b> 或 <b>ALTER DATABASE RESTORE DEFAULT CALIBRATION</b> 语句的信息。</li> <li>• <b>LAST_DTT</b> - 有关最新对 <b>dbspace</b> 执行的 <b>DTT</b> 校准操作的信息。即有关最新执行 <b>ALTER DATABASE CALIBRATE</b> 或 <b>ALTER DATABASE RESTORE DEFAULT CALIBRATION</b> 语句的信息。</li> <li>• <b>LAST_BACKUP</b> - 有关上次备份的信息，包括备份的日期和时间、备份类型、备份的文件及执行备份的数据库服务器的版本。</li> </ul>
object_id	UNSIGNED INT	对于 <b>DTT</b> 和 <b>LAST_DTT</b> 以外的任何操作，该列的值都将为 0。对于 <b>DTT</b> 和 <b>LAST_DTT</b> 操作，这是在 <b>SYSDBSpace</b> 系统视图中定义的 <b>dbspace</b> 的 <b>dbspace_id</b> 。
sub_operation	CHAR(128)	对于 <b>DTT</b> 和 <b>LAST_DTT</b> 以外的任何操作，该列的值将是一组空单引号 ("")。对于 <b>DTT</b> 和 <b>LAST_DTT</b> 操作，该列包含对 <b>dbspace</b> 执行的子操作的类型。值包括： <ul style="list-style-type: none"> <li>• <b>DTT_SET</b> - 已设置 <b>dbspace</b> 校准。</li> <li>• <b>DTT_UNSET</b> - 已将 <b>dbspace</b> 校准恢复为缺省设置。</li> </ul>
version	CHAR(128)	用于执行此操作的数据库服务器版本与编译版本。
platform	CHAR(128)	执行该操作的操作系统。
first_time	TIMESTAMP	第一次在某一特定操作系统下以特定软件版本启动数据库的本地日期和时间。

列名	数据类型	说明
last_time	TIMESTAMP	最近一次在某一特定操作系统下以特定软件版本启动数据库的本地日期和时间。
details	LONG VARCHAR	此列存储例如启动数据库服务器所使用的命令行选项或数据库启用的功能位等信息。该信息供技术支持使用。
first_time_utc	TIMESTAMP WITH TIME ZONE	第一次在某一特定操作系统下以特定软件版本启动数据库的 UTC 日期和时间。
last_time_utc	TIMESTAMP WITH TIME ZONE	最近一次在某一特定操作系统下以特定软件版本启动数据库的 UTC 日期和时间。

### 基础系统表上的约束

PRIMARY KEY (operation, object\_id, version, platform)

### **SYSIDX** 系统视图

SYSIDX 系统视图的每行都定义数据库中的一个逻辑索引。该视图的基础系统表为 ISYSIDX。

列名	数据类型	说明
table_id	UNSIGNED INT	唯一标识包含该索引的表。
index_id	UNSIGNED INT	标识表中索引的唯一编号。
object_id	UNSIGNED BIGINT	索引的内部 ID，在数据库中唯一标识索引。
phys_index_id	UNSIGNED INT	标识用于实现逻辑索引的基础物理索引。对于临时表或远程表上的索引，该值为 NULL。否则，该值与 <b>SYSPHYSIDX</b> 系统视图窗中物理索引的 object_id 相对应。
dbspace_id	SMALLINT	包含索引的文件的 ID。该值对应 <b>SYSDBSPACE</b> 系统视图中的一个条目。
index_category	TINYINT	索引的类型。值包括： <ul style="list-style-type: none"> <li>• 1 - 主键</li> <li>• 2 - 外键</li> <li>• 3 - 次级索引（包括唯一约束）</li> <li>• 4 - 文本索引</li> </ul>

列名	数据类型	说明
"unique"	TINYINT	指出索引是唯一索引 (1)、唯一约束 (2)、保留 (3)、非唯一索引 (4) 还是具有 WITH NULLS NOT DISTINCT 的唯一索引。唯一索引可以避免索引表中的两行在索引列中具有相同的值。
index_name	CHAR(128)	索引的名称。
not_enforced	CHAR(1)	仅供系统使用。
file_id	SMALLINT	不建议使用。此列存在于 SYSVIEW 中，但不存在于基础系统表 ISYSIDX 中。此列的内容与 dbspace_id 相同，并且是为了兼容而提供的。请改用 dbspace_id。

### 基础系统表上的约束

```
PRIMARY KEY (table_id, index_id)
```

```
FOREIGN KEY (table_id) REFERENCES SYS.ISYSTAB (table_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (table_id, phys_index_id) REFERENCES SYS.ISYSPHYSIDX  
(table_id, phys_index_id)
```

```
UNIQUE INDEX (index_name, table_id, index_category)
```

### **SYSIDXCOL** 系统视图

**SYSIDXCOL** 系统视图的每行描述在 **SYSIDX** 系统视图中描述的索引的一列。该视图的基础系统表为 **ISYSIDXCOL**。

列名	数据类型	说明
table_id	UNSIGNED INT	标识对其应用了该索引的表。
index_id	UNSIGNED INT	标识对其应用该列的索引。table_id 与 index_id 共同标识 <b>SYSIDX</b> 系统视图中描述的一个索引。
sequence	SMALLINT	索引中的每一列都有一个指定的从 0 开始的唯一编号。这些编号的顺序决定了列在索引中的相对重要性。最重要的列的序列号为 0。
column_id	UNSIGNED INT	标识表中的哪列被索引。table_id 与 column_id 共同标识 <b>SYSCOLUMN</b> 系统视图中描述的一列。
"order"	CHAR(1)	指出该列在索引中是升序 (A) 还是降序 (D)。对于文本索引此值为 NULL。
primary_col- umn_id	UNSIGNED INT	与该外键列相对应的主键列的 ID。对于非外键列，值为 NULL。

*基础系统表上的约束*

```
PRIMARY KEY (table_id, index_id, column_id)
```

```
FOREIGN KEY (table_id, index_id) REFERENCES SYS.ISYSIDX (table_id, index_id)
```

```
FOREIGN KEY (table_id, column_id) REFERENCES SYS.ISYSTABCOL (table_id, column_id)
```

**SYSINDEX 兼容性视图 (不建议使用)**

提供 **SYSCOLUMN** 视图的目的是为了与提供 **SYSINDEX** 系统表的旧版本软件兼容。但以前的 **SYSINDEX** 系统表已被 **ISYSIDX** 系统表所取代，并且您应当使用与该系统表对应的 **ISYSIDX** 系统视图。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSINDEX"
as select b.table_id,
  b.index_id,
  b.object_id,
  p.root,
  b.dbSPACE_id,
  case b."unique"
  when 1 then 'Y'
  when 2 then 'U'
  when 3 then 'M'
  when 4 then 'N'
  when 5 then 'Y'
  else 'I'
  end as "unique",
  t.creator,
  b.index_name,
  r.remarks,
  10 as hash_limit,
  b.dbSPACE_id as file_id
  from(SYS.ISYSIDX as b left outer join SYS.ISYSINDEX as p
on(b.table_id = p.table_id and b.phys_index_id = p.phys_index_id))
  left outer join SYS.ISYSREMARK as r on(b.object_id =
r.object_id)
  ,SYS.ISYSTAB as t
  where t.table_id = b.table_id
  and b.index_category = 3
```

**SYSINDEXES 统一视图**

**SYSINDEXES** 视图的每一行都描述数据库中的一个索引。您可以使用 **SYSIDX** 和 **SYSIDXCOL** 系统视图替代该视图。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSINDEXES"( icreator,
  iname, fname, creator, tname, indextype,
```

```

colnames,interval,level_num )
as select u.user_name,idx.index_name,dbs.dbspace_name,u.user_name,
tab.table_name,
case idx.index_category
when 1 then 'Primary Key'
when 2 then 'Foreign Key'
when 3 then(
if idx."unique" = 4 then 'Non-unique'
else if idx."unique" = 2 then 'UNIQUE constraint'
else if idx."unique" = 5 then 'UNIQUE NULLS NOT DISTINCT'
else 'UNIQUE'
endif
endif
endif) when 4 then 'Text Index' end,(select
list(string(c.column_name,
if icx."order"='A' then 'ASC' else 'DESC' endif) order by
icx.table_id asc,icx.index_id asc,icx.sequence asc)
from SYS.ISYSIDXCOL as icx
join SYS.ISYSTABCOL as c on(
c.table_id = icx.table_id
and c.column_id = icx.column_id)
where icx.index_id = idx.index_id
and icx.table_id = idx.table_id),
0,0
from SYS.ISYSTAB as tab
join SYS.ISYSDBSPACE as dbs on(tab.dbspace_id = dbs.dbspace_id)
join SYS.ISYSIDX as idx on(idx.table_id = tab.table_id)
join SYS.ISYSUSER as u on u.user_id = tab.creator

```

### **SYSINDEXES ASE 兼容性视图**

每个聚簇索引、每个非聚簇索引、每个没有聚簇索引的表以及每个包含文本或图像列的表在 `sysindexes` 中都有对应的一行。

每个基于函数的索引或在计算列上创建的索引在此表中都有对应的一行。

此视图由用户 `DBO` 所有。

### **另请参见**

- 每个 Adaptive Server 数据库中的表 (第 751 页)
- `SYSCOLUMNS ASE` 兼容性视图 (第 665 页)
- `SYS COMMENTS ASE` 兼容性视图 (第 665 页)
- `SYSIQOBJECTS ASE` 兼容性视图 (第 691 页)
- `SYSIQVINDEXT ASE` 兼容性视图 (第 695 页)
- `SYSOBJECTS ASE` 兼容性视图 (第 702 页)
- `SYSTYPES ASE` 兼容性视图 (第 741 页)
- `SYSUSERS ASE` 兼容性视图 (第 747 页)

### **SYSINFO 兼容性视图 (不建议使用)**

SYSINFO 视图指出了创建数据库时定义的数据库特性。它始终只有一行。该视图可通过内置函数来获得且不是保存在目录中。以下是 SYSINFO 视图的定义：

```
ALTER VIEW "SYS"."SYSINFO"( page_size,
    encryption,
    blank_padding,
    case_sensitivity,
    default_collation,
    database_version )
as select db_property('PageSize'),
    if db_property('Encryption') <> 'None' then 'Y' else 'N' endif,
    if db_property('BlankPadding') = 'On' then 'Y' else 'N' endif,
    if db_property('CaseSensitive') = 'On' then 'Y' else 'N' endif,
    db_property('Collation'),
    NULL
```

### **SYSIQBACKUPHISTORY 系统视图**

此视图以可读格式显示 ISYSIQBACKUPHISTORY 中的组信息。此视图中的每一行均描述了一个成功完成的特定备份操作。

SYSIQBACKUP 视图反映类型、子类型和 bkp\_virtual 列的等效字符串值。

列名	列类型	列约束	描述
bu_id	unsigned bigint	NOT NULL	操作检查点的事务标识符。备份操作的备份 ID。
bu_time	timestamp	NOT NULL	备份记录中记录的备份操作的时间。
type	tinyint	NOT NULL	备份类型：0 = FULL 1 = INCREMENTAL 2 = INCREMENTAL SINCE FULL
selective_type	tinyint	NOT NULL	备份子类型：0 = ALL (备份所有 dbfile) 1 = READ/WRITE ONLY (备份所有读写文件) 2 = READ ONLY (备份特定的只读文件)
virtual_type	tinyint	NOT NULL	备份虚拟类型：0 = NONE 1 = DECOUPLED 2 = ENCAPSULATED
dependson_id	unsigned bigint	NULL	对 FULL 备份为 NULL
cmd	long varchar	NOT NULL	命令的完整文本
creator	char(128)	NOT NULL	发出备份命令的用户
version	unsigned int	NOT NULL	备份版本

基础系统表的约束：

主键 (bu\_id)

另请参见

- `sp_iqbackupdetails` 过程 (第 387 页)

### **SYSIQBACKUPHISTORYDETAIL** 系统视图

此视图描述备份时数据库中存在的 `dbfile` 记录。此视图中的每一行均描述了一个成功完成的特定备份操作。

此视图以可读格式显示 `ISYSIQBACKUPHISTORYDETAIL` 中的组信息。每个列的列约束都为 `NOT NULL`。

列名	列类型	描述
<code>bu_id</code>	<code>unsigned bigint</code>	操作检查点的事务标识符。备份操作的备份 ID。
<code>dbspace_id</code>	<code>smallint</code>	与此 <code>dbfile</code> 记录关联的 <code>dbspace</code> ID。
<code>dbfile_id</code>	<code>smallint</code>	持续备份操作过程中, <code>dbspace</code> 中存在的 <code>dbfile</code> ID
<code>dbspace_rwstatus</code>	<code>char(1)</code>	T 表示读写
<code>dbspace_createid</code>	<code>unsigned bigint</code>	创建 <code>dbspace</code> 的事务的事务 ID
<code>dbspace_alterid</code>	<code>unsigned bigint</code>	将 <code>dbspace</code> 标记为 RO 的事务 ID。如果未标记, 则为创建 ID
<code>dbspace_online</code>	<code>char(1)</code>	T 表示联机
<code>dbfile_rwstatus</code>	<code>char(1)</code>	T 表示读写
<code>dbfile_createid</code>	<code>unsigned bigint</code>	创建此 <code>dbfile</code> 的事务的事务 ID
<code>dbfile_alterid</code>	<code>unsigned bigint</code>	最后一个更改此 <code>dbfile</code> 的读写状态的事务的事务 ID
<code>is_backed_up</code>	<code>char(1)</code>	表示 <code>dbfile</code> 是在此备份中备份的
<code>start_block</code>	<code>unsigned bigint</code>	<code>dbfile</code> 的起始块
<code>num_blocks</code>	<code>unsigned bigint</code>	<code>dbfile</code> 中块的总数
<code>num_blocks_backed_up</code>	<code>unsigned bigint</code>	已备份的块的总数
<code>dbspace_name</code>	<code>char(128)</code>	<code>dbspace</code> 名称
<code>dbfile_name</code>	<code>char(128)</code>	<code>dbfile</code> 的逻辑文件名
<code>dbfile_path</code>	<code>long varchar</code>	文件的物理路径

基础系统表的约束:

主键 (`bu_id`、`dbfile_id`)

外键 (`txn_id`) 引用 `SYS.ISYSBACKUPHISTORY`

**SYSIQCOLUMN 系统视图 (不建议使用)**

SYSIQCOLUMN 已由 SYSIQTABCOL 系统视图取代。

另请参见

- SYSIQTABCOL 系统视图 (第 694 页)

**SYSIQDBFILE 系统视图**

以可读格式显示 ISYSIQDBFILE 中的组信息。

**注意：** 此视图将取代不建议使用的 SYSIQFILE 系统视图。

列名	列类型	描述
dbfile_id	small int	dbfile 的唯一 ID
start_block	rowid	第一个块的编号
block_count	rowid	此文件 (dbspace) 的块数
reserve_size	rowid	dbspace 的预分配文件系统空间
allocated	char(1)	定义段是预先分配 (T) 还是自动分配 (F)
data_offset	unsigned int	标识 SAP Sybase IQ 数据开始的字节位置, 此位置相对于原始分区的开头。
create_time	timestamp	创建文件的日期和时间
last_modified	timestamp	上次修改文件的日期和时间
read_write	char(1)	T 表示读写
online	char(1)	T 表示联机
create_txn_id	unsigned bigint	创建 dbfile 的事务 ID
alter_txn_id	unsigned bigint	上次修改读写状态的事务 ID
server_id	unsigned int	Multiplex 服务器名
file_name	text	Multiplex 服务器用于打开的 dbspace 的名称

基础系统表的约束:

外键 (server\_id) 引用 SYS.ISYSIQMPXSERVER

唯一 (server\_id、file\_name)

另请参见

- SYSIQFILE 系统视图 (不建议使用) (第 685 页)



**SYSIQDBSPACE 系统视图**

以可读格式显示 ISYSIQDBSPACE 中的组信息。

列名	列类型	描述
dbspace_id	small int	数据库中的每个 dbspace 都有一个指定的唯一编号 (dbspace ID)
last_modified	timestamp	上次修改 dbspace 的读写状态的时间
segment_type	char(8)	段类型: Main、Temp 或 Msg
read_write	char(1)	'T' - 可读写; 'F' - 只读
online	char(1)	'T' - 联机; 'F' - 脱机
create_txn_id	unsigned bigint	创建 dbspace 的事务 ID
alter_txn_id	unsigned bigint	上次修改 read_write 状态的事务 ID
striping_on	char(1)	'T' - 磁盘条带化开; 'F' - 磁盘条带化关
stripe_size_kb	unsigned int	在磁盘条带化算法移动到下一个 dbfile 之前要写入 dbspace 的每个文件的字节数 (KB)
is_rlv_store	char(1)	'T' - dbspace 是一个 RLV 存储 dbspace; 'F' - dbspace 是一个 MIN、SHARED_TEMP 或 TEMPORARY 存储 dbspace。

基础系统表的约束:

主键 (dbspace\_id)

外键 (dbspace\_id) 引用 SYS.ISYSDBSPACE(dbspace\_id)

**SYSIQFILE 系统视图 (不建议使用)**

SYSIQFILE 已由 SYSIQDBFILE 系统视图取代。

另请参见

- SYSIQDBFILE 系统视图 (第 684 页)

**SYSIQIDX 系统视图**

以可读格式显示 ISYSIQIDX 中的组信息。SYSIQIDX 视图中的每行均描述一个 IQ 索引。

**注意:** 此视图将取代不建议使用的 SYSIQINDEX 系统视图。

列名	列类型	描述
table_id	unsigned int	表号唯一标识了该索引所应用的表

列名	列类型	描述
index_id	unsigned int	一个特定表的每条索引都有一个指定的唯一索引号
index_type	char(4)	索引类型
index_owner	char(4)	索引所有者
max_key	unsigned int	供内部使用
identity_location	hs_vdorecid	供内部使用
identity_size	unsigned int	供内部使用
identity_location_size	unsigned int	供内部使用
link_index_id	unsigned int	供内部使用
delimited_by	varchar(1024)	(仅限 WD 索引) 用于将列的字符串解析为要存储在该列的 WD 索引中的单词的分隔符列表
limit	unsigned int	(仅限 WD 索引) WD 索引的最大单词长度

基础系统表的约束:

主键 (table\_id、 index\_id)

外键 (table\_id、 index\_id) 引用 SYS.ISYIDX

外键 (link\_table\_id、 link\_index\_id、 table\_id、 index\_id) 引用 SYS.ISYSIDX

### **SYSIQINFO** 系统视图

以可读格式显示 ISYSIQINFO 中的组信息。

使用 **CREATE DATABASE** 创建 SAP Sybase IQ 数据库时, ISYSIQINFO 系统表指示数据库特性为定义特性。它始终只有一行。

列名	列类型	描述
create_time	TIMESTAMP NOT NULL	创建数据库的日期和时间
update_time	TIMESTAMP NOT NULL	最后一次更新的日期和时间
file_format_version	UNSIGNED INT NOT NULL	此数据库的文件的文件格式编号
cat_format_version	UNSIGNED INT NOT NULL	此数据库的目录格式编号
sp_format_version	UNSIGNED INT NOT NULL	此数据库的存储过程格式编号

列名	列类型	描述
block_size	UNSIGNED INT NOT NULL	为数据库指定的块大小
chunk_size	UNSIGNED INT NOT NULL	由块大小确定的每一页的块数以及为数据库指定的页大小
file_format_date	CHAR(10) NOT NULL	上次更改文件格式编号的日期
dbsig	BINARY(136) NOT NULL	供目录在内部使用
commit_txn_id	unsigned bigint	供内部使用
rd_commit_txn_id	unsigned bigint	供内部使用
multiplex name	CHAR(128) NULL	此数据库是其成员的 Multiplex 的名称
last_multiplex_mode	TINYINT NULL	(SAP Sybase IQ 16.0 中未使用的列) 最后打开目录读写的服务器模式。以下值之一。 <ul style="list-style-type: none"> <li>• 0 - 单节点。</li> <li>• 1 - 读取器。</li> <li>• 2 - 事务协调器。</li> <li>• 3 - 写入器。</li> </ul>

### **YSIQLOGICALSERVER 系统视图**

呈现 ISYSIQLOGICALSERVER 系统表的可读版本。

ISYSIQLOGICALSERVER 系统表存储逻辑服务器信息和关联逻辑服务器策略信息。

列名	列类型	描述
ls_id	UNSIGNED BIGINT NOT NULL	逻辑服务器的 ID 号。
ls_object_id	UNSIGNED BIGINT NOT NULL	逻辑服务器对象 ID 号。
ls_policy_id	UNSIGNED BIGINT NOT NULL	逻辑服务器策略的 ID 号。
ls_name	CHAR(128) NOT NULL UNIQUE	逻辑服务器名称。

基础系统表上的约束：

- 主键 (ls\_id)
- object\_id 外键 (ISYSOBJECT)
- ls\_policy\_id 外键 (ISYSIQLSPOLICY)

**SYSIQLOGINPOLICYLSINFO 系统视图**

呈现 ISYSIQLOGINPOLICYLSINFO 表的可读版本。

ISYSIQLOGINPOLICYLSINFO 系统表存储登录策略逻辑服务器分配信息。

列名	列类型	描述
login_policy_id	UNSIGNED BIGINT NOT NULL	登录策略的 ID 号。
ls_id	UNSIGNED BIGINT NOT NULL	逻辑服务器的 ID 号。

基础系统表上的约束：

- 主键 (login\_policy\_id、ls\_id)
- login\_policy\_id 外键 (ISYSLOGINPOLICY)
- ls\_id 外键 (ISYSIQLOGICALSERVER)

**SYSIQLSLOGINPOLICIES 统一视图**

描述了基于登录策略的所有逻辑服务器分配。

该统一系统视图显示来自 SYSIQLOGICALSERVER、ISYSIQLOGINPOLICYLSINFO 和 ISYSLOGINPOLICY 的信息。

列名	列类型	描述
ls_id	UNSIGNED BIGINT NOT NULL	逻辑服务器标识符。
ls_name	CHAR(128)	逻辑服务器名称。
login_policy_id	UNSIGNED BIGINT NOT NULL	登录策略的 ID 号。
login_policy_name	char(128)	登录策略的名称。

**SYSIQLSLOGINPOLICYOPTION 系统视图**

呈现表 ISYSIQLSLOGINPOLICYOPTION 的可读格式版本。

ISYSIQLSLOGINPOLICYOPTION 表存储登录策略选项值的逻辑服务器级设置。

列名	列类型	描述
login_policy_id	UNSIGNED BIGINT NOT NULL	登录策略的 ID 号。
ls_id	UNSIGNED BIGINT NOT NULL	逻辑服务器标识符。
login_option_name	CHAR(128) NOT NULL	登录策略选项的名称。
login_option_value	LONG VARCHAR NOT NULL	登录策略选项的值。

基础系统表上的约束：

- 主键 (login\_policy\_id、ls\_id、login\_option\_name)
- login\_policy\_id 外键 (ISYSLOGINPOLICY)
- ls\_id 外键 (ISYSIQLOGICALSERVER)

### **SYSIQLSMEMBER 系统视图**

显示用于存储逻辑服务器成员资格信息的 ISYSIQLSMEMBER 表中的组信息。

ISYSIQLSMEMBER 存储逻辑服务器及其对应的 Multiplex 服务器。

列名	列类型	描述
ls_id	UNSIGNED BIGINT NOT NULL	逻辑服务器的 ID 号。
logical_membership_type	TINYINT NOT NULL	逻辑成员资格的类型。
mpx_server_id	UNSIGNED INT NOT NULL	Multiplex 服务器的 ID 号。
membership_info	UNSIGNED INT NOT NULL	成员资格信息。

基础系统表上的约束：

- 主键 (ls\_id、logical\_membership\_id、mpx\_server\_id)
- ls\_id 外键 (ISYSIQLOGICALSERVER)

对于使用 Multiplex 服务器名定义的逻辑服务器成员资格，logical\_membership\_type 值为 0，mpx\_server\_id 值为该 Multiplex 服务器的 ID。

对于协调器的逻辑成员资格，mpx\_server\_id 值为 0，logical\_membership\_type 值为 1。

### **SYSIQLSMEMBERS 统一视图**

描述了所有用户定义的逻辑服务器成员资格。

列名	列类型	描述
ls_id	UNSIGNED BIGINT NOT NULL	逻辑服务器的 ID 号。
ls_name	CHAR(128) NOT NULL	逻辑服务器的名称。
server_id	UNSIGNED INT NOT NULL	对于使用服务器名称定义的成员资格，为成员的 Multiplex 服务器标识符；对于协调器的逻辑成员资格，为 0。
server_name	CHAR(128) NOT NULL	对于使用服务器名称定义的成员资格，为成员的 Multiplex 服务器名称；对于协调器的逻辑成员资格，为 'LOGICAL COORDINATOR'。
membership_type	TINYINT NOT NULL	对于使用服务器名称定义的成员资格，为 0；对于协调器的逻辑成员资格，为 1。

### **SYSIQLSPOLICY 系统视图**

呈现 ISYSIQLSPOLICY 表的可读格式版本。

ISYSIQLSPOLICY 系统表存储逻辑服务器策略。

列名	列类型	描述
ls_policy_Id	UNSIGNED BIGINT NOT NULL	逻辑服务器策略的 ID 号。
ls_policy_name	CHAR(128) NOT NULL UNIQUE	逻辑服务器策略名称。

基础系统表上的约束：

- 主键 (ls\_policy\_id)
- object\_id 外键 (ISYSOBJECT)

### **SYSIQLSPOLICYOPTION 系统视图**

呈现表 ISYSIQLSPOLICYOPTION 的可读格式版本。

ISYSIQLSPOLICYOPTION 表存储逻辑服务器策略选项。

列名	列类型	描述
ls_policy_id	UNSIGNED BIGINT NOT NULL	登录策略的 ID 号。
ls_policy_option_name	CHAR(128) NOT NULL	逻辑服务器策略选项名称。
ls_policy_option_value	LONG VARCHAR NOT NULL	逻辑服务器策略选项值。

基础系统表上的约束：

- 主键 (ls\_policy\_id、ls\_policy\_option\_name)
- ls\_policy\_id 外键 (ISYSIQLSPOLICY)

### **SYSIQMPXSERVER 系统视图**

显示表 ISYSIQMPXSERVER 的可读版本。ISYSIQMPXSERVER 系统表存储给定 Multiplex 节点的成员资格属性和版本状态数据。

列名	列类型	描述
server_id	UNSIGNED INT NOT NULL	服务器的 ID 号。
server_name	CHAR(128) NOT NULL	服务器名称。必须唯一且不区分大小写。
role	TINYINT NOT NULL	协调器、读取程序或写入程序。
status	TINYINT NOT NULL	已排除或已包含。

列名	列类型	描述
current_version	UNSIGNED BIGINT NULL	服务器的当前版本 ID。
active_version	LONG BINARY NULL	服务器上活动版本的列表（已编码）。
connection_info	LONG VARCHAR NULL	用于公共域连接的包含主机名和端口对的字符串，以分号分隔。
db_path	LONG VARCHAR NOT NULL	服务器数据库文件的完整路径。
private_connection_info	LONG VARCHAR NULL	用于专用网络连接的包含主机名和端口对的字符串，以分号分隔。

基础系统表上的约束：

- 主键 (server\_id)

### **SYSIQMPXSERVERAGENT 系统视图**

显示表 ISYSIQMPXSERVERAGENT 的可读版本。ISYSIQMPXSERVERAGENT 系统表存储指定 Multiplex 服务器的代理连接定义。

列名	列类型	描述
server_id	UNSIGNED INT NOT NULL	服务器的 ID 号。
agent_connection_info	LONG VARCHAR NOT NULL	每个 Multiplex 节点上用于 SCC IQ 代理连接的包含主机名和端口对的字符串，以分号分隔。
agent_user_name	LONG VARCHAR NOT NULL	包含 SCC IQ 代理用户名的字符串。
agent_pwd	VARBINARY(1024) NOT NULL	包含 SCC IQ 代理加密口令的字符串。

基础系统表上的约束：

- 主键 (server\_id)
- 外键 (server\_id) 引用 SYS.SYSIQMPXSERVER

### **SYSIQOBJECTS ASE 兼容性视图**

sysiqobjects 为每个系统表、用户表、视图、过程、触发器、事件、约束、域 (sysdomain)、域 (sysusertype)、列和索引显示一行。此视图由用户 DBO 所有。

另请参见

- 每个 Adaptive Server 数据库中的表（第 751 页）

- SYSCOLUMNS ASE 兼容性视图 (第 665 页)
- SYSCOMMENTS ASE 兼容性视图 (第 665 页)
- SYSINDEXES ASE 兼容性视图 (第 681 页)
- SYSIQVINDEXT ASE 兼容性视图 (第 695 页)
- SYSOBJECTS ASE 兼容性视图 (第 702 页)
- SYSTYPES ASE 兼容性视图 (第 741 页)
- SYSUSERS ASE 兼容性视图 (第 747 页)

### **SYSIQPARTITIONCOLUMN 系统视图**

以可读格式显示 ISYSIQPARTITIONCOLUMN 中的组信息。

```
ALTER VIEW "SYS"."SYSIQPARTITIONCOLUMN"
as select * from SYS.ISYSIQPARTITIONCOLUMN
```

SYSIQPARTITIONCOLUMN 视图中的每一行均描述了 SYSPARTITIONSCHEME 视图中描述的分区表中的 SYSIQPARTITION 视图中描述的分区中的一列。

SYSIQPARTITIONCOLUMN 仅描述未存储在分区的数据空间上的列的分区。

列名	列类型	描述
partitioned_object_id	unsigned bigint	分配给每个分区对象 (表) 的唯一 ID。
partition_id	unsigned int	标识分区表中的分区。
column_id	unsigned int	列的列 ID。
dbspace_id	smallint	存储分区的此列的 dbspace 的 dbspace ID。

基础系统表的约束:

主键 (partitioned\_object\_id、 partition\_id、 column\_id)

外键 (partitioned\_object\_id、 partition\_id) 引用 SYS.ISYSPARTITION

外键 (dbspace\_id) 引用 SYS.ISYSDBSPACE

### **SYSIQRLVMERGEHISTORY 系统视图**

每次开始合并 RLV 存储库和 IQ 主存储库时, 都会为每个启用行级版本控制 (RLV) 的表添加一个日志条目。日志条目在合并完成时更新。

列名	列类型	描述
merge_id	unsigned bigint	唯一的日志条目标识符
table_id	unsigned int	Sys.systable 系统表的外键
start_time	timestamp	合并开始时间
end_time	timestamp	合并结束时间



列名	列类型	描述
status	char(9)	STARTED   COMPLETED   FAILED
return_code	tinyint	合并完成后的 SQL 代码
merge_type	char(9)	合并触发原因: AUTOMATIC   DML   DDL   SHUTDOWN   USER
merge_mode	char(12)	BLOCKING   NON-BLOCKING
merge_detail	varchar(255)	附加信息 (如果提供), 例如错误信息
rows_inserted	unsigned bigint	由于合并而插入的行数
rows_updated	unsigned bigint	由于合并而更新的行数
rows_deleted	unsigned bigint	由于合并而删除的行数
rows_forwarded	unsigned bigint	合并时未提交的行数

### **SYSIQRVLOG** 系统视图

以可读格式显示 ISYSIQRVLOG 中的组信息。SYSIQRVLOG 视图中的每一行都对应于一个启用 RLV 的表的日志。table\_id 为 0 的行表示服务器范围的提交日志。

列名	列类型	描述
stream_id	unsigned int	日志流标识符。
table_id	unsigned int	指示日志流所属的表。NULL 指示提交日志流。
partition_low	unsigned int	与上次激活该日志时使用的分区映射相对应。
partition_high	int	与上次激活该日志时使用的分区映射相对应。
identity_location	unsigned bigint	日志流标识块的位置。

### **SYSIQTAB** 系统视图

以可读格式显示 ISYSIQTAB 中的组信息。SYSIQTAB 视图中的每一行均描述了一个 IQ 表。

```
ALTER VIEW "SYS"."SYSIQTAB"
as select * from SYS.ISYSIQTAB
```

**注意:** 此视图将取代不建议使用的 SYSIQTABLE 系统视图。

列名	列类型	描述
table_id	unsigned int	每个表都有一个属于主键的唯一编号 (表号)。
block_map	hs_blockmapidentity	供内部使用。

列名	列类型	描述
block_map_size	unsigned int	供内部使用。
vdo	hs_vdoidentity	供内部使用。
vdoid_size	unsigned int	供内部使用。
info_location	hs_vdorecid	尚未使用。始终为零。
info_recid_size	unsigned int	尚未使用。始终为零。
info_location_size	unsigned int	尚未使用。始终为零。
commit_txn_id	unsigned bigint	供内部使用。
txn_id	unsigned bigint	供内部使用。
update_time	timestamp	最后一次修改 IQ 表的日期和时间。
is_rlv	char(1)	'T' - RLV 存储在表上启用; 'F' - RLV 存储不在表上启用。

基础系统表的约束:

主键 (table\_id)

另请参见

- SYSIQTABLE 系统视图 (不建议使用) (第 695 页)

### **SYSIQTABLECOL 系统视图**

以可读格式显示 ISYSIQTABLECOL 中的组信息。SYSIQTABLECOL 视图中的每一行均描述 IQ 表中的一列。

```
ALTER VIEW "SYS"."SYSIQTABLECOL"
as select * from SYS.ISYSIQTABLECOL
```

**注意:** 此视图将取代不建议使用的 SYSIQCOLUMN 系统视图。

列名	列类型	描述
link_table_id	unsigned int	供内部使用。
link_column_id	unsigned int	供内部使用。
max_length	unsigned int	表示该列允许的最大长度。
approx_unique_count	rowid	此列的唯一值 (基数) 的近似数字。
cardinality	rowid	此列的唯一值 (基数) 的实际数字。
has_data	char(1)	表示该列包含数据 (T/F)。

列名	列类型	描述
is_nbit	char(1)	表示列是 NBit (T) 还是 Flat FP (F)。

基础系统表的约束:

主键 (table\_id)

另请参见

- SYSIQCOLUMN 系统视图 (不建议使用) (第 684 页)

### **SYSIQTAB 系统视图 (不建议使用)**

SYSIQTAB 已由 SYSIQTAB 系统视图取代。

另请参见

- SYSIQTAB 系统视图 (第 693 页)

### **SYSIQVINDEX ASE 兼容性视图**

sysiqvindex 提供与每个非 FP IQ 索引相对应的一行。

此视图由用户 DBO 所有。

另请参见

- 每个 Adaptive Server 数据库中的表 (第 751 页)
- SYSCOLUMNS ASE 兼容性视图 (第 665 页)
- SYS COMMENTS ASE 兼容性视图 (第 665 页)
- SYSINDEXES ASE 兼容性视图 (第 681 页)
- SYSIQOBJECTS ASE 兼容性视图 (第 691 页)
- SYSOBJECTS ASE 兼容性视图 (第 702 页)
- SYSTYPES ASE 兼容性视图 (第 741 页)
- SYSUSERS ASE 兼容性视图 (第 747 页)

### **SYSIXCOL 兼容性视图 (不建议使用)**

提供 SYSIXCOL 视图的目的是为了与提供 SYSIXCOL 系统表的旧版本软件兼容。但 SYSIXCOL 系统表已被 ISYSIDXCOL 系统表所取代, 与后者对应的是 SYSIDXCOL 系统视图。您应改用 SYSIDXCOL 系统视图。

SYSIXCOL 的每一行都描述索引中的一列。在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息, 请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSIXCOL"
  as select a.table_id,
         a.index_id,
         a.sequence,
         a.column_id,
         a."order"
```

```

from SYS.ISYSIDXCOL as a
     ,SYS.ISYSIDX as b
where a.table_id = b.table_id
     and a.index_id = b.index_id
     and b.index_category = 3

```

### **SYSJAR 系统视图**

SYSJAR 系统视图的每行定义在数据库中存储的一个 JAR 文件。该视图的基础系统表为 ISYSJAR。

列名	数据类型	说明
jar_id	INTEGER	标识 JAR 文件的唯一编号。
object_id	UNSIGNED BIGINT	JAR 文件的内部 ID，在数据库中唯一标识该文件。
creator	UNSIGNED INT	JAR 文件创建者的用户号。可通过 INSTALL JAVA 语句的 AS USER 子句进行设置。
jar_name	LONG VARCHAR	JAR 文件的名称。
jar_file	LONG VARCHAR	此列现已不再使用，并包含 NULL。
update_time	TIMESTAMP	上次更新 JAR 文件的本地时间。
update_time_utc	TIMESTAMP WITH TIME ZONE	上次更新 JAR 文件的 UTC 时间。

#### *基础系统表上的约束*

```
PRIMARY KEY (jar_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH
UNIQUE FULL
```

```
UNIQUE INDEX (jar_name)
```

### **SYSJARCOMPONENT 系统视图**

SYSJAR 系统视图中的每一行都定义一个 JAR 文件组件。该视图的基础系统表为 ISYSJARCOMPONENT。

列名	数据类型	说明
component_id	INTEGER	包含组件 ID 的主键。
jar_id	INTEGER	包含 JAR 的 ID 号的字段。
component_name	LONG VARCHAR	组件的名称。
component_type	CHAR(1)	此列现已不再使用，并包含 NULL。
contents	LONG BINARY	JAR 文件的字节代码。

*基础系统表上的约束*

```
PRIMARY KEY (component_id)
```

```
FOREIGN KEY (jar_id) REFERENCES SYS.ISYSJAR (jar_id)
```

**SYSJAVACLASS 系统视图**

**SYSJAVACLASS** 系统视图的每行描述在数据库中存储的一个 Java 类。该视图的基础系统表为 **ISYSJAVACLASS**。

列名	数据类型	说明
class_id	INTEGER	Java 类的唯一编号。也是表的主键。
object_id	UNSIGNED BIGINT	JAR 类的内部 ID，在数据库中唯一标识该类。
creator	UNSIGNED INT	类的创建者的用户号。可通过 <b>INSTALL JAVA</b> 语句的 <b>AS USER</b> 子句进行设置。
jar_id	INTEGER	产生类的 JAR 文件的 ID。
class_name	LONG VARCHAR	Java 类的名称。
public	CHAR(1)	指出该类是公共类 (Y) 还是专用类 (N)。
component_id	INTEGER	<b>SYSJARCOMPONENT</b> 系统视图中组件的 ID。
update_time	TIMESTAMP	类的上次更新本地时间。
update_time_utc	TIMESTAMP WITH TIME ZONE	类的上次更新 UTC 时间。

*基础系统表上的约束*

```
PRIMARY KEY (class_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (component_id) REFERENCES SYS.ISYSJARCOMPONENT  
(component_id)
```

**SYSLDAPSERVER 系统视图**

以可读格式显示 **ISYSLDAPSERVER** 系统表中的信息。

**ISYSLDAPSERVER** 系统表为 LDAP 服务器定义了一系列属性。

列名	列类型	描述
ldsrv_id	UNSIGNED BIGINT NOT NULL	LDAP 服务器的唯一标识符，同时也是主键，由登录策略用来引用 LDAP 服务器。
ldsrv_name	CHAR(128) NOT NULL	指派给 LDAP 服务器的名称。
ldsrv_state	CHAR(9) NOT NULL	LDAP 服务器的只读状态： 1 - RESET 2 - READY 3 - ACTIVE 4 - FAILED 5 - SUSPENDED  <b>注意：</b> 数字值存储在系统表中；相应的文本值在系统视图中显示。
ldsrv_start_tls	TINYINT NOT NULL	控制是否使用传送层安全 (TLS) 连接到 LDAP 服务器。将此协议与 TRUSTED_CERTIFICATE_FILE 结合使用，为 LDAP 服务器上的连接和搜索提供经过加密的通信。 有效范围：1 (ON) 或 0 (OFF)。 缺省值为 0 (OFF)。
ldsrv_num_retries	TINYINT NOT NULL	控制 LDAP 服务器允许在执行多少次验证尝试后返回故障或启动故障转移（如指定故障转移）。 有效范围：1-60 缺省值为 3。
ldsrv_timeout	UNSIGNED INT NOT NULL	控制连接或搜索的超时值，以毫秒为单位。 有效范围：1 - 3600000（1 小时）。 缺省值为 10000。
ldsrv_last_state_change	TIMESTAMP NOT NULL	表示上一次更改状态的时间。无论 LDAP 服务器的当地时区是多少，该值都以协调通用时间 (UTC) 形式存储。
ldsrv_search_url	CHAR(1024) NULL	LDAP URL，用于根据用户 ID 查找用户的可分辨名称 (DN)。
ldsrv_auth_url	CHAR(1024) NULL	LDAP 搜索字符串，用于为给定用户 ID 的用户查找 DN。
ldsrv_access_dn	CHAR(1024) NULL	DN，用于访问 LDAP 服务器进行搜索以获取用户 ID 的 DN。

列名	列类型	描述
ldsrv_access_dn_pwd	VARBINARY(1024) NULL	访问帐户的口令。在磁盘上存储此口令时采用对称加密形式。

### **SYSLOGINMAP** 系统视图

**SYSLOGINMAP** 系统视图为可使用集成登录或 Kerberos 登录连接到数据库的每个用户都包含一行。为此，访问此视图将受到限制。该视图的基础系统表为 **ISYSLOGINMAP**。

列名	数据类型	说明
login_mode	TINYINT	登录的类型：1 表示集成登录，2 表示 Kerberos 登录。
login_id	VARCHAR(1024)	集成登录用户配置文件名或映射到 database_uid 的 Kerberos 主体。
object_id	UNSIGNED BIGINT	唯一标识符，用户 ID 与数据库用户 ID 间的每个映射各一个。
database_uid	UNSIGNED INT	登录 ID 所映射到的数据库用户 ID。

#### *基础系统表上的约束*

```
PRIMARY KEY (login_mode, login_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (database_uid) REFERENCES SYS.ISYSUSER (user_id)
```

### **SYSLOGINPOLICY** 系统视图

该视图的基础系统表为 **ISYSLOGINPOLICY**。

列名	数据类型	说明
login_policy_id	UNSIGNED BIGINT	登录策略的唯一标识符。
login_policy_name	CHAR(128)	登录策略的名称。

#### *基础系统表上的约束*

```
PRIMARY KEY (login_policy_id)
```

```
FOREIGN KEY (login_policy_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
UNIQUE INDEX (login_policy_name)
```

**SYSLOGINPOLICYOPTION 系统视图**

该视图的基础系统表为 ISYSLOGINPOLICYOPTION。

列名	数据类型	说明
login_policy_id	UNSIGNED BIGINT	登录策略的唯一标识符。
login_option_name	CHAR(128)	登录策略的名称。
login_option_value	LONG VARCHAR	登录策略被创建时的值。

*基础系统表上的约束*

```
PRIMARY KEY (login_policy_id, login_option_name)
```

```
FOREIGN KEY (login_policy_id) REFERENCES SYS.ISYSLOGINPOLICY  
(login_policy_id)
```

**SYSLOGINS ASE 兼容性视图**

此视图由用户 DBO 所有。SYSLOGINS 为每个 Adaptive Server 用户帐户包含一行。

**SYSMVOPTION 系统视图**

SYSMVOPTION 系统视图的每行描述了实例化视图或文本索引在其创建时一个选项值的设置。在 SYSMVOPTIONNAME 系统视图中可以找到选项的名称。该视图的基础系统表为 ISYSMVOPTION。

列名	数据类型	说明
view_object_id	UNSIGNED BIGINT	实例化视图的对象 ID。
option_id	UNSIGNED INT	标识数据库中的选项的唯一编号。要查看选项名称，请参见 SYSMVOPTIONNAME 系统视图。
option_value	LONG VARCHAR	创建实例化视图时的选项值。

*基础系统表上的约束*

```
PRIMARY KEY (view_object_id, option_id)
```

```
FOREIGN KEY (view_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
FOREIGN KEY (option_id) REFERENCES SYS.ISYSMVOPTIONNAME (option_id)
```



**SYSMVOPTIONNAME 系统视图**

SYSMVOPTION 系统视图的每行给出了实例化视图或文本索引在其创建时的名称选项值。在 SYSMVOPTION 系统视图中可以找到选项的值。该视图的基础系统表为 ISYSMVOPTIONNAME。

列名	数据类型	说明
option_id	UNSIGNED INT	在数据库中唯一标识选项的编号。
option_name	CHAR(128)	选项的名称。

*基础系统表上的约束*

PRIMARY KEY (option\_id)

UNIQUE INDEX (option\_name)

**SYSOBJECT 系统视图**

SYSOBJECT 系统视图中的每行都描述一个数据库对象。该视图的基础系统表为 ISYSOBJECT。

列名	数据类型	说明
object_id	UNSIGNED BIGINT	对象的内部 ID，在数据库中唯一标识该对象。
status	TINYINT	对象的状态。值包括： <ul style="list-style-type: none"> <li>• <b>1 (有效)</b> - 该对象可供数据库服务器使用。此状态与 [已启用] 同义。即，如果启用一个对象，则其状态变为 [有效]。</li> <li>• <b>2 (无效)</b> - 在一项内部操作之后尝试重新编译对象失败，例如，在对其依赖的某个对象进行模式变更修改之后。只要语句中一引用对象，数据库服务器就继续尝试重新编译该对象。</li> <li>• <b>4 (禁用)</b> - 该对象已被用户显式禁用，例如使用 ALTER TABLE...DISABLE VIEW DEPENDENCIES 语句。</li> </ul>
object_type	TINYINT	对象的类型。
creation_time	TIMESTAMP	对象创建的本地日期和时间。
object_type_str	CHAR(128)	对象的类型。
creation_time_utc	TIMESTAMP WITH TIME ZONE	对象创建的 UTC 日期和时间。

*基础系统表上的约束*

```
PRIMARY KEY (object_id)
```

**SYSOBJECTS ASE 兼容性视图**

每个表、视图、存储过程、扩展存储过程、日志、规则、缺省值、触发器、检查约束、参照约束、计算列、基于函数的索引键、临时对象以及其它形式的编译对象在 sysobjects 中都有对应的一行。

此视图由用户 DBO 所有。

对象类型为 N 时的每个分区条件 ID 在该视图中也有对应的一行。

**另请参见**

- 每个 Adaptive Server 数据库中的表 (第 751 页)
- SYSCOLUMNS ASE 兼容性视图 (第 665 页)
- SYS COMMENTS ASE 兼容性视图 (第 665 页)
- SYSINDEXES ASE 兼容性视图 (第 681 页)
- SYSIQOBJECTS ASE 兼容性视图 (第 691 页)
- SYSIQVINDEXT ASE 兼容性视图 (第 695 页)
- SYSTYPES ASE 兼容性视图 (第 741 页)
- SYSUSERS ASE 兼容性视图 (第 747 页)

**SYSOPTION 系统视图**

SYSOPTION 系统视图包含一些选项，每个存储在数据库中的选项设置对应一行。对于给定的选项，每个用户都可以有自己的设置。此外，PUBLIC 角色的设置定义了缺省设置，以用于没有个人设置的用户。该视图的基础系统表为 ISYSOPTION。

列名	数据类型	说明
user_id	UNSIGNED INT	对其应用该选项设置的用户号。
"option"	CHAR(128)	选项的名称。
"setting"	LONG VARCHAR	选项的当前设置。

*基础系统表上的约束*

```
PRIMARY KEY (user_id, "option")
```

```
FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)
```

### **SYSOPTIONS 统一视图**

**SYSOPTIONS** 视图的每一行都描述了用 **SET** 命令创建的一个选项。对于每个选项，每个用户都可以有自己的设置。此外，**PUBLIC** 用户的设置定义了缺省设置，以用于没有自己的设置的用户。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSOPTIONS"( user_name,"option",setting )
  as select u.user_name,opt."option",opt.setting
  from SYS.ISYSOPTION as opt
  join SYS.ISYSUSER as u on opt.user_id = u.user_id
```

### **SYSOPTSTAT 系统视图**

**SYSOPTSTAT** 系统视图存储开销模型校准信息（由 **ALTER DATABASE CALIBRATE** 语句计算）。该视图的内容仅供内部使用并可通过 **sa\_get\_dtt** 系统过程最有效地访问。该视图的基础系统表为 **ISYSOPTSTAT**。

列名	数据类型	说明
stat_id	UNSIGNED INT	仅供系统使用。
group_id	UNSIGNED INT	仅供系统使用。
format_id	SMALLINT	仅供系统使用。
data	LONG BINARY	仅供系统使用。

#### *基础系统表上的约束*

```
PRIMARY KEY (stat_id, group_id, format_id)
```

### **SYSPARTITION 系统视图**

以可读格式显示 **ISYSPARTITION** 中的组信息。

**SYSPARTITION** 视图中的每行都描述数据库中的一个分区对象（表或索引）。该视图的基础系统表为 **ISYSPARTITION**。

列名	列类型	描述
partitioned_object_id	unsigned bigint	分配给每个分区对象（表）的唯一编号。
partition_id	unsigned int	标识分区表中的分区。
partition_object_id	unsigned bigint	每个表分区本身都是一个对象，并从表对象或索引对象为其分配了唯一编号。

列名	列类型	描述
partition_values	long varchar	包含范围或列表分区的分区条件。 对于范围分区而言，值包含此分区的上限。 对于列表分区而言，值包含以 ‘,’ 分隔的值列表。位置是分区的顺序号。
位置	unsigned int	分区的序号。 就范围分区而言，对于位置 2 及以上，(position-1) 的分区包含独占下限。
partition_name	char(128)	分区的名称。

基础系统表的约束:

主键 (partitioned\_object\_id、 partition\_id)

唯一 (partition\_object\_id、 position)

外键 (partition\_object\_id) 引用 SYS.ISYSOBJECT

外键 (partitioned\_object\_id) 引用 SYS.ISYSOBJECT

### **SYSPARTITIONKEY 系统视图**

以可读格式显示 ISYSPARTITIONKEY 中的组信息。

SYSPARTITIONKEY 视图中的每一行均描述了数据库中的一个分区对象（表或索引）。

```
ALTER VIEW "SYS"."SYSPARTITIONKEY"
as select * from SYS.ISYSPARTITIONKEY
```

列名	列类型	描述
partitioned_object_id	unsigned bigint	每个分区对象（表）都被指派了一个唯一的对象号。
column_id	unsigned int	列 ID 将表列标识为分区键的一部分。
位置	smallint	此列在分区键中的位置。位置从 0 开始。位置 0 表示分区键中的第 1 列。

基础系统表的约束:

主键 (partitioned\_object\_id、 column\_id)

外键 (partitioned\_object\_id) 引用 SYS.ISYSOBJECT

**SYSPARTITIONS 系统视图**

以可读格式显示 ISYSPARTITIONS 系统表中的组信息。

SYSPARTITIONS 视图中的每行都描述数据库中的一个分区对象（表或索引）。该视图的基础系统表为 ISYSPARTITIONS。

```
ALTER VIEW "SYS"."SYSPARTITIONS"
as select * from SYS.ISYSPARTITIONS
```

列名	数据类型	描述
table_id	unsigned int	索引对应的表的对象 ID。
partition_id	unsigned int	标识分区表中的分区。
partition_object_id	unsigned bigint	每个表分区本身都是一个对象，并从表对象或索引对象中分配了唯一编号。
partition_dbpace_id	smallint	分区所在的 dbspace 的对象 ID。
partition_values	long varchar	包含此范围分区的上限。
position	unsigned int	分区的序号。
partition_name	char(128)	分区的名称

基础系统表上的约束：

主键 (partitioned\_object\_id、 partition\_id)

外键 (partitioned\_object\_id) 引用 SYS.ISYSOBJECT

外键 (partitioned\_object\_id) 引用 SYS.ISYSOBJECT

**SYSPARTITIONSCHEME 系统视图**

以可读格式显示 ISYSPARTITIONSCHEME 中的组信息。

SYSPARTITIONSCHEME 视图中的每行都描述数据库中的一个分区对象（表或索引）。

```
ALTER VIEW "SYS"."SYSPARTITIONSCHEME"
as select * from SYS.ISYSPARTITIONSCHEME
```

列名	列类型	描述
partitioned_object_id	unsigned bigint	每个分区对象（表）都分配有一个唯一编号。
partition_method	tinyint	此表的分区方法。有效值：1 - 范围，3 - 散列（不使用 2）。

列名	列类型	描述
subpartition_method	tinyint	此表的子分区方法。有效值: NULL - 没有子分区, 1 - 范围分区, 3 - 散列分区 (不使用 2)。

基础系统表的约束:

主键 (partitioned\_object\_id)

外键 (partitioned\_object\_id) 引用 SYS.ISYSOBJECT

### **SYSPHYSIDX 系统视图**

SYSPHYSIDX 系统视图的每行定义数据库中的一个物理索引。该视图的基础系统表为 SYSPHYSIDX。

列名	数据类型	说明
table_id	UNSIGNED INT	索引对应的表的对象 ID。
phys_index_id	UNSIGNED INT	表中物理索引的唯一编号。
root	INTEGER	标识数据库文件中物理索引的根页的位置。
key_value_count	UNSIGNED INT	索引中不同键值的数量。
leaf_page_count	UNSIGNED INT	叶索引页的数量。
depth	UNSIGNED SMALLINT	物理索引的深度 (层数)。
max_key_distance	UNSIGNED INT	仅供系统使用。
seq_transitions	UNSIGNED INT	仅供系统使用。
rand_transitions	UNSIGNED INT	仅供系统使用。
rand_distance	UNSIGNED INT	仅供系统使用。
allocation_bitmap	LONG VARBIT	仅供系统使用。
long_value_bitmap	LONG VARBIT	仅供系统使用。

基础系统表上的约束

```
PRIMARY KEY (table_id, phys_index_id)
```

### **SYSPROCAUTH 统一视图**

SYSPROCAUTH 视图的每一行都描述了针对过程授予的一组权限。您可以使用 SYSROCPERM 系统视图替代该视图。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息, 请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSPROCAUTH" ( grantee,
    creator, procname )
as select u1.user_name, u2.user_name, p.proc_name
```

```

from SYS.ISYSPROCEDURE as p
  join SYS.ISYSPROCPERM as pp on(p.proc_id = pp.proc_id)
  join SYS.ISYSUSER as u1 on u1.user_id = pp.grantee
  join SYS.ISYSUSER as u2 on u2.user_id = p.creator

```

### **SYSPROCEDURE 系统视图**

SYSPROCEDURE 系统视图的每行描述数据库中的一个过程。该视图的基础系统表为 ISYSPROCEDURE。

列名	数据类型	说明
proc_id	UNSIGNED INT	每个过程都有一个指定的唯一编号（过程号）。
creator	UNSIGNED INT	过程的所有者。
object_id	UNSIGNED BIGINT	过程的内部 ID，在数据库中唯一标识该过程。
proc_name	CHAR(128)	过程的名称。一个创建者不能有两个同名过程。
proc_defn	LONG VARCHAR	过程的定义。
remarks	LONG VARCHAR	有关过程的注释。该值存储在 ISYSREMARK 系统表中。
replicate	CHAR(1)	此属性仅供内部使用。
srvid	UNSIGNED INT	如果该过程是远程数据库服务器上过程的代理，则表示该远程服务器。
source	LONG VARCHAR	过程的保留源。该值存储在 ISYSSOURCE 系统表中。
avg_num_rows	FLOAT	当该过程出现在 FROM 子句中时，收集到的用于查询优化的信息。
avg_cost	FLOAT	当该过程出现在 FROM 子句中时，收集到的用于查询优化的信息。
stats	LONG BINARY	当该过程出现在 FROM 子句中时，收集到的用于查询优化的信息。

#### 基础系统表上的约束

```
PRIMARY KEY (proc_id)
```

```
FOREIGN KEY (srvid) REFERENCES SYS.ISYSSEVER (srvid)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH
UNIQUE FULL
```

```
FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)
```

```
UNIQUE INDEX (proc_name, creator)
```

**SYSPROCPARM 系统视图**

SYSPROCPARM 系统视图的每行描述数据库中过程的一个参数。该视图的基础系统表为 ISYSPROCPARM。

列名	数据类型	说明
proc_id	UNSIGNED INT	唯一标识该参数所属的过程。
parm_id	SMALLINT	每个过程都从 1 开始对参数进行编号。参数编号的顺序对应于定义参数时采取的顺序。对于函数，第一个参数具有该函数的名称并代表该函数的返回值。
parm_type	SMALLINT	参数类型将为下列类型之一： <ul style="list-style-type: none"> <li>• 0 - 普通参数 (变量)</li> <li>• 1 - 结果变量 - 与返回结果集的过程一起使用</li> <li>• 2 - SQLSTATE 错误值</li> <li>• 3 - SQLCODE 错误值</li> <li>• 4 - 函数的返回值</li> </ul>
parm_mode_in	CHAR(1)	指出该参数是否向过程 (IN 或 INOUT 参数) 提供值。
parm_mode_out	CHAR(1)	指出该参数从过程 (OUT 或 INOUT 参数) 返回值还是在 RESULT 子句中返回列。
domain_id	SMALLINT	用 SYSDOMAIN 系统视图中列出的数据类型号标识参数的数据类型。
width	BIGINT	包含字符串参数的长度、数字参数的精度或任何其它数据类型的存储字节数。
scale	SMALLINT	对于数字数据类型是小数点后的位数。对于所有其它数据类型，该列的值为 1。
user_type	SMALLINT	参数的用户类型 (如果适用)。
parm_name	CHAR(128)	过程参数的名称。
"default"	LONG VARCHAR	参数的缺省值。仅供参考。
remarks	LONG VARCHAR	始终返回 NULL。前提是允许将上一版本的 ODBC 驱动程序与较新的个人数据库服务器配合使用。
base_type_str	VARCHAR(32767)	表示参数物理类型的加批注类型字符串。

*基础系统表上的约束*

PRIMARY KEY (proc\_id, parm\_id)

FOREIGN KEY (proc\_id) REFERENCES SYS.ISYSPROCEDURE (proc\_id)

FOREIGN KEY (domain\_id) REFERENCES SYS.ISYSDOMAIN (domain\_id)



```
FOREIGN KEY (user_type) REFERENCES SYS.ISYSUSERTYPE (type_id)
```

### **SYSROCPARMS 统一视图**

**SYSROCPARMS** 视图中的每一行都描述数据库中某个过程的一个参数。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSROCPARMS"( creator,
    procname,paramname,param_id,paramtype,parammode,paramdomain,
    length,scale,"default",user_type )
as select
up.user_name,p.proc_name,pp.param_name,pp.param_id,pp.param_type,
    if pp.param_mode_in = 'Y' and pp.param_mode_out = 'N' then 'IN'
    else if pp.param_mode_in = 'N' and pp.param_mode_out = 'Y' then
'OUT'
        else 'INOUT'
    endif
endif,dom.domain_name,pp.width,pp.scale,pp."default",ut.type_name
from SYS.SYSROCPARM as pp
join SYS.ISYSPROCEDURE as p on p.proc_id = pp.proc_id
join SYS.ISYSUSER as up on up.user_id = p.creator
join SYS.ISYSDOMAIN as dom on dom.domain_id = pp.domain_id
left outer join SYS.ISYSUSERTYPE as ut on ut.type_id =
pp.user_type
```

### **SYSROCPERM 系统视图**

**SYSROCPERM** 系统视图的每行都描述一个被授予过程 EXECUTE 特权的用户。该视图的基础系统表为 **ISYSROCPERM**。

列名	数据类型	说明
proc_id	UNSIGNED INT	过程号唯一标识被授予 EXECUTE 特权的进程。
grantee	UNSIGNED INT	被授予特权者的用户号。

#### *基础系统表上的约束*

```
PRIMARY KEY (proc_id, grantee)
```

```
FOREIGN KEY (grantee) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (proc_id) REFERENCES SYS.ISYSPROCEDURE (proc_id)
```

### **SYSROCS 统一视图**

**SYSROCS** 视图显示过程或函数的名称、其创建者的名称以及为过程或函数记录的任何注释。

在以下的 **ALTER VIEW** 语句中提供了组成该视图的表和列。

```
ALTER VIEW "SYS"."SYSROCS"( creator,
    procname,remarks )
as select u.user_name,p.proc_name,r.remarks
```

```

from SYS.ISYSPROCEDURE as p
  join SYS.ISYSUSER as u on u.user_id = p.creator
  left outer join SYS.ISYSREMARK as r on (p.object_id =
r.object_id)
    
```

**SYSPROXYTAB 系统视图**

SYSPROXYTAB 系统视图的每行描述一个代理表的远程参数。该视图的基础系统表为 ISYSPROXYTAB。

列名	数据类型	说明
table_object_id	UNSIGNED BIGINT	代理表的对象 ID。
existing_obj	CHAR(1)	指出远程服务器上先前是否存在代理表。
srvid	UNSIGNED INT	与代理表相关的远程服务器的唯一 ID。
remote_location	LONG VARCHAR	远程服务器上代理表的位置。

*基础系统表上的约束*

```
PRIMARY KEY (table_object_id)
```

```
FOREIGN KEY (table_object_id) REFERENCES ISYSOBJECT (object_id)
MATCH UNIQUE FULL
```

```
FOREIGN KEY (srvid) REFERENCES SYS.ISYSSEVER (srvid)
```

**SYSPUBLICATION 系统视图**

SYSPUBLICATION 系统视图中的每一行都描述一个发布。该视图的基础系统表为 ISYSPUBLICATION。

列名	数据类型	说明
publication_id	UNSIGNED INT	唯一标识发布的编号。
object_id	UNSIGNED BIGINT	发布的内部 ID，在数据库中唯一标识该发布。
creator	UNSIGNED INT	发布的所有者。
publication_name	CHAR(128)	发布的名称。
remarks	LONG VARCHAR	有关发布的注释。该值存储在 ISYSREMARK 系统表中。
type	CHAR(1)	不建议使用此列。

列名	数据类型	说明
sync_type	UNSIGNED INT	发布的同步类型。值包括： <ul style="list-style-type: none"> <li><b>0 (日志扫描)</b> - 这是一个常规发布，它使用事务日志上载自上次上载以来发生更改的全部相关数据。</li> <li><b>1 (脚本式上载)</b> - 对于该发布，将忽略事务日志并由用户使用存储过程定义上载。有关存储过程的信息存储在 <b>ISYSSYNCSCRIPT</b> 系统表中。</li> <li><b>2 (仅下载)</b> - 这是一个仅下载发布；不上载任何数据。</li> </ul>

### 基础系统表上的约束

PRIMARY KEY (publication\_id)

FOREIGN KEY (object\_id) REFERENCES SYS.ISYSOBJECT (object\_id) MATCH UNIQUE FULL

FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user\_id)

UNIQUE INDEX (publication\_name, creator)

### **SYSPUBLICATIONS** 统一视图

SYSPUBLICATIONS 视图中的每一行都描述一个发布。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSPUBLICATIONS"
as select u.user_name as creator,
p.publication_name,
r.remarks,
p.type,
case p.sync_type
when 0 then 'logscan'
when 1 then 'scripted upload'
when 2 then 'download only'
else 'invalid'
end as sync_type
from SYS.ISYSPUBLICATION as p
join SYS.ISYSUSER as u on u.user_id = p.creator
left outer join SYS.ISYSREMARK as r on (p.object_id =
r.object_id)
```

### **SYREMARK** 系统视图

SYREMARK 系统视图的每行描述一个对象的一个注释。该视图的基础系统表为 ISYSREMARK。

列	数据类型	说明
object_id	UNSIGNED BIGINT	具有关联注释的对象的内部 ID。

列	数据类型	说明
remarks	LONG VARCHAR	与对象关联的注释。

*基础系统表上的约束*

```
PRIMARY KEY (object_id)
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH
UNIQUE FULL
```

**SYSREMOTEPTION 系统视图**

SYSREMOTEPTION 系统视图的每一行都描述一个消息链接参数的值。该视图的基础系统表为 ISYSREMOTEPTION。

该视图中的一些列包含可能比较敏感的数据。SYSREMOTEPTION2 视图提供了访问此视图中除可能的敏感列以外的数据的公共方法。

列	数据类型	说明
option_id	UNSIGNED INT	消息链接参数的标识号。
user_id	UNSIGNED INT	为其设置参数的用户 ID。
"setting"	VARCHAR(255)	消息链接参数的值。

*基础系统表上的约束*

```
PRIMARY KEY (option_id, user_id)
FOREIGN KEY (option_id) REFERENCES SYS.ISYSREMOTEPTIONTYPE
(option_id)
FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)
```

**SYSREMOTEPTION2 统一视图**

以一种可读性更强的形式联合显示 SYSREMOTEPTION 和 SYSREMOTEPTIONTYPE 中的列。

对于没有 SELECT ANY TABLE 系统特权的用户，设置列中的值将隐藏。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSREMOTEPTION2"
as select ISYSREMOTEPTION.option_id,
ISYSREMOTEPTION.user_id,
SYS.HIDE_FROM_NON_DBA(ISYSREMOTEPTION.setting) as setting
from SYS.ISYSREMOTEPTION
```

**SYSREMOPTIONS 统一视图**

SYSREMOPTIONS 视图的每一行都描述 消息链接参数的值。

对于没有 SELECT ANY TABLE 系统特权的用户，设置列中的值将隐藏。  
SYSREMOPTION2 视图提供了访问非敏感数据的公共方法。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSREMOPTIONS"
  as select srt.type_name,
         sup.user_name,
         srot."option",
         SYS.HIDE FROM NON_DBA(sro.setting) as setting
  from SYS.ISYSREMOPTIONTYPE as srt
       ,SYS.ISYSREMOPTIONTYPE as srot
       ,SYS.ISYSREMOPTION as sro
       ,SYS.ISYSUSER as sup
  where srt.type_id = srot.type_id
        and srot.option_id = sro.option_id
        and sro.user_id = sup.user_id
```

**SYSREMOPTIONTYPE 系统视图**

SYSREMOPTIONTYPE 系统视图的每一行都描述一个 消息链接参数。该视图的基础系统表为 ISYSREMOPTIONTYPE。

列	数据类型	说明
option_id	UNSIGNED INT	消息链接参数的标识号。
type_id	SMALLINT	使用该参数的消息类型的标识号。
option	VARCHAR(128)	消息链接参数的名称。

*基础系统表上的约束*

```
PRIMARY KEY (option_id)
```

```
FOREIGN KEY (type_id) REFERENCES SYS.ISYSREMOPTIONTYPE (type_id)
```

**SYSREMOPTIONTYPE 系统视图**

该视图的基础系统表为 ISYSREMOPTIONTYPE。

列名	数据类型	说明
type_id	SMALLINT	标识使用 所支持的哪种消息系统向该用户发送消息。
object_id	UNSIGNED BIGINT	远程类型的内部 ID，在数据库中唯一标识该远程类型。
type_name	CHAR(128)	消息系统的名称。

列名	数据类型	说明
publisher_address	LONG VARCHAR	远程数据库发布者的地址。
remarks	LONG VARCHAR	有关远程类型的注释。该值存储在 ISYSREMARK 系统表中。

### 基础系统表上的约束

PRIMARY KEY (type\_id)

FOREIGN KEY (object\_id) REFERENCES SYS.ISYSOBJECT (object\_id) MATCH UNIQUE FULL

UNIQUE INDEX (type\_name)

### **SYSREMOTETYPES** 统一视图

SYSREMOTETYPES 视图的每一行都描述一种 消息类型，包括发布者地址。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSREMOTETYPES"
  as select rt.type_id,rt.type_name,rt.publisher_address,rm.remarks
  from SYS.ISYSREMOTETYPE as rt
  left outer join SYS.ISYSREMARK as rm on(rt.object_id =
  rm.object_id)
```

### **SYSREMOTEUSER** 系统视图

SYSREMOTEUSER 系统视图的每一行都描述一个具有 REMOTE 系统特权的用户 ID (预订者)，以及发送自/至该用户的消息的状态。该视图的基础系统表为 ISYSREMOTEUSER。

列名	数据类型	说明
user_id	UNSIGNED INT	具有 REMOTE 特权的用户的用户号。
consolidate	CHAR(1)	指出是授予了用户 CONSOLIDATE 特权 (Y) 还是 REMOTE 特权 (N)。
type_id	SMALLINT	标识使用 哪种消息系统向此用户发送消息。
address	LONG VARCHAR	消息要发送到的地址。该地址必须符合 address_type。
frequency	CHAR(1)	发送 消息的频率。
send_time	TIME	下一次将消息发送给该用户的时间。
log_send	UNSIGNED BIGINT	仅将消息发送给 log_send 大于 log_sent 的预订者。
time_sent	TIMESTAMP	将最新的消息发送到该预订者的本地时间。

列名	数据类型	说明
log_sent	UNSIGNED BIGINT	最近的发送操作的日志偏移。
confirm_sent	UNSIGNED BIGINT	此预订者最近一次确认的操作的日志偏移。
send_count	INTEGER	已发送的消息数。
resend_count	INTEGER	用于确保在预订者数据库上只应用一次消息的计数器。
time_received	TIMESTAMP	收到来自该预订者的最新消息的本地时间。
log_received	UNSIGNED BIGINT	预订者数据库中当前数据库上最近收到的操作的日志偏移。
confirm_received	UNSIGNED BIGINT	预订者数据库中已发送确认消息的最近操作的日志偏移。
receive_count	INTEGER	已接收到的消息数量。
rereceive_count	INTEGER	用于确保在当前数据库上只应用一次消息的计数器。
time_sent_utc	TIMESTAMP WITH TIME ZONE	将最新的消息发送到该预订者的 UTC 时间。
time_received_utc	TIMESTAMP WITH TIME ZONE	收到来自该预订者的最新消息的 UTC 时间。

### 基础系统表上的约束

```
PRIMARY KEY (user_id)
```

```
FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (type_id) REFERENCES SYS.ISYSREMOTETYPE (type_id)
```

```
UNIQUE INDEX (type_id, address)
```

### **SYSREMOTEUSERS** 统一视图

**SYSREMOTEUSERS** 视图的每一行都描述一个具有 **REMOTE** 系统特权的用户 ID (预订者)，以及发送自/至该用户的消息的状态。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSREMOTEUSERS" AS SELECT u.user_name,
r.consolidate, t.type_name, r.address, r.frequency, r.send_time,
(if r.frequency = 'A' then NULL
else if r.frequency = 'P' then
if r.time_sent IS NULL then CURRENT_TIMESTAMP
```

```

else (select min( minutes( dateadd(mi,
PROPERTY('TimeZoneAdjustment'), a.time_sent),
60*hour(a.send_time) + minute( seconds( a.send_time, 59 ) ) ) )
FROM SYS.ISYSREMOTEUSER a WHERE a.frequency = 'P' AND a.send_time =
r.send_time ) endif
else if CURRENT DATE + r.send_time > coalesce( dateadd(mi,
PROPERTY('TimeZoneAdjustment'), r.time_sent), CURRENT TIMESTAMP)
then CURRENT DATE + r.send_time
else CURRENT DATE + r.send_time + 1
endif endif endif) as next_send, r.log_send ,
dateadd(mi, PROPERTY('TimeZoneAdjustment'), r.time_sent)
as time_sent , r.log_send, r.confirm_send, r.send_count,
r.resend_count,
dateadd(mi, PROPERTY('TimeZoneAdjustment'), r.time_received) as
time_received ,
r.log_received, r.confirm_received, r.receive_count,
r.rereceive count ,
TODATETIMEOFFSET( r.time_sent, 0 ) as time_sent_utc ,
TODATETIMEOFFSET( r.time_received, 0 ) as time_received_utc
FROM SYS.ISYSREMOTEUSER r JOIN SYS.ISYSUSER u ON ( u.user_id =
r.user_id ) JOIN SYS.ISYSREMOType t ON ( t.type_id = r.type_id )

```

### **SYSROLEGRANTTEXT 系统视图**

SYSROLEGRANTTEXT 系统视图包含与 SET USER 和 CHANGE PASSWORD 系统特权相关的语法扩展，并且与 SYSROLEGRANT 系统视图相关。

列名	数据类型	描述
grant_id	UNSIGNED INT	用于标识每个 GRANT 语句的 ID。
user_id	UNSIGNED INT	在特定扩展的授权中，在 user-list 或 role-list 中指定的 user_id。

当您使用 user-list 选项或 ANY WITH ROLES role-list 选项授予或撤消 SET USER 或 CHANGE PASSWORD 特权时，此视图将用扩展语法中的值更新。

*基础系统表上的约束*

主键 (grant\_id、 user\_id)

### **SYSROLEGRANT 系统视图**

SYSROLEGRANT 系统视图为每个授予的系统定义角色或用户定义角色包含一行。该视图的基础系统表为 ISYSROLEGRANT。

列名	数据类型	描述
grant_id	UNSIGNED INT	用于标识每个 GRANT 语句的 ID。
role_id	UNSIGNED INT	按照 ISYSUSER 被授予的角色的 ID。



列名	数据类型	描述
grantee	UNSIGNED INT	按照 ISYSUSER 被授予角色的用户的 ID。
grant_type	TINYINT	<p>使用 3 位数描述授予类型。第一位数表示是否已授予特权。第二位数表示是否已给予管理权限。第三位数表示系统特权是否可继承。</p> <ul style="list-style-type: none"> <li>• <b>001</b> - 已授予特权，但无继承权限和管理权限。仅适用于除 SYS_AUTH_DBA_ROLE 和 SYS_AUTH_REMOVE_DBA_ROLE 之外的不可继承的旧权限。</li> <li>• <b>101</b> - 已授予特权，并具有继承权限，但没有管理权限。</li> <li>• <b>110</b> - 仅授予了管理权限。</li> <li>• <b>111</b> - 已授予特权，并具有继承权限和管理权限。</li> <li>• <b>001</b> - 已授予特权，并具有管理权限，但没有继承权限。仅适用于旧权限 SYS_AUTH_DBA_ROLE 和 SYS_AUTH_REMOVE_DBA_ROLE。</li> </ul>
grant_scope	TINYINT	<p>由 SET USER 和 CHANGE PASSWORD 用于设置授予范围。可以是以下一个或多个值：</p> <ul style="list-style-type: none"> <li>• <b>001</b> - 用户列表。</li> <li>• <b>010</b> - ANY WITH ROLES</li> <li>• <b>110</b> - ANY</li> </ul>
grantor	CHAR(128)	角色授予者的唯一标识符。

### 基础系统表上的约束

主键 (grant\_ID)

UNIQUE 索引 (role\_id、grantee、grant\_scope)

### **SYSROLEGRANTS** 系统视图

SYSROLEGRANTS 系统视图与 SYSROLEGRANT 系统视图相同，但另外又包含了以下两列：角色名称（不仅仅是角色 ID）以及被授予者名称（不仅仅是用户 ID）。

列名	数据类型	描述
grant_id	UNSIGNED INT	所发出的每个授权语句的唯一标识符。
role_id	UNSIGNED INT	为用户授予的角色的唯一标识符（在 ISYSUSER 表中进行定义）。

列名	数据类型	描述
role_name	CHAR(128)	role_id 值所对应的角色的名称。
grantee	UNSIGNED INT	被授予角色的用户的唯一标识符。
grantee_name	CHAR(128)	grantee 值所对应的被授予者的名称。
grant_type	TINYINT	标识如何授予角色及其基础特权。值包括： <ul style="list-style-type: none"> <li>• 1 - 已授予基础特权，但无管理权限和特权继承权限。</li> </ul> <hr/> <p><b>注意：</b> 该值适用于除 <code>SYS_AUTH_DBA_ROLE</code> 和 <code>SYS_AUTH_REMOVE_DBA_ROLE</code> 之外的所有不可继承的旧角色。</p> <hr/> <ul style="list-style-type: none"> <li>• 3 - 已授予基础特权，有管理权限，但无特权继承权限。</li> </ul> <hr/> <p><b>注意：</b> 该值仅适用于不可继承的旧角色 <code>SYS_AUTH_DBA_ROLE</code> 和 <code>SYS_AUTH_REMOVE_DBA_ROLE</code>。</p> <hr/> <ul style="list-style-type: none"> <li>• 5 - 已授予基础特权，无管理权限，但有特权继承权限。</li> <li>• 6 - 仅授予基础特权的权限。</li> <li>• 7 - 已授予基础特权，有管理权限和特权继承权限。</li> <li>•</li> </ul>
grant_scope	TINYINT	定义授权适用的范围。值包括： <ul style="list-style-type: none"> <li>• 1 - 用户列表</li> <li>• 2 - 被授予指定角色中的成员资格的任何用户</li> <li>• 3 - 所有用户</li> </ul> <hr/> <p><b>注意：</b> 该值仅适用于 <code>SET USER</code> 和 <code>CHANGE PASSWORD</code> 系统特权，并可以存储这些值的任意有效组合。</p>
grantor	CHAR(128)	角色授予者的唯一标识符。

### **SYSSCHEDULE 系统视图**

SYSSCHEDULE 系统视图中的每一行都描述事件的触发时间（由 CREATE EVENT 的 SCHEDULE 子句指定）。该视图的基础系统表为 ISYSSCHEDULE。

列名	数据类型	说明
event_id	UNSIGNED INT	指派给每个事件的唯一编号。
sched_name	VARCHAR(128)	与事件的调度相关联的名称。
recurring	TINYINT	指出调度是否重复。

列名	数据类型	说明
start_time	TIME	调度开始时间。
stop_time	TIME	调度停止时间（如果使用了 BETWEEN）。
start_date	DATE	调度事件执行的第一个日期。
days_of_week	TINYINT	用于指出在星期几调度事件的位掩码：  x01 = 星期日 x02 = 星期一 x04 = 星期二 x08 = 星期三 x10 = 星期四 x20 = 星期五 x40 = 星期六
days_of_month	UNSIGNED INT	用于指出在该月中的哪天调度事件的位掩码。一些示例包括：  x01 = 1 号 x02 = 2 号 x40000000 = 31 号 x80000000 = 每月的最后 1 天
interval_units	CHAR(10)	EVERY 指定的间隔单位：  HH = 小时 NN = 分钟 SS = 秒
interval_amt	INTEGER	EVERY 指定的周期。

### 基础系统表上的约束

PRIMARY KEY (event\_id, sched\_name)

FOREIGN KEY (event\_id) REFERENCES SYS.ISYSEVENT (event\_id)

### SYSSERVER 系统视图

SYSSERVER 系统视图中的每一行都描述一个远程服务器。该视图的基础系统表为 ISYSSERVER。

**注意：** 以前版本的目录中包含一个 SYSSERVERS 系统表。该表已重命名为 ISYSSERVER（无 'S'），并且是本视图的基础表。

列名	数据类型	说明
srvid	UNSIGNED INT	远程服务器的标识符。
srvname	VARCHAR(128)	远程服务器的名称。
srvclass	LONG VARCHAR	服务器类（在 CREATE SERVER 语句中指定）。
srvinfo	LONG VARCHAR	服务器信息。
srvreadonly	CHAR(1)	服务器是否为只读。

*基础系统表上的约束*

PRIMARY KEY (srvid)

**SYSSOURCE 系统视图**

SYSSOURCE 系统视图中的每一行都包含 SYBJECT 系统视图中所列的一个对象的源代码（如果适用）。该视图的基础系统表为 ISYSSOURCE。

列名	数据类型	说明
object_id	UNSIGNED BIGINT	要定义其源代码的对象的内部 ID。
source	LONG VARCHAR	当创建对象时，如果 preserve_source_format 数据库选项为 On，则该列包含该对象的原始源代码。

*基础系统表上的约束*

PRIMARY KEY (object\_id)

FOREIGN KEY (object\_id) REFERENCES SYS.ISYBJECT (object\_id) MATCH UNIQUE FULL

**SYSSPATIALREFERENCESYSTEM 系统视图**

SYSSPATIALREFERENCESYSTEM 系统视图中的每一行都描述了一个在数据库中定义的 SRS。该视图的基础系统表为 ISYSSPATIALREFERENCESYSTEM。

此视图提供的信息量略微不同于 ST\_SPATIAL\_REFERENCE\_SYSTEMS 系统视图。

**注意：** 空间数据、空间参照系和空间测量单位只能在目录存储库中使用。

列名	数据类型	描述
object_id	UNSIGNED BIGINT	仅供系统使用。
owner	UNSIGNED INT	SRS 的所有者。
srs_name	CHAR(128)	SRS 的名称。

列名	数据类型	描述
srs_id	INTEGER	空间参照系的数字标识符 (SRID)。
round_earth	CHAR(1)	SRS 类型是 ROUND EARTH (Y) 还是 PLANAR (N)。
axis_order	CHAR(12)	描述数据库服务器如何相对于纬度和经度来解释点 (例如当使用 ST_Lat 和 ST_Long 方法时)。对于非地理空间参照系, 轴顺序为 x/y/z/m。对于地理空间参照系, 缺省轴顺序为经度/纬度/z/m, 同样支持纬度/经度/z/m 顺序。
snap_to_grid	DOUBLE	定义执行计算时使用的网格大小。
tolerance	DOUBLE	定义比较点时使用的精度。
semi_major_axis	DOUBLE	对于 ROUND EARTH SRS 的从椭圆体中心到赤道的距离。
semi_minor_axis	DOUBLE	对于 ROUND EARTH SRS 的从椭圆体中心到两极的距离。
inv_flattening		
		用于 ROUND EARTH SRS 中椭圆体的反转展平。 反转展平 (f) 是一个数学值, 定义椭球体的极点指向其赤道的挤压程度。值的范围从无展平 (一个正圆) 到完全展平 (一条直线)。反转展平是 1/f 的值, 如下所示: $1/f = (\text{semi\_major\_axis}) / (\text{semi\_major\_axis} - \text{semi\_minor\_axis})$
min_x	DOUBLE	坐标允许的 x 的最小值。
max_x	DOUBLE	坐标允许的 x 的最大值。
min_y	DOUBLE	坐标允许的 y 的最小值。
max_y	DOUBLE	坐标允许的 y 的最大值。
min_z	DOUBLE	坐标允许的 z 的最小值。
max_z	DOUBLE	坐标允许的 z 的最大值。
min_m	DOUBLE	坐标允许的 m 的最小值。
max_m	DOUBLE	坐标允许的 m 的最大值。
organization	LONG VARCHAR	创建空间参照系使用的坐标系的组织的名称。

列名	数据类型	描述
organization_coord- sys_id	INTEGER	创建坐标系的组织为坐标系提供的 ID。
srs_type	CHAR(11)	按 SQL/MM 标准定义的 SRS 类型。值可以是以下各项之一： <ul style="list-style-type: none"> <li>• <b>GEOGRAPHIC</b> - 此类型是针对于拥有纬度，经度（和高程）轴的基于地理配准坐标系的 SRS。这些 SRS 是 PLANAR 或 ROUND EARTH 类型。</li> <li>• <b>PROJECTED</b> - 此类型是针对于没有纬度和经度轴的基于地理配准坐标系的 SRS。这些 SRS 是 PLANAR 类型。</li> <li>• <b>ENGINEERING</b> - 此类型是针对于基于非地理配准的坐标系的 SRS。这些 SRS 是 PLANAR 类型。</li> <li>• <b>GEOCENTRIC</b> - 不支持。</li> <li>• <b>COMPOUND</b> - 不支持。</li> <li>• <b>VERTICAL</b> - 不支持。</li> </ul> 如果 srs_type 为空，则未指定类型。
linear_unit_of_ measure	UNSIGNED BIGINT	空间参照系使用的线性测量单位。
angular_unit_of_ measure	UNSIGNED BIGINT	空间参照系使用的角度测量单位。
count_in_use	UNSIGNED BIGINT	仅供内部使用。
polygon_format	LONG VARCHAR	多边形内的圆环方向。CounterClockwise、ClockWise 或 EvenOdd 格式之一。
storage_format	LONG VARCHAR	数据是以规范化格式（内部）、非规范化格式（原始）还是两种格式皆有（混合）的形式来存储。
definition	LONG VARCHAR	以 OGC 标准定义的格式的空间参照系的 WKT 定义。
transform_definition	LONG VARCHAR	当从此 SRS 到另一 SRS 转换数据时使用的转换定义设置。

基础系统表上的约束

```
PRIMARY KEY (object_id)
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id)
FOREIGN KEY (linear_unit_of_measure) REFERENCES
```

```

SYS.ISYSUNITOFMEASURE (object_id)
FOREIGN KEY (angular_unit_of_measure) REFERENCES
SYS.ISYSUNITOFMEASURE (object_id)
FOREIGN KEY (owner) REFERENCES SYS.ISYSUSER (user_id)
UNIQUE CONSTRAINT (srs_name)
UNIQUE CONSTRAINT (srs_id)

```

### **SYSSQLSERVERTYPE 系统视图**

SYSSQLSERVERTYPE 系统视图包含有关与 Adaptive Server Enterprise 兼容的信息。该视图的基础系统表为 ISYSSQLSERVERTYPE。

列名	数据类型	说明
ss_user_type	SMALLINT	Adaptive Server Enterprise 用户类型。
ss_domain_id	SMALLINT	Adaptive Server Enterprise 域 ID。
ss_type_name	VARCHAR(30)	Adaptive Server Enterprise 类型名称。
primary_sa_domain_id	SMALLINT	相应的 SAP Sybase IQ 主域 ID。
primary_sa_user_type	SMALLINT	相应的 SAP Sybase IQ 主用户类型。

基础系统表上的约束

```
PRIMARY KEY (ss_user_type)
```

### **SYSSUBPARTITIONKEY 系统视图**

以可读格式显示 ISYSSUBPARTITIONKEY 系统表中的组信息。

SYSSUBPARTITIONKEY 系统视图为 ISYSPARTITIONSCHEME 视图中描述的分区表中的 ISYSPARTITION 视图所描述分区的每列包含一行。

列名	数据类型	描述
partitioned_object_id	unsigned bigint	向每个分区对象（表或索引）分配的唯一编号。
column_id	unsigned int	可标识表的哪一列是部分分区键，此外 partitioned_object_id 和 column_id 会标识 SYSTABCOL 系统视图中描述的一列。
position	smallint	此列在分区键中的位置。0 表示分区键中的第一列。1 表示分区键中的第二列。

基础系统表上的约束：

主键 (partitioned\_object\_id、column\_id)

外键 (partitioned\_object\_id) 引用 SYS.ISYSOBJECT

**SYSSUBSCRIPTION 系统视图**

SYSSUBSCRIPTION 系统视图中的每一行均介绍一个用户 ID（必须具有 REMOTE 系统特权）对一个发布的预订。该视图的基础系统表为 ISYSSUBSCRIPTION。

列名	数据类型	说明
publication_id	UNSIGNED INT	此用户 ID 所预订的发布的标识符。
user_id	UNSIGNED INT	发布预订者的用户 ID。
subscribe_by	CHAR(128)	预订的 SUBSCRIBE BY 表达式（如果有）的值。
created	UNSIGNED BIGINT	事务日志中创建预订的位置所在的偏移。
started	UNSIGNED BIGINT	事务日志中启动预订的位置所在的偏移。

**基础系统表上的约束**

```
PRIMARY KEY (publication_id, user_id, subscribe_by)
```

```
FOREIGN KEY (publication_id) REFERENCES SYS.ISYSPUBLICATION  
(publication_id)
```

```
FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)
```

**SYSSUBSCRIPTIONS 统一视图**

每行均介绍一个用户 ID（必须具有 REMOTE 系统特权）对一个发布的预订。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSSUBSCRIPTIONS"  
  as select p.publication_name,u.user_name,s.subscribe_by,s.created,  
          s.started  
  from SYS.ISYSSUBSCRIPTION as s  
        join SYS.ISYSPUBLICATION as p on(p.publication_id =  
s.publication_id)  
        join SYS.ISYSUSER as u on u.user_id = s.user_id
```

**SYSSYNC 系统视图**

SYSSYNC 系统视图包含与同步相关的信息。该视图中的一些列包含可能比较敏感的数据。为此，访问此视图将受到限制。SYSSYNC2 视图提供了访问此视图中除可能的敏感列以外的数据的公共方法。该视图的基础系统表为 ISYSSYNC。

列名	数据类型	说明
sync_id	UNSIGNED INT	唯一标识行的编号。
type	CHAR(1)	此值始终为 D。
publication_id	UNSIGNED INT	在 SYSPUBLICATION 系统视图中找到的 publication_id。



列名	数据类型	说明
progress	UNSIGNED BIGINT	上次成功上载的日志偏移。
site_name	CHAR(128)	用户名。
"option"	LONG VARCHAR	同步选项。
server_connect	LONG VARCHAR	服务器的地址或 URL。
server_conn_type	LONG VARCHAR	进行同步时使用的通信协议（如 TCP/IP）。
last_download_time	TIMESTAMP	指出上次从 服务器接收下载流的时间。
last_upload_time	TIMESTAMP	指出上次成功上载信息的时间（在服务器上测量的时间）。缺省值为 jan-1-1900。
created	UNSIGNED BIGINT	创建预订的位置所在的日志偏移。
log_sent	UNSIGNED BIGINT	日志进度，到该进度为止的信息已被上载。不必接收到上载确认即可更新该列中的条目。
generation_number	INTEGER	对于基于文件的下载，是指接收到的此预订的上一个世代号。缺省值为 0。
extended_state	VARCHAR(1024)	仅供内部使用。
script_version	CHAR(128)	指出 CREATE 和 ALTER SYNCHRONIZATION SUBSCRIPTION 语句以及 START SYNCHRONIZATION SCHEMA CHANGE 语句使用的脚本版本。
subscription_name	CHAR(128)	预订的名称。
server_protocol	UNSIGNED BIGINT	仅供内部使用。

### 基础系统表上的约束

```
PRIMARY KEY (sync_id)
```

```
FOREIGN KEY (publication_id) REFERENCES SYS.ISYSPUBLICATION  
(publication_id)
```

```
UNIQUE INDEX (publication_id, site_name)
```

```
UNIQUE INDEX (subscription_name)
```

### **SYSSYNC2 统一视图**

SYSSYNC2 视图提供了访问 SYSSYNC 系统视图中可找到的数据（与同步相关的信息）的公共方法—而不会公开可能比较敏感的数据。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

对于没有 **SELECT ANY TABLE** 系统特权的用户，**server\_connect** 和 **option** 列中的值将隐藏。

```
ALTER VIEW "SYS"."SYSSYNC2"
  as select ISYSSYNC.sync_id,
    ISYSSYNC.type,
    ISYSSYNC.publication_id,
    ISYSSYNC.progress,
    ISYSSYNC.site_name,
    SYS.HIDE_FROM_NON_DBA(ISYSSYNC."option") as "option",
    SYS.HIDE_FROM_NON_DBA(ISYSSYNC.server_connect) as
server_connect,
    ISYSSYNC.server_conn_type,
    ISYSSYNC.last_download_time,
    ISYSSYNC.last_upload_time,
    ISYSSYNC.created,
    ISYSSYNC.log_sent,
    ISYSSYNC.generation_number,
    ISYSSYNC.extended_state,
    ISYSSYNC.script_version,
    ISYSSYNC.subscription_name
  from SYS.ISYSSYNC
```

### **SYSSYNCPUBLICATIONDEFAULTS** 统一视图

**SYSSYNCPUBLICATIONDEFAULTS** 视图提供与同步中涉及的发布关联的缺省同步设置。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

对于没有 **SELECT ANY TABLE** 系统特权的用户，**server\_connect** 和 **option** 列中的值将隐藏。

```
ALTER VIEW "SYS"."SYSSYNCPUBLICATIONDEFAULTS"
  as select s.sync_id,
    p.publication_name,
    SYS.HIDE_FROM_NON_DBA(s."option") as "option",
    SYS.HIDE_FROM_NON_DBA(s.server_connect) as server_connect,
    s.server_conn_type
  from SYS.ISYSSYNC as s join SYS.ISYSPUBLICATION as p
on(p.publication_id = s.publication_id) where
  s.site_name is null
```

### **SYSSYNCS** 统一视图

**SYSSYNCS** 视图包含与同步相关的信息。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

对于没有 **SELECT ANY TABLE** 系统特权的用户，**server\_connect** 和 **option** 列中的值将隐藏。

```
ALTER VIEW "SYS"."SYSSYNCS"
  as select p.publication_name,s.progress,s.site_name,
```

```

SYS.HIDE_FROM_NON_DBA(s."option") as "option",
SYS.HIDE_FROM_NON_DBA(s.server_connect) as server_connect,
s.server_conn_type,s.last_download_time,
s.last_upload_time,s.created,s.log_sent,s.generation_number,
s.extended_state
from SYS.ISYSSYNC as s
  left outer join SYS.ISYSPUBLICATION as p
    on p.publication_id = s.publication_id

```

### **SYSSYNCSCRIPT 系统视图**

SYSSYNCSCRIPT 系统视图的每一行都标识一个脚本式上载的存储过程。该视图几乎与 SYSSYNCSCRIPTS 视图完全相同，只是该视图的值以其原始格式显示。

该视图的基础系统表为 ISYSSYNCSCRIPT。

列名	数据类型	说明
pub_object_id	UNSIGNED BIGINT	脚本所属的发布的对象 ID。
table_object_id	UNSIGNED BIGINT	应用脚本的表的对象 ID。
type	UNSIGNED INT	上载过程的类型。
proc_object_id	UNSIGNED BIGINT	用于发布的存储过程的对象 ID。

#### *基础系统表上的约束*

```
PRIMARY KEY (pub_object_id, table_object_id, type)
```

```
FOREIGN KEY (pub_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
FOREIGN KEY (table_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
FOREIGN KEY (proc_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

### **SYSSYNCSCRIPTS 统一视图**

SYSSYNCSCRIPTS 视图的每一行都标识一个脚本式上载的存储过程。该视图几乎与 SYSSYNCSCRIPT 视图完全相同，只是其值以人工可读的格式显示，而不是显示原始数据。

```

ALTER VIEW "SYS"."SYSSYNCSCRIPTS"
  as select p.publication_name,
    t.table_name,
    case s.type
      when 0 then 'upload insert'
      when 1 then 'upload delete'
      when 2 then 'upload update'
      else 'unknown'
    end as type,
    c.proc name
  from SYS.ISYSSYNCSCRIPT as s
    join SYS.ISYSPUBLICATION as p on p.object_id = s.pub_object_id
    join SYS.ISYSTAB as t on t.object_id = s.table_object_id
    join SYS.ISYSPROCEDURE as c on c.object_id = s.proc_object_id

```

**SYSSYNCSUBSCRIPTIONS 统一视图**

SYSSYNCSUBSCRIPTIONS 视图包含与 同步预订关联的同步设置。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

对于没有 SELECT ANY TABLE 系统特权的用户，server\_connect 和 option 列中的值将隐藏。

```
ALTER VIEW "SYS"."SYSSYNCSUBSCRIPTIONS"
as select s.sync_id,
  p.publication_name,
  s.progress,
  s.site_name,
  SYS.HIDE_FROM_NON_DBA(s."option") as "option",
  SYS.HIDE_FROM_NON_DBA(s.server_connect) as server_connect,
  s.server_conn_type,
  s.last_download_time,
  s.last_upload_time,
  s.created,
  s.log_sent,
  s.generation_number,
  s.extended_state
from SYS.ISYSSYNC as s join SYS.ISYSPUBLICATION as p
on(p.publication_id = s.publication_id)
where s.publication_id is not null and
s.site_name is not null and exists
(select 1 from SYS.SYSSYNCSUSERS as u
where s.site_name = u.site_name)
```

**SYSSYNCSUSERS 统一视图**

与 同步用户关联的同步设置视图。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

对于没有 SELECT ANY TABLE 系统特权的用户，server\_connect 和 option 列中的值将隐藏。

```
ALTER VIEW "SYS"."SYSSYNCSUSERS"
as select ISYSSYNC.sync_id,
  ISYSSYNC.site_name,
  SYS.HIDE_FROM_NON_DBA(ISYSSYNC."option") as "option",
  SYS.HIDE_FROM_NON_DBA(ISYSSYNC.server_connect) as
server_connect,
  ISYSSYNC.server_conn_type
from SYS.ISYSSYNC where
ISYSSYNC.publication_id is null
```

**SYSTAB 系统视图**

SYSTAB 系统视图的每一行都描述数据库中的一个表或视图。视图的附加信息可在 SYSVIEW 系统视图中找到。该视图的基础系统表为 ISYSTAB。

列名	数据类型	说明
table_id	UNSIGNED INT	每个表都有一个指定的唯一编号（表号）。
dbspace_id	SMALLINT	指出哪个 dbspace 包含该表的值。
count	UNSIGNED BIGINT	表或实例化视图中的行数。该值在每次成功的检查点操作中都会更新。该数字用于优化数据库访问。对于非实例化视图或远程表，count 始终为 0。
creator	UNSIGNED INT	表或视图的所有者的用户号。
table_page_count	INTEGER	该基础表使用的主页总数。
ext_page_count	INTEGER	该基础表使用的扩展页总数。
commit_action	INTEGER	对于全局临时表，0 表示创建表时指定了 ON COMMIT PRESERVE ROWS 子句，1 表示创建表时指定了 ON COMMIT DELETE ROWS 子句（临时表的缺省行为），而 3 表示创建表时指定了 NOT TRANSACTIONAL 子句。对于非临时表，commit_action 始终为 0。
share_type	INTEGER	对于全局临时表，4 表示创建表时指定了 SHARE BY ALL 子句，而 5 表示创建表时未指定 SHARE BY ALL 子句。对于非临时表，share_type 始终为 5，因为在创建非临时表时无法指定 SHARE BY ALL 子句。
object_id	UNSIGNED BIGINT	表的对象 ID。
last_modified_at	TIMESTAMP	上次修改表中数据的本地时间。该列仅在检查点时间更新。
table_name	CHAR(128)	表或视图的名称。一个创建者不能有两个同名的表或视图。

列名	数据类型	说明
table_type	TINYINT	表或视图的类型。值包括： <ul style="list-style-type: none"> <li>• 1 - 基表</li> <li>• 2 - 实例化视图</li> <li>• 3 - 全局临时表</li> <li>• 4 - 本地临时表</li> <li>• 5 - 文本索引基表</li> <li>• 6 - 文本索引全局临时表</li> <li>• 21 - 视图</li> </ul>
replicate	CHAR(1)	此值仅供内部使用。
server_type	TINYINT	基础表的数据位置。值包括： <ul style="list-style-type: none"> <li>• 1 - 本地服务器</li> <li>• 3 - 远程服务器</li> </ul>
tab_page_list	LONG VARBIT	仅供内部使用。包含表的信息的页集，用位图表示。
ext_page_list	LONG VARBIT	仅供内部使用。包含表的行扩展和大对象 (LOB) 页的页集，用位图表示。
pct_free	UNSIGNED INT	表的 PCT_FREE 规范 (如果已经指定了一个) ; 否则为 NULL。
clustered_index_id	UNSIGNED INT	表的聚簇索引的 ID。如果没有聚簇索引，则该字段为 NULL。
encrypted	CHAR(1)	是否加密表或实例化视图。
last_modified_tsn	UNSIGNED BIGINT	分配给修改表的事务的序列号。该列仅在检查点时间更新。
current_schema	UNSIGNED INT	表的当前模式版本。
file_id	SMALLINT	不建议使用。此列存在于 SYSVIEW 中，但不存在于基础系统表 ISYSTAB 中。此列的内容与 dbspace_id 相同，并且是为了兼容而提供的。请改用 dbspace_id。
table_type_str	CHAR(13)	table_type 的可读值。值包括： <ul style="list-style-type: none"> <li>• <b>BASE</b> - 基表</li> <li>• <b>MAT VIEW</b> - 实例化视图</li> <li>• <b>GBL TEMP</b> - 全局临时表</li> <li>• <b>VIEW</b> - 视图</li> </ul>

列名	数据类型	说明
last_modified_at_utc	TIMESTAMP WITH TIME ZONE	上次修改表中数据的 UTC 时间。该列仅在检查点时间更新。

### 基础系统表上的约束

```
FOREIGN KEY (dbspace_id) REFERENCES SYS.ISYSDBSPACE (dbspace_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
PRIMARY KEY (table_id)
```

```
FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)
```

```
UNIQUE INDEX (table_name, creator)
```

### **SYSTABAUTH** 统一视图

SYSTABAUTH 视图包含来自 SYSTABLEPERM 系统视图的信息，但采用了更加易读的格式。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSTABAUTH" ( grantor,
    grantee, screator, stname, tcreator, tname,
    selectauth, insertauth, deleteauth,
    updateauth, updatecols, alterauth, referenceauth,
    loadauth, truncateauth )
as select u1.user_name, u2.user_name, u3.user_name, tab1.table_name,
    u4.user_name, tab2.table_name, tp.selectauth, tp.insertauth,
    tp.deleteauth, tp.updateauth, tp.updatecols, tp.alterauth,
    tp.referenceauth, tp.loadauth, tp.truncateauth
from SYS.ISYSTABLEPERM as tp
    join SYS.ISYSUSER as u1 on u1.user_id = tp.grantor
    join SYS.ISYSUSER as u2 on u2.user_id = tp.grantee
    join SYS.ISYSTAB as tab1 on tab1.table_id = tp.stable_id
    join SYS.ISYSUSER as u3 on u3.user_id = tab1.creator
    join SYS.ISYSTAB as tab2 on tab2.table_id = tp.stable_id
    join SYS.ISYSUSER as u4 on u4.user_id = tab2.creator
```

### **SYSTABCOL** 系统视图

SYSTABCOL 系统视图为数据库中每个表和视图的每列都包含一行。该视图的基础系统表为 ISYSTABCOL。

列名	数据类型	说明
table_id	UNSIGNED INT	该列所属的表或视图的对象 ID。

列名	数据类型	说明
column_id	UNSIGNED INT	列的 ID。各个表的列的编号都是从 1 开始。 当使用 SELECT * 时 column_id 值决定结果集中的列的顺序。当未提供列名的列表时它也决定 INSERT 语句的列的顺序。
domain_id	SMALLINT	该列的数据类型，由 SYSDOMAIN 系统视图中所列的数据类型编号标识。
nulls	CHAR(1)	指出该列中是否允许 NULL 值。
width	BIGINT	字符串列的长度、数字列的精度或任何其它数据类型的存储字节数。
scale	SMALLINT	对于 NUMERIC 或 DECIMAL 数据类型的列，是指小数点后的位数。对于字符串列，值 1 表示字符长度语义，而 0 表示字节长度语义。
object_id	UNSIGNED BIGINT	表列的对象 ID。
max_identity	BIGINT	如果该列为 AUTOINCREMENT、IDENTITY 或 GLOBAL AUTOINCREMENT 列，则为列的最大值。
column_name	CHAR(128)	列的名称。
"default"	LONG VARCHAR	列的缺省值。如果指定，则仅在 INSERT 语句没有指定列值时才使用该值。
user_type	SMALLINT	使用用户定义的数据类型定义列时的数据类型。
column_type	CHAR(1)	列的类型（C 表示计算列，而 R 表示其它）。
compressed	TINYINT	该列是否以压缩格式存储。
collect_stats	TINYINT	系统是否自动收集并更新该列的统计信息。
inline_max	SMALLINT	行中存储 BLOB 的最大字节数。NULL 值表示已应用缺省值，或该列不是字符或二进制类型。非 NULL 的 inline_max 值与使用 CREATE TABLE 或 ALTER TABLE 语句为列指定的 INLINE 值相对应。
inline_long	SMALLINT	当 BLOB 的大小超过 inline_max 值时，在行中存储的 BLOB 的重复字节数。NULL 值表示已应用缺省值，或该列不是字符或二进制类型。非 NULL 的 inline_long 值与使用 CREATE TABLE 或 ALTER TABLE 语句为列指定的 PREFIX 值相对应。



列名	数据类型	说明
lob_index	TINYINT	是否对列中超过内部阈值大小（约八个数据库页）的 BLOB 值建立索引。NULL 值表示已应用缺省值，或该列不是 BLOB 类型。值 1 表示将建立索引。值 0 表示不建立索引。非 NULL 的 lob_index 值与使用 CREATE TABLE 或 ALTER TABLE 语句为列指定了 INDEX 还是 NO INDEX 相对应。
base_type_str	VARCHAR(32,767)	表示列的物理类型的加批注类型字符串。
nonmaterialized_value	LONG BINARY	仅供内部使用。
start_schema	UNSIGNED INT	包含此列的表模式的第一版。

### 基础系统表上的约束

```
PRIMARY KEY (table_id, column_id)
```

```
FOREIGN KEY (table_id) REFERENCES SYS.ISYSTAB (table_id)
```

```
FOREIGN KEY (domain_id) REFERENCES SYS.ISYSDOMAIN (domain_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (user_type) REFERENCES SYS.ISYSUSERTYPE (type_id)
```

### **SYSTABLE 兼容性视图（不建议使用）**

提供 SYSTABLE 视图的目的是为了与提供 SYSTABLE 系统表的旧版本软件兼容。但以前的 SYSTABLE 系统表已被 ISYSTAB 系统表所取代，并且您应当使用与该系统表对应的 ISYSTAB 系统视图。

SYSTABLE 视图的每一行都描述数据库中的一个表。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSTABLE"  
as select b.table_id,  
       b.file_id,  
       b.count,  
       0 as first_page,  
       b.commit_action as last_page,  
       COALESCE(ph.root,0) as primary_root,  
       b.creator,  
       0 as first_ext_page,  
       0 as last_ext_page,  
       b.table_page_count,  
       b.ext_page_count,  
       b.object_id,  
       b.table_name,  
       b.table_type_str as table_type,
```

```

v.view_def,
r.remarks,
b.replicate,
p.existing_obj,
p.remote_location,
'T' as remote_objtype,
p.srvid,
case b.server_type
when 1 then 'SA'
when 2 then 'IQ'
when 3 then 'OMNI'
else 'INVALID'
end as server_type,
10 as primary_hash_limit,
0 as page_map_start,
s.source,
b."encrypted"
from SYS.SYSTAB as b
  left outer join SYS.ISYSREMARK as r on(b.object_id =
r.object_id)
  left outer join SYS.ISYSSOURCE as s on(b.object_id =
s.object_id)
  left outer join SYS.ISYSVIEW as v on(b.object_id =
v.view_object_id)
  left outer join SYS.ISYSPROXYTAB as p on(b.object_id =
p.table_object_id)
  left outer join(SYS.ISYSIDX as i left outer join SYS.ISYSPHYSIDX
as ph on(i.table_id = ph.table_id
  and i.phys_index_id = ph.phys_index_id)) on(b.table_id =
i.table_id and i.index_category = 1
  and i.index_id = 0)

```

### **SYSTABLEPERM 系统视图**

通过 GRANT 语句授予表和视图的特权存储在 SYSTABLEPERM 系统视图中。该视图中的每一行都对应于一个表、一个授予特权的用户 ID (grantor) 和一个被授予权限的用户 ID (grantee)。该视图的基础系统表为 ISYSTABLEPERM。

列名	数据类型	说明
stable_id	UNSIGNED INT	应用了特权的表或视图的表号。
grantee	UNSIGNED INT	接收特权的用户 ID 的用户号。
grantor	UNSIGNED INT	授予特权的用户 ID 的用户号。
selectauth	CHAR(1)	指出是否已授予 SELECT 特权。可能的值为 Y、N 或 G。有关这些值的含义的详细信息，请参见下面的注释区域。
insertauth	CHAR(1)	指出是否已授予 INSERT 特权。可能的值为 Y、N 或 G。有关这些值的含义的详细信息，请参见下面的注释区域。

列名	数据类型	说明
deleteauth	CHAR(1)	指出是否已授予 DELETE 特权。可能的值为 Y、N 或 G。有关这些值的含义的详细信息，请参见下面的注释区域。
updateauth	CHAR(1)	指出是否已为表中的所有列授予 UPDATE 特权。可能的值为 Y、N 或 G。有关这些值的含义的详细信息，请参见下面的注释区域。
updatecols	CHAR(1)	指出是否仅为基础表中的某些列授予了 UPDATE 特权。如果 updatecols 的值为 Y，则 SYSCOLPERM 系统视图中将有一行或多行为该列授予 UPDATE 特权。
alterauth	CHAR(1)	指出是否已授予 ALTER 特权。可能的值为 Y、N 或 G。有关这些值的含义的详细信息，请参见下面的注释区域。
referenceauth	CHAR(1)	指出是否已授予 REFERENCE 特权。可能的值为 Y、N 或 G。有关这些值的含义的详细信息，请参见下面的注释区域。
loadauth	CHAR(1)	指出是否已授予 LOAD 特权。可能的值为 Y、N 或 G。有关这些值的含义的详细信息，请参见下面的注释区域。
truncateauth	CHAR(1)	指出是否已授予 TRUNCATE 特权。可能的值为 Y、N 或 G。有关这些值的含义的详细信息，请参见下面的注释区域。
loadauth	CHAR(1)	指出是否已授予 LOAD 特权。可能的值为 Y、N 或 G。有关这些值的含义的详细信息，请参见下面的注释区域。
truncateauth	CHAR(1)	指出是否已授予 TRUNCATE 特权。可能的值为 Y、N 或 G。有关这些值的含义的详细信息，请参见下面的注释区域。

### 注释

有多种可以授予的特权。每种特权都可以有下列三个值之一。

- **N** - 否，授予者没有授予被授予者该特权。
- **Y** - 是，授予者已授予被授予者该特权。
- **G** - 被授予者已被授予该特权并能向其他用户授予同样的特权。

**注意：** 另一个授予者可能已授予被授予者操作同一个表的特权。如果是这样，可在 SYSTABLEPERM 系统视图的不同行中找到该信息。

### 基础系统表上的约束

```
PRIMARY KEY (stable_id, grantee, grantor)
```

```
FOREIGN KEY (stable_id) REFERENCES SYS.ISYSTAB (table_id)
```

```
FOREIGN KEY (grantor) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (grantee) REFERENCES SYS.ISYSUSER (user_id)
```

**SYSTEXTCONFIG 系统视图**

SYSTEXTCONFIG 系统视图中的每一行都描述了一个文本配置对象，以与全文搜索功能配合使用。该视图的基础系统表为 ISYSTEXTCONFIG。

列名	数据类型	说明
object_id	UNSIGNED BIGINT	文本配置对象的对象 ID。
creator	UNSIGNED INT	文本配置对象的创建者。
term_breaker	TINYINT	用于将字符串分隔为术语或字的算法。对于 <b>GENERIC</b> ，值为 0；对于 <b>NGRAM</b> ，值为 1。 <b>GENERIC</b> 算法会将所有由一个或多个字母数字字符构成并由非字母数字分隔的字符串视为一个术语。 <b>NGRAM</b> 用于近似匹配或不使用空格分隔术语的文档。
stemmer	TINYINT	仅供内部使用。
min_term_length	TINYINT	术语所允许的最小长度（以字符数表示）。忽略长度小于 <b>min_term_length</b> 的术语。  <b>MINIMUM TERM LENGTH</b> 设置仅对 <b>GENERIC</b> 术语断开器有意义。对于 <b>NGRAM</b> 文本索引，将忽略该设置。
max_term_length	TINYINT	对于 <b>GENERIC</b> 文本索引，术语所允许的最大长度（以字符数表示）。忽略长度大于 <b>max_term_length</b> 的术语。  对于 <b>NGRAM</b> 文本索引，该设置是用于断开术语的 <b>n</b> 元语法词的长度。
collation	CHAR(128)	仅供内部使用。
text_config_name	CHAR(128)	文本配置对象的名称。
prefilter	LONG VARCHAR	外部前置过滤器库的函数和库的名称。
postfilter	LONG VARCHAR	仅供内部使用。
char_stoplist	LONG VARCHAR	在 <b>CHAR</b> 列上执行全文搜索时要忽略的术语。还可以从文本索引中忽略这些术语。在从 <b>default_char</b> 创建文本配置对象时使用此列。
nchar_stoplist	LONG NVARCHAR	在 <b>NCHAR</b> 列上执行全文搜索时要忽略的术语。还可以从文本索引中忽略这些术语。在从 <b>default_nchar</b> 创建文本配置对象时使用此列。

列名	数据类型	说明
external_term_breaker	LONG VARCHAR	外部术语断开器库的函数和库的名称。

### 基础系统表上的约束

PRIMARY KEY (object\_id)

FOREIGN KEY (object\_id) REFERENCES SYS.ISYSOBJECT (object\_id) MATCH  
UNIQUE FULL

FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user\_id)

UNIQUE INDEX (creator, text\_config\_name)

### **SYSTEXTIDX** 系统视图

SYSTEXTIDX 系统视图中的每一行都描述一个文本索引。该视图的基础系统表为 ISYSTEXTIDX。

列名	数据类型	说明
index_id	UNSIGNED BIGINT	SYSDIX 中文本索引的对象 ID。
sequence	UNSIGNED INT	仅供内部使用。
status	UNSIGNED INT	仅供内部使用。
text_config	UNSIGNED BIGINT	SYSTEXTCONFIG 中文本配置对象的对象 ID。
next_handle	UNSIGNED INT	仅供内部使用。
last_handle	UNSIGNED INT	仅供内部使用。
deleted_length	UNSIGNED BIGINT	文本索引中已删除的索引值的总大小。
pending_length	UNSIGNED BIGINT	将要在下次刷新时添加到文本索引的索引值的总大小。
refresh_type	TINYINT	刷新的类型。以下值之一： <ul style="list-style-type: none"> <li>• 1 - MANUAL</li> <li>• 2 - AUTO</li> <li>• 3 - IMMEDIATE</li> </ul>
refresh_interval	UNSIGNED INT	AUTO REFRESH 间隔（以分钟为单位）。
last_refresh	TIMESTAMP	上次刷新的本地时间。
last_refresh_utc	TIMESTAMP WITH TIME ZONE	上次刷新的 UTC 时间。

### 基础系统表上的约束

PRIMARY KEY (index\_id, sequence)

```
FOREIGN KEY (index_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
FOREIGN KEY (text_config) REFERENCES SYS.ISYSOEXTXTCONFIG (object_id)
```

### **SYSTEXTIDXTAB** 系统视图

**SYSTEXTIDXTAB** 系统视图中的每一行都描述了已生成的表，而该表是文本索引的一部分。该视图的基础系统表为 **ISYSOEXTXTIDXTAB**。

列名	数据类型	说明
index_id	UNSIGNED BIGINT	仅供内部使用。
sequence	UNSIGNED INT	仅供内部使用。
table_type	UNSIGNED INT	仅供内部使用。
table_id	UNSIGNED INT	仅供内部使用。

#### *基础系统表上的约束*

```
PRIMARY KEY (index_id, sequence, table_type)
```

```
FOREIGN KEY (index_id, sequence) REFERENCES SYS.ISYSOEXTXTIDXTAB (index_id, sequence)
```

```
FOREIGN KEY (table_id) REFERENCES SYS.ISYSOEXTXTIDXTAB (table_id)
```

### **SYSTRIGGER** 系统视图

**SYSTRIGGER** 系统视图的每一行都描述数据库中的一个触发器。该视图还包含自动为具有参照触发操作（例如 **ON DELETE CASCADE**）的外键定义创建的触发器。该视图的基础系统表为 **ISYSOEXTXTIDXTAB**。

列名	数据类型	说明
trigger_id	UNSIGNED INT	<b>SYSTRIGGER</b> 视图中触发器的唯一编号。
table_id	UNSIGNED INT	触发器所属的表的表 ID。
object_id	UNSIGNED BIGINT	数据库中触发器的对象 ID。

列名	数据类型	说明
event	CHAR(1)	此操作将导致触发器的触发。 <ul style="list-style-type: none"> <li>• <b>A</b> - INSERT, DELETE</li> <li>• <b>B</b> - INSERT, UPDATE</li> <li>• <b>C</b> - UPDATE COLUMNS</li> <li>• <b>D</b> - DELETE</li> <li>• <b>E</b> - DELETE, UPDATE</li> <li>• <b>I</b> - INSERT</li> <li>• <b>M</b> - INSERT, DELETE, UPDATE</li> <li>• <b>U</b> - UPDATE</li> </ul>
trigger_time	CHAR(1)	触发器相对于事件触发时的时间。 <ul style="list-style-type: none"> <li>• <b>A</b> - AFTER (行级触发器)</li> <li>• <b>B</b> - BEFORE (行级触发器)</li> <li>• <b>I</b> - INSTEAD OF (行级触发器)</li> <li>• <b>K</b> - INSTEAD OF (语句级触发器)</li> <li>• <b>R</b> - RESOLVE</li> <li>• <b>S</b> - AFTER (语句级触发器)</li> </ul>
trigger_order	SMALLINT	相同类型 (插入、更新或删除) 的多个触发器设置为同时触发时 (只应用于 BEFORE 或 AFTER 触发器) 触发器的触发顺序。
foreign_table_id	UNSIGNED INT	包含有参照触发操作 (如 ON DELETE CASCADE) 的外键定义的表的 ID。foreign_table_id 值反映 ISYSIDX.table_id 值。
foreign_key_id	UNSIGNED INT	foreign_table_id 引用的表的外键 ID。foreign_key_id 值反映 ISYSIDX.index_id 值。
referential_action	CHAR(1)	外键所定义的操作。该单字符值对应于创建外键时指定的操作。 <ul style="list-style-type: none"> <li>• <b>C</b> - CASCADE</li> <li>• <b>D</b> - SET DEFAULT</li> <li>• <b>N</b> - SET NULL</li> <li>• <b>R</b> - RESTRICT</li> </ul>
trigger_name	CHAR(128)	触发器的名称。一个表中不能有两个同名的触发器。
trigger_defn	LONG VARCHAR	用于创建触发器的命令。
remarks	LONG VARCHAR	有关触发器的注释。该值存储在 ISYSREMARK 系统表中。





```

endif, trig.trigger_defn
from SYS.ISYSTRIGGER as trig
    join SYS.ISYSTAB as tab on(tab.table_id = trig.table_id)
    join SYS.ISYSUSER as u on u.user_id = tab.creator where
trig.foreign_table_id is null

```

### **SYSTYPEMAP 系统视图**

SYSTYPEMAP 系统视图包含 SYSSQLSERVERTYPE 系统视图中条目的兼容性映射值。该视图的基础系统表为 ISYSTYPEMAP。

列名	数据类型	说明
ss_user_type	SMALLINT	包含 Adaptive Server Enterprise 用户类型。
sa_domain_id	SMALLINT	包含相应的 SAP Sybase IQ domain_id。
sa_user_type	SMALLINT	包含相应的 SAP Sybase IQ 用户类型。
nullable	CHAR(1)	该类型是否允许 NULL 值。

#### *基础系统表上的约束*

```
FOREIGN KEY (sa_domain_id) REFERENCES SYS.ISYSDOMAIN (domain_id)
```

### **SYSTYPES ASE 兼容性视图**

每个系统提供的数据类型和用户定义数据类型在 systypes 中都有对应的一行。域（由规则定义）和缺省值将在其存在时提供。

此视图由用户 DBO 所有。不能修改用于说明系统提供的数据类型的行。

#### **另请参见**

- 每个 Adaptive Server 数据库中的表（第 751 页）
- SYSCOLUMNS ASE 兼容性视图（第 665 页）
- SYSCOMMENTS ASE 兼容性视图（第 665 页）
- SYSINDEXES ASE 兼容性视图（第 681 页）
- SYSIQOBJECTS ASE 兼容性视图（第 691 页）
- SYSIQVINDEXT ASE 兼容性视图（第 695 页）
- SYSOBJECTS ASE 兼容性视图（第 702 页）
- SYSUSERS ASE 兼容性视图（第 747 页）

**SYSUSER 系统视图**

**SYSUSER** 系统视图的每一行都描述系统中的一个用户。**Standalone** 角色也同样存储于该视图中，但仅有 **user\_id**、**object\_id**、**user\_name** 和 **user\_type** 几列对这些角色而言是有意义的。该视图的基础系统表为 **ISYSUSER**。

列名	数据类型	说明
user_id	UNSIGNED INT	被分配至登录策略的用户的唯一标识符。
object_id	UNSIGNED BIGINT	用户在数据库中的唯一标识符。
user_name	CHAR(128)	用户的登录名。
password	BINARY(128)	用户的口令。出于安全性考虑，该列中的数据仅对具有 <b>SELECT ANY TABLE</b> 系统特权的用户可见。
login_policy_id	UNSIGNED BIGINT	登录策略的唯一标识符。
expired_password_on_login	TINYINT	指示下次登录时用户的口令是否到期的值。
password_creation_time	TIMESTAMP	创建用户口令的本地时间。
failed_login_attempts	UNSIGNED INT	用户在帐户被锁定前可以登录失败的次数。
last_login_time	TIMESTAMP	用户上次登录时的本地时间。
user_type	TINYINT	指示用户是常规用户、角色、还是扩展为角色的用户的值。并可指示用户、角色、或扩展角色是否可变更（可变）或删除。可能值包括： <ul style="list-style-type: none"> <li>• 1 - 不可变的系统角色。</li> <li>• 5 - 可变的系统角色。</li> <li>• 9 - 不可变但可删除的系统角色。</li> <li>• 12 - 可变并可删除的用户。</li> <li>• 13 - 可变并可删除的角色。</li> <li>• 14 - 扩展为角色的可变并可删除的用户。</li> </ul>
user_dn	CHAR(1024)	在某域内或多个域间唯一的用户的 LDAP 可分辨名称 (DN) 标识符。DN 用于通过 LDAP 服务器进行验证。
user_dn_cached_at	TIMESTAMP	<b>user_dn</b> 列上次高速缓存的时间。此值用于确定是否清除旧 DN。无论数据库服务器本地时区是什么，该值都以协调通用时间 (UTC) 形式存储。

列名	数据类型	说明
password_creation_time_utc	TIMESTAMP WITH TIME ZONE	创建用户口令的 UTC 时间。
last_login_time_utc	TIMESTAMP WITH TIME ZONE	用户上次登录的 UTC 时间。
dual_password	BINARY(128)	双重用户口令的第一和/或第二部分。出于安全性考虑，该列中的数据仅对具有 <b>SELECT ANY TABLE</b> 系统特权的用户可见。
lock_time	TIMESTAMP	用户由于尝试登录失败而被锁定的时间戳。

### 基础系统表上的约束

```
PRIMARY KEY (user_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (login_policy_id) REFERENCES SYS.ISYSLOGINPOLICY  
(login_policy_id)
```

```
UNIQUE INDEX (user_name)
```

### **SYSUSERAUTH** 兼容性视图 (不建议使用)

**SYSUSERAUTH** 视图的每一行描述一个用户，但不公开其 `user_id`。而是用用户名来标识每个用户。由于该视图显示口令，您必须具有 **SELECT ANY TABLE** 系统特权才能查看其数据。

提供 **SYSUSERAUTH** 视图的目的是为了与旧版本软件兼容。改用 **SYSROLEGRANTS** 统一视图。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

**注意：** 尽管该视图的标题包含文字 `auth` (代表 `authorities`)，但是安全模式仍基于角色和特权。因此视图中的数据使用来自视图定义中所提及表和视图中的角色信息进行编译。

```
ALTER VIEW "SYS"."SYSUSERAUTH" ( name,  
    password,resourceauth,dbaauth,scheduleauth,user_group )  
    as select  
SYSUSERPERM.user_name,SYSUSERPERM.password,SYSUSERPERM.resourceauth  
,SYSUSERPERM.dbaauth,SYSUSERPERM.scheduleauth,SYSUSERPERM.user_grou  
P  
    from SYS.SYSUSERPERM
```

**SYSUSERAUTHORITY 兼容性视图 (不建议使用)**

提供 SYSUSERAUTHORITY 视图的目的是为了与旧版本软件兼容。改用 SYSROLEGRANTS 统一视图。

SYSUSERAUTHORITY 系统视图的每行描述授予一个用户 ID 的一个权限。

**注意：** 尽管该视图的标题包含文字 **authority**，但是安全模式仍基于角色和特权。因此视图中的数据使用来自视图定义中所提及表和视图中的角色信息进行编译。

```
ALTER VIEW "SYS"."SYSUSERAUTHORITY" as
  select ISYSROLEGRANT.grantee as user_id,
     sp_auth_sys_role_info.auth
  from SYS.ISYSROLEGRANT
     natural join dbo.sp_auth_sys_role_info()
  where ISYSROLEGRANT.grant_type <> (0x02|0x04) and
     not ISYSROLEGRANT.grantee = any(select
sp_auth_sys_role_info.role_id from dbo.sp_auth_sys_role_info())
union
  select ISYSUSER.user_id,
     cast('GROUP' as varchar(20)) as auth
  from SYS.ISYSUSER
     where ISYSUSER.user_name
in( 'SYS','PUBLIC','diagnostics','SYS_SPATIAL_ADMIN_ROLE','rs_syssta
bgroup','SA_DEBUG','dbo' ) union
  select ISYSUSER.user_id,
     cast('GROUP' as varchar(20)) as auth
  from SYS.ISYSUSER
     where ISYSUSER.user_type = (0x02|0x04|0x08) union
  select cast(opt.setting as unsigned integer) as user_id,
     cast('PUBLISH' as varchar(20)) as auth
  from SYS.ISYSOPTION as opt
     where opt."option" like '%db_publisher%' and opt.setting not like
'%-1%'
```

**SYSUSERLIST 兼容性视图 (不建议使用)**

提供 SYSUSERAUTH 视图的目的是为了与旧版本软件兼容。

SYSUSERLIST 视图的每一行都描述一个用户，但不公开其 **user\_id** 和口令。每个用户都由其用户名来标识。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSUSERLIST"( name,
  resourceauth, dbaauth, scheduleauth, user_group )
as select
SYSUSERPERM.user_name, SYSUSERPERM.resourceauth, SYSUSERPERM.dbaauth,
SYSUSERPERM.scheduleauth, SYSUSERPERM.user_group
  from SYS.SYSUSERPERM
```

**SYSUSERMESSAGE 系统视图**

**SYSUSERMESSAGE** 系统视图中的每一行都包含有关错误情况的用户定义消息。该视图的基础系统表为 **ISYSUSERMESSAGE**。

**注意：** 以前版本的目录中包含一个 **SYSUSERMESSAGES** 系统表。该表已重命名为 **ISYSUSERMESSAGE** (无 'S')，并且是本视图的基础表。

列名	数据类型	说明
error	INTEGER	错误情况的唯一标识号。
uid	UNSIGNED INT	定义消息的用户号。
description	VARCHAR(255)	对应于错误情况的消息。
langid	SMALLINT	保留。

*基础系统表上的约束*

```
FOREIGN KEY (uid) REFERENCES SYS.ISYSUSER (user_id)
```

```
UNIQUE CONSTRAINT (error, langid)
```

**SYSUSEROPTIONS 统一视图**

**SYSUSEROPTIONS** 视图包含了对每个用户都有效的选项设置。如果用户没有某个选项的设置，该视图将显示该选项的公共设置。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSUSEROPTIONS"( user_name,
  "option",setting )
as select u.user_name,
  o."option",
  isnull((select s.setting
    from SYS.ISYSOPTION as s
    where s.user_id = u.user_id
    and s."option" = o."option"),
  o.setting)
from SYS.SYSOPTIONS as o,SYS.ISYSUSER as u
where o.user_name = 'PUBLIC'
```

**SYSUSERPERM 兼容性视图 (不建议使用)**

**SYSUSERPERM** 视图的每一行都描述一个用户 ID。您必须具有 **SELECT ANY TABLE** 系统特权才能查看此视图中的数据。

不建议使用该视图，因为它只显示以前版本中可用的特权和权限。您应该更改应用程序以使用 **SYSROLEGRANTS** 统一视图。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```

ALTER VIEW "SYS"."SYSUSERPERM"
as select b.user_id,
  b.object_id,
  b.user_name,
  b.password,
  if AA.resourceauth is not null and AA.resourceauth > 0 then
    'Y' else 'N' endif as resourceauth,
  if AA.dbauth is not null and AA.dbauth > 0 then
    'Y' else 'N' endif as dbauth,
  'N' as scheduleauth,
  if exists(select * from SYS.ISYSOPTION as opt
    where opt."option" like '%db_publisher%' and opt.setting not
like '%-1'
    and b.user_id = cast(opt.setting as integer)) then
    'Y' else 'N' endif as publishauth,
  if AA.remotedbauth is not null and AA.remotedbauth > 0 then
    'Y' else 'N' endif as remotedbauth,
  if b.user_type = (0x02|0x04|0x08) or b.user_name
in( 'SYS','PUBLIC','diagnostics','SYS_SPATIAL_ADMIN_ROLE','rs_systa
bgroup','SA_DEBUG','dbo' ) then
    'Y' else 'N' endif as user_group,
  r.remarks
from SYS.ISYSUSER as b
  left outer join SYS.ISYSREMARK as r on(b.object_id =
r.object_id)
  left outer join(select sum(if sp_auth_sys_role_info.auth =
'RESOURCE' then 1 else 0 endif) as resourceauth,
  sum(if sp_auth_sys_role_info.auth = 'DBA' then 1 else 0 endif)
as dbauth,
  sum(if sp_auth_sys_role_info.auth = 'REMOTE DBA' then 1 else 0
endif) as remotedbauth,
  ISYSROLEGRANT.grantee
  from SYS.ISYSROLEGRANT natural join
dbo.sp_auth_sys_role_info()
  where ISYSROLEGRANT.grant_type <> (0x02|0x04)
  and sp_auth_sys_role_info.auth in( 'DBA','RESOURCE','REMOTE
DBA' )
  group by ISYSROLEGRANT.grantee) as AA
on(AA.grantee = b.user_id)

```

### **SYSUSERPERMS 兼容性视图 (不建议使用)**

不建议使用该视图，因为它只显示以前版本中可用的特权和权限。您应该更改应用程序以使用 **SYSROLEGRANTS** 统一视图。

**SYSUSERPERM** 视图的每一行都描述一个用户 ID。但不包括口令信息。所有用户都被允许从该视图中读取。

在以下的 **SQL** 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```

ALTER VIEW "SYS"."SYSUSERPERMS"
as select
SYSUSERPERM.user_id,SYSUSERPERM.user_name,SYSUSERPERM.resourceauth,
SYSUSERPERM.dbauth,

```

```
SYSUSERPERM.scheduleauth,SYSUSERPERM.user_group,SYSUSERPERM.publish
auth,SYSUSERPERM.remotedbaauth,SYSUSERPERM.remarks
from SYS.SYSUSERPERM
```

### **SYSUSERTYPE 系统视图**

SYSUSERTYPE 系统视图的每一行都包含对一种用户定义数据类型的描述。该视图的基础系统表为 ISYSUSERTYPE。

列名	数据类型	说明
type_id	SMALLINT	用户定义数据类型的唯一标识号。
creator	UNSIGNED INT	数据类型所有者的用户号。
domain_id	SMALLINT	该用户定义数据类型所依据的数据类型，由 SYSDOMAIN 系统视图中列出的数据类型号指出。
nulls	CHAR(1)	用户定义的数据类型是否允许空值。可能的值为 Y、N 或 U。值为 U 表示未指定空性。
width	BIGINT	字符串列的长度、数字列的精度或任何其它数据类型的存储字节数。
scale	SMALLINT	数字数据类型列的小数点后的位数，对于所有其它数据类型该值为零。
type_name	CHAR(128)	数据类型的名称。
"default"	LONG VARCHAR	数据类型的缺省值。
"check"	LONG VARCHAR	数据类型的 CHECK 条件。
base_type_str	VARCHAR(32767)	表示用户类型的物理类型的加批注类型字符串。

#### *基础系统表上的约束*

```
PRIMARY KEY (type_id)
```

```
FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (domain_id) REFERENCES SYS.ISYSDOMAIN (domain_id)
```

```
UNIQUE CONSTRAINT (type_name)
```

### **SYSUSERS ASE 兼容性视图**

数据库中允许的每个用户以及每个组或角色在 sysusers 中都有对应的一行。

此视图由用户 DBO 所有。

#### **另请参见**

- 每个 Adaptive Server 数据库中的表 (第 751 页)
- SYSCOLUMNS ASE 兼容性视图 (第 665 页)
- SYSCOMMENTS ASE 兼容性视图 (第 665 页)
- SYSINDEXES ASE 兼容性视图 (第 681 页)

- SYSIQOBJECTS ASE 兼容性视图 (第 691 页)
- SYSIQVINDEXT ASE 兼容性视图 (第 695 页)
- SYSOBJECTS ASE 兼容性视图 (第 702 页)
- SYSTYPES ASE 兼容性视图 (第 741 页)

### **SYSVIEW 系统视图**

SYSVIEW 系统视图的每一行都描述数据库中的一个视图。

您可以在 SYSTAB 系统视图中找到有关视图的其它信息。该视图的基础系统表为 ISYSVIEW。

还可以使用 **sa\_materialized\_view\_info** 系统过程获取可读格式的实例化视图信息。仅 IQ 目录存储库中的 SQL Anywhere 表支持实例化视图。

列名	数据类型	描述
view_object_id	UNSIGNED BIGINT	视图的对象 ID。
view_def	LONG VARCHAR	视图的定义 (查询说明)。
mv_build_type	TINYINT	目前未使用。
mv_refresh_type	TINYINT	为视图定义的刷新类型。可能的值为 IMMEDIATE (1) 和 MANUAL (2)。
mv_use_in_optimization	TINYINT	实例化视图是否可用于查询优化 (0 表示不能用于优化, 1 表示可以用于优化)
mv_last_refreshed_at	TIMESTAMP	表示上次刷新实例化视图的本地日期和时间。
mv_known_stale_at	TIMESTAMP	实例化视图失效的本地时间。该值对应于检测到一个基表发生更改的时间。值为 0 表示该视图为最新的, 或已失效但数据库服务器尚未对此进行标记, 因为自该视图失效以来一直未使用。使用 <b>sa_materialized_view_info</b> 系统过程来确定实例化视图的状态。
mv_last_refreshed_tsn	UNSIGNED BIGINT	分配给刷新实例化视图的事务的序列号。
mv_last_refreshed_at_utc	TIMESTAMP WITH TIME ZONE	表示上次刷新实例化视图的 UTC 日期和时间。



列名	数据类型	描述
mv_known_stale_at_utc	TIMESTAMP WITH TIME ZONE	实例化视图失效的 UTC 时间。该值对应于检测到一个基表发生更改的时间。值为 0 表示该视图为最新的，或已失效但数据库服务器尚未对此进行标记，因为自该视图失效以来一直未使用。使用 sa_materialized_view_info 系统过程来确定实例化视图的状态。当 mv_last_refreshed_at 为 0 时，该列包含 0，而当 mv_last_refreshed_at 为 NULL 时，该列包含 NULL。

### 基础系统表上的约束

```
PRIMARY KEY (view_object_id)
```

```
FOREIGN KEY (view_object_id) references SYS.ISYSOBJECT (object_id)
MATCH UNIQUE FULL
```

### **SYSVIEWS** 统一视图

**SYSVIEWS** 视图的每一行都描述一个视图，包括其视图定义。

在以下的 SQL 语句中提供了组成该视图的表和列。要了解有关特定表或列的详细信息，请使用该视图定义下所提供的链接。

```
ALTER VIEW "SYS"."SYSVIEWS"( vcreator,
    viewname,viewtext )
as select u.user_name,t.table_name,v.view_def
    from SYS.ISYSTAB as t
        join SYS.ISYSVIEW as v on(t.object_id = v.view_object_id)
        join SYS.ISYSUSER as u on(u.user_id = t.creator)
```

### **SYSWEBSERVICE** 系统视图

**SYSWEBSERVICE** 系统视图的每一行都包含对一种 Web 服务的描述。该视图的基础系统表为 **ISYSWEBSERVICE**。

列名	数据类型	说明
service_id	UNSIGNED INT	Web 服务的唯一标识号。
object_id	UNSIGNED BIGINT	Web 服务的 ID。
service_name	CHAR(128)	分配给 Web 服务的名称。
service_type	VARCHAR(40)	服务类型，例如 RAW、HTTP、XML、SOAP 或 DISH。
auth_required	CHAR(1)	是否所有请求都必须包含有效用户名和口令。
secure_required	CHAR(1)	是否接受不安全的连接（例如 HTTP），或仅接受安全连接（例如 HTTPS）。

列名	数据类型	说明
url_path	CHAR(1)	控制对 URL 的解释。
user_id	UNSIGNED INT	如果启用验证，则标识有权使用该服务的用户或用户组。如果禁用验证，则指定处理请求时使用的帐户。
parameter	LONG VARCHAR	标识要包含在 DISH 服务中的 SOAP 服务的前缀。
statement	LONG VARCHAR	响应请求时始终要执行的 SQL 语句。如果为 NULL，则执行在每个请求中包含的任意语句。DISH 类型的服务将忽略此参数。
remarks	LONG VARCHAR	有关 Web 服务的注释。该值存储在 ISYSREMARK 系统表中。
enabled	CHAR(1)	指示 web 服务当前是启用还是禁用（请参见 CREATE SERVICE）。

### 基础系统表上的约束

```
PRIMARY KEY (service_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
UNIQUE CONSTRAINT (service_name)
```

### **Transact-SQL 兼容性视图**

Adaptive Server 和 SAP Sybase IQ 的系统目录不同，表示两种产品用途不同。

在 Adaptive Server 中，存在一个 master 数据库，其中包含一组保存有应用于服务器上所有数据库的信息的系统表。Master 数据库中可能存在许多数据库，而且每个数据库都有另外的系统表与之关联。

在 SAP Sybase IQ 中，每个数据库独立存在，并且包含自己的系统表。没有 master 数据库包含关于数据库集合的系统信息。每台服务器都可以同时运行多个数据库，可根据需要动态装载和卸载每个数据库。

Adaptive Server 和 SAP Sybase IQ 系统目录不同。Adaptive Server 系统表和视图由特殊用户 dbo 拥有，并且部分存在于 master 数据库中，部分存在于 **sybsecurity** 数据库中，部分存在于每个单独的数据库中；SAP Sybase IQ 系统表和视图由特殊用户 sys 拥有，并分别存在于每个数据库中。

为了便于准备兼容应用程序，SAP Sybase IQ 可提供一组由特殊用户 dbo 拥有的视图，这些视图对应于 Adaptive Server 系统表和视图。如果体系结构差异使得特定 Adaptive Server 表或视图的内容在 SAP Sybase IQ 环境中没有意义，则视图为空，仅包含列名和数据类型。

这些主题列出了 Adaptive Server 系统表及其在 SAP Sybase IQ 系统目录中的实现。每个 DBMS 中的 dbo 为所有表的所有者。

另请参见

- ASE T-SQL 兼容性视图（第 654 页）

每个 Adaptive Server 数据库中的表

不是所有的 Adaptive Server 系统表均在 SAP Sybase IQ 系统目录中实现。

表 127. 每个 Adaptive Server 数据库中的表

表名	描述	是否包含数据?	由 SAP Sybase IQ 支持?
sysalternates	每个映射到一个数据库用户的用户占一行	否	否
syscolumns	表或视图中的每一列都占一行，过程中的每个参数都占一行。在 SAP Sybase IQ 中，查询时使用所有者名称 dbo，即 dbo.syscolumns。	是	是
syscomments	每个视图、规则、缺省值和过程占一行或多行，以给出 SQL 定义语句。	是	是
sysconstraints	与表或列相关联的每个参照约束和检查约束在其中都有相应的一行。	否	否
sysdepends	由过程、视图引用的每个过程、视图或表占一行。	否	否
sysindexes	每个聚簇索引或非聚簇索引都占一行，每个没有索引的表都占一行，每个包含文本或图像数据的表另外占一行。在 SAP Sybase IQ 中，查询时使用所有者名称 dbo，即 dbo.sysindexes。	是	是
sysiqobjects	每个系统表、用户表、视图、过程、触发器、事件、约束、域 (sysdomain)、域 (sysusertype)、列和索引各显示一行。	是	是
sysiqvindex	每个非 fp iq 索引占一行。	是	是
syskeys	每个主键、外键或公用键各显示一行；由用户设置（不由 Adaptive Server 维护）。	否	否
syslogs	事务日志。	否	否
sysobjects	每个表、视图、过程、规则、缺省值、日志和临时对象（仅在 tempdb 中）占一行。	仅包含兼容数据	是
sysprocedures	每个视图、规则、缺省值和过程占一行，以给出内部定义。	否	否
sysprotects	用户权限信息。	否	否
sysreferences	在表或列上声明的每个参照完整性约束在其中都有相应的一行。	否	否

表名	描述	是否包含数据?	由 SAP Sybase IQ 支持?
sysroles	将服务器范围内的角色映射到本地数据库组。	否	否
syssegments	每个段（命名的磁盘区段集）在其中都有相应的一行。	否	否
systhresholds	为数据库定义的每个阈值在其中都有相应的一行。	否	否
systypes	每个系统提供的数据类型和用户定义数据类型占一行。	是	是
sysusers	数据库中允许的每个用户在其中都有相应的一行。	是	是

另请参见

- SYSCOLUMNS ASE 兼容性视图（第 665 页）
- SYSCOMMENTS ASE 兼容性视图（第 665 页）
- SYSINDEXES ASE 兼容性视图（第 681 页）
- SYSIQOBJECTS ASE 兼容性视图（第 691 页）
- SYSIQVINDEXT ASE 兼容性视图（第 695 页）
- SYSOBJECTS ASE 兼容性视图（第 702 页）
- SYSTYPES ASE 兼容性视图（第 741 页）
- SYSUSERS ASE 兼容性视图（第 747 页）

Adaptive Server Master 数据库中的表

不是所有的 Adaptive Server master 数据库表均在 SAP Sybase IQ 系统目录中实现。

**表 128. ASE master 数据库表**

表名	描述	是否包含数据?	由 SAP Sybase IQ 支持?
syscharsets	每个字符集或排序顺序占一行	否	否
sysconfigures	可以由用户设置的每个配置参数占一行	否	否
syscurconfigs	有关服务器当前正在使用的配置参数的信息	否	否
sysdatabases	服务器上的每个数据库占一行	否	否
sysdevices	每个磁带转储设备、磁盘转储设备、用于数据库的磁盘和用于数据库的磁盘分区在其中都有相应的一行	否	否
sysengines	每个当前联机的服务器占一行	否	否

表名	描述	是否包含数据?	由 SAP Sybase IQ 支持?
syslanguages	服务器已知的每种语言（美国英语除外）占一行	否	否
syslocks	有关活动锁的信息	否	否
sysloginroles	每个拥有系统定义角色的服务器登录占一行	否	否
syslogins	每个有效用户帐户占一行	是	是
sysmessages	每个系统错误或警告占一行	否	否
sysprocesses	有关服务器进程的信息	否	否
sysremotelogins	每个远程用户占一行	否	否
sysrvroles	每个服务器范围内的角色占一行	否	否
sys.servers	每个远程服务器占一行	否	否
sysusages	每个分配给数据库的磁盘区段占一行	否	否

#### Adaptive Server Subsecurity 数据库中的表

不是所有的 Adaptive Server subsecurity 数据库表均在 SAP Sybase IQ 系统目录中实现。

**表 129. ASE subsecurity 数据库表**

表名	描述	是否包含数据?	由 SAP Sybase IQ 支持?
sysaudits	每个审计记录占一行	否	否
sysauditoptions	每个全局审计选项占一行	否	否



## 与其它 Sybase 数据库的兼容性

该部分的主题用于简化从其它 SAP Sybase 数据库到 SAP Sybase IQ 的迁移，以及用于充当创建与 Adaptive Server 或 SQL Anywhere 兼容的 SAP Sybase IQ 应用程序的指南。

SAP Sybase IQ 的每个新版本中都涉及兼容性特性。该部分将 SAP Sybase IQ 与 Adaptive Server 和 SQL Anywhere 进行比较。

### SQL Anywhere 的说明

---

SAP Sybase IQ 是 SQL Anywhere 的扩展。

大多数情况下，SQL 语法、函数、选项、实用程序、过程和其它功能对于这两种产品而言都是相同的。但是，仍有一些重要区别。不要认为 SQL Anywhere 文档中描述的所有特性在 SAP Sybase IQ 中也受支持。使用 SAP Sybase IQ 文档。

### Transact-SQL 支持概述

---

SAP Sybase IQ，如 SQL Anywhere，支持 *Transact-SQL*（Sybase Adaptive Server 所支持的 SQL 方言）中相当大的一部分。

在 SAP Sybase IQ 中，Transact-SQL 支持的目标是提供应用程序可移植性。许多应用程序、存储过程和批处理文件都可以写入，以便用于 Adaptive Server 和 SAP Sybase IQ 数据库。

旨在写入应用程序以便使用 Adaptive Server 和 SAP Sybase IQ。通常情况下，现有的 Adaptive Server 应用程序需要进行一些更改才能在 SQL Anywhere 或 SAP Sybase IQ 数据库上运行。

SAP Sybase IQ 通过以下形式支持 Transact-SQL：

- 大部分 SQL 语句在 SAP Sybase IQ 和 Adaptive Server 之间均兼容。
- 对于某些语句（尤其是在过程和批处理语句中使用的过程语言），将支持单独的 Transact-SQL 语句以及 SAP Sybase IQ 先前版本中所支持的语法。对于这些语句，SQL Anywhere 和 SAP Sybase IQ 支持两种 SQL 方言。在本附录中，我们将这两种方言称为 Transact-SQL 和 Watcom-SQL。
- 过程或批处理以 Transact-SQL 或 Watcom-SQL 术语执行。在整个批处理语句或过程中，必须仅使用一种术语的控制语句。例如，每种方言都有不同的流控制语句。

在处理现有数据时，SAP Sybase IQ 支持大多数 Transact-SQL 语言元素、函数和语句。

另外，SAP Sybase IQ 还支持大多数 Transact-SQL 存储过程语言（**CREATE PROCEDURE** 语法、控制语句等），以及 Transact-SQL 数据定义语言语句的很多（但不是所有）方面。

每种产品所支持的体系结构和配置功能存在着设计上的差异。设备管理、用户管理和维护任务（如备份）往往是因系统而异的。但是，即使在这种情况下，SAP Sybase IQ 仍将 Transact-SQL 系统表作为视图来提供（在视图中，SAP Sybase IQ 中没有任何意义的表将不显示任何行）。此外，SAP Sybase IQ 为某些更常见的管理任务提供了一组系统过程。

## Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 体系结构

---

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 是互补产品，其体系结构用于满足不同的目的。

SAP Sybase IQ 是专为数据仓库应用和分析处理设计的高性能决策支持服务器。SQL Anywhere 适合在工作组或部门服务器所需管理较少时作为个人数据库使用。Adaptive Server 适合用作大型数据库的企业级服务器，它侧重于事务处理。

本节描述了这三种产品间的体系结构差异。还描述 SAP Sybase IQ 和 SQL Anywhere 包括的用于兼容数据库管理的类似 Adaptive Server 的工具。

### 服务器和数据库

Adaptive Server 中的服务器和数据库之间的关系不同于 SAP Sybase IQ 和 SQL Anywhere 中服务器和数据库之间的关系。

在 Adaptive Server 中，每个数据库都存在于服务器中，而每个服务器都可以包含多个数据库。用户可以具有服务器的登录权限并且可以连接到服务器。然后，如果用户拥有相应的权限，则可以连接到该服务器上的任何数据库。全系统范围的系统表保存在 master 数据库中，这些表包含服务器上所有数据库共有的信息。

在 SAP Sybase IQ 中，不存在等效 Adaptive Server master 数据库。相反，每个数据库都是一个独立的实体，它们包含所有系统表。用户可以具有数据库（而不是服务器）的连接权限。当用户建立连接后，他或她将连接到单独的数据库。没有在 master 数据库级别维护的全系统范围的系统表集。每台 SAP Sybase IQ 数据库服务器都可以动态地启动和停止用户可维护独立连接的数据库。SAP 强烈建议每台服务器只运行一个 SAP Sybase IQ 数据库。

SQL Anywhere 和 SAP Sybase IQ 为支持 Transact-SQL 和 Open Server 提供了一些工具，允许以类似于 Adaptive Server 的方式执行某些任务。但是，这些工具的实际实现方式存在一些差异。



## 空间分配和设备管理

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 最初及后来使用不同的模型来管理设备和分配磁盘空间，反映了这些产品的不同用法。

例如：

- 在 Adaptive Server 中，最初使用 **DISK INIT** 分配数据库设备中的空间，然后在一个或多个数据库设备上创建数据库。您可以使用 **ALTER DATABASE** 或使用阈值自动添加更多空间。
- 在 SAP Sybase IQ 中，最初通过在 **CREATE DATABASE** 语句中列出原始设备来分配空间。可以使用 **CREATE DBSPACE** 手动添加更多空间。虽然您不能自动添加空间，但是可以创建事件以在实际需要空间之前提醒 DBA。SAP Sybase IQ 还可以使用文件系统空间。SAP Sybase IQ 不支持 Transact-SQL **DISK** 语句，如 **DISK INIT**、**DISK MIRROR**、**DISK REFIT**、**DISK REINIT**、**DISK REMIRROR** 和 **DISK UNMIRROR**。
- SQL Anywhere 和 SAP Sybase IQ 类似，只不过 **CREATE DATABASE** 语句会占用文件系统文件而不是原始设备列表。SQL Anywhere 允许您使用名为 **dbinit** 的命令实用程序对数据库进行初始化。SAP Sybase IQ 可提供此实用程序的扩展版本（名为 **iqinit**）来初始化 SAP Sybase IQ 数据库。

## 系统表、目录存储库和 IQ 主存储库

SAP Sybase IQ 数据库是连接数据存储库。

连接存储库由以下内容构成：

- 目录存储库包括系统表和存储过程，并位于与 SQL Anywhere 兼容的表集中。
- 永久性 IQ 主存储库是 SAP Sybase IQ 表集。表数据存储在索引中。
- 临时存储包括一组临时表，数据库服务器使用这些临时表进行排序以及其它临时处理。

目录区别和兼容性功能包括：

- SQL Anywhere 和 SAP Sybase IQ 针对目录（表、列等）使用不同的 Adaptive Server 模式。
- SQL Anywhere 和 SAP Sybase IQ 将提供模拟 Adaptive Server 系统表相关部分的兼容性视图，但是使用时会有一些性能影响。
- 在 Adaptive Server 中，数据库所有者（用户 ID **dbo**）拥有目录对象。
- 在 SQL Anywhere 和 SAP Sybase IQ 中，系统所有者（用户 ID **sys**）拥有目录对象。

**注意：** **dbo** 用户 ID 拥有由 SAP Sybase IQ 提供的 Adaptive Server 兼容系统视图。

## 数据类型

---

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 以不同的方式处理数据类型。

本节讨论了各种数据类型的兼容性信息。

---

**注意：** 本节中未包括的数据类型当前受全部三种产品支持。

---

### 位数据类型

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对 BIT 数据类型的支持有所不同。区别如下：

- SQL Anywhere 只允许 0 或 1。
- Adaptive Server 和 SAP Sybase IQ 隐式将整数数据类型转换为 BIT。非零值存储为 1 (TRUE)。

### 字符数据类型兼容性

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 允许 CHAR 和 VARCHAR 数据，但是各个产品处理这些类型的方式不同。

- SAP Sybase IQ 将所有字符串视为 VARCHAR，即使在以空格填充的数据库中也是如此。
- Adaptive Server 和 SAP Sybase IQ 区分 CHAR（固定长度）数据和 VARCHAR（可变长度）数据。

Adaptive Server 剪裁 VARCHAR 值的尾随空白。SAP Sybase IQ 根据数据和操作形式剪裁 VARCHAR 值的尾随空白。

插入 CHAR 或 VARCHAR 时：

- SQL Anywhere 允许将整数数据类型插入 CHAR 或 VARCHAR（隐式转换）中。
- Adaptive Server 和 SAP Sybase IQ 需要显式转换。

按如下方式确定列的最大大小：

- Adaptive Server CHAR 和 VARCHAR 取决于逻辑页大小，而逻辑页大小可以是 2K、4K、8K 和 16K。例如：
  - 2K 页大小允许列为单个行那么大，大约是 1962 字节。
  - 4K 页大小允许的列大小约为 4010 字节。
- SQL Anywhere 支持最大为 32K-1 的 CHAR 和 VARCHAR，以及最大为 2GB 的 LONG VARCHAR。
- SQL Anywhere 支持名称 LONG VARCHAR 以及其同义词 TEXT，而 Adaptive Server 仅支持名称 TEXT，不支持名称 LONG VARCHAR。

- SAP Sybase IQ 支持最多为 32K-1 个字节的 CHAR 和 VARCHAR。  
SAP Sybase IQ 还支持最多 512TB（SAP Sybase IQ 页大小为 128KB）和 2PB（SAP Sybase IQ 页大小为 512KB）的 LONG VARCHAR。有关 SAP Sybase IQ 中 LONG VARCHAR 数据类型的信息，请参见《非结构化数据分析》。
- Adaptive Server 支持 NCHAR、NVARCHAR、UNICHAR、UNIVARCHAR 数据类型。N 代表多字节字符集；UNI 代表单字节字符集。
- SQL Anywhere 和 SAP Sybase IQ 支持 CHAR 和 VARCHAR 数据类型的 Unicode，而不支持单独数据类型的 Unicode。
- 为使 SAP Sybase IQ 和 Adaptive Server 兼容，请始终指定字符数据类型的长度。

#### 另请参见

- 字符数据类型（第 345 页）

## 二进制数据类型

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对二进制数据类型的支持有所不同。

表 130. 二进制数据类型支持的大小

数据类型	Adaptive Server	SQL Anywhere	SAP Sybase IQ
BINARY	< 页大小	32KB - 1	255
VARBINARY	< 页大小	32KB - 1	32KB - 1
LONG BINARY*	不受支持	2GB - 1	512TB (IQ 页大小为 128KB) 2PB (IQ 页大小为 512KB)
IMAGE	2GB	2GB - 1	使用 LONG BINARY*

\*有关 SAP Sybase IQ 中 LONG BINARY 数据类型的信息，请参见《非结构化数据分析》。此功能需要单独的许可证。

投影时，Adaptive Server 和 SAP Sybase IQ 会以不同方式显示二进制数据：

- SAP Sybase IQ 支持 Adaptive Server 和 SQL Anywhere 显式格式。
- 如果在 BINARY 字段中输入 '123'，则 SQL Anywhere 会以字节格式显示，如 '123'；Adaptive Server 会以半字节格式显示，如 '0x616263'。

#### 另请参见

- 二进制数据类型（第 352 页）
- NEWID 函数 [杂项]（第 247 页）
- STRTOUUID 函数 [字符串]（第 317 页）
- UUIDTOSTR 函数 [字符串]（第 331 页）
- 字符数据类型（第 345 页）

## 日期、时间、日期时间和时间戳数据类型

尽管 Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 全支持某些形式的日期和时间数据，但是仍存在一些差异。

- SQL Anywhere 和 SAP Sybase IQ 支持 4 字节的日期和时间数据类型。
- Adaptive Server 支持 8 字节的 `datetime` 类型以及以二进制形式 (8) 实现的作为用户定义数据类型 (域) 的 `timestamp`。
- SQL Anywhere 和 SAP Sybase IQ 支持 8 字节 `timestamp` 类型以及以时间戳形式实现的 8 字节 `datetime` 域。SQL Anywhere/SAP Sybase IQ 的毫秒精度 `datetime` 数据类型不同于 Adaptive Server。

日期的显示格式具有不同的缺省值：

- Adaptive Server 缺省以 "MMM-DD-YYYY" 格式显示日期，但可以通过设置选项来进行更改。
- SQL Anywhere 和 SAP Sybase IQ 缺省为 ISO "YYYY-MM-DD" 格式，但可以通过设置选项来进行更改。

时间转换如下：

- Adaptive Server 会根据秒的小数部分是由冒号还是句点分隔来决定将存储在字符串中的时间转换为内部时间的方式。
- 无论分隔符是什么，SQL Anywhere 和 SAP Sybase IQ 都会以相同的方式转换时间。

当您向 `DATETIME` 列插入时间时：

- Adaptive Server 和 SAP Sybase IQ 缺省提供 "1st January 1900"。
- SQL Anywhere 缺省提供当前日期。

使用 `INSERT...LOCATION` 插入到含 `DATETIME` 列的 SAP Sybase IQ 表时，根据 Adaptive Server 数据库的更改，检索 `TIME` 和 `DATETIME` 值。`INSERT...LOCATION` 语句使用 Open Client，Open Client 的 `DATETIME` 精度为 1/300 秒。

例如，假设以下值存储在 Adaptive Server 数据库的表列中：

```
2004-11-08 10:37:22.823
```

当使用 `INSERT...LOCATION` 在 SAP Sybase IQ 表中检索并存储该值时，它会变为：

```
2004-11-08 10:37:22.823333
```

### **Adaptive Server 中的日期时间值和时间值的兼容性**

使用 `INSERT...LOCATION` 在 Adaptive Server 中检索的 `DATETIME` 或 `TIME` 值基于 Open Client 的日期时间精度可能有所不同。

例如 Adaptive Server 数据库中的 `DATETIME` 值为 '2012-11-08 10:37:22.823'。当使用 `INSERT...LOCATION` 在 SAP Sybase IQ 中检索并存储该值时，它会变为 '2012-11-08 10:37:22.823333'。

## **BIGTIME 和 BIGDATETIME 支持**

对于组件集成服务 (CIS) 和 **INSERT...LOCATION**, SAP Sybase IQ 支持 Adaptive Server 数据类型 BIGTIME 和 BIGDATETIME。

- 具有 Adaptive Server 的组件集成服务 - aseodbc 服务器类代理表映射到包含数据类型为 BIGTIME 和 BIGDATETIME 的列的 Adaptive Server 表。

创建映射到 Adaptive Server 表的代理表时, 如果未指定映射, 则缺省情况下 BIGDATETIME 列会映射到 TIMESTAMP 列。缺省情况下, BIGTIME 列将映射到 TIME 列。

Asejdbc 服务器类不支持数据类型 BIGTIME 和 BIGDATETIME。

- **INSERT...LOCATION - INSERT...LOCATION** 命令用于将数据从包含数据类型为 BIGTIME 和 BIGDATETIME 的列的 Adaptive Server 表装载到 SAP Sybase IQ 表。SAP Sybase IQ 将 Adaptive Server 数据类型 BIGTIME 插入到 SAP Sybase IQ 数据类型 TIME。

SAP Sybase IQ 将 Adaptive Server 数据类型 BIGDATETIME 插入到 SAP Sybase IQ 数据类型 DATETIME、DATE、TIME 和 TIMESTAMP。

## **数值数据类型**

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 的缺省精度或小数位数不同。

- 在 Adaptive Server 中, 缺省精度为 18, 小数位数为 0。
- 在 SQL Anywhere 中, 缺省精度为 30, 小数位数为 6。
- 在 SAP Sybase IQ 中, 缺省精度为 126, 小数位数为 38。因为这些缺省值对于 TDS 和某些客户端工具来说太大, 通常要为 SAP Sybase IQ 精确数值类型指定精度和小数位数。

## **文本数据类型**

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对 TEXT 数据的支持不同。

- Adaptive Server 最多支持 2GB 的 LONG VARBINARY (SQL Anywhere 中为 LONG BINARY) 和 TEXT。SQL Anywhere 不支持 LONG VARBINARY 的列类型, 而是使用 LONG BINARY 达到相同目的。SQL Anywhere 最多支持 2GB 的 LONG BINARY 和 TEXT。
- SAP Sybase IQ 支持最多 32KB - 1 的 VARCHAR。SAP Sybase IQ 还支持最多 512TB (IQ 页大小为 128KB) 和 2PB (IQ 页大小为 512KB) 的 LONG VARCHAR 数据。有关 SAP Sybase IQ 中 LONG VARCHAR 数据类型的信息, 请参见《非结构化数据分析》。

## 图像数据类型

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对 IMAGE 数据的支持有所不同。

- Adaptive Server 和 SQL Anywhere 支持最多 2GB 的 IMAGE。
- SAP Sybase IQ 支持最多 512TB (IQ 页大小为 128KB) 和 2PB (IQ 页大小为 512KB) 的 LONG BINARY 数据。有关 SAP Sybase IQ 中 LONG BINARY 数据类型的信息, 请参见《非结构化数据分析》。

## Java 数据类型

Adaptive Server 允许在数据库中使用 Java 数据类型。而 SQL Anywhere 和 SAP Sybase IQ 不允许。

## Data Definition Language (数据定义语言)

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 在数据库和数据库对象的创建方式上有所不同。

## 使用 CREATE DATABASE 语句创建与 Transact-SQL 兼容的数据库

使用 Interactive SQL 来创建与 Transact-SQL 兼容的数据库。

例如, 在 Interactive SQL 中键入以下语句:

```
CREATE DATABASE 'db-name.db' CASE RESPECT BLANK PADDING ON
```

## 区分大小写

数据库中的区分大小写指的是数据、标识符和口令要区分大小写。

### 数据的区分大小写

数据的区分大小写特性会反映在索引、查询结果等内容中。

创建数据库时, 应确定 SAP Sybase IQ 数据在比较时是否区分大小写。缺省情况下, 虽然数据始终保持输入时的大小写, 但 SAP Sybase IQ 数据库在比较时区分大小写。

Adaptive Server 是否区分大小写取决于 Adaptive Server 系统上安装的排序顺序。通过重新配置 Adaptive Server 的排序顺序, 可更改单字节字符集的区分大小写特性。

### 标识符的区分大小写

标识符包括表名、列名、用户 ID 等。

SAP Sybase IQ 不支持区分大小写标识符。输入在 Adaptive Server 中, 标识符的区分大小写特性遵循数据的区分大小写特性。

在 Adaptive Server 中, 用户定义数据类型的名称区分大小写。在 SAP Sybase IQ 中, 它们区分大小写。

## 用户 ID 和口令区分大小写

口令的区分大小写的处理方式与其它标识符有所不同。

在 SAP Sybase IQ 和 SAP Sybase SQL Anywhere® 中，新建数据库中的所有口令均区分大小写，与数据库是否区分大小写无关。缺省用户 ID 为 DBA，该用户的口令为小写的 *sql*。

重建现有数据库时，SAP Sybase IQ 和 SQL Anywhere 按以下条件确定口令的大小写：

- 如果数据库最初是在不区分大小写的数据库中输入的，则口令会依然不区分大小写。
- 如果口令最初是在区分大小写的数据库中输入的，则大写和混合大小写的口令依然区分大小写。如果口令全部以小写输入，则口令不区分大小写。
- 对现有口令和新口令进行的更改均区分大小写。

在 SAP Adaptive Server® Enterprise 中，用户 ID 和口令的区分大小写特性遵循服务器的区分大小写特性。

## 确保兼容的对象名

每个数据库对象在特定的命名空间中都必须有唯一的名称。

在此命名空间之外，允许出现重复的名称。同 SQL Anywhere 与 SAP Sybase IQ 相比，Adaptive Server 中的某些数据库对象占用不同的命名空间。

### *表名唯一性*

数据库中具有以下表名唯一性要求：

- 对于 SAP Sybase IQ 和 SQL Anywhere，表名在给定所有者的数据库中必须唯一。例如，user1 和 user2 均可创建名为 employee 的表；通过完全限定名称，如 user1.employee 和 user2.employee，可以提供唯一性。
- 对于 Adaptive Server，表名在数据库中以及对于所有者都必须唯一。

### *索引名称唯一性*

表中具有索引名称唯一性要求。在所有这三种产品中，索引由创建时所在表的所有者拥有。索引名称在给定表中必须唯一，但任何两个表都可以具有同名的索引，即使对于同一所有者也是如此。例如，在所有这三种产品中，表 t1 和 t2 可以具有同名索引，无论它们是由相同用户还是不同用户拥有都是如此。

### *重命名索引和外键*

SAP Sybase IQ 允许您使用 **ALTER INDEX** 语句对显式创建的索引、索引的外键角色名以及外键进行重命名。SQL Anywhere 允许您使用 **ALTER INDEX** 语句对索引、外键角色名以及外键进行重命名。Adaptive Server 不允许您重命名这些对象。

## 使用 **CREATE TABLE** 语句时的注意事项

创建表以实现兼容性时，请注意以下兼容性注意事项：空值处理、检查约束、参照完整性、缺省值、标识列、计算列、临时表和表位置。

### *列中的 NULL*

为实现 NULL 的兼容处理：

- 如果列定义中没有声明 NOT NULL，则 SQL Anywhere 和 SAP Sybase IQ 假定列可以为空值。可通过将数据库选项 **ALLOW\_NULLS\_BY\_DEFAULT** 设置为与 Transact-SQL 兼容的设置 OFF 来更改此行为。
- SQL Anywhere 和 SAP Sybase IQ 假定 BIT 列不能为 NULL。
- 如果未声明 NULL，Adaptive Server 假定列不能为空值。

### *检查约束*

SAP Sybase IQ 强制对基表、全局临时表和局部临时表以及用户定义的数据类型执行检查约束。用户可以记录检查完整性约束违规，并指定在 **LOAD** 语句回退前可以发生的违规数。

SAP Sybase IQ 不允许创建它无法评估的检查约束，如由用户定义的函数、代理表或非 SAP Sybase IQ 表组成的检查约束。对于那些无法评估的约束，当在 **LOAD**、**INSERT** 或 **UPDATE** 语句中首次使用定义了相应检查约束的表时，系统便会检测到这些约束。SAP Sybase IQ 不允许使用包含以下内容的检查约束：

- 子查询
- 指定主机语言参数、SQL 参数或列作为数据值目标的表达式
- 集函数
- 对非确定性函数或修改数据的函数的调用

Adaptive Server 和 SQL Anywhere 强制执行 **CHECK** 约束。SQL Anywhere 允许在检查约束中使用子查询。

SAP Sybase IQ 支持允许将约束封装在数据类型定义中的用户定义数据类型。

### *参照完整性约束*

支持强制执行完整性的操作，如下所示：

- SQL Anywhere 支持所有 ANSI 操作：SET NULL、CASCADE、DEFAULT、RESTRICT。
- Adaptive Server 支持其中的两个操作：SET NULL、DEFAULT。

---

**注意：** 通过使用触发器而非参照完整性可在 Adaptive Server 中实现 CASCADE。

---

- SAP Sybase IQ 仅支持 RESTRICT 操作。
- SAP Sybase IQ 不支持 NOT NULL FOREIGN KEY。
- SAP Sybase IQ 具有列不能同时作为候选键和外键的限制。



### 列中的缺省值

缺省值支持有如下区别：

- Adaptive Server 和 SQL Anywhere 支持为列指定缺省值。
- 仅 SQL Anywhere 支持 DEFAULT UTC TIMESTAMP。
- SAP Sybase IQ 支持为列指定缺省值（特殊值 DEFAULT UTC TIMESTAMP 和 DEFAULT CURRENT UTC TIMESTAMP 除外）。SAP Sybase IQ 还会忽略 DEFAULT\_TIMESTAMP\_INCREMENT 数据库选项的设置。

### 标识列

标识列支持有如下区别：

- SAP Sybase IQ 支持 IDENTITY 或 DEFAULT AUTOINCREMENT 作为缺省值。SAP Sybase IQ 支持精度为任意值、小数位数为 0 的任何数值类型的标识列，且该列可以为 NULL。SAP Sybase IQ 标识列必须为正值，且受数据类型范围的限制。SAP Sybase IQ 支持每个表一个标识列，并要求将数据库选项 IDENTITY\_INSERT 设置为显式插入和更新的表名称。要删除含 IDENTITY 列的表，不能将 IDENTITY\_INSERT 设置为指向该表。添加标识列时，该表可以包含数据。使用 SELECT INTO 派生的表不具有标识/自动增量列。SAP Sybase IQ 视图不能包含 IDENTITY/DEFAULT AUTOINCREMENT 列。
- SQL Anywhere 支持 AUTOINCREMENT 缺省值。SQL Anywhere 支持小数位数和精度为任意允许值的任何数值类型的标识列。标识列值可以是正数、负数或零，受数据类型范围的限制。SQL Anywhere 支持每个表有任意数量的标识列，但不要求使用 identity\_insert 执行显式插入、删除和更新。添加标识列时，表必须为空。可将 SQL Anywhere 标识列改为非标识列，也可以将非标识列改为标识列。可在 [\[\[unresolved text-ref: SQL Anywhere\]\]](#) 视图中添加或删除 AUTOINCREMENT 列。
- Adaptive Server 支持每个表一个标识列。Adaptive Server 标识列只能是小数位数为 0、最大精度为 38 的数值数据类型。它们必须为正数，且受数据类型范围的限制，也不能为空值。Adaptive Server 要求使用 identity\_insert 执行显式插入和删除，但不要求使用该命令执行标识列的更新。添加标识列时，该表可以包含数据。Adaptive Server 用户不能显式设置为标识列选择的下一个值。Adaptive Server 视图不能包含 IDENTITY/AUTOINCREMENT 列。在某些条件下使用 SELECT INTO 时，如果要从中选择内容的表中包含标识/自动增量列，Adaptive Server 会允许结果表中也包含这些列。

### 计算列

计算列支持有如下区别：

- SQL Anywhere 支持能够创建索引的计算列。
- Adaptive Server 和 SAP Sybase IQ 不支持计算列。

### 临时表

您可以通过以下方法创建临时表：在 **CREATE TABLE** 语句中的表名前加上井号 (#) 但不指定所有者。这些临时表是 SAP Sybase IQ 声明的临时表，只能在当前连接中使用。

### 定位表

Adaptive Server 和 SAP Sybase IQ 中表的实际放置方式有所不同。SAP Sybase IQ 支持 *ON segment-name* 子句，但 *segment-name* 指的是 SAP Sybase IQ dbspace。

### sp\_iqstatus 过程的输出

```
Sybase IQ (TM) Copyright (c) 1992-2013 by SAP AG or an SAP affiliate
company. All rights reserved.
Version: 16.0.0.562/130821/P/Mainline/Sun_x64/OS 5.10/64bit/
2013-08-21 06:15:41
Time Now: 2013-08-21 06:27:14.150
Build Time: 2013-08-21 06:15:41
File Format: 23 on 03/18/1999
Server mode: IQ Server
Catalog Format: 2
Stored Procedure Revision: 1
Page Size: 65536/4096blks/16bpp
Number of Main DB Files: 2
Main Store Out Of Space: N
Number of Cache Dbspace Files: 5
Number of Shared Temp DB Files: 0
Shared Temp Store Out Of Space: N
Number of Local Temp DB Files: 1
Local Temp Store Out Of Space: N
DB Blocks: 1-25600 IQ_SYSTEM_MAIN
DB Blocks: 522208-547807 MainUser
Cache Dbspace Blocks: 1-5120 ssd_dev_1
Cache Dbspace Blocks: 522208-527327 ssd_dev_2
Cache Dbspace Blocks: 1044416-1049535 ssd_dev_3
Cache Dbspace Blocks: 1566624-1571743 ssd_dev_4
Cache Dbspace Blocks: 2088832-2093951 ssd_dev_5
Local Temp Blocks: 1-25600 IQ_SYSTEM_TEMP
Create Time: 2013-08-21 06:27:05.444
Update Time: 2013-08-21 06:27:14.035
Main IQ Buffers: 1588, 100Mb
Temporary IQ Buffers: 1588, 100Mb
Main IQ Blocks Used: 5250 of 38400, 13%=20Mb, Max Block#: 5313
Cache Dbspace IQ Blocks Used: 197 of 25600, 0%=0Mb, Max Block#: 0
Shared Temporary IQ Blocks Used: 0 of 0, 0%=0Mb, Max Block#: 0
Local Temporary IQ Blocks Used: 65 of 12800, 0%=0Mb, Max Block#: 0
Main Reserved Blocks Available: 12800 of 12800, 100%=50Mb
Shared Temporary Reserved Blocks Available: 0 of 0, 0%=0Mb
Local Temporary Reserved Blocks Available: 12800 of 12800, 100%=50Mb
IQ Dynamic Memory: Current: 292mb, Max: 308mb
Main IQ Buffers: Used: 18, Locked: 0
Temporary IQ Buffers: Used: 4, Locked: 0
Main IQ I/O: I: L459/P9 O: C21/D33/P22 D:1 C:100.0
```

```

Temporary IQ I/O: I: L320/P0 O: C54/D59/P8 D:50 C:100.0
Other Versions: 0 = 0Mb
Active Txn Versions: 0 = C:0Mb/D:0Mb
Last Full Backup ID: 0
Last Full Backup Time:
Last Backup ID: 0
Last Backup Type: None
Last Backup Time:
DB Updated: 0
Blocks in next ISF Backup: 0 Blocks: =0Mb
Blocks in next ISI Backup: 0 Blocks: =0Mb
IQ large memory space: 2048Mb
IQ large memory flexible percentage: 50
IQ large memory flexible used: 0Mb
IQ large memory inflexible percentage: 90
IQ large memory inflexible used: 0Mb
IQ large memory anti-starvation percentage: 50
DB File Encryption Status: OFF
RLV memory limit: 2048Mb
RLV memory used: 0Mb

```

## 使用 CREATE DEFAULT、CREATE RULE 和 CREATE DOMAIN 语句时的注意事项

SAP Sybase IQ 提供了合并规则的替代方式。

- Adaptive Server 支持使用 Create Default 和 Create Rule 语句来创建指定缺省值。
- SQL Anywhere 和 SAP Sybase IQ 支持使用 **CREATE DOMAIN** 语句来达到相同的目的。

## 使用 CREATE TRIGGER 语句时的注意事项

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对触发器的支持有所不同。

- SQL Anywhere 支持行级和语句级触发器。
- Adaptive Server 仅支持语句级触发器。
- SAP Sybase IQ 不支持触发器。

**注意：**触发器实际上是存储过程，在 **INSERT**、**UPDATE** 或 **DELETE** 之前或之后立即作为同一事务的一部分自动运行，可用于触发相关更改（例如，当雇员转到不同部门时，自动更新雇员的经理的姓名）。它还用于编写审计追踪，以标识哪些修改使数据库发生了何种更改以及更改时间。

## 使用 CREATE INDEX 语句时的注意事项

在 Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 中，CREATE INDEX 语法略微不同。

- Adaptive Server 和 SQL Anywhere 使用下列语法支持聚簇索引或非聚簇索引：

```

CREATE [UNIQUE] [CLUSTERED] INDEX name
ON table (column,...)
ON dbspace

```

Adaptive Server 还允许使用 **NONCLUSTERED** 关键字，但是对于两种产品，缺省值为 **NONCLUSTERED**。

- Adaptive Server **CREATE INDEX** 语句在 SQL Anywhere 中使用，因为 SQL Anywhere 允许使用（但会忽略）关键字 **FILLFACTOR**、**IGNORE\_DUP\_KEY**、**SORTED\_DATA**、**IGNORE\_DUP\_ROW** 和 **ALLOW\_DUP\_ROW**。
- SQL Anywhere **CREATE INDEX** 语法支持索引顾问使用 **VIRTUAL** 关键字，但不支持使用该关键字执行实际查询。
- SAP Sybase IQ 支持七种专用的索引类型：**LF**、**HG**、**HNG**、**DATE**、**TIME**、**DTTM** 以及 **WD**。SAP Sybase IQ 还支持对具有相同数据类型、精度和小数位数的两列间的关系使用 **CMP** 索引。SAP Sybase IQ 缺省创建 **HG** 索引，除非在以下 **CREATE INDEX** 语句中指定索引类型：

```
CREATE [UNIQUE] [type] INDEX name  
ON table (column,...)
```

## 用户、组/角色以及权限

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 在用户和角色/组的模型方面存在一些差异。

在 Adaptive Server 中，用户连接到服务器，每个用户需要该服务器的登录 ID 和口令以及他们想要访问的每个数据库的用户 ID。

SQL Anywhere 和 SAP Sybase IQ 用户不需要服务器登录 ID。所有的 SQL Anywhere 和 SAP Sybase IQ 用户将收到数据库的用户 ID 和口令。

### 用户角色/组

为了可以同时向多个用户授予权限，SQL Anywhere 和 SAP Sybase IQ 支持用户角色，而 Adaptive Server 支持用户组。尽管基本角色和组相同，但是仍存在一些行为差异：

- Adaptive Server 只允许每个用户为一个组的成员。
- SQL Anywhere 和 SAP Sybase IQ 允许用户成为多个角色的成员，还允许存在角色层次结构。

这三种产品都拥有一个 **public** 角色或组，用于定义缺省权限。每个用户都会自动成为 **public** 角色或组的成员。

### 数据库对象权限

在所有这三种产品中，用于授予对各个数据库对象的权限的 **GRANT** 和 **REVOKE** 语句非常相似。

- 这三种产品都允许在数据库表和视图上拥有 **SELECT**、**INSERT**、**DELETE**、**UPDATE** 和 **REFERENCES** 权限以及在数据库表的选定列拥有 **UPDATE** 权限。SQL Anywhere 和 SAP Sybase IQ 还允许在数据库表和视图上拥有 **LOAD** 和 **TRUNCATE** 权限。

例如，以下语句在所有这三种产品中都有效：

```
GRANT INSERT, DELETE
ON TITLES
TO MARY, SALES
```

此语句向用户 MARY 和 SALES 角色或组授予对 TITLES 表使用 **INSERT** 和 **DELETE** 语句的权限。

- 所有这三种产品都允许针对存储过程授予 **EXECUTE** 权限。
- **Adaptive Server** 还对附加项支持 **GRANT** 和 **REVOKE**：
  - 对象：表中的列、视图中的列以及存储过程
  - 用户能力：**CREATE DATABASE**、**CREATE DEFAULT**、**CREATE PROCEDURE**、**CREATE RULE**、**CREATE TABLE**、**CREATE VIEW**
- **SQL Anywhere** 和 **SAP Sybase IQ** 需要用户具有 **MANAGE ANY OBJECT PRIVILEGE** 系统特权来授予数据库对象权限。（紧密对应的 **Adaptive Server** 权限是 **GRANT ALL**，它由数据库所有者使用。）
- 这三种产品都支持 **WITH GRANT OPTION** 子句，该子句允许被授予权限的用户向其他用户授予权限，但是 **SAP Sybase IQ** 和 **SQL Anywhere** 不允许 **WITH GRANT OPTION** 用于 **GRANT EXECUTE** 语句。

#### 数据库范围权限

**Adaptive Server** 针对数据库范围的用户权限使用不同的模型。

- 假定 **SYS\_AUTH\_DBA\_ROLE** 兼容性角色尚未迁移到角色的层次结构中以满足客户安全需求，**SQL Anywhere** 和 **SAP Sybase IQ** 将使用 **SYS\_AUTH\_DBA\_ROLE** 兼容性角色以便允许用户在数据库内具有完全权限。
- **Adaptive Server** 中的系统管理员享有服务器上所有数据库的这一权限。
- 数据库所有者必须使用 **Adaptive Server SETUSER** 语句获取对其他用户所拥有的对象的权限。

#### 添加用户

**Adaptive Server** 添加用户需要两步：首先是 **sp\_addlogin**，接着是 **sp\_adduser**。

**SQL Anywhere** 和 **SAP Sybase IQ** 添加用户需要一步。

尽管不需要添加或删除用户，但是 **SAP Sybase IQ** 登录管理存储过程允许用户具有相应的系统特权来添加或删除 **SAP Sybase IQ** 用户帐户。启用 **SAP Sybase IQ** 用户管理时，这些 **SAP Sybase IQ** 用户帐户允许控制用户连接和口令到期。

尽管 **SQL Anywhere** 和 **SAP Sybase IQ** 允许 **Adaptive Server** 系统过程来管理用户和组，但是某些情况下这些过程的确切语法和函数有所不同。

#### 另请参见

- **Adaptive Server** 系统过程（第 645 页）

## 装载格式

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对装载格式的支持有所不同。

- SAP Sybase IQ 可处理 ASCII、BINARY 和 BCP 装载格式。
- 除 ASCII 和 BINARY 以外，SQL Anywhere 还允许您导入 dBase、Excel、FoxPro 和 Lotus 文件格式。
- SAP Sybase IQ 可处理 ASCII、BINARY 和 BCP 装载格式。

---

**注意：** SAP Sybase IQ 和 SQL Anywhere **LOAD** 语句的语法基于 BCP，旨在提供完全相同的功能。

---

## Transact-SQL 兼容性选项

使用 **SET OPTION** 语句设置 SAP Sybase IQ 数据库选项。

请参见参考：语句和选项中的 Transact-SQL 兼容性选项。

## Data Manipulation Language (数据操纵语言)

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 的查询需求不同。

### 编写可移植 SQL 的一般原则

即使多个服务器支持给定的 SQL 语句，每个系统上的缺省行为也可能不尽相同。

有关编写兼容的 SQL 语句的一般原则包括：

- 在编写将在多个数据库管理系统上使用的 SQL 语句时，应该使您的 SQL 语句尽可能地清晰。
- 详细地写出所有可用选项，而不是使用缺省行为。
- 使用括号来明确语句中的执行顺序，而不是假定运算符有相同的缺省优先级。
- 使用在变量名前加 @ 符号这一 Transact-SQL 约定，以实现 Adaptive Server 的可移植性。
- 在过程和批处理中声明变量和游标要紧跟在 **BEGIN** 语句之后。这是 SAP Sybase IQ 的要求，但是 Adaptive Server 允许在过程或批处理中的任何位置进行声明。
- 不要在数据库中使用 Adaptive Server 或 SAP Sybase IQ 中的任何保留字作为标识符。

### 编写兼容查询的标准

要编写在 SAP Sybase IQ 数据库和 Adaptive Server 数据库上都能运行的查询，应符合以下两个条件：

- 查询中的数据类型、表达式和搜索条件必须兼容。
- **SELECT** 语句本身的语法必须兼容。

SAP Sybase IQ 支持 Transact-SQL **SELECT** 语句的以下子集。

### 语法

```

SELECT [ ALL | DISTINCT ] select-list
...[ INTO #temporary-table-name ]
...[ FROM table-spec,
...   table-spec, ... ]
...[ WHERE search-condition ]
...[ GROUP
...     BY column-name, ... ]
...[ HAVING search-condition ]
...| [ ORDER
...     BY expression [ ASC | DESC ], ... ]      |
...| [ ORDER
...     BY integer [ ASC | DESC ], ... ]      |

```

### 参数

```

select-list:
{ table-name.* }...
{ * }...
{ expression }...
{ alias-name = expression }...
{ expression as identifier }...
{ expression as T_string }...

```

```

table-spec:
[ owner. ]table-name
...   [ [ AS ] correlation-name ]
...

```

```

alias-name:
identifier | 'string' | "string"

```

下面各节详细说明了编写兼容查询时需要注意的一些项。

### 另请参见

- Transact-SQL 过程中的变量（第 780 页）

## 子查询支持

SAP Sybase IQ 当前提供的子查询支持与 Adaptive Server 和 SQL Anywhere 提供的子查询支持略有不同。

Adaptive Server 和 SQL Anywhere 支持 **ON** 子句中的子查询；SAP Sybase IQ 当前不支持。

**UNION** 在子查询中的支持情况如下：

- SQL Anywhere 在相关和不相关子查询中均支持 **UNION**。
- SAP Sybase IQ 仅在不相关子查询中支持 **UNION**。

- Adaptive Server 在任何子查询中都不支持 **UNION**。

SQL Anywhere 支持许多其它位置的子查询，这些位置的标量值可能显示在语法中。关于可以在何处指定子查询，Adaptive Server 和 SAP Sybase IQ 遵循 ANSI 标准。

## **GROUP BY 子句支持**

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对 **GROUP BY ALL** 数据的支持不同。

- Adaptive Server 支持 **GROUP BY ALL**，返回所有可能组，包括 **WHERE** 子句和 **HAVING** 子句。对于所有集合，这些子句均具有 NULL 值。
- SQL Anywhere 不支持 **GROUP BY ALL** Transact-SQL 扩展。

按如下方式支持 **GROUP BY** 子句中的 **ROLLUP** 和 **CUBE**：

- SAP Sybase IQ 和 SQL Anywhere 支持在 **GROUP BY** 子句中使用 **ROLLUP** 和 **CUBE**。
- Adaptive Server 当前不支持 **ROLLUP** 和 **CUBE**。

Adaptive Server 支持在 **SELECT** 子句中对非组列进行投影。这称为按语义扩展的组，将返回一组值。SAP Sybase IQ 支持按语义扩展的组，而 SQL Anywhere 不支持按语义扩展的组。只有 SQL Anywhere 支持使用 List() 集合返回值列表。

## **COMPUTE 子句支持**

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对 **COMPUTE** 的支持不同。

- Adaptive Server 支持 Transact-SQL **COMPUTE** 子句。
- SQL Anywhere 和 SAP Sybase IQ 不支持 Transact-SQL **COMPUTE** 子句，因为该子句不在 ANSI 标准中，且大多数第三方前端工具均提供此功能。

## **WHERE 子句支持**

在对 Contains() 谓词的支持以及对 Like() 谓词中尾随空白的处理方面，Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 之间的 **WHERE** 子句有所不同。

- SAP Sybase IQ 支持使用 **Contains()** 谓词在字符数据内搜索单词（与 MS SQL Server 和 Verity 中的 Contains 类似）。SAP Sybase IQ 使用 WORD 索引和 TEXT 索引优化这些搜索（如果可能）。
- Adaptive Server 不支持 **Contains()**。

## **Transact-SQL 外连接支持**

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对外连接语法的支持有所不同。

- Adaptive Server 完全支持外连接的 \*= 和 =\* Transact-SQL 语法。
- SQL Anywhere 和 SAP Sybase IQ 支持 Transact-SQL 外连接，但拒绝一些具有潜在不明确性的复杂 Transact-SQL 外连接。



- SAP Sybase IQ 不支持链式（嵌套的）Transact-SQL 外连接。对于此类型的多个外连接，请使用 ANSI 语法。

---

**注意：** 不建议在 SQL Anywhere 和 SAP Sybase IQ 中使用 Transact-SQL 外连接语法。

---

有关 Transact-SQL 外连接的详细信息（包括 ANSI 语法替代项），请参见白皮书 Transact-SQL 外连接的语义和兼容性，您可从 SAP Community Network 获得该白皮书。尽管是针对 SQL Anywhere 编写的，文档中的信息也适用于 SAP Sybase IQ。

## ANSI 连接支持

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对 ANSI 连接语法的支持有所不同。

- SAP Sybase IQ 当前不支持 **ON** 子句中的子查询。
- Adaptive Server 和 SQL Anywhere 支持 **ON** 子句中的子查询。
- 使用 ANSI 连接语法的查询的 **FROM** 子句中的 **CONTAINS** 条件受支持，但可能具有不理想的性能。仅当需要每个 **CONTAINS** 子句中的“score”列时，才能将外连接用于 **FROM** 子句中的 **CONTAINS**。否则，应将 **CONTAINS** 移到 **ON** 条件或 **WHERE** 子句中。

FULL OUTER JOIN 支持情况如下：

- SQL Anywhere 和 SAP Sybase IQ 支持 **FULL OUTER JOIN**。
- Adaptive Server 不支持 **FULL OUTER JOIN**。

## 空值比较支持

Adaptive Server 拥有 Transact-SQL 扩展，允许使用谓词比较空值。

例如，{col} = Null 表示 {col} Is Null。

SQL Anywhere 和 SAP Sybase IQ 使用 ANSI 语义进行空值比较，除非将 ANSINULL 选项设置为 OFF，在这种情况下，该比较与 Adaptive Server 兼容。

---

**注意：** SQL Anywhere 8.0 及更高版本添加了对 TDS\_EMPTY\_STRING\_AS\_NULL 的支持，为将空字符串映射到空值提供了 Adaptive Server 兼容性。

---

## 零长度字符串支持

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对零长度字符串的处理方式有所不同。

- Adaptive Server 将零长度字符串作为空值处理。  
Adaptive Server 用户将空字符串存储为一个空格。
- SQL Anywhere 和 SAP Sybase IQ 遵循零长度字符串的 ANSI 语义，即零长度字符串是实数值而不是空值。

## **HOLDLOCK、SHARED 和 FOR BROWSE 支持**

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对 HOLDLOCK、SHARED 和 FOR BROWSE 语法的支持有所不同。

- Adaptive Server 支持 HOLDLOCK、SHARED 和 FOR BROWSE 语法。
- SQL Anywhere 支持 HOLDLOCK 而不支持 SHARED 或 FOR BROWSE。
- SAP Sybase IQ 不支持这些关键字。

## **SQL 函数支持**

SAP Sybase IQ 支持 SQL Anywhere 和 Adaptive Server 所支持的大多数函数，但是仍有一些区别。

- Adaptive Server 支持在 **PatIndex()** 中使用 **USING CHARACTERS | USING BYTES** 语法；而 SQL Anywhere 和 SAP Sybase IQ 不支持。
- Adaptive Server 支持 **Reverse()** 函数；而 SQL Anywhere 和 SAP Sybase IQ 不支持。
- Adaptive Server 支持 **Len()** 作为 **Length()** 的替代语法；SQL Anywhere 不支持该替代语法。
- Adaptive Server 支持 **Square()** 和 **Str\_Replace()** Microsoft 兼容性函数；SQL Anywhere 不支持。
- SAP Sybase IQ 支持 **Str\_Replace()**。
- Adaptive Server 和 SQL Anywhere 支持使用 **TSEQUAL()** 来比较两个时间戳以确定修改时间；而 SAP Sybase IQ 不支持 **TSEQUAL()**。（**TSEQUAL** 在 SAP Sybase IQ 表级版本控制模型中不相关。）
- SAP Sybase IQ 支持 **ROWID()**；而 Adaptive Server 和 SQL Anywhere 不支持。
- 除 Adaptive Server 用于数据类型转换的 **Convert()** 外，SQL Anywhere 和 SAP Sybase IQ 还支持 **Cast()**。

---

**注意：** **Cast()** 是符合 ANSI 标准的名称。

- SQL Anywhere 和 SAP Sybase IQ 支持将 **Lcase()** 和 **Ucase()** 作为 **Lower()** 和 **Upper()** 的同义词；而 Adaptive Server 不支持。
- SQL Anywhere 和 SAP Sybase IQ 支持 **Locate()** 字符串函数；而 Adaptive Server 不支持。
- SQL Anywhere 支持 **IsDate()** 和 **IsNumeric()** 函数以测试将字符串转换成相应数据类型的能力；而 Adaptive Server 不支持。SAP Sybase IQ 支持 **IsDate()**。您可以在 SAP Sybase IQ 中使用 **IsNumeric**，但需要考虑 CIS 函数补偿性能注意事项。
- SQL Anywhere 支持 **NEWID**、**STRTOUID** 和 **UUIDTOSTR** 函数；Adaptive Server 不支持。这些是 SAP Sybase IQ 中的本地函数，因此无需考虑 CIS 函数补偿性能注意事项。

---

**注意：** 某些 SQL 函数（包括 **SOUNDEX** 和 **DIFFERENCE** 字符串函数）以及日期函数在 SAP Sybase IQ 和 SQL Anywhere 中的操作方式有所不同。SAP Sybase IQ 数据库选项 **ASE\_FUNCTION\_BEHAVIOR** 指定一些 SAP Sybase IQ 数据类型转换函数（包括 **HEXTOINT** 和 **INTTOHEX**）的输出与 Adaptive Server 函数的输出一致。

---

## **OLAP 函数支持**

Adaptive Server 当前不支持 OLAP 函数。但 SAP Sybase IQ 和 SQL Anywhere 支持。

SAP Sybase IQ 当前支持以下 OLAP 函数：

- **Corr()**
- **Covar\_Pop()**
- **Covar\_Samp()**
- **Cume\_Dist**
- **Dense\_Rank()**
- **Exp\_Weighted\_Avg**
- **First\_Value**
- **Last\_Value**
- **Median**
- **Ntile()**
- **Percent\_Rank()**
- **Percentile\_Cont()**
- **Percentile\_Disc()**
- **Rank()**
- **Regr\_Avgx()**
- **Regr\_Avgy()**
- **Regr\_Intercept()**
- **Regr\_R2**
- **Regr\_Slope()**
- **Regr\_Sxx()**
- **Regr\_Sxy()**
- **Regr\_Syy()**
- **StdDev()**
- **Stddev\_Pop**
- **Stddev\_Samp**
- **Var\_Pop**
- **Var\_Samp**
- **Variance()**
- **Weighted\_Avg**

SQL Anywhere 支持所有的 SAP Sybase IQ OLAP 函数。

Adaptive Server 当前不支持 OLAP 函数。

CIS 函数补偿不支持 OLAP 函数。

---

**注意：**对 OLAP 函数的支持是 Sybase 产品发展的一个快速发展的领域。

---

## 系统函数支持

SAP Sybase IQ 和 SQL Anywhere 不支持某些 Adaptive Server 系统函数。

SQL Anywhere 和 SAP Sybase IQ 不支持这些 Adaptive Server 系统函数：

- **curunreservedpgs()** - 数据库空间上的可用页数。
- **data\_pgs()** - 表或索引使用的页数。
- **host\_id()** - 服务器进程的 UNIX pid。
- **hos\_name()** - 运行服务器的计算机的名称。
- **lct\_admin()** - 管理事务管理器的“最后机会阈值”。
- **reserved\_pgs()** - 分配给表或索引的页数。
- **rowcnt()** - 指定表中的行数。
- **valid\_name()** - 使用的名称（例如，表名）是否有效。
- **valid\_user()** - 如果该用户具有连接权限，则返回 TRUE。
- **ptn\_data\_pgs()** - 分区中的数据页数。
- **index\_colorder()** - 返回索引中的列顺序。

## 用户定义的函数支持

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对用户定义函数 (UDF) 的支持有所不同：

- SQL Anywhere 支持使用 SQL、Java 和 C 编写的 UDF。
- Adaptive Server 仅支持使用 Java 编写的 UDF。
- SAP Sybase IQ 支持使用外部 C/C++ 及 Java UDF 作为许可选项。SAP Sybase IQ 通过 CIS 查询分解来支持 Interactive SQL UDF，但是存在一些性能影响。

## 解释日期的算术表达式的区别

SQL Anywhere 和 SAP Sybase IQ 将日期的算术表达式解释为各种日期函数的速记注释。Adaptive Server 不会这样。

- “日期 +/- 整数”等效于 **Dateadd()**。
- “日期 - 日期”等效于 **Datediff()**。
- “日期 + 时间”会利用这两个项创建一个时间戳。

## SELECT INTO 语句支持

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 中的 SELECT INTO 语句所允许的表类型有所不同。

请考虑以下示例语句：

```
select into table1 from table2
```

- Adaptive Server 允许 *table1* 为永久表、临时表或代理表。Adaptive Server 还支持 **SELECT INTO EXISTING TABLE**。
- SQL Anywhere 和 SAP Sybase IQ 允许 *table1* 为永久表或临时表。只有当选择到 *table* 中并指定多个列时，系统才会创建永久表。**SELECT INTO** 无论指定多少列，*#table*（不指定所有者）始终创建临时表。**SELECT INTO** 单列表会选择到宿主变量中。

## 可更新的视图支持

Adaptive Server 和 SQL Anywhere 对视图的定义比 ANSI 更加自由，无需请求 **WITH CHECK** 选项即可进行更新。

SQL Anywhere 提供 **ANSI\_UPDATE\_CONSTRAINTS** 选项来控制是将更新限制为 SQL92 所支持的那些更新，还是采用一组更加自由的规则。

SAP Sybase IQ 仅允许对可以展平的单表视图执行 **UPDATE**。SAP Sybase IQ 不支持 **WITH CHECK**。

## 支持 UPDATE 和 DELETE 中的 FROM 子句

SAP Sybase IQ、Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 均支持在 **UPDATE** 和 **DELETE** 中对多个表执行 **FROM** 子句。

## Transact-SQL 过程语言概述

存储过程语言是在存储过程和批处理中使用的 SQL 的一部分。

除基于 SQL92 的 Watcom-SQL 方言外，SQL Anywhere 和 SAP Sybase IQ 还支持 Transact-SQL 存储过程语言中的很大一部分。

## Transact-SQL 存储过程概述

由于其基于 ISO/ANSI 标准草案，SQL Anywhere 和 SAP Sybase IQ 存储过程语言在许多方面不同于 Transact-SQL 方言。

它们的许多概念和功能都相似，但语法却不同。SQL Anywhere 和 SAP Sybase IQ 支持 Transact-SQL 通过提供方言间的自动转换来利用类似的概念。但是，过程必须仅使用两种方言中的一种来编写，而不能混用这两种方言。

SQL Anywhere 和 SAP Sybase IQ 对 Transact-SQL 存储过程的支持包括多个不同的方面，例如：

- 传递参数
- 返回结果集
- 返回状态信息
- 提供参数的缺省值
- 控制语句
- 错误处理

## **Transact-SQL 批处理概述**

在 Transact-SQL 中，批处理是一起提交并作为一个组依次执行的 SQL 语句组。

批处理可以存储在命令文件中。SQL Anywhere 和 SAP Sybase IQ 中的 ISQL 实用程序以及 Adaptive Server 中的 isql 实用程序为交互执行批处理提供相似的功能。

过程中使用的控制语句也可以在批处理中使用。SQL Anywhere 和 SAP Sybase IQ 支持在批处理中使用控制语句，并支持以类似于 Transact-SQL 的方式使用以 GO 语句终止的非分隔语句组来表示批处理结束。

对于命令文件中存储的批处理语句，SQL Anywhere 和 SAP Sybase IQ 支持在命令文件中使用参数。Adaptive Server 不支持使用参数。

## **过程和批处理中的 SQL 语句**

SAP Sybase IQ 所支持的某些 SQL 语句只在一种方言中使用，而不在另一种方言中使用。

您不能在一个过程或批处理语句中混用两种术语。这意味着：

- 可以在一个批处理或过程中同时包含仅限 Transact-SQL 的语句以及属于两种术语的语句。
- 可以在一个批处理或过程中同时包含 Adaptive Server 不支持的语句以及两种服务器都支持的语句。
- 不能在一个批处理或过程中同时包含仅限 Transact-SQL 的语句以及仅限 SAP Sybase IQ 的语句。

Transact-SQL 过程或批处理中可以使用未以分号分隔的 SQL 语句。有关各个语句的详细信息，请参见《参考：语句和选项》。

Transact-SQL 兼容性已得到改进；以前接受的不正确的 SQL 语法现在将失败并返回错误。

### **IF 语句中的表达式子查询**

Adaptive Server 和 SQL Anywhere 支持对变量和由表达式子查询返回的标量值进行比较。

例如：

```
create procedure testIf () begin declare var4 int; set var4 = 10; if var4 = (select MIN (a_il) from a) then set var4 = 100; end if; end;
```

### **CASE 语句支持**

SAP Sybase IQ 和 SQL Anywhere 在是否允许使用 CASE 语句方面有所不同。

Adaptive Server 中不支持 CASE 语句，该产品仅支持 CASE 表达式。

**另请参见**

- 表达式 (第 27 页)

**行级游标操作支持**

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 支持游标与 UPDATE 和 DELETE 配合使用。

请考虑以下示例：

```
UPDATE WHERE CURRENT OF {cursor}
```

```
DELETE WHERE CURRENT OF {cursor}
```

在 SAP Sybase IQ 中，只有敏感性游标、仅针对单个表的游标以及链式游标才能进行更新。禁止使用可更新持有游标。SAP Sybase IQ 中的可更新游标具有表锁。

**PRINT 命令支持**

Adaptive Server、SQL Anywhere 和 SAP Sybase IQ 对 PRINT 的支持有所不同。

PRINT 的效果取决于客户端：

- Adaptive Server **PRINT** 始终向客户端发送消息。
- 在 SQL Anywhere 和 SAP Sybase IQ 中，**PRINT** 向客户端发送消息以建立 Open Client 和 JDBC 连接。
- 依赖于 **PRINT** 的 Adaptive Server 存储过程使用 Interactive SQL 在 SAP Sybase IQ 中运行。

---

**注意：** Interactive SQL 使用不建议使用的 iAnywhere JDBC 驱动程序。

---

**存储过程的自动转换**

除了支持 Transact-SQL 替代语法之外，SQL Anywhere 和 SAP Sybase IQ 还提供了用于在 Watcom-SQL 方言和 Transact-SQL 方言之间进行语句转换的辅助工具。

返回有关 SQL 语句的信息并可实现 SQL 语句自动转换的函数包括：

**表 131. 实现自动转换的函数**

功能	描述
SQLDialect(statement)	返回 Watcom-SQL 或 Transact-SQL。
WatcomSQL(statement)	返回语句的 Watcom-SQL 语法。
TransactSQL(statement)	返回语句的 Transact-SQL 语法。

这些是函数，因此可以从 ISQL 使用 **SELECT** 语句对其进行访问。例如，以下语句返回值 Watcom-SQL：

```
SELECT SqlDialect('select * from Employees')
```

## Transact-SQL 过程的结果集

---

SQL Anywhere、SAP Sybase IQ 过程和 Transact-SQL 过程返回结果集的方式有所不同。

SQL Anywhere 和 SAP Sybase IQ 使用 **RESULT** 子句指定返回的结果集。

在 Transact-SQL 过程中，第一个查询的列名或别名将返回到调用环境中。

以下 Transact-SQL 过程说明了 Transact-SQL 存储过程如何返回结果集：

```
CREATE PROCEDURE showdept (@deptname varchar(30))
AS
    SELECT Employees.Surname, Employees.GivenName
    FROM Departments, Employees
    WHERE Departments.DepartmentName = @deptname
    AND Departments.DepartmentID = Employees.DepartmentID
```

以下为相应的 SQL Anywhere 或 SAP Sybase IQ 过程：

```
CREATE PROCEDURE showdept(in deptname varchar(30))
RESULT ( lastname char(20), firstname char(20))
BEGIN
    SELECT Employees.Surname, Employees.GivenName
    FROM Departments, Employees
    WHERE Departments.DepartmentName = deptname
    AND Departments.DepartmentID = Employee.DepartmentID
END
```

客户端工具向客户端提供多个结果的方式略有不同：

- **isql** 在一个流中显示所有结果。
- **Interactive SQL** 在单独的选项卡上显示每个结果集。您必须在“选项”菜单中启用此功能。将它设为永久更改，然后重新启动或重新连接到 **Interactive SQL**。

## Transact-SQL 过程中的变量

---

SQL Anywhere 和 SAP Sybase IQ 为过程中的变量赋值的方式不同于 Transact-SQL。

SQL Anywhere 和 SAP Sybase IQ 使用 **SET** 语句为过程中的变量赋值。

在 Transact-SQL 中，系统将使用含空表列表的 **SELECT** 语句来赋值。下面的简单过程说明了 Transact-SQL 语法的工作原理：

```
CREATE PROCEDURE multiply
    @mult1 int,
    @mult2 int,
    @result int output
AS
SELECT @result = @mult1 * @mult2
```



可以按如下方式来调用该过程：

```
CREATE VARIABLE @product int
go
EXECUTE multiply 5, 6, @product OUTPUT
go
```

当该过程执行后，变量 *@product* 的值为 30。

变量声明的顺序和保留有一些区别：

- 在 **Adaptive Server** 中，您可以在存储过程正文中的任何位置声明变量。变量的保留时间为该过程的持续时间。
- 在 **SQL Anywhere** 和 **SAP Sybase IQ** 中，您必须在复合语句的开头声明变量（即，紧跟在 **BEGIN...END** 对中的 **BEGIN** 之后）。变量的保留时间仅为该复合语句的持续时间。

### 另请参见

- 编写兼容查询的标准（第 770 页）

## Transact-SQL 过程中的错误处理

Watcom-SQL 术语中的缺省过程错误处理方式不同于 Transact-SQL 术语中的缺省过程错误处理方式。

缺省情况下，Watcom-SQL 术语过程在它们遇到错误时退出，并且将 **SQLSTATE** 值和 **SQLCODE** 值返回到调用环境。

您可以使用 **EXCEPTION** 语句在 Watcom-SQL 存储过程中构置显式错误处理功能，也可以使用 **ON EXCEPTION RESUME** 语句指示过程在遇到错误时继续执行下一条语句。

当 Transact-SQL 术语过程遇到错误时，将从下一个语句继续执行。全局变量 **@@error** 保存最近执行过的语句的错误状态。您可以在语句之后检查这一变量，以强制从过程中返回。例如，以下语句将导致在出错时退出：

```
IF @@error != 0 RETURN
```

当该过程执行完之后，返回值将表明该过程是成功还是失败。此返回状态是一个整数，并且可以按如下方式对其进行访问：

```
DECLARE @status INT
EXECUTE @status = proc_sample
IF @status = 0
    PRINT 'procedure succeeded'
ELSE
    PRINT 'procedure failed'
```

下表介绍了内置过程返回值及其含义：

表 132. 内置过程返回值

值	含义
0	过程执行时没有发生错误
-1	缺失对象
-2	数据类型错误
-3	进程被选作死锁牺牲品
-4	权限错误
-5	语法错误
-6	杂类用户错误
-7	资源错误，如空间不足
-8	非致命内部问题
-9	达到系统限制
-10	致命内部不一致性
-11	致命内部不一致性
-12	表或索引损坏
-13	数据库损坏
-14	硬件错误

**RETURN** 语句可用于返回其它整数，这些整数由用户定义其含义。

### 在过程中使用 **RAISERROR** 语句

**RAISERROR** 语句是用于生成用户定义的错误的 Transact-SQL 语句。其功能与 **SIGNAL** 语句的功能类似。

**RAISERROR** 语句本身并不会导致从过程中退出，但它可以同 **RETURN** 语句或 **@@error** 全局变量的测试组合在一起控制用户定义的错误之后的执行过程。

如果您将 **ON\_TSQL\_ERROR** 数据库选项设置为 **CONTINUE**，**RAISERROR** 将不再发出有关执行结束错误的信号。相反，该过程会完成并存储 **RAISERROR** 状态码和消息，然后返回最近的 **RAISERROR**。如果导致 **RAISERROR** 的过程是从其它过程中调用的，**RAISERROR** 将在最外部的调用过程终止后返回。

当该过程终止后，您将丢失中间的 **RAISERROR** 状态和代码。如果错误在返回时随 **RAISERROR** 一起发生，则系统将返回错误信息，并且 **RAISERROR** 信息会丢失。应用程序可以通过在不同的执行点检查 **@@error** 全局变量来查询中间 **RAISERROR** 状态。

## Watcom-SQL 术语中与 Transact-SQL 类似的错误处理

您可以使 Watcom-SQL 术语过程以类似于 Transact-SQL 的方式处理错误。

向 **CREATE PROCEDURE** 语句提供 **ON EXCEPTION RESUME** 子句：

```
CREATE PROCEDURE sample_proc()
ON EXCEPTION RESUME
BEGIN
    ...
END
```

**ON EXCEPTION RESUME** 子句的存在将防止执行显式的异常处理代码，因此应该避免同时使用这两个子句。

## SQL Anywhere 和 SAP Sybase IQ 的区别和共享功能

SAP Sybase IQ 和 SQL Anywhere 在启动和管理数据库和服务器、数据库选项支持、DDL 支持以及 DML 支持方面存在一些区别。

有关其它信息，请在使用产品时始终参考 SAP Sybase IQ 文档集。请在使用 SQL Anywhere 时参考 SQL Anywhere 文档集，SAP Sybase IQ 文档仅在涉及特定功能时才引用 SQL Anywhere 文档

## SQL Anywhere 服务器和数据库启动与管理

SAP Sybase IQ 和 SQL Anywhere 启动和管理数据库与服务器的方式有所不同。

- SAP Sybase IQ 使用服务器启动命令 **start\_iq**，而不是 SQL Anywhere 网络服务器启动命令。
- SAP Sybase IQ 不支持个人服务器。
- SAP Sybase IQ 支持很多 SQL Anywhere 服务器命令行选项，但并不是所有命令行选项。其它服务器选项在 SAP Sybase IQ 中受支持，在 SQL Anywhere 中不支持。
- SAP Sybase IQ 提供 **stop\_iq** 实用程序 (UNIX) 以关闭服务器。
- SAP Sybase IQ 和 SQL Anywhere 在是否允许在 **BACKUP DATABASE** 和 **RESTORE DATABASE** 语句中使用子句方面有所不同。
- SAP Sybase IQ 中仅支持使用 SQL Remote 执行 Multiplex 操作。

SAP Sybase IQ 支持很多 SQL Anywhere 数据库管理实用程序，但并非所有管理实用程序：

- SAP Sybase IQ 不支持以下 SQL Anywhere 实用程序：
  - **backup**
  - **compression**
  - **console**
  - **initialization**

- **license**
- **log transfer**
- **log translation**
- **rebuild**
- **spawn**
- 一些 **transaction log** 选项 (**-g**、**-il**、**-ir**、**-n**、**-x**、**-z**)
- **uncompression**
- **unload**
- **upgrade**
- **write file**
- SAP Sybase IQ 仅支持对目录存储库使用 SQL Anywhere **validation** 实用程序。要验证 IQ 主存储库，请使用 **sp\_iqcheckdb**。

## SQL Anywhere 数据定义语言 (DDL) 区别

SQL Anywhere 和 SAP Sybase IQ 在 DDL 行为方面有所不同。

- 在 **ALTER TABLE** 语句的 **DELETE/DROP** 或 **PRIMARY KEY** 子句中，SAP Sybase IQ 会执行 **RESTRICT** 操作（如果存在关联外键，则报告错误）。SQL Anywhere 始终执行 **CASCADE** 操作。
- 类似地，如果存在关联的外键约束，**DROP TABLE** 会在 SAP Sybase IQ 中报告错误。
- SAP Sybase IQ 不支持这些 DDL 语句：**CREATE COMPRESSED DATABASE**、**CREATE TRIGGER**、**SETUSER**。
- SAP Sybase IQ 支持语句级参照完整性，而不是 SQL Anywhere 使用 **CREATE TABLE** 语句的 **CHECK ON COMMIT** 子句支持的事务级完整性。
- SAP Sybase IQ 表不能有引用 SQL Anywhere（或目录）表的外键，而 SQL Anywhere 表不能有引用 SAP Sybase IQ 表的外键。
- 在 SAP Sybase IQ 数据库中，只能在 SQL Anywhere 表中创建发布。
- 在 **CREATE DATABASE** 中，区分大小写和归类的缺省值不同。在 SAP Sybase IQ 中，缺省值为 **CASE RESPECT** 和 **ISO\_BINENG** 归类；在 SQL Anywhere 中，缺省值为 **CASE IGNORE** 以及从操作系统的语言和字符集推导出的归类。
- SAP Sybase IQ 不支持 SQL Anywhere 所支持的 **CREATE ENCRYPTED DATABASE** 和 **CREATE DECRYPTED DATABASE** 命令。请参见管理：用户管理和安全。

## SQL Anywhere 数据操作语言 (DDL) 区别

并不是所有的 SQL Anywhere DML 对象和语法都受 SAP Sybase IQ 支持。

- SAP Sybase IQ 不支持以下 DML 和过程语句：
  - **EXPLAIN**
  - **GET DATA**
  - **INPUT**

- **PREPARE TO COMMIT**
- **PUT**
- **READTEXT**
- **ROLLBACK TRIGGER**
- **SYSTEM**
- **UNLOAD TABLE**
- **VALIDATE TABLE**

---

**注意：** 一组提取选项与 **UNLOAD TABLE** 的作用类似。

---

- SAP Sybase IQ 支持 **INSERT...LOCATION** 语法；而 SQL Anywhere 不支持。
- SAP Sybase IQ 和 SQL Anywhere 中的 **LOAD TABLE** 选项有所不同。
- SAP Sybase IQ 中的 **OPEN** 语句不支持 **BLOCK** 和 **ISOLATION LEVEL** 子句。
- SAP Sybase IQ 不支持触发器。
- 事务使用、隔离级别、检查点、自动生成 **COMMIT** 以及游标支持这些方面在 SAP Sybase IQ 和 SQL Anywhere 中有所不同。
- 从 SAP Sybase IQ 中的存储过程执行 **SELECT** 时，需要考虑 CIS 函数补偿性能注意事项。
- SAP Sybase IQ 忽略 Adaptive Server **SELECT** 语句中完全限定名的数据库名称限定符，如含 `<database name>.<owner>.<table name>` 的 **FROM** 子句。例如，SAP Sybase IQ 将查询 `SELECT * FROM XXX..TEST` 解释为 `SELECT * FROM TEST`。

## SAP Adaptive Server® Enterprise 和 SAP Sybase IQ 的区别和共享功能

---

SAP Sybase IQ 和 Adaptive Server 在存储过程支持和视图支持方面有所不同。

有关其它信息，请在使用产品时始终参考 SAP Sybase IQ 文档集。请在使用 Adaptive Server 时参考 Adaptive Server 文档集，SAP Sybase IQ 文档仅在涉及特定功能时才引用 Adaptive Server 文档。

### Adaptive Server 存储过程

某些存储过程不受 SAP Sybase IQ 支持。

SAP Sybase IQ 不再支持这些 Adaptive Server 存储过程：

- **sp\_addserver**
- **sp\_configure**
- **sp\_estspace**
- **sp\_help**
- **sp\_helpuser**

与其它 Sybase 数据库的兼容性

- **sp\_who**

SAP Sybase IQ 不再支持以下目录过程:

- **sp\_column\_privileges**
- **sp\_databases**
- **sp\_datatype\_info**
- **sp\_server\_info**

## **Adaptive Server 系统视图**

某些视图不受 SAP Sybase IQ 支持。

SAP Sybase IQ 不再支持这些 Adaptive Server 视图:

- sysalternates
- sysaudits
- sysauditoptions
- sysconstraints
- syscharsets
- sysconfigures
- syscurconfigs
- sysdatabases
- sysdepends
- sysdevices
- sysengines
- syskeys
- syslanguages
- syslocks
- syslogs
- sysloginroles
- sysmessages
- sysprocedures
- sysprocesses
- sysprotects
- sysreferences
- sysremotelogins
- sysroles
- syssegments
- sysservers
- sysrvroles
- systhresholds
- sysusages

*列名区别*

Adaptive Server 视图 SYSTYPES 中使用的列名为 "allownulls"。SAP Sybase IQ 视图 SYSTYPES 中使用的列名为 "allowsnulls"。





# 索引

## A

ABS 函数 133  
 ACOS 函数 133  
 Adaptive Server  
   兼容性 755, 761  
 ALL  
   条件 46  
 AND 条件 47  
 ANSI\_SUBSTRING 选项 321  
 ANY  
   条件 46  
 ARGN 函数 134  
 ASCII 函数 135  
 ASCII 值 135, 146  
 ASIN 函数 136  
 ATAN 函数 137  
 ATAN2 函数 137  
 AVG 函数 138  
 安全模型 381  
 安装目录  
   关于 3

## B

BETWEEN 条件 47  
 BFILE 函数  
   说明 139  
 BIGDATETIME 数据类型  
   兼容性 761  
 BIGINTTOHEX 函数 139  
 BIGTIME 数据类型  
   兼容性 761  
 BINARY 数据类型 353  
 BIT 数据类型  
   Transact-SQL 357  
   兼容性 758  
 BIT\_LENGTH 函数 140  
 BLOB 数据类型  
   LIKE 条件 50  
 BYTE\_LENGTH 函数 141  
 BYTE\_LENGTH64 函数  
   说明 142  
 BYTE\_SUBSTR 函数  
   说明 143

BYTE\_SUBSTR64 函数  
   说明 143  
 百分点  
   使用 NTILE 函数计算 253  
 保留字 17  
   列表 17  
 备份  
   活动较少期间 563  
   检查点操作期间 563  
 备份操作  
   摘要 389  
 备份历史记录文件  
   位置 9  
 本地计算机  
   环境设置 14  
 比较  
   关于 39  
 比较日期和时间 361  
 变量  
   Transact-SQL 780  
   关于 66  
   局部 66  
   连接级 67  
   全局 66, 68  
 标识  
   外部库 583  
 标识符  
   SAP SQL Anywhere 中的最大长度 24  
   SQL 语法 24  
   关于 24  
   区分大小写 762  
   唯一性 763  
 标识列  
   兼容性 765  
   支持作为缺省值 765  
 标准偏差  
   函数 310  
   样本函数 312  
   总体函数 311  
 标准偏差函数 307  
 表  
   iq\_dummy 75  
   Transact-SQL 764  
   显示有关信息 441

## 索引

表达式 27

CASE 35

Transact-SQL 36

长度 (以字节为单位) 164

转换为时间戳 180

表达式子查询

IF 语句中 778

并置字符串 31

## C

CASE 表达式 35

NULLIF 函数 255

CAST 函数 143, 366

Catalog 存储

监控 518

CEIL 函数 144

CEILING 函数 145

char

拆分为术语 562

CHAR 函数 146

CHAR 数据类型

关于 345

CHAR\_LENGTH 函数 147

CHAR\_LENGTH64 函数

说明 148

CHARACTER VARYING 数据类型

关于 345

剪裁尾随空白 347, 758

CHARACTER 数据类型

关于 345

CHARINDEX 函数 148

CHECK 条件

Transact-SQL 764

CHECKPOINT 语句

在检查点操作期间备份 563

CLOB 数据类型 50

COALESCE 函数 149

COL\_LENGTH 函数 150

COL\_NAME 函数 151

COMPUTE 子句

Transact-SQL 772

CONNECTION\_PROPERTY 函数 152

CONTAINS 条件

具有 TEXT 索引 48

具有 WD 索引 48

CONVERT 函数 153, 366

日期到整数的转换 156

日期到字符串的转换 156

整数到日期的转换 156

字符串到日期的转换 156

CORR 函数 157

COS 函数 158

cosine 158

COT 函数 159

COUNT 函数 162

COVAR\_POP 函数 160

COVAR\_SAMP 函数 161

CPU 利用率

数据库一致性检查程序 392

CREATE DECRYPTED DATABASE 语句 784

CREATE DEFAULT 语句

不支持 767

CREATE DOMAIN 语句

Transact-SQL 兼容性 767

使用 362

语法 364

CREATE ENCRYPTED DATABASE 语句 784

CREATE INDEX 语句

SAP Sybase IQ 767

Transact-SQL 767

CREATE RULE 语句

不支持 767

CREATE TABLE 语句

Transact-SQL 764

CREATE TRIGGER

不受支持 767

CUBE 操作

GROUPING 函数 201

CUME\_DIST 函数 163

CURRENT DATABASE

特殊值 62

CURRENT DATE

缺省值 62

特殊值 62

CURRENT PUBLISHER

缺省值 63

特殊值 63

CURRENT TIME

缺省值 63

特殊值 63

CURRENT TIMESTAMP

缺省值 63

特殊值 63

CURRENT USER

缺省值 64

特殊值 64

参数选择 134

- 参照完整性约束
  - 不支持 CASCADE 764
  - 兼容性 764
- 策略
  - 逻辑服务器 690
- 插入
  - SQL Anywhere 784
- 查询
  - Transact-SQL 770
- 拆分
  - 术语 562, 588
- 常量
  - Transact-SQL 37
  - 表达式中 28
- 成员资格
  - 逻辑服务器 689
- 触发器
  - 不受支持 767
- 窗口函数
  - 窗口分区 78
  - 窗口函数类型 78
  - 窗口名称或说明 78
- 窗口函数, 定义 78
- 窗口集合函数 77
- 窗口类型 78
- 窗口名称 78
- 窗口说明 78
- 创建
  - 数据类型 362, 364
- 存储过程
  - Adaptive Server Enterprise 785
  - sa\_char\_terms 562
  - sa\_external\_library\_unload 580
  - sa\_list\_external\_library 583
  - sa\_nchar\_terms 588
  - sa\_text\_index\_stats 618
  - sa\_text\_index\_vocab 619
  - sp\_iqbackupdetails 387
  - sp\_iqbackupsommary 389
  - sp\_iqclient\_lookup 400
  - sp\_iqmpxdumptlvlog 472
  - sp\_iqmpxinconnpoolinfo 474
  - sp\_iqmpxincheartbeatinfo 475
  - sp\_iqmpxincstatistics 476
  - sp\_iqmpxinfo 477
  - sp\_iqrestoreaction 500
  - sp\_iqsetcompression 504
  - sp\_iqshowcompression 506

- 存储过程语言
  - 概述 777
- 错误
  - Transact-SQL 781, 783
- 错误消息
  - ERRORMSG 函数 189
  - 获取文本 189

## D

- DATALENGTH 函数 164
- DATE 函数 165
- DATE 数据类型 358
- DATE\_ORDER 选项 361
- DATEADD 函数 165
- DATECEILING 函数 167
- DATEDIFF 函数 169
- DATEFLOOR 函数 171
- DATEFORMAT 函数 174
- DATENAME 函数 175
- DATEPART 函数 176
- DATEROUND 函数 177
- DATETIME 函数 180
- DAY 函数 180
- DAYNAME 函数 181
- DAYS 函数 182
- DB\_ID 函数 183
- DB\_NAME 函数 184
- DB\_PROPERTY 函数 185
- dbcc
  - 线程使用 392
- DBCC
  - 输出 392
  - 数据库验证 392
  - 性能 392
  - 运行时间 392
- DBCC\_LOG\_PROGRESS 选项 392
- dbinit 实用程序 757
- dbspace
  - 阻止读写操作 392
- DDL
  - SQL Anywhere 784
- DECIMAL data type 348
- DEGREES 函数 186
- DENSE\_RANK 函数 186
- DIFFERENCE 函数 188
- DISK 语句
  - 不支持 757

## 索引

### DML

SQL Anywhere 784

DOW 函数 189

DUMMY 表 652

大对象数据

LIKE 条件 50

代码页

和数据存储 346

当前用户

环境设置 14

登录策略

分配用户 469

复制 413, 468

逻辑服务器分配 688

登录管理

sp\_expireallpasswords 628

sp\_iqaddlogin 386

sp\_iqcopyloginpolicy 413, 468

定义窗口 78

对象

确定 ID 257

确定名称 257

显示有关信息 441

重命名 498

## E

ELSE

IF 表达式 34

ENDIF

IF 表达式 34

ERRORMSG 函数

SQL 语法 189

EVENT\_CONDITION 函数 190

EVENT\_CONDITION\_NAME 函数 192

EVENT\_PARAMETER 函数 192

EXISTS 条件 48

EXP 函数 194

EXP\_WEIGHTED\_AVG 函数 194

二进制数据

兼容性 759

## F

FIRST\_VALUE 函数 196

FLOAT 数据类型 350

FLOOR 函数 198

FOR BROWSE 语法

Transact-SQL 774

FP 索引

验证 392

FROM 子句 87

UPDATE 和 DELETE 777

发布者

SQL Remote 63

反斜杠

不允许在 SQL 标识符中使用 24

反余弦值 133

反正切值 137

反正切值比率 137

反正弦值 136

返回值

过程 781

分布函数 77

分隔 SQL 字符串 24

分离查询谓词 43

分配单元 505

分配映射

重置 392

分区

一致性检查 392

分区表

验证 392

分析函数 77

DENSE\_RANK 186

NTILE 253

PERCENT\_RANK 261

PERCENTILE\_CONT 262

PERCENTILE\_DISC 264

RANK 273

服务

注册表条目 14

服务器

属性 92

服务器管理

SQL Anywhere 和 IQ 783

辅助服务器

共享临时存储使用情况 505

## G

GETDATE 函数 199

GRAPHICAL\_PLAN 函数 199

GROUP BY

兼容性 772

GROUP\_MEMBER 函数

SQL 语法 202

GROUPING 函数 201

**GUID**

- NEWID 函数的 SQL 语法 247
- STRTOUUID 函数的 SQL 语法 317
- UIDTOSTR 函数的 SQL 语法 331

## 共享临时存储

- 显示使用情况分布 505

## 估计值

- 优化程序 54

## 挂起的连接 478

## 关键字

- SQL 17
- 列表 17

## 归类

- SORTKEY 函数 300

## 规则

- Transact-SQL 767

## 过程

- Transact-SQL 779
- Transact-SQL 概述 777
- 错误处理 781, 783
- 返回值 781
- 显示参数信息 489
- 显示有关信息 441, 487
- 转换 779

## 过程语言

- 概述 777

**H**

## HEXTOBIGINT 函数 203

## HEXTOINT 函数 204

- ASE\_FUNCTION\_BEHAVIOR 选项 204

## HOLDLOCK 语法

- Transact-SQL 774

## HOUR 函数 205

## HOURS 函数 206

## HTML\_PLAN 209

## HTML\_PLAN 函数 209

## HTTP 函数 86

## 函数

- ABS 函数 133
- ACOS 函数 133
- ARGN 函数 134
- ASCII 函数 135
- ASIN 函数 136
- ATAN 函数 137
- ATAN2 函数 137
- AVG 函数 138
- BFILE 函数 139

## BIGINTTOHEX 函数 139

## BIT\_LENGTH 函数 140

## BYTE\_LENGTH 函数 141

## BYTE\_LENGTH64 函数 142

## BYTE\_SUBSTR 函数 143

## BYTE\_SUBSTR64 函数 143

## CAST 函数 143

## CEIL 函数 144

## CEILING 函数 145

## CHAR 函数 146

## CHAR\_LENGTH 函数 147

## CHAR\_LENGTH64 函数 148

## CHARINDEX 函数 148

## COALESCE 函数 149

## COL\_LENGTH 函数 150

## COL\_NAME 函数 151

## CONNECTION\_PROPERTY 函数 152

## CONVERT 函数 153

## CORR 函数 157

## COS 函数 158

## COT 函数 159

## COUNT 函数 162

## COVAR\_POP 函数 160

## COVAR\_SAMP 函数 161

## CUME\_DIST 函数 163

## DATALENGTH 函数 164

## DATE 函数 165

## DATEADD 函数 165

## DATECEILING 函数 167

## DATEDIFF 函数 169

## DATEFLOOR 函数 171

## DATEFORMAT 函数 174

## DATENAME 函数 175

## DATEPART 函数 176

## DATETIME 函数 177

## DATETIME 函数 180

## DAY 函数 180

## DAYNAME 函数 181

## DAYS 函数 182

## DB\_ID 函数 183

## DB\_NAME 函数 184

## DB\_PROPERTY 函数 185

## DEGREES 函数 186

## DENSE\_RANK 函数 186

## DIFFERENCE 函数 188

## DOW 函数 189

## ERRORMSG 函数 SQL 语法 189

## EVENT\_CONDITION 函数 190

- EVENT\_CONDITION\_NAME 函数 192
- EVENT\_PARAMETER 函数 192
- EXP 函数 194
- EXP\_WEIGHTED\_AVG 函数 194
- FIRST\_VALUE 函数 196
- FLOOR 函数 198
- GETDATE 函数 199
- GRAPHICAL\_PLAN 199
- GROUP\_MEMBER 函数 SQL 语法 202
- GROUPING 函数 SQL 语法 201
- HEXTOBIGINT 函数 203
- HEXTOINT 函数 204
- HOUR 函数 205
- HOURS 函数 206
- HTML\_PLAN 函数 209
- HTTP 86
- IFNULL 函数 216
- INDEX\_COL 函数 216
- INSERTSTR 函数 217
- INTTOHEX 函数 218
- ISDATE 函数 SQL 语法 219
- ISNULL 函数 220
- ISNUMERIC 函数 SQL 语法 221
- LAST\_VALUE 223
- LCASE 函数 225
- LEFT 函数 228
- LEN 函数 SQL 语法 229
- LENGTH 函数 230, 233
- LIST 函数 231
- LOCATE 函数 234
- LOG 函数 235
- LOG10 函数 236
- LOWER 函数 237
- LTRIM 函数 238
- MAX 函数 239
- MEDIAN 函数 240
- MIN 函数 241
- MINUTE 函数 242
- MINUTES 函数 242
- MOD 函数 244
- MONTH 函数 244
- MONTHNAME 函数 245
- MONTHS 函数 246
- NEWID 函数 SQL 语法 247
- NEXT\_CONNECTION 函数 248
- NEXT\_DATABASE 函数 250
- NOW 函数 253
- NTILE 函数 253
- NULLIF 函数 255
- NUMBER 函数 256
- OBJECT\_ID 函数 257
- OBJECT\_NAME 函数 257
- OCTET\_LENGTH 函数 258
- PATINDEX 函数 259
- PERCENT\_RANK 函数 261
- PERCENTILE\_CONT 函数 262
- PERCENTILE\_DISC 函数 264
- PI 函数 266
- POWER 函数 266
- PROPERTY 函数 267
- PROPERTY\_DESCRIPTION 函数 268
- PROPERTY\_NAME 函数 269
- PROPERTY\_NUMBER 函数 270
- QUARTER 函数 270
- RADIANS 函数 271
- RAND 函数 272
- RANK 函数 273
- REGR\_AVGX 函数 274
- REGR\_AVGY 函数 275
- REGR\_COUNT 函数 276
- REGR\_INTERCEPT 函数 277
- REGR\_R2 函数 278
- REGR\_SLOPE 函数 280
- REGR\_SXX 函数 281
- REGR\_SXY 函数 282
- REGR\_SYY 函数 283
- REMAINDER 函数 284
- REPEAT 函数 285
- REPLACE 函数 286
- REPLICATE 函数 288
- REVERSE 函数 SQL 语法 289
- RIGHT 函数 290
- ROUND 函数 291
- ROWID 函数 293
- RTRIM 函数 295
- SAP Sybase IQ 扩展 373
- SECOND 函数 296
- SECONDS 函数 297
- SIGN 函数 298
- SIMILAR 函数 298
- SIN 函数 299
- SORTKEY 函数 300
- SOUNDEX 函数 304
- SPACE 函数 307
- SQLFLAGGER 函数 307
- SQRT 函数 309

SQUARE 函数 309  
 STDDEV 函数 310  
 STDDEV\_POP 函数 311  
 STDDEV\_SAMP 函数 312  
 STR 函数 314  
 STR\_REPLACE 函数 SQL 语法 315  
 STRING 函数 316  
 STRTOUUID 函数 SQL 语法 317  
 STUFF 函数 318  
 SUBSTR 函数 319  
 SUBSTRING 函数 319  
 SUBSTRING64 函数 322  
 SUM 函数 322  
 SUSER\_ID 函数 323  
 SUSER\_NAME 函数 324  
 TAN 函数 324  
 today 652  
 TODAY 函数 325  
 Transact-SQL 774  
 TRIM 函数 326  
 TRUNCNUM 函数 326  
 UCASE 函数 327  
 UPPER 函数 328  
 USER\_ID 函数 329  
 USER\_NAME 函数 330  
 UUIDTOSTR 函数 SQL 语法 331  
 VAR\_POP 函数 332  
 VAR\_SAMP 函数 333  
 VARIANCE 函数 334  
 WEEKS 函数 336  
 WEIGHTED\_AVG 函数 337  
 WIDTH\_BUCKET 函数 339  
 YEAR 函数 341  
 YEARS 函数 342  
 YMD 函数 343  
 按字母顺序排列的列表 132  
 窗口化集合 77  
   分布 77  
   分析 77  
   集合 75  
   排名 77  
   日期和时间 83  
   数据类型转换 83  
   数值 77, 87  
   统计 77  
   一致的结果 87  
   用户定义的 131  
   杂项 132

字符串 88  
 函数, 集合  
   GROUPING 201  
 函数, 数据类型转换  
   ISDATE 219  
 函数, 杂项  
   ERRORMSG 189  
   ISNUMERIC 221  
   NEWID 247  
 函数, 字符串 209, 229, 289, 315  
   STRTOUUID 317  
   UUIDTOSTR 331  
 环境变量  
   ILOGDIR16 9  
   PATH 12  
   SQLCONNECT 12  
   SYBASE 12  
   SYBASE\_JRE 13  
   SYBASE\_OCS 13  
   关于 6  
 缓冲区高速缓存  
   使用 sp\_iqsysmon 监控 518  
 恢复操作  
   一致状态 500

**I**  
 IF 表达式 34  
 IFNULL 函数 216  
 IMAGE 数据类型 759  
   兼容性 762  
 IN 条件 49  
 INDEX\_COL 函数 216  
 INDEX\_PREFERENCE 选项 57  
 INSERTSTR 函数 217  
 INTEGER 数据类型 349  
 INTTOHEX 函数 218  
   ASE\_FUNCTION\_BEHAVIOR 选项 218  
 IQ 主存储库 757  
 iq\_dummy 表 75, 652  
   获取一致的结果 83  
 IQ\_USE\_DIRECTIO 环境变量 10  
 IQCHARSET 环境变量 7  
 IQDIR16 环境变量 7  
 iqinit 实用程序 757  
 IQLANG 8  
 IQLANG 环境变量 8  
 ILOGDIR16 环境变量 9  
 IQTMP16 环境变量 10

## 索引

IS NULL 条件 50  
ISDATE 函数  
    SQL 语法 219  
ISNULL 函数 220  
ISNUMERIC 函数  
    SQL 语法 221

## J

Java  
    用户定义的函数 132  
Java 数据类型  
    兼容性 762  
Java 运行时环境  
    设置 13  
JAVA\_HOME 环境变量 11  
集合函数 75  
    AVG 138  
    COUNT 162  
    LIST 231  
    MAX 239  
    MIN 241  
    STDDEV 310  
    STDDEV\_POP 311  
    STDDEV\_SAMP 312  
    SUM 322  
    VAR\_POP 332  
    VAR\_SAMP 333  
    VARIANCE 334  
计算列  
    不支持 765  
兼容性  
    Adaptive Server 755  
    Adaptive Server 数据类型 761  
    参照完整性约束 764  
监控  
    sp\_iqsysmon 过程 518  
剪裁尾随空白 347, 758  
检查约束 764  
    Transact-SQL 兼容性 764  
    强制执行 764  
建议  
    存储 450  
    清除 450  
    显示 450  
键  
    显示有关信息 485  
结果集  
    Transact-SQL 780

禁用  
    压缩 504  
局部变量  
    关于 66  
绝对值 133

## K

可移植 SQL 770  
空白  
    剪裁尾随 758  
    剪裁尾随空白 347  
空值  
    关于 73  
空值比较  
    Transact-SQL 773  
口令  
    到期 628  
    区分大小写 763  
    添加或修改 484

## L

language\_code 8  
LAST USER  
    特殊值 64  
LAST\_VALUE 函数 223  
LCASE 函数 225  
LD\_LIBRARY\_PATH 环境变量 11  
LEFT 函数 228  
LEN 函数  
    SQL 语法 229  
LENGTH 函数 230, 233  
LIBPATH 环境变量 11  
LIKE 条件 50  
    LONG BINARY 数据 50  
    LONG VARCHAR 数据 50  
    大对象数据 50  
LIST 函数 231  
LOB 的数据压缩 504  
    更改设置 504  
    显示设置 506  
LOB 数据的压缩 504  
    更改设置 504  
    显示设置 506  
LOB 数据类型  
    LIKE 条件 50  
LOB 压缩  
    更改设置 504



- 禁用 504
- 启用 504
- 显示设置 506
- LOCATE 函数 234
- LOG 函数 235
- LOG10 函数 236
- logarithm (base 10) 236
- LONG BINARY 数据类型 759, 762
  - LIKE 条件 50
- LONG VARCHAR 数据类型 50
- LOWER 函数 237
- LTRIM 函数 238
- LVC 单元 392
- 类型转换 366
- 立即刷新 618, 619
- 连接
  - Transact-SQL 772
    - 确定 ID 号 248
    - 属性 92
    - 外连接运算符 34
    - 显示有关信息 556
    - 自动 372
  - 连接等同性条件 60
  - 连接级变量
    - 关于 67
  - 连接属性值 152
  - 连接信息
    - sp\_iqcontext 410
  - 连接运算符
    - ANSI 773
    - Transact-SQL 772
  - 链式外连接 773
- 列
  - 和用户定义数据类型 363
  - 命名 29
- 列表
  - 外部库 583
- 列长度 150
- 列名称 151
- 列缺省值
  - 不支持 765
- 临时表
  - Transact-SQL 766
- 零长度字符串
  - Transact-SQL 773
- 逻辑服务器
  - 系统表 687, 689
- 逻辑服务器策略
  - 系统表 687, 690

- 逻辑服务器策略选项
  - 系统表 690
- 逻辑服务器登录策略选项
  - 系统表 688

## M

- MAX 函数 239
- MEDIAN 函数 240
- MIN 函数 241
- MINUTE 函数 242
- MINUTES 函数 242
- MOD 函数 244
- MONEY 数据类型 352
- MONTH 函数 244
- MONTHNAME 函数 245
- MONTHS 函数 246
- MPXServerName 列 406
- Multiplex
  - 检查配置 480
  - 逻辑服务器成员资格 689
  - 系统过程 406
  - 显示版本信息 480
- 密钥
  - 验证 392
- 命名空间
  - 索引 763
- 模式匹配
  - 关于 50
  - 和归类 52
  - 限制 51
- 目录
  - Adaptive Server 的兼容性 757
  - 系统表 649
- 目录存储库
  - SAP Sybase IQ 757
- 目录结构 3

## N

- nchar
  - 拆分为术语 588
- NEWID 函数
  - SQL 语法 247
- NEXT\_CONNECTION 函数 248
- NEXT\_DATABASE 函数 250
- NOT 条件 54
- NOW 函数 253
- NTILE 函数 253

## 索引

### NULL

Transact-SQL 兼容性 764

NULLIF 函数 36, 255

NUMBER 函数 256

内存

使用 sp\_iqsysmon 监控 518

## O

OBJECT\_ID 函数 257

OBJECT\_NAME 函数 257

OCTET\_LENGTH 函数 258

OLAP

DENSE\_RANK 函数 186

GROUPING 函数 201

NTILE 函数 253

PERCENT\_RANK 函数 261

PERCENTILE\_CONT 函数 262

PERCENTILE\_DISC 函数 264

RANK 函数 273

STDDEV 函数 310

窗口规范 78

窗口函数 78

窗口函数类型 78

窗口集合函数 77

窗口名称 78

分布函数 77

排名函数 77

数值函数 77

统计函数 77

行比较函数 77

OLAP OVER 子句 78

OLAP 函数

兼容性 775

行间函数 81

ON EXCEPTION RESUME 子句

Transact-SQL 783

Open Client 设置 13

OR 关键字 47

OVER 子句 78

## P

PATH 环境变量 12

PATINDEX 函数 259

PERCENT\_RANK 函数 261

PERCENTILE\_CONT 函数 262

PERCENTILE\_DISC 函数 264

PI 函数 266

POWER 函数 266

PRINT 命令

Transact-SQL 779

PROPERTY 函数 267

PROPERTY\_DESCRIPTION 函数 268

PROPERTY\_NAME 函数 269

PROPERTY\_NUMBER 函数 270

排名函数 77

批处理

Transact-SQL 概述 778

编写 778

撇号

字符串中 26

平方根函数 309

平均 138

## Q

QUARTER 函数 270

QUOTED\_IDENTIFIER 选项 38

启用

压缩 504

嵌套外连接 773

区分大小写

Transact-SQL 兼容性 762

比较条件 40

标识符 762

和模式匹配 52

口令 763

数据 762

数据库 762

用户 ID 763

用户定义数据类型 762

区域设置

设置 8

全局变量

关于 66, 68

兼容性 69

列表 68

全局唯一标识符

NEWID 函数的 SQL 语法 247

权限

Adaptive Server 768

缺省值

CURRENT DATABASE 62

CURRENT DATE 62

CURRENT PUBLISHER 63

CURRENT TIME 63

CURRENT TIMESTAMP 63

CURRENT USER 64  
 LAST USER 64  
 TIMESTAMP 65  
 Transact-SQL 767  
 USER 65  
 不支持 765

## R

RADIANS 函数 271  
 RAISERROR 语句  
   ON EXCEPTION RESUME 783  
   Transact-SQL 782  
 RAND 函数 272  
 RANK 函数 273  
 REGR\_AVGX 函数 274  
 REGR\_AVGY 函数 275  
 REGR\_COUNT 函数 276  
 REGR\_INTERCEPT 函数 277  
 REGR\_R2 函数 278  
 REGR\_SLOPE 函数 280  
 REGR\_SXX 函数 281  
 REGR\_SXY 函数 282  
 REGR\_SYY 函数 283  
 REMAINDER 函数 284  
 REPEAT 函数 285  
 REPLACE 函数 286  
   在 SELECT INTO 语句中 31, 155, 217, 225,  
     228, 237, 238, 285, 286, 288, 290,  
     295, 326–328  
 REPLICATE 函数 288  
 resetclocks  
   sp\_iqcheckdb 选项 392  
 REVERSE 函数  
   SQL 语法 289  
 RIGHT 函数 290  
 ROLLUP 操作  
   GROUPING 函数 201  
 ROUND 函数 291  
 ROWID 函数 293  
 RTRIM 函数 295  
 日期  
   查询 360  
   将字符串解释为日期 361  
   确定当前日期 253, 325  
   算术表达式 776  
 日期和时间函数 83  
   DATE 165  
   DATEADD 165  
   DATECEILING 167  
   DATEDIFF 169  
   DATEFLOOR 171  
   DATEFORMAT 174  
   DATENAME 175  
   DATEPART 176  
   DATEROUND 177  
   DATETIME 180  
   DAY 180  
   DAYNAME 181  
   DAYS 182  
   DOW 189  
   GETDATE 199  
   HOUR 205  
   HOURS 206  
   IQ 功能 371  
   MINUTE 242  
   MINUTES 242  
   MONTH 244  
   MONTHNAME 245  
   MONTHS 246  
   NOW 253  
   QUARTER 270  
   SECOND 296  
   SECONDS 297  
   TODAY 325  
   WEEKS 336  
   YEAR 341  
   YEARS 342  
   YMD 343  
   获取一致的结果 83  
 日期和时间数据类型  
   兼容性 760  
 日期时间和时间数据类型  
   兼容性 760

## S

sa\_char\_terms 存储过程 562  
 sa\_checkpoint\_execute 系统过程 563  
 sa\_conn\_list 系统过程 568  
 sa\_describe\_shapefile 系统过程 573  
 sa\_external\_library\_unload 存储过程 580  
 sa\_get\_ldapservers\_status 系统过程 582  
 sa\_get\_table\_definition 系统过程 385  
 sa\_list\_external\_library 存储过程 583  
 sa\_locks 系统过程 584  
 sa\_nchar\_terms 存储过程 588  
 sa\_performance\_diagnostics 系统过程 589

- sa\_text\_index\_stats 存储过程 618
- sa\_text\_index\_vocab 存储过程 619
- SAP Sybase IQ 用户管理
  - sp\_iqdroplogin 431
- SECOND 函数 296
- SECONDS 函数 297
- SELECT INTO
  - Transact-SQL 776
  - 使用 REPLACE 函数 31, 155, 217, 225, 228, 237, 238, 285, 286, 288, 290, 295, 326–328
- SELECT 语句
  - Transact-SQL 770
- SET OPTION 语句
  - Transact-SQL 770
- SHARED 语法
  - Transact-SQL 774
- SIGN 函数 298
- SIGNAL 语句
  - Transact-SQL 782
- SIMILAR 函数 298
- SIN 函数 299
- SMALLDATETIME 数据类型 358
- SMALLMONEY 数据类型 352
- SOME 条件 46
- SORTKEY 函数 300
- SOUNDEX 函数 304
- sp\_auth\_sys\_role\_info 623
- sp\_displayroles 系统过程 624
- sp\_expireallpasswords 系统过程 628
- sp\_has\_role 函数 305
- sp\_iq\_reset\_identity 系统过程 499
- sp\_iqaddlogin 系统过程 386
- sp\_iqbackupdetails 存储过程 387
- sp\_iqbackupsummary 存储过程 389
- sp\_iqbrestoreaction 存储过程 500
- sp\_iqcardinality\_analysis 系统过程 390
- sp\_iqcheckdb
  - DBCC\_LOG\_PROGRESS 选项 392
  - resetclocks 选项 392
  - 分配模式 392
  - 检查模式 392
  - 沙漏模式 392
  - 示例输出 392
  - 输出 392
  - 性能 392
  - 验证模式 392
  - 语法 392
  - 运行时间 392
- sp\_iqcheckdb 系统过程 392
- sp\_iqcheckoptions 系统过程 398
- sp\_iqcolumn 系统过程 401
- sp\_iqcolumnmetadata 过程 404
- sp\_iqcolumnuse 系统过程 404
- sp\_iqconnection 系统过程 406
- sp\_iqcontext 系统过程 410
- sp\_iqcopyloginpolicy 系统过程 413, 468
- sp\_iqcursorinfo 系统过程 413
- sp\_iqdatatype 系统过程 415
- sp\_iqdbsize 系统过程 418
- sp\_iqdbspace 系统过程 419
- sp\_iqdbspaceinfo 系统过程 423
- sp\_iqdbspaceobjectinfo 系统过程 426
- sp\_iqdbstatistics 系统过程 430
- sp\_iqdroplogin 系统过程 431
- sp\_iqemptyfile 系统过程 432
- sp\_iqestdbspaces 系统过程 433
- sp\_iqestspace 系统过程 435
- sp\_iqevent 系统过程 436
- sp\_iqfile 系统过程 438
- sp\_iqhelp 系统过程 441
- sp\_iqindex 系统过程 447
- sp\_iqindex\_alt 系统过程 447
- sp\_iqindexadvice 系统过程 450
- sp\_iqindexfragmentation 系统过程 452
- sp\_iqindexinfo
  - 显示索引信息 454
- sp\_iqindexinfo 系统过程 453
- sp\_iqindexmetadata 系统过程 455
- sp\_iqindexrebuildwidedata 系统过程 458
- sp\_iqindexsize 系统过程 459
- sp\_iqindexuse 系统过程 461
- sp\_iqlocks 系统过程 465
- sp\_iqmergerlvstore 系统过程 468
- sp\_iqmodifylogin 469
- sp\_iqmodifylogin 系统过程 469
- sp\_iqmpxdumptlvlog 存储过程 472
- sp\_iqmpxfilestatus 系统过程 473
- sp\_iqmpxinconnpoolinfo 存储过程 474
- sp\_iqmpxinheartbeatinfo 存储过程 475
- sp\_iqmpxincstatistics 存储过程 476
- sp\_iqmpxinfo 存储过程 477
- sp\_iqmpxsuspendedconninfo 系统过程 478
- sp\_iqmpxvalidate 系统过程 480
- sp\_iqmpxversioninfo 系统过程 480
- sp\_iqobjectinfo 系统过程 481
- sp\_iqpassword 系统过程 484

- sp\_iqpkkeys 系统过程 485
- sp\_iqprocedure 系统过程 487
- sp\_iqprocparm 系统过程 489
- sp\_iqrebuildindex 系统过程 492, 503
- sp\_iqrename 系统过程 498
- sp\_iqrlvmemory 系统过程 502
- sp\_iqsetcompression 存储过程 504
- sp\_iqsetcompression 系统过程 379
- sp\_iqsharedtempdistrib 系统过程 505
- sp\_iqshowcompression 存储过程 506
- sp\_iqshowcompression 系统过程 379
- sp\_iqshowpsex 系统过程 507
- sp\_iqspaceinfo 系统过程 509
  - 示例输出 509
- sp\_iqspaceused 系统过程 510
- sp\_iqstatistics 系统过程 512
- sp\_iqstatus 系统过程 515
  - 示例输出 515
- sp\_iqsysmon 系统过程 518
- sp\_iqtable 系统过程 539
- sp\_iqtablesize 系统过程 543
- sp\_iqtableuse 系统过程 544
- sp\_iqtransaction 系统过程 545
- sp\_iqunusedcolumn 系统过程 549
- sp\_iqunusedindex 系统过程 550
- sp\_iqunusedtable 系统过程 551
- sp\_iqversionuse 系统过程 552
- sp\_iqview 系统过程 554
- sp\_iqwho 系统过程 556
- sp\_iqworkmon 系统过程 559
- sp\_objectpermission 系统过程 629
- sp\_sys\_priv\_role\_info 643
- SPACE 函数 307
- SQL
  - IQ 术语区别 371
  - 用户定义的函数 131
- SQL Anywhere 755
  - 参照完整性约束 764
- SQL 函数
  - ERRORMSG 函数语法 189
  - GRAPHICAL\_PLAN 函数语法 199
  - GROUP\_MEMBER 函数语法 202
  - GROUPING 函数语法 201
  - HTML\_PLAN 函数语法 209
  - ISDATE 函数语法 219
  - ISNUMERIC 函数语法 221
  - LEN 函数语法 229
  - NEWID 函数语法 247
  - REVERSE 函数语法 289
  - STR\_REPLACE 函数语法 315
  - STRTOUUID 函数语法 317
  - UUIDTOSTR 函数语法 331
  - 兼容性 774
- SQL 语法
  - CURRENT DATABASE 特殊值 62
  - CURRENT PUBLISHER 特殊值 63
  - CURRENT USER 特殊值 64
  - LAST USER 特殊值 64
  - TIMESTAMP 特殊值 65
  - USER 特殊值 65
  - 标识符 24
- SQL 语句
  - CREATE DECRYPTED DATABASE 784
  - CREATE ENCRYPTED DATABASE 784
- SQL92 一致性 371
- SQLCODE
  - 特殊值 64
- SQLCONNECT 环境变量 12
- SQLFLAGGER 函数 307
- SQLSTATE
  - 特殊值 65
- SQRT 函数 309
- SQUARE 函数 309
- STDDEV 函数 310
- STDDEV\_POP 函数 311
- STDDEV\_SAMP 函数 312
- STR 函数 314
- STR\_REPLACE 函数
  - SQL 语法 315
- STRING 函数 316
- STRTOUUID 函数
  - SQL 语法 317
- STUFF 函数 318
- SUBSTR 函数 319
- SUBSTRING 函数 319
- SUBSTRING64 函数
  - 说明 322
- SUM 函数 322
- SUSER\_ID 函数 323
- SUSER\_NAME 函数 324
- SYBASE 环境变量 12
- SYBASE\_JRE 环境变量 13
- SYBASE\_OCS 环境变量 13
- SYSIQBACKUPHISTORY 系统视图 682
- SYSIQBACKUPHISTORYDETAIL 系统视图
  - 683

## 索引

- SYSIQDBFILE 系统视图 684
- SYSIQDBSPACE 系统视图 685
- SYSIQIDX 系统视图 685
- SYSIQPARTITIONCOLUMN 系统视图 692
- SYSIQRLVMERGEHISTORY 系统视图 692
- SYSIQRVLOG 系统视图 693
- SYSIQTAB 系统视图 693
- SYSPARTITION 系统视图 703
- SYSPARTITIONKEY 系统视图 704
- SYSPARTITIONS 系统视图 705
- SYSPARTITIONSCHEME 系统视图 705
- SYSROLEGRANTTEXT 系统视图 716
- SYSSEPARTITIONKEY 系统视图 723
- 三值逻辑
  - 关于 42
  - 空值 73
- 沙漏模式 392
- 设备
  - 管理 757
- 时间
  - 查询 360
- 实例
  - 外部库 583
- 实用程序
  - SQL Anywhere 783
- 事件
  - EVENT\_CONDITION 函数 190
  - EVENT\_CONDITION\_NAME 函数 192
  - EVENT\_PARAMETER 函数 192
  - 显示有关信息 436, 441
- 事务
  - 挂起 478
- 事务管理
  - 使用 sp\_iqsysmon 监控 518
- 视图
  - 可更新的 777
  - 显示有关信息 441
- 属性
  - ID 说明 268
  - 服务器 92
  - 服务器级 267
  - 连接 92
  - 确定名称 269
  - 确定属性号 270
  - 数据库 115
- 术语
  - 拆分 562, 588
  - 行位置 562, 588

- 数据
  - 区分大小写 762
- 数据库
  - 区分大小写 762
  - 确定 ID 号 183, 250
  - 确定名称 184
  - 属性 115
  - 属性值 185
  - 系统表 649
  - 系统过程 379
- 数据库对象
  - 标识 24
  - 确定 ID 257
  - 确定名称 257
- 数据库空间
  - 管理 757
- 数据库选项
  - DATE\_ORDER 361
  - QUOTED\_IDENTIFIER 38
- 数据类型
  - Adaptive Server 758
  - IMAGE 759, 762
  - LONG BINARY 759, 762
  - SAP Sybase IQ 758
  - SQL Anywhere 758
  - TEXT 346, 758, 761
  - UNIQUEIDENTIFIERSTR 345
  - 创建 364
  - 二进制 352
  - 和兼容性 367
  - 日期和时间 358
  - 数值 348
  - 显示有关信息 415, 441
  - 用户定义 362
  - 字符 345
- 数据类型兼容性
  - IMAGE 数据 762
  - Java 数据 762
  - TEXT 数据 761
  - 二进制数据 759
  - 日期和时间数据 760
  - 日期时间和时间数据 760
  - 数值数据 761
  - 位数据 758
  - 字符数据 758
- 数据类型转换
  - BIT 到 BINARY 368
  - BIT 到 CHAR 369
  - BIT 到 VARBINARY 368

- BIT 到 VARCHAR 369
- CHAR 到 BIT 369
- VARCHAR 到 BIT 369
- 关于 366
- 函数 83
- 数据类型转换函数 83
  - BIGINTTOHEX 139
  - CAST 143
  - CONVERT 153
  - HEXTOBIGINT 203
  - HEXTOINT 204
  - INTTOHEX 218
- 数学表达式 30
- 数值函数 77, 87
  - ABS 133
  - ACOS 133
  - ASIN 136
  - ATAN 137
  - ATAN2 137
  - CEIL 144
  - CEILING 145
  - COS 158
  - COT 159
  - DEGREES 186
  - EXP 194
  - FLOOR 198
  - LOG 235
  - LOG10 236
  - MOD 244
  - PI 266
  - POWER 266
  - RADIANS 271
  - RAND 272
  - REMAINDER 284
  - ROUND 291
  - SIGN 298
  - SIN 299
  - SQRT 309
  - SQUARE 309
  - TAN 324
  - TRUNCNUM 326
  - WIDTH\_BUCKET 339
  - 一致的结果 87
- 数值数据类型
  - 兼容性 761
- 数字 28
- 数字的对数 235
- 刷新
  - 立即 618, 619

- 双引号
  - 不允许在 SQL 标识符中使用 24
  - 数据库对象 24
- 搜索条件
  - ALL 或 ANY 条件 46
  - BETWEEN 条件 47
  - CONTAINS 条件 48
  - EXISTS 条件 48
  - IN 条件 49
  - IS NULL 条件 50
  - LEADING SUBSTRING SEARCH 条件 50
  - LIKE 条件 50
  - NOT 条件 54
  - 比较条件 40
  - 关于 39
  - 三值逻辑 42
  - 真值条件 54
  - 子查询 43
- 算术表达式 30
  - 日期 776
- 索引
  - Adaptive Server 767
  - SAP Sybase IQ 767
  - SQL Anywhere 767
  - Transact-SQL 763
- 索引首选项提示 56
- 锁
  - 显示 465

## T

- TAN 函数 324
- tangent 324
- TEXT 数据类型 346, 758
  - 兼容性 758, 761
- TEXT 索引
  - 统计信息 618, 619
- THEN
  - IF 表达式 34
- TIME 数据类型 358
- TIMESTAMP
  - 数据类型 358
  - 数据类型兼容性 760
  - 特殊值 65
  - 转换表达式 180
- TINYINT 数据类型 349
- TLV 日志
  - 显示内容 472
- TODAY 函数 325, 652

**Transact-SQL**

- 比较条件 41
- 编写可移植 SQL 770
- 变量 780
- 表达式 36
- 参照完整性约束 764
- 常量 37
- 创建兼容的数据库 762
- 概述 755
- 关于 755
- 过程 777
- 过程语言概述 777
- 结果集 780
- 局部变量 67
- 连接 772
- 批处理 778
- 外连接运算符 34
- 系统目录 750
- 用户定义数据类型 366
- 逐位运算符 32
- 字符串 38

**Transact-SQL 兼容性**

- 数据库 762

**TRIM 函数 326**

**TRUNCNUM 函数 326**

**特殊值**

- CURRENT DATABASE 62
- CURRENT DATE 62
- CURRENT PUBLISHER 63
- CURRENT TIME 63
- CURRENT TIMESTAMP 63
- CURRENT USER 64
- LAST USER 64
- SQLCODE 64
- SQLSTATE 65
- TIMESTAMP 65
- USER 65

**特殊字符**

- 字符串中 26

**提示**

- 索引首选项 56
- 执行阶段 58, 59

**体系结构**

- Adaptive Server Enterprise 756
- SQL Anywhere 756

**条件**

- 用户提供的用于查询的条件 54

**条件提示字符串 55**

**通用唯一标识符**

- NEWID 函数的 SQL 语法 247

**统计函数 77**

**统计信息**

- TEXT 索引 618, 619

**U**

**UCASE 函数 327**

**UNION**

- 在子查询中 771

**UNIQUEIDENTIFIER 数据类型 356**

**UNIQUEIDENTIFIERSTR 数据类型**

- 关于 345

**UPPER 函数 328**

**USER**

- 特殊常量 652
- 特殊值 65

**USER\_ID 函数 329**

**USER\_NAME 函数 330**

**UUID**

- NEWID 函数的 SQL 语法 247
- STRTOUUID 函数的 SQL 语法 317
- UIDTOSTR 函数的 SQL 语法 331

**UIDTOSTR 函数**

- SQL 语法 331

**V**

**VAR\_POP 函数 332**

**VAR\_SAMP 函数 333**

**VARBINARY 数据类型 353**

**VARCHAR 数据类型**

- 关于 345
- 剪裁尾随空白 347, 758

**VARIANCE 函数 334**

**W**

**WEEKS 函数 336**

**WEIGHTED\_AVG 函数 337**

**WHERE 子句**

- Transact-SQL 772

**WIDTH\_BUCKET 函数 339**

**WITHIN GROUP 子句 80**

**外部库**

- 列表 583
- 卸载 580



## 外连接

Transact-SQL 772

和子查询 29

链式 773

嵌套 773

运算符 34

## 尾随空白

剪裁 347, 758

## 位长度 140

## 谓词

分离 43

关于 39

## 文件

位置 4

## 文字

最大长度 375

**X**

## 系统变量 68

## 系统表

Adaptive Server 的兼容性 757

DUMMY 652

ISYSIQLLOGICALSERVER 687

ISYSIQLLOGINPOLICYLSINFO 688

ISYSIQLSLOGINPOLICIES 688

ISYSIQLSLOGINPOLICYOPTION 688

ISYSIQLSMEMBER 689

ISYSIQLSMEMBERS 689

ISYSIQLSPOLICY 690

ISYSIQLSPOLICYOPTION 690

关于 649

显示信息 441

## 系统过程

sa\_checkpoint\_execute 563

sa\_conn\_list 568

sa\_describe\_shapefile 573

sa\_get\_table\_definition 385

sa\_locks 584

sa\_performance\_diagnostics 589

sp\_expireallpasswords 628

sp\_iqaddlogin 386

sp\_iqbackupdetails 387

sp\_iqbackupsummary 389

sp\_iqcardinality\_analysis 390

sp\_iqcheckdb 392

sp\_iqcheckoptions 398

sp\_iqclient\_lookup 400

sp\_iqcolumn 401

sp\_iqcolumnuse 404

sp\_iqconnection 406

sp\_iqcontext 410

sp\_iqcopyloginpolicy 413, 468

sp\_iqcursorinfo 413

sp\_iqdatatype 415

sp\_iqdbsize 418

sp\_iqdbspaceobjectinfo 426

sp\_iqdbstatistics 430

sp\_iqdroplogin 431

sp\_iqemptyfile 432

sp\_iquestdbspaces 433

sp\_iquestspace 435

sp\_iqevent 436

sp\_iqfile 438

sp\_iqhelp 441

sp\_iqindex 447

sp\_iqindex\_alt 447

sp\_iqindexadvice 450

sp\_iqindexsize 459

sp\_iqindexuse 461

sp\_iqmodifylogin 469

sp\_iqmpxfilestatus 473

sp\_iqmpxinconnpoolinfo 474

sp\_iqmpxinheartbeatinfo 475

sp\_iqmpxincstatistics 476

sp\_iqmpxinfo 477

sp\_iqmpxsuspendedconninfo 478

sp\_iqobjectinfo 481

sp\_iqpassword 484

sp\_iqpkeys 485

sp\_iqprocedure 487

sp\_iqprocparm 489

sp\_iqrename 498

sp\_iqrestoreaction 500

sp\_iqsetcompression 379

sp\_iqshowcompression 379

sp\_iqshowpsex 507

sp\_iqspaceinfo 509

sp\_iqspaceused 510

sp\_iqstatistics 512

sp\_iqstatus 515

sp\_iqsysmon 518

sp\_iqtable 539

sp\_iqtablesize 543

sp\_iqtableuse 544

sp\_iqtransaction 545

sp\_iqunusedcolumn 549

sp\_iqunusedindex 550

- sp\_iqunusedtable 551
- sp\_iqversionuse 552
- sp\_iqview 554
- sp\_iqwho 556
- sp\_iqworkmon 559
- 关于 379
- 显示有关信息 441
- 系统函数 90
  - COL\_LENGTH 150
  - COL\_NAME 151
  - CONNECTION\_PROPERTY 152
  - DATALength 164
  - DB\_ID 183
  - DB\_NAME 184
  - DB\_PROPERTY 185
  - EVENT\_CONDITION 190
  - EVENT\_CONDITION\_NAME 192
  - EVENT\_PARAMETER 192
  - INDEX\_COL 216
  - NEXT\_CONNECTION 248
  - NEXT\_DATABASE 250
  - OBJECT\_ID 257
  - OBJECT\_NAME 257
  - PROPERTY 267
  - PROPERTY\_DESCRIPTION 268
  - PROPERTY\_NAME 269
  - PROPERTY\_NUMBER 270
  - SUSER\_ID 323
  - SUSER\_NAME 324
  - Transact-SQL 776
  - USER\_ID 329
  - USER\_NAME 330
- 系统角色
  - 迁移兼容性角色 380
- 系统目录
  - Adaptive Server 的兼容性 757
  - Transact-SQL 750
- 系统视图
  - Adaptive Server 786
  - SYSCOLUMNS ASE 兼容性视图 665
  - SYSINDEXES ASE 兼容性视图 681
  - SYSIQBACKUPHISTORY 682
  - SYSIQBACKUPHISTORYDETAIL 683
  - SYSIQCOLUMN 684
  - SYSIQDBFILE 684
  - SYSIQDBSPACE 685
  - SYSIQFILE 685
  - SYSIQIDX 685
  - SYSIQINFO 686
  - SYSIQITAB 693
  - SYSIQLLOGICALSERVER 687
  - SYSIQLLOGINPOLICYLSINFO 688
  - SYSIQLSLOGINPOLICIES 688
  - SYSIQLSLOGINPOLICYOPTION 688
  - SYSIQLSMEMBER 689
  - SYSIQLSMEMBERS 689
  - SYSIQLSPOLICY 690
  - SYSIQLSPOLICYOPTION 690
  - SYSIQMPXSERVER 690
  - SYSIQMPXSERVERAGENT 691
  - SYSIQOBJECTS ASE 兼容性视图 691
  - SYSIQPARTITIONCOLUMN 692
  - SYSIQLVLMERGEHISTORY 692
  - SYSIQRVLOG 693
  - SYSIQTABCOL 694
  - SYSIQTABLE 系统视图 695
  - SYSIQVINDEX ASE 兼容性视图 695
  - SYSLOGINS ASE 兼容性视图 700
  - SYSOBJECTS ASE 兼容性视图 702
  - SYSPARTITION 703
  - SYSPARTITIONKEY 704
  - SYSPARTITIONS 705
  - SYSPARTITIONSCHEME 705
  - SYSSUBPARTITIONKEY 723
  - SYSTYPES ASE 兼容性视图 741
  - SYSUSERS ASE 兼容性视图 747
  - SYSVIEW 748
  - 统一 653
- 线程
  - dbcc 392
- 协调器
  - 共享临时存储使用情况 505
  - 挂起的事务 478
- 卸载
  - 外部库 580
- 星期几 (DOW) 189
- 行
  - 计数 162
- 性能
  - sp\_iqshowpsex 连接信息 507
  - sp\_iqsysmon 过程 518
  - 监控 518
- 选项
  - DBCC\_LOG\_PROGRESS 392
  - QUOTED\_IDENTIFIER 38
  - 登录策略 690
  - 逻辑服务器登录策略 688

## 选择性

- 提示 56
- 显式 54
- 用户提供的条件 54

**Y**

YEAR 函数 341

YEARS 函数 342

YMD 函数 343

## 验证

- 分区表 392
- 密钥 392
- 索引 392

样本方差函数 333

## 一致性检查

- 分区 392

## 一致状态

- 恢复 500

## 引号

- SQL 标识符 24
- 数据库对象 24
- 字符串 26

## 用户

- 删除 431
- 添加 386
- 显示有关信息 556
- 修改 469

## 用户 ID

- Adaptive Server 768
- 区分大小写 763
- 通过用户名确定 323, 329

用户定义的函数 131

- 兼容性 776

用户定义数据类型

- CREATE DOMAIN 语句 364
- Transact-SQL 366
- 关于 362
- 区分大小写 762

## 用户管理

- 请参见 登录管理

## 用户名

- 通过用户 ID 确定 324, 330

用户提供的关于连接等同性条件的提示 60

用户提供的条件

- 查询 54

用户提供的条件提示、准则和用法 62

用户提供的条件提示字符串 55

用户提供的条件选择性 54

## 优化程序

- 估计值 54
- 用户定义的选择性 54

## 游标

- SAP Sybase IQ 中的行级 779
- Transact-SQL 779
- 显示有关信息 413

有用性提示 59

余切值 159

## 语法

- CURRENT DATABASE 特殊值 62
- CURRENT PUBLISHER 特殊值 63
- CURRENT USER 特殊值 64
- LAST USER 特殊值 64
- SQL 标识符 24
- TIMESTAMP 特殊值 65
- USER 特殊值 65

## 语句

- CREATE DEFAULT 767
- CREATE DOMAIN 767
- CREATE RULE 767
- CREATE TABLE 764
- DISK INIT 757
- DISK MIRROR 757
- DISK REFIT 757
- DISK REINIT 757
- DISK REMIRROR 757
- DISK UNMIRROR 757
- RAISERROR 782, 783
- SELECT 770
- SIGNAL 782

## 语言

- 指定 8

## 预取

- 使用 sp\_iqsysmon 监控 518

## 域 364

- 关于 362

## 运算符

- 比较运算符 40
- 优先级 34

运算符的优先级 34

**Z**

杂项函数 132

- ARGN 134
- COALESCE 149
- IFNULL 216
- ISNULL 220

- NULLIF 255
- NUMBER 256
- ROWID 293
- SQLFLAGGER 307
- 摘要 387, 474–477
- 执行阶段提示 58
- 指数函数 194
- 中括号
  - SQL 标识符 24
  - 数据库对象 24
- 重命名对象
  - sp\_iqrename 过程 498
- 逐位运算符 32
- 主高速缓存
  - 验证 392
- 主键
  - UUID 和 GUID 247
  - 生成唯一值 247
  - 使用 UUID 生成唯一值 247
  - 显示有关信息 485
- 主数据库
  - 不支持 756
- 注册表条目
  - 关于 14
- 注释
  - 注释指示符 72
- 装载格式
  - Transact-SQL 和 SQL Anywhere 770
- 子查询
  - Adaptive Server 771
  - SAP Sybase IQ 771
  - SAP Sybase IQ 实现 372
  - SQL Anywhere 771
  - 表达式中 29
  - 分离 43
  - 搜索条件中的 43
- 子句
  - ON EXCEPTION RESUME 783
- 字符串 26, 28
  - SOUNDEX 函数 304
  - Transact-SQL 38
  - 并置 31, 288, 316
  - 并置运算符 31
  - 插入 217
  - 长度 141, 147
  - 常量 26, 28
  - 返回子字符串 319
  - 分隔符 38
  - 关于 26
  - 决定长度 230, 233
  - 确定相似性 298
  - 删除空白 326
  - 删除前导空白 238
  - 删除尾随空白 295
  - 特殊字符 26
  - 替换子字符串 286
  - 位置 148
  - 文字字符串 26
  - 转换为大写形式 327
  - 转换为小写 225, 237
- 字符串函数 88
  - ASCII 135
  - BFILE 139
  - BIT\_LENGTH 140
  - BYTE\_LENGTH 141
  - BYTE\_LENGTH64 142
  - BYTE\_SUBSTR 143
  - BYTE\_SUBSTR64 143
  - CHAR 146
  - CHAR\_LENGTH 147
  - CHAR\_LENGTH64 148
  - CHARINDEX 148
  - DIFFERENCE 188
  - INSERTSTR 217
  - LCASE 225
  - LEFT 228
  - LENGTH 230, 233
  - LOCATE 234
  - LOWER 237
  - LTRIM 238
  - OCTET\_LENGTH 258
  - PATINDEX 259
  - REPEAT 285
  - REPLACE 286
  - REPLICATE 288
  - RIGHT 290
  - RTRIM 295
  - SIMILAR 298
  - SORTKEY 300
  - SOUNDEX 304
  - SPACE 307
  - STR 314
  - STRING 316
  - STUFF 318
  - SUBSTR 319
  - SUBSTRING 319
  - SUBSTRING64 322
  - TRIM 326

- UCASE 327
- UPPER 328
- 字符串文字
  - 最大长度 375
- 字符集
  - 指定 7
- 字符数据
  - 兼容性 758

- 字节长度 258
- 字母字符
  - 定义 24
- 自动连接
  - 和外键 371
- 总体方差函数 332
- 组
  - Adaptive Server 768

