



参考：构件块、表和过程

Sybase IQ 15.4

文档 ID: DC01134-01-1540-01

最后修订日期: 2011 年 11 月

版权所有 © 2011 Sybase, Inc. 保留所有权利。

除非新版本或技术声明中另有说明, 否则本出版物适用于 Sybase 软件及所有后续版本。本文档中的信息如有更改, 恕不另行通知。本出版物中描述的软件按许可证协议提供, 其使用或复制必须符合协议条款。

要订购其它文档, 美国和加拿大的客户请拨打客户服务部门电话 (800) 685-8225 或发传真至 (617) 229-9845。

持有美国许可证协议的其它国家/地区的客户可通过上述传真号码与客户服务部门联系。所有其它国际客户请与 Sybase 子公司或当地分销商联系。仅在软件的定期发布日期提供升级内容。未经 Sybase, Inc. 的事先书面许可, 不得以任何形式、任何手段 (电子的、机械的、手工的、光学的或其它手段) 复制、传播或翻译本出版物的任何部分。

可在 <http://www.sybase.com/detail?id=1011207> 上的 Sybase 商标页中查看 Sybase 商标。Sybase 和列出的标记均是 Sybase, Inc. 的商标。® 表示已在美国注册。

SAP 和此处提及的其它 SAP 产品与服务及其各自的徽标是 SAP AG 在德国和世界各地其它几个国家/地区的商标或注册商标。

Java 和基于 Java 的所有标记都是 Sun Microsystems, Inc. 在美国和其它国家/地区的商标或注册商标。

Unicode 和 Unicode 徽标是 Unicode, Inc. 的注册商标。

本书中提到的所有其它公司和产品名均可能是与之相关的相应公司的商标。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568。

目录

读者	1
文件位置和安装设置	3
安装目录结构	3
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
IQDIR15 环境变量	7
IQPORT 环境变量	8
IQLANG 环境变量	8
IQLOGDIR15 环境变量	10
IQTIMEOUT 环境变量	10
IQTMP15 环境变量	11
JAVA_HOME 环境变量	11
LIBPATH 或 LD_LIBRARY_PATH 环境变量	12
PATH 环境变量	12
SQLCONNECT 环境变量	13
SYBASE 环境变量	13
\$SYBASE_JRE6_32、\$SYBASE_JRE6_64 和 \$SYBASE_JRE5_64 环境变量	13
SYBASE_OCS 环境变量	14
注册表条目	14
当前用户和本地计算机设置	14
注册表结构	15
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	20
字符串	21
表达式	22
表达式中的常量	23
表达式中的列名	23
表达式中的子查询	24
SQL 运算符	24
IF 表达式	27
CASE 表达式	28
表达式和常量的兼容性	29
搜索条件	31
比较条件	32
搜索条件中的子查询	34
ALL 或 ANY 条件	37
BETWEEN 条件	38
LIKE 条件	38
IN 条件	42
CONTAINS 条件	42
EXISTS 条件	43
IS DISTINCT FROM 和 IS NOT DISTINCT FROM 搜索条件	43
IS NULL 条件	44
带有逻辑运算符的条件	44
NOT 条件	45
真值条件	45
三值逻辑	45
用户提供的条件提示	46
特殊值	51
CURRENT DATABASE 特殊值	52
CURRENT DATE 特殊值	52
CURRENT PUBLISHER 特殊值	52
CURRENT TIME 特殊值	52
CURRENT TIMESTAMP 特殊值	53

CURRENT USER 特殊值	53
LAST USER 特殊值	54
SQLCODE 特殊值	54
SQLSTATE 特殊值	54
TIMESTAMP 特殊值	54
USER 特殊值	55
变量	55
局部变量	55
连接级变量	57
全局变量	57
注释	61
空值	62
SQL 数据类型	65
字符数据类型	65
存储大小	66
字符集和代码页	66
索引	67
VARCHAR 数据和尾随空白	67
对于超过 255 字节的 CHAR 和 VARCHAR 数据 的限制	68
字符数据兼容性	68
长字符串	68
数值数据类型	68
数值数据类型的用法	69
二进制数据类型	72
二进制数据类型的用法	73
位数据类型	77
位数据兼容性	78
日期和时间数据类型	78
日期和时间数据类型的用法	78
域	83
简单域	83
CREATE DOMAIN 语句	84
域兼容性	85

数据类型转换	86
字符串到日期时间转换的兼容性	86
导出的日期的兼容性	87
从 BIT 到 BINARY 数据类型的转换	87
BIT 和 CHAR/VARCHAR 数据类型之间的转换	88
SQL 函数	91
集合函数	91
分析函数	93
窗口化集合函数的用法	95
排名函数的用法	96
统计聚合分析函数的用法	97
分布函数的用法	97
行间函数的用法	98
数据类型转换函数	98
日期和时间函数	99
日期分量	102
HTTP 函数	103
数值函数	104
字符串函数	105
系统函数	107
连接属性	110
可用于服务器的属性	111
可用于每个数据库的属性	111
SQL 和 Java 用户定义的函数	112
SQL 中用户定义的函数	112
Java 中用户定义的函数	112
时序和预测函数	113
杂项函数	113
按字母顺序排列的函数列表	113
ABS 函数 [数值]	114
ACOS 函数 [数值]	114
ARGN 函数 [杂项]	115
ASCII 函数 [字符串]	116
ASIN 函数 [数值]	117

ATAN 函数 [数值]	118
ATAN2 函数 [数值]	118
AVG 函数 [集合]	119
BFILE 函数 [数据提取]	120
BIGINTTOHEX 函数 [数据类型转换]	121
BIT_LENGTH 函数 [字符串]	121
BYTE_LENGTH 函数 [字符串]	122
BYTE_LENGTH64 函数	124
BYTE_SUBSTR64 和 BYTE_SUBSTR 函数	124
CAST 函数 [数据类型转换]	124
CEIL 函数 [数值]	125
CEILING 函数 [数值]	126
CHAR 函数 [字符串]	127
CHAR_LENGTH 函数 [字符串]	128
CHAR_LENGTH64 函数	129
CHARINDEX 函数 [字符串]	129
COALESCE 函数 [杂项]	130
COL_LENGTH 函数 [系统]	131
COL_NAME 函数 [系统]	132
CONNECTION_PROPERTY 函数 [系统]	133
CONVERT 函数 [数据类型转换]	134
CORR 函数 [集合]	138
COS 函数 [数值]	139
COT 函数 [数值]	140
COVAR_POP 函数 [集合]	140
COVAR_SAMP 函数 [集合]	142
COUNT 函数 [集合]	143
CUME_DIST 函数 [分析]	144
DATALLENGTH 函数 [系统]	145
DATE 函数 [日期和时间]	146
DATEADD 函数 [日期和时间]	146
DATECEILING 函数 [日期和时间]	148
DATEDIFF 函数 [日期和时间]	150
DATEFLOOR 函数 [日期和时间]	152

DATEFORMAT 函数 [日期和时间]	155
DATENAME 函数 [日期和时间]	156
DATEPART 函数 [日期和时间]	157
DATEROUND 函数 [日期和时间]	158
DATETIME 函数 [日期和时间]	161
DAY 函数 [日期和时间]	161
DAYNAME 函数 [日期和时间]	162
DAYS 函数 [日期和时间]	163
DB_ID 函数 [系统]	164
DB_NAME 函数 [系统]	165
DB_PROPERTY 函数 [系统]	166
DEGREES 函数 [数值]	167
DENSE_RANK 函数 [分析]	167
DIFFERENCE 函数 [字符串]	169
DOW 函数 [日期和时间]	170
ERRORMSG 函数 [杂项]	170
EVENT_CONDITION 函数 [系统]	171
EVENT_CONDITION_NAME 函数 [系统]	173
EVENT_PARAMETER 函数 [系统]	173
EXP 函数 [数值]	175
EXP_WEIGHTED_AVG 函数 [集合]	175
FIRST_VALUE 函数 [集合]	177
FLOOR 函数 [数值]	179
GETDATE 函数 [日期和时间]	179
GRAPHICAL_PLAN 函数 [字符串]	180
GROUPING 函数 [集合]	182
GROUP_MEMBER 函数 [系统]	183
HEXTOBIGINT 函数 [数据类型转换]	184
HEXTOINT 函数 [数据类型转换]	185
HOUR 函数 [日期和时间]	187
HOURS 函数 [日期和时间]	188
HTML_DECODE 函数 [HTTP]	189
HTML_ENCODE 函数 [HTTP]	189
HTML_PLAN 函数 [字符串]	190

HTTP_DECODE 函数 [HTTP]	191
HTTP_ENCODE 函数 [HTTP]	192
HTTP_HEADER 函数 [HTTP]	192
HTTP_VARIABLE 函数 [HTTP]	192
IFNULL 函数 [杂项]	193
INDEX_COL 函数 [系统]	193
INSERTSTR 函数 [字符串]	194
INTTOHEX 函数 [数据类型转换]	195
ISDATE 函数 [日期和时间]	196
ISNULL 函数 [杂项]	197
ISNUMERIC 函数 [杂项]	198
LAG 函数 [分析]	199
LAST_VALUE 函数 [集合]	201
LCASE 函数 [字符串]	203
LEAD 函数 [分析]	204
LEFT 函数 [字符串]	205
LEN 函数 [字符串]	206
LENGTH 函数 [字符串]	207
LIST 函数 [集合]	209
LN 函数 [数值]	209
LOCATE 函数 [字符串]	210
LOG 函数 [数值]	211
LOG10 函数 [数值]	212
LOWER 函数 [字符串]	213
LTRIM 函数 [字符串]	214
MAX 函数 [集合]	214
MEDIAN 函数 [集合]	215
MIN 函数 [集合]	217
MINUTE 函数 [日期和时间]	217
MINUTES 函数 [日期和时间]	218
MOD 函数 [数值]	219
MONTH 函数 [日期和时间]	220
MONTHNAME 函数 [日期和时间]	221
MONTHS 函数 [日期和时间]	221

NEWID 函数 [杂项]	223
NEXT_CONNECTION 函数 [系统]	224
NEXT_DATABASE 函数 [系统]	226
NEXT_HTTP_HEADER 函数 [HTTP]	227
NEXT_HTTP_VARIABLE 函数 [HTTP]	227
NOW 函数 [日期和时间]	227
NTILE 函数 [分析]	228
NULLIF 函数 [杂项]	229
NUMBER 函数 [杂项]	231
OBJECT_ID 函数 [系统]	232
OBJECT_NAME 函数 [系统]	232
OCTET_LENGTH 函数 [字符串]	233
PATINDEX 函数 [字符串]	234
PERCENT_RANK 函数 [分析]	236
PERCENTILE_CONT 函数 [分析]	237
PERCENTILE_DISC 函数 [分析]	239
PI 函数 [数值]	241
POWER 函数 [数值]	241
PROPERTY 函数 [系统]	242
PROPERTY_DESCRIPTION 函数 [系统]	243
PROPERTY_NAME 函数 [系统]	243
PROPERTY_NUMBER 函数 [系统]	244
QUARTER 函数 [日期和时间]	245
RADIANS 函数 [数值]	246
RAND 函数 [数值]	246
RANK 函数 [分析]	247
REGR_AVGX 函数 [集合]	249
REGR_AVGY 函数 [集合]	250
REGR_COUNT 函数 [集合]	251
REGR_INTERCEPT 函数 [集合]	252
REGR_R2 函数 [集合]	253
REGR_SLOPE 函数 [集合]	254
REGR_SXX 函数 [集合]	256
REGR_SXY 函数 [集合]	257

REGR_SYY 函数 [集合]	258
REMAINDER 函数 [数值]	259
REPEAT 函数 [字符串]	260
REPLACE 函数 [字符串]	261
REPLICATE 函数 [字符串]	263
REVERSE 函数 [字符串]	264
RIGHT 函数 [字符串]	265
ROUND 函数 [数值]	266
ROW_NUMBER 函数 [分析]	267
ROWID 函数 [杂项]	268
RTRIM 函数 [字符串]	269
SECOND 函数 [日期和时间]	270
SECONDS 函数 [日期和时间]	271
SIGN 函数 [数值]	272
SIMILAR 函数 [字符串]	273
SIN 函数 [数值]	274
SORTKEY 函数 [字符串]	274
SOUNDEX 函数 [字符串]	278
SPACE 函数 [字符串]	279
SQLFLAGGER 函数 [杂项]	280
SQRT 函数 [数值]	281
SQUARE 函数 [数值]	281
STDDEV 函数 [集合]	282
STDDEV_POP 函数 [集合]	284
STDDEV_SAMP 函数 [集合]	285
STR 函数 [字符串]	286
STR_REPLACE 函数 [字符串]	287
STRING 函数 [字符串]	289
STRTOUUID 函数 [字符串]	290
STUFF 函数 [字符串]	291
SUBSTRING 函数 [字符串]	292
SUBSTRING64 函数 [字符串]	293
SUM 函数 [集合]	293
SUSER_ID 函数 [系统]	294

SUSER_NAME 函数 [系统]	295
TAN 函数 [数值]	296
TODAY 函数 [日期和时间]	297
TRIM 函数 [字符串]	297
TRUNCNUM 函数 [数值]	298
TS_ARMA_AR 函数 [时序]	299
TS_ARMA_CONST 函数 [时序]	299
TS_ARMA_MA 函数 [时序]	299
TS_AUTOCORRELATION 函数 [时序]	299
TS_AUTO_ARIMA 函数 [时序]	300
TS_AUTO_ARIMA_OUTLIER 函数 [时序]	300
TS_AUTO_ARIMA_RESULT_AIC 函数 [时序] ...	300
TS_AUTO_ARIMA_RESULT_AICC 函数 [时序]	301
TS_AUTO_ARIMA_RESULT_BIC 函数 [时序] ...	301
TS_AUTO_ARIMA_RESULT_FORECAST_VA LUE 函数 [时序]	301
TS_AUTO_ARIMA_RESULT_FORECAST_ER ROR 函数 [时序]	301
TS_AUTO_ARIMA_RESULT_MODEL_D 函数 [时序]	302
TS_AUTO_ARIMA_RESULT_MODEL_P 函数 [时序]	302
TS_AUTO_ARIMA_RESULT_MODEL_Q [时序]	302
TS_AUTO_ARIMA_RESULT_MODEL_S 函数 [时序]	302
TS_AUTO_ARIMA_RESULT_RESIDUAL_SIG MA [时序]	302
TS_AUTO_UNI_AR 函数 [时序]	303
TS_BOX_COX_XFORM 函数 [时序]	303
TS_DIFFERENCE 函数 [时序]	303
TS_DOUBLE_ARRAY [时序]	303
TS_ESTIMATE_MISSING 函数 [时序]	304

TS_GARCH 函数 [时序]	304
TS_GARCH_RESULT_A 函数 [时序]	304
TS_GARCH_RESULT_AIC 函数 [时序]	304
TS_GARCH_RESULT_USER [时序]	305
TS_INT_ARRAY [时序]	305
TS_LACK_OF_FIT 函数 [时序]	305
TS_LACK_OF_FIT_P 函数 [时序]	305
TS_MAX_ARMA_AR 函数 [时序]	306
TS_MAX_ARMA_CONST 函数 [时序]	306
TS_MAX_ARMA_LIKELIHOOD 函数 [时序]	306
TS_MAX_ARMA_MA 函数 [时序]	306
TS_OUTLIER_IDENTIFICATION 函数 [时序]	307
TS_PARTIAL_AUTOCORRELATION 函数 [时 序]	307
TS_VWAP 函数 [时序]	307
UCASE 函数 [字符串]	308
UPPER 函数 [字符串]	309
USER_ID 函数 [系统]	310
USER_NAME 函数 [系统]	310
UUIDTOSTR 函数 [字符串]	311
VAR_POP 函数 [集合]	312
VAR_SAMP 函数 [集合]	314
VARIANCE 函数 [集合]	315
WEEKS 函数 [日期和时间]	317
WEIGHTED_AVG 函数 [集合]	318
WIDTH_BUCKET 函数 [数值]	320
YEAR 函数 [日期和时间]	321
YEARS 函数 [日期和时间]	322
YMD 函数 [日期和时间]	324
与其它 SQL 术语的区别	325
日期	325
完整性	325
连接	326
更新	326

更改表	326
有时不允许使用子查询	326
附加函数	327
游标	327
物理限制	329
系统过程	333
存储过程的语法规则	333
了解存储过程报告的统计信息	334
系统存储过程	334
sa_char_terms 系统过程	335
sa_dependent_views 系统过程	335
sa_external_library_unload 过程	336
sa_list_external_library 过程	336
sa_nchar_terms 系统过程	336
sa_text_index_vocab 过程	336
sa_get_user_status 系统过程	337
sp_expireallpasswords 过程	338
sp_iqaddlogin 过程	338
sp_iqbackupdetails 过程	340
sp_iqbackupsummary 过程	342
sp_iqcardinality_analysis 过程	343
sp_iqcheckdb 过程	346
sp_iqcheckoptions 过程	352
sp_iqclient_lookup 过程	354
sp_iqcolumn 过程	356
sp_iqcolumnuse 过程	358
sp_iqconnection 过程	359
sp_iqconstraint 过程	362
sp_iqcontext 过程	364
sp_iqcopyloginpolicy 过程	366
sp_iqcursorinfo 过程	367
sp_iqdatatype 过程	369
sp_iqdbsize 过程	372
sp_iqdbspace 过程	373

sp_iqdbspaceinfo 过程	376
sp_iqdbspaceobjectinfo 过程	379
sp_iqdbstatistics 过程	383
sp_iqdroplogin 过程	384
sp_iqemptyfile 过程	385
sp_iqestjoin 过程	386
sp_iqestdbspaces 过程	387
sp_iqestspace 过程	389
sp_iqevent 过程	390
sp_iqfile 过程	393
sp_iqhelp 过程	395
sp_iqindex 和 sp_iqindex_alt 过程	401
sp_iqindexadvice 过程	404
sp_iqindexfragmentation 过程	406
sp_iqindexinfo 过程	407
sp_iqindexmetadata 过程	409
sp_iqindexsize 过程	410
sp_iqindexuse 过程	413
sp_iqjoinindex 过程	415
sp_iqjoinindexsize 过程	418
sp_iqlmconfig 过程	419
sp_iqllocks 过程	421
sp_iqmodifyadmin 过程	423
sp_iqmodifylogin 过程	424
sp_iqmpxcheckdqpconfig 过程	425
sp_iqmpxfilestatus 过程	427
sp_iqmpxinconnpoolinfo 过程	427
sp_iqmpxinheartbeatinfo 过程	427
sp_iqmpxinfo 过程	427
sp_iqmpxvalidate 过程	427
sp_iqmpxversioninfo 过程	427
sp_iqobjectinfo 过程	428
sp_iqpassword 过程	430
sp_iqpkeys 过程	431

sp_iqprocedure 过程	434
sp_iqprocparm 过程	436
sp_iqrebuildindex 过程	439
sp_iqrename 过程	441
sp_iq_reset_identity 过程	443
sp_iqrestoreaction 过程	444
sp_iqrowdensity 过程	445
sp_iqsharedtempdistrib 过程	446
sp_iqshowpsexec 过程	446
sp_iqspaceinfo 过程	448
sp_iqspaceused 过程	449
sp_iqstatistics 过程	451
sp_iqstatus 过程	453
sp_iqsysmon 过程	455
sp_iqtable 过程	460
sp_iqtablesize 过程	464
sp_iqtableuse 过程	465
sp_iqtransaction 过程	466
sp_iqunusedcolumn 过程	469
sp_iqunusedindex 过程	470
sp_iqunusedtable 过程	472
sp_iqversionuse 过程	473
sp_iqview 过程	474
sp_iqwho 过程	476
sp_iqworkmon 过程	479
目录存储过程	481
sa_ansi_standard_packages 系统过程	481
sa_audit_string 系统过程	482
sa_checkpoint_execute 系统过程	482
sa_conn_activity 系统过程	483
sa_conn_info 系统过程	485
sa_conn_list 系统过程	487
sa_conn_properties 系统过程	488
sa_db_info 系统过程	488

sa_db_properties 系统过程	489
sa_disable_auditing_type 系统过程	490
sa_disk_free_space 系统过程	491
sa_enable_auditing_type 系统过程	492
sa_eng_properties 系统过程	493
sa_flush_cache 系统过程	494
sa_make_object 系统过程	495
sa_rowgenerator 系统过程	496
sa_server_option 系统过程	498
sa_set_http_header 系统过程	506
sa_set_http_option 系统过程	507
sa_table_page_usage 系统过程	511
sa_validate 系统过程	511
sa_verify_password 系统过程	512
sp_login_environment 系统过程	513
sp_remote_columns 系统过程	513
sp_remote_exported_keys 系统过程	514
sp_remote_imported_keys 系统过程	515
sp_remote_primary_keys 系统过程	517
sp_remote_tables 系统过程	517
sp_servercaps 系统过程	518
sp_tsql_environment 系统过程	519
Adaptive Server Enterprise 系统和 Catalog 存储过程	520
Adaptive Server Enterprise 系统过程	521
Adaptive Server Enterprise Catalog 存储过程	522
SQL Anywhere 支持的过程	523
系统表和视图	525
系统表	525
SYS.DUMMY 表与 IQ_DUMMY 表	528
系统视图	529
统一视图	529
兼容性视图	529

ASE T-SQL 兼容性视图	530
SYSARTICLE 系统视图	530
SYSARTICLECOL 系统视图	530
SYSARTICLECOLS 统一视图	531
SYSARTICLES 统一视图	531
SYSCAPABILITIES 统一视图	531
SYSCAPABILITY 系统视图	532
SYSCAPABILITYNAME 系统视图	532
SYSCATALOG 统一视图	532
SYSCHECK 系统视图	533
SYSCOLAUTH 统一视图	533
SYSCOLLATION 兼容性视图 (不建议使用)	
.....	534
SYSCOLLATIONMAPPINGS 兼容性视图 (不建 议使用)	534
SYSCOLPERM 系统视图	534
SYSCOLSTAT 系统视图	535
SYSCOLSTATS 统一视图	535
SYSCOLUMN 兼容性视图 (不建议使用)	536
SYSCOLUMNS 统一视图	536
SYSCOLUMNS ASE 兼容性视图	537
SYSCOMMENTS ASE 兼容性视图	537
SYSCONSTRAINT 系统视图	538
SYSDBFIL 系统视图	538
SYSDBSpace 系统视图	539
SYSDBSpacePERM 系统视图	539
SYSDependency 系统视图	540
SYSDomain 系统视图	540
SYSEvent 系统视图	540
SYSEVENTTYPE 系统视图	541
SYSEXTERNENV 系统视图	542
SYSEXTERNENVOBJECT 系统视图	542
SYSEXTERNLOGIN 系统视图	543
SYSDFile 兼容性视图 (不建议使用)	543

SYSFKCOL 兼容性视图（不建议使用）	544
SYSFKEY 系统视图	544
SYSFOREIGNKEY 兼容性视图（不建议使用）	545
SYSFOREIGNKEYS 统一视图	546
SYSGROUP 系统视图	546
SYSGROUPS 统一视图	547
SYSHISTORY 系统视图	548
SYSIDX 系统视图	549
SYSIDXCOL 系统视图	550
SYSINDEX 兼容性视图（不建议使用）	551
SYSINDEXES 统一视图	551
SYSINDEXES ASE 兼容性视图	552
SYSINFO 兼容性视图（不建议使用）	552
SYSIQBACKUPHISTORY 系统视图	553
SYSIQBACKUPHISTORYDETAIL 系统视图	553
SYSIQCOLUMN 系统视图（不建议使用）	554
SYSIQDBFILE 系统视图	554
SYSIQDBSPACE 系统视图	555
SYSIQFILE 系统视图（不建议使用）	556
SYSIQIDX 系统视图	556
SYSIQINFO 系统视图	557
SYSIQJOINIDX 系统视图	558
SYSIQJOININDEX 系统视图（不建议使用）	559
SYSIQJOINIXCOLUMN 系统视图	559
SYSIQJOINIXTABLE 系统视图	560
SYSIQLOGICALSERVER 系统视图	560
SYSIQLOGINPOLICYLSINFO 系统视图	560
SYSIQLSLOGINPOLICYOPTION 系统视图	561
SYSIQLSMEMBER 系统视图	561
SYSIQLSMEMBERS 统一视图	561
SYSIQLSLOGINPOLICIES 统一视图	561
SYSIQLSPOLICY 系统视图	561
SYSIQLSPOLICYOPTION 系统视图	561

SYSIQMPXSERVER 系统视图	561
SYSIQOBJECTS ASE 兼容性视图	562
SYSIQPARTITIONCOLUMN 系统视图	562
SYSIQTAB 系统视图	563
SYSIQTABCOL 系统视图	563
SYSIQTABLE 系统视图（不建议使用）	564
SYSIQVINDEXT ASE 兼容性视图	564
SYSIXCOL 兼容性视图（不建议使用）	565
SYSJAR 系统视图	565
SYSJARCOMPONENT 系统视图	566
SYSJAVACLASS 系统视图	566
SYSLOGINMAP 系统视图	567
SYSLOGINPOLICY 系统视图	567
SYSLOGINPOLICYOPTION 系统视图	567
SYSLOGINS ASE 兼容性视图	568
SYSMVOPTION 系统视图	568
SYSMVOPTIONNAME 系统视图	568
SYSOBJECT 系统视图	569
SYSOBJECTS ASE 兼容性视图	569
SYSOPTION 系统视图	570
SYSOPTIONS 统一视图	570
SYSOPTSTAT 系统视图	570
SYSPARTITION 系统视图	571
SYSPARTITIONKEY 系统视图	571
SYSPARTITIONSCHEME 系统视图	572
SYSPHYSIDX 系统视图	572
SYSPROCAUTH 统一视图	573
SYSPROCEDURE 系统视图	573
SYSPROCPRM 系统视图	574
SYSPROCPRMS 统一视图	575
SYSPROCPRM 系统视图	576
SYSPROCS 统一视图	576
SYSPROXYTAB 系统视图	577
SYSPUBLICATION 系统视图	577

SYSPUBLICATIONS 统一视图	578
SYSREMARK 系统视图	579
SYSREMOTEOPTION 系统视图	579
SYSREMOTEOPTION2 统一视图	579
SYSREMOTEOPTIONS 统一视图	580
SYSREMOTEOPTIONTYPE 系统视图	580
SYSREMOTETYPE 系统视图	580
SYSREMOTETYPES 统一视图	581
SYSREMOTEUSER 系统视图	581
SYSREMOTEUSERS 统一视图	582
SYSSCHEDULE 系统视图	583
SYSSERVER 系统视图	583
SYSSOURCE 系统视图	584
SYSSQLSERVERTYPE 系统视图	584
SYSSUBPARTITIONKEY 系统视图	584
SYSSUBSCRIPTION 系统视图	584
SYSSUBSCRIPTIONS 统一视图	585
SYSSYNC 系统视图	585
SYSSYNC2 统一视图	586
SYSSYNCPUBLICATIONDEFAULTS 统一视图	587
SYSSYNCS 统一视图	587
SYSSYNCSCRIPT 系统视图	587
SYSSYNCSCRIPTS 统一视图	588
SYSSYNCSUBSCRIPTIONS 统一视图	588
SYSSYNCUSERS 统一视图	589
SYSTAB 系统视图	589
SYSTABAUTH 统一视图	591
SYSTABCOL 系统视图	592
SYSTABLE 兼容性视图（不建议使用）	593
SYSTABLEPERM 系统视图	594
SYSTEXTCONFIG 系统视图	595
SYSTEXTIDX 系统视图	597
SYSTEXTIDXTAB 系统视图	598

SYSTRIGGER 系统视图	598
SYSTRIGGERS 统一视图	600
SYSTYPEMAP 系统视图	600
SYSTYPES ASE 兼容性视图	601
SYSUSER 系统视图	601
SYSUSERAUTH 兼容性视图（不建议使用）	602
SYSUSERAUTHORITY 系统视图	602
SYSUSERLIST 兼容性视图（不建议使用）	603
SYSUSERMESSAGE 系统视图	603
SYSUSEROPTIONS 统一视图	603
SYSUSERPERM 兼容性视图（不建议使用）	604
SYSUSERPERMS 兼容性视图（不建议使用）	604
SYSUSERTYPE 系统视图	605
SYSUSERS ASE 兼容性视图	606
SYSVIEW 系统视图	606
SYSVIEWS 统一视图	606
SYSWEBSERVICE 系统视图	607
Transact-SQL 兼容性视图	608
与其它 Sybase 数据库的兼容性	613
关于 SQL Anywhere	613
Transact-SQL 支持概述	613
Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 体系结构	614
服务器和数据库	614
空间分配和设备管理	615
系统表、Catalog 存储和 IQ 存储	615
管理角色	616
数据类型	617
位数据类型	617
字符数据类型	617
二进制数据类型	618
日期、时间、日期时间和时间戳数据类型	619

数值数据类型	620
文本数据类型	620
图像数据类型	621
Java 数据类型	621
Data Definition Language (数据定义语言)	621
从 Sybase Central 创建与 Transact-SQL 兼容的 数据库	621
使用 CREATE DATABASE 语句创建与 Transact- SQL 兼容的数据库	621
区分大小写	622
确保兼容的对象名	622
使用 CREATE TABLE 语句时的注意事项	623
使用 CREATE DEFAULT、CREATE RULE 和 CREATE DOMAIN 语句时的注意事项	625
使用 CREATE TRIGGER 语句时的注意事项	625
使用 CREATE INDEX 语句时的注意事项	626
用户、组和权限	626
装载格式	628
Transact-SQL 兼容性选项	628
Data Manipulation Language (数据操纵语言)	628
编写可移植 SQL 的一般原则	628
编写兼容查询的标准	629
子查询支持	630
GROUP BY 子句支持	630
COMPUTE 子句支持	630
WHERE 子句支持	631
Transact-SQL 外连接支持	631
ANSI 连接支持	631
空值比较支持	631
零长度字符串支持	632
HOLDLOCK、SHARED 和 FOR BROWSE 支持	632
SQL 函数支持	632
OLAP 函数支持	633

系统函数支持	634
用户定义的函数支持	634
解释日期的算术表达式的区别	635
SELECT INTO 语句支持	635
可更新的视图支持	635
支持 UPDATE 和 DELETE 中的 FROM 子句	635
Transact-SQL 过程语言概述	636
Transact-SQL 存储过程概述	636
Transact-SQL 批处理概述	636
过程和批处理中的 SQL 语句	636
存储过程的自动转换	638
Transact-SQL 过程的结果集	638
Transact-SQL 过程中的变量	639
Transact-SQL 过程中的错误处理	640
在过程中使用 RAISERROR 语句	641
Watcom-SQL 术语中与 Transact-SQL 类似的错误处理	641
SQL Anywhere 和 Sybase IQ 的区别 和 共享的功能	642
SQL Anywhere 服务器和数据库启动与管理	642
SQL Anywhere 数据库选项	642
SQL Anywhere 数据定义语言 (DDL) 区别	643
SQL Anywhere 数据操作语言 (DML) 区别	643
Adaptive Server Enterprise 和 Sybase IQ 的区别和共享的功能	644
Adaptive Server Enterprise 存储过程	644
Adaptive Server Enterprise 系统视图	644
索引	647

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 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》。

文件位置和安装设置

以下主题描述了 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 Sybase IQ 时，可能创建多个目录。本节说明了目录的结构。

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，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。Sybase IQ 附带的其它工具在 Sybase 目录 下面具有唯一子目录。本节仅描述 Sybase IQ 的子目录 结构。

缺省情况下，Sybase IQ 目录是 `IQ-15_3`。`IQ-15_3` 的位置因 Sybase IQ 的 安装位置而异。`IQ-15_3` 目录在 UNIX 上还被环境变量 `$IQDIR15` 引用，而在 Windows 上则被 `%IQDIR15%` 引用。

Sybase IQ 目录下有很多目录和 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 文件为 `iqdemo.db`、`iqdemo.iq`、`iqdemo.iqmsg` 和 `iqdemo.iqtmp`。演示数据库本身并不随附在 Sybase IQ 中。
- **子目录 /demo/adata** – 包含可用于创建 15.x iqdemo 数据库的 15.x 数据。子目录 `/demo/demodata` 包含 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/mkqiqdemo.bat` 或 UNIX 上的 `demo/mkqiqdemo.sh` 可创建 15.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-15_3/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 SQL Anywhere® 的 C++ 示例。由于 SQL Anywhere 和 Sybase IQ 共享公用代

码, 因此您可以修改这些示例以便用于 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-15_3 子目录 /bin64、/lib64、/logfiles、/res 和 /tix。在 Windows 上, 这些子目录包括 IQ-15_3 子目录 \h、\install、\java 和 \bin32。

Sybase IQ 如何定位文件

Sybase IQ 必须查找并访问几种类型的文件, 才能启动和运行。系统上可能驻留有一些同名的目录或文件。

为了确保使用正确文件, 了解 Sybase IQ 如何查找这些文件 是很重要的。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。这些文件是运行 Sybase IQ 所必需的。如果找 到不正确的 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 \$IQDIR15/lib64 或 \$SYBASE/\$SYBASE_OCS/lib64 中, 而在 Windows 上则可能位于 %IQDIR15%\bin32 或 %SYBASE%\SYBASE_OCS\dll 中。空目录 \$IQDIR15/usrlib 用于 以自定义库和修补程序取代缺省库, 因为 start_iq 所具有的 usrlib 位于 常规库目录之前。Sybase IQ 同时使用 Adaptive Server Enterprise 和 SQL Anywhere 库。如果系统中已安装了这些产品中任意一种产品, 则应注意它们的安装目录, 以避免混淆。
- **Interface 文件** - 为运行 Sybase IQ 所必需的。例如, UNIX 上的 .odbc.ini 和 utility_db.ini, 以及 Windows 上的 util_db.ini。有关这些文件的详细信息, 请参见《系统管理指南第一卷》和《安装和配置指南》。
- **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** - 存储有关服务器和已连接的数据库上当前会话的信息。例如, 服务器日志可能命名为 %ALLUSERSPROFILE%/SybaseIQ/IQ15_3/logfiles/yourservername.0006.srvlog。数据库日志 (例如 %ALLUSERSPROFILE%/SybaseIQ/IQ-15_3/demo/iqdemo.log) 是在您连接到数据库时创建的。有关这些文件的详细信息, 请参见《安装和配置指南》。
- **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** - 由 Sybase IQ 创建, 用于存储对查询执行排序这样的操作的临时信息。

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，Sybase IQ 使用简单算法来定位文件。在其它情况下，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

Sybase IQ 程序（包括数据库服务器和管理实用程序）对必需文件（例如 DLL 或共享库）执行广泛搜索。在这些情况中，Sybase IQ 程序按以下顺序查找文件：

1. *可执行文件目录* - 包含程序可执行文件的目录。以及，具有相对于程序可执行文件目录的以下路径的目录：
 - 可执行文件目录的父目录。
 - 名为 **scripts** 的父目录的子目录。
2. *当前工作目录* - 程序启动时，它具有当前工作目录（从中启动程序的目录）。在此目录中搜索所需的文件。
3. *位置注册表条目* - 在 Windows 安装中，Sybase IQ 添加一条 **LOCATION** 注册表条目。先搜索指明的目录，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）
4. *特定于系统的目录* - 这包括包含公用操作系统文件的目录，例如 Windows 上的 Windows 目录和 Windows\system 目录。
5. *CLASSPATH 目录* - 对于 Java 文件，系统将搜索 CLASSPATH 环境变量中列出的目录，以查找文件。
6. *PATH 目录* - 系统将搜索系统路径和用户路径中的目录，以查找文件。
7. *LIBRARY PATH 目录* - 系统将搜索 LIBPATH 环境变量中列出的目录，以查找共享库。

环境变量

Sybase IQ 使用环境变量来存储各种信息类型；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-15_3/IQ-15_3.sh
```
2. 对于 C shell:

```
source $SYBASE/IQ-15_3/IQ-15_3.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
```

IQDIR15 环境变量

IQDIR15 用于标识 Sybase IQ 目录的位置，以及该目录下的其它目录和文件的位置。

- **\$IQDIR15/bin[64]/util_db.ini** 包含实用程序数据库 **utility_db** 的登录 ID 和口令。安装程序允许您将这些值从其缺省值（登录 ID “DBA” 和口令 “sql”）更改为其它值。
- **\$IQDIR15/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。通过设置 **IQLOGDIR15** 环境变量，可以覆盖此缺省值。
- **\$IQDIR15/demo** 是 **iqdemo** 数据库文件所在的位置。

设置

```
IQDIR15 = ${SYBASE}/IQ-15_3
```

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。

IQPORT 环境变量

IQPORT 覆盖 Sybase IQ 代理 端口号的缺省值，此数字用于 Sybase IQ 插件 和代理之间的通信。

注意：一旦代理启动，则不能更改端口值。

在任何给定的端口上搜索代理进程时，1099 都是插件缺省值。如果插件没有在此端口上找到代理，则它会显示提示，以便您指定正确的端口值。

设置

```
IQPORT = 5556
```

操作系统

可选。如果用户未和环境源文件中指定 IQPORT，则端口号缺省为 1099。可以在插件启动之前更改此缺省值。可以通过在启动 Sybase Central 时向 **scjview** 提供 **-DIQPORT** 参数来设置此变量。例如：

```
scjview -DIQPORT=3345
```

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
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	俄语	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

语言	ISO_639 语言代码	语言标签	替代标签
土耳其语	TR	turkish	不适用
乌克兰语	UK	ukrainian	不适用

IQLOGDIR15 环境变量

IQLOGDIR15 环境变量定义各种日志文件的位置。IQLOGDIR15 不是由安装程序设置的。

- 服务器日志位于 \$IQLOGDIR15 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 IQLOGDIR15 设置为有效和可写的目录，则大多数实用程序（包括 `start_iq`）将使用用于所有服务器日志的缺省位置 `$IQDIR15/logfiles`。

设置

```
IQLOGDIR15 = path
```

操作系统

可选。

IQTIMEOUT 环境变量

IQTIMEOUT 覆盖缺省等待时间（五分钟）。

参数 *nnn* 是 Sybase IQ 代理要等待的 分钟数。例如：

- 等待 45 分钟（Korn 或 Bourne shell）：

```
IQTIMEOUT=45 export IQTIMEOUT
```

- 等待一小时 (C shell)：

```
setenv IQTIMEOUT 60
```

Sybase IQ 代理会无限期地等待进程完成。建议您在为具有非常大的 Catalog 存储的 Multiplex 创建或同步 查询服务器时设置等待时间。对于大型 Catalog 存储，同步过程中的数据库备份部分所需时间会有所延长，而增加等待时间可满足较大规模的同步。

注意： 请在启动 Sybase IQ 代理之前设置 IQTIMEOUT。

设置

```
IQTIMEOUT = nnn
```

操作系统

可选，但在 Multiplex 环境中推荐使用。请参见《使用 Sybase IQ Multiplex》。

IQTMP15 环境变量

IQTMP15 环境变量不是由安装程序 设置的。IQTMP15 由 Sybase IQ 用于指示 保存临时文件的目录。

IQTMP15 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，这将允许 IQTMP15 目录清除当客户端连接关闭时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。这些目录和文件仅在连接期间需要用到。必须使连接到服务器的所有用户对该目录具有写入权限。

注意：由 IQTMP15 定义其位置的临时文件由客户端和服务器使用。此变量不控制 IQ 临时存储的缺省位置。**CREATE DATABASE** 语句控制 IQ 临时存储的缺省位置。请参见《参考：语句和选项》。

警告！ 不要将 IQTMP15 设置为 \$SYBASE 或 \$IQDIR15。

如果未显式设置 IQTMP15，则 Sybase IQ 代理会将 IQTMP15 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 Sybase IQ 将使用共享的内存连接，而不是网络连接。

若要避免在使用共享内存时发生冲突：

- 创建每台服务器专用的临时目录。通过在两种环境中都显式设置 IQTMP15 环境变量，确保每个本地客户端都使用与其服务器相同的临时目录。
- 在 .odbc.ini 文件（在 UNIX 上）中为每台服务器创建一个数据源名称，并提供详细的连接信息。请参见《安装和配置指南》。
- 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

```
IQTMP15 = temp_directory
```

操作系统

在 UNIX 上可选。在 Windows 平台上不使用。

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 通常是在安装 VM 时创建的。

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 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 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：在 UNIX 系统上，在 SQLCONNECT 中而不是在命令行上指定连接参数可以提供更好的安全性。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，此方法尤为有用。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。因为口令采用明文形式，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《系统管理指南：第一卷》>“连接和通信参数”>“连接参数”。

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。

必须在 UNIX 系统上安装 Sybase IQ 之前设置 SYBASE 变量。此变量是在 UNIX 系统上使用 Sybase Central 所必需的。

设置

```
SYBASE = path
```

操作系统

必需。

\$SYBASE_JRE6_32、\$SYBASE_JRE6_64 和 \$SYBASE_JRE5_64 环境变量

此变量指定 Sybase IQ 的 Sybase Central 插件所使用的 Java 运行时环境的位置。

对于 Windows 和 UNIX，该环境变量为 \$SYBASE_JRE6_32 或 \$SYBASE_JRE6_64。

对于 AIX/Linux/IBM，该变量为 \$SYBASE_JRE5_64。

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定位适当的 JRE。`JAVA_HOME` 具有优先权。在 Windows 上，安装程序会在安装 Open Client 软件开发工具包时设置该变量。

设置

```
SYBASE_JRE= "${SYBASE}/shared/jre-6_0"
```

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-15_3"
```

*操作系统
必需。*

注册表条目

在 Windows 操作系统上，Sybase IQ 使用若干个注册表 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！ Sybase 建议 *不要* 修改注册表，因为不正确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。

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。

Sybase IQ 安装程序允许您选择其所进行的设置是仅针对当前用户的，还是属于本地计算机级别的。

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 Sybase IQ 程序作为服务运行，则应确保在本地计算机级别进行设置。

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，Sybase 建议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。

Sybase IQ 注册表条目保存在 HKEY_LOCAL_MACHINE 项的下述位置中：

```
SOFTWARE
  Sybase
    IQ 15.4
```

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 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-15_4
```

Sybase IQ 注册表包括已安装应用程序的其它条目。Sybase Central 注册表包含有关 Sybase Central 版本和已安装的插件的信息。

SQL 语言元素

以下主题提供 Sybase IQ SQL 的语言 元素和约定的详细说明。

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 Sybase IQ 文档中，关键字都以大写字母显示。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 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。与这些保留字只在大小写形式上有所不同的所有字符串都是保留字。

表 1. SQL 保留字

add	all	alter	和
any	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	压缩	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	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	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	选项	options
或	order	others	out
outer	over	passthrough	precision
prepare	primary	print	privileges
proc	procedure	publication	raiserror
readtext	real	reference	references
刷新	release	remote	remove
rename	reorganize	resource	restore
restrict	return	revoke	right
rollback	rollup	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	unsigned
update	updating	user	using
validate	values	varbinary	varbit
varchar	variable	varying	view
wait	waitfor	when	where
while	window	with	within
work	writetext	xml	

另请参见

- 标识符（第 20 页）
- quoted_identifier 选项（第 30 页）
- 搜索条件中的子查询（第 34 页）
- 表达式中的列名（第 23 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。

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，您都可以用中括号来分隔标识符。

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。

有关服务器名和数据库名称的其它限制，请参见《实用程序指南》中的 `-n start_iq` 服务器选项。

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 30 页)
- 搜索条件中的子查询 (第 34 页)
- 表达式中的列名 (第 23 页)

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 32 页)
- 表达式 (第 22 页)
- 空值 (第 62 页)
- 搜索条件 (第 31 页)
- 三值逻辑 (第 45 页)

- SQL 运算符（第 24 页）
- 搜索条件中的子查询（第 34 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 Enterprise 不支持 IF 条件。
- Adaptive Server Enterprise 目前不支持 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 32 页)
- 空值 (第 62 页)
- 搜索条件 (第 31 页)
- 字符串 (第 21 页)
- 三值逻辑 (第 45 页)
- SQL 运算符 (第 24 页)
- 搜索条件中的子查询 (第 34 页)
- 特殊值 (第 51 页)
- CASE 语句支持 (第 637 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。

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

- 搜索条件中的子查询（第 34 页）
- 保留字（第 17 页）
- 标识符（第 20 页）

表达式中的子查询

子查询是用括号括起来的 **SELECT** 语句。该 **SELECT** 语句能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零个或一个值；

在顶级 **SELECT** 的 **SELECT** 列表或 **UPDATE** 语句的 **SET** 子句中，可以在能够使用列名的任意位置使用标量子查询。但是，子查询不能在条件表达式（**CASE**、**IF**、**NULLIF**、**ARGN**）中出现。

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;
```

SQL 运算符

以下主题描述了 Sybase IQ 中提供的算术运算符、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

- 比较条件（第 32 页）
- 表达式（第 22 页）
- 空值（第 62 页）
- 搜索条件（第 31 页）
- 字符串（第 21 页）
- 三值逻辑（第 45 页）

算术运算符

Sybase IQ 中提供了以下算术运算符。

表 2. 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。

运算符	描述
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。
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。

字符串运算符

Sybase IQ 中提供了以下字符串运算符。

表 3. 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。

字符串并置运算符的结果数据类型为 **LONG VARCHAR**。如果在 **SELECT INTO** 语句中使用字符串并置运算符，您必须具有非结构化数据分析选件许可证，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 Enterprise 支持 + 运算符。

另请参见

- REVERSE 函数 [字符串]（第 264 页）

逐位运算符

在 Sybase IQ 和 Adaptive Server Enterprise 中，可以对所有无增减的整数 数据类型使用以下逐位运算符。

运算符	描述
&	AND
	OR

运算符	描述
$\wedge$	异或
$\sim$	NOT

AND 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

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
1	0	1
1	1	1

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

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

位	~ 位
0	1

连接运算符

除了在 **FROM** 子句中使用表表达式的 ISO/ANSI SQL 连接语法，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**。
- 字符串并置 - 在 Sybase IQ 中使用 **+** 并置运算符时，需确保操作数已显式设置为字符串，而不是依赖隐式数据转换。例如，以下查询返回整数 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，**+** 运算符表示并置，而不是加法。

Adaptive Server Enterprise 不支持 **||** 并置运算符。

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，Sybase 建议您使用括号明确运算顺序，而不是依赖 Adaptive Server Enterprise 与 Sybase IQ 之间相同的运算符优先级。

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。

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 "GROUP0".Products
```

另请参见

- **NULLIF** 函数 [杂项]（第 229 页）
- 用于简写 **CASE** 表达式的 **NULLIF** 函数（第 29 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 28 页）
- NULLIF 函数 [杂项]（第 229 页）

表达式和常量的兼容性

以下主题介绍了 Adaptive Server Enterprise 和 Sybase IQ 之间的表达式和 常量的兼容性。

表达式的兼容性

下表介绍了 Adaptive Server Enterprise 和 Sybase IQ 之间的表达式兼容性。

下表只是一个指南，而且，“同时适用”的标记并不表示在任何情况下出于任何目的，表达式均以相同的方式执行。有关详细说明，请参见 Adaptive Server Enterprise 文档和 Sybase IQ 文档中各个表达式 的相关内容。

在下表中，**expr** 表示表达式，**op** 表示运算符。

表 4. Adaptive Server Enterprise 与 Sybase IQ 之间的表达式 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 Sybase IQ

常量的兼容性

下表介绍了 Adaptive Server Enterprise 和 Sybase IQ 之间的常量兼容性。

下表只是一个指南，而且，“同时适用”的标记并不表示在任何情况下出于任何目的，表达式均以相同的方式执行。有关详细说明，请参见 Adaptive Server Enterprise 文档和 Sybase IQ 文档中各个表达式的相关内容。

表 5. Adaptive Server Enterprise 与 Sybase IQ 之间的常量 兼容性

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

分隔字符串的缺省解释

缺省情况下，Adaptive Server Enterprise 和 Sybase IQ 为分隔字符串赋予不同的含义：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。

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 Enterprise 则遵循如下约定：用引号引起来的字符串是常量，而在缺省情况下不允许使用分隔标识符，并将其按字符串处理。

quoted_identifier 选项

Adaptive Server Enterprise 和 Sybase IQ 都提供了一个 **quoted_identifier** 选项，通过该选项，可以更改对分隔字符串的解释。缺省情况下，**quoted_identifier** 选项在 Adaptive Server Enterprise 中设置为 OFF，在 Sybase IQ 中设置为 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 Enterprise 连接选项不支持 Transact-SQL **SET** 语句，但 **quoted_identifier** 选项支持 **SET**。

Sybase IQ 或 Adaptive Server Enterprise 中的以下语句将 **quoted_identifier** 选项的设置更改为 ON：

```
SET quoted_identifier ON
```

quoted_identifier 选项设置为 ON 时，Adaptive Server Enterprise 允许用引号分隔表、视图和列的名称。在 Adaptive Server Enterprise 中不能分隔 其它对象名。

Sybase IQ 或 Adaptive Server Enterprise 中的以下语句将 **quoted_identifier** 选项的设置更改为 OFF：

```
SET quoted_identifier OFF
```

在 Adaptive Server Enterprise 和 Sybase IQ 中，只要各个 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（缺省的 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" (
    col1 char(5)
);
INSERT "user" ( col1 )
VALUES ( 'abcde' );
```

如果在 **quoted_identifier** 选项为 OFF（缺省的 Adaptive Server Enterprise 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 20 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 ] DISTINCT FROM
| expression IS [ NOT ] NULL expression
| 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 32 页）
- 表达式（第 22 页）
- 空值（第 62 页）
- 字符串（第 21 页）
- 三值逻辑（第 45 页）
- SQL 运算符（第 24 页）
- 搜索条件中的子查询（第 34 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 Sybase IQ 中提供的比较运算符。

运算符	描述
=	等于
>	大于
<	小于
>=	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：

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

兼容性

- 尾随空白 - Adaptive Server Enterprise 在进行比较时会忽略字符数据中的任何尾随空白。Sybase IQ 在比较字符串时的行为由“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，Sybase IQ 数据库与 Adaptive Server Enterprise 数据库一样，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 Sybase IQ 数据库的大小写区分。

另请参见

- 表达式 (第 22 页)
- 空值 (第 62 页)
- 搜索条件 (第 31 页)
- 字符串 (第 21 页)

- 三值逻辑（第 45 页）
- SQL 运算符（第 24 页）
- 搜索条件中的子查询（第 34 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。

Sybase IQ 只支持在非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。

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_linenum = @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 23 页）
- 保留字（第 17 页）
- 标识符（第 20 页）
- 比较条件（第 32 页）
- 表达式（第 22 页）
- 空值（第 62 页）
- 搜索条件（第 31 页）
- 字符串（第 21 页）
- 三值逻辑（第 45 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。Sybase IQ 支持这些子查询，这些子查询可以是相关的（包含

对在外部查询中出现且不能单独计算的表的引用），也可以是非相关的（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。请参见《参考：语句和选项》。

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 Enterprise 和 Sybase IQ 之间是兼容的。只有 Sybase IQ 支持将 **SOME** 用作 **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 Sybase IQ 和 Adaptive Server Enterprise 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。

如果有 **WD** 索引，则某些 **LIKE** 谓词会执行得更快。

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

所有其它字符都必须完全匹配。

例如，搜索条件：

```
name LIKE 'a%b_'
```

对于名称以字母 **a** 开头并以字母 **b** 作为倒数第二个字符的所有行，为 **TRUE**。

如果指定一个 *escape-expr*，它的值必须为单个字符。字符可以位于百分号、下划线、左中括号或 *pattern* 中的其它转义字符之前，以防止特殊字符具有特殊的含义。以这种方式转义之后，百分号与百分号匹配，下划线与下划线匹配。

支持的模式

支持长度小于等于 126 个字符的所有模式。

支持长度介于 127 和 254 字符之间的某些模式，但仅在特定条件下支持。请参见下面的各个小节以了解相关示例。

不支持长度大于等于 255 个字符的所有模式。

长度介于 127 和 254 个字符之间的模式: 示例 1

在模式中存在相邻常量字符的特定情况下，支持长度介于 127 和 254 个字符之间的模式。字符串模式中的每个常量字符都需要两个字节，即使该字符是单字节字符也是如此。**LIKE** 谓词中的字符串模式必须小于 256 字节（或 255/2 个字符），否则会出现以下错误：

There was an error reading the results of the SQL statement. The displayed results may be incorrect or incomplete. Cannot compile Like pattern: either bad pattern or pattern too long.

Sybase IQ 将相邻常量字符折叠成一个字符。例如，考虑以下 **LIKE** 谓词，该谓词的字符串长度为 130 个字符：

```
select col2 from tablen where col2 like
'123456789012345678901234567890123456789012345678901234567890123456
789012345678901234567890123456789012345678901234567890123456%%%' ;
```

Sybase IQ 将字符串末尾的四个相邻常量字符 %%% 折叠成一个 % 字符，从而将字符串长度从 130 个字符减少到 127 个字符。这小于 256 字节（或 255/2 个字符）的最大长度，并且不会产生错误。

因此，如果您的 **LIKE** 谓词在字符串中包含相邻常量，则只要折叠字符串的总长度小于 256 字节（或 255/2 个字符），系统就支持长度介于 127 和 254 个字符之间的模式。

长度介于 127 和 254 个字符之间的模式：示例 2

在此示例中，常量字符 7890 替换了 130 个字符的 **LIKE** 谓词末尾的四个相邻常量字符 %%%%：

```
select col2 from tablen where col2 like
'123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890' ;
```

在这种情况下，不折叠任何字符。字符串长度仍为 130 个字符，并且 Sybase IQ 会产生错误。

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
'123456789012345678901234567890123456789012345678901234567890123456
789012345678901234567890123456789012345678901234567890123456____' ;
```

Sybase IQ 不折叠相邻特殊字符。字符串长度仍为 130 个字符，并且 Sybase IQ 会产生错误。

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
'123456789012345678901234567890123456789012345678901234567890123456
789012345678901234567890123456789012345678901234567890123456[1-3]' ;
```

LIKE 谓词的长度（以字节为单位）按以下方式进行计算：126（对于常量字符）* 2 + 1（对于中括号中的 1）+ 1（对于中括号中的 3）+ 2（对于 Set 语句和 Range 语句表达式）。

计算结果为 256 字节，因此 Sybase IQ 会产生 错误。

搜索字符集合中的某个字符

可以通过在中括号内列出一组字符，指定对这些字符进行查找。例如，以下条件查找字符串 *smith* 和 *smyth*：

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

搜索字符范围中的某个字符

通过在中括号内列出范围的两端（用连字符分隔），可以指定要查找的字符范围。例如，以下条件查找字符串 *bough* 和 *rough*，但不查找 *tough*：

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


字符串范围 `[a-z]` 解释为“大于或等于 `a` 且小于或等于 `z`”，其中大于和小于运算在数据库归类中执行。有关归类中的字符排序的信息，请参见《系统管理指南：第一卷》>“国际语言和字符集”。

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，以下条件查找字符串 *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

可以在中括号内将范围和集合组合起来。例如，以下条件查找字符串 *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 (^) 可指定从搜索中排除的字符范围。例如，以下条件查找字符串 *tough*，但不查找字符串 *rough* 或 *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 Sybase IQ 支持 **ESCAPE** 子句。

注意：有关对大对象数据和变量的 **LIKE** 谓词支持的信息，请参见《Sybase IQ 中的非结构化数据分析》>“非结构化数据查询”。

用户必须专门获得使用许可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件的详细信息，请参见《Sybase IQ 中的非结构化数据分析》。

另请参见

- **PATINDEX** 函数 [字符串] (第 234 页)
- **LOCATE** 函数 [字符串] (第 210 页)

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 Enterprise 和 Sybase IQ 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，每个表达式正好包含一个单词。该单词的长度不能超过列的单词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 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《Sybase IQ 中的非结构化数据分析》。

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 Enterprise 和 Sybase IQ 之间是兼容的。

IS DISTINCT FROM 和 IS NOT DISTINCT FROM 搜索条件

IS DISTINCT FROM 和 IS NOT DISTINCT FROM 搜索条件可以优化搜索, 求值结果为 TRUE 或 FALSE。

IS [NOT] DISTINCT FROM 条件的语法是:

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

Remarks

如果 *expression1* 与 *expression2* 相等或者两个表达式均为 NULL, 则 **IS NOT DISTINCT FROM** 搜索条件的结果值为 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 语法使含义相反。即, 如果表达式 1 不等于表达式 2, 且至少一个表达式并非 NULL, 则 **IS DISTINCT FROM** 的结果值为 TRUE。这相当于以下语句:

```
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 标准中进行定义。**IS DISTINCT FROM** 谓词是 SQL/2008 标准的特性 T151 “DISTINCT 谓词”。**IS NOT DISTINCT FROM** 谓词是 SQL/2008 标准的特性 T152 “带负值的 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 Enterprise 和 Sybase IQ 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 Sybase IQ 和 Adaptive Server Enterprise 之间是兼容的。

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 Sybase IQ 支持真值条件。

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 32 页）
- 表达式（第 22 页）
- 空值（第 62 页）
- 搜索条件（第 31 页）
- 字符串（第 21 页）
- SQL 运算符（第 24 页）
- 搜索条件中的子查询（第 34 页）

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。

Sybase IQ 查询优化程序使用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的因素是条件的选择性；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 Studio® 支持用户提供的选择性 估计。

Adaptive Server Enterprise 不支持用户提供的选择性 估计。

另请参见

- 选择性提示 (第 47 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。

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

- 用户提供的条件选择性 (第 46 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。只有以下条件成立时，索引首选项才能对条件的执行产生影响：指定的索引类型存在于该列中，而且在评估关联条件时该索引类型可以使用；并非所有索引类型对于所有条件均可使用。请参见《参考：语句和选项》。

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
```

INDEX_PREFERENCE 选项

控制对用于查询的索引的选择。

允许值

-10 到 10

默认值

0

范围

设置此选项无需 DBA 权限。可针对单个连接或 PUBLIC 组进行临时设置。设置将立即生效。

描述

Sybase IQ 优化程序一般会选择最佳可用索引来处理本地 WHERE 子句谓词和其它能够在 IQ 索引内部完成的操作。INDEX_PREFERENCE 用来覆盖 用于测试目的的优化程序选项；在大多数环境下，不应当更改它。

表 6. INDEX_PREFERENCE 有效值

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
-9	避免 TIME 索引
-10	避免 DTTM 索引

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 Sybase IQ 查询引擎中，有四个可以评估条件的不同执行阶段：不变、延迟、绑定和水平。

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进行有关要使用的最佳连接顺序和连接算法的决策。由于这是最早的执行阶段，因此用户无法迫使条件进入此阶段，但可以迫使条件离开此阶段进入后面的阶段。

延迟 - 有些条件在查询的其它某部分执行之前无法进行评估。这些延迟条件将在其连接的查询节点首次被读取时进行一次评估。这些条件分为两类：非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同性连接条件。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 - 某些条件（如包含同一表中两列的条件）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
```

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 Studio 不支持用户提供的条件提示字符串。

Adaptive Server Enterprise 不支持用户提供的条件提示字符串。

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。

以下示例请求散列连接：

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

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。

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 22 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 54 页)
- CURRENT TIMESTAMP 特殊值 (第 53 页)
- CURRENT TIME 特殊值 (第 52 页)
- 日期和时间数据类型 (第 78 页)
- 检索日期和时间 (第 80 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 54 页)

- **CURRENT TIMESTAMP** 特殊值 (第 53 页)
- **CURRENT DATE** 特殊值 (第 52 页)
- 日期和时间数据类型 (第 78 页)
- 检索日期和时间 (第 80 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 54 页)
- **CURRENT TIME** 特殊值 (第 52 页)
- **CURRENT DATE** 特殊值 (第 52 页)
- 日期和时间数据类型 (第 78 页)
- 检索日期和时间 (第 80 页)
- **CURRENT USER** 特殊值 (第 53 页)
- **LAST USER** 特殊值 (第 54 页)
- **USER** 特殊值 (第 55 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 53 页)
- **LAST USER** 特殊值 (第 54 页)
- **USER** 特殊值 (第 55 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 53 页）
- **CURRENT TIMESTAMP** 特殊值（第 53 页）
- **USER** 特殊值（第 55 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： Sybase IQ 不支持 **UTC TIMESTAMP** 或 **CURRENT UTC TIMESTAMP** 的 **DEFAULT** 值，IQ 也不支持数据库选项 **DEFAULT_TIMESTAMP_INCREMENT**。每次尝试插入或者更新 **UTC TIMESTAMP** 或 **CURRENT UTC TIMESTAMP** 类型的列的 **DEFAULT** 值时，Sybase IQ 都会产生错误。

数据类型

TIMESTAMP

另请参见

- **CURRENT TIMESTAMP** 特殊值（第 53 页）
- **CURRENT TIME** 特殊值（第 52 页）
- **CURRENT DATE** 特殊值（第 52 页）
- 日期和时间数据类型（第 78 页）
- 检索日期和时间（第 80 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 53 页）
- **CURRENT TIMESTAMP** 特殊值（第 53 页）
- **LAST USER** 特殊值（第 54 页）

变量

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。下面的批处理语句无效，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 Enterprise 和 Sybase IQ 都支持局部变量。在 Adaptive Server Enterprise 中，所有变量都必须用一个 @ 符号作为前缀。在 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。

作用域 - 在 Sybase IQ 和 Adaptive Server Enterprise 中，局部变量的作用域是不同的。Sybase IQ 支持使用 **DECLARE** 语句声明批处理中的局部变量。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 Sybase IQ 中，每个 **DECLARE** 语句只能声明一个变量。在 Adaptive Server Enterprise 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 Enterprise 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。

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 Sybase IQ 中提供的全局变量。

表 7. Sybase IQ 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。
<code>@@fetch_status</code>	包含从上一 <code>fetch</code> 语句所生成的状态信息。 <code>@@fetch_status</code> 可包含以下值 • 0 - 读取语句成功完成。 • -1 - 读取语句导致错误。 • -2 - 结果集中没有其它数据。 此功能与 <code>@@sqlstatus</code> 相同，只不过它返回的值不同。这是为了与 Microsoft SQL Server 兼容。
<code>@@identity</code>	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， <code>@@identity</code> 都会被重置。如果一个语句插入多行， <code>@@identity</code> 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 <code>@@identity</code> 会设置为 0。 <code>@@identity</code> 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。 <code>@@identity</code> 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。
<code>@@isolation</code>	当前隔离级别。 <code>@@isolation</code> 采用活动级别的值。
<code>@@procid</code>	当前执行的过程的存储过程 ID。
<code>@@rowcount</code>	受上一语句影响的行数。应在该语句后立即检查 <code>@@rowcount</code> 的值。插入、更新和删除会将 <code>@@rowcount</code> 设置为受影响的行数。对于游标， <code>@@rowcount</code> 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 <code>@@rowcount</code> 重置为零。

变量名	含义
<i>@@servername</i>	当前数据库服务器的名称。
<i>@@sqlstatus</i>	包含从上一 FETCH 语句所生成的状态信息。
<i>@@version</i>	当前 Sybase IQ 版本的 版本号。

Adaptive Server Enterprise Sybase IQ 中支持的全局变量

下表 包括 Sybase IQ 中支持的所有 Adaptive Server Enterprise 全局变量。列表中 不包括 Sybase IQ 不支持的 Adaptive Server Enterprise 全局变量。

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 8. Sybase IQ 中支持的 Adaptive Server Enterprise 全局 变量

全局变量	返回
<i>@@char_convert</i>	返回 0。
<i>@@client_csname</i>	在 Adaptive Server Enterprise 中，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；否则包含最近 使用的字符集的名称。在 Sybase IQ 中，返回 NULL。
<i>@@client_csid</i>	在 Adaptive Server Enterprise 中，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，它包含 syscharsets 中最近使用的 客户端字符集 ID。在 Sybase IQ 中返回 -1。
<i>@@connections</i>	自服务器上次启动后的登录数。
<i>@@cpu_busy</i>	在 Adaptive Server Enterprise 中，返回 Adaptive Server Enterprise 自上次 启动后，CPU 运行 Adaptive Server Enterprise 所花费的时间（以时钟周 期为单位）。在 Sybase IQ 中，返回 0。
<i>@@error</i>	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： if @@error != 0 return 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 <i>@@error</i>，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。

全局变量	返回
<code>@@identity</code>	在 Adaptive Server Enterprise 中, 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, <code>@@identity</code> 都会被重置。如果语句插入 多行, <code>@@identity</code> 会为插入的最后一行 反映 IDENTITY 值。如果受影响的表不包含 IDENTITY 列, 则 <code>@@identity</code> 会设置为 0。 <code>@@identity</code> 的值 不受 INSERT 或 SELECT INTO 语句失败的影响, 也不受包含失败语句的事务回退的 影响。 <code>@@identity</code> 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。
<code>@@idle</code>	在 Adaptive Server Enterprise 中, 返回服务器自 上次启动起 Adaptive Server Enterprise 空闲的时间 (以时钟周期为单位)。在 Sybase IQ 中, 返回 0。
<code>@@io_busy</code>	在 Adaptive Server Enterprise 中, 返回自服务器上上次启动后 Adaptive Server Enterprise 在输入和输出 操作上花费的时间 (以时钟周期为单位)。在 Sybase IQ 中, 返回 0。
<code>@@isolation</code>	连接的当前隔离级别。在 Adaptive Server Enterprise 中, <code>@@isolation</code> 采用 活动级别的值。
<code>@@langid</code>	在 Adaptive Server Enterprise 中, 定义 当前所用语言的本地语言 ID。在 Sybase IQ 中, 返回 0。
<code>@@language</code>	在 Adaptive Server Enterprise 中, 定义 当前所用语言的名称。在 Sybase IQ 中, 返回 “English”。
<code>@@maxcharlen</code>	在 Adaptive Server Enterprise 中, 返回 Adaptive Server Enterprise 的缺省 字符集中字符的 最大长度 (以字节为单位)。在 Sybase IQ 中, 返回 1。
<code>@@max_connections</code>	对于网络服务器, 返回活动客户端 (不是数据库连接, 因为各客户端 可支持多个连接) 的最大数目。对于 Adaptive Server Enterprise, 返回 与服务器的最大 连接数。
<code>@@ncharsize</code>	在 Adaptive Server Enterprise 中, 返回国家/地区 字符的平均长度 (以字节为单位)。在 Sybase IQ 中, 返回 1。
<code>@@nestlevel</code>	在 Adaptive Server Enterprise 中, 返回当前执行的 嵌套级别 (最初为 0)。每当存储过程 (或触发器) 调用另一个存储过程 (或触发器) 时, 嵌套级别就会增加 1。在 Sybase IQ 中, 返回 -1。
<code>@@pack_received</code>	在 Adaptive Server Enterprise 中, 返回服务器 自上次启动起 Adaptive Server Enterprise 所读取的输入包 的数目。在 Sybase IQ 中, 返回 0。
<code>@@pack_sent</code>	在 Adaptive Server Enterprise 中, 返回服务器 自上次启动起 Adaptive Server Enterprise 所写入的输出包 的数目。在 Sybase IQ 中, 返回 0。
<code>@@packet_errors</code>	在 Adaptive Server Enterprise 中, 返回 Adaptive Server Enterprise 发送 和接收包时所出现 的错误数。在 Sybase IQ 中, 返回 0。

全局变量	返回
@@procid	当前执行的过程的存储过程 ID。
@@servername	本地 Adaptive Server Enterprise 或 Sybase IQ 服务器的名称。
@@spid	在 Adaptive Server Enterprise 中, 返回当前进程的服务器进程 ID 号。在 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 FETCH 语句所生成的状态信息。@@sqlstatus 可能包含以下值: • 0 - FETCH 语句成功完成。 • 1 - FETCH 语句产生错误。 • 2 - 结果集中没有其它数据。
@@thresh_hysteresis	在 Adaptive Server Enterprise 中, 返回激活一个阈值所需的可用空间变化。在 Sybase IQ 中, 返回 0。
@@timeticks	在 Adaptive Server Enterprise 中, 返回每个时钟周期中的微秒数。每个时钟周期的时间量与计算机有关。在 Sybase IQ 中, 返回 0。
@@total_errors	在 Adaptive Server Enterprise 中, 返回 Adaptive Server Enterprise 读取或写入时所出现的错误数。在 Sybase IQ 中, 返回 0。
@@total_read	在 Adaptive Server Enterprise 中, 返回服务器自上次启动起 Adaptive Server Enterprise 所读取的磁盘数。在 Sybase IQ 中, 返回 0。
@@total_write	在 Adaptive Server Enterprise 中, 返回服务器自上次启动起 Adaptive Server Enterprise 所写入的磁盘数。在 Sybase IQ 中, 返回 0。
@@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 BEGIN TRANSACTION 都会增加事务计数。
@@transtate	在 Adaptive Server Enterprise 中, 返回语句执行后事务的当前状态。在 Sybase IQ 中, 返回 -1。
@@version	有关 Adaptive Server Enterprise 或 Sybase IQ 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。

Sybase IQ 中提供了以下注释指示符:

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。Sybase 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 Enterprise 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** 包含空值。

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。

在数值表达式中使用时，空值还有一个有趣的属性。*任何*包含空值的数值表达式的结果都是空值。这意味着，如果将 **NULL** 值添加到某个数字中，则结果为 **NULL** 值而不是数字。如果要将空值视为 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** 语句将空值插入到 **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 32 页)
- 表达式 (第 22 页)
- 搜索条件 (第 31 页)
- 字符串 (第 21 页)
- 三值逻辑 (第 45 页)
- SQL 运算符 (第 24 页)
- 搜索条件中的子查询 (第 34 页)

SQL 数据类型

SQL 数据类型定义了对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 9. 字符数据类型

字符数据类型	描述
CHAR	最大长度为 <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”。 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。 无论是否指定 BLANK PADDING 选项，系统均会用空白填充所有 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。
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。此数据类型用于在进行远程数据访问时映射 Microsoft SQL Server 的 Uniqueidentifier 列。

注意： 作为单独许可的选件，Sybase IQ 支持长度范围为零 (0) 到 512TB（千吉字节）的字符大对象 (CLOB) 数据（对于大小为 128KB 的 IQ 页），或零 (0) 到 2PB（千万亿字节）的字符大对象 (CLOB) 数据（对于大小为 512KB 的 IQ 页）。最大长度为 4GB 乘以数据库页大小。请参见《Sybase IQ 中的非结构化数据分析》。

另请参见

- 二进制数据类型（第 72 页）
- NEWID 函数 [杂项]（第 223 页）
- STRTOUUID 函数 [字符串]（第 290 页）
- UUIDTOSTR 函数 [字符串]（第 311 页）
- 二进制数据类型（第 618 页）
- 字符数据类型（第 617 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 10. 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/OFFLOAD 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
```

- 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

Sybase IQ 和 Adaptive Server Enterprise 与 SQL Anywhere 之间存在字符数据兼容性差异。

- Adaptive Server Enterprise 不支持 **CHAR** 的替换形式 **CHARACTER (n)**。
- Sybase IQ 不支持 Adaptive Server Enterprise 提供的 **NCHAR**、**NVARCHAR**、**UNICHAR** 和 **UNIVARCHAR** 数据类型。Sybase IQ 支持 **CHAR** 和 **VARCHAR** 数据类型的 Unicode。
- Sybase IQ 支持的 **LONG VARCHAR** 数据类型的长度比 SQL Anywhere 支持的该数据类型要长。请参见《Sybase IQ 中的非结构化数据分析》。
- 为了使 Sybase IQ 和 Adaptive Server Enterprise 相互兼容，务必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（请参见“SQL 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。碰到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 11. 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 和 9223372036854775807（带符号）之间，或者是从 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字数据类型，其精度在算术运算后保持不变。

数值数据类型	描述
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。
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，而标度值的范围是 0 到精度值。缺省值是：标度 = 38，精度 = 126。为了确保精确，系统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标度。
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 个字节存储空间的单精度浮点数（带符号）。绝对非零值的范围是 1.175494351e-38 到 3.402823466e+38。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 12. 十进制数字的存储大小

Precision	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：

$$4 + 2 * (\text{int}(((\text{prec} - \text{scale}) + 3) / 4) + \text{int}((\text{scale} + 3) / 4) + 1)$$

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 的数量，接着是小数点，其后是 *scale* 定义的 9 的数量。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

如果针对空值到 **NUMERIC** 的显式转换既未指定精度也未指定标度，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’ 的结果数据类型超过支持的 最大数值精度 255。请为数值函数 ‘location’ 的精度设置正确的值”

数值数据兼容性

Sybase IQ 和 Adaptive Server Enterprise 与 SQL Anywhere 之间存在数值数据兼容性差异。

- 在嵌入式 SQL 中，将 **TINYINT** 列提取到 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 Enterprise 12.5.x 版不支持 不带符号的整数。您可以将 Sybase IQ 的不带符号的 整数映射到 Adaptive Server Enterprise 带符号的整数或数值数据，数据将进行隐式转换。
 - 将 **IQ UNSIGNED SMALLINT** 数据映射到 ASE **INT**
 - 如果您有负值，请将 **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 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标度设置，因为这些缺省设置是因产品而异的：

数据库	缺省精度	缺省标度
Sybase IQ	126	38
Adaptive Server Enterprise	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 Enterprise，*p* 小于或等于 15 时使用 **REAL**，而 *p* 大于 15 时使用 **DOUBLE**。对于 Sybase IQ，分界值与平台有关，但所有平台上的分界值均大于 22。
- Sybase IQ 包括两种用户定义数据类型，即 **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 Enterprise 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 223 页）
- **STRTOUUID** 函数 [字符串]（第 290 页）
- **UUIDTOSTR** 函数 [字符串]（第 311 页）
- 字符数据类型（第 65 页）
- 二进制数据类型（第 618 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 个十六进制数字。虽然缺省长度为 1 字节，但 Sybase 还是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 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。

表 13. 二进制数据类型

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 个字节。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充到列长度 <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) 个字节。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** 列均用零填充以达到列的全部宽度。在所有 **VARBINARY** 列中，系统均会截断尾随零。

以下示例创建了一个表，该表中含有使用空值和非空值定义的 **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充或截断。

```
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
0x0123000000	0x0123000000	0x0123	0x0123

bnot	bnull	vbnot	vbnull
0x0000000000	0x0000000000	0x00	0x00
0x3030323731	0x3030323731	0x3030323731	0x3030323731

因为每个存储字节都包含 2 个十六进制数字，所以 Sybase IQ 希望二进制 条目由字符 “0x” 和跟在该字符后面的偶数个数字 组成。当 “0x” 后跟 奇数个数字时，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，系统会使用零填充值，直到达到字段的全部长度：

```
INSERT zeros VALUES (0x0, 0x0, 0x0, 0x0); SELECT * FROM zeros
```

bnot	bnull	vbnot	vbnull
0x0000000000	0x0000000000	0x00	0x00

如果输入值不包括 “0x”，Sybase IQ 会假定该值为 ASCII 值并将其转换。例如：

```
CREATE TABLE sample (col_bin BINARY(8));
INSERT sample VALUES ('002710000000ae1b');
SELECT * FROM sample;
```

col_bin
0x3030323731303030

注意： 在上面的示例中，确保将 string_rtruncation 选项设置为 “off”。

选择 BINARY 值时，必须以填充零指定值或使用 CAST 函数。例如：

```
SELECT * FROM zeros WHERE bnot = 0x0123000000;
```

或

```
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组。

例如，如果将 0x1234 或 1234 从平面文件读取到 二进制列中，Sybase IQ 会将值存储为十六进制 1234。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 14. 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 98 页）
- 数据类型转换（第 86 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。

- 您不能将集合函数 **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** 数据 仅支持缺省索引、**CMP** 索引和 **TEXT** 索引类型。
- 长度等于或小于 8 字节的 **BINARY** 和 **VARBINARY** 数据支持位操作。

二进制数据兼容性

Sybase IQ、SQL Anywhere 和 Adaptive Server Enterprise 中对二进制数据尾随零的处理各不相同。

表 15. 尾随零的处理

数据类型	Sybase IQ	SQL Anywhere	Adaptive Server Enterprise
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 Enterprise、SQL Anywhere 和 Sybase IQ 都支持 **STRING_RTRUNCATION** 数据库选项，该选项会影响截断 **INSERT** 或 **UPDATE** 字符串时的错误消息报告。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 Enterprise 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个字节进行位操作。Sybase IQ 支持对二进制类型数据前八个字节进行位操作。

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 **UUIDTOSTR()** 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 Enterprise 支持。
- 向后兼容性 - 在 Sybase IQ 版本 12.7 之前创建的数据库中，系统通过 CIS 函数补偿来支持 **STRTOUUID**、**UUIDTOSTR** 和 **NEWID** 函数。在版本 15.3 和更高版本中，**STRTOUUID**、**UUIDTOSTR** 和 **NEWID** 函数是本地 Sybase IQ 函数。

二进制大对象数据

作为单独许可的选项，Sybase IQ 支持长度范围为零 (0) 到 512TB（千吉字节）的字符大对象 (BLOB) 数据（对于大小为 128KB 的 IQ 页），或零 (0) 到 2PB（千万亿字节）的字符大对象 (BLOB) 数据（对于大小为 512KB 的 IQ 页）。

最大长度为 4GB 乘以数据库页大小。请参见《Sybase IQ 中的非结构化数据分析》。

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	Sybase IQ 和 Adaptive Server Enterprise

用法

BIT 仅存储值 0 或 1。

将任意非零值插入到 **BIT** 列中都会在该列中存储一个 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 EnterpriseBIT 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 54 页）
- `CURRENT_TIMESTAMP` 特殊值（第 53 页）
- `CURRENT_TIME` 特殊值（第 52 页）
- `CURRENT_DATE` 特殊值（第 52 页）
- 检索日期和时间（第 80 页）
- `BIGTIME` 和 `BIGDATETIME` 支持（第 620 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。

表 16.

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，以便最早日期为 0001-01-01。DATE 值需要 4 字节的存储空间。
DATETIME	一个以 <code>TIMESTAMP</code> 的形式实现的域。之所以提供 <code>DATETIME</code> ，主要是为了与 Adaptive Server Enterprise 兼容。
SMALLDATETIME	一个以 <code>TIMESTAMP</code> 的形式实现的域。之所以提供 <code>SMALLDATETIME</code> ，主要是为了与 Adaptive Server Enterprise 兼容。

日期和时间数据类型	描述
TIME	一天中的时间，包含小时、分钟、秒，以及秒的小数部分。小数部分存储为 6 位小数。 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。由于此原因，在依赖高于秒的精确度的 WHERE 子句比较中，不要使用 TIME 数据类型。）
TIMESTAMP	时间点，包含年、月、日、小时、分钟、秒，以及秒的小数部分。小数部分存储为 6 位小数。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是从 0001-01-01 00:00:00.000000 到 9999-12-31 23:59:59.999999。超出范围 1600-02-28 23:59:59 到 7911-01-01 00:00:00 的 **TIMESTAMP** 数据的显示可能不完整，但完整的日期时间值存储在数据库中，通过首先将该数据转换为字符串，可以查看完整的值。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 86 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 17. 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。
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 54 页)
- `CURRENT TIMESTAMP` 特殊值 (第 53 页)
- `CURRENT TIME` 特殊值 (第 52 页)
- `CURRENT DATE` 特殊值 (第 52 页)
- 日期和时间数据类型 (第 78 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根据用户的 `DATE_ORDER` 设置错误地解释日期。

用法

格式为 `yyyy/mm/dd` 或 `yyyy-mm-dd` 的日期始终被视为日期 (无论 `DATE_ORDER` 设置是什么)。您可以使用其它字符作为分隔符；例如，问号、空格或逗号。只要不同用户使用的 `DATE_ORDER` 设置可能会不同，就应该使用此格式。例如，在存储过程中，使用明确的日期格式可以防止根据用户的 `DATE_ORDER` 设置错误地解释日期。

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，可以使用更加灵活的日期格式。Sybase IQ 可将许多字符串解释为格式。具体的解释取决于 **DATE_ORDER** 数据库选项的设置。**DATE_ORDER** 数据库选项可以拥有值 “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** 语句来更改此值。

数据库选项 **DATE_ORDER** 决定数据库是将字符串 10/11/12 解释为 Oct 11 1912、Nov 12 1910 还是 Nov 10 1912。日期字符串的年、月和日应该由某个字符（例如，“/”、“-”或空格）分隔开并按 **DATE_ORDER** 选项指定的顺序显示。

您可以将年以 2 位数或 4 位数的形式提供。**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，系统会将 50 解释为 1950，而将 49 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。

Sybase 建议您务必使用 4 位数格式指定年。

只要 **DATE_ORDER** 的设置正确，以下字符串便都是有效日期：

```
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。

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 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。

创建数据类型时需要资源权限。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。

语法

```
CREATE { DOMAIN | DATATYPE } domain-name data-type
... [ NOT ] NULL ]
... [ DEFAULT default-value ]
```

参数

- **domain-name:** - 标识符
- **data-type:** - 具有精度和小数位数的内置数据类型
- **default-value:** - *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:** - **CURRENT** { **DATE** | **TIME** | **TIMESTAMP** | **USER** | **PUBLISHER** } | **USER**

示例

- **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空值：

```
CREATE DOMAIN address CHAR( 35 ) NULL
```

用法

用户定义数据类型是内置数据类型的别名，包括适用的精度和小数位数。它们提高了方便程度并有利于数据库的一致性。

Sybase 建议您使用 **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自动成为该数据类型的所有者。在 **CREATE DATATYPE** 语句中不能指定所有者。用户定义数据类型的名称必须是唯一的，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。与内置数据类型的名称一样，用户定义数据类型的名称始终不区分大小写。

缺省情况下，用户定义数据类型允许使用空值，除非 **allow_nulls_by_default** 选项设置为 OFF。此时，缺省情况下新的用户定义数据类型不允许空值。在用户定义数据类型上

创建的列可否为空值取决于该用户定义数据类型的定义设置，而不是引用该列时的 **allow_nulls_by_default** 选项设置。任何在列定义中显式设置的空值或非空值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。在数据类型上定义的任何列都将继承 **DEFAULT** 值规范。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。有关使用列 **DEFAULT** 值的详细信息，请参见《系统管理指南第一卷》>“数据完整性”>“列缺省值有助于保持数据完整性”。

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、本地临时表和用户定义数据类型，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。必须是数据类型的所有者或者具有 **DBA** 权限才能删除用户定义数据类型。

另请参见《参考：构件块、表和过程》>“SQL 数据类型”。

副作用：

- 自动提交

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase — Adaptive Server Enterprise 不支持。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

必须具有 **RESOURCE** 权限。

域兼容性

Sybase IQ 和 Adaptive Server Enterprise 与 SQL Anywhere 之间存在域兼容性差异。

- 命名约束和缺省值 - 在 Sybase IQ 中，使用基本数据类型（并可以选择使用空值或非空值条件）创建用户定义数据类型。命名约束和命名缺省值不受支持。
- 创建数据类型 - 在 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 Enterprise 中，必须使用 **sp_addtype**。**sp_addtype** 和从 SQL Anywhere 继承的其它存储过程的所有者是 **dbo**。使用 SQL Anywhere 存储过程创建的任何对象的创建者也是 **dbo**，因此没有 **DBA** 权限的用户无法更改或删除使用 **sp_addtype** 创建的域。您必须具有 **DBA** 权限才能更改或删除使用 **CREATE DOMAIN** 创建的域。

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)** - 与 **CAST(value AS CHAR)** 类似，只不过 **string(NULL)** 是空字符串 ("")，而 **CAST(NULL AS CHAR)** 是空值。

另请参见

- 数据类型转换函数 （第 98 页）
- 存储大小 （第 75 页）

字符串到日期时间转换的兼容性

在 Sybase IQ 和 Adaptive Server Enterprise 中，将字符串转换为日期和时间数据类型时的行为有一些不同。

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 Sybase IQ 和 Adaptive Server Enterprise 都使用缺省日期 January 1, 1900。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 Enterprise 会根据值的前缀是句点还是冒号以不同方式解释值。如果前缀是冒号，则值表示秒的千分之几。如果前

缀是句点，则 1 位数表示十分之几，2 位数表示百分之几，3 位数表示千分之几。无论分隔符是什么，Sybase IQ 和 SQL Anywhere 解释值的方式都是相同的。

- Adaptive Server Enterprise 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，Sybase IQ 转换毫秒值的方式都与 Adaptive Server Enterprise 转换前缀为句点的值的方式相同：

```
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 78 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 Enterprise 支持使用空白作为第一位数；Sybase IQ 支持零或空白。

有关如何将此类数据从 Adaptive Server Enterprise 装载到 Sybase IQ 中的详细信息，请参见《系统管理指南：第一卷》>“数据导入和导出”。

从 BIT 到 BINARY 数据类型的转换

Sybase IQ 支持从 BIT 到 BINARY 以及从 BIT 到 VARBINARY 的隐式转换和显式转换，并且与这些转换的 Adaptive Server Enterprise 支持兼容。

对于比较运算符、算术运算以及 INSERT 和 UPDATE 语句，Sybase IQ 会将 BIT 隐式转换为 BINARY 并将 BIT 隐式转换为 VARBINARY 数据类型。

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，其余字节将用零填充。例如，BIT 值 1 转换为 BINARY(n) 字符串“0x0100……00”（拥有 2ⁿ 个半位元组）。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 个半位元组）。

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，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 --- 0x010000000000000000 (18
nibbles) 0x000000000000000000 (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 ---
0x010000000000000000 (18 nibbles) 0x000000000000000000 (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、算术运算以及 **INSERT** 和 **UPDATE** 语句，Sybase IQ 支持 **BIT** 和 **CHAR** 以及 **BIT** 和 **VARCHAR** 数据类型之间的隐式转换。

以下示例说明了 **BIT** 和 **CHAR** 以及 **BIT** 和 **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 tvvarchar SELECT c2 FROM tbar SELECT c2, char_length(c2) FROM
tvvarchar c2, char_length(tvvarchar.c2) -----
'1',1 '0',1
```

从 VARCHAR 到 BIT 的隐式转换:

```
INSERT tbit SELECT c2 FROM tvvarchar 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 tvvarchar SELECT CONVERT(VARCHAR(9), c2) FROM tbar SELECT c2,
char_length(c2) FROM tvvarchar c2, char_length(tvvarchar.c2)
----- '1',1 '0',1
```

从 VARCHAR 到 BIT 的显式转换:

```
INSERT tbit SELECT CONVERT (BIT, c2) FROM tvvarchar SELECT c2 FROM
tbit c2 -- 0 1
```


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。

将函数与 Sybase IQ 一起使用时，除非另外说明，否则，任何接收空值作为参数的函数都会返回空值。

如果省略 **FROM** 子句，或者查询中的所有表都位于 **SYSTEM dbspace** 中，则 **SQL Anywhere** 将取代 Sybase IQ 处理查询，并具有不同的行为，特别是关于语法和语义限制以及选项设置的影响。有关可能应用于处理的规则，请参见 **SQL Anywhere** 文档。

如果您的查询不需要 **FROM** 子句，则您可以通过添加子句 “**FROM iq_dummy**” 来强制 Sybase IQ 处理查询，其中 **iq_dummy** 是您在数据库中创建的含一行、一列的表。

另请参见

- 附加函数（第 327 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。

注意： 时序功能仅可用于 RAP - Trading Edition Enterprise。请参见《时序指南》。

有关使用 OLAP 的详细信息，请参见《系统管理指南：第二卷》。

有关 **LONG BINARY** 和 **LONG VARCHAR** 数据类型的集合函数支持的信息，请参见《Sybase IQ 中的非结构化数据分析》。

表 18. 集合函数

集合函数	参数
AVG	([DISTINCT] { <i>column-name</i> <i>numeric-expr</i> })
CORR	(dependent-expression, independent-expression)
COUNT	(*)
COUNT	([DISTINCT] { <i>column-name</i> <i>numeric-expr</i> })
COVAR_POP	(dependent-expression, independent-expression)
COVAR_SAMP	(dependent-expression, independent-expression)
CUME_DIST	()
EXP_WEIGHTED_AVG	(<i>expression</i> , <i>period-expression</i>)
FIRST_VALUE	(expression)
LAST_VALUE	(expression)
LIST	([DISTINCT] <i>string-expression</i> [, ' <i>delimiter-string</i> '] [ORDER BY <i>order-by-expression</i> [ASC DESC], ...)
MAX	([DISTINCT] { <i>column-name</i> <i>numeric-expr</i> })
MEDIAN	(expression)
MIN	([DISTINCT] { <i>column-name</i> <i>numeric-expr</i> })

集合函数	参数
REGR_AVGX	(dependent-expression, independent-expression)
REGR_AVGY	(dependent-expression, independent-expression)
REGR_COUNT	(dependent-expression, independent-expression)
REGR_INTERCEPT	(dependent-expression, independent-expression)
REGR_R2	(dependent-expression, independent-expression)
REGR_SLOPE	(dependent-expression, independent-expression)
REGR_SXX	(dependent-expression, independent-expression)
REGR_SXY	(dependent-expression, independent-expression)
REGR_SYY	(dependent-expression, independent-expression)
STDDEV	([ALL] expression)
SUM	([DISTINCT] { column-name numeric-expr })
VARIANCE	([ALL] expression)
WEIGHTED_AVG	(expression, period-expression)

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 93 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 Enterprise 不支持排名函数和 逆分布分析函数。

下表列出了分析函数及其参数。与某些集合函数不同的是，您不能在窗口函数中指定 **DISTINCT**。

表 19. 分析函数

函数	参数
AVG	({ <i>column-name</i> <i>numeric-expr</i> })
COUNT	(*)
COUNT	({ <i>column-name</i> <i>expression</i> })
DENSE_RANK	()
GROUPING *	({ GROUPING <i>group-by-expression</i> })
MAX	({ <i>column-name</i> <i>expression</i> })
MIN	({ <i>column-name</i> <i>expression</i> })
NTILE	(<i>integer</i>)
PERCENT_RANK	()
PERCENTILE_CONT	(<i>numeric-expr</i>)
PERCENTILE_DISC	(<i>numeric-expr</i>)
RANK	()
ROW_NUMBER	()
STDDEV	([ALL] <i>expression</i>)
STDDEV_POP	([ALL] <i>expression</i>)
STDDEV_SAMP	([ALL] <i>expression</i>)
SUM	({ <i>column-name</i> <i>expression</i> })
VAR_POP	([ALL] <i>expression</i>)
VAR_SAMP	([ALL] <i>expression</i>)
VARIANCE	([ALL] <i>expression</i>)

* 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 91 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。

窗口化扩展通过窗口名称或规范指定窗口函数的类型，并应用于单个查询表达式范围内的分区结果集。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...)

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 LONG BINARY 和 LONG VARCHAR 数据类型的分析函数支持的信息，请参见《Sybase IQ 中的非结构化数据分析》>“函数支持”。

另请参见

- CORR 函数 [集合]（第 138 页）
- COUNT 函数 [集合]（第 143 页）
- EXP_WEIGHTED_AVG 函数 [集合]（第 175 页）
- FIRST_VALUE 函数 [集合]（第 177 页）
- GROUPING 函数 [集合]（第 182 页）
- LAST_VALUE 函数 [集合]（第 201 页）
- MAX 函数 [集合]（第 214 页）
- MEDIAN 函数 [集合]（第 215 页）
- MIN 函数 [集合]（第 217 页）
- REGR_AVGX 函数 [集合]（第 249 页）
- REGR_COUNT 函数 [集合]（第 251 页）
- REGR_INTERCEPT 函数 [集合]（第 252 页）
- REGR_R2 函数 [集合]（第 253 页）
- REGR_SLOPE 函数 [集合]（第 254 页）
- REGR_SXX 函数 [集合]（第 256 页）

- REGR_SXY 函数 [集合] (第 257 页)
- REGR_SYY 函数 [集合] (第 258 页)
- STDDEV 函数 [集合] (第 282 页)
- SUM 函数 [集合] (第 293 页)
- STDDEV_POP 函数 [集合] (第 284 页)
- STDDEV_SAMP 函数 [集合] (第 285 页)
- VAR_POP 函数 [集合] (第 312 页)
- VAR_SAMP 函数 [集合] (第 314 页)
- VARIANCE 函数 [集合] (第 315 页)
- WEIGHTED_AVG 函数 [集合] (第 318 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列、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 子句指定对其执行排序的参数以及每组中行的排序顺序。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 选择列表中或者 **SELECT** 语句的 **ORDER BY** 子句 中使用。Rank 函数可以在视图中或联合中使用。不能在 子查询、**HAVING** 子句中，或者 **UPDATE** 或 **DELETE** 语句的 选择列表中使用 rank 函数。在 Sybase IQ 15.3 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）指定对其执行百分点函数的表达式以及每组中行的排序顺序。此 **ORDER BY** 子句只在 **WITHIN GROUP** 子句中使用，而不是用于 **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。**LAG** 函数提供在表或分区中 **CURRENT ROW** 前面且与之相距给定物理偏移量的行的访问。**LEAD** 函数提供在表或分区中 **CURRENT ROW** 后面且与之相距给定物理偏移量的行的访问。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。请参见《系统管理指南：第二卷》> “使用 OLAP” > “分析函数” > “窗口化” > “行间函数”。

行间函数需要使用 **OVER (ORDER_BY)** 子句。

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。

下表列出了数据类型转换函数及其参数。

表 20. 数据类型转换函数

数据类型转换函数	参数
BIGINTTOHEX	(<i>integer-expression</i>)
CAST	(<i>expression</i> AS <i>data type</i>)
CONVERT	(<i>data type</i> , <i>expression</i> [, <i>format-style</i>])
HEXTOBIGINT	(<i>hexadecimal-string</i>)
HEXTOINT	(<i>hexadecimal-string</i>)
INTTOHEX	(<i>integer-expr</i>)
ISDATE	(<i>string</i>)
ISNUMERIC	(<i>string</i>)

描述

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 86 页）
- 存储大小（第 75 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。

下表列出了日期和时间函数及其参数。

语法 1

表 21. 日期和时间函数

日期和时间函数	参数
DATE	(<i>expression</i>)
DATECEILING	(<i>date-part</i> , <i>datetime-expr</i> , [<i>multiple-expr</i>])
DATEFLOOR	(<i>date-part</i> , <i>datetime-expr</i> , [<i>multiple-expr</i>])
DATEFORMAT	(<i>datetime-expr</i> , <i>string-expr</i>)
DATENAME	(<i>date-part</i> , <i>date-expr</i>)
DATEROUND	(<i>date-part</i> , <i>datetime-expr</i> , [<i>multiple-expr</i>])
DATETIME	(<i>expression</i>)
DAY	(<i>date-expr</i>)
DAYNAME	(<i>date-expr</i>)
DAYS	(<i>date-expr</i>)
DAYS	(<i>date-expr</i> , <i>date-expr</i>)
DAYS	(<i>date-expr</i> , <i>integer-expr</i>)
DOW	(<i>date-expr</i>)
HOURL	(<i>datetime-expr</i>)
HOURS	(<i>datetime-expr</i>)
HOURS	(<i>datetime-expr</i> , <i>datetime-expr</i>)
HOURS	(<i>datetime-expr</i> , <i>integer-expr</i>)
ISDATE	(<i>string</i>)
MINUTE	(<i>datetime-expr</i>)

日期和时间函数	参数
MINUTES	(<i>datetime-expr</i>)
MINUTES	(<i>datetime-expr</i> , <i>datetime-expr</i>)
MINUTES	(<i>datetime-expr</i> , <i>integer-expr</i>)
MONTH	(<i>date-expr</i>)
MONTHNAME	(<i>date-expr</i>)
MONTHS	(<i>date-expr</i>)
MONTHS	(<i>date-expr</i> , <i>date-expr</i>)
MONTHS	(<i>date-expr</i> , <i>integer-expr</i>)
NOW	(*)
QUARTER	(<i>date-expr</i>)
SECOND	(<i>datetime-expr</i>)
SECONDS	(<i>datetime-expr</i>)
SECONDS	(<i>datetime-expr</i> , <i>datetime-expr</i>)
SECONDS	(<i>datetime-expr</i> , <i>integer-expr</i>)
TODAY	(*)
WEEKS	(<i>date-expr</i>)
WEEKS	(<i>date-expr</i> , <i>date-expr</i>)
WEEKS	(<i>date-expr</i> , <i>integer-expr</i>)
YEAR	(<i>date-expr</i>)
YEARS	(<i>date-expr</i>)
YEARS	(<i>date-expr</i> , <i>date-expr</i>)
YEARS	(<i>date-expr</i> , <i>integer-expr</i>)
YMD	(<i>year-num</i> , <i>month-num</i> , <i>day-num</i>)

语法 2

表 22. 与 Transact-SQL 兼容的日期和时间函数

与 Transact-SQL 兼容的日期和时间函数	参数
DATEADD	(<i>date-part</i> , <i>numeric-expression</i> , <i>date-expr</i>)
DATEDIFF	(<i>date-part</i> , <i>date-expr1</i> , <i>date-expr2</i>)
DATENAME	(<i>date-part</i> , <i>date-expr</i>)
DATEPART	(<i>date-part</i> , <i>date-expr</i>)
GETDATE	()

描述

Sybase IQ 提供了两类日期和时间函数，这两类函数可以互换使用，但具有不同的样式。其中一组与 Transact-SQL 兼容。

语法 1 中列出的日期和时间函数允许操作时间单位。大多数时间单位（如 MONTH）都具有四个用于操作时间的函数，但只使用两个名称（如 MONTH 和 MONTHS）。

语法 2 中列出的函数是 Transact-SQL 日期和时间函数。它们提供了用于访问和操作日期和时间函数的替代方式。

您应该先将日期函数的参数转换为日期，然后再使用它们。例如，下面的示例是不正确的：

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

下面的示例是正确的：

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

Sybase IQ 没有与 SQL Anywhere 相同的常量或数据类型升级，但与其共享公用用户界面。如果发出不包含 FROM 子句的 SELECT 语句，则该语句将传递到 SQL Anywhere。下面的语句由 SQL Anywhere 以独占方式处理：

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

下列语句（由 Sybase IQ 处理）对于 WEEKS 函数采用不同的起点 并返回与以上语句不同的结果：

```
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 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 不具有单个开始日期。下列语句（第一个由 SQL Anywhere 处理，第二个由 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 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： 创建一个只有一列、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 Sybase IQ 处理并且生成一致的结果。

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。

表 23. 日期分量值

日期分量	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
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。

日期分量	缩写	值
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'`

有关指定哪一天是一周的第一天的详细信息，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”>“DATE_FIRST_DAY_OF_WEEK 选项”。

兼容性
若要与 Adaptive Server Enterprise 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 146 页）
- DATECEILING 函数 [日期和时间]（第 148 页）
- DATEDIFF 函数 [日期和时间]（第 150 页）
- DATEFLOOR 函数 [日期和时间]（第 152 页）
- DATEPART 函数 [日期和时间]（第 157 页）
- DATENAME 函数 [日期和时间]（第 156 页）
- DATEROUND 函数 [日期和时间]（第 158 页）

HTTP 函数

使用 HTTP 函数可便于处理 Web 服务中的 HTTP 请求。
下表列出了所有 HTTP 函数及其参数。

表 24. HTTP 函数

HTTP 函数	参数
HTML_DECODE	(<i>string</i>)
HTML_ENCODE	(<i>string</i>)
HTTP_DECODE	(<i>string</i>)
HTTP_ENCODE	(<i>string</i>)
HTTP_HEADER	(<i>header-field-name</i>)
HTTP_VARIABLE	(<i>var-name</i> [[, <i>instance</i>], <i>header-field</i>])

HTTP 函数	参数
NEXT_HTTP_HEADER	(<i>header-name</i>)
NEXT_HTTP_VARIABLE	(<i>var-name</i>)

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。

函数

Sybase IQ 没有与 SQL Anywhere 相同的常量或数据类型升级，但与其共享公用用户界面。如果发出 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，都一定要在 **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。

下表列出了数值函数及其参数。

表 25. 数值函数

数值函数	参数
ABS	(<i>numeric-expr</i>)
ACOS	(<i>numeric-expr</i>)
ASIN	(<i>numeric-expr</i>)
ATAN	(<i>numeric-expr</i>)
ATAN2	(<i>numeric-expr1</i> , <i>numeric-expr2</i>)
CEIL	(<i>numeric-expr</i>)
CEILING	(<i>numeric-expr</i>)
COS	(<i>numeric-expr</i>)
COT	(<i>numeric-expr</i>)
DEGREES	(<i>numeric-expr</i>)
EXP	(<i>numeric-expr</i>)
FLOOR	(<i>numeric-expr</i>)
LN	(<i>numeric-expr</i>)

数值函数	参数
LOG	(<i>numeric-expr</i>)
LOG10	(<i>numeric-expr</i>)
MOD	(<i>dividend, divisor</i>)
PI	(*)
POWER	(<i>numeric-expr1, numeric-expr2</i>)
RADIANS	(<i>numeric-expr</i>)
RAND	([<i>integer-expr</i>])
REMAINDER	(<i>numeric-expr, numeric-expr</i>)
ROUND	(<i>numeric-expr, integer-expr</i>)
SIGN	(<i>numeric-expr</i>)
SIN	(<i>numeric-expr</i>)
SQRT	(<i>numeric-expr</i>)
SQUARE	(<i>numeric-expr</i>)
TAN	(<i>numeric-expr</i>)
“TRUNCATE”	(<i>numeric-expr, integer-expr</i>)
TRUNCNUM	(<i>numeric-expression, integer-expression</i>)
WIDTH_BUCKET	(<i>expression, min_value, max_value, num_buckets</i>)

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 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 IQ 页大小或多字节归类更有可能出现错误。

下表列出了字符串函数及其参数。

表 26. 字符串函数

字符串函数	参数
ASCII	(<i>string-expr</i>)
BIT_LENGTH	(<i>column-name</i>)
BYTE_LENGTH	(<i>string-expr</i>)
CHAR	(<i>integer-expr</i>)
CHAR_LENGTH	(<i>string-expr</i>)
CHARINDEX	(<i>string-expr1</i> , <i>string-expr2</i>)
DIFFERENCE	(<i>string-expr1</i> , <i>string-expr2</i>)
GRAPHICAL_PLAN	(<i>string-expr</i>)
HTML_PLAN	(<i>string-expr</i>)
INSERTSTR	(<i>numeric-expr</i> , <i>string-expr1</i> , <i>string-expr2</i>)
LCASE	(<i>string-expr</i>)
LEFT	(<i>string-expr</i> , <i>numeric-expr</i>)
LEN	(<i>string-expr</i>)
LENGTH	(<i>string-expr</i>)
LOCATE	(<i>string-expr1</i> , <i>string-expr2</i> [, <i>numeric-expr</i>])
LOWER	(<i>string-expr</i>)
LTRIM	(<i>string-expr</i>)
OCTET_LENGTH	(<i>column-name</i>)
PATINDEX	('% <i>pattern</i> %', <i>string_expr</i>)

字符串函数	参数
REPEAT	(<i>string-expr</i> , <i>numeric-expr</i>)
REPLACE	(<i>original-string</i> , <i>search-string</i> , <i>replace-string</i>)
REVERSE	(<i>expression</i> <i>uchar_expr</i>)
REPLICATE	(<i>string-expr</i> , <i>integer-expr</i>)
RIGHT	(<i>string-expr</i> , <i>numeric-expr</i>)
RTRIM	(<i>string-expr</i>)
SIMILAR	(<i>string-expr1</i> , <i>string-expr2</i>)
SORTKEY	(<i>string-expression</i> [, { <i>collation-id</i> <i>collation-name</i> [(<i>collation-tailoring-string</i>)] }])
SOUNDEX	(<i>string-expr</i>)
SPACE	(<i>integer-expr</i>)
STR	(<i>numeric_expr</i> [, <i>length</i> [, <i>decimal</i>]])
STR_REPLACE	(<i>string_expr1</i> , <i>string_expr2</i> , <i>string_expr3</i>)
STRING	(<i>string1</i> [, <i>string2</i> , ..., <i>string99</i>])
STUFF	(<i>string-expr1</i> , <i>start</i> , <i>length</i> , <i>string-expr2</i>)
SUBSTRING	(<i>string-expr</i> , <i>integer-expr</i> [, <i>integer-expr</i>])
TRIM	(<i>string-expr</i>)
UCASE	(<i>string-expr</i>)
UPPER	(<i>string-expr</i>)

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《Sybase IQ 中的非结构化数据分析》>“函数支持”。

另请参见

- 物理限制（第 329 页）

系统函数

系统函数返回系统信息。

下表列出了系统函数及其参数。

表 27. 系统函数

系统函数	参数
COL_LENGTH	(<i>table-name</i> , <i>column-name</i>)
COL_NAME	(<i>table-id</i> , <i>column-id</i> [, <i>database-id</i>])
CONNECTION_PROPERTY	(({ <i>property-id</i> <i>property-name</i> } [, <i>connection-id</i>]))
DATALength	(<i>expression</i>)
DB_ID	([<i>database-name</i>])
DB_NAME	([<i>database-id</i>])
DB_PROPERTY	(({ <i>property-id</i> <i>property-name</i> } [, { <i>database-id</i> <i>database-name</i> }]))
EVENT_CONDITION	(<i>condition-name</i>)
EVENT_CONDITION_NAME	(<i>integer</i>)
EVENT_PARAMETER	(<i>context-name</i>)
GROUP_MEMBER	(<i>group-name-string-expression</i> [, <i>user-name-string-expression</i>])
INDEX_COL	(<i>table-name</i> , <i>index-id</i> , <i>key_#</i> [, <i>user-id</i>])
NEXT_CONNECTION	([<i>connection-id</i>] [, <i>database-id</i>])
NEXT_DATABASE	(({ NULL <i>database-id</i> }))
OBJECT_ID	(<i>object-name</i>)
OBJECT_NAME	(<i>object-id</i> [, <i>database-id</i>])
PROPERTY	(({ <i>property-id</i> <i>property-name</i> }))
PROPERTY_DESCRIPTION	(<i>property-id</i> <i>property-name</i>)
PROPERTY_NAME	(<i>property-id</i>)
PROPERTY_NUMBER	(<i>property-name</i>)
SUSER_ID	([<i>user-name</i>])
SUSER_NAME	([<i>user-id</i>])
USER_ID	([<i>user-name</i>])
USER_NAME	([<i>user-id</i>])

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 Sybase IQ 函数。当对 Sybase IQ 表的查询包括系统函数时，查询的性能会降低。

兼容性

下表显示了 Adaptive Server Enterprise 系统函数及其在 Sybase IQ 中的状态：

表 28. Sybase IQ 中的 ASE 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	总是返回 0
show_role	总是返回空
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

函数	状态
lct_admin	未实现
reserved_pgs	未实现
rowcnt	未实现
used_pgs	未实现
valid_name	未实现
valid_user	未实现

注释

- 有些系统函数在 Sybase IQ 中是作为 系统存储过程来实现的。
- db_id、db_name、datalength、suser_id 和 suser_name 函数是作为内置函数来实现的。

连接属性

检索特定连接属性的值或所有连接属性的值。

连接属性适用于单个连接。有关所有连接属性的说明，请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “连接属性、数据库属性和数据库服务器属性” > “连接属性”。连接属性 **QueryBypassedCosted**、**QueryBypassedOptimized**、**QueryDescribedOptimizer** 和 **StatementPostAnnotatesSimple** 适用于 SQL Anywhere 对象，但不适用于 IQ 表。

注意： 此参考指向 SQL Anywhere 文档。

示例

使用 **connection_property** 系统函数来检索单个连接属性的值。下面的语句返回已通过当前连接从文件中读取的页数：

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

使用 **sa_conn_properties** 系统过程来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 242 页）
- **PROPERTY_NAME** 函数 [系统]（第 243 页）
- **PROPERTY_NUMBER** 函数 [系统]（第 244 页）
- **CONNECTION_PROPERTY** 函数 [系统]（第 133 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。

服务器属性作为整体应用于整个服务器中。有关所有服务器属性的说明，请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “连接属性、数据库属性和数据库服务器属性” > “数据库服务器属性”。

Server Edition 属性返回 SQL Anywhere 版本，而非 Sybase IQ 版本。若要显示 Sybase IQ 许可证信息，请使用 **sp_iqlmconfig** 系统过程。

Server Edition 属性返回 SQL Anywhere 版本，而非 Sybase IQ 版本。若要显示 Sybase IQ 许可证信息，请使用 **sp_iqlmconfig** 系统过程。

示例

使用 **property** 系统函数来检索单个服务器属性的值。下面的语句返回用于保存主堆的高速缓存页的数量：

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

使用 **sa_eng_properties** 系统过程来检索所有服务器属性的值。

```
call sa_eng_properties
```

另请参见

- **sp_iqlmconfig** 过程（第 419 页）
- **PROPERTY** 函数 [系统]（第 242 页）
- **PROPERTY_NAME** 函数 [系统]（第 243 页）
- **PROPERTY_NUMBER** 函数 [系统]（第 244 页）
- **CONNECTION_PROPERTY** 函数 [系统]（第 133 页）

可用于每个数据库的属性

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。

有关所有数据库属性的说明，请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “连接属性、数据库属性和数据库服务器属性” > “数据库服务器属性”。

服务器属性 **QueryBypassedCosted**、**QueryBypassedOptimized**、**QueryDescribedOptimizer** 和 **StatementPostAnnotatesSimple** 仅针对 SQL Anywhere 表的查询进行了更新。

示例

使用 **db_property** 系统函数来检索单个数据库属性的值。下面的语句返回当前数据库的页大小：

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

使用 **sa_db_properties** 系统过程来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 242 页)
- `PROPERTY_NAME` 函数 [系统] (第 243 页)
- `PROPERTY_NUMBER` 函数 [系统] (第 244 页)
- `CONNECTION_PROPERTY` 函数 [系统] (第 133 页)

SQL 和 Java 用户定义的函数

在 Sybase IQ 中有两种创建用户定义的 函数的机制。您可以使用 SQL 语言编写函数，也可以使用 Java。

注意： 用户定义的函数由 SQL Anywhere 来处理。它们不能利用 Sybase IQ 的性能特性。包含用户定义的函数的查询 比不包含用户定义的函数的查询的运行速度至少慢 10 倍。

在极少数的情况下，SQL Anywhere 和 Sybase IQ 之间存在的语义区别会使同一个查询返回不同的结果（如果该查询是使用用户定义的 函数执行的）。例如，Sybase IQ 将 `CHAR` 和 `VARCHAR` 数据类型视为 各不相同的数据类型，而 SQL Anywhere 按 `VARCHAR` 的处理方式 来处理 `CHAR` 数据。

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的函数的视图中使用 `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。

有关创建 SQL 函数的详细信息，请参见《系统管理指南第二卷》>“使用过程和批处理”。

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 类和类方法的详细信息，请参见“SQL Anywhere Server - 编程”>“数据库中的 Java”。

时序和预测函数

时序和预测函数是专门为财务时序分析而设计的。《时序指南》中介绍了时序和预测函数。

注意： 时序和预测功能仅可用于 RAP - Trading Edition Enterprise。

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。下表列出了杂项函数及其参数。

表 29. 杂项函数

杂项函数	参数
ARGN	(<i>integer-expr</i> , <i>expression</i> [, ...])
COALESCE	(<i>expression</i> , <i>expression</i> [, <i>expression</i> ...])
IFNULL	(<i>expression1</i> , <i>expression2</i> [, <i>expression3</i>])
ISNULL	(<i>expression</i> , <i>expression</i> [, <i>expression</i> ...])
NULLIF	(<i>expression1</i> , <i>expression2</i>)
NUMBER	(*)
ROWID	(<i>table-name</i>)

兼容性

Adaptive Server Enterprise 仅支持 COALESCE、ISNULL、和 NULLIF 函数。

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

数字表达式的绝对值。

数字表达式数据类型	返回
INT	INT
FLOAT	FLOAT
DOUBLE	DOUBLE
NUMERIC	NUMERIC

示例

下面的语句返回值 66:

```
SELECT ABS( -66 ) 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。

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 30. 参数

参数	描述
numeric-expression	角度的余弦值。

返回

DOUBLE

示例

下面的语句返回值 1.023945:

```
SELECT ACOS( 0.52 ) 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。

另请参见

- COS 函数 [数值] (第 139 页)
- ATAN2 函数 [数值] (第 118 页)
- ATAN 函数 [数值] (第 118 页)
- ASIN 函数 [数值] (第 117 页)
- COT 函数 [数值] (第 140 页)
- SIN 函数 [数值] (第 274 页)
- TAN 函数 [数值] (第 296 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 31. 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）。

示例

下面的语句返回值 6:

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

用法

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 Enterprise 兼容。

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

表 32. 参数

参数	描述
string-expression	字符串。

返回

SMALLINT

示例

当归类序列设置为缺省值 ISO_BINENG 时，下面的语句返回值 90:

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

用法

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。
- Sybase - 与 Adaptive Server Enterprise 兼容。

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

表 33. 参数

参数	描述
numeric-expression	角度的正弦值。

返回

DOUBLE

示例

下面的语句返回值 0.546850。

```
SELECT ASIN( 0.52 ) 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。

另请参见

- COS 函数 [数值]（第 139 页）
- ATAN2 函数 [数值]（第 118 页）
- ATAN 函数 [数值]（第 118 页）
- ACOS 函数 [数值]（第 114 页）
- COT 函数 [数值]（第 140 页）
- SIN 函数 [数值]（第 274 页）
- TAN 函数 [数值]（第 296 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

表 34. 参数

参数	描述
numeric-expression	角度的正切值。

返回

DOUBLE

示例

下面的语句返回值 0.479519:

```
SELECT ATAN( 0.52 ) 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。

另请参见

- COS 函数 [数值]（第 139 页）
- ATAN2 函数 [数值]（第 118 页）
- ASIN 函数 [数值]（第 117 页）
- ACOS 函数 [数值]（第 114 页）
- COT 函数 [数值]（第 140 页）
- SIN 函数 [数值]（第 274 页）
- TAN 函数 [数值]（第 296 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

示例

下面的语句返回值 0.00866644968879073143:

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

标准和兼容性

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

另请参见

- COS 函数 [数值] (第 139 页)
- ATAN 函数 [数值] (第 118 页)
- ASIN 函数 [数值] (第 117 页)
- ACOS 函数 [数值] (第 114 页)
- COT 函数 [数值] (第 140 页)
- SIN 函数 [数值] (第 274 页)
- TAN 函数 [数值] (第 296 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

表 35. 参数

参数	描述
numeric-expression	要针对一组行计算其平均值的值。

参数	描述
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。

示例

下面的语句返回值 49988.6：

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

用法

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 Enterprise 兼容。

另请参见

- COUNT 函数 [集合]（第 143 页）
- SUM 函数 [集合]（第 293 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。

用法

IQ 数据抽取工具包含 **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** 不一定要与数据抽取工具一起使用。

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。

请参见《Sybase IQ 中的非结构化数据分析》> “函数支持”。

BIGINTTOHEX 函数 [数据类型转换]

在 VARCHAR(16) 中返回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
<i>integer-expression</i>	要转换成十六进制的整数。

示例

返回值 0000000000000009:

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

返回值 FFFFFFFF7:

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

用法

BIGINTTOHEX 接受计算结果为 **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，直到达到 16 位数字（最大数目）。对于所有无标度的整数数据类型，都可以将其作为整数表达式接受。

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。如果列是使用正标度值声明的，则无法截断该列。如果转换失败，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 Enterprise 兼容。

另请参见

- **HEXTOBIGINT** 函数 [数据类型转换]（第 184 页）
- **HEXTOINT** 函数 [数据类型转换]（第 185 页）
- **INTTOHEX** 函数 [数据类型转换]（第 195 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

表 36. 参数

参数	描述
column-name	列的名称

返回

INT

用法

空值参数的返回值为空。

BIT_LENGTH 函数支持 所有 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 《Sybase IQ 中的非结构化数据分析》 > “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 Enterprise 不支持。

另请参见

- **BYTE_LENGTH** 函数 [字符串] （第 122 页）
- **CHAR_LENGTH** 函数 [字符串] （第 128 页）
- **COL_LENGTH** 函数 [系统] （第 131 页）
- **DATALength** 函数 [系统] （第 145 页）
- **LEN** 函数 [字符串] （第 206 页）
- **LENGTH** 函数 [字符串] （第 207 页）
- **OBJECT_NAME** 函数 [系统] （第 232 页）
- **OCTET_LENGTH** 函数 [字符串] （第 233 页）
- **STR_REPLACE** 函数 [字符串] （第 287 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

示例

它返回值 12:

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

用法

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，则可以将此函数与大对象数据一起使用。

请参见《Sybase IQ 中的非结构化数据分析》> “函数支持”。

标准和兼容性

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

另请参见

- BIT_LENGTH 函数 [字符串] (第 121 页)
- CHAR_LENGTH 函数 [字符串] (第 128 页)
- COL_LENGTH 函数 [系统] (第 131 页)
- DATALENGTH 函数 [系统] (第 145 页)
- LEN 函数 [字符串] (第 206 页)
- LENGTH 函数 [字符串] (第 207 页)
- OBJECT_NAME 函数 [系统] (第 232 页)
- OCTET_LENGTH 函数 [字符串] (第 233 页)
- STR_REPLACE 函数 [字符串] (第 287 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。

用法

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《Sybase IQ 中的非结构化数据分析》> “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。

用法

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《Sybase IQ 中的非结构化数据分析》> “函数支持”。

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。

示例

下面的函数可确保使用字符串作为日期：

```
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
```

用法

如果未指定字符串类型的长度，Sybase IQ 将选择适当的长度。如果没有为十进制转换指定精度和标度，则数据库服务器将选择适当的值。

如果针对空值到 **NUMERIC** 的显式转换既未指定精度也未指定标度，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 Enterprise 兼容。

另请参见

- CONVERT 函数 [数据类型转换]（第 134 页）
- HOURS 函数 [日期和时间]（第 188 页）
- MINUTES 函数 [日期和时间]（第 218 页）
- MONTHS 函数 [日期和时间]（第 221 页）
- REPLACE 函数 [字符串]（第 261 页）
- SECOND 函数 [日期和时间]（第 270 页）
- WEEKS 函数 [日期和时间]（第 317 页）
- YEAR 函数 [日期和时间]（第 321 页）
- YEARS 函数 [日期和时间]（第 322 页）
- DAYS 函数 [日期和时间]（第 163 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， CEIL 会产生错误。返回值与所提供的值属于相同的数据类型。

用法

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。

请参见《系统管理指南第一卷》>“国际语言和字符集”。

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 179 页)
- CEILING 函数 [数值] (第 126 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

示例

下面的语句返回值 60.00000:

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

下面的语句返回值 123:

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

下面的语句返回值 124.00:

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

下面的语句返回值 -123.00:

```
SELECT CEILING( -123.45 ) 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。

另请参见

- FLOOR 函数 [数值] (第 179 页)
- CEIL 函数 [数值] (第 125 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

示例

下面的语句返回值 “Y” :

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

下面的语句返回值 “S” :

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

用法

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 Enterprise 兼容。

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
string-expression	要计算其长度的字符串。

返回

INT

用法

返回的长度中包括尾随空格字符。

NULL 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 **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，则可以将此函数与大对象数据一起使用。请参见《Sybase IQ 中的非结构化数据分析》>“函数支持”。

示例

下面的语句返回值 8：

SELECT CHAR_LENGTH('Chemical') FROM iq_dummy

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

- **BIT_LENGTH** 函数 [字符串]（第 121 页）
- **BYTE_LENGTH** 函数 [字符串]（第 122 页）

- COL_LENGTH 函数 [系统] (第 131 页)
- DATALENGTH 函数 [系统] (第 145 页)
- LEN 函数 [字符串] (第 206 页)
- LENGTH 函数 [字符串] (第 207 页)
- OBJECT_NAME 函数 [系统] (第 232 页)
- OCTET_LENGTH 函数 [字符串] (第 233 页)
- STR_REPLACE 函数 [字符串] (第 287 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）。

用法

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《Sybase IQ 中的非结构化数据分析》>“函数支持”。

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
string-expression1	要搜索的字符串。该字符串限制为 255 个字节。
string-expression2	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

示例

语句:

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

返回下列值:

Surname	GivenName
Klobucher	James
Kuo	Felicia
Kelly	Moir

用法

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，则可以将此函数与大对象数据一起使用。请参见《Sybase IQ 中的非结构化数据分析》>“函数支持”。

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

- SUBSTRING 函数 [字符串] (第 292 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

表 37. 参数

参数	描述
表达式	任意表达式。

返回

ANY

示例

下面的语句返回值 34:

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

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

- ISNULL 函数 [杂项] (第 197 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。

示例

返回列长度 35:

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

标准和兼容性

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

另请参见

- BIT_LENGTH 函数 [字符串] (第 121 页)
- BYTE_LENGTH 函数 [字符串] (第 122 页)
- CHAR_LENGTH 函数 [字符串] (第 128 页)
- DATALENGTH 函数 [系统] (第 145 页)
- LEN 函数 [字符串] (第 206 页)

- LENGTH 函数 [字符串] (第 207 页)
- OBJECT_NAME 函数 [系统] (第 232 页)
- OCTET_LENGTH 函数 [字符串] (第 233 页)
- STR_REPLACE 函数 [字符串] (第 287 页)

COL_NAME 函数 [系统]

返回列名。

语法

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

参数

表 38. 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。

示例

下面的语句返回列名 **lname**。Customers 表的对象 ID 为 100209，这由 **OBJECT_ID** 函数返回。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，这由 **DB_ID** 函数返回。

```
SELECT COL_NAME( 100209, 3, 0 ) FROM iq_dummy
```

下面的语句返回列名 **city**。

```
SELECT COL_NAME ( 100209, 5 )FROM iq_dummy
```

标准和兼容性

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

另请参见

- DB_ID 函数 [系统] (第 164 页)
- DB_NAME 函数 [系统] (第 165 页)
- DB_PROPERTY 函数 [系统] (第 166 页)
- NEXT_DATABASE 函数 [系统] (第 226 页)
- OBJECT_ID 函数 [系统] (第 232 页)
- OBJECT_NAME 函数 [系统] (第 232 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

示例

下面的语句返回所维护的已预备语句的数目，例如 4：

```
SELECT connection_property( 'PrepStmt' )FROM iq_dummy
```

用法

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 Enterprise 兼容。

另请参见

- PROPERTY 函数 [系统]（第 242 页）
- PROPERTY_NAME 函数 [系统]（第 243 页）
- PROPERTY_NUMBER 函数 [系统]（第 244 页）
- 可用于服务器的属性（第 111 页）

- 可用于每个数据库的属性（第 111 页）
- 连接属性（第 110 页）
- sp_iqshowpsexec 过程（第 446 页）
- sp_iqcontext 过程（第 364 页）

CONVERT 函数 [数据类型转换]

返回转换成提供的数据库类型的表达式。

语法

```
CONVERT ( data-type, expression [ , format-style ] )
```

参数

参数	描述
data-type	表达式将要转换成的数据类型。
表达式	要转换的表达式。
format-style	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，format-style 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 39. 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]yyymmdd
-	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.ssssssAM (或 PM)
37	137	hh:nn:ss.ssssss
38	138	mmm dd yy[yy] hh:nn:ss.ssssssAM (或 PM)
39	139	mmm dd yy[yy] hh:nn:ss.ssssss
40	140	[yy]yy-mm-dd hh:nn:ss.ssssss
-	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。

示例
下列语句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
.....

```
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
```

下列语句说明了格式样式 365 的用法，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
```

下面的语句说明了到整数的转换，并返回值 5。

```
SELECT CONVERT( integer, 5.2 ) FROM iq_dummy
```

用法

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件许可证，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 Enterprise 和 SQL Anywhere 兼容，但格式样式 365 除外，该样式是仅限 Sybase IQ 的扩展。

另请参见

- **CAST** 函数 [数据类型转换] (第 124 页)
- **HOURS** 函数 [日期和时间] (第 188 页)
- **MINUTES** 函数 [日期和时间] (第 218 页)
- **MONTHS** 函数 [日期和时间] (第 221 页)
- **REPLACE** 函数 [字符串] (第 261 页)
- **SECOND** 函数 [日期和时间] (第 270 页)
- **WEEKS** 函数 [日期和时间] (第 317 页)
- **YEAR** 函数 [日期和时间] (第 321 页)

- YEARS 函数 [日期和时间] （第 322 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 40. 参数

参数	描述
dependent-expression	受 independent-expression 影响的变量。
independent-expression	影响结果的变量。

返回

DOUBLE

示例

下面的示例执行相关，看年龄是否与收入水平有关。以下函数返回值 0.440227:

```
SELECT CORR( Salary, ( YEAR( NOW( ) ) - YEAR( BirthDate ) ) ) FROM  
Employees;
```

用法

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。

另请参见

- 窗口化集合函数的用法（第 95 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 41. 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。

示例

下面的语句返回值 0.86781:

```
SELECT COS( 0.52 ) 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。

另请参见

- ATAN2 函数 [数值]（第 118 页）
- ATAN 函数 [数值]（第 118 页）
- ASIN 函数 [数值]（第 117 页）
- ACOS 函数 [数值]（第 114 页）
- COT 函数 [数值]（第 140 页）
- SIN 函数 [数值]（第 274 页）

- TAN 函数 [数值] (第 296 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 42. 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。

示例

下面的语句返回值 1.74653：

```
SELECT COT( 0.52 ) 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。

另请参见

- COS 函数 [数值] (第 139 页)
- ATAN2 函数 [数值] (第 118 页)
- ATAN 函数 [数值] (第 118 页)
- ASIN 函数 [数值] (第 117 页)
- ACOS 函数 [数值] (第 114 页)
- SIN 函数 [数值] (第 274 页)
- TAN 函数 [数值] (第 296 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

表 43. 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。

示例

下面的示例度量员工年龄与工资之间的关联强度。以下函数返回值 73785.840059:

```
SELECT COVAR_POP( Salary, ( YEAR( NOW( ) ) - YEAR( BirthDate ) ) )
FROM Employees;
```

用法

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：

$$(SUM(x*y) - SUM(x) * SUM(y) / n) / n$$

其中 *x* 表示 *dependent-expression*，*y* 表示 *independent-expression*。

请参见“SQL Anywhere Server - SQL 用法”>“查询和修改数据”>“OLAP 支持”>“SQL Anywhere 中的窗口函数”>“行计算函数”>“集合函数的数学公式”。

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。

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

表 44. 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。

示例

下面的示例度量员工年龄与工资之间的关联强度。以下函数返回值 74782.946005:

```
SELECT COVAR_SAMP( Salary, ( 2008 - YEAR( BirthDate ) ) ) FROM Employees;
```

用法

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。

请参见“SQL Anywhere Server - SQL 用法”>“查询和修改数据”>“OLAP 支持”>“SQL Anywhere 中的窗口函数”>“行计算函数”>“集合函数的数学公式”。

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。

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 expression 不为空值的行数。
DISTINCT column-name	返回 column-name 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

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
```

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

- AVG 函数 [集合] （第 119 页）
- SUM 函数 [集合] （第 293 页）
- 窗口化集合函数的用法 （第 95 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。

示例

下面的示例返回一个结果集，该结果集提供居住在加利福尼亚的员工的工资累积分布：

```
SELECT DepartmentID, Surname, Salary,CUME_DIST() OVER (PARTITION BY
DepartmentIDORDER BY Salary DESC) "Rank"FROM Employees WHERE State IN
('CA');
```

返回的结果集为：

表 45. 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

用法

Sybase IQ 计算大小为 N 的 S 集合中 x 值的累积分布，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。

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

用法

表 46. 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

示例

返回值 35，即 `company_name` 列中最长的字符串：

```
SELECT MAX( DATALENGTH( company_name ) )  
FROM Customers
```

标准和兼容性

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

另请参见

- BIT_LENGTH 函数 [字符串]（第 121 页）
- BYTE_LENGTH 函数 [字符串]（第 122 页）
- CHAR_LENGTH 函数 [字符串]（第 128 页）
- COL_LENGTH 函数 [系统]（第 131 页）

- LEN 函数 [字符串] (第 206 页)
- LENGTH 函数 [字符串] (第 207 页)
- OBJECT_NAME 函数 [系统] (第 232 页)
- OCTET_LENGTH 函数 [字符串] (第 233 页)
- STR_REPLACE 函数 [字符串] (第 287 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 47. 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

示例

下面的语句返回值 1988-11-26 作为日期。

```
SELECT DATE( '1988-11-26 21:20:53' ) FROM iq_dummy
```

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。

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

示例

下面的语句返回值 1995-11-02 00:00:00.000:

```
SELECT DATEADD( MONTH, 102, '1987/05/02' ) FROM iq_dummy
```

下面的语句返回值 2009-11-10 14:57:52.722016:

```
SELECT DATEADD(MICROSECOND, 15, '2009-11-10 14:57:52.722001') FROM iq_dummy
```

下面的语句返回值 1985-05-02 00:00:00.123456:

```
SELECT DATEADD(MICROSECOND, 123456, '1985/05/02') FROM iq_dummy
```

下面的语句返回值 1985-05-01 23:59:59.876544:

```
SELECT DATEADD(MICROSECOND, -123456, '1985/05/02') FROM iq_dummy
```

下面的语句返回值 2009-11-03 11:10:42.033192:

```
SELECT DATEADD(MCS, 2, '2009-11-03 11:10:42.033190') FROM iq_dummy
```

用法

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。

另请参见

- DATECEILING 函数 [日期和时间]（第 148 页）
- DATEDIFF 函数 [日期和时间]（第 150 页）
- DATEFLOOR 函数 [日期和时间]（第 152 页）
- DATEPART 函数 [日期和时间]（第 157 页）
- DATENAME 函数 [日期和时间]（第 156 页）
- DATEROUND 函数 [日期和时间]（第 158 页）
- 日期分量（第 102 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
date-part	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	(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 date-part 参数指定的单位的倍数是多少。例如，您可以使用 multiple-expression 指定要将数据的间隔调整为 200 微秒或 10 分钟。 如果 multiple-expression 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 date-part 的有效值，则 Sybase IQ 会产生错误。如果 multiple-expression 计算结果为空值，则该函数的结果也为空值。

示例

下面的语句返回值 August 13, 2009 10:40.00.000AM:

```
SELECT DATECEILING( MI, 'August 13, 2009, 10:32.00.132AM', 10) FROM iq_dummy
```

下面的语句返回值 August 13, 2009 10:32.35.456800 AM:

```
SELECT DATECEILING( US, 'August 13, 2009, 10:32.35.456789AM', 200 ) FROM iq_dummy
```

下面的语句返回值 August 13, 2009 10:32.35.600000 AM:

```
SELECT DATECEILING( US, 'August 13, 2009, 10:32.35.456789AM', 200000 ) FROM iq_dummy
```

下面的语句返回值 August 13, 2009 10:32.35.456789 AM:

```
SELECT DATECEILING( US, 'August 13, 2009, 10:32.35.456789AM') FROM iq_dummy
```

用法

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

如果您对微秒、毫秒、秒、分钟或小时日期分量指定 *multiple-expression*，则 Sybase IQ 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，如果指定一个两分钟的倍数，则 Sybase IQ 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、季度或年日期分量指定 *multiple-expression*，则 Sybase IQ 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 Sybase IQ 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 Sybase IQ 舍入为周日期分量的倍数，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 — Adaptive Server Enterprise 或 SQL Anywhere 不支持。

另请参见

- DATEADD 函数 [日期和时间] (第 146 页)
- DATEDIFF 函数 [日期和时间] (第 150 页)
- DATEFLOOR 函数 [日期和时间] (第 152 页)
- DATEPART 函数 [日期和时间] (第 157 页)
- DATENAME 函数 [日期和时间] (第 156 页)
- DATEROUND 函数 [日期和时间] (第 158 页)
- 日期分量 (第 102 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

表 48. 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 <i>date-expression1</i> 以返回两个参数间的日期分量的数量。

返回

INT

示例

下面的语句返回 1:

```
SELECT DATEDIFF( HOUR, '4:00AM', '5:50AM' )
FROM iq_dummy
```

下面的语句返回 102:

```
SELECT DATEDIFF( MONTH, '1987/05/02', '1995/11/15' )
FROM iq_dummy
```

下面的语句返回 0:

```
SELECT DATEDIFF( DAY, '00:00', '23:59' ) FROM iq_dummy
```

下面的语句返回 4:

```
SELECT DATEDIFF( DAY, '1999/07/19 00:00', '1999/07/23
23:59' ) FROM iq_dummy
```

下面的语句返回 0:

```
SELECT DATEDIFF( MONTH, '1999/07/19', '1999/07/23' )
FROM iq_dummy
```

下面的语句返回 1:

```
SELECT DATEDIFF( MONTH, '1999/07/19', '1999/08/23' )
FROM iq_dummy
```

下面的语句返回 4:

```
SELECT DATEDIFF(MCS, '2009-11-03 11:10:42.033185', '2009-11-03
11:10:42.033189') FROM iq_dummy
```

下面的语句返回 15:

```
SELECT DATEDIFF(MICROSECOND, '2009-11-10 14:57:52.722001',
'2009-11-10 14:57:52.722016') FROM iq_dummy
```

下面的语句返回 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
```

用法

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** 作为日期分量时，**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，下面的语句返回 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 是月的第一天，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 也是月的第一天，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。

另请参见

- DATEADD 函数 [日期和时间]（第 146 页）
- DATECEILING 函数 [日期和时间]（第 148 页）
- DATEFLOOR 函数 [日期和时间]（第 152 页）
- DATEPART 函数 [日期和时间]（第 157 页）
- DATENAME 函数 [日期和时间]（第 156 页）
- DATEROUND 函数 [日期和时间]（第 158 页）
- 日期分量（第 102 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。

参数	描述
multiple-expression	(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 date-part 指定的单位的倍数是多少。例如, 您可以使用 multiple-expression 指定要将数据的间隔调整为 200 微秒或 10 分钟。 如果 multiple-expression 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 date-part 的有效值, 则 Sybase IQ 会产生错误。如果 multiple-expression 计算结果为空值, 则该函数的结果也为空值。

示例

- 下面的语句返回值 August 13, 2009 10:35:00.000AM:

```
SELECT DATEFLOOR( MINUTE, 'August 13, 2009 10:35:22.123AM') FROM iq_dummy
```

- 下面的语句返回值 August 13, 2009 10:32:35.456600 AM:

```
SELECT DATEFLOOR( US, 'August 13, 2009, 10:32:35.456789AM', 200 ) FROM iq_dummy
```

- 下面的语句返回值 August 13, 2009 10:32:35.400000 AM:

```
SELECT DATEFLOOR( US, 'August 13, 2009, 10:32:35.456789AM', 200000 ) FROM iq_dummy
```

- 下面的语句返回值 August 13, 2009 10:32:35.456789 AM:

```
SELECT DATEFLOOR( US, 'August 13, 2009, 10:32:35.456789AM') FROM iq_dummy
```

用法

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

如果您对微秒、毫秒、秒、分钟或小时日期分量指定 *multiple-expression*, 则 Sybase IQ 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，如果指定一个两分钟的倍数，则 Sybase IQ 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、季度或年日期分量指定 *multiple-expression*，则 Sybase IQ 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 Sybase IQ 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 Sybase IQ 舍入为周日期分量的倍数，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 — Adaptive Server Enterprise 或 SQL Anywhere 不支持。

另请参见

- DATEADD 函数 [日期和时间] (第 146 页)
- DATECEILING 函数 [日期和时间] (第 148 页)
- DATEDIFF 函数 [日期和时间] (第 150 页)
- DATEPART 函数 [日期和时间] (第 157 页)
- DATENAME 函数 [日期和时间] (第 156 页)
- DATEROUND 函数 [日期和时间] (第 158 页)
- 日期分量 (第 102 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

表 49. 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

示例

下面的语句返回类似 “Jan 01, 1989” 的字符串值：

```
SELECT DATEFORMAT( start_date, 'Mmm dd, yyyy' ) from Employees;
```

下面的语句返回字符串 “Feb 19, 1987”：

```
SELECT DATEFORMAT( CAST ( '1987/02/19' AS DATE ), 'Mmm Dd, yyyy' )  
FROM iq_dummy
```

用法

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，Sybase IQ 会隐式将字符串转换为日期、时间或时间戳数据类型，因此不需要像上面的示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;
```

若要设置用于从数据库中检索的日期的格式，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”>“DATE_FORMAT 选项”。

标准和兼容性

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

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 50. 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

示例

下面的语句返回值 May：

```
SELECT DATENAME( MONTH , '1987/05/02' )  
FROM iq_dummy
```

下面的语句返回值 722,001：

```
SELECT DATENAME(MICROSECOND, '2009-11-10 14:57:52.722001') FROM  
iq_dummy
```

下面的语句返回值 777,777：

```
SELECT DATENAME(MICROSECOND, '2000/07/07 07:07:07.777777') FROM  
iq_dummy
```

下面的语句返回值 33,189：

```
SELECT DATENAME(MCS, '2009-11-03 11:10:42.033189') FROM iq_dummy
```

用法

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。

另请参见

- DATEADD 函数 [日期和时间]（第 146 页）
- DATECEILING 函数 [日期和时间]（第 148 页）
- DATEDIFF 函数 [日期和时间]（第 150 页）
- DATEFLOOR 函数 [日期和时间]（第 152 页）
- DATEPART 函数 [日期和时间]（第 157 页）
- DATEROUND 函数 [日期和时间]（第 158 页）
- 日期分量（第 102 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 51. 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

示例

下面的语句返回值 5:

```
SELECT DATEPART( MONTH, '1987/05/02' )
FROM iq_dummy
```

下面的语句返回值 722,001:

```
SELECT DATEPART(MICROSECOND, '2009-11-10 14:57:52.722001') FROM iq_dummy
```

下面的语句返回值 777,777:

```
SELECT DATEPART(MICROSECOND, '2000/07/07 07:07:07.777777') FROM iq_dummy
```

下面的语句返回值 33,189:

```
SELECT DATEPART(MCS, '2009-11-03 11:10:42.033189') FROM iq_dummy
```

用法

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。

另请参见

- DATEADD 函数 [日期和时间]（第 146 页）
- DATECEILING 函数 [日期和时间]（第 148 页）
- DATEDIFF 函数 [日期和时间]（第 150 页）
- DATEFLOOR 函数 [日期和时间]（第 152 页）
- DATENAME 函数 [日期和时间]（第 156 页）
- DATEROUND 函数 [日期和时间]（第 158 页）
- 日期分量（第 102 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。
datetime-expression	包含要计算的值的日期、时间或日期时间表达式。

参数	描述
multiple-expression	(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。 如果 <i>multiple-expression</i> 计算结果为零、计算结果为负数、是一个显式空值常量，或者不是指定 <i>date-part</i> 的有效值，则 Sybase IQ 会产生错误。如果 <i>multiple-expression</i> 计算结果为空值，则该函数的结果也为空值。

示例

下面的语句返回 `August 13, 2009, 10:30.000AM`：

```
SELECT DATEROUND( MI, 'August 13, 2009 10:33.123AM', 10) FROM
iq_dummy
```

下面的语句返回 `August 13, 2009 10:32:35.456600 AM`：

```
SELECT DATEROUND( US, 'August 13, 2009, 10:32:35.456500AM', 200 )
FROM iq_dummy
```

下面的语句返回 `August 13, 2009 10:32:35.456789 AM`：

```
SELECT DATEROUND( US, 'August 13, 2009, 10:32:35.456789AM') FROM
iq_dummy
```

下面的语句返回 `August 13, 2009 10:32:35.456400 AM`：

```
SELECT DATEROUND( US, 'August 13, 2009, 10:32:35.456499AM', 200 )
FROM iq_dummy
```

用法

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

如果您对微秒、毫秒、秒、分钟或小时日期分量指定 *multiple-expression*，则 Sybase IQ 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，如果指定一个两分钟的倍数，则 Sybase IQ 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、季度或年日期分量指定 *multiple-expression*，则 Sybase IQ 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 Sybase IQ 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 Sybase IQ 舍入为周日期分量的倍数，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 — Adaptive Server Enterprise 或 SQL Anywhere 不支持。

另请参见

- DATEADD 函数 [日期和时间] (第 146 页)
- DATECEILING 函数 [日期和时间] (第 148 页)
- DATEDIFF 函数 [日期和时间] (第 150 页)
- DATEFLOOR 函数 [日期和时间] (第 152 页)
- DATEPART 函数 [日期和时间] (第 157 页)
- DATENAME 函数 [日期和时间] (第 156 页)
- 日期分量 (第 102 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

表 52. 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

示例

下面的语句返回含有值 1998-09-09 12:12:12.000 的时间戳：

```
SELECT DATETIME( '1998-09-09 12:12:12.000' ) FROM iq_dummy
```

标准和兼容性

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

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 53. 参数

参数	描述
date-expression	日期。

返回

SMALLINT

示例

下面的语句返回值 12:

```
SELECT DAY( '2001-09-12' ) 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。

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 54. 参数

参数	描述
date-expression	日期。

返回

VARCHAR

示例

下面的语句返回值 Saturday:

```
SELECT DAYNAME ( '1987/05/02' ) FROM iq_dummy
```

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。

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 55.

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。

示例

下面的语句返回整数值 729948：

```
SELECT DAYS( '1998-07-13 06:07:12' ) FROM iq_dummy
```

下面的语句返回整数值 -366，这是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
```

下面的语句返回值 1999-07-14：

```
SELECT DAYS( CAST( '1998-07-13' AS DATE ), 366 )
FROM iq_dummy
```

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

- CAST 函数 [数据类型转换] (第 124 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

表 56. 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

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
```

标准和兼容性

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

另请参见

- COL_NAME 函数 [系统] (第 132 页)
- DB_NAME 函数 [系统] (第 165 页)
- DB_PROPERTY 函数 [系统] (第 166 页)
- NEXT_DATABASE 函数 [系统] (第 226 页)
- OBJECT_ID 函数 [系统] (第 232 页)
- OBJECT_NAME 函数 [系统] (第 232 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

表 57. 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

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
```

用法

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 Sybase IQ 实现的 Adaptive Server Enterprise 函数。

另请参见

- COL_NAME 函数 [系统]（第 132 页）
- DB_ID 函数 [系统]（第 164 页）
- DB_PROPERTY 函数 [系统]（第 166 页）
- NEXT_DATABASE 函数 [系统]（第 226 页）
- OBJECT_ID 函数 [系统]（第 232 页）
- OBJECT_NAME 函数 [系统]（第 232 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 58. 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

示例

下面的语句返回当前数据库的页大小（以字节为单位）。

```
SELECT DB_PROPERTY( 'PAGESIZE' ) FROM iq_dummy
```

用法

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。

另请参见

- COL_NAME 函数 [系统]（第 132 页）
- DB_ID 函数 [系统]（第 164 页）
- DB_NAME 函数 [系统]（第 165 页）
- NEXT_DATABASE 函数 [系统]（第 226 页）
- OBJECT_ID 函数 [系统]（第 232 页）

- OBJECT_NAME 函数 [系统] (第 232 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 59. 参数

参数	描述
<i>numeric-expression</i>	以弧度表示的角度。

返回

返回由 *numeric-expression* 所指定的角的度数。

DOUBLE

示例

下面的语句返回值 29.793805:

```
SELECT DEGREES( 0.52 ) 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。

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 60. 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

示例

下面的语句说明了 **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

用法

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** 子句指定对其执行排序的参数以及每组中行的排序顺序。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。

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 — Adaptive Server Enterprise 或 SQL Anywhere 不支持。

另请参见

- [RANK 函数 \[分析\]](#)（第 247 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 61. 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 278 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 62. 参数

参数	描述
date-expression	日期。

返回

SMALLINT

示例

下面的语句返回值 5:

```
SELECT DOW( '1998-07-09' ) FROM iq_dummy
```

用法

如果需要星期一（或另一天）成为一周中的第一天，请参见《参考：语句和选项》>“按字母顺序排列的选项列表”>“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。

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

表 63. 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

示例
下面的语句返回 SQLCODE -813 的错误消息：

```
select errormsg( -813 )
```

返回值
包含错误消息的字符串。如果未提供参数，则会提供当前状态的错误消息。将创建任何替代项（例如表名和列名）。
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 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 — Adaptive Server Enterprise 不支持。

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。
请参见《参考：语句和选项》> “SQL 语句”> “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 64. 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 65. 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 显示系统临时文件（由 TEMP 或 IQTMP15 环境变量指向该文件）而不是 IQ 临时存储中的可用空间。
TempFreeSpace	兆字节	TempDiskSpace	
TempSize	兆字节	GrowTemp	

返回

INT

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
```

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_PARAMETER 函数 [系统] (第 173 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。

请参见《参考：语句和选项》>“SQL 语句”>“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

用法

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。

请参见《参考：语句和选项》> “SQL 语句”> “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 66. 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 connection_property('id')
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 171 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 67. 参数

参数	描述
<code>numeric-expression</code>	指数。

返回

DOUBLE

示例

下面的语句返回值 3269017.3724721107:

```
SELECT EXP( 15 ) 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。

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。

用法

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。

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，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。

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。

示例

下面的示例返回位于佛罗里达州的员工的工资指数加权平均值，其中最近聘用的员工的工资占该平均值的权重最大。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 68. 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
100	Gowda	59,840.000	60,621.875000
400	Francis	53,870.000	61,403.750000

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

另请参见

- `WEIGHTED_AVG` 函数 [集合] （第 318 页）
- 窗口化集合函数的用法 （第 95 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
表达式	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。

用法

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。

示例

下面的示例返回每位员工的工资与在同一部门中最近聘用的员工工资之间的关系（用百分比表示）：

```
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 69. FIRST_VALUE 结果集

DepartmentID	EmployeeID	Percentage
500	1,658	100.000000000000000000000000000000
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	100.000000000000000000000000000000
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

在此示例中，员工 1658 为部门 500 的第一行，表示员工 1658 是该部门中最近聘用的，因此收到的百分比为 100%。部门 500 中的其余员工的百分比是相对于员工 1658 的百分比进行计算的。例如，员工 1570 的收入大约是员工 1658 的收入的 139%。

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。

另请参见

- 窗口化集合函数的用法（第 95 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 70. 参数

参数	描述
numeric-expression	数字，通常是浮点数。

返回

DOUBLE

示例

下面的语句返回值 123.00:

```
SELECT FLOOR ( 123 ) FROM iq_dummy
```

下面的语句返回值 123:

```
SELECT FLOOR ( 123.45 ) FROM iq_dummy
```

下面的语句返回值 -124.00。

```
SELECT FLOOR ( -123.45 ) 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。

另请参见

- CEILING 函数 [数值]（第 126 页）
- CEIL 函数 [数值]（第 125 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

示例
下面的语句返回系统日期和时间。

```
SELECT GETDATE( ) FROM iq_dummy
```

用法
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。

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 SELECT 语句，但也可以是 UPDATE 或 DELETE 、 INSERT SELECT 或 SELECT INTO 语句。
statistics-level	一个整数。statistics-level 可以是： • 0 - 仅优化程序估计值（缺省值）。 • 2 - 包括节点统计在内的详细统计。 • 3 - 详细统计。
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。

参数	描述
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件许可证，或使用 **CAST** 并将 **GRAPHICAL_PLAN** 设置为正确的数据类型和大小。

用法

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，DBA 必须针对该用户将 **QUERY_PLAN_TEXT_ACCESS** 选项设置为 ON。

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”>“**QUERY_PLAN_TEXT_ACCESS** 选项”。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”>“**QUERY_PLAN_TEXT_CACHING** 选项”。

如果 **QUERY_PLAN_TEXT_ACCESS** 为 ON，并且服务器上维护的高速缓存提供有字符串表达式所对应的查询计划，则系统会将查询计划从高速缓存返回给您。

如果高速缓存中未提供有查询计划，并且您有权在客户端上查看计划，则系统会生成带有优化程序估计值的查询计划（**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。

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”>“**NOEXEC** 选项”。

当用户请求尚未执行的查询计划时，系统不会在高速缓存中提供该查询计划。而会返回带有优化程序估计值但无 **QUERY_PLAN_AFTER_RUN** 统计信息的查询计划。

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”>“**QUERY_PLAN_AFTER_RUN** 选项”。

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 Sybase IQ 查询打开的游标的查询计划。可分别使用 **DECLARE CURSOR** 和 **OPEN CURSOR** 声明和打开游标。若要获取最近打开 游标的查询计划，请使用：

```
SELECT GRAPHICAL_PLAN ( );
```

在 **QUERY_PLAN_AFTER_RUN** 选项为 **OFF** 的情况下，该计划会在 **OPEN CURSOR** 或 **CLOSE CURSOR** 后显示。但是，如果 **QUERY_PLAN_AFTER_RUN** 为 **ON**，则必须先执行 **CLOSE CURSOR**，然后才能请求该计划。

有关在 Interactive SQL 的“计划查看器”窗口中查看针对 SQL 语句的 查询优化程序执行计划的信息，请参见“SQL Anywhere Server - 数据库管理”>“管理数据库”>“SQL Anywhere 图形管理工具”>“使用 Interactive SQL”>“使用计划查看器查看图形计划”。

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、设置为 **ASIS** 的 **HEXADECIMAL** 子句来获取已设置格式的 计划输出，则系统会在不执行任何转义的情况下写入字符值，即使值包含控制字符也是如此。**ASIS** 对于包含格式设置字符（如制表符或回车）的文本很有用。

```
SELECT GRAPHICAL_PLAN ('SELECT * FROM Employees');OUTPUT to 'C:\gplan.xml' HEXADECIMAL ASIS quote '';
```

下面的示例从高速缓存中返回查询计划（如果有）：

```
SELECT GRAPHICAL_PLAN ( );
```

标准和兼容性

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

另请参见

- **HTML_PLAN** 函数 [字符串]（第 190 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 ROLLUP 或 CUBE 关键字的 GROUP BY 子句的查询结果集中）。该函数标识通过 ROLLUP 或 CUBE 操作添加到结果集中的小计行。

当前，Sybase IQ 不支持使用 **PERCENTILE_CONT** 或 **PERCENTILE_DISC** 函数 来执行 **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 — Adaptive Server Enterprise 不支持。

另请参见

- 窗口化集合函数的用法（第 95 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
group-name-string-expression	标识要考虑的组。
user-name-string-expression	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 71. 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。输入内容可以采用以下形式，前缀中包括小写或大写 “0x”，或者不使用前缀： <i>0xhex-string</i> <i>0Xhex-string</i> <i>hex-string</i>

示例

下面的语句返回值 4294967287：

```
SELECT HEXTOBIGINT ( '0xffffffff7' ) FROM iq_dummy  
SELECT HEXTOBIGINT ( '0Xffffffff7' ) FROM iq_dummy  
SELECT HEXTOBIGINT ( 'ffffffff7' ) FROM iq_dummy
```


用法

HEXTOBIGINT 函数接受十六进制整数，并返回 **BIGINT** 等效数字。可采用以下形式提供十六进制整数：**CHAR** 和 **VARCHAR** 值表达式以及 **BINARY** 和 **VARBINARY** 表达式。

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。

对于输入时出现的数据类型转换失败，Sybase IQ 会返回错误，但如果 **CONVERSION_ERROR** 选项设置为 OFF 则例外。当 **CONVERSION_ERROR** 为 OFF 时，输入无效的十六进制值后，系统将返回空值。

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]”。

在以下情况下，系统会返回错误：如果 **BINARY** 或 **VARBINARY** 值超过 8 字节，并且 **CHAR** 或 **VARCHAR** 值超过 16 个字符，后面附加了 “0x” 的值例外。

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

- **BIGINTTOHEX** 函数 [数据类型转换] (第 121 页)
- **HEXTOINT** 函数 [数据类型转换] (第 185 页)
- **INTTOHEX** 函数 [数据类型转换] (第 195 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
hexadecimal-string	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：
	<i>0xhex-string</i>
	<i>0Xhex-string</i>
	<i>hex-string</i>

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

示例

下面的语句返回值 420：

```
SELECT HEXTOINT ( '0x1A4' ) FROM iq_dummy
SELECT HEXTOINT ( '0X1A4' ) FROM iq_dummy
SELECT HEXTOINT ( '1A4' ) FROM iq_dummy
```

用法

如果输入无效的十六进制值，Sybase IQ 会返回错误，除非 **CONVERSION_ERROR** 选项为 OFF。当 **CONVERSION_ERROR** 为 OFF 时，输入无效的十六进制值后，系统将返回空值。

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]”。

数据库选项 **ASE_FUNCTION_BEHAVIOR** 指定 Sybase IQ 函数（包括 **INTTOHEX** 和 **HEXTOINT**）的输出与 Adaptive Server Enterprise 函数的输出一致。

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”>“**ASE_FUNCTION_BEHAVIOR** 选项”。

如果 **ASE_FUNCTION_BEHAVIOR** 选项为 ON：

- Sybase IQHEXTPOINT 假设 输入的是一个包含 8 个字符的十六进制字符串； 如果长度短于 8 个字符，则系统会用零填充字符串左侧。
- Sybase IQHEXTPOINT 最多接受前缀为 0x 的 16 个字符（共 18 个字符）； 使用时请小心，因为较大的输入值会导致整数值 溢出 32 位带符号整数输出的大小。
- 系统假设 Sybase IQHEXTPOINT 函数的输出的数据类型为 32 位带符号整数。
- Sybase IQHEXTPOINT 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，Sybase IQHEXTPOINT 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 Enterprise 兼容。

另请参见

- BIGINTTOHEX 函数 [数据类型转换]（第 121 页）
- HEXTOBIGINT 函数 [数据类型转换]（第 184 页）
- INTTOHEX 函数 [数据类型转换]（第 195 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 72. 参数

参数	定义
<i>datetime-expression</i>	日期/时间。

返回

SMALLINT

示例

下面的语句返回值 21:

```
SELECT HOUR( '1998-07-09 21:12:13' ) FROM iq_dummy
```

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。

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 73. 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

示例

下面的语句返回值 17518758:

```
SELECT HOURS( '1998-07-13 06:07:12' ) FROM iq_dummy
```

下面的语句返回值 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
```

下面的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
```

用法

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。

另请参见

- CAST 函数 [数据类型转换] (第 124 页)
- CONVERT 函数 [数据类型转换] (第 134 页)
- MINUTES 函数 [日期和时间] (第 218 页)
- MONTHS 函数 [日期和时间] (第 221 页)
- REPLACE 函数 [字符串] (第 261 页)
- SECOND 函数 [日期和时间] (第 270 页)
- WEEKS 函数 [日期和时间] (第 317 页)
- YEAR 函数 [日期和时间] (第 321 页)
- YEARS 函数 [日期和时间] (第 322 页)

HTML_DECODE 函数 [HTTP]

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 )
```

注意： 需要考虑 CIS 函数补偿性能方面的一些事宜。

HTML_DECODE 函数是 SQL Anywhere 函数。请参见“SQL Anywhere Server - SQL 参考” > “SQL 函数” > “函数” > “HTML_DECODE 函数 [HTTP]”。

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 **HTML_DECODE**，您必须具有非结构化数据分析选件许可证，或使用 **CAST** 并将 **HTML_DECODE** 设置为正确的数据类型和大小。

HTML_ENCODE 函数 [HTTP]

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 )
```

注意： 需要考虑 CIS 函数补偿性能方面的一些事宜。

HTML_ENCODE 函数是 SQL Anywhere 函数。请参见“SQL Anywhere Server - SQL 参考” > “SQL 函数” > “函数” > “HTML_ENCODE 函数 [HTTP]”。

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 **HTML_ENCODE**，您必须具有非结构化数据分析选项许可证，或使用 **CAST** 并将 **HTML_ENCODE** 设置为正确的数据类型和大小。

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 SQL 语句。该语句主要是 SELECT 语句，但也可以是 UPDATE 或 DELETE 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，并且服务器上维护的高速缓存提供有字符串表达式所对应的查询计划，则系统会将查询计划从高速缓存返回给您。

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”>“**QUERY_PLAN_TEXT_ACCESS** 选项”。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”>“**QUERY_PLAN_TEXT_CACHING** 选项”。

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 Sybase IQ 查询打开的游标的查询计划。若要获取最近打开的游标的查询计划，请使用：

SELECT HTML_PLAN () ;

在 **QUERY_PLAN_AFTER_RUN** 选项为 OFF 的情况下，该计划会在 **OPEN CURSOR** 或 **CLOSE CURSOR** 后显示。但是，如果 **QUERY_PLAN_AFTER_RUN** 为 ON，则必须先执行 **CLOSE CURSOR**，然后才能请求该计划。

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”>“QUERY_PLAN_AFTER_RUN 选项”。

有关在 Interactive SQL 的“计划查看器”窗口中查看针对 SQL 语句的查询优化程序执行计划的信息，请参见“SQL Anywhere Server - 数据库管理”>“管理数据库”>“SQL Anywhere 图形管理工具”>“使用 Interactive SQL”>“使用计划查看器查看图形计划”。

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.
```

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 的子句 **HEXADECIMAL ASIS** 对于包含格式设置字符（如制表符或回车）的文本很有用。设置为 **ASIS** 时，值按原样写入，不发生任何转义，即使值中包含控制字符也是如此。

下面的示例从高速缓存中返回 **HTML** 查询计划（如果有）。

```
SELECT HTML_PLAN ( );
```

标准和兼容性

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

另请参见

- GRAPHICAL_PLAN 函数 [字符串]（第 180 页）

HTTP_DECODE 函数 [HTTP]

对与 HTTP 一起使用的字符串中的特殊字符进行解码。

语法

```
HTTP_DECODE ( string )
```

注意： 需要考虑 CIS 函数补偿性能方面的一些事宜。

HTTP_DECODE 函数是 SQL Anywhere 函数。请参见“SQL Anywhere Server - SQL 参考”>“SQL 函数”>“函数”>“HTML_DECODE 函数 [HTTP]”。

HTTP_ENCODE 函数 [HTTP]

对与 HTTP 一起使用的字符串中的特殊字符进行编码。

语法

```
HTTP_ENCODE ( string )
```

注意： 需要考虑 CIS 函数补偿性能方面的一些事宜。

HTTP_ENCODE 函数是 SQL Anywhere 函数。请参见“SQL Anywhere Server - SQL 参考” > “SQL 函数” > “函数” > “HTTP_ENCODE 函数 [HTTP]”。

HTTP_HEADER 函数 [HTTP]

获取 HTTP 标头的值。

语法

```
HTTP_HEADER ( field-name )
```

注意： 需要考虑 CIS 函数补偿性能方面的一些事宜。

HTTP_HEADER 函数是 SQL Anywhere 函数。请参见“SQL Anywhere Server - SQL 参考” > “SQL 函数” > “函数” > “HTTP_HEADER 函数 [HTTP]”。

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 **HTTP_HEADER**，您必须具有非结构化数据分析选件许可证，或使用 **CAST** 并将 **HTTP_HEADER** 设置为正确的数据类型和大小。

HTTP_VARIABLE 函数 [HTTP]

获取一个 HTTP 变量的值。

语法

```
HTTP_VARIABLE ( var-name [ [ , instance ] , header-field ] )
```

注意： 需要考虑 CIS 函数补偿性能方面的一些事宜。

HTTP_VARIABLE 函数是 SQL Anywhere 函数。请参见“SQL Anywhere Server - SQL 参考” > “SQL 函数” > “函数” > “HTTP_VARIABLE 函数 [HTTP]”。

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 **HTTP_VARIABLE**，您必须具有非结构化数据分析选件许可证，或使用 **CAST** 并将 **HTTP_VARIABLE** 设置为正确的数据类型和大小。

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 74. 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。

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
```

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。

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 Sybase IQ 实现的 Adaptive Server Enterprise 函数。

另请参见

- OBJECT_ID 函数 [系统]（第 232 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

表 75. 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件许可证，或使用 **CAST** 并将 **INSERTSTR** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “backoffice”：

```
SELECT INSERTSTR( 0, 'office ', 'back' ) FROM iq_dummy
```

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase — Adaptive Server Enterprise 中不支持。STUFF 函数具有相同的作用，并且在 Adaptive Server Enterprise 和 Sybase IQ 中均受支持。

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

表 76. 参数

参数	描述
integer-expression	要转换成十六进制的整数。

返回

VARCHAR

示例

下面的语句返回值 3B9ACA00：

```
SELECT INTTOHEX( 1000000000 ) FROM iq_dummy
```

下面的语句返回值 00000002540BE400：

```
SELECT INTTOHEX ( 10000000000 ) FROM iq_dummy
```

用法

如果用于 **INTTOHEX** 转换的输入的数据转换失败，则 Sybase IQ 返回错误，除非 **CONVERSION_ERROR** 选项为 **OFF**。在这种情况下，结果为空。

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]”。

数据库选项 `ASE_FUNCTION_BEHAVIOR` 指定 Sybase IQ 函数（包括 `INTTOHEX` 和 `HEXTOINT`）的输出与 Adaptive Server Enterprise 函数的输出一致。
`ASE_FUNCTION_BEHAVIOR` 的缺省值为 `OFF`。

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”>“`ASE_FUNCTION_BEHAVIOR` 选项”。

如果禁用 `ASE_FUNCTION_BEHAVIOR` 选项（值为 `OFF`）：

- `INTTOHEX` 的输出与 SQL Anywhere 兼容。
- 根据输入，`INTTOHEX` 的输出可以是 8 位数字或 16 位数字，并用零填充左侧；返回数据类型是 `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。

如果启用 `ASE_FUNCTION_BEHAVIOR` 选项（值为 `ON`）：

- `INTTOHEX` 的输出与 Adaptive Server Enterprise 兼容。
- `INTTOHEX` 的输出始终是 8 位数字，并用零填充左侧；返回数据类型是 `VARCHAR`。
- `INTTOHEX` 的输出没有“0x”或“0X”前缀。
- Sybase IQ `INTTOHEX` 假设输入是一个 32 位带符号整数；更大的值可能会溢出，并且可能会导致发生转换错误。例如，语句：

```
SELECT INTTOHEX( 1000000000 ) FROM iq_dummy
```

将返回值 `3B9ACA00`。但语句：

```
SELECT INTTOHEX( 100000000000 ) FROM iq_dummy
```

导致发生转换错误。

标准和兼容性

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

另请参见

- `BIGINTTOHEX` 函数 [数据类型转换]（第 121 页）
- `HEXTOBIGINT` 函数 [数据类型转换]（第 184 页）
- `HEXTOINT` 函数 [数据类型转换]（第 185 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 77. 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

示例

下面的示例测试 **birth_date** 列是否含有有效日期，将无效日期以空值形式返回，而将有效日期以 **date** 格式返回。

```
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
```

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。
 - Adaptive Server Enterprise 不支持。

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 78. 参数

参数	描述
表达式	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。

示例

下面的语句返回值 -66:

```
SELECT ISNULL( NULL , -66, 55, 45, NULL, 16 ) FROM iq_dummy
```

用法

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。

另请参见

- **COALESCE** 函数 [杂项] (第 130 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 )
```

注意： 需要考虑 CIS 函数补偿性能方面的一些事宜。

参数

表 79. 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

用法

为获得最佳性能，请不要在谓词中使用 **ISNUMERIC**，因为在这种情况下，将由产品的 SQL Anywhere 部分对其进行处理，因而无法利用 Sybase IQ 的性能特性。

示例

下面的示例测试 **height_in_cms** 列是否含有有效数值数据，将无效数值数据以空值形式返回，而将有效数值数据以 **int** 格式返回。

```
data height_in_cms
-----
asde
asde
180
156

select case
  when isnumeric(height_in_cms)=0
  then NULL
  else cast(height_in_cms as int)
end
from MyData
```

标准和兼容性

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

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
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。
default	<i>offset</i> 值超过表或分区的基数范围时要返回的值。
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。
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。

用法

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 的缺省值是 **NULL**。*default* 的数据类型必须可隐式转换为 *value_expr* 值的数据类型，否则 Sybase IQ 将产生转换错误。

示例

下面的示例返回雇员表中的工资数据，按部门 ID 对数据进行分区，并根据雇员开始日期对数据进行排序。**LAG** 函数返回上一行中的工资值（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
...			

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

另请参见

- LEAD 函数 [分析] (第 204 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
表达式	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。

用法

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。

示例

下面的示例返回每位雇员的工资，以及该部门中工资最高的雇员的姓名：

```
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 80. LAST_VALUE 结果集

employee_name	Salary	DepartmentID	highest_paid
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
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
.....			

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。

另请参见

- 窗口化集合函数的用法（第 95 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

表 81. 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件许可证，或使用 **CAST** 并将 **LCASE** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “lower case”：

```
SELECT LCASE( 'LOWER Case' ) FROM iq_dummy
```

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

- Sybase — **LCASE** 在 Adaptive Server Enterprise 中不受支持； 您可以使用 **LOWER** 来获得相同的功能。

另请参见

- **LEFT** 函数 [字符串] (第 205 页)
- **LOWER** 函数 [字符串] (第 213 页)
- **REPLACE** 函数 [字符串] (第 261 页)
- **REVERSE** 函数 [字符串] (第 264 页)
- **RIGHT** 函数 [字符串] (第 265 页)
- **UCASE** 函数 [字符串] (第 308 页)
- **UPPER** 函数 [字符串] (第 309 页)

LEAD 函数 [分析]

返回表或表分区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
<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。
<i>default</i>	<i>offset</i> 值超过表或分区的范围时要返回的值。
<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。

用法

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 的缺省值是 **NULL**。*default* 的数据类型必须可隐式转换为 *value_expr* 值的数据类型，否则 Sybase IQ 将产生转换错误。

示例

下面的示例返回雇员表中的工资数据，按部门 ID 对数据进行分区，并根据雇员开始日期对数据进行排序。**LEAD** 函数返回下一行中的工资值（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

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

另请参见

- LAG 函数 [分析]（第 199 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 82. 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件许可证，或使用 **CAST** 并将 **LEFT** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “choco”：

```
SELECT LEFT( 'chocolate', 5 ) FROM iq_dummy
```

用法

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件许可证，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。

另请参见

- LCASE 函数 [字符串] (第 203 页)
- LOWER 函数 [字符串] (第 213 页)
- REPLACE 函数 [字符串] (第 261 页)
- REVERSE 函数 [字符串] (第 264 页)
- RIGHT 函数 [字符串] (第 265 页)
- UCASE 函数 [字符串] (第 308 页)
- UPPER 函数 [字符串] (第 309 页)

LEN 函数 [字符串]

采用一个参数作为 **BINARY** 或 **STRING** 类型的输入并返回由数据库的归类序列定义的、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 83. 参数

参数	描述
<i>string_expr</i>	要计算其值的字符串表达式。

示例

下面的示例返回值 3152:

```
select len(Photo) from Productswhere ID = 500
```

用法

此函数等效于 **CHAR_LENGTH** (*string_expression*)。

权限

任何用户都可以执行 **LEN**。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。

另请参见

- BIT_LENGTH 函数 [字符串] (第 121 页)
- BYTE_LENGTH 函数 [字符串] (第 122 页)
- CHAR_LENGTH 函数 [字符串] (第 128 页)
- COL_LENGTH 函数 [系统] (第 131 页)
- DATALENGTH 函数 [系统] (第 145 页)
- LENGTH 函数 [字符串] (第 207 页)
- OBJECT_NAME 函数 [系统] (第 232 页)
- OCTET_LENGTH 函数 [字符串] (第 233 页)
- STR_REPLACE 函数 [字符串] (第 287 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 84. 参数

参数	描述
string-expression	字符串。

返回

INT

示例

下面的语句返回值 9:

```
SELECT LENGTH( 'chocolate' ) FROM iq_dummy
```

用法

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。如果字符串是 **BINARY** 数据类型，则 **LENGTH** 函数的行为与 **BYTE_LENGTH** 的行为相同。

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。

另请参见

- **BIT_LENGTH** 函数 [字符串] (第 121 页)
- **BYTE_LENGTH** 函数 [字符串] (第 122 页)
- **CHAR_LENGTH** 函数 [字符串] (第 128 页)
- **COL_LENGTH** 函数 [系统] (第 131 页)
- **DATALength** 函数 [系统] (第 145 页)
- **LEN** 函数 [字符串] (第 206 页)
- **OBJECT_NAME** 函数 [系统] (第 232 页)
- **OCTET_LENGTH** 函数 [字符串] (第 233 页)
- **STR_REPLACE** 函数 [字符串] (第 287 页)

LIST 函数 [集合]

返回以逗号分隔的值列表。

语法

```
LIST(  
[DISTINCT] string-expresssion  
[, '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。

LIST 函数 [集合] 是 SQL Anywhere 函数。请参见“SQL Anywhere Server - SQL 参考” > “SQL 函数” > “函数” > “LIST 函数 [集合]”。

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件许可证，或使用 **CAST** 并将 **LIST** 设置为正确的数据类型和大小。

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	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， LN 函数会产生错误。返回值为 DOUBLE 数据类型。

用法

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 211 页)
- LOG10 函数 [数值] (第 212 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
string-expression1	要搜索的字符串。
string-expression2	要搜索的字符串。该字符串限制为 255 个字节。
numeric-expression	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， LOCATE 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* 是 32 位带符号整数。

返回

INT

示例

下面的语句返回值 8:

```
SELECT LOCATE( 'office party this week - rsvp as soon as possible',
'party', 2 ) FROM iq_dummy
```

在第二个示例中，要搜索的 *numeric-expression* 起始偏移量是负数。

```
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
```

用法

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应超过 255 个字节。如果第二个参数是长字符串，此函数返回空值。

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《Sybase IQ 中的非结构化数据分析》> “函数支持”。

标准和兼容性

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

另请参见

- PATINDEX 函数 [字符串]（第 234 页）
- LIKE 条件（第 38 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 85. 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**。

示例

下面的语句返回值 3.912023:

```
SELECT LOG( 50 ) 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。

另请参见

- LN 函数 [数值] (第 209 页)
- LOG10 函数 [数值] (第 212 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 86. 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。

示例

下面的语句返回值 1.698970。

```
SELECT LOG10( 50 ) 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。

另请参见

- LN 函数 [数值] (第 209 页)
- LOG 函数 [数值] (第 211 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 87. 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件许可证，或使用 **CAST** 并将 **LOWER** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “lower case”：

```
SELECT LOWER( 'LOWER Case' ) FROM iq_dummy
```

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

- LCASE 函数 [字符串]（第 203 页）
- LEFT 函数 [字符串]（第 205 页）
- REPLACE 函数 [字符串]（第 261 页）
- REVERSE 函数 [字符串]（第 264 页）
- RIGHT 函数 [字符串]（第 265 页）
- UCASE 函数 [字符串]（第 308 页）

- UPPER 函数 [字符串] (第 309 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件许可证，或使用 **CAST** 并将 **LTRIM** 设置为正确的数据类型和大小。

示例

下面的语句返回删除了所有前导空白的值 “Test Message”：

```
SELECT LTRIM( ' Test Message' ) 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。

另请参见

- RTRIM 函数 [字符串] (第 269 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
expression	要计算其最大值的表达式。这通常是列名称。
DISTINCT column-name	返回的值与 MAX (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。

示例

下面的语句返回值 138948.000，该值表示 **Employees** 表中的最高工资：

```
SELECT MAX ( Salary )
FROM Employees
```

用法

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。

另请参见

- 窗口化集合函数的用法（第 95 页）
- MIN 函数 [集合]（第 217 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。

用法

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**。

示例

下面的查询返回佛罗里达的每个部门的工资中值：

```
SELECT DepartmentID, Surname, Salary,
MEDIAN(Salary) OVER (PARTITION BY DepartmentID) "Median"
FROM Employees
WHERE State IN ('FL')
```

返回的结果为：

表 88. 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

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

另请参见

- 窗口化集合函数的用法（第 95 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。

示例

下面的语句返回值 24903.000，该值表示 Employees 表中的最低工资：

```
SELECT MIN ( Salary )
FROM Employees
```

用法

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。

另请参见

- 窗口化集合函数的用法（第 95 页）
- MAX 函数 [集合]（第 214 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
<code>datetime-expression</code>	日期/时间值。

返回

SMALLINT

示例

下面的语句返回值 22:

```
SELECT MINUTE( '1998-07-13 12:22:34' ) 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。

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
<code>datetime-expression</code>	日期和时间。
<code>integer-expression</code>	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

示例

它返回值 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
```

用法

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。

另请参见

- CAST 函数 [数据类型转换] (第 124 页)
- CONVERT 函数 [数据类型转换] (第 134 页)
- HOURS 函数 [日期和时间] (第 188 页)
- MONTHS 函数 [日期和时间] (第 221 页)
- REPLACE 函数 [字符串] (第 261 页)
- SECOND 函数 [日期和时间] (第 270 页)
- WEEKS 函数 [日期和时间] (第 317 页)
- YEAR 函数 [日期和时间] (第 321 页)
- YEARS 函数 [日期和时间] (第 322 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

示例

下面的语句返回值 2:

```
SELECT MOD( 5, 3 ) FROM iq_dummy
```

用法

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 — Adaptive Server Enterprise 中不支持。 % 运算符在 Adaptive Server Enterprise 中作为模运算符使用。

另请参见

- REMAINDER 函数 [数值] (第 259 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

示例

下面的语句返回值 7:

```
SELECT MONTH( '1998-07-13' ) FROM iq_dummy
```

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。

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
<i>date-expression</i>	日期时间值。

返回

VARCHAR

示例

当 **DATE_ORDER** 选项设置为缺省值 *ymd* 时，下面的语句返回值 **September**。

```
SELECT MONTHNAME( '1998-09-05' ) FROM iq_dummy
```

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”>“**DATE_ORDER** 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。

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
date-expression	日期和时间。
integer-expression	要添加到 <i>date-expression</i> 中的月数。如果 <i>integer-expression</i> 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 <i>date-expression</i> 显式转换为 datetime 数据类型。

返回

INT

TIMESTAMP

示例

下面的语句返回值 23982:

```
SELECT MONTHS( '1998-07-13 06:07:12' ) FROM iq_dummy
```

下面的语句返回值 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
```

下面的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
```

用法

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。

另请参见

- **CAST** 函数 [数据类型转换] (第 124 页)
- **CONVERT** 函数 [数据类型转换] (第 134 页)
- **HOURS** 函数 [日期和时间] (第 188 页)
- **MINUTES** 函数 [日期和时间] (第 218 页)
- **REPLACE** 函数 [字符串] (第 261 页)
- **SECOND** 函数 [日期和时间] (第 270 页)
- **WEEKS** 函数 [日期和时间] (第 317 页)
- **YEAR** 函数 [日期和时间] (第 321 页)
- **YEARS** 函数 [日期和时间] (第 322 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

示例

下面的语句创建表 **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，但前提是该列的当前值为空值。

```
CREATE TABLE t1 (uid_col int);
UPDATE t1
    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。

用法

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
- **INSERT...VALUES** 的 **VALUES** 子句

可以使用 **NEWID** 函数生成的值作为 Sybase IQ 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 — Adaptive Server Enterprise 不支持。

另请参见

- 二进制数据类型 （第 72 页）
- STRTOUUID 函数 [字符串] （第 290 页）
- UUIDTOSTR 函数 [字符串] （第 311 页）
- 字符数据类型 （第 65 页）
- 二进制数据类型 （第 618 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 NEXT_CONNECTION 调用返回。如果 <i>connection-id</i> 为空值，则 NEXT_CONNECTION 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 NEXT_CONNECTION 返回下一个连接，而不考虑数据库。

用法

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 );
```

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
<i>database-id</i>	一个整数，它指定数据库的 ID 号。

返回

INT

示例

下面的语句返回值 0，这是第一个数据库值：

```
SELECT NEXT_DATABASE( NULL ) FROM iq_dummy
```

下面的语句返回空值，表示服务器上没有其它的数据库：

```
SELECT NEXT_DATABASE( 0 ) FROM iq_dummy
```

用法

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。

另请参见

- COL_NAME 函数 [系统]（第 132 页）
- DB_ID 函数 [系统]（第 164 页）
- DB_NAME 函数 [系统]（第 165 页）
- DB_PROPERTY 函数 [系统]（第 166 页）
- OBJECT_ID 函数 [系统]（第 232 页）
- OBJECT_NAME 函数 [系统]（第 232 页）

NEXT_HTTP_HEADER 函数 [HTTP]

获得下一个 HTTP 标头名称。

语法

```
NEXT_HTTP_HEADER ( header-name )
```

注意： 需要考虑 CIS 函数补偿性能方面的一些事宜。

NEXT_HTTP_HEADER 函数是 SQL Anywhere 函数。请参见“SQL Anywhere Server - SQL 参考” > “SQL 函数” > “函数” > “NEXT_HTTP_HEADER 函数 [HTTP]”。

注意： 此参考指向 SQL Anywhere 文档。

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 NEXT_HTTP_HEADER，您必须具有非结构化数据分析选件许可证，或使用 **CAST** 并将 NEXT_HTTP_HEADER 设置为正确的数据类型和大小。

NEXT_HTTP_VARIABLE 函数 [HTTP]

获得下一个 HTTP 变量的名称。

语法

```
NEXT_HTTP_VARIABLE ( var-name )
```

注意： 需要考虑 CIS 函数补偿性能方面的一些事宜。

NEXT_HTTP_VARIABLE 函数是 SQL Anywhere 函数。请参见“SQL Anywhere Server - SQL 参考” > “SQL 函数” > “函数” > “NEXT_HTTP_VARIABLE 函数 [HTTP]”。

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 NEXT_HTTP_VARIABLE，您必须具有非结构化数据分析选件许可证，或使用 **CAST** 并将 NEXT_HTTP_VARIABLE 设置为正确的数据类型和大小。

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

示例

下面的语句返回当前日期和时间。

```
SELECT NOW(*) FROM iq_dummy
```

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。

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。

示例

下面的示例使用 **NTILE** 函数确定汽车经销商的销售状态。将经销商根据各自销售的汽车数量分为四组。根据汽车销售额排名，**ntile** 为 1 的经销商为所有经销商中最靠前的 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

若要按销售量查找处于前 10% 的汽车经销商，请在示例 **SELECT** 语句中指定 **NTILE(10)**。类似地，若要按销售量查找处于前 50% 的汽车经销商，请指定 **NTILE(2)**。

用法

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** 子句指定对其执行排序的参数以及每组中行的排序顺序。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 — Adaptive Server Enterprise 或 SQL Anywhere 不支持

另请参见

- PERCENTILE_CONT 函数 [分析]（第 237 页）
- PERCENTILE_DISC 函数 [分析]（第 239 页）
- YEAR 函数 [日期和时间]（第 321 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。

示例

下面的语句返回 a:

```
SELECT NULLIF( 'a', 'b' ) FROM iq_dummy
```

下面的语句返回空值:

```
SELECT NULLIF( 'a', 'a' ) FROM iq_dummy
```

用法

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。

另请参见

- CASE 表达式 (第 28 页)
- 用于简写 CASE 表达式的 NULLIF 函数 (第 29 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

用法

仅在选择列表中或者 UPDATE 语句的 SET 子句中使用 NUMBER 函数。例如，下面的语句使用比上一行大 1 的数字更新 seq_id 列的每一行。该数字按 ORDER BY 子句指定的顺序应用。

```
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。

示例

下面的语句返回此编号列表：

number(*)
1
2
3
4
5

```
SELECT NUMBER( * )
FROM Departments
WHERE DepartmentID > 10
```

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。

OBJECT_ID 函数 [系统]

返回对象 ID。

语法

```
OBJECT_ID ( object-name )
```

参数

表 89. 参数

参数	描述
object-name	对象的名称。

示例

下面的语句返回 *Customers* 表的对象 ID 100209:

```
SELECT OBJECT_ID ('CUSTOMERS') FROM iq_dummy
```

标准和兼容性

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

另请参见

- COL_NAME 函数 [系统] (第 132 页)
- DB_ID 函数 [系统] (第 164 页)
- DB_NAME 函数 [系统] (第 165 页)
- DB_PROPERTY 函数 [系统] (第 166 页)
- NEXT_DATABASE 函数 [系统] (第 226 页)
- OBJECT_NAME 函数 [系统] (第 232 页)
- INDEX_COL 函数 [系统] (第 193 页)

OBJECT_NAME 函数 [系统]

返回对象名称。

语法

```
OBJECT_NAME ( object-id [ , database-id ] )
```


参数

表 90. 参数

参数	描述
object-id	对象 ID。
database-id	数据库 ID。

示例

下面的语句返回名称 “customer”：

```
SELECT OBJECT_NAME ( 100209 ) FROM iq_dummy
```

标准和兼容性

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

另请参见

- BIT_LENGTH 函数 [字符串] (第 121 页)
- BYTE_LENGTH 函数 [字符串] (第 122 页)
- CHAR_LENGTH 函数 [字符串] (第 128 页)
- COL_LENGTH 函数 [系统] (第 131 页)
- DATALENGTH 函数 [系统] (第 145 页)
- LEN 函数 [字符串] (第 206 页)
- LENGTH 函数 [字符串] (第 207 页)
- OCTET_LENGTH 函数 [字符串] (第 233 页)
- STR_REPLACE 函数 [字符串] (第 287 页)
- COL_NAME 函数 [系统] (第 132 页)
- DB_ID 函数 [系统] (第 164 页)
- DB_NAME 函数 [系统] (第 165 页)
- DB_PROPERTY 函数 [系统] (第 166 页)
- NEXT_DATABASE 函数 [系统] (第 226 页)
- OBJECT_ID 函数 [系统] (第 232 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。

用法

空值参数的返回值为空。

OCTET_LENGTH 函数支持 所有 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《Sybase IQ 中的非结构化数据分析》>“函数支持”。

标准和兼容性

- Sybase — SQL Anywhere 或 Adaptive Server Enterprise 不支持。

另请参见

- BIT_LENGTH 函数 [字符串] (第 121 页)
- BYTE_LENGTH 函数 [字符串] (第 122 页)
- CHAR_LENGTH 函数 [字符串] (第 128 页)
- COL_LENGTH 函数 [系统] (第 131 页)
- DATALENGTH 函数 [系统] (第 145 页)
- LEN 函数 [字符串] (第 206 页)
- LENGTH 函数 [字符串] (第 207 页)
- OBJECT_NAME 函数 [系统] (第 232 页)
- STR_REPLACE 函数 [字符串] (第 287 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， PATINDEX 返回一 (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

示例

下面的语句返回值 2:

```
SELECT PATINDEX( '%hoco%', 'chocolate' ) FROM iq_dummy
```

下面的语句返回值 11:

```
SELECT PATINDEX ( '%4_5_', '0a1A 2a3A 4a5A' ) FROM iq_dummy
```

用法

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 91. 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 32 位不带符号的整数位置。

如果您有权使用非结构化数据分析功能，则可以将此函数与大对象数据一起使用。

请参见《Sybase IQ 中的非结构化数据分析》>“函数支持”。

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

- LIKE 条件（第 38 页）
- LOCATE 函数 [字符串]（第 210 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。

示例

下面的语句说明了 **PERCENT_RANK** 函数的用法：

```
SELECT s_supkey, SUM(s_acctBal) AS sum_acctBal,
PERCENT_RANK() OVER ( ORDER BY SUM(s_acctBal) DESC )
AS percent_rank_all FROM supplier GROUP BY s_supkey;
```

s_supkey	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

用法

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

$$(R_x - 1) / (N_{totalRow} - 1)$$

其中 R_x 是组中某行的秩位置， $N_{totalRow}$ 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** 子句指定对其执行排序的参数以及每组中行的排序顺序。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 — Adaptive Server Enterprise 或 SQL Anywhere 不支持。

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。

示例

下面的示例使用 **PERCENTILE_CONT** 函数确定某区域中要进入前 10% 应该达到的汽车销售量。

示例中使用了下面的数据集：

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

下面的 **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中要进入前 10 个百分点应该达到的汽车销售量：

region	percentile_cont	Northeast	840
Northwest	740	South	470

用法

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）指定对其执行百分点函数的表达式以及每组中行的排序顺序。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，而不是用于 **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 — Adaptive Server Enterprise 或 SQL Anywhere 不支持。

另请参见

- NTILE 函数 [分析] (第 228 页)
- PERCENTILE_DISC 函数 [分析] (第 239 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。
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。

示例

下面的示例使用 **PERCENTILE_DISC** 函数确定某区域中要进入前 10% 应该达到的汽车销售量。

示例中使用了下面的数据集：

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

下面的 **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中要进入前 10 个百分点应该达到的汽车销售量：

region	percentile_cont	Northeast	900
Northwest	800	South	500

用法

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_DISC** 支持可在 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）指定对其执行百分点函数的表达式以及每组中行的排序顺序。此 **ORDER BY** 子句只在 **WITHIN GROUP** 子句中使用，而不是用于 **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 — Adaptive Server Enterprise 或 SQL Anywhere 不支持。

另请参见

- NTILE 函数 [分析]（第 228 页）
- PERCENTILE_CONT 函数 [分析]（第 237 页）

PI 函数 [数值]

返回数值 PI。

语法

```
PI ( * )
```

返回

DOUBLE

示例

下面的语句返回值 3.141592653……

```
SELECT PI ( * ) FROM iq_dummy
```

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase — Adaptive Server Enterprise 中支持 **PI()** 函数，但 **PI(*)** 则不受支持。

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

示例

下面的语句返回值 64:

```
SELECT Power( 2, 6 ) FROM iq_dummy
```

用法

将 *numeric-expression1* 增加至乘方 *numeric-expresson2*。

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。

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 92.

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 PROPERTY_NUMBER 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

示例

下面的语句返回当前数据库服务器的名称：

```
SELECT PROPERTY( 'Name' ) FROM iq_dummy
```

用法

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。

另请参见

- PROPERTY_NAME 函数 [系统]（第 243 页）
- PROPERTY_NUMBER 函数 [系统]（第 244 页）
- CONNECTION_PROPERTY 函数 [系统]（第 133 页）

- 可用于服务器的属性 （第 111 页）
- 可用于每个数据库的属性 （第 111 页）
- 连接属性 （第 110 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 PROPERTY_NUMBER 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

示例

下面的语句返回说明“索引插入数”：

```
SELECT PROPERTY_DESCRIPTION( 'IndAdd' ) FROM iq_dummy
```

用法

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。

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

示例

下面的语句返回与属性号 126 关联的属性。它指向的实际属性因版本而异。

```
SELECT PROPERTY_NAME( 126 ) FROM iq_dummy
```

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

- PROPERTY 函数 [系统] (第 242 页)
- PROPERTY_NUMBER 函数 [系统] (第 244 页)
- CONNECTION_PROPERTY 函数 [系统] (第 133 页)
- 可用于服务器的属性 (第 111 页)
- 可用于每个数据库的属性 (第 111 页)
- 连接属性 (第 110 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
property-name	属性名称。

返回

INT

示例

下面的语句返回一个整数值。实际的值因版本而异。

```
SELECT PROPERTY_NUMBER( 'PAGESIZE' ) FROM iq_dummy
```

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

- PROPERTY 函数 [系统] (第 242 页)
- PROPERTY_NAME 函数 [系统] (第 243 页)
- CONNECTION_PROPERTY 函数 [系统] (第 133 页)
- 可用于服务器的属性 (第 111 页)
- 可用于每个数据库的属性 (第 111 页)
- 连接属性 (第 110 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

示例

如果将 **DATE_ORDER** 选项设置为缺省值 *ymd*，则下面的语句返回值 2：

```
SELECT QUARTER ( '1987/05/02' ) FROM iq_dummy
```

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”>“DATE_ORDER 选项”。

用法

下表列出了一年的各个季度中的日期。

表 93. 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。

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

示例

下面的语句返回 0.5236（近似值）：

```
SELECT RADIANS( 30 ) FROM iq_dummy
```

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。

RAND 函数 [数值]

返回一个 DOUBLE 精度的随机数字 x，其中 0 <= 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

示例

下面的语句返回某个表的 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;
```

下面的语句返回 941392926249216914（近似值）：

```
SELECT RAND( 4 ) FROM iq_dummy
```

用法

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： **RAND** 返回的值会因您是否使用 **FROM** 子句以及被引用表是在 **SYSTEM** 中创建的还是在 **IQ** 存储中创建的而异。

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。

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

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

用法

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** 子句指定对其执行排序的参数以及每组中行的排序顺序。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 Enterprise 或 SQL Anywhere 不支持。

另请参见

- DENSE_RANK 函数 [分析] (第 167 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

用法

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*)

请参见“SQL Anywhere Server - SQL 用法” > “查询和修改数据” > “OLAP 支持” > “SQL Anywhere 中的窗口函数” > “行计算函数” > “集合函数的数学公式”。

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。

示例

下面的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;
```

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。

另请参见

- 窗口化集合函数的用法（第 95 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
dependent-expression	受独立变量影响的变量。
independent-expression	影响结果的变量。

返回

DOUBLE

用法

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)
```

请参见“SQL Anywhere Server - SQL 用法”>“查询和修改数据”>“OLAP 支持”>“SQL Anywhere 中的窗口函数”>“行计算函数”>“集合函数的数学公式”。

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。

示例

下面的示例计算独立变量雇员工资的平均值：以下函数返回值 49988.6232：

```
SELECT REGR_AVGY( Salary, ( YEAR( NOW( ) ) - YEAR( BirthDate ) ) )FROM Employees;
```

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。

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
<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

INTEGER

用法

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。

示例

下面的示例返回一个值，表示用于拟合回归线的非空值对的数量。以下函数返回 75：

```
SELECT REGR_COUNT( Salary, ( YEAR( NOW() ) -  
YEAR( BirthDate ) ) )FROM Employees;
```

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。

另请参见

- 窗口化集合函数的用法（第 95 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
<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

用法

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*：

$$\text{AVG}(y) - \text{REGR\_SLOPE}(y, x) * \text{AVG}(x)$$

请参见“SQL Anywhere Server - SQL 用法” > “查询和修改数据” > “OLAP 支持” > “SQL Anywhere 中的窗口函数” > “行计算函数” > “集合函数的数学公式”。

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。

示例

下面的示例返回值 1874.5805688517603：

```
SELECT REGR_INTERCEPT( Salary, ( YEAR( NOW() ) -
YEAR( BirthDate ) ) )FROM Employees;
```

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。

另请参见

- 窗口化集合函数的用法（第 95 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

用法

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。

在消除所有 *dependent-expression* 或 *independent-expression* 为空值的值对之后，系统会将 **REGR_R2** 应用于 *dependent-expression* 和 *independent-expression* 对的集合。

Sybase IQ 随后会应用下面的算法：

- 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*。

请参见“SQL Anywhere Server - SQL 用法”>“查询和修改数据”>“OLAP 支持”>“SQL Anywhere 中的窗口函数”>“行计算函数”>“集合函数的数学公式”。

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。

示例

下面的示例返回值 0.19379959710325653：

```
SELECT REGR_R2( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) )FROM
Employees;
```

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。

另请参见

- 窗口化集合函数的用法（第 95 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

用法

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)
```

请参见“SQL Anywhere Server - SQL 用法”>“查询和修改数据”>“OLAP 支持”>“SQL Anywhere 中的窗口函数”>“行计算函数”>“集合函数的数学公式”。

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。

示例

下面的示例返回值 935.3429749445614：

```
SELECT REGR_SLOPE( Salary, ( YEAR( NOW() ) -  
YEAR( BirthDate ) ) )FROM Employees;
```

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。

另请参见

- 窗口化集合函数的用法（第 95 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

用法

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)
```

请参见“SQL Anywhere Server - SQL 用法” > “查询和修改数据” > “OLAP 支持” > “SQL Anywhere 中的窗口函数” > “行计算函数” > “集合函数的数学公式”。

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。

示例

下面的示例返回值 5916.4800000000105：

```
SELECT REGR_SXX( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) )FROM  
Employees;
```


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。

另请参见

- 窗口化集合函数的用法（第 95 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

用法

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)
```

请参见“SQL Anywhere Server - SQL 用法” > “查询和修改数据” > “OLAP 支持” > “SQL Anywhere 中的窗口函数” > “行计算函数” > “集合函数的数学公式”。

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。

示例

下面的示例返回值 5533938.004400015。

```
SELECT REGR_SXY( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) )FROM Employees;
```

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。

另请参见

- 窗口化集合函数的用法（第 95 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
<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

用法

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。

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)
```

请参见“SQL Anywhere Server - SQL 用法” > “查询和修改数据” > “OLAP 支持” > “SQL Anywhere 中的窗口函数” > “行计算函数” > “集合函数的数学公式”。

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。

示例

下面的示例返回值 26、708、672、843.3002：

```
SELECT REGR_SYY( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) )FROM Employees;
```

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。

另请参见

- 窗口化集合函数的用法（第 95 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

示例

下面的语句返回值 2：

```
SELECT REMAINDER( 5, 3 ) FROM iq_dummy
```

用法

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。

另请参见

- MOD 函数 [数值] (第 219 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**，您必须具有非结构化数据分析选件许可证，或使用 **CAST** 并将 **REPEAT** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “repeatrepeatrepeat”：

```
SELECT REPEAT( 'repeat', 3 ) FROM iq_dummy
```

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。

另请参见

- REPLACE 函数 [字符串] (第 261 页)
- REPLICATE 函数 [字符串] (第 263 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	要搜索的字符串。此字符串可以为任意长度。
search-string	要搜索并被 <i>replace-string</i> 替换的字符串。该字符串限制为 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	替换字符串，该字符串将替换 <i>search-string</i> 。该字符串可为任意长度。如果 <i>replace-string</i> 是空字符串，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件许可证，或使用 **CAST** 并将 **REPLACE** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “xx.def.xx.ghi”：

```
SELECT REPLACE( 'abc.def.abc.ghi', 'abc', 'xx' ) FROM iq_dummy
```

下面的语句生成包含 **ALTER PROCEDURE** 语句的结果集，这些语句在执行时会对引用已重命名的表的存储过程进行修复。（表名必须唯一才能使用。）

```
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
```

用法

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件许可证，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：

- 声明一个本地临时表，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，请使用 **CAST** 函数。例如，

```
CREATE TABLE aa(a CHAR(5)); INSERT INTO aa VALUES( 'CCCCC' ); COMMIT;
SELECT a, CAST(REPLACE(a, 'C', 'ZZ') AS CHAR(5)) FROM aa;
```

标准和兼容性

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

另请参见

- **CAST** 函数 [数据类型转换] (第 124 页)
- **CONVERT** 函数 [数据类型转换] (第 134 页)
- **HOURS** 函数 [日期和时间] (第 188 页)
- **MINUTES** 函数 [日期和时间] (第 218 页)
- **MONTHS** 函数 [日期和时间] (第 221 页)
- **SECOND** 函数 [日期和时间] (第 270 页)
- **WEEKS** 函数 [日期和时间] (第 317 页)
- **YEAR** 函数 [日期和时间] (第 321 页)
- **YEARS** 函数 [日期和时间] (第 322 页)
- **LCASE** 函数 [字符串] (第 203 页)
- **LEFT** 函数 [字符串] (第 205 页)
- **LOWER** 函数 [字符串] (第 213 页)
- **REVERSE** 函数 [字符串] (第 264 页)
- **RIGHT** 函数 [字符串] (第 265 页)
- **UCASE** 函数 [字符串] (第 308 页)

- UPPER 函数 [字符串] (第 309 页)
- REPEAT 函数 [字符串] (第 260 页)
- REPLICATE 函数 [字符串] (第 263 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件许可证，或使用 **CAST** 并将 **REPLICATE** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “repeatrepeatrepeat”：

```
SELECT REPLICATE( 'repeat', 3 ) FROM iq_dummy
```

用法

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件许可证，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。

另请参见

- REPEAT 函数 [字符串] (第 260 页)
- REPLACE 函数 [字符串] (第 261 页)

REVERSE 函数 [字符串]

采用一个参数作为 **BINARY** 或 **STRING** 类型的输入，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、变量或常量表达式。

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件许可证，或使用 **CAST** 并将 **REVERSE** 设置为正确的数据类型和大小。

示例 1

```
select reverse("abcd")
----
dcba
```

示例 2

```
select reverse(0x12345000)
-----
0x00503412
```

用法

- **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。

权限

任何用户都可以执行 **REVERSE**。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。

另请参见

- 字符串运算符（第 25 页）
- LCASE 函数 [字符串]（第 203 页）
- LEFT 函数 [字符串]（第 205 页）
- LOWER 函数 [字符串]（第 213 页）
- REPLACE 函数 [字符串]（第 261 页）
- RIGHT 函数 [字符串]（第 265 页）
- UCASE 函数 [字符串]（第 308 页）
- UPPER 函数 [字符串]（第 309 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件许可证，或使用 **CAST** 并将 **RIGHT** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “olate”：

```
SELECT RIGHT( 'chocolate', 5 ) FROM iq_dummy
```

用法

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。

另请参见

- LCASE 函数 [字符串] (第 203 页)
- LEFT 函数 [字符串] (第 205 页)
- LOWER 函数 [字符串] (第 213 页)
- REPLACE 函数 [字符串] (第 261 页)
- REVERSE 函数 [字符串] (第 264 页)
- UCASE 函数 [字符串] (第 308 页)
- UPPER 函数 [字符串] (第 309 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
numeric-expression	传递给函数的待舍入数字。
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

示例

下面的语句返回值 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)
123.4570	round (a.n,3)
123.4600	round (a.n,2)
123.5000	round (a.n,1)

值	ROUND (值)
123.0000	round (a.n, 0)
120.0000	round (a.n, -1)
100.0000	round (a.n, -2)
0.0000	round (a.n, -3)

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

- TRUNCNUM 函数 [数值]（第 298 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。
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。

用法

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。

示例

下面的示例返回雇员表中的工资数据，按部门 ID 对结果集进行分区，并根据雇员开始日期对数据进行排序。**ROW_NUMBER** 函数为每一行分配一个行号，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

标准和兼容性

- SQL - 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T611。

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

示例

下面的语句返回行 ID 值 1 到 10:

```
SELECT ROWID( “PRODUCTS” ) FROM PRODUCTS
```

rowid(Products)
1
2

rowid(Products)
3
.
.
.
10

下面的语句返回所有产品 ID 值小于 400 的行的产品 ID 和行 ID 值：

```
SELECT PRODUCTS.ID, ROWID ( PRODUCTS )
FROM PRODUCTS
WHERE PRODUCTS.ID < 400
```

ID	rowid(Products)
300	1
301	2
302	3

下面的语句删除所有行 ID 值大于 50 的行：

```
DELETE FROM PRODUCTS
WHERE ROWID ( PRODUCTS ) > 50
```

用法

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。

ROWID 函数的限制在于它无法使用该表的连接索引，因而得不到通常使用该连接索引所带来的性能优势。

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。

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件许可证，或使用 **CAST** 并将 **RTRIM** 设置为正确的数据类型和大小。

示例

下面的语句返回所有尾随空白均已删除的字符串 “Test Message”。

```
SELECT RTRIM( 'Test Message      ' ) 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。

另请参见

- LTRIM 函数 [字符串]（第 214 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
datetime-expression	日期/时间值。

返回

SMALLINT

示例

下面的语句返回值 5:

```
SELECT SECOND( '1998-07-13 08:21:05' ) 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。

另请参见

- CAST 函数 [数据类型转换] (第 124 页)
- CONVERT 函数 [数据类型转换] (第 134 页)
- HOURS 函数 [日期和时间] (第 188 页)
- MINUTES 函数 [日期和时间] (第 218 页)
- MONTHS 函数 [日期和时间] (第 221 页)
- REPLACE 函数 [字符串] (第 261 页)
- WEEKS 函数 [日期和时间] (第 317 页)
- YEAR 函数 [日期和时间] (第 321 页)
- YEARS 函数 [日期和时间] (第 322 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
<i>datetime-expression</i>	日期和时间。
<i>integer-expression</i>	要添加到 <i>datetime-expression</i> 中的秒数。如果 <i>integer-expression</i> 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 <i>datetime-expression</i> 显式转换为 <i>datetime</i> 数据类型。

返回

INTEGER

TIMESTAMP

示例

下面的语句返回值 3600:

```
SELECT ( SECONDS( '1998-07-13 06:07:12' ) -  
SECONDS( '1998-07-13 05:07:12' ) ) FROM iq_dummy
```

下面的语句返回值 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
```

下面的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
```

用法

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。

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

示例

下面的语句返回值 -1:

```
SELECT SIGN( -550 ) FROM iq_dummy
```

用法

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。

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
<i>string-expression1</i>	要比较的第一个字符串。
<i>string-expression2</i>	要比较的第二个字符串。

返回

SMALLINT

示例

下面的语句返回值 75:

```
SELECT SIMILAR( 'toast', 'coast' ) FROM iq_dummy
```

这表示两个值有 75% 相似。

用法

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。

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

示例

下面的语句返回值 0.496880:

```
SELECT SIN( 0.52 ) 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。

另请参见

- COS 函数 [数值]（第 139 页）
- ATAN2 函数 [数值]（第 118 页）
- ATAN 函数 [数值]（第 118 页）
- ASIN 函数 [数值]（第 117 页）
- ACOS 函数 [数值]（第 114 页）
- COT 函数 [数值]（第 140 页）
- TAN 函数 [数值]（第 296 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	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 STRING。 如果 string-expression 为空值，则 SORTKEY 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。 SORTKEY 函数可处理的输入字符串没有长度限制。SORTKEY 的结果始终限制为 1024 字节和 VARBINARY 数据类型。如果实际结果超过 1024 字节，则只包含前 1024 字节。
collation-name	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 char_collation 或 db_collation（两者等效）以生成与数据库正在使用的 CHAR 归类所用排序键相同的排序键。 同样，可以指定别名 NCHAR_COLLATION 以生成与数据库正在使用的 NCHAR 归类所用排序键相同的排序键。但是，Sybase IQ 不支持将 NCHAR_COLLATION 用于 Sybase IQ 特有的对象。支持对 Sybase IQ 服务器上的 SQL Anywhere 对象使用 NCHAR_COLLATION。
collation-id	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 Adaptive Server Enterprise 归类，这些归类可按照对应的归类 ID 进行引用。
collation-tailoring-string	(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， 'UCA(locale=es;case=LowerFirst;accent=respect)' 指定这些选项的语法与 CREATE DATABASE 语句的 COLLATION 子句相同。. 请参见《参考：语句和选项》>“SQL 语句”>“CREATE DATABASE 语句”。 注意： 在指定 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，则仅支持区分大小写定制。

返回

BINARY

示例

下面的语句查询 **Employees** 表并返回所有雇员的 **FirstName** 和 **Surname**（按使用 dict 归类 [拉丁语-1、英语、法语、德语字典] 的 **Surname** 列的排序键值进行排序）：

```
SELECT Surname, GivenName FROM Employees ORDER BY SORTKEY( Surname, 'dict' );
```

用法

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 函数可保证它针对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。

有效的归类如下：

- 若要查看按标签列出的 Sybase IQ 支持的归类，请执行 **iqinit -l**。
- 下表列出了 Adaptive Server Enterprise 归类。

说明	归类名称	归类 ID
缺省 Unicode 多语言	default	0
CP 850 方案：不区分变音	altnoacc	39
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	二进制	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

说明	归类名称	归类 ID
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	cymnocs	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	hunnocs	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
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与 **SORTKEY** 函数的缺省定制设置不同。若要从 **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 Sybase IQ 创建的排序键值不包含使用 15.0 和更高版本创建的相同值。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 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。

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

示例

下面的语句返回两个数字，分别表示每个名称的声音。每个参数的 **SOUNDEX** 值都是 3827。

```
SELECT SOUNDEX( 'Smith' ), SOUNDEX( 'Smythe' ) FROM iq_dummy
```

SOUNDEX ('Smith') 等同于 **SOUNDEX** ('Smythe')。

用法

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 Enterprise 兼容，但 Adaptive Server Enterprise 返回 CHAR(4) 结果，而 Sybase IQ 返回整数。

另请参见

- DIFFERENCE 函数 [字符串] (第 169 页)

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
integer-expression	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件许可证，或使用 **CAST** 并将 **SPACE** 设置为正确的数据类型和大小。

示例

下面的语句返回包含 10 个空格的字符串：

```
SELECT SPACE( 10 ) 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。

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-stand-ard-string	测试遵从性所使用的标准级别。可能的值与 SQL_FLAGGER_ERROR_LEVEL 数据库选项的值相同： • SQL:2003/Core 测试核心 SQL/2003 语法的一致性。 • SQL:2003/Package 测试完整 SQL/2003 语法的一致性。 • SQL:1999/Core 测试核心 SQL/1999 语法的一致性。 • SQL:1999/Package 测试完整 SQL/1999 语法的一致性。 • SQL:1992/Entry 测试初级 SQL/1992 语法的一致性。 • SQL:1992/Intermediate 测试中级 SQL/1992 语法的一致性。 • SQL:1992/Full 测试完整 SQL/1992 语法的一致性。
sql-state-ment-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件许可证，或使用 **CAST** 并将 **SQLFLAGGER** 设置为正确的数据类型和大小。

示例

下面的语句显示在找到禁用扩展后返回的消息示例：

```
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'。

下面的语句返回 “00000”，因为它不包含禁用扩展：

```
SELECT SQLFLAGGER( 'SQL:2003/Package', 'SELECT dummy_col FROM
sys.dummy' );
```

用法

您也可以使用 `iqsqlpp` SQL 预处理器实用程序标记不是 SQL92 指定集合组成部分的任意嵌入式 SQL。请参见《实用程序指南》>“`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。

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
<code>numeric-expression</code>	要计算其平方根的数字。

返回

DOUBLE

示例

下面的语句返回值 3：

```
SELECT SQRT( 9 ) 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。

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， SQUARE 函数会产生错误。返回值为 DOUBLE 数据类型。

用法

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

示例

假定以下数据：

```
SELECT Salary FROM Employees WHERE DepartmentID = 300
```

Salary
51432.000
57090.000
42300.000
43700.00
36500.000

Salary
138948.000
31200.000
58930.00
75400.00

下面的语句返回值 32617.8446712838471:

```
SELECT STDDEV ( Salary ) FROM Employees
WHERE DepartmentID = 300
```

假定以下数据:

```
SELECT UnitPrice FROM Products WHERE Name = 'Tee Shirt'
```

名称	UnitPrice
Tee Shirt	9.00
Tee Shirt	14.00
Tee Shirt	14.00

下面的语句返回值 2.88675134594813049:

```
SELECT STDDEV ( UnitPrice ) FROM Products
WHERE Name = 'Tee Shirt'
```

用法

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。

另请参见

- 窗口化集合函数的用法 (第 95 页)
- **STDDEV_SAMP** 函数 [集合] (第 285 页)

- VARIANCE 函数 [集合] （第 315 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

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	平均值	方差
2000	1	25.775148	14.2794
2000	2	27.050847	15.0270
.....			

用法

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。

另请参见

- 窗口化集合函数的用法 （第 95 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

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	平均值	方差
2000	1	25.775148	14.3218
2000	2	27.050847	15.0696
.....			

用法

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。

另请参见

- 窗口化集合函数的用法（第 95 页）
- STDDEV 函数 [集合]（第 282 页）
- VARIANCE 函数 [集合]（第 315 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

示例

下面的语句返回的字符串共有 10 个字符，其中 1234 前面有 6 个空格：

```
SELECT STR( 1234.56 ) FROM iq_dummy
```

下面的语句返回结果 1234.5：

```
SELECT STR( 1234.56, 6, 1 ) FROM iq_dummy
```

用法

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。

STR_REPLACE 函数 [字符串]

采用三个参数作为 **BINARY** 或 **STRING** 类型的输入，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

表 94. 参数

参数	描述
string_expr1	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。
string_expr2	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。
string_expr3	替换字符串表达式，表示为 CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY 或 BINARY 数据类型。

示例 1

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)
```

示例 2

将所有空格替换为 “toyota”

```
select str_replace ("chevy, ford, mercedes", "", "toyota")
-----
chevy,toyotaford,toyotamercedes
(1 row(s) affected)
```

示例3

接受第三个参数中的空值并将其视为尝试用空值替换 *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)
```

用法

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，在进行进一步的计算之前，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。如果所有参数都是分配给常量的列或宿主变量，则 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 Sybase IQ 因在编译表达式时参数值未知而无法计算结果长度，则使用的结果长度为 255。
- **RESULT_LEN** 永远不会超过 32767。

权限

任何用户都可以执行 **STR_REPLACE**。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。

另请参见

- **BIT_LENGTH** 函数 [字符串] (第 121 页)
- **BYTE_LENGTH** 函数 [字符串] (第 122 页)
- **CHAR_LENGTH** 函数 [字符串] (第 128 页)
- **COL_LENGTH** 函数 [系统] (第 131 页)
- **DATALength** 函数 [系统] (第 145 页)
- **LEN** 函数 [字符串] (第 206 页)
- **LENGTH** 函数 [字符串] (第 207 页)
- **OBJECT_NAME** 函数 [系统] (第 232 页)

- OCTET_LENGTH 函数 [字符串] （第 233 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件许可证，或使用 **CAST** 并将 **STRING** 设置为正确的数据类型和大小。

示例

下面的语句返回值 testing123:

```
SELECT STRING( 'testing', NULL, 123 )  
FROM iq_dummy
```

用法

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。

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 <code>xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</code> 的字符串

返回

UNIQUEIDENTIFIER

示例

```
CREATE TABLE T (  
  pk uniqueidentifier primary key,  
  c1 int);  
INSERT INTO T (pk, c1)  
VALUES (STRTOUUID  
  ('12345678-1234-5678-9012-123456789012'), 1);
```

用法

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 Adaptive Server EnterpriseSybase IQ 数据库。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase — Adaptive Server Enterprise 不支持。

另请参见

- 二进制数据类型（第 72 页）
- NEWID 函数 [杂项]（第 223 页）
- UUIDTOSTR 函数 [字符串]（第 311 页）
- 字符数据类型（第 65 页）
- 二进制数据类型（第 618 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
string-expression1	要用 STUFF 函数修改的字符串。
start	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。
length	要删除的字符数。
string-expression2	要插入的字符串。若要使用 STUFF 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

示例

下面的语句返回值 “chocolate pie”：

```
SELECT STUFF( 'chocolate cake', 11, 4, 'pie' )  
FROM iq_dummy
```

用法

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。

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。
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件许可证，或使用 **CAST** 并将 **STRING** 设置为正确的数据类型和大小。

示例

下面的语句返回 “back”：

```
SELECT SUBSTRING ( 'back yard', 1 , 4 )  
FROM iq_dummy
```

下面的语句返回 yard：

```
SELECT SUBSTR ( 'back yard', -1 , -4 )  
FROM iq_dummy
```

下面的语句返回 0x2233：

```
SELECT SUBSTR ( 0x112233445566, 2, 2 )  
FROM iq_dummy
```

用法

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。

请参见《Sybase IQ 中的非结构化数据分析》> “函数支持”。

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase — Adaptive Server Enterprise 不支持 **SUBSTR**。请改用 **SUBSTRING**。

另请参见

- CHARINDEX 函数 [字符串] (第 129 页)

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。

用法

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《Sybase IQ 中的非结构化数据分析》> “函数支持”。

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。
DISTINCT <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

示例

下面的语句返回值 3749146.740:

```
SELECT SUM( salary )  
FROM Employees
```

用法

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。

另请参见

- AVG 函数 [集合] (第 119 页)
- COUNT 函数 [集合] (第 143 页)
- 窗口化集合函数的用法 (第 95 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

示例

下面的语句返回用户标识号 1:

```
SELECT SUSER_ID ( 'DBA' ) FROM iq_dummy
```

下面的语句返回用户标识号 0:

```
SELECT SUSER_ID ( 'SYS' ) FROM iq_dummy
```

标准和兼容性

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

另请参见

- SUSER_NAME 函数 [系统] (第 295 页)
- USER_ID 函数 [系统] (第 310 页)
- USER_NAME 函数 [系统] (第 310 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件许可证，或使用 **CAST** 并将 **SUSER_NAME** 设置为正确的数据类型和大小。

示例

下面的语句返回值 DBA:

```
SELECT SUSER_NAME ( 1 ) FROM iq_dummy
```

下面的语句返回值 SYS:

```
SELECT SUSER_NAME ( 0 ) FROM iq_dummy
```

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

- Sybase - 为 Sybase IQ 实现的 Sybase IQAdaptive Server Enterprise 函数。在 Adaptive Server Enterprise 中，**SUSER_NAME** 返回服务器用户名。

另请参见

- SUSER_ID 函数 [系统] (第 294 页)
- USER_ID 函数 [系统] (第 310 页)
- USER_NAME 函数 [系统] (第 310 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

示例

它返回值 .572561:

```
SELECT TAN( 0.52 ) 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。

另请参见

- COS 函数 [数值] (第 139 页)
- ATAN2 函数 [数值] (第 118 页)
- ATAN 函数 [数值] (第 118 页)
- ASIN 函数 [数值] (第 117 页)
- ACOS 函数 [数值] (第 114 页)
- COT 函数 [数值] (第 140 页)
- SIN 函数 [数值] (第 274 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

示例

下列语句根据系统时钟返回当前日期。

```
SELECT TODAY( * ) FROM iq_dummy
```

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。

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件许可证，或使用 **CAST** 并将 **TRIM** 设置为正确的数据类型和大小。

示例

下面的语句返回没有前导或尾随空白的值 “chocolate pie”。

```
SELECT TRIM( '   chocolate   ' ) FROM iq_dummy
```

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**。

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

示例

下面的语句返回值 600:

```
SELECT TRUNCNUM( 655, -2 ) FROM iq_dummy
```

下面的语句返回值 655.340:

```
SELECT TRUNCNUM( 655.348, 2 ) FROM iq_dummy
```

用法

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。

另请参见

- **ROUND** 函数 [数值] (第 266 页)

TS ARMA_AR 函数 [时序]

计算用于自回归移动平均值 (ARMA) 模型的参数的最小二乘估计值，并返回请求的自回归估计值。

语法

```
TS_ARMA_AR (timeseries_expression, ar_count, ar_elem, method)  
OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS ARMA_CONST 函数 [时序]

计算用于自回归移动平均值 (ARMA) 模型的参数的最小二乘估计值，并返回估计的常量。

语法

```
TS_ARMA_CONST (timeseries_expression, method)  
OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS ARMA_MA 函数 [时序]

计算用于自回归移动平均值 (ARMA) 模型的参数的最小二乘估计值，并返回请求的移动平均估计值。

语法

```
TS_ARMA_MA (timeseries_expression, ma_count, ma_elem, method)  
OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS AUTOCORRELATION 函数 [时序]

计算固定时序的示例自相关函数。

语法

```
TS_AUTOCORRELATION (timeseries_expression, lagmax, lag_elem)  
OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA 函数 [时序]

确定乘积季节性自回归整合移动平均值 (ARIMA) 模型的参数，并生成合并了其效应持续至时序结束之后的离群值的预测。

语法

```
TS_AUTO_ARIMA(< time_value >,< timeseries_expression > [< max_lag >
[< critical > [< epsilon > [< criterion > [< confidence > [<
model > [< n_predictions >]]]]]]])
OVER ( window-spec )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_OUTLIER 函数 [时序]

与 **TS_AUTO_ARIMA** 集合函数一样，**TS_AUTO_ARIMA_OUTLIER** 也接受输入时序，并自动确定乘积季节性自回归整合移动平均值 (ARIMA) 模型的参数。

但是，**TS_AUTO_ARIMA** 使用 ARIMA 模型来预测超出输入集的值，而 **TS_AUTO_ARIMA_OUTLIER** 使用 ARIMA 模型来识别输入时序的统计离群值元素，并返回每个元素的离群值类型。

语法

```
TS_AUTO_ARIMA_OUTLIER(< time_value >,< timeseries_expression > [<
```

```
max_lag > [< critical > [< epsilon > [< criterion > [<
confidence > [< model > [< delta >]]]]]]])
OVER ( window-spec )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_RESULT_AIC 函数 [时序]

检索 **TS_AUTO_ARIMA** 生成的 Akaike 的信息准则 (AIC) 输出参数。

语法

```
TS_AUTO_ARIMA_RESULT_AIC( auto_arima_result )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_RESULT_AICC 函数 [时序]

检索 TS_AUTO_ARIMA 生成的修正的 AIC (AICC) 输出参数。

语法

```
TS_AUTO_ARIMA_RESULT_AICC( auto_arima_result )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_RESULT_BIC 函数 [时序]

检索 TS_AUTO_ARIMA 生成的 Bayesian 信息准则 (BIC) 输出参数。

语法

```
TS_AUTO_ARIMA_RESULT_BIC( auto_arima_result )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_RESULT_FORECAST_VALUE 函数 [时序]

检索 TS_AUTO_ARIMA 生成的没有离群值的请求序列的预测值。

语法

```
TS_AUTO_ARIMA_RESULT_FORECAST_VALUE( auto_arima_result ,  
model_element_number )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_RESULT_FORECAST_ERROR 函数 [时序]

检索 TS_AUTO_ARIMA 生成的原始输入序列的预测标准差值。

语法

```
TS_AUTO_ARIMA_RESULT_FORECAST_ERROR( auto_arima_result ,  
forecast_element_number )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_RESULT_MODEL_D 函数 [时序]

检索 TS_AUTO_ARIMA 在计算 ARIMA 模型说明时生成的 d 值。

语法

```
TS_AUTO_ARIMA_RESULT_MODEL_D( auto_arima_result )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_RESULT_MODEL_P 函数 [时序]

检索 TS_AUTO_ARIMA 在计算 ARIMA 模型说明时生成的 p 值。

语法

```
TS_AUTO_ARIMA_RESULT_MODEL_P( auto_arima_result )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_RESULT_MODEL_Q 函数 [时序]

检索 TS_AUTO_ARIMA 在计算 ARIMA 模型说明时生成的 q 值。

语法

```
TS_AUTO_ARIMA_RESULT_MODEL_Q( auto_arima_result )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_RESULT_MODEL_S 函数 [时序]

检索 TS_AUTO_ARIMA 在计算 ARIMA 模型说明时生成的 s 值。

语法

```
TS_AUTO_ARIMA_RESULT_MODEL_S( auto_arima_result )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA 函数 [时序]

检索没有离群值的数据点的残差标准差。

语法

```
TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA( auto_arima_result )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_AUTO_UNI_AR 函数 [时序]

执行单变量自回归时序模型的自动选择和拟合。

语法

```
TS_AUTO_UNI_AR (timeseries_expression, ar_count, ar_elem, method)  
OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_BOX_COX_XFORM 函数 [时序]

执行正向和反向 Box-Cox 幂转换。

语法

```
TS_BOX_COX_XFORM (timeseries_expression, power [, shift [,  
inverse] ]) OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_DIFFERENCE 函数 [时序]

区分季节性或非季节性时序。

语法

```
TS_DIFFERENCE (timeseries_expression, period1 [, period2 [, ...period  
10] ]) OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_DOUBLE_ARRAY 函数 [时序]

TS_GARCH 函数的支持函数。构造包含 3 到 10 个常量双精度浮点值的逻辑数组，并返回单个 varbinary 值。

语法

```
TS_DOUBLE_ARRAY(xguess1, xguess2, xguess3, [ ... [ , xguess10] ... ] )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_ESTIMATE_MISSING 函数 [时序]

估计时序中的缺失值并将它们作为散布在原始时序中的新时序返回。

语法

```
TS_ESTIMATE_MISSING ( timeseries_expression, method )
```

```
OVER ( window-spec )
```

注意： 此函数仅可用于 RAP – Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_GARCH 函数 [时序]

计算 GARCH(p,q) 模型的参数估计值。

语法

```
TS_GARCH ( <time series expression> , <garch_count> , <arch_count> ,  
<xguess_binary_encoding> [, <max_sigma> ] )  
OVER ( window-spec )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_GARCH_RESULT_A 函数 [时序]

TS_GARCH 函数的支持函数。检索 **TS_GARCH** 集合函数生成的对数似然输出参数 *A*。

语法

```
TS_GARCH_RESULT_A ( ts_garch_result )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_GARCH_RESULT_AIC 函数 [时序]

TS_GARCH 函数的支持函数。检索 **TS_GARCH** 集合函数生成的 Akaike 信息准则输出参数 *AIC*。

语法

```
TS_GARCH_RESULT_AIC ( ts_garch_result )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_GARCH_RESULT_USER [时序]

TS_GARCH 函数的支持函数。访问描述 **GARCH(p, q)** 模型的逻辑数组中的每个元素。

语法

```
TS_GARCH_RESULT_USER( ts_garch_result , model_element_number )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_INT_ARRAY [时序]

TS_AUTO_ARIMA 函数和 **TS_AUTO_ARIMA_OUTLIER** 函数的支持函数。构造编码为 varbinary 值的常量整数值的逻辑数组。

语法

```
TS_INT_ARRAY( int1 , int2 , int3 , int4 , [...], int10 [...] )
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_LACK_OF_FIT 函数 [时序]

在给定的适当的相关函数的情况下针对单变量时序或传递函数执行失拟检验。

语法

```
TS_LACK_OF_FIT(timeseries_expression, p_value, q_value, lagmax,  
[tolerance])
```

```
OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_LACK_OF_FIT_P 函数 [时序]

对单变量时序执行失拟检验。此函数与 **TS_LACK_OF_FIT** 函数相同，只不过它返回的是 q 的 p 值，而不是返回 q。

语法

```
TS_LACK_OF_FIT_P(timeseries_expression, p_value, q_value, lagmax,  
[tolerance])
```

```
OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_MAX_ARMA_AR 函数 [时序]

计算单变量 ARMA（自回归移动平均值）时序模型中参数的精确最大似然估计值，并返回请求的自回归估计值。

语法

```
TS_MAX_ARMA_AR (timeseries_expression, ar_count, ar_elem)
```

```
OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_MAX_ARMA_CONST 函数 [时序]

计算单变量 ARMA（自回归移动平均值）时序模型中参数的精确最大似然估计值，并返回常量估计值。

语法

```
TS_MAX_ARMA_CONST (timeseries_expression)
```

```
OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_MAX_ARMA_LIKELIHOOD 函数 [时序]

计算单变量 ARMA（自回归移动平均值）时序模型中参数的精确最大似然估计值，并返回拟合模型的似然值 (ln)。

语法

```
TS_MAX_ARMA_LIKELIHOOD (timeseries_expression)
```

```
OVER (window-spec)
```

注意： 此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_MAX_ARMA_MA 函数 [时序]

计算单变量 ARMA（自回归移动平均值）时序模型中参数的精确最大似然估计值，并返回请求的移动平均估计值。

语法

```
TS_MAX_ARMA_MA (timeseries_expression, ma_count, ma_elem)
```

```
OVER (window-spec)
```

注意：此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_OUTLIER_IDENTIFICATION 函数 [时序]

检测并确定离群值，同时估计某个时序中的模型参数的值，其中，基础无离群值序列遵循常规季节性或非季节性 ARMA 模型。

语法

```
TS_OUTLIER_IDENTIFICATION (timeseries_expression, p_value, q_value,  
s_value, d_value, [, delta_value[, critical_value]])
```

```
OVER (window-spec)
```

注意：此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_PARTIAL_AUTOCORRELATION 函数 [时序]

计算固定时序的示例偏自相关函数。

语法

```
TS_PARTIAL_AUTOCORRELATION (timeseries_expression, lagmax, lag_elem)
```

```
OVER (window-spec)
```

注意：此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

TS_VWAP 函数 [时序]

VWAP 表示成交量加权平均价。**TS_VWAP** 计算特定时间范围内交易价值与交易总量之间的比率。

VWAP 是已定义交易范围的平均股价的测量值。可以使用 **TS_VWAP** 作为简单集合函数和 OLAP 样式集合函数。

与其它时序函数不同，**TS_VWAP** 不调用 IMSL 库。

语法 1

```
TS_VWAP (price_expression, volume_expression)
```

语法 2

```
TS_VWAP (price_expression, volume_expression)
```

```
OVER (window-spec)
```

注意：此函数仅可用于 RAP - Trading Edition Enterprise。有关此函数的详细信息，请参见《时序指南》。

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件许可证，或使用 **CAST** 并将 **UCASE** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “CHOCOLATE”：

```
SELECT UCASE( 'ChocoLate' ) FROM iq_dummy
```

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - Adaptive Server Enterprise 不支持 **UCASE**，但 **UPPER** 可通过兼容方式提供相同的功能。

另请参见

- LCASE 函数 [字符串]（第 203 页）
- LEFT 函数 [字符串]（第 205 页）
- LOWER 函数 [字符串]（第 213 页）
- REPLACE 函数 [字符串]（第 261 页）
- REVERSE 函数 [字符串]（第 264 页）
- RIGHT 函数 [字符串]（第 265 页）
- UPPER 函数 [字符串]（第 309 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件许可证，或使用 **CAST** 并将 **UPPER** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “CHOCOLATE”：

```
SELECT UPPER( 'ChocoLate' ) 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。

另请参见

- LCASE 函数 [字符串]（第 203 页）
- LEFT 函数 [字符串]（第 205 页）
- LOWER 函数 [字符串]（第 213 页）
- REPLACE 函数 [字符串]（第 261 页）
- REVERSE 函数 [字符串]（第 264 页）
- RIGHT 函数 [字符串]（第 265 页）
- UCASE 函数 [字符串]（第 308 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

示例

下面的语句返回用户标识号 1:

```
SELECT USER_ID ( 'DBA' ) FROM iq_dummy
```

下面的语句返回用户标识号 0:

```
SELECT USER_ID ( 'SYS' ) FROM iq_dummy
```

标准和兼容性

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

另请参见

- SUSER_ID 函数 [系统] (第 294 页)
- SUSER_NAME 函数 [系统] (第 295 页)
- USER_NAME 函数 [系统] (第 310 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件许可证，或使用 **CAST** 并将 **USER_NAME** 设置为正确的数据类型和大小。

示例

下面的语句返回值 “DBA”：

```
SELECT USER_NAME ( 1 ) FROM iq_dummy
```

下面的语句返回值 “SYS”：

```
SELECT USER_NAME ( 0 ) FROM iq_dummy
```

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 为 Sybase IQ 实现的 Adaptive Server Enterprise 函数。在 Adaptive Server Enterprise 中，USER_NAME 返回用户名，而不是服务器用户名。

另请参见

- SUSER_ID 函数 [系统]（第 294 页）
- SUSER_NAME 函数 [系统]（第 295 页）
- USER_ID 函数 [系统]（第 310 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 95. 参数

参数	描述
uuid-expression	唯一标识符值。

返回

VARCHAR

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);  
INSERT INTO T3 (pk, c1)  
values (0x12345678123456789012123456789012, 1)  
SELECT UUIDTOSTR(pk) FROM T3
```

用法

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。

另请参见

- 二进制数据类型（第 72 页）
- NEWID 函数 [杂项]（第 223 页）
- STRTOUUID 函数 [字符串]（第 290 页）
- 字符数据类型（第 65 页）
- 二进制数据类型（第 618 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

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	平均值	方差
2000	1	25.775148	203.9021
2000	2	27.050847	225.8109
.....			

用法

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。

另请参见

- 窗口化集合函数的用法（第 95 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

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	平均值	方差
2000	1	25.775148	205.1158
2000	2	27.050847	227.0939
.....			

用法

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 Sybase IQ 12.7 和更高版本中包含一个元素的输入集，NULL 将返回 NULL。在 12.7 以前的版本中，NULL 返回零。

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 — Adaptive Server Enterprise 不支持。

另请参见

- 窗口化集合函数的用法（第 95 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

示例

假定以下数据：

```
SELECT Salary FROM Employees WHERE DepartmentID = 300
```

salary
51432.000
57090.000
42300.000
43700.00
36500.000
138948.000
31200.000
58930.00

salary
75400.00

下面的语句返回值 1063923790.99999994:

```
SELECT VARIANCE ( Salary ) FROM Employees
WHERE DepartmentID = 300
```

假定以下数据:

```
SELECT UnitPrice FROM Products WHERE name = 'Tee Shirt'
```

UnitPrice
9.00
14.00
14.00

下面的语句返回值 8.33333333333334327:

```
SELECT VARIANCE ( UnitPrice ) FROM Products
WHERE name = 'Tee Shirt'
```

用法

用于计算 **VARIANCE** 的公式是

$$var = \frac{n \sum x^2 - (\sum x)^2}{n(n-1)}$$

VARIANCE 返回双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。

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。

另请参见

- 窗口化集合函数的用法 (第 95 页)
- STDDEV 函数 [集合] (第 282 页)
- STDDEV_SAMP 函数 [集合] (第 285 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
datetime-expression	日期和时间。
integer-expression	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。会忽略小时、分钟和秒钟。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。

示例

下面的语句返回值 104278:

```
SELECT WEEKS( '1998-07-13 06:07:12' ) FROM iq_dummy
```

下面的语句返回值 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
```

下面的语句返回时间戳值 1999-06-16 21:05:07.000:

```
SELECT WEEKS( CAST( '1999-05-12 21:05:07'
AS TIMESTAMP ), 5) FROM iq_dummy
```

用法

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。会忽略小时、分钟和秒钟。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。

另请参见

- CAST 函数 [数据类型转换] (第 124 页)
- CONVERT 函数 [数据类型转换] (第 134 页)
- HOURS 函数 [日期和时间] (第 188 页)
- MINUTES 函数 [日期和时间] (第 218 页)
- MONTHS 函数 [日期和时间] (第 221 页)
- REPLACE 函数 [字符串] (第 261 页)
- SECOND 函数 [日期和时间] (第 270 页)
- YEAR 函数 [日期和时间] (第 321 页)
- YEARS 函数 [日期和时间] (第 322 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 (<i>expression</i>)
OVER (<i>window-spec</i>)

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。

用法

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

$$WMA_M = \frac{np_M + (n - 1)p_{M-1} + \cdots + 2p_{M-n+2} + p_{M-n+1}}{n + (n - 1) + \cdots + 2 + 1}$$

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。

示例

下面的示例按部门返回位于佛罗里达州的员工的工资加权平均值，其中最近聘用的员工的工资占该平均值的权重最大：

```
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 96. **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

标准和兼容性

- SQL - ISO/ANSI SQL 语法的供应商扩展。

另请参见

- **EXP_WEIGHTED_AVG** 函数 [集合]（第 175 页）

- 窗口化集合函数的用法（第 95 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
expression	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。
min_value	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。
max_value	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。
num_buckets	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。

示例

以下示例基于 **credit_limit** 列为样本表中马萨诸塞州的客户创建十表元直方图，并为每名客户返回表元号（“Credit Group”）。其信用额度超过最大值的客户将被分配到溢出表元 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，表元将成为半开半闭区间。例如：**WIDTH_BUCKET** (*credit_limit*, 5000, 0, 5)。在此示例中，表元号 1 的上下限为 (4000, 5000]，表元号 2 的上下限为 (3000, 4000]，表元号 5 的上下限为 (0, 1000]。上溢表元的编号为 0 (5000, +infinity)，下溢表元的编号为 6 (-infinity, 0]。

用法

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。

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
<i>datetime-expression</i>	日期和时间。

返回

SMALLINT

示例

下面的语句返回值 1998：

```
SELECT YEAR( '1998-07-13 06:07:12' ) FROM iq_dummy
```

用法

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。

另请参见

- CAST 函数 [数据类型转换] (第 124 页)
- CONVERT 函数 [数据类型转换] (第 134 页)
- HOURS 函数 [日期和时间] (第 188 页)
- MINUTES 函数 [日期和时间] (第 218 页)
- MONTHS 函数 [日期和时间] (第 221 页)
- REPLACE 函数 [字符串] (第 261 页)
- SECOND 函数 [日期和时间] (第 270 页)
- WEEKS 函数 [日期和时间] (第 317 页)
- YEARS 函数 [日期和时间] (第 322 页)
- NTILE 函数 [分析] (第 228 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**。

示例

下面的语句返回值 1998:

```
SELECT YEARS( '1998-07-13 06:07:12' ) FROM iq_dummy
```

下面的语句返回 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
```

下面的语句返回 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
```

用法

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。

另请参见

- CAST 函数 [数据类型转换] (第 124 页)
- CONVERT 函数 [数据类型转换] (第 134 页)
- HOURS 函数 [日期和时间] (第 188 页)
- MINUTES 函数 [日期和时间] (第 218 页)
- MONTHS 函数 [日期和时间] (第 221 页)
- REPLACE 函数 [字符串] (第 261 页)
- SECOND 函数 [日期和时间] (第 270 页)
- WEEKS 函数 [日期和时间] (第 317 页)
- YEAR 函数 [日期和时间] (第 321 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。
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

示例

下面的语句返回值 1998-06-12:

```
SELECT YMD( 1998, 06, 12 ) FROM iq_dummy
```

如果值超出它们的正常范围，则相应地调整日期。例如，下面的语句返回值 1993-03-01:

```
SELECT YMD( 1992, 15, 1 ) FROM iq_dummy
```

下面的语句返回值 1993-02-28:

```
SELECT YMD ( 1992, 15, 1-1 ) FROM iq_dummy
```

下面的语句返回值 1992-02-29:

```
SELECT YMD ( 1992, 3, 1-1 ) FROM iq_dummy
```

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。

与其它 SQL 术语的区别

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 Sybase IQ 功能是许多其它 SQL 实现未能提供的。

日期

Sybase IQ 具有 date、time 和 timestamp 类型，这些类型包括年、月、日、小时、分钟、秒和秒的小数部分。插入或更新日期字段时，或与日期 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 97. 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

Sybase IQ 同时支持实体完整性和参照完整性。

这是通过对 **CREATE TABLE** 和 **ALTER TABLE** 进行的以下两种扩展实现的。

```
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。之后，IQ 会强制实现主键的唯一性，并确保主键中的任何一列都不包含 NULL 值。

FOREIGN KEY 子句定义此表与另一个表之间的关系。该关系由此表中的一列（或多列）表示，列中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完整性 -- 只要修改了这些列或在此表中插入了行，便会对这些列进行检查，以确保有一列或

与其它 SQL 术语的区别

多列为空值，或这些值与另一个表的主键中某行的对应列匹配。有关详细信息，请参见 **CREATE TABLE** 语句。

连接

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，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

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：

```
ADD column data-type
MODIFY column data-type
DELETE column
RENAME new-table-name
RENAME old-column TO new-column
```

使用 **MODIFY** 可以更改字符列的最大长度，也可以从一种数据类型转换为另一种数据类型。

有时不允许使用子查询

与 SQL Anywhere 不同，Sybase IQ 不允许子查询 出现在允许使用表达式的任何位置。

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。Sybase IQ 不支持 SQL Anywhere 扩展。

许多 SQL 实现只允许将子查询放置在 比较运算符的右边。例如，以下命令在 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

Sybase IQ 支持几个在 ANSI SQL 定义中没有的函数。

另请参见

- SQL 函数 (第 91 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

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 98. Sybase IQ 数据库对象 的大小和数量限制

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	由操作系统文件 大小决定。
dbspace 大小	裸设备： 最大大小为 4TB。 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。 Sybase IQ 不推荐在 NAS（网络 附加存储）设备上创建 dbspace。
字段大小	对于 BINARY 为 255 字节，对于 VARBINARY 为 32,767 字节 32,767（对于 CHAR、VARCHAR） 对于 LONG BINARY 和 LONG VARCHAR，上限为 512 TB（适用于 128 KB 的页）或 1 PB（适用于 512 KB 的页）。
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 -gss 增加目录堆栈大小，并使用 -c 或 -ch 加 -cl 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，文本限制在 IQ 目录页大小。要打印长命令，可以使用增加的 -gp 设置启动服务器，不过通常 Sybase 建议使用缺省值 -gp 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 _n 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
每个表的列数	Sybase IQ 最多支持一个表中包含 45,000 列。如果表中的列数多于 10,000，则会发生性能下降。每个表中允许 NULL 的列数限制大约为 8*（数据库页大小 - 30）。
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 2^{32} （大约 4,000,000）个。
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
每个连接索引的表数（可以在一个查询块中连接的表数）	31
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	Sybase 建议限制为 页大小的一半。
表大小	受数据库大小的限制。

另请参见

- 字符串函数（第 105 页）

系统过程

使用 Sybase IQ 数据库中系统提供的存储过程 来检索系统信息。

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，这包括上述两类过程。
请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用”> “系统过程”。
- Transact-SQL 系统和类过程。

《Sybase IQ 中的非结构化数据分析》> “存储过程 支持”中介绍了专门与大对象 数据相关的系统存储过程（包括 `sp_iqsetcompression` 和 `sp_iqshowcompression`）。

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。

允许一些变化形式，因为产品既支持 Sybase IQ SQL 语法又支持 Transact-SQL 语法。如果需要与 Transact-SQL 兼容，请确保使用 Transact-SQL 语法。

表 99. 存储过程语法变化形式

语法	语法类型	解释
<code>procedure_name ('param')</code>	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>	Sybase IQ	如果将参数包含在引号中，则括号是可选项。

语法	语法类型	解释
procedure_name param	Transact-SQL	如果省略 参数两侧的引号，则还必须省略括号。 注意： 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>， iqtablesize 'dba.emp1' sp_ 需要用引号将参数引起来，而 iqtablesize emp1 sp_ 则不需要。
procedure_name	Sybase IQ IQ 或 Transact-SQL	如果在 Interactive SQL 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。
call procedure_name (param=' value')	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 Sybase IQ 在执行相应过程时 的状态的有关信息。

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。

若要监控 Sybase IQ 在更长时段内的使用情况，请使用 Sybase IQ 监控器， 它用于按您指定的间隔收集和报告从监控器启动 到其停止时段内的统计信息。

系统存储过程

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 拥有。

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。

sa_char_terms 系统过程

将 CHAR 字符串拆分为多个术语，并分别在单独的行中返回各个术语及其位置。

请参见《Sybase IQ 中的非结构化数据分析》>“存储过程支持”>“管理 TEXT 索引中的术语”>“sa_char_terms 系统过程”。

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 CHARACTER 参数指定表或视图的名称。
- **owner_name** - 使用此可选的 CHARACTER 参数指定 *tbl_name* 的所有者。

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

使用此过程获取相关视图的 ID 的列表。或者，也可以在语句中使用该过程返回有关视图的详细信息，如视图名称。

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。另外：

- *tbl_name* 为可选项，其缺省值为空。
- 如果 *owner* 和 *tbl_name* 均为空，则将返回具有相关视图的所有表的相关信息。
- 如果 *tbl_name* 为空，但指定了 *owner*，则将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* 为空，则将返回具有指定名称的所有表的相关信息。

缺省情况下，执行该过程不需要任何权限，而且假定 PUBLIC 组可以访问该目录。DBA 可以根据需要控制对视图和/或目录的访问。

权限

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_external_library_unload 过程

卸载外部库。

请参见《Sybase IQ 中的非结构化数据分析》>“存储过程支持”>“外部库标识”>“sa_external_library_unload 系统过程”。

sa_list_external_library 过程

列出服务器上当前装载的外部库。

请参见《Sybase IQ 中的非结构化数据分析》>“存储过程支持”>“标识外部库”>“sa_list_external_library 过程”。

sa_nchar_terms 系统过程

将 NCHAR 字符串拆分为多个术语，并分别在单独的行中返回各个术语及其位置。

请参见《Sybase IQ 中的非结构化数据分析》>“存储过程支持”>“管理 TEXT 索引中的术语”>“sa_nchar_terms 系统过程”。

sa_text_index_vocab 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。

请参见《Sybase IQ 中的非结构化数据分析》>“存储过程支持”>“管理 TEXT 索引中的术语”>“sa_text_index_vocab 系统过程”。

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( )
```

参数

无

结果集

列名	数据类型	说明
user_id	UNSIGNED INT	标识用户的唯一编号。
user_name	CHAR(128)	用户的名称。
连接	INT	该用户当前建立的连接数。
failed_logins	UNSIGNED INT	用户进行的登录失败重试次数。
last_login_time	TIMESTAMP	用户上次登录的时间。
locked	TINYINT	指示用户帐户是否锁定。
reason_locked	LONG VARCHAR	帐户被锁定的原因。

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。

不具备 DBA 权限的用户可以通过创建和执行由 DBA 拥有的包装过程获取用户信息。

权限

需要具有 DBA 权限才能查看所有用户的信息。没有 DBA 权限的用户可以查看其自身的信息。另外，不具备 DBA 权限的用户可以通过执行由 DBA 拥有的包装过程查看关于其他用户的信息。

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;
```

sp_expireallpasswords 过程

导致所有用户口令立即过期。

语法 1

```
call sp_expireallpasswords
```

语法 2

```
sp_expireallpasswords
```

权限

需要具有 DBA 或 USER ADMIN 权限。

示例

导致所有用户口令立即过期：

```
call sp_expireallpasswords
```

另请参见

- sp_iqaddlogin 过程（第 338 页）
- sp_iqcopyloginpolicy 过程（第 366 页）
- sp_iqmodifyadmin 过程（第 423 页）
- sp_iqmodifylogin 过程（第 424 页）
- sp_iqpassword 过程（第 430 页）

sp_iqaddlogin 过程

向指定的 登录策略添加一个新的 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 ]
```

用法

表 100. 参数

参数	描述
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 (口令不过期)。
policy_name	(可选) 按照命名登录策略创建用户。如果未指定，则按照 root 登录策略创建用户。

使用 **sp_iqaddlogin** 创建并设置为在一天后过期的 **username_in/pwd** 在第二天全天有效，而在随后的一天无效。换句话说，今天创建并设置为在 *n* 天后过期的登录在日期变为第 *(n+1)* 天时将不再可用。

权限
需要具有 **DBA** 权限。

描述
添加新的 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。如果用户对于数据库已经拥有用户 ID，但不在 **ISYSUSER** 中（例如，如果使用 **GRANT CONNECT** 语句或 Sybase Central 添加了用户 ID），则 **sp_iqaddlogin** 会将该用户 添加到该表中。

如果在调用过程时 未指定登录策略的名称，则 Sybase IQ 将为用户分配 root 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。Sybase 建议使用 **CREATE USER** 创建新用户，但为了实现向后兼容性，还支持 **sp_iqaddlogin**。

请参见《参考：语句和选项》>“SQL 语句”>“CREATE USER 语句”。

示例
这些调用将按照名为 **expired_password** 的登录策略添加用户 **rose**，口令为 **irk324**。此示例假设 **expired_password** 登录策略已经存在。

```
call sp_iqaddlogin('rose','irk324','ON','expired_password')
sp_iqaddlogin 'rose','irk324','ON','expired_password'
```

另请参见

- sp_expireallpasswords 过程（第 338 页）
- sp_iqcopyloginpolicy 过程（第 366 页）
- sp_iqmodifyadmin 过程（第 423 页）
- sp_iqmodifylogin 过程（第 424 页）
- sp_iqpassword 过程（第 430 页）
- sp_iqdroplogin 过程（第 384 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

表 101. 参数

参数	描述
backup_id	指定备份操作的事务标识符。

注意： 可以从 SYSIQBACKUPHISTORY 表中获取 backup_id 值。运行以下查询： select * from sysiqbackuphistory

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

sp_iqbackupdetails 返回以下内容：

表 102. sp_iqbackupdetails 列

列名	描述
backup_id	备份事务的标识符。
backup_time	备份的时间。
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”。

列名	描述
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	SYSIQBACKUPHISTORYDETAIL 的 dbspace 的名称。如果该数据库空间名称与给定 dbspace_id 的 SYSDBSPACE 中的数据库空间名称匹配，则该名称存在。否则为“空值”。
dbspace_rwstatus	“读写”或“只读”。
dbspace_createid	Dbpace 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。值为“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	正在备份的 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	“读写”或“只读”。
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 (KB)。
dbfile_backup_size	dbfile 备份的大小 (KB)。
dbfile_path	SYSBACKUPDETAIL 的 dbfile 路径，如果该路径与给定 dbspace_id 和 dbfile_id 的 SYSDBFILE 中的物理文件路径（“file_name”）匹配，则该路径存在。否则为“空值”。

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 553 页）

sp_iqbackupsummary 过程

总结执行的备份操作。

语法

```
sp_iqbackupsummary [ timestamp or backup_id ]
```

参数

表 103. 参数

参数	描述
timestamp or backup_id	指定报告备份操作的间隔。如果指定时间戳或备份 ID，则只返回那些具有大于或等于输入时间的 backup_time 的记录。如果不指定时间戳，则过程返回 ISYSIQBACKUPHISTORY 中的所有备份记录。

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

sp_iqbackupsummary 返回以下内容：

表 104. 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'
```

sp_iqcardinality_analysis 过程

通过分析表中各列的基数并建议要添加的索引, 来帮助您生成索引。

sp_iqcardinality_analysis 可以生成可选的 SQL 脚本, 该脚本包含用于创建索引的可立即执行的 SQL 语句。

sp_iqcardinality_analysis 单独运行索引顾问。索引顾问根据实际查询中列的使用情况提出相关建议, 但仅在该列具有 LF 或 HG 索引时考虑基数, 并且非重复计数允许重新构建单字节和双字节 FP 索引。**sp_iqcardinality_analysis** 在任何情况下均考虑基数, 但不在其索引建议列表中包含索引顾问建议。

sp_iqcardinality_analysis 不在其建议列表中包含 HNG 和 CMP 索引。建议的索引包括:

- LF 索引
- HG 索引
- 1 字节 FP

- 双字节 FP
- 3 字节 FP
- 将双字节 FP 转换为单字节 FP
- 主键约束 (HG 索引)
- 唯一约束 (HG 索引)
- DATE/TIME/DTTM 索引
- WD 索引
- 词索引

注意： 针对宽表执行 `sp_iqcardinality_analysis` 可能要花费几分钟时间，具体取决于列数据类型的复杂性以及表上已经存在的索引。

语法

```
sp_iqcardinality_analysis ([ 'table_name' ], [ 'table_owner' ], [ 'script' ])
```

参数

参数	描述
table_name	表的名称。
table_owner	表所有者的姓名。如果未指定此参数，该过程将查找当前用户所拥有的表。
脚本	生成一个 SQL 脚本并显示索引建议，该脚本中包含用于创建索引的可立即执行的 SQL 语句。如果未指定该选项，主控台将显示： • table_name • table_owner • column_name • cardinality • index_type • index recommendation

用法

如果未指定任何参数，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 ' ';
```

权限

需要具有 DBA 权限。

示例 1

```
sp_iqcardinality_analysis 'onebytefp', 'DBA'
```


表 105. 主控台输出示例 1

table_name	table_owner	column_name	cardinality	index type	index recommendation
onebytefp	DBA	c1	10	单字节 FP	-- “c1” 列不具有单字节 FP 索引并且基数小于 256。-- 可以创建单字节 FP 索引。-- 调用以下存储过程： sp_iqrebuildindex 'onebytefp','column c1 255'
onebytefp	DBA	c1	10	LF	-- “c1” 列不具有 LF 索引并且基数小于 1000。-- 可以使用 CREATE INDEX 语句创建 LF 索引： CREATE LF INDEX IQ_T400_c1_LF ON DBA.onebytefp (c1)

示例 2

```
sp_iqcardinality_analysis 'onebytefp', 'DBA', 'script'
```

表 106. 主控台输出示例 2

index recommendation
-- “c1” 列不具有单字节 FP 索引并且基数小于 256。-- 可以创建单字节 FP 索引。-- 调用以下存储过程： sp_iqrebuildindex 'onebytefp','column c1 255'。-- “c1” 列不具有 LF 索引并且基数小于 1000。-- 可以使用 CREATE INDEX 语句创建 LF 索引： CREATE LF INDEX IQ_T400_c1_LF ON onebytefp (c1)

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。

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 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 ]'
```

这是 **sp_iqcheckdb** 的常规语法。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。下面分别列出 每种模式的语法。如果参数字符串中既没有指定模式也没有指定目标，则 Sybase IQ 会返回错误消息：

```
At least one mode and target must be specified to DBCC.
```

参数

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* }

分配模式

```
sp_iqcheckdb 'allocation target [ resources resource-percent ]'
```

检查模式

```
sp_iqcheckdb 'check target [ resources resource-percent ]'
```

检验模式

```
sp_iqcheckdb 'verify target [ resources resource-percent ]'
```

沙漏模式

```
sp_iqcheckdb 'dropleaks target [ resources resource-percent ]'
```

用法

表 107. 参数

参数	描述
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 <i>index-type</i> 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 <i>owner</i>，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 <i>table</i>，则 <i>index-name</i> 必须是唯一的。
table-name	可能包含所有者限定符：[owner.]table-name 如果未指定 <i>owner</i>，则系统将以该顺序替代当前用户及数据库所有者 (dbo)。 <i>table-name</i> 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 <i>table-name</i> 或 <i>index-name</i> 参数用双引号引起来： <pre>sp_iqcheckdb 'check index "dbo.sstab.i2" resources 75'</pre> 如果未指定 <i>owner</i>，则系统将以该顺序替代当前用户及数据库所有者 (dbo)。 <i>table-name</i> 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 <i>table-name</i> 或 <i>index-name</i> 参数用双引号引起来： <pre>sp_iqcheckdb 'check index "dbo.sstab.i2" resources 75'</pre>
partition-name	<i>partition-name</i> 参数不包含限定符。如果它包含空格，请将其放在双引号内。 分区过滤器会导致 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	<i>dbspace-name</i> 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。

参数	描述
resource-percent	输入参数 <i>resource-percent</i> 必须是大于零的整数。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 <i>resource-percent</i> = 100（缺省值），则每个 CPU 创建一个线程。如果 <i>resource-percent</i> > 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。

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。

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

`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。

分配模式

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 `dbspace` 的块映射信息检查分配情况。不检查索引一致性。

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）。

检测数据库或 `dbspace` 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）。

当目标是分区表时，分配模式将：

- 检查所有表分区分位图的元数据
- 检查表分配位图的元数据
- 检验块映射条目与表的分配位图是否一致
- 检验表的分区分位图没有重叠
- 检查在表的分区分位图中定义的行是否构成表的存在位图的超集
- 检查在表的分区分位图中定义的行是否构成表的分配位图的超集

注意： 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 `sp_iqcheckdb` 不能检查所有分配问题。

在以下情况下，请以分配模式运行：

- 若要检测重复块或无主块（使用数据库或者特定表或索引作为目标）
- 如果遇到页头错误

DBCC 选项 **resetclocks** 仅用于 分配模式。**resetclocks** 与强制恢复 一起使用，以将 Multiplex 辅助服务器转换成 事务协调器。有关 Multiplex 功能的信息，请参见《使用 Sybase IQ Multiplex》。一旦数据库内部版本控制时钟 落后，**resetclocks** 将更正这些时钟的值。除非曾咨询过 Sybase IQ 技术支持，否则请勿将 **resetclocks** 选项用于任何其它目的。

resetclocks 选项必须在单用户模式下运行，并且只能与 DBCC 语句 **allocation database** 一起使用。**resetclocks** 的语法为：

```
sp_iqcheckdb 'allocation database resetclocks'
```

检查模式

检验可以读取整个数据库、特定索引、特定索引类型、特定表、特定分区或特定 dbspace 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位图。

如果在运行查询时返回元数据错误、计数为空错误或离散值个数错误，则在检查模式下运行。

检验模式

检验整个数据库、特定索引、特定索引类型、特定表、特定分区或特定数据库空间的非 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，则检验模式检测以下不一致问题：

- 缺失键 - 某键在 FP 中，但不在非 FP 索引中。
- 附加键 - 某键在非 FP 索引中，但不在 FP 索引中。
- 缺失行 - 某行在 FP 中，但不在非 FP 索引中。
- 附加行 - 某行在非 FP 索引中，但不在 FP 索引中。

如果指定的目标仅包含 FP 页的一个子集，则检验模式仅可以检测以下不一致问题：

- 缺失键
- 缺失行

如果目标是分区表，则检验模式还检验已将表或表分区中的每一行分配给正确的分区。

如果在运行查询时返回元数据错误、计数为空错误或离散值个数错误，则在检验模式下运行。

注意： **sp_iqcheckdb** 不检查参照完整性，或在违背参照完整性的情况下进行修复。

沙漏模式

当 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。

有关在 Multiplex 上运行沙漏模式的信息，请参见《使用 Sybase IQ Multiplex》。

以下示例说明 **sp_iqcheckdb** 过程的用法。

示例 1

在此示例中，**sp_iqcheckdb** 将检查整个数据库的分配情况：

```
sp_iqcheckdb 'allocation database'
```

示例 2

在第二个示例中，**sp_iqcheckdb** 将对索引 **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，如下命令所示：

```
sp_iqcheckdb 'verify index i1 index i2 index dbo.t1.i3'
```

示例 3

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。在下面的示例中，**sp_iqcheckdb** 将对表 **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'
```

示例 4

此示例检查数据库中类型为 **FP** 的所有索引：

```
sp_iqcheckdb 'check indextype FP database'
```

示例 5

下面的示例检验表 **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'
```

示例 6

以下示例说明 **sp_iqcheckdb** 输出中三条“LVC 单元”消息中的一条消息：

```
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：

- Unowned LVC cells
- 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 是宽度大于 255 的 VARCHAR 或 VARBINARY 列。LONG BINARY (BLOB) 和 LONG VARCHAR (CLOB) 也使用 LVC。

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性能的高低而有所不同。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数据库空间目标数。

下表总结了四种 **sp_iqcheckdb** 模式的操作和输出。

表 108. 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	仅输出分配统计信息	每小时 4 TB
检查	分配错误 大部分索引错误	所有可用统计信息	每小时 60GB
检验	分配错误 所有索引错误	所有可用统计信息	每小时 15GB
沙漏	分配错误	仅输出分配统计信息	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 (*****) 指示，并且仅当检测到错误时才会显示。

sp_iqcheckdb 的输出 还会复制到 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。

输出示例

下面是当您运行 **sp_iqcheckdb 'allocation database'** 并且有泄漏空间时所看到的输出示例。泄漏的空间是按照数据库空闲列表（内部分配映射）分配但 DBCC 发现不属于任何数据库对象的块。在此示例中，DBCC 报告了 32 处泄漏块。

Stat	Value	Flags
=====	=====	
=====		
DBCC Allocation Mode Report		
=====	=====	
=====		

** DBCC Status		Errors Detected	*****
=====		=====	
=====			
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			
=====		=====	
=====			
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			

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

权限

无。DBA 会看到为所有组和用户设置的所有永久性选项，并会看到为 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。

用法

不需要参数。对已更改为非缺省值的每个选项返回一行。输出将先按选项名称然后按用户名排序。

描述

对于已连接的用户，**sp_iqcheckoptions** 存储过程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。**sp_iqcheckoptions** 将考虑所有 Sybase IQ 和 SQL Anywhere 数据库选项。Sybase IQ 会修改某些 SQL Anywhere 选项缺省值，并且这些修改后的缺省值会成为新的缺省值。除非新的 Sybase IQ 缺省值再次发生更改，否则 **sp_iqcheckoptions** 不会列出该选项。

当 **sp_iqcheckoptions** 运行时，DBA 会查看为所有组和用户设置的所有永久性选项，并查看为 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 109. sp_iqcheckoptions 列

列名	描述
User_name	已设置其选项的用户或组的名称。在创建数据库时，系统将为 public 组设置所有选项。系统将显示为除 public 以外的组或用户设置的任何选项。
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，系统将临时选项 APPEND_LOAD 设置为 ON，并且将 mygroup 组的选项 MAX_WARNINGS 设置为 9。系统将用户 joel 的 MAX_WARNINGS 设置为临时值 55。

在第一个示例中，**sp_iqcheckoptions** 由 DBA 运行。

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	Append_Load	ON	OFF	Temporary
DBA	Checkpoint_time	20	60	Temporary
DBA	Connection_authent	Company=MyComp;		Temporary

DBA	Login_procedure	Application=DBTools;Signa		
nent		DBA.sp_iq_proce	sp_login_envir	Perma
PUBLIC	Login_procedure	DBA.sp_iq_proce		
nent		sp_login_envir		Perma
mygroup	Max_Warnings	9	281474976710655	Permanent
DBA	Thread_count	25	0	Temporary

在第二个示例中，**sp_iqcheckoptions** 由用户 **joel** 运行。

User_name	Option_name	Current_value	Default_value	Optio
n_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		
joel	Login_procedure	DBA.sp_iq_proce	sp_login_envir	Perma
nent				
PUBLIC	Login_procedure	DBA.sp_iq_proce	sp_login_envir	Perma
nent				
joel	Max_Warnings	55	281474976710655	Temporary
joel	Thread_count	25	0	Temporary

sp_iqclient_lookup 过程

允许客户端应用程序确定对负责某个特定数据流的 Sybase IQ 用户帐户，就像是从源自某个特定客户端 IP 地址/端口的网络分析程序中观察。

语法

```
sp_iqclient_lookup [ 'IPaddress' ], [ Port ], [ UserID ]
```

参数

表 110. 参数

参数	描述
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 Sybase IQ 用户 ID。

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

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。

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' ]
```

用法

语法	描述
语法 1	如果指定 <i>table_owner</i> ，而不指定 <i>table_name</i> ，则必须用空值代替 <i>table_name</i> 。例如， sp_iqcolumn NULL,DBA。
语法 2	参数可以任何顺序进行指定。将 ' <i>table_name</i> '和 ' <i>table_owner</i> '用单引号引起来。

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

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。

表 111. sp_iqcolumn 列

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

列名	描述
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
标度	数值数据类型的标度
nulls	如果列可以包含 NULLS 则为 “Y”，如果列不能包含 NULLS 则为 “N”
default	如果列是标识/自动增量列，则是 'Identity/Autoincrement'，如果不是，则为空。
cardinality	按索引进行的离散值个数（如果已知）
est_cardinality	不同的值的估计数目，如果是在 MINIMIZE_STORAGE 选项为 ON 的情况下创建了列，则自动设置为 255，或者是在 IQ UNIQUE 约束的 CREATE TABLE 中指定的用户提供值。
位置	TEMP = IQ 临时存储，MAIN = IQ main 存储，SYSTEM = Catalog 存储
isPartitioned	如果列属于分区表并有一个或多个分区的 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 COMMENT 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 362 页）
- sp_iqdatatype 过程（第 369 页）
- sp_iqevent 过程（第 390 页）
- sp_iqhelp 过程（第 395 页）
- sp_iqindex 和 sp_iqindex_alt 过程（第 401 页）
- sp_iqjoinindex 过程（第 415 页）
- sp_iqpkeys 过程（第 431 页）
- sp_iqprocparm 过程（第 436 页）
- sp_iq_reset_identity 过程（第 443 页）
- sp_iqtable 过程（第 460 页）
- sp_iqview 过程（第 474 页）

sp_iqcolumn 过程示例

使用示例作为 **sp_iqcolumn** 用法的参考。

以下两种语法变化形式都会返回 **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
nulls  default
Departments  GROUPO      DepartmentID     integer      4      0
N      (NULL)
Departments  GROUPO      DepartmentName   char          40     0
N      (NULL)
Departments  GROUPO      DepartmentHead   integer      4      0
Y      (NULL)
cardinality  est_cardinality  location  isPartitioned  remarks  check
5            5            Main      N              (NULL)   (NULL)
0            5            Main      N              (NULL)   (NULL)
5            5            Main      N              (NULL)   (NULL)
```

以下语法变化形式会返回表所有者 DBA 拥有的所有表中的所有列。

```
sp_iqcolumn table_owner='DBA'
```

sp_iqcolumnuse 过程

报告负载访问的列的详细使用信息。

语法

```
sp_iqcolumnuse
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

不会报告在 SYSTEM 中创建的表的列。

表 112. 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
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）。

有关索引的详细信息，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”>“INDEX_ADVISOR 选项”。

示例

sp_iqcolumnuse 过程的输出样本：

TableName	NRef	ColumnName	Owner	UID	LastDT
orders		o_orderdate	DBA		20070917
22:41:22..	13			151	
orders		o_shippriority	DBA		20070917
22:41:22..	13			154	
lineitem		l_orderkey	DBA		20070917
22:41:22..	13			186	
lineitem		l_extendedp..	DBA		20070917
22:41:22..	13			191	
lineitem		l_discount	DBA		20070917
22:41:22..	13			192	
lineitem		l_shipdate	DBA		20070917
22:41:22..	13			196	
#tmp1		expression	DBA	10000000001218	20070917
22:57:36..	1				
#tmp1		expression	DBA	10000000001222	20070917
22:41:58..	1				
...					

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 404 页）
- sp_iqindexuse 过程（第 413 页）
- sp_iqtableuse 过程（第 465 页）
- sp_iqunusedcolumn 过程（第 469 页）
- sp_iqunusedindex 过程（第 470 页）
- sp_iqunusedtable 过程（第 472 页）
- sp_iqworkmon 过程（第 479 页）

sp_iqconnection 过程

显示有关连接和版本的信息，包括 哪些用户使用临时 dbspace、哪些用户使用 版本、连接在 Sybase IQ 中做什么、连接状态、数据库版本状态等等。

语法

sp_iqconnection [*connhandle*]

用法

输入参数 *connhandle* 相当于 **Number** 连接属性并且是连接的 ID 号。

connection_property 系统函数返回连接 ID：

```
SELECT connection_property ( 'Number' )
```

当用有效输入参数 *connhandle* 调用时，**sp_iqconnection** 仅对该连接返回一行。

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

sp_iqconnection 对每个活动连接 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 Sybase IQ 引擎的 Sybase IQ 端的连接数据。各行按 ConnCreateTime 排序。

MPXServerName 列存储与 Multiplex 节点间通信 (INC) 有关的信息，如下所示：

表 113. MPXServerName 列值

服务器运行在	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	- 空值用于本地/用户连接 - 包含用于每一 INC 连接（根据需要或专用检测信号连接）的辅助节点服务器名（连接源）的值。
Multiplex 辅助服务器	- 空值用于本地/用户连接 - 包含协调器服务器名（连接源）的值。

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 Sybase IQ 的连接 属性。此示例 显示如何指定特定于 IQ 的连接参数，其中的 **myconnection** 为 IQ 连接名：

```
p.put("RemotePWD",",",CON=myconnection");
```

请参见 SQL Anywhere “Server - 编程”。

表 114. sp_iqconnection 列

列名	描述
ConnHandle	连接的 ID 号。
Name	服务器的名称。
Userid	连接的用户 ID。

列名	描述
LastReqTime	指定连接的最后一个请求的启动时间。
ReqType	最后一个请求的类型的字符串。
IQCmdType	在 Sybase IQ 端执行的当前命令（如果有）。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 存储中的数据的 DDL 和 DML 命令、内部 IQ 游标命令和特殊控制命令（例如 OPEN 和 CLOSE DB 、 BACKUP 、 RESTORE 等）组成。
LastIQCmdTime	此连接上最后一个 IQ 命令在 Sybase IQ 引擎的 IQ 端上的启动或完成时间。
IQCursors	该连接上在 IQ 存储中打开的游标数。
LowestIQCursorState	IQ 游标状态（如果有）。如果在连接上存在多个游标，则显示的状态是所有游标的最低游标状态；也就是说，距离完成最远。游标状态反映内部 Sybase IQ 实现详细信息，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。如名称所示，游标状态在操作结束时更改。例如，PREPARED 状态指示游标正在执行。
IQthreads	当前分配给连接的 Sybase IQ 线程数。某些线程可能已分配，但处于空闲状态。此列可以帮助您确定哪些连接正在使用大多数资源。
TxnID	连接上当前事务的事务 ID。该事务 ID 与 .iqmsg 文件中的 BeginTxn、CmtTxn 和 PostCmtTxn 消息以及打开数据库时记录的 Txn ID Seq 中显示的事务 ID 相同。
ConnCreateTime	创建连接的时间。
TempTableSpaceKB	此连接在处理 IQ 临时表中存储的数据时所用的 IQ 临时存储区空间的字节数 (KB)。
TempWorkSpaceKB	此连接在处理排序、散列和临时位图等工作空间时所用的 IQ 临时存储空间的字节数 (KB)。作为 Sybase IQ 临时表的索引的一部分的位图或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数连接 ID。这是服务器会话内唯一单调增加的整数。
satoiq_count	内部计数器，用于显示从 Sybase IQ 引擎的 SQL Anywhere 端到 IQ 端的穿过数。这在确定连接活动时可能偶尔有用。结果集将在缓冲区中以行返回，并且不会每行增加 satoiq_count 或 iqtosa_count 一次。
iqtosa_count	内部计数器，用于显示从 Sybase IQ 引擎的 IQ 端到 SQL Anywhere 端的穿过数。这在确定连接活动时可能偶尔有用。

列名	描述
CommLink	连接的通信链接。这是 Sybase IQ 支持的网络协议之一，对于相同计算机连接则是本地协议。
NodeAddr	客户端/服务器连接中客户端的节点。
LastIdle	请求间隔时钟数。
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。如果非 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，则为空值。

示例

以下是 `sp_iqconnection` 输出的示例：

ConnHandle	Name	Userid	LastReqTime		ReqType		
1	'SQL_DBC_100525210'	'DBA'	'2011-03-28	09:29:24.466'	'OPEN'		
IQCmdType		LastIQCmdTime	IQCursors	LowestIQCursorState			
'IQUILITYOPENCURSOR'		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	LSName
34	43	2	'local'	'	244	(NULL)	Finance_LS

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' ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

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

下面的输出样本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 356 页）
- sp_iqdatatype 过程（第 369 页）
- sp_iqevent 过程（第 390 页）
- sp_iqhelp 过程（第 395 页）
- sp_iqindex 和 sp_iqindex_alt 过程（第 401 页）
- sp_iqjoinindex 过程（第 415 页）
- sp_iqpkeys 过程（第 431 页）
- sp_iqprocparm 过程（第 436 页）
- sp_iq_reset_identity 过程（第 443 页）
- sp_iqtable 过程（第 460 页）
- sp_iqview 过程（第 474 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 ]
```

用法

输入参数 *connhandle* 相当于 **Number** 连接属性并且是连接的 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。

权限

需要具有 **DBA** 权限。必须向没有 **DBA** 权限的用户授予 **EXECUTE** 权限才能运行存储过程。

如果您对作为分布式查询处理 (DQP) 工作线程的领导节点的服务器没有 **MULTIPLEX ADMIN** 或 **DBA** 权限，系统将不会显示这些线程。

描述

DBA 使用 **sp_iqcontext** 可以在任何给定时刻 确定哪些语句在系统上运行， 并可以标识发出相应语句的用户和连接。通过 这些信息，可以使用此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和事务信息
- 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
- 使用连接信息将 **sp_iqcontext** 中的语句 文本与 **.iqmsg** 文件中的等效行 匹配，后者在当 **Sybase IQ** 可以收集相应信息时包含 相应的查询计划
- 将语句文本与可能产生的 **IQ** 堆栈跟踪 (**stktrc-yyyyymmdd-hhnnss_#.iq**) 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 115. sp_iqcontext 列

列名	描述
ConnOrCursor	CONNECTION、CURSOR 或 DQP。
ConnHandle	连接的 ID 号，或者 0（对于 DQP）。
名称	服务器的名称（领导节点名称）。

列名	描述
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。这是服务器会话内唯一单调增加的整数。
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，表示 IQGovernPriority 不适用于此操作。用数据库选项 IQGOVERN_PRIORITY 设置 IQGovernPriority 值。 对于 DQP 连接，此列显示“无命令”。
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”。
属性	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带参数的情况下发出 `sp_iqcontext` 时的输出，该输出是对当前所有连接产生的结果。出于空间考虑，列名被截断。

```
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 显示了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大 4096 个字符) 。用户命令片段之前的 2 表示这是中优先级查询。

连接句柄 (对于此示例中的第一个连接为 2) 标识 `-zr` 日志中的结果。IQ 连接 ID (对于此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) 显示了 DQP 工作线程。该工作线程连接正在运行两个不变的条件。

最后一行 (TxnID 10001) 显示了连接正在运行散列连接。

另请参见

- CONNECTION_PROPERTY 函数 [系统] (第 133 页)
- sp_iqshowpsexec 过程 (第 446 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'
```

语法 3

```
sp_iqcopyloginpolicy existing-policy-name, new-policy-name
policy_name ]
```

用法

表 116. 参数

参数	描述
existing policy name	要复制的登录策略。

参数	描述
new policy name	要创建的新登录策略的名称 (CHAR(128))。

权限

需要具有 DBA 权限。

示例

以下存储过程通过从名为 *lockeduser* 的现有登录策略复制登录策略选项值创建名为 *root* 的新登录策略。

```
call sp_iqcopyloginpolicy ('root','lockeduser')
```

另请参见

- sp_expireallpasswords 过程（第 338 页）
- sp_iqaddlogin 过程（第 338 页）
- sp_iqmodifyadmin 过程（第 423 页）
- sp_iqmodifylogin 过程（第 424 页）
- sp_iqpassword 过程（第 430 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 ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

参数	描述
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 将返回指定连接中所有游标的相关信息。

可以在没有任何参数的情况下调用 **sp_iqcursorinfo** 过程。如果没有指定任何参数，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。如果两个参数都

加以指定，则 **sp_iqcursorinfo** 将报告具有指定名称并且在指定连接中的所有游标的相关信息。

如果不指定第一个参数，但指定第二个参数，则必须用 **NULL** 替换省略的参数。例如，**sp_iqcursorinfo NULL, 1**。

表 117. **sp_iqcursorinfo** 用法示例

语法	输出
sp_iqcursorinfo	显示有关当前在服务器上打开的所有游标的信息
sp_iqcursorinfo 'cursor1'	显示有关在所有连接中名为 cursor1 的所有游标的信息
sp_iqcursorinfo NULL, 3	显示有关在连接 3 中的所有游标的信息
sp_iqcursorinfo 'cursor2', 4	显示有关在连接 4 中名为 cursor2 的所有游标的信息

描述

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 118. **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
IQConnID	作为 .iqmsg 文件中所有消息的一部分显示的十位数连接 ID。此数字是服务器会话内唯一单调增加的整数。
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

列名	描述
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

请参见《系统管理指南第二卷》>“使用过程和批处理”>“在过程中使用游标”。有关事务中的游标的信息，请参见《系统管理指南第一卷》>“事务和版本控制”>“事务中的游标”。

有关 CURSOR 数据库选项，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”。

有关 CURSOR 语句，请参见《参考：语句和选项》>“SQL 语句”中的条目。

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	
---	200000000	30000000000	DBA	test1	call
proc1()	call				
proc2()					

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 ]
```

权限
需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 119. 参数

参数	描述
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: 仅显示有关系统定义的数据类型（由用户 SYS 或 dbo 拥有的数据类型）的信息 • ALL: 显示有关用户和系统数据类型的信息 • 其它任何值: 显示有关用户数据类型的信息

可以在没有任何参数的情况下调用 **sp_iqdatatype** 过程。如果不指定参数，则缺省情况下仅显示有关用户定义数据类型（dbo 或 SYS 不拥有的数据类型）的信息。

如果未指定前两个参数中的任何一个参数，但指定了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 120. sp_iqdatatype 用法示例

语法	输出
sp_iqdatatype	显示数据库中所有用户定义数据类型的相关信息
sp_iqdatatype country_t	显示名为 country_t 的用户定义数据类型的相关信息
sp_iqdatatype non_existing_type	不返回任何行，因为数据类型 non_existing_type 不存在。
sp_iqdatatype NULL, DBA	显示 DBA 拥有的所有用户定义数据类型的相关信息
sp_iqdatatype country_t, DBA	显示 DBA 拥有的数据类型 country_t 的相关信息
sp_iqdatatype rowid	rowid 是系统定义的数据类型。如果没有同样名为 rowid 的用户定义数据类型，则不返回任何行。（缺省情况下，仅返回用户定义数据类型。）

语法	输出
sp_iqdatatype rowid, SYS	不返回任何行，因为数据类型 rowid 不是用户定义数据类型（缺省情况下，仅返回用户定义数据类型）。
sp_iqdatatype NULL, NULL, SYSTEM	显示所有系统定义的数据类型（由 dbo 或 SYS 拥有）的相关信息
sp_iqdatatype rowid, NULL, SYSTEM	显示系统数据类型 rowid 的相关信息
sp_iqdatatype NULL, NULL, 'ALL'	显示有关用户定义数据类型和系统数据类型的信息

描述

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 121. sp_iqdatatype 列

列名	描述
type_name	数据类型的名称
creator	数据类型的所有者
nulls	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。
width	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 356 页）
- sp_iqconstraint 过程（第 362 页）
- sp_iqevent 过程（第 390 页）
- sp_iqhelp 过程（第 395 页）
- sp_iqindex 和 sp_iqindex_alt 过程（第 401 页）
- sp_iqjoinindex 过程（第 415 页）
- sp_iqpkeys 过程（第 431 页）
- sp_iqprocparm 过程（第 436 页）
- sp_iq_reset_identity 过程（第 443 页）
- sp_iqtable 过程（第 460 页）
- sp_iqview 过程（第 474 页）

sp_iqdbsize 过程

显示当前数据库的大小。

语法

```
sp_iqdbsize([ main ] )
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

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。

表 122. sp_iqdbsize 列

列名	描述
数据库	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，最初这是以 MB 为单位指定的。使用相应 IQ 页大小 和该 IQ 页大小的对应块大小将此 MB 数量转换为块数。Physical Blocks 列反映每个 Sybase IQ dbspace 大小的累计总数，以块为单位表示。
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 sp_iqdbsize 列中的 Physical Blocks）乘以块大小。块大小取决于 IQ 页大小。

列名	描述
Pages	在内存中表示在表和连接索引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 Compressed Pages（下一 sp_iqdbsize 列）的值。
Compressed Pages	将表和连接索引中的数据 以及这些对象的元数据存储在磁盘上所需的总 IQ 页数。此值始终小于或等于 Pages（前一 sp_iqdbsize 列）的值，因为当将 IQ 页从内存写入磁盘时 Sybase IQ 会压缩页。 sp_iqdbsize Compressed Pages 列表示压缩的页数。
NBlocks	用于存储表和连接索引中的数据的总块大小。此值始终小于或等于 sp_iqdbsize Physical Blocks 值。
Catalog Blocks	用于存储表和连接索引的元数据的总块大小。

可以在 Multiplex 环境中使用此过程。请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用”> “系统过程”> “sp_iqdbsize 过程”。

示例

显示数据库 iqdemo 的大小信息：

```
sp_iqdbsize
Database
PhysicalBlocks KBytes Pages CompressedPages NBlocks CatalogBlocks
=====
/system1/sybase/IQ-15_3/demo/iqdemo.db
1280 522 688 257 1119 18
```

sp_iqdbspace 过程

显示有关每个 IQ dbspace 的详细信息。

语法

```
sp_iqdbspace [ dbspace-name ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

可以使用 **sp_iqdbspace** 中的信息确定是否必须移动数据，对于已移动的数据，旧版本是否已释放。**sp_iqdbspace** 显示以下信息：

表 123. sp_iqdbspace 列

列名	描述
DBSpaceName	在 CREATE DBSPACE 语句中指定的数据库空间的名称。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。
DBSpaceType	dbspace 的类型 (MAIN 、 SHARED_TEMP 或 TEMPORARY) 。
Writable	T (可写入) 或 F (不可写入) 。
Online	T (联机) 或 F (脱机) 。
Usage	dbspace 中所有文件当前使用的 dbspace 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	可以添加到 dbspace 中所有文件的保留空间总数。
NumFiles	dbspace 中的文件数。
NumRWFiles	dbspace 中的读/写文件数。
Stripingon	T (On) 或 F (Off)。
StripeSize	如果磁盘条带化打开，则这是在移动到下一个 dbspace 之前写入 dbspace 的数据量。
BlkTypes	由用户数据和内部系统结构共同使用的空间。
OkToDrop	“Y” 表示 dbspace 可以删除；否则为 “N” 。

BlkTypes 块类型标识符的值：

表 124. sp_iqdbspace 块类型

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

标识符	块类型
I	索引建议存储
M	Multiplex CM*
O	旧版本
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。

有关 Multiplex 功能的信息，请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用”> “系统过程”> “sp_iqdbspace 过程”。

示例

以下输出显示有关数据库空间的信息。

```
sp_iqdbspace;
```

注意： 以下示例演示 **iqdemo** 数据库中的对象，以便更好地描述输出。请注意，**iqdemo** 包括一个名为 **iq_main** 的用户数据库空间示例，您自己的数据库中可能不存在该示例。

DBSpaceName	DBSpaceType	Writable	Online	Usage	TotalSize	Reserve	NumFiles	Num
IQ_MAIN	MAIN	T	T	55	75M	200M	1	1
IQ_SYSTEM_MAIN	MAIN	T	T	21	300M	50M	1	1
IQ_SYSTEM_TEMP	TEMPORARY	T	T	1	100M	50M	1	1

另请参见

- sp_iqindexinfo 过程（第 407 页）
- sp_iqdbspaceinfo 过程（第 376 页）
- sp_iqspaceinfo 过程（第 448 页）

sp_iqdbspaceinfo 过程

显示在指定的表或连接索引中使用的每个对象和子对象的大小。

语法

```
sp_iqdbspaceinfo [ dbspace-name ] [ , owner_name ] [ ,  
object_name ] [ , object-type ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

参数	描述
dbspace_name	如果指定，则 sp_iqdbspaceinfo 显示在指定的数据库空间中有任何组件的每个表的一行。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，则 sp_iqdbspaceinfo 仅显示使用指定所有者的表和连接索引的输出。如果未指定，则 sp_iqdbspaceinfo 显示数据库中所有用户的表和连接索引的相关信息。
object_name	表或连接索引的名称。如果未指定，则 sp_iqdbspaceinfo 显示有关数据库中所有表和连接索引的信息。
object_type	有效的对象类型包括 table （有关表信息）或 joinindex （有关连接索引信息）。如果未指定，则对象类型缺省为 table 。

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。

sp_iqdbspaceinfo 存储过程支持解释 *dbspace_name*、*object_name* 和 *owner_name* 的通配符。它按与 **LIKE** 子句匹配查询内部的模式相同的方式，显示匹配给定模式的所有数据库空间的信息。

描述

sp_iqdbspaceinfo 为 DBA 显示每个数据库空间中驻留的对象使用的空间数量。DBA 可以使用此信息确定要删除某个 dbspace，必须先重新分配哪些对象。子对象列显示以后跟后缀 B、K、M、G、T 或 P 的整数数量报告的大小，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。

对于表，**sp_iqdbspaceinfo** 显示所有子对象的子对象大小信息（使用带有后缀 B、K、M、G、T 或 P 的整数数量）。对于连接索引，该过程显示连接索引及其所有相关子对象的大小信息。输出按 *dbspace_name*、*object_name* 和 *owner_name* 排序。

表 125. sp_iqdbspaceinfo 列

列名	描述
dbspace_name	dbspace 的名称。
object_type	对象的类型（仅限于 table 或 joinindex ）。
owner	对象所有者的名称。
object_name	位于 dbspace 上的对象（仅限于表和连接索引的对象）的名称。
object_id	对象的全局对象 ID。
id	对象的表 ID 或连接索引 ID。
columns	给定 dbspace 上的列存储空间的大小。
索引	给定 dbspace 上的索引存储空间的大小。不要使用系统生成的索引（例如，唯一约束中的 HG 索引或 FP 索引）。
元数据	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	表示数据库空间处于联机状态 (Y) 还是脱机状态 (N)。

注意： 如果对使用 -r 开关（只读）启动的服务器运行 sp_iqdbspaceinfo，会看到错误“Msg 13768, Level 14, State 0: SQL Anywhere Error -757: Modifications not permitted for read-only database”。应出现此行为。其它存储过程（例如 sp_iqdbspace、sp_iqfile、sp_iqdbspaceobjectinfo 或 sp_iqobjectinfo）上不会发生此错误。

可以在 Multiplex 环境中使用此过程。请参见《使用 Sybase IQ Multiplex》>“Multiplex 引用”>“系统过程”>“sp_iqdbspaceinfo 过程”。

示例

显示数据库的所有数据库空间内所有表中的所有对象和子对象的大小：

注意： 以下三个示例演示 iqdemo 数据库中的对象，以便更好地描述输出。请注意，iqdemo 包括一个名为 iq_main 的用户数据库空间示例，您自己的数据库中可能不存在该示例。

sp_iqdbspaceinfo						
dbspace_name	object_type	owner	object_name	object_id	id	
columns						
iq_main	table	DBA	emp1	3689	741	96K

iq_main		table	DBA	iq_dummy	3686	740	24K
iq_main		table	DBA	sale	3698	742	96K
iq_main		table	GROUP0	Contacts	3538		732
iq_main		table	GROUP0	Customers	3515		731
iq_main		table	GROUP0	Departments	3632	738	72K
iq_main		table	GROUP0	Employees	3641		739
iq_main		table	GROUP0	FinancialCodes	3612		736
iq_main		table	GROUP0	FinancialData	3621	737	96K
iq_main		table	GROUP0	Products			
3593	735	272K					
iq_main		table	GROUP0	SalesOrderItems	3580		734
iq_main		table	GROUP0	SalesOrders	3565		733
144K							
indexes	metadata	primary_key	unique_constraint	foreign_key	dbspace_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		

显示数据库的指定数据库空间中指定用户拥有的所有对象和子对象的大小:

sp_iqdbspaceinfo iq_main, GROUP0

dbspace_name	object_type	owner	object_name	object_id	id
columns					
iq_main	table	GROUP0	Contacts	3538	732
288K					
iq_main	table	GROUP0	Customers	3515	731
240K					
iq_main	table	GROUP0	Departments	3632	738
iq_main	table	GROUP0	Employees	3641	739
408K					
iq_main	table	GROUP0	FinancialCodes	3612	736
72K					
iq_main	table	GROUP0	FinancialData	3621	737
iq_main	table	GROUP0	Products	3593	735
272K					
iq_main	table	GROUP0	SalesOrderItems	3580	734
120K					
iq_main	table	GROUP0	SalesOrders	3565	733
144K					

indexes	metadata	primary_key	unique_constraint	foreign_key	dbspace
ace_online	0B	5.45M	24K	0B	48K
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

显示数据库的指定数据库空间中指定用户拥有的指定对象及其子对象的大小：

`sp_iqdbspaceinfo iq_main,GROUP0,Departments`

dbspace_name	object_type	owner	object_name	object_id	id
columns					
iq_main	table	GROUP0	Departments	3632	738
72K					
indexes	metadata	primary_key	unique_constraint	foreign_key	dbspace
ace_online					
0B	1.78M	24K	0B	48K	Y

另请参见

- `sp_iqindexinfo` 过程（第 407 页）
- `sp_iqdbspace` 过程（第 373 页）
- `sp_iqspaceinfo` 过程（第 448 页）

sp_iqdbspaceobjectinfo 过程

列出给定 `dbspace` 的类型表和连接索引的对象及其子对象（列、索引、元数据、主键、唯一约束、外键和分区）。

语法

`sp_iqdbspaceobjectinfo [ dbspace-name ] [ , owner_name ] [ , object_name ] [ , object-type ]`

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

所有参数都是可选的，并可能提供独立于其它参数的值的任何参数。

表 126. 参数

参数	描述
dbspace-name	如果指定，则 <code>sp_iqdbspaceobjectinfo</code> 仅显示指定数据库空间的输出。否则，显示数据库中所有 <code>dbspace</code> 的信息。

参数	描述
owner-name	对象的所有者。如果指定，则 sp_iqdbspaceobjectinfo 仅显示使用指定所有者的表和连接索引的输出。如果未指定，则 sp_iqdbspaceobjectinfo 显示数据库中所有用户的表和连接索引的相关信息。
object-name	表或连接索引的名称。如果未指定，则 sp_iqdbspaceobjectinfo 显示有关数据库中所有表和连接索引的信息。
object-type	有效的对象类型包括 table （有关表信息）或 joinindex （有关连接索引信息）。如果未指定，则对象类型缺省为 table 。

sp_iqdbspaceobjectinfo 存储过程支持解释 *dbspace_name*、*object_name* 和 *owner_name* 的通配符。它按与 **LIKE** 子句匹配查询内部的模式相同的方式，显示匹配给定模式的所有数据库空间的信息。

描述

对于表，**sp_iqdbspaceobjectinfo** 显示所有相关子对象的摘要信息。对于连接索引，该过程显示连接索引及其所有相关子对象的大小信息。存储过程的输出将按 *dbspace_name*、*owner* 和 *object_name* 排序。

sp_iqdbspaceobjectinfo 根据输入参数值显示以下信息：

表 127. **sp_iqdbspaceobjectinfo** 列

列名	描述
dbspace_name	dbspace 的名称。
dbspace_id	dbspace 的标识符。
object_type	表或连接索引。
owner	对象所有者的名称。
object_name	位于 dbspace 上的对象（仅限于表和连接索引的对象）的名称。
object_id	对象的全局对象 ID。
id	对象的表 ID 或连接索引 ID。
columns	位于给定 dbspace 上的表列的数量。如果一列或列分区之一位于 dbspace 上，则认为该 dbspace 上存在该列或列分区。结果以 n/N（超出表中总列数 N 之外的 n 在给定的 dbspace 上）的格式显示。
索引	位于给定 dbspace 上的表中用户定义的索引数。以 n/N（超出表中总索引数 N 之外的 n 在给定的 dbspace 上）的格式显示。如果是唯一约束，则不包含系统生成的索引，例如 FP 索引和 HG 索引。

列名	描述
元数据	布尔字段 (Y/N)，表示是否子对象的元数据信息也位于此 dbspace 上。
primary_key	布尔字段 (1/0)，表示是否表的主键（如果有）也位于此 dbspace 上。
unique_constraint	位于给定 dbspace 上的表中唯一约束的数量。以 n/N（超出表中总唯一约束数 N 之外的 n 在给定的 dbspace 上）的格式显示。
foreign_key	位于给定 dbspace 上的表中外键的数量。以 n/N（超出表中总外键数 N 之外的 n 在给定的 dbspace 上）的格式显示。
分区	位于给定 dbspace 上的表中分区数量。以 n/N（超出表中总分区数 N 之外的 n 在给定的 dbspace 上）的格式显示。

示例

注意： 以下两个示例演示 **iqdemo** 数据库中的对象，以便更好地描述输出。请注意，**iqdemo** 包括一个名为 **iq_main** 的用户数据库空间示例，您自己的数据库中可能不存在该示例。

显示有关数据库中某个特定数据库空间的信息：

sp_iqdbspaceobjectinfo iq_main						
dbspace_name	dbspace_id	object_type	owner	object_name	object_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	GROUP0	Contacts	3538	732 12/12
iq_main	16387	table	GROUP0	Customers	3515	731 10/10
iq_main	16387	table	GROUP0	Departments	3632	738 3/3
iq_main	16387	table	GROUP0	Employees	3641	739 21/21
iq_main	16387	table	GROUP0	FinancialCodes	3612	736 3/3
iq_main	16387	table	GROUP0	FinancialData	3621	737 4/4
iq_main	16387	table	GROUP0	Products	3593	735 8/8
iq_main	16387	table	GROUP0	SalesOrderItems	3580	734 5/5
iq_main	16387	table	GROUP0	SalesOrders	3565	733 6/6
indexes	metadata	primary_key	unique_constraint	foreign_key	partitions	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

显示有关数据库的某个特定数据库空间中指定用户拥有的对象的信息：

sp_iqdbspaceobjectinfo iq_main,GROUP0

dbspace_name	dbspace_id	object_type	owner	object_name	object_
id id columns					
iq_main	16387	table	GROUP0	Contacts	3538
732 2/12					
iq_main	16387	table	GROUP0	Customers	3515
731 10/10					
iq_main	16387	table	GROUP0	Departments	3632
738 3/3					
iq_main	16387	table	GROUP0	Employees	3641
739 21/21					
iq_main	16387	table	GROUP0	FinancialCodes	3612
736 3/3					
iq_main	16387	table	GROUP0	FinancialData	3621
737 4/4					
iq_main	16387	table	GROUP0	Products	3593
735 8/8					
iq_main	16387	table	GROUP0	SalesOrderItems	3580
734 5/5					
iq_main	16387	table	GROUP0	SalesOrders	3565
733 6/6					
indexes metadata	primary_key	unique_constraint	foreign_key	part	
itions 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	

使用 **sp_iqdbspaceobjectinfo** 可构造可用于移动对象的命令。在此示例中，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;
```

sp_iqdbstatistics 过程

报告最近一次执行 **sp_iqcheckdb** 的结果。

语法

```
sp_iqdbstatistics
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

显示在最近一次执行 **sp_iqcheckdb** 时收集的数据库统计信息。

有关 **sp_iqcheckdb** 的用法和 **sp_iqcheckdb** 输出解释的详细信息，请参见《系统管理指南第一卷》>“系统恢复和数据库修复”。

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	Errors Detected	*****
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		
=====	=====	
=====		

sp_iqdroplogin 过程

删除 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' )
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 128. 参数

参数	描述
userid	要删除的用户的用户 ID。

描述

sp_iqdroplogin 删除指定的用户。

示例

以下存储过程调用将删除用户 **rose**。

```
sp_iqdroplogin 'rose'  
sp_iqdroplogin rose  
call sp_iqdroplogin ('rose')
```

另请参见

- **sp_iqaddlogin** 过程（第 338 页）

sp_iqemptyfile 过程

清空 dbfile 并将该 dbfile 文件中的对象移动到同一个数据库空间中的另一个可用的读写 dbfile。

语法

```
sp_iqemptyfile ( logical-file--name )
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

sp_iqemptyfile 清空 dbfile。dbspace 必须是只读的才可以执行 **sp_iqemptyfile** 过程。该过程将文件中的对象移动到同一个数据库空间中的另一个可用的读写 dbfile。如果没有其它读写 dbfile 可用，则 Sybase IQ 将显示错误消息。

注意：在 Multiplex 环境中，只能在事务协调器上运行 **sp_iqemptyfile**。必须有一个读写数据库空间可用才能成功执行该过程。

示例

以下示例清空 dbfile **dbfile1**：

```
sp_iqemptyfile 'dbfile1'
```

sp_iqestjoin 过程

估计为您指定的表创建连接索引所需的空间大小。

语法

```
sp_iqestjoin ( table1_name, table1_row_#, table2_name,
table2_row_#, relation, iq_page_size )
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

根据所连接的表返回连接索引使用的空间大小。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）。

如果指定未经限定的表名，则确保您是连接的表的所有者。如果您不是表所有者，则为每个表提供一个限定表名，例如 “owner.tablename”。

表 129. sp_iqestjoin 的参数

名称	数据类型	描述
table1_name	char(256)	连接中第一个表的名称。
table1_row_#	int	参与连接的第一个表中的行数。
table2_name	char(256)	连接中第二个表的名称。
table2_row_#	int	参与连接的第二个表中的行数。
relation	char(9)	连接类型，可以是 “one>>many” 或 “one>>one”（单词与运算符之间不留空格）。缺省值为 “one>>many”。
iq_page_size	smallint	为数据库的 IQ 段定义的页大小（必须是 2 的乘方且介于 1024 和 524288 之间；缺省值为 131072）。

示例

```
call sp_iqestjoin ( 'Customers', 1500000, 'SalesOrders', 15000000,
'one>>many', 65536 )
```

Cases	Indexsize	Create time	Msg
Table1:Customers			
Rows: 1500000			
列:			

Cases	Indexsize	Create time	Msg
8			
Width:			
223			
Table2:SalesOrders			
Rows: 15000000			
列:			
9			
Width:			
134			
IQpagesize:			
65536			
Min Case	48001024	3h0m/CPU	
Max Case	95449088	9h6m/CPU	
Avg Case	70496256	5h53m/CPU	

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 )
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

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（否则估计值将不正确）。

表 130. sp_iquestdbspaces 的参数

名称	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 的乘方且介于 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)。

用法

sp_iquestdbspaces 将显示四种类型的建议，具体取决于数据中有多少是唯一数据：

表 131. 建议

建议	描述
min	如果数据几乎没有变化，则可以选择只创建建议大小为 min 的数据库空间段。对于变化量最小的数据，这些建议反映数据可能的最佳压缩。
avg	如果数据的变化量为平均水平，则可以创建建议大小为 min 的数据库空间段，还可以创建建议大小为 avg 的其它段。
max	如果数据的变化度较高（有很多唯一值），则可以创建建议大小为 min 、 avg 和 max 的数据库空间段。
spare	如果不确定数据中唯一值的数量，则可以创建建议大小为 min 、 avg 、 max 和 spare 的数据库空间段。装载数据后，您随时可以删除未使用的段，但是创建的段过少可能会花费一些时间。

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	

dbspace files	类型	Size	Msg
5	max	2053697536	
6	spare	1200001024	

本例估计的是一个 12GB 数据库所需的 dbspace 段的大小和数量。如果预计数据几乎不存在唯一性，Sybase IQ 建议您最少创建 3 个段（列为 **min**）以获得最佳数据压缩。如果数据的变化量为平均水平，则应多创建 1 个段（列为 **avg**）。变化量很大（有很多唯一值，需要进行大范围索引）的数据可能需要再多创建 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，可以删除任何未使用的 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。

Sybase 强烈建议您创建 spare dbspace 段，因为这样可以在稍后删除未使用的段。

- 1. 针对您预计经常进行连接的所有表对运行 **sp_iquestjoin**。
- 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_iquestspace**。
- 5. 对 **sp_iquestspace** 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_iquestdbspaces** 中的 *db_size_in_bytes* 参数。

sp_iquestspace 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_iquestspace ( table_name, #_of_rows, iq_page_size )
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

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_iquestspace** 参数。

表 132. 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）

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 ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 133. 参数

参数	描述
event-name	事件的名称。
event-owner	事件的所有者。
event-type	事件的类型。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

可以在没有任何参数的情况下调用 **sp_iqevent** 过程。如果未指定任何参数，则缺省情况下只显示有关用户事件（**dbo** 或 **SYS** 不拥有的事件）的信息。

如果未指定前两个参数中的任何一个参数，但指定了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 134. sp_iqevent 用法示例

语法	输出
sp_iqevent	显示有关数据库中所有用户事件的信息
sp_iqevent e1	显示事件 e1 的相关信息
sp_iqevent non_existing_event	不返回任何行，因为事件 non_existing_event 不存在
sp_iqevent NULL, DBA	显示 DBA 拥有的所有事件的相关信息
sp_iqevent e1, DBA	显示 DBA 拥有的事件 e1 的相关信息
sp_iqevent ev_iqbegintxn	ev_iqbegintxn 是系统定义的事件。如果没有同样名为 ev_iqbegintxn 的用户定义的事件，则不返回任何行。（缺省情况下，只返回用户定义的事件。）
sp_iqevent ev_iqbegintxn, dbo	不返回任何行，因为事件 ev_iqbegintxn 不是用户事件（缺省情况下，只返回用户事件）。
sp_iqevent NULL, NULL, SYSTEM	显示所有系统事件（由 dbo 或 SYS 拥有）的相关信息
sp_iqevent ev_iqbegintxn, NULL, SYSTEM	显示系统事件 ev_iqbegintxn 的相关信息
sp_iqevent ev_iqbegintxn, dbo, ALL	显示 dbo 拥有的系统事件 ev_iqbegintxn 的相关信息

描述

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 135. sp_iqevent 列

列名	描述
event_name	事件的名称
event_owner	事件的所有者
event_type	对于系统事件，是 SYSEVENTTYPE 系统表中列出的事件类型

列名	描述
enabled	指明是否允许触发事件 (Y/N)
action	事件处理程序定义
condition	用于控制事件处理程序触发的 WHERE 条件
位置	允许事件触发的位置: • C = 统一 • R = 远程 • A = 所有
remarks	注释字符串

示例

显示有关用户定义的事件 **e1** 的信息:

sp_iqevent e1				
event_name	event_owner	event_type	enabled	action
e1	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...
condition	location	remarks (NULL)	A	begin Declare
(NULL)	A	(NULL)		_Catalog...
				(NULL)

另请参见

- sp_iqcolumn 过程 (第 356 页)
- sp_iqconstraint 过程 (第 362 页)
- sp_iqdatatype 过程 (第 369 页)
- sp_iqhelp 过程 (第 395 页)
- sp_iqindex 和 sp_iqindex_alt 过程 (第 401 页)
- sp_iqjoinindex 过程 (第 415 页)
- sp_iqpkeys 过程 (第 431 页)
- sp_iqprocparm 过程 (第 436 页)
- sp_iq_reset_identity 过程 (第 443 页)
- sp_iqtable 过程 (第 460 页)
- sp_iqview 过程 (第 474 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 ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

sp_iqfile 显示数据库空间中每个 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。

sp_iqfile 显示以下信息：

表 136. sp_iqfile 列

列名	描述
DBSpaceName	在 CREATE DBSPACE 语句中指定的数据库空间的名称。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。
DBFileName	逻辑文件名。
Path	物理文件或裸分区的位置。
SegmentType	数据库空间的类型 (MAIN 或 TEMPORARY) 。
RWMode	数据库空间的模式：读写 (RW) 或只读 (RO)。
联机	T (联机) 或 F (脱机) 。
用法	dbspace 中该文件当前使用的 dbspace 所占的百分比。
DBFileSize	文件或裸分区的当前大小。对于裸分区，此大小值可以小于物理大小。
Reserve	dbspace 中可添加到该文件的保留空间。
StripeSize	如果磁盘条带化打开，则这是在移动到下一个文件之前写入文件的数据量。
BlkTypes	由用户数据和内部系统结构共同使用的空间。
FirstBlk	分配给文件的第一个 IQ 块号。
LastBlk	分配给文件的最后一个 IQ 块号。
OkToDrop	“Y” 表示可以删除文件；否则为 “N” 。

块类型标识符的值：

表 137. sp_iqfile 块类型

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
M	Multiplex CM*
O	旧版本
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

显示有关数据库空间中文件的信息：

```
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'

'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'
```

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 ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 138. 参数

参数	描述
obj-name	对象的名称。
obj-owner	对象的所有者。
obj-category	指定对象类别的可选参数。 列、约束和索引是与表关联的，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并非是允许值之一，则会返回 “Invalid object category” 错误。
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: 显示有关所有对象的信息 缺省情况下，只显示有关非系统对象的信息。如果指定的对象类型不是 SYSTEM 或 ALL ，则会返回 “Invalid object type” 错误。

表 139. sp_iqhelp 的 obj-category 参数值

object-type 参数	详细说明
“table”	对象是基表
“view”	对象是视图
“procedure”	对象是存储过程或函数
“event”	对象是事件
“datatype”	对象是系统数据类型或用户定义数据类型

可以在没有任何参数的情况下调用 **sp_iqhelp** 过程。如果未指定任何参数，**sp_iqhelp** 将显示有关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）的信息。

如果前三个参数均未指定，但指定了序列中的下一个参数，则必须用 **NULL** 替代省略的参数。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（值为空值时除外）。

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 140. sp_iqhelp 用法示例

语法	输出
sp_iqhelp	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 t1, u1, “table”	显示用户 u1 拥有的表 t1 以及与 t1 关联的列、索引和约束的相关信息
sp_iqhelp NULL, u1, “view”	显示用户 u1 拥有的视图 v1 以及与 v1 关联的列的相关信息
sp_iqhelp sp2	显示过程 sp2 和 sp2 的参数的相关信息
sp_iqhelp e1	显示事件 e1 的相关信息
sp_iqhelp dt1	显示数据类型 dt1 的相关信息
sp_iqhelp NULL, NULL, NULL, SYSTEM	显示所有系统对象（由 dbo 或 SYS 拥有）的相关摘要信息

语法	输出
sp_iqhelp non_existing_obj	返回错误 “Object ‘non_existing_obj’ not found”，因为对象 non_existing_obj 不存在
sp_iqhelp NULL, non_existing_user	返回错误 “User ‘non_existing_user’ not found”，因为用户 non_existing_user 不存在
sp_iqhelp t1, NULL, “apple”	返回错误 “Invalid object category ‘apple’ ”，因为 “apple” 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 “USER” 不是 <i>obj-type</i> 的允许值

描述

sp_iqhelp 存储过程可显示 IQ 数据库中系统对象及数据类型和用户定义的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、连接索引、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 141. 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
view	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
event	event_name、event_creator、enabled、location、event_type、action、external_action、condition、remarks

对象类型	显示的列
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 142. sp_iqhelp 详细信息

对象类型	描述	列
table	显示有关指定基表及其列、索引、约束和连接索引（如果该表参与任何连接索引）的信息	• 表列：table_name、table_owner、server_type、location、table_constraints、remarks • 列列：column_name、domain_name、width、scale、nulls、default、check、pkey、user_type、cardinality、est_cardinality、remarks • 索引列：index_name、column_name、index_type、unique_index、location、remarks • 约束列：constraint_name（角色）、column_name、index_name、constraint_type、foreigntable_name、foreigntable_owner、foreigncolumn_name、foreignindex_name、location • 连接索引列：joinindex_name、creator、left_table_name、left_table_owner、left_column_name、join_type、right_table_name、right_table_owner、right_column_name、key_type、valid、remarks
view	显示有关指定视图及其列的信息	• 视图列：view_name、view_creator、view_def、server_type、location、remarks • 列列：column_name、domain_name、width、scale、nulls、default、check、pkey、user_type、cardinality、est_cardinality、remarks
存储过程	显示有关指定过程及其参数的信息	• 过程列：proc_name、proc_creator、proc_defn、replicate、srvid、remarks • 参数列：parameter_name、type、width、scale、default、mode

对象类型	描述	列
函数	显示有关指定函数及其参数的信息	- 函数列: proc_name、proc_creator、proc_defn、replicate、srvid、remarks - 参数列: parameter_name、type、width、scale、default、mode
event	显示有关指定事件的信息	事件列: event_name、event_creator、enabled、location、event_type、action、external_action、condition、remarks
数据类型	显示有关指定数据类型的信息	数据类型列: type_name、creator、nulls、width、scale、default、check

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:

<code>sp_iqhelp</code>	<code>sp_customer_list</code>	<code>proc_name</code>	<code>proc_owner</code>	<code>proc_defn</code>
=====	=====	=====	<code>sp_customer_list</code>	<code>DBA create</code>
<code>procedure DBA.sp_customer_list()</code>				
<code>integer company_name char(35))</code>				
<code>begin</code>				
<code>select id company_name from Customers</code>				
<code>end replicate srvid remarks</code>				
=====	=====	=====	<code>N</code>	<code>(NULL)</code>
<code>parm_name</code>	<code>parm_type</code>	<code>parm_mode</code>	<code>domain_name</code>	<code>width scale</code>
=====	=====	=====	=====	=====
<code>id</code>	<code>result</code>	<code>out</code>	<code>integer</code>	<code>4 0</code>
<code>company_name</code>	<code>result</code>	<code>out</code>	<code>char</code>	<code>35 0 default</code>
=====	<code>(NULL)</code>			

另请参见

- `sp_iqcolumn` 过程 (第 356 页)
- `sp_iqconstraint` 过程 (第 362 页)
- `sp_iqdatatype` 过程 (第 369 页)
- `sp_iqevent` 过程 (第 390 页)
- `sp_iqindex` 和 `sp_iqindex_alt` 过程 (第 401 页)
- `sp_iqjoinindex` 过程 (第 415 页)
- `sp_iqpkeys` 过程 (第 431 页)
- `sp_iqprocparm` 过程 (第 436 页)
- `sp_iq_reset_identity` 过程 (第 443 页)
- `sp_iqtable` 过程 (第 460 页)
- `sp_iqview` 过程 (第 474 页)

sp_iqhelp 与 Adaptive Server Enterprise 的兼容性

Sybase IQ `sp_iqhelp` 存储过程与 Adaptive Server Enterprises `sp_help` 过程类似，后者显示有关 `SYSOBJECTS` 系统表中列出的所有数据库对象的信息，以及有关系统数据类型和用户定义数据类型的信息。

在支持的对象类型和对象命名空间方面，Sybase IQ 与 Adaptive Server Enterprise 在体系结构方面存在一些差异。在 Adaptive Server Enterprise 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中。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）则存储在不同的系统表中，并且位于不同的命名空间中。例如，在 Sybase IQ 中，一个表可以与某个事件或某个存储过程具有相同名称。

由于 Sybase IQ 与 Adaptive Server Enterprise 在体系结构方面存在差异，Sybase IQ `sp_iqhelp` 支持的对象类型和语法不同于 Adaptive Server Enterprises `sp_help` 支持的对象和语法；但是，这两个存储过程显示的有关数据库对象的信息类型是类似的。

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' ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 143. 参数

参数	描述
语法 1	如果未指定前两个参数中的任何一个参数，但指定了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	您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。

描述

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 144. 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
index_type	索引类型的缩写形式（例如 HG 、 LF ）
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 COMMENT 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 356 页）
- sp_iqconstraint 过程（第 362 页）
- sp_iqdatatype 过程（第 369 页）
- sp_iqevent 过程（第 390 页）
- sp_iqhelp 过程（第 395 页）
- sp_iqjoinindex 过程（第 415 页）
- sp_iqpkeys 过程（第 431 页）
- sp_iqprocparm 过程（第 436 页）
- sp_iq_reset_identity 过程（第 443 页）
- sp_iqtable 过程（第 460 页）
- sp_iqview 过程（第 474 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。

以下两种语法变化形式均返回名为 **DepartmentID** 的列中的所有索引：

```
call sp_iqindex (NULL, 'DepartmentID')
sp_iqindex column_name='DepartmentID'
```

table_name	table_owner	column_name	index_type	index_name	unique_index	location	dbspace_id	remarks
Departments	GROUPO	DepartmentID	FP	ASIQ_IDX_T201_C1_FP	N	Main	16387	(NULL)
Departments	GROUPO	DepartmentID	HG	ASIQ_IDX_T201_C1_HG	U	Main	16387	(NULL)
Employees	GROUPO	DepartmentID	FP	ASIQ_IDX_T202_C5_FP	N	Main	16387	(NULL)

以下两种语法变化形式均返回表所有者 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	index_name	unique_index	location	dbspace_id	remarks
Departments	GROUPO	Department-HeadID	FP	ASIQ_IDX_T201_C3_FP	N	Main	16387	(NULL)
Departments	GROUPO	DepartmentID	FP	ASIQ_IDX_T201_C1_FP	N	Main	16387	(NULL)
Departments	GROUPO	DepartmentID	HG	ASIQ_IDX_T201_C1_HG	U	Main	16387	(NULL)
Departments	GROUPO	Department-Name	FP	ASIQ_IDX_T201_C2_FP	N	Main	16387	(NULL)

sp_iqindex_alt 语法的以下两种变化形式均返回包含列 **City** 的表 **Employees** 中的所有索引。索引 **emp_loc** 是列 **City** 和 **State** 的多列索引。对于多列索引，**sp_iqindex_alt** 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	index_name	unique_index	dbspace_id	remarks
Employees	GROUPO	City	FP	ASIQ_IDX_T452_C7_FP	N	16387	(NULL)
Employees	GROUPO	City	HG	emp_loc	N	16387	(NULL)
Employees	GROUPO	状态	HG	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	index_name	unique_index	dbspace_id	位置	remarks
Employees	GROUPO	City	FP	ASIQ_IDX_T452_C7_FP	N	16387	Main	(NULL)
Employees	GROUPO	City,State	HG	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 ] )
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 **EXECUTE** 权限才能运行存储过程。

用法

表 145. 参数

参数	描述
resetflag	允许调用者清除索引建议存储。如果 <i>resetflag</i> 不为零，则系统会在检索完最后一行后删除所有建议。

描述

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 146. 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 147. 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 358 页)
- sp_iqindexuse 过程 (第 413 页)
- sp_iqtableuse 过程 (第 465 页)
- sp_iqunusedcolumn 过程 (第 469 页)
- sp_iqunusedindex 过程 (第 470 页)

- `sp_iqunusedtable` 过程（第 472 页）
- `sp_iqworkmon` 过程（第 479 页）

sp_iqindexfragmentation 过程

报告有关 Sybase IQ 索引中的 B 树、GArray 和位图结构内 占用的页空间百分比的信息。

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 [...]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 148. 参数

参数	描述
table-name	目标 table <i>table-name</i> 用于报告指定表中的所有非缺省索引。
index-name	目标 index <i>index-name</i> 用于报告指定索引。每个 <i>index-name</i> 都是限定索引名。可以指定表中的多个索引，但必须使用指定的每个索引重复 index 关键字。

示例

报告表 `Customers` 中非唯一 **HG** 索引 `cidhg` 的内部索引碎片：

```
dbo.sp_iqindexfragmentation ( 'index customers.cidhg ' )
```

指数	索引类型	B 树节点页数	GARRAY_FILL_FACTOR_PERCENT
dba.customers.cidhg	HG	3	75
SQLCODE	0		
填充百分比	B 树页数	GArray 页数	位图页数
0 - 10%	0	0	0
11 - 20%	0	0	0
21 - 30%	0	0	0

31-40%	0	0	22
41 - 50%	0	0	0
51 - 60%	0	0	10
61 - 70%	2	0	120
71 - 80%	138	3	64
81 - 90%	24	122	14
91 - 100%	18	1	0

根据此输出，在非唯一 **HG** 索引 **cidhg** 中的 182 个 B 树页中，填充百分比介于 61% 和 70% 之间的有 2 个，填充百分比为 71% 至 80% 的有 138 个，填充百分比为 81% - 90% 的有 24 个，填充百分比为 91% - 100% 的有 18 个。GArray 和位图页使用情况的报告方式与此相同。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。这些类型为 **HG**、**LF**、**WD**、**DATE** 和 **DTTM**。

如果在针对此索引执行存储过程的过程中出错，SQLCODE 将为非零值。

另请参见

- `sp_iqindexmetadata` 过程（第 409 页）
- `sp_iqindexinfo` 过程（第 407 页）
- `sp_iqindexsize` 过程（第 410 页）
- `sp_iqrebuildindex` 过程（第 439 页）
- `sp_iqrowdensity` 过程（第 445 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数据库空间中，则 **sp_iqindexinfo** 返回在所有数据库空间中使用的空间，如下例所示。

语法

```
sp_iqindexinfo '{ database
| [ table table-name | index index-name ] [...] }
[ resources resource-percent ]'
```

权限

需要具有 **DBA** 权限。必须向没有 **DBA** 权限的用户授予 **EXECUTE** 权限才能运行存储过程。

用法

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或索引参数。如果已指定表名，**sp_iqindexinfo** 会返回表中所有索引的信息。如果已指定索引名，则仅返回该索引的信息。

不能按名称指定连接索引。请使用 **database** 关键字来显示连接索引。

如果指定的 *table-name* 或 *index-name* 不明确或无法找到该对象，则返回一个错误。

缺省情况下，在 **Multiplex** 数据库中，**sp_iqindexinfo** 显示辅助节点上有关共享的 IQ 存储的信息。如果指定了单个表或索引，则会自动选中要显示的存储。

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。

可以在 **Multiplex** 环境中使用此过程。请参见《使用 Sybase IQ Multiplex》。

描述

sp_iqindexinfo 会向 DBA 显示给定对象驻留在哪些数据库空间上。DBA 可用该信息来确定哪些数据库空间必须具有 **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 149. sp_iqindexinfo 列

列名	描述
对象	表、索引或连接索引的名称
Dbospace_name	dbospace 的名称
ObjSize	此 dbospace 中此对象的数据大小
DBSpPct	该对象所使用的 dbospace 的百分比
MinBlk	由此 dbospace 中的此对象使用的第一个块
MaxBlk	由此 dbospace 中的此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 dbospace 调整为较小的大小

示例

显示有关表 **t2** 的索引信息：

```
sp_iqindexinfo 'table t2';
```

对象	dbospace_name	ObjSize	DBSpPct	MinBlk	MaxBlk
t2	IQ_SYSTEM_MAIN	32K	1	84	107

对象	dbspace_name	ObjSize	DBSpPct	MinBlk	MaxBlk
t2	dbspacedb2	160K	2	1045495	1045556
t2	dbspacedb3	8K	1	2090930	2090930
t2.DBA.ASIQ_IDX_T430_C1_FP	IQ_SYSTEM_MAIN	136K	2	126	321
t2.DBA.ASIQ_IDX_T430_C1_FP	dbspacedb3	152K	2	2091032	2091053
t2.DBA.t2c1hng	dbspacedb2	136K	2	1045537	1045553

另请参见

- sp_iqdbspace 过程（第 373 页）
- sp_iqdbspaceinfo 过程（第 376 页）
- sp_iqspaceinfo 过程（第 448 页）
- sp_iqindexmetadata 过程（第 409 页）
- sp_iqindexfragmentation 过程（第 406 页）
- sp_iqindexsize 过程（第 410 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' ] ] }
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

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。

描述

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 150. sp_iqindexmetadata 输出行

索引类型	返回的元数据
CMP, DATE, DTTM, TIME	Type, Version
FP	Type, Version, LookupPages, Style, LookupEntries, 1stLookupPage, LargeLOBs, SmallLOBs, IQ Unique, LOB Compression (仅限于列数据类型为 LONG VARCHAR 或 LONG BINARY 的情况)
HG	Type, Version, Distinct Keys
HNG	Type, Version, BitsPerBlockmap, NumberOfBits
LD	Type, Version<Id>, Version, Distinct Keys
LF	Type, Version, IndexStatus, NumberOfBlockmaps, BitsPerBlockmap, Distinct Keys
WD	Type, Version, KeySize, Delimiters, DelimiterCount, MaxKeyWordLength, PermitEmptyWord

示例

以下命令显示了有关 **HG** 索引 hg_index_col54 的索引信息:

```
sp_iqindexmetadata 'hg_index_col54' , 'metal' , 'DBA';
```

'DBA',	'metal'	'hg_index_col54'
'Type',	'HG',	"
'Version',	'2',	"
'Distinct Keys',	'0',	"

另请参见

- sp_iqindexfragmentation 过程 (第 406 页)
- sp_iqindexinfo 过程 (第 407 页)
- sp_iqindexsize 过程 (第 410 页)

sp_iqindexsize 过程

给出指定索引的大小。

语法

```
sp_iqindexsize [ [ owner.] table.] index_name
```

权限
需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

表 151. 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。
类型	索引类型。
信息	报告其 KBytes、Pages 和 Compressed Pages 的 IQ 索引的组件。组件因索引类型而异。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_T452_C19_FP

Username	Indexname	类型	信息	KBytes	Pages	Compressed Pages
DBA	Employees.ASIQ_IDX_T452_C19_FP	FP	总	288	4	2
DBA	Employees.ASIQ_IDX_T452_C19_FP	FP	vdo	0	0	0

Username	Indexname	类型	信息	KBytes	Pages	Compressed Pages
DBA	Employees.ASIQ_IDX_T452_C19_FP	FP	bt	0	0	0
DBA	Employees.ASIQ_IDX_T452_C19_FP	FP	garray	0	0	0
DBA	Employees.ASIQ_IDX_T452_C19_FP	FP	bm	136	2	1
DBA	Employees.ASIQ_IDX_T452_C19_FP	FP	barray	152	2	1
DBA	Employees.ASIQ_IDX_T452_C19_FP	FP	dpstore	0	0	0
DBA	Employees.ASIQ_IDX_T452_C19_FP	FP	largelob	0	0	0
DBA	Employees.ASIQ_IDX_T452_C19_FP	FP	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
GROUPO	GROUPO.Employees.ti	TEXT	总	896	12	6
GROUPO	GROUPO.Employees.ti	TEXT	vdo	0	0	0
GROUPO	GROUPO.Employees.ti	TEXT	bt	304	4	2
GROUPO	GROUPO.Employees.ti	TEXT	garray	152	2	1
GROUPO	GROUPO.Employees.ti	TEXT	bm	136	2	1
GROUPO	GROUPO.Employees.ti	TEXT	barray	152	2	1

Username	Indexname	类型	信息	KBytes	Pages	Compressed Pages
GROUPO	GROUPO.Employees.ti	TEXT	dpstore	0	0	0
GROUPO	GROUPO.Employees.ti	TEXT	largelob	0	0	0
GROUPO	GROUPO.Employees.ti	TEXT	txtPst	304	4	2

另请参见

- `sp_iqindexmetadata` 过程 (第 409 页)
- `sp_iqindexfragmentation` 过程 (第 406 页)
- `sp_iqindexinfo` 过程 (第 407 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`

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

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。

系统不会报告在 **SYSTEM** 中创建的表的索引。

表 152. `sp_iqindexuse` 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
UID**	索引唯一标识符

列名	描述
类型	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	NOpt
NQry NConstraint						
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 358 页）
- sp_iqindexadvice 过程（第 404 页）
- sp_iqtableuse 过程（第 465 页）
- sp_iqunusedcolumn 过程（第 469 页）
- sp_iqunusedindex 过程（第 470 页）
- sp_iqunusedtable 过程（第 472 页）
- sp_iqworkmon 过程（第 479 页）

sp_iqjoinindex 过程

显示与连接索引有关的信息。

语法

```
sp_iqjoinindex [ left-table-name ], [ left-column-name ], [ left-table-owner ], [ right-table-name ], [ right-column-name ], [ right-table-owner ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

可以在没有任何参数的情况下调用 **sp_iqjoinindex** 过程。如果未指定参数，**sp_iqjoinindex** 会显示 IQ 基表上所有连接索引的信息。连接索引表始终都是 IQ 基表。连接索引表不能是临时表、远程表或代理表。

表 153. 参数

参数	描述
left-table-name	构成连接操作左侧的表的名称。
left-column-name	构成连接左侧某部分的列的名称。
left-table-owner	构成连接操作左侧的表的所有者。
right-table-name	构成连接操作右侧的表的名称。
right-column-name	构成连接右侧某部分的列的名称。
right-table-owner	构成连接操作右侧的表的所有者。

如果未指定前五个参数中的任何一个参数，但指定了序列中的下一个参数，则必须使用 NULL 替代省略的参数。例如，**sp_iqjoinindex NULL, NULL, NULL, t2, n2, DB'** 和 **sp_iqjoinindex t1, NULL, NULL, t2**。

表 154. sp_iqjoinindex 用法示例

语法	输出
sp_iqjoinindex	显示与所有连接索引有关的信息
sp_iqjoinindex t1, NULL, DBA	显示 DBA 所拥有的 t1 构成操作左侧的所有连接索引的信息

语法	输出
sp_iqjoinindex t2, n1, DBA	显示以 DBA 所拥有的表 t2 的列 n1 作为连接左侧的连接索引信息
sp_iqjoinindex NULL, NULL, DBA, NULL, NULL, DBA	显示其左侧和右侧表皆为 DBA 所有的连接索引的信息
sp_iqjoinindex NULL, NULL, NULL, t2, NULL, NULL	显示表 t2 位于连接操作右侧的所有连接索引的信息
sp_iqjoinindex t1, n1, DBA, t2, n1, DBA	显示符合以下条件的连接索引的信息：左侧为 DBA 所拥有的表 t1 的列 n1，右侧为 DBA 所拥有的表 t2 的列 n1
sp_iqjoinindex non_existing_table	不返回任何行，因为表 non_existing_table 不存在
sp_iqjoinindex NULL, NULL, non_existing_user	不返回任何行，因为用户 non_existing_user 不存在

描述

sp_iqjoinindex 存储过程显示与数据库中连接索引有关的信息。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 *left-table-name*，**sp_iqjoinindex** 就会显示由该表构成连接左侧的所有连接索引。如果指定 *left-table-owner*，**sp_iqjoinindex** 将仅返回左侧表属于指定所有者的连接索引。如果未指定任何参数，**sp_iqjoinindex** 将显示有关数据库中所有连接索引的信息。

sp_iqjoinindex 过程返回以下列中的信息：

表 155. sp_iqjoinindex 列

列名	描述
joinindex_name	连接索引的名称。
creator	连接索引的所有者。
left_table_name	构成连接操作左侧的表的名称。
left_table_owner	构成连接操作左侧的表的所有者名称。
left_column_name	构成连接左侧某部分的列的名称。
join_type	当前支持的值只有 “=”。
right_table_name	构成连接操作右侧的表的名称。
right_table_owner	构成连接操作右侧的表的所有者名称。
right_column_name	构成连接右侧某部分的列的名称。

列名	描述
key_type	定义键上的连接类型： - NATURAL：自然连接 - KEY：键连接 - ON：左外/右外/完全连接
有效的	指示是否需要同步此连接索引。“Y”表示不需要同步，“N”表示需要同步。
remarks	注释字符串。
dbspace_id	指定连接索引所驻留的 dbspace 的名称。

示例

显示表 t1 构成连接操作左侧的连接索引的信息：

```
sp_iqjoinindex t1
joindex_name creator left_table_name left_table_owner left_column_
name
join_type right_table_name right_table_owner right_column_name key_t
ype valid dbspace_id remarks
t1_t2_t3_join DBA t1 DBA n1
= t2 DBA n1 NATURAL
Y 16387 (NULL)
```

显示表 t2 构成连接操作左侧的连接索引的信息：

```
sp_iqjoinindex t2
joindex_name creator left_table_name left_table_owner left_column_
name
join_type right_table_name right_table_owner right_column_name key_t
ype valid dbspace_id remarks
t1_t2_t3_join DBA t2 DBA n1
= t3 DBA n1 NATURAL
Y (NULL) t1_t2_t3_join DBA t2 DBA name
= t3 DBA name NATURAL
Y 16387 ((NULL))
```

显示符合以下条件的连接索引的信息：左侧为 DBA 所拥有的表 t2 的列 name，右侧为 DBA 所拥有的表 t3 的列 name：

```
sp_iqjoinindex t2, name, DBA, t3, name, DBA
joindex_name creator left_table_name left_table_owner left_column_
name
join_type right_table_name right_table_owner right_column_name key_t
ype valid dbspace_id remarks
t1_t2_t3_join DBA t2 DBA name
= t3 DBA name NATURAL
Y 16387 ((NULL))
```

另请参见

- sp_iqcolumn 过程 (第 356 页)
- sp_iqconstraint 过程 (第 362 页)
- sp_iqdatatype 过程 (第 369 页)
- sp_iqevent 过程 (第 390 页)
- sp_iqhelp 过程 (第 395 页)
- sp_iqindex 和 sp_iqindex_alt 过程 (第 401 页)
- sp_iqpkeys 过程 (第 431 页)
- sp_iqprocparm 过程 (第 436 页)
- sp_iq_reset_identity 过程 (第 443 页)
- sp_iqtable 过程 (第 460 页)
- sp_iqview 过程 (第 474 页)

sp_iqjoinindexsize 过程

给出指定连接索引的大小。

语法

sp_iqjoinindexsize (join_index_name)

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

以字节、千字节和 NBlocks (IQ 块) 为单位返回索引的总大小。还返回将连接索引保存在内存中所需的页数，以及将连接索引在磁盘上压缩后的 IQ 页数。对于此过程，必须指定 join_index_name 参数。

表 156. sp_iqjoinindexsize 列

列名	描述
Username	连接索引的所有者
JoinIndexName	返回其结果的连接索引
Number of Tables	连接索引中的表数
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
NBlocks	IQ 块数

示例

```
sp_iqjoinindexsize ( 't1t2' )
```

Username	JoinIndexName	Number of Tables	KBytes	Pages	Compressed Pages	NBlocks
DBA	t1t2	2	13	15	4	26

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 157. “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 158. “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 159. 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 160. 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 161. 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 |
```

权限

需要 DBA 授权。

用法

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 111 页）

sp_iqlocks 过程

显示与数据库中 IQ 存储和 Catalog 存储中的锁有关的信息。

语法

```
sp_iqlocks ([connection,] [[owner.]table_name] max_locks,]  
[sort_order])
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

您可以指定来限制结果的可选 **sp_iqlocks** 参数：

表 162. 可选 sp_iqlocks 参数

参数	数据类型	描述
<i>connection</i>	整数	连接 ID。使用该选项时，该过程仅返回与指定连接的锁有关的信息。缺省值为零，使用该值将返回与所有连接有关的信息。
<i>owner.table_name</i>	char(128)	表名。使用该选项，该过程仅返回与指定表的锁有关的信息。缺省值为空值，使用该值将返回与数据库中的所有表有关的信息。如果未指定 owner，系统则假定该过程的调用者拥有该表。
<i>max_locks</i>	整数	返回其信息的锁的最大数目。缺省值为 0，它返回所有锁的信息。
<i>sort_order</i>	char(1)	返回信息的顺序： • C 表示按连接排序（缺省值） • T 表示按 table_name 排序

描述

显示与数据库中当前锁有关的信息。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：

表 163. sp_iqlocks 列

列	数据类型	描述
conn_name	VARCHAR(128)	当前连接的名称。
conn_id	INTEGER	拥有锁的连接 ID。
user_id	CHAR(128)	与该连接 ID 相关联的用户。
table_type	CHAR(6)	表的类型。该类型是 BASE（对于表）、GLBTMP（对于全局临时表）或 MVIEW（对于物化视图）。 仅 IQ Catalog 存储中的 SQL Anywhere 表支持 物化视图。
creator	VARCHAR(128)	表的所有者。
table_name	VARCHAR(128)	持有锁的表。
index_id	INTEGER	索引 ID 或空值。
lock_class	CHAR(8)	指示锁类型的字符串： • S - 共享。 • SW - 共享和写入。 • EW - 排它和写入。 • E - 排它。 • P - 幻像。 • A - 反幻像。 • W - 写入。 所有列出的锁必须有 S、E、EW 或 SW 中的一个，还可能有 P 和/或 A。幻像和反幻像锁还有一个限定符 T 或 *： • T - 该锁针对有序扫描。 • * - 该锁针对所有扫描。 • <i>nnn</i> - 索引号；该锁针对特定索引。 Sybase IQ 先获取共享锁，再获取写入锁。如果连接具有排它锁，则不会显示共享锁。对于写入锁，如果连接同时具有排它、共享和写入三种锁，则会显示 EW。
lock_duration	CHAR(11)	锁持续时间。Transaction、Position 或 Connection 之一。
lock_type	CHAR(9)	标识锁的值（取决于锁类）。
row_identifier	UNSIGNED BIGINT	行或空值的标识符。

如果 **sp_iqlocks** 无法找到拥有表锁的用户的连接 ID 或用户名，它就会以 0（零）来表示该连接 ID 并以 **User unavailable** 来表示用户名。

注意：排它、幻像或反幻像锁可以用于 SQL Anywhere 表，但不能用于 Sybase IQ 表。除非已在 Catalog 存储中显式撤消了表上的锁，否则您将永远无法在 Sybase IQ 数据库中看到这些类型的锁（或其限定符 T、* 及 *nnn*）。有关锁定在 SQL Anywhere 表中的工作方式的信息，请参见“SQL Anywhere Server – SQL 用法”。

示例

此示例显示了在 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 全是缺省选项，因此其输出按连接排序 显示了所有锁。

```
call sp_iqlocks()
```

conn_name	conn_id	user_id	table_type	creator	table_name
con1	70187172	'mary'	BASE	DBA	t1

index_id	lock_class	lock_duration	lock_type	row_iden
tifier				
ASIQ_IDX_T452_C19_FP	Table	Position	Table	1

sp_iqmodifyadmin 过程

将命名登录策略的某个选项设置为特定值。如果未指定任何登录策略，则设置根策略的选项。在 Multiplex 中，**sp_iqmodifyadmin** 采用 Multiplex 服务器名的可选参数。

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' ,
```

用法

表 164. 参数

参数	描述
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	其登录策略选项需要更改的登录策略的名称。

权限

需要具有 DBA 权限。

示例

将名为 *lockeduser* 的策略的登录选项 *locked* 设置为 ON:

```
call sp_iqmodifyadmin ('locked','on','lockeduser')
```

在名为 *Writer1* 的 Multiplex 服务器上将名为 *lockeduser* 的策略的登录选项 *locked* 设置为 ON:

```
call sp_iqmodifyadmin ('locked','on','lockeduser','Writer1')
```

另请参见

- *sp_expireallpasswords* 过程 (第 338 页)
- *sp_iqaddlogin* 过程 (第 338 页)
- *sp_iqcopyloginpolicy* 过程 (第 366 页)
- *sp_iqmodifylogin* 过程 (第 424 页)
- *sp_iqpassword* 过程 (第 430 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']
```

权限

需要具有 DBA 权限。

用法

表 165. 参数

参数	描述
userid	存放待修改的帐户名的变量。
login_policy_name	(可选) 指定将向用户分配的登录策略的名称。如果未指定任何登录策略名称，则向用户分配 root 登录策略。

示例

将用户 joe 分配给名为 expired_password 的登录策略:

```
sp_iqmodifylogin 'joe', 'expired_password'
```

将用户 joe 分配给 root 登录策略:

```
call sp_iqmodifylogin ('joe')
```

另请参见

- sp_expireallpasswords 过程 (第 338 页)
- sp_iqaddlogin 过程 (第 338 页)
- sp_iqcopyloginpolicy 过程 (第 366 页)
- sp_iqmodifyadmin 过程 (第 423 页)
- sp_iqpassword 过程 (第 430 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 失败，请运行 **sp_iqmpxcheckdqpconfig** 以确定 DQP 配置问题是否导致查询分配失败。

语法

```
sp_iqmpxcheckdqpconfig
```

权限

执行该过程不需要特权。

描述

表 166. 列说明

列名	描述
DiagMsgID	唯一标识诊断消息

列名	描述
描述	描述在 DQP 配置中发现的问题的诊断消息

表 167. 诊断消息

DiagMsgID	描述
0	在 DQP 配置中没有发现问题
1	数据库为 Simplex
2	Multiplex 正在单节点配置模式下运行
3	登录策略选项 dqp_enabled 设置为 OFF
4	临时 dqp_enabled 连接选项设置为 OFF
5	逻辑服务器上下文仅有一个成员节点
6	由于其在逻辑服务器中的指定成员资格当前无效，因此协调器不参与 DQP
7	由于其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，原因是 Root 逻辑服务器策略中的 ALLOW_COORDINATOR_AS_MEMBER 选项设置为 OFF
8	IQ_SHARED_TEMP dbspace 中没有 dbfile
9	IQ_SHARED_TEMP dbspace 中的所有 dbfile 均为只读文件
10	IQ_SHARED_TEMP dbspace 动态脱机

示例

sp_iqmpxcheckdqpconfig 过程的输出样本：

diagmsgid	description
3	Login policy option dqp_enabled is set to OFF
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 Logical server policy set to OFF
8	There is no dbfile in IQ_SHARED_TEMP dbspace

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。

共享的 dbfile 包括 **IQ_SYSTEM_MAIN**、**IQ_SHARED_TEMP** 和用户 dbspace 中的所有文件。在辅助节点上执行时，仅显示当前节点的文件状态。用户必须获得 **Multiplex Grid Option** 使用许可才能运行辅助节点。有关包括语法在内的完整信息，请参见《使用 Sybase IQ Multiplex》。

sp_iqmpxincconnpoolinfo 过程

如果在协调节点上运行，显示每一节点的 **INC** 连接归集状态。如果在辅助节点上执行，仅显示当前节点的 **INC** 连接归集状态。

用户必须获得 **Multiplex Grid Option** 使用许可才能运行 辅助节点。有关 **sp_iqmpxincconnpoolinfo** 语法 和完整说明，请参见《使用 Sybase IQ Multiplex》。

sp_iqmpxincheartbeatinfo 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。

用户必须获得 **Multiplex Grid Option** 使用许可才能运行 辅助节点。有关 **sp_iqmpxincheartbeatinfo** 语法 和完整说明，请参见《使用 Sybase IQ Multiplex》。

sp_iqmpxinfo 过程

如果在协调节点上运行，显示所有节点的完整 **Multiplex** 配置信息。如果在辅助节点上运行，仅显示该节点的完整 **Multiplex** 配置信息。

用户必须获得 **Multiplex Grid Option** 使用许可才能运行 辅助节点。有关 **sp_iqmpxinfo** 语法 和完整说明，请参见《使用 Sybase IQ Multiplex》。

sp_iqmpxvalidate 过程

检查 **Multiplex** 配置中是否有不一致。

有关 **sp_iqmpxvalidate** 语法 和完整说明，请参见《使用 Sybase IQ Multiplex》。

sp_iqmpxversioninfo 过程

显示此服务器的当前版本信息。

有关 **sp_iqmpxversioninfo** 语法 和完整说明，请参见《使用 Sybase IQ Multiplex》。

sp_iqobjectinfo 过程

返回数据库对象和子对象的分区和 dbspace 赋值。

语法

```
sp_iqobjectinfo [ owner_name ] [ , object_name ] [ , object-type ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 168. 参数

参数	描述
owner_name	对象的所有者。如果指定，则 sp_iqobjectinfo 仅显示使用指定所有者的表和连接索引的输出。如果未指定，则 sp_iqobjectinfo 显示数据库中所有用户的表和连接索引的相关信息。
object_name	表或连接索引的名称。如果未指定，则 sp_iqobjectinfo 显示有关数据库中所有表和连接索引的信息。
object-type	有效的对象类型为 table （缺省值）或 joinindex 。 如果 object-type 是表，则它必须用引号引起来。

所有参数都是可选的，且任一参数均可独立于其它参数的值提供。

Sybase 建议您将输入参数与 **sp_iqobjectinfo** 一起使用；如果使用输入参数，而不是在查询的 **WHERE** 子句中使用谓词，则可以查询 **sp_iqobjectinfo** 的结果，并且效果更好。例如，查询 A 编写如下：

```
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 为重新编写的查询 A 以使用 **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 返回结果的速度快于查询 A。当输入参数传递给 **sp_iqobjectinfo** 时，过程将比较和连接系统表中较少的记录，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并会返回较小的结果集，这样查询中使用的谓词数较少。

sp_iqobjectinfo 存储过程支持解释 *owner_name*、*object_name* 和 *object_type* 的通配符。它按与 **LIKE** 子句匹配查询内部的模式相同的方式，显示匹配给定模式的所有数据库空间的信息。

描述

返回特定或所有数据库对象（仅限于表和连接索引类型的对象）及其子对象的所有分区和 dbspace 赋值。子对象为列、索引、主键、唯一约束和外键。

表 169. sp_iqobjectinfo 列

列名	描述
owner	对象所有者的名称。
object_name	位于 dbspace 上的对象（仅限于表和连接索引类型的对象）的名称。
sub_object_name	位于 dbspace 上的对象的名称。
object_type	对象的类型（列、索引、主键、唯一约束、外键、分区、连接索引或表）。
object_id	对象的全局对象 ID。
id	对象的表 ID 或连接索引 ID。
dbspace_name	对象所在的 dbspace 的名称。将为分区对象的特殊元行显示字符串 “[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： 以下两个示例演示 **iqdemo** 数据库中的对象，以便更好地描述输出。请注意，**iqdemo** 包括一个名为 **iq_main** 的用户数据库空间示例，您自己的数据库中可能不存在该示例。

显示有关特定用户拥有的特定数据库对象和子对象的分区和数据库空间赋值的信息：

sp_iqobjectinfo GROUP0, Departments				
owner	object_name	sub_object_name	object_type	obj
ect_id	id			
GROUP0	Departments	(NULL)	table	3
632	738			
GROUP0	Departments	DepartmentID	column	3
633	738			
GROUP0	Departments	DepartmentName	column	3
634	738			
GROUP0	Departments	DepartmentHeadID	column	3
635	738			
GROUP0	Departments	DepartmentsKey	primary	
key 83	738			
GROUP0	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)
```

显示有关 *object-type*table 的特定用户拥有的特定数据库对象和子对象的分区和数据库空间赋值的信息：

```
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)
```

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' ]
```

权限

设置自己的口令不需要任何权限；设置其他用户的口令则需要 DBA 或 PERMS ADMIN 权限。

用法

表 170. 参数

参数	描述
caller_password	口令。在您更改自己的口令时，该口令是您的旧口令。当 DBA 或 PERMS ADMIN 更改另一用户的口令时，caller_password 是 DBA 或 PERMS ADMIN 口令。
new_password	用户的新口令，或 loginname 的新口令。
user_name	DBA 或 PERMS ADMIN 要更改其口令的用户的登录名。更改自己的口令时，请勿指定 user_name。

描述

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。DBA 或 PERMS ADMIN 权限可以更改任何现有用户的口令。

创建用户的首选方式是使用 **CREATE USER** 语句。

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'
```

仅当已登录用户具有 DBA 或 PERMS ADMIN 特权或用户是 joe 本人时，才会将用户 joe 的口令从 epr45 更改为 pdi032：

```
call sp_iqpassword ('epr45', 'pdi932', 'joe')
```

另请参见

- sp_expireallpasswords 过程（第 338 页）
- sp_iqaddlogin 过程（第 338 页）
- sp_iqcopyloginpolicy 过程（第 366 页）
- sp_iqmodifyadmin 过程（第 423 页）
- sp_iqmodifylogin 过程（第 424 页）

sp_iqpkeys 过程

按表、列、表所有者或为数据库中 的所有 Sybase IQ 表显示与主键和主键约束 有关的信息。

语法

```
sp_iqpkeys { [ table-name ], [ column-name ], [ table-owner ] }
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 171. 参数

参数	描述
table-name	基表或全局临时表的名称。如果指定，该过程将仅返回与在指定表上定义的主键有关的信息。
column-name	列的名称。如果指定，该过程将仅返回与指定列上的主键有关的信息。
table-owner	表的所有者或表。如果指定，该过程将仅返回属于指定所有者的表上的主键的相关信息。

可指定一个或多个参数。如果未指定前两个参数中的任何一个参数，但指定了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。如果指定的某个参数无效，则输出中不会显示任何行。

表 172. sp_iqkeys 用法示例

语法	输出
sp_iqkeys sales	显示表 sales 上定义的主键的相关信息
sp_iqkeys sales, NULL, DBA	显示 DBA 拥有的表 sales 上定义的主键的相关信息
sp_iqkeys sales, store_id, DBA	显示 DBA 拥有的表 sales 的列 store_id 上定义的主键的相关信息
sp_iqkeys NULL, NULL, DBA	显示 DBA 拥有的所有表上定义的主键的相关信息

描述

sp_iqkeys 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 173. sp_iqkeys 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

列名	描述
constraint_name	主键约束的名称
constraint_id	主键约束 ID

注意： `sp_iqpkkeys` 存储过程仅位于 使用 Sybase IQ 版本 12.6 或 更高版本创建的数据 库中。

示例

显示在表 `sales1` 的列上定义的主键：

```
sp_iqpkkeys sales1
table_name table_owner column_name column_id constraint_name constra
int_id
sales1          DBA          store_id      1          MA114          114
```

显示在表 `sales2` 的列上定义的主键：

```
sp_iqpkkeys sales2
table_name table_owner column_name column_id constraint_name constra
int_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_iqpkkeys sales2, store_id
table_name table_owner column_name column_id constraint_name constra
int_id
sales2          DBA          store_id      1          MA115          115
```

另请参见

- `sp_iqcolumn` 过程 (第 356 页)
- `sp_iqconstraint` 过程 (第 362 页)
- `sp_iqdatatype` 过程 (第 369 页)
- `sp_iqevent` 过程 (第 390 页)
- `sp_iqhelp` 过程 (第 395 页)
- `sp_iqindex` 和 `sp_iqindex_alt` 过程 (第 401 页)
- `sp_iqjoinindex` 过程 (第 415 页)
- `sp_iqprocparm` 过程 (第 436 页)
- `sp_iq_reset_identity` 过程 (第 443 页)
- `sp_iqtable` 过程 (第 460 页)
- `sp_iqview` 过程 (第 474 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 ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 174. 参数

参数	描述
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

可以在没有任何参数的情况下调用 **sp_iqprocedure** 过程。如果未指定任何参数，则缺省情况下只显示有关用户定义的过程（dbo 或 SYS 不拥有的过程）的信息。

如果未指定前两个参数中的任何一个参数，但指定了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 175. sp_iqprocedure 用法示例

语法	输出
sp_iqprocedure	显示 dbo 或 SYS 不拥有的数据库中的所有过程的相关信息
sp_iqprocedure sp_test	显示有关过程 sp_test 的信息
sp_iqprocedure non_existing_proc	不返回任何行，因为过程 non_existing_proc 不存在
sp_iqprocedure NULL, DBA	显示 DBA 拥有的所有过程的相关信息

语法	输出
sp_iqprocedure sp_test, DBA	显示 DBA 拥有的过程 sp_test 的相关信息
sp_iqprocedure sp_iqtable	过程 sp_iqtable 不是系统过程。如果没有同样名为 sp_iqtable 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 sp_iqtable 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_iqprocedure sp_iqtable, NULL, 'YSTEM	显示有关系统过程 sp_iqtable 的信息
sp_iqprocedure sp_iqtable, dbo, ALL	显示有关由 dbo 所拥有的系统过程 sp_iqtable 的信息

描述

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 176. 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

显示有关用户定义的过程 **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
```

显示有关由用户 DBA 所拥有的所有过程的信息：

```
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
```

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 ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 177. 参数

参数	描述
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

可以不带参数调用 **sp_iqprocparm**。如果未指定任何参数，则显示用户定义的过程（dbo 或 SYS 不拥有的过程）的输入/输出和结果参数。

如果未指定前两个参数中的任何一个参数，但指定了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 178. sp_iqprocparm 用法示例

语法	输出
sp_iqprocparm	显示 dbo 或 SYS 不拥有的数据库中的所有过程的参数
sp_iqprocparm sp_test	显示有关过程 sp_test 的信息
sp_iqprocparm non_existing_proc	不返回任何行，因为过程 non_existing_proc 不存在
sp_iqprocparm NULL, DBA	显示由 DBA 所拥有的所有过程的参数
sp_iqprocparm sp_test, DBA	显示由 DBA 所拥有的过程 sp_test 的参数
sp_iqprocparm sp_iqtable	sp_iqtable 是一个系统过程。如果没有同样名为 sp_iqtable 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_iqprocparm sp_iqtable, dbo	不返回任何行，因为过程 sp_iqtable 不是用户过程。（缺省情况下，仅返回用户过程。）
sp_iqprocparm NULL, NULL, SYSTEM	显示所有系统过程（由 dbo 或 SYS 拥有）的参数
sp_iqprocparm sp_iqtable, NULL, SYSTEM	显示系统过程 sp_iqtable 的参数
sp_iqprocparm sp_iqtable, dbo, ALL	显示由 dbo 所拥有的系统过程 sp_iqtable 的参数

描述

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 179. 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

列名	描述
parm_type	参数类型为下列值之一： - 普通参数（变量） - 结果变量：与返回结果集的过程一起使用 - SQLSTATE 错误值 - SQLCODE 错误值
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： - in：参数为过程提供一个值 - out：参数返回一个值 - inout：参数既提供一个值又返回一个值 - 空值：参数既不提供值也不返回值
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
标度	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

显示有关用户定义过程 `sp_test` 的参数信息：

```
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 356 页）

- `sp_iqconstraint` 过程（第 362 页）
- `sp_iqdatatype` 过程（第 369 页）
- `sp_iqevent` 过程（第 390 页）
- `sp_iqhelp` 过程（第 395 页）
- `sp_iqindex` 和 `sp_iqindex_alt` 过程（第 401 页）
- `sp_iqjoinindex` 过程（第 415 页）
- `sp_iqpkeys` 过程（第 431 页）
- `sp_iq_reset_identity` 过程（第 443 页）
- `sp_iqtable` 过程（第 460 页）
- `sp_iqview` 过程（第 474 页）

`sp_iqrebuildindex` 过程

使用 **CREATE TABLE** 语句中指定的原始 **IQ UNIQUE** 值重建表的一个或多个索引，或使用新的 **IQ UNIQUE** 值来更改所需的存储空间和/或查询性能。

若要重建缺省索引以外的索引，请指定该索引名。

语法

`sp_iqrebuildindex (table_name, index_clause)`

权限

要运行此过程，必须具有 **EXECUTE** 权限。必须拥有表的 **INSERT** 权限，才能对该表重建索引。

用法

表 180. 参数

参数	描述
table_name	要对其重建索引的表的部分限定名或完全限定名。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	一个或多个下列字符串，以空格进行分隔： column <i>column_name</i> [<i>count</i>] index <i>index_name</i> 每个 <i>column_name</i> 或 <i>index_name</i> 必须引用指定表中的某列或某个索引。如果多次指定 <i>column_name</i> 或 <i>index_name</i>，该过程就会返回一条错误，并且不重建任何索引。 <i>count</i> 是一个非负值，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的值的估计数量。不同的值的数量会影响查询速度和存储要求。

必须指定关键字 **column** 和 **index**。这些关键字是不区分大小写的。

注意： 此过程不支持 **TEXT** 索引。要重建 **TEXT** 索引，必须删除并重新创建该索引。

描述

如果指定列名，该过程不需任何索引名即可重建该列的缺省索引。在这种情况下，如果指定列名的同时还指定了由 Sybase IQ 分配的缺省索引的名称，系统会返回一条错误。如果省略了 *column_name* 后面的 *count*，则系统会使用值 0（零）作为缺省值。

如果缺省索引是单字节索引，**sp_iqrebuildindex** 始终会将缺省索引重建为单字节索引，无论用户指定的 **IQ UNIQUE** 值是多少。

对于单字节缺省索引，如果 *column_name (count)* 中指定的值为 0 或大于 256，则系统会使用列的基数值来更新 **SYS.SYSIQCOLUMN** 中的 **approx_unique_count** 列。

sp_iqrebuildindex 将对数据类型为 **LONG VARCHAR (CLOB)** 的列重建 **WD** 索引。

如果缺省索引为双字节，而指定的 *count* 为 0 或大于 65536 的值，列的基数值将会决定是否将缺省索引重建为单字节或双字节索引，并用该值来更新 **SYS.SYSIQCOLUMN** 中的 **approx_unique_count** 列。

如果指定的是一个非零的 **IQ UNIQUE** 值，缺省索引就会重建为单字节、双字节或平面缺省索引，不过前文所述情况除外。

如果指定 **IQ UNIQUE** 的值为零或不指定 **IQ UNIQUE** 的值，**MINIMIZE_STORAGE** 选项会控制如何重建索引：

- 如果 **MINIMIZE_STORAGE** 选项设置为 **ON**，索引就会先重建为单字节缺省索引，然后再根据需要转换为双字节或平面索引。
- 如果 **MINIMIZE_STORAGE** 设置为 **OFF**，则系统会使用该数据类型的缺省值来重建索引。

示例

重建列 *Surname* 的缺省索引：

```
sp_iqrebuildindex 'emp1', 'column dept_id'
```

或：

```
call sp_iqrebuildindex ('emp1', 'column dept_id')
```

创建列 **c1** 的平面缺省索引：

```
CREATE TABLE mytable (c1 int IQ UNIQUE 1000000000)
```

将缺省单字节索引转换为双字节索引：

```
sp_iqrebuildindex 'mytable', 'column c1 1024'
```

或：

```
call sp_iqrebuildindex ('mytable', 'column c1 1024')
```


注意： 当 `sp_iqrebuildindex` 在大型 HG 索引上运行时，用户可能会发现性能暂时降低。

另请参见

- `sp_iqindexfragmentation` 过程（第 406 页）
- `sp_iqrowdensity` 过程（第 445 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 ]
```

权限

必须是表的所有者，或者具有 DBA 授权或对象的 ALTER 权限。要求对对象具有独占访问权限。

用法

表 181. 参数

参数	描述
object-name	用户创建的对象原始名称。 或者， <i>owner-name</i> 可以指定为 <i>object-name</i> 的一部分，即 <i>owner-name.object-name</i> ，其中 <i>owner-name</i> 是要重命名的对象的所有者名称。如果未指定 <i>owner-name</i> ，调用 <code>sp_iqrename</code> 的用户就会被假定为对象的所有者。只有调用 <code>sp_iqrename</code> 的用户拥有重命名该对象所需的权限时，才能成功重命名该对象。 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， <i>object-name</i> 的形式可以是 <i>table-name.object-name</i> 或 <i>owner-name.table-name.object-name</i> 。
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是唯一的。
object-type	可选参数，用于指定要重命名的用户所创建对象的类型，亦即对象 <i>object-name</i> 的类型。 <i>object-type</i> 参数可用大写或小写字母进行指定。

object-type 参数的值：

表 182. sp_iqrename object-type 参数值

object-type 参数	详细说明
column	要重命名的对象是列
index	要重命名的对象是索引
constraint	要重命名的对象是一个唯一约束、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！ 您必须相应地更改与要通过 **sp_iqrename** 重命名的对象相关的对象（过程、函数以及视图）的定义。**sp_iqrename** 过程不会自动更新依赖对象的定义。您必须手动更改这些定义。

描述

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。

如果尝试用对于某类型的对象并不唯一的名称来重命名此对象，则 **sp_iqrename** 将返回消息“项已存在”。

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“功能不受支持”。

您也可以使用 **ALTER TABLE** 语句和 **ALTER INDEX** 语句的 **RENAME** 子句来进行重命名。请参见《参考：语句和选项》。

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** 上的索引 **idindex** 重命名为 **isbnindex**：

```
sp_iqrename books.idindex, isbnindex, index
```

将表 **books** 上的主键约束 **prim_id** 重命名为 **prim_isbn**：

```
sp_iqrename books.prim_id, prim_isbn, constraint
```

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)
```

用法

必须指定 *table_name*、*table owner* 和 *value*。

权限

不需要。

描述

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 356 页)
- `sp_iqconstraint` 过程 (第 362 页)
- `sp_iqdatatype` 过程 (第 369 页)
- `sp_iqevent` 过程 (第 390 页)
- `sp_iqhelp` 过程 (第 395 页)
- `sp_iqindex` 和 `sp_iqindex_alt` 过程 (第 401 页)
- `sp_iqjoinindex` 过程 (第 415 页)
- `sp_iqpkeys` 过程 (第 431 页)
- `sp_iqprocparm` 过程 (第 436 页)
- `sp_iqtable` 过程 (第 460 页)
- `sp_iqview` 过程 (第 474 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 ]
```

用法

表 183. 参数

参数	描述
timestamp	指定过去的日期目标。

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

如果不能使数据库达到时间戳的一致状态，sp_iqrestoreaction 将返回错误。否则，建议恢复将使数据库回到一致状态的操作。

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 184. sp_iqrestoreaction 列

列名	描述
sequence_number	要执行的步骤的顺序
backup_id	备份事务的标识符
backup_archive_list	备份中存档文件的列表
backup_time	备份执行的时间
virtual_type	虚拟备份的类型：“Non-virtual”、“Decoupled”或“Encapsulated”

列名	描述
restore_dbSPACE	可以为空。表示要从备份存档中恢复的所有 dbSPACE
restore_dbfile	可以为空。表示要从备份存档中恢复的给定 dbSPACE 中的所有 dbfile
backup_comment	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	c:\\\\temp\\\\b1	2008-09-23
14:48:05.01	2	c:\\\\temp\\\\b2.inc	2008-09-23
14:48:13.0	3	c:\\\\temp\\\\b3.inc	2008-09-23

virtual_type	restore_dbSPACE	restore_dbfile	backup_comment
Nonvirtual			
Nonvirtual			
Nonvirtual			

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 (...))
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

参数	描述
table-name	目标表 <i>table-name</i> 用于报告指定表中的所有列。
column-name	目标列 <i>column-name</i> 用于报告目标表中的所有指定列。可指定多个目标列, 但每次都必须重复使用相应关键字。

必须指定关键字 **table** 和 **column**。这些关键字是不区分大小写的。

描述

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。该过程以 $0 < density < 1$ 的数值形式返回密度。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。

示例

报告表 *SalesOrders* 中列 *ID* 的行密度：

```
sp_iqrowdensity('column group0.SalesOrders.ID')
```

Tablename	ColumnName	IndexType	Density
'GROUPO.SalesOrders'	'ID'	'Flat style FP'	'1.0'

另请参见

- sp_iqindexfragmentation 过程（第 406 页）
- sp_iqrebuildindex 过程（第 439 页）

sp_iqsharedtempdistrib 过程

返回有关共享临时空间在 Multiplex 的节点之间是如何分布的以及该空间是可用的还是已隔离的诊断信息。

用户必须获得 Multiplex Grid Option 使用许可才能运行辅助节点。有关 **sp_iqsharedtempdistrib** 语法和完整说明，请参见《使用 Sybase IQ Multiplex》。

sp_iqshowpsexex 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_iqshowpsexex [ connection-id ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 185. 参数

参数	描述
connection-id	表示连接 ID 的整数。 如果指定 <i>connection-id</i> , 则 sp_iqshowpsex e 将仅返回指定连接的相关信息。 如果未指定 <i>connection-id</i> , 则 sp_iqshowpsex e 将返回所有连接的相关信息。 如果指定的 <i>connection-id</i> 不存在, 则 sp_iqshowpsex e 不返回任何行。

描述

sp_iqshowpsexe 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 186. sp_iqshowpsex 列

列名	描述
connectionid	连接 ID
应用程序	打开连接的客户端应用程序的相关信息。包括 AppInfo 连接属性信息: HOST : 客户端计算机的主机名; EXE : 客户端可执行文件的名称 (仅限 Windows); APPINFO : 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特别长的查询。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), 用于约束用户查询使用的临时 IQ 数据库空间。缺省值为 2000MB。
max_cursors	数据库选项 MAX_CURSOR_COUNT 的值, 用于指定资源管理器来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。

列名	描述
max_statements	数据库选项 MAX_STATEMENT_COUNT 的值，用于指定资源管理器来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。

有关列中引用的数据库选项的信息，请参见《参考：语句和选项》。

注意： **AppInfo** 属性在 Open Client 或 jConnect 应用程序（例如 Interactive SQL 或 Sybase Central）中可能不可用。如果 **AppInfo** 属性不可用，则 **application** 列为空。

示例

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 ID 为 2 的连接的资源使用情况：

```
sp_iqshowpsexec 2 connectionid application
                  2 HOST=G00DGUY-XP;EXE=C:\\Program Files\\Sybase\\
                  IQ-15_3\\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 133 页）
- sp_iqcontext 过程（第 364 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] [...] ']
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

针对当前数据库显示对象名、每个对象使用的块数和数据库空间的名称。

sp_iqspaceinfo 不需要任何参数。

sp_iqspaceinfo 返回的信息对于管理数据库空间很有帮助。

可以在 Multiplex 环境中使用此过程。请参见《使用 Sybase IQ Multiplex》。

示例

下面的输出来自 **iqdemo** 数据库上运行的 **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 407 页）
- **sp_iqdbspace** 过程（第 373 页）
- **sp_iqdbspaceinfo** 过程（第 376 页）

sp_iqspaceused 过程

显示 IQ 存储、IQ 临时存储以及 IQ 全局和本地共享临时存储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)
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 **EXECUTE** 权限才能运行存储过程。

用法

sp_iqspaceused 以 **unsigned bigint out** 参数返回八个值。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 **IQ main** 存储空间和 **IQ** 临时存储空间的数量。

描述

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。

表 187. sp_iqspaceused 列

列名	描述
mainKB	IQ main 存储空间的总大小（以 KB 为单位）。
mainKBUsed	数据库使用的 IQ main 存储 空间的大小（以 KB 为单位）。（辅助 Multiplex 节点返回 '(Null)'。）
tempKB	IQ 临时存储空间的总大小（以 KB 为单位）。
tempKBUsed	IQ 临时存储空间的总大小（以 KB 为单位）。
shTempTotalKB	IQ 全局共享临时存储空间的总大小（以 KB 为单位）。
shTempTotalKBUsed	IQ 全局共享临时存储空间的总大小（以KB为单位）。（辅助 Multiplex 节点返回 '(Null)'。）
shTempLocalKB	IQ 本地共享临时存储空间的总大小（以 KB 为单位）。
shTempLocalKBUsed	数据库使用的 IQ 本地共享临时存储空间的大小（以 KB 为单位）。

示例

sp_iqspaceused 需要八个输出参数。下面的示例显示用户定义的存储过程 **myspace** 的创建过程，该存储过程先声明这八个输出参数，然后调用 **sp_iqspaceused**：

```
create 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; call sp_iqspaceused(mt,mu,tt,tu,gt,gu,lt,lu); select
cast(mt/1024 as unsigned bigint) as mainMB, cast(mu/1024 as
unsigned bigint) as mainusedMB, mu*100/mt as
mainPerCent, cast(tt/1024 as unsigned bigint) as
tempMB, cast(tu/1024 as unsigned bigint) as
tempusedMB, tu*100/tt as tempPerCent; cast(gt/1024 as
unsigned bigint) as shTempTotalKB, cast(gu/1024 as unsigned
bigint) as shTempTotalKBUsed, gu*100/gt as
globalshtempPerCent; cast(lt/1024 as unsigned bigint) as
shTempLocalMB, cast(lu/1024 as unsigned bigint) as
shTempLocalKBUsed, lu*100/lt as localshtempPerCent; end
```

若要显示 **sp_iqspaceused** 的输出，请运行过程 **myspace**：

```
myspace
```

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 ]
```

用法

参数	描述
stat_name	(可选) VARCHAR 参数指定统计的名称。

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

如果提供了 **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。不带任何参数进行调用时，**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 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 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 IQ 服务器使用的 CPU 系统时间（以秒为单位）	Second

stat_num	stat_name	stat_desc	stat_unit
3	ThreadsFree	可用 IQ 线程数	不适用
4	ThreadsInUse	正在使用的 IQ 线程数	不适用
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	main 高速缓存当前大小（以 MB 为单位）	MB
8	MainCacheFinds	main 高速缓存查找请求总数	不适用
9	MainCacheHits	main 高速缓存命中总数	不适用
10	MainCachePagesPinned	main 高速缓存已固定页数	页码
11	MainCachePagesPinnedPercentage	main 高速缓存已固定页的百分比	%
12	MainCachePagesDirtyPercentage	main 高速缓存脏页的百分比	%
13	MainCachePagesInUsePercentage	main 高速缓存正在使用的页的百分比	%
14	TempCacheCurrentSize	临时高速缓存当前大小（以 MB 为单位）	MB
15	TempCacheFinds	临时高速缓存查找请求总数	不适用
16	TempCacheHits	临时高速缓存命中总数	不适用
17	TempCachePagesPinned	临时高速缓存已固定页数	页码
18	TempCachePagesPinnedPercentage	临时高速缓存已固定页的百分比	%
19	TempCachePagesDirtyPercentage	临时高速缓存脏页的百分比	%
20	TempCachePagesInUsePercentage	临时高速缓存正在使用的页的百分比	%
21	MainStoreDiskReads	从 main 存储读取的字节数（以 KB 为单位）	KB
22	MainStoreDiskWrites	写入 main 存储的字节数（以 KB 为单位）	KB

stat_num	stat_name	stat_desc	stat_unit
23	TempStoreDiskReads	从 main 存储读取的字节数 (以 KB 为单位)	KB
24	TempStoreDiskWrites	写入 main 存储的字节数 (以 KB 为单位)	KB
25	ConnectionsTotalConnections	自服务器启动以来的连接总数	不适用
26	ConnectionsTotalDisconnections	自服务器启动以来的断开连接总数	不适用
27	ConnectionsActive	活动连接数	不适用
28	OperationsWaiting	等待 IQ 资源调控器的操作数	不适用
29	OperationsActive	IQ 资源调控器接纳的活动并发操作数	不适用
30	OperationsActiveLoadTableStatements	活动 LOAD TABLE 语句数	不适用

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%'
```

sp_iqstatus 过程

显示当前数据库的各种 相关 Sybase IQ 状态信息。

语法

```
sp_iqstatus
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

显示当前数据库的相关状态信息, 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、备份信息等等。

sp_iqstatus 显示 main 和临时存储的空间不足状态。如果存储区发生空间不足状况, **sp_iqstatus** 会在该存储区的空间不足状态显示值中显示 Y。

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。

若要显示可通过删除连接回收的空间，请使用 **sp_iqstatus** 并将返回的两行中的结果相加：

```
(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 不显示将在下一个检查点释放的块。但是，这些块作为类型 X 出现在 **sp_iqdbspace** 输出中。

可以在 Multiplex 环境中使用此过程。请参见《使用 Sybase IQ Multiplex》。

示例

注意： 以下示例演示 **iqdemo** 数据库中的对象，以便更好地描述输出。请注意，**iqdemo** 包括一个名为 **iq_main** 的用户数据库空间示例，您自己的数据库中可能不存在该示例。

下面的输出来自 **sp_iqstatus** 存储过程：

```
Sybase IQ (TM)                Copyright (c) 1992-2010 by Sybase, Inc.
                                All rights reserved.
Version:                        15.3.0/090416/P/MS/Windows/2010/
                                32bit/2010-04-16 02:11:41
Time Now:                       2010-04-21 13:48:22.319
Build Time:                     2010-04-16 02:15:39
File Format:                    23 on 03/18/1999
Server mode:                    IQ Server
Catalog Format:                  2
Stored Procedure Revision:      1
Page Size:                      131072/8192blks/16bpp
Number of Main DB Files :       2
Main Store Out Of Space:        N
Number of Local Temp DB Files : 1
Local Temp Store Out Of Space:  N
DB Blocks: 1-3200               IQ_SYSTEM_MAIN
DB Blocks: 1045440-1055039      iq_main
Local Temp Blocks: 1-1600       IQ_SYSTEM_TEMP
Create Time:                    2009-04-03 11:30:20.674
Update Time:                    2009-04-03 11:34:33.040
Main IQ Buffers:                255, 32Mb
Temporary IQ Buffers:           191, 24Mb
```

```

Main IQ Blocks Used:                    5915 of 11200, 52%=46Mb, Max
Block#:105278
Local Temporary IQ Blocks Used:        65 of 800, 8%=0Mb, Max
Block#: 0
Main Reserved Blocks Available:        1600 of 1600, 100%=6Mb
Local Temporary Reserved Blocks Available: 6400 of 6400,
100%=50Mb
IQ Dynamic Memory:                     Current: 69mb, Max: 70mb
Main IQ Buffers:                       Used: 17, Locked: 0
Temporary IQ Buffers:                 Used: 4, Locked: 0
Main IQ I/O:                           I: L1581/P14 O: C3/D163/P161
D:34 C:97.1
Temporary IQ I/O:                     I: L6627/P0 O: C1086/D1166/P83
D:1082 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:                           1
Blocks in next ISF Backup:             0 Blocks: =0Mb
Blocks in next ISI Backup:             0 Blocks: =0Mb
DB File Encryption Status:             OFF

```

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: 逻辑页读取 (“查找”)
- P: 物理页读取
- O: 输出
- C: 创建的页
- D: 弄脏的页
- P: 物理写入
- D: 损坏的页
- C: 压缩比

另请参见

- sp_iqtransaction 过程 (第 466 页)
- sp_iqversionuse 过程 (第 473 页)

sp_iqsysmon 过程

监控 Sybase IQ 的多个组件, 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
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

批处理模式用法

参数	描述
start_monitor	开始监控。
stop_monitor	停止监控并显示报告。
time-period	监控的时间段。必须采用 HH:MM:SS 格式。
section(s)	要由 sp_iqsysmon 显示的一个部分或多个部分的缩写。如果指定的部分多于一个，则必须用空格分隔各部分的缩写，并且列表必须用单引号或双引号引起来。缺省情况下显示所有部分。 对于与 IQ 存储有关的部分，可以分别通过在部分缩写之前添加前缀 “m” 或 “t” 来指定主存储或临时存储。如果没有前缀，则会同时对两个存储进行监控。例如，如果指定 “mbufman”，则仅监控 IQ 主存储缓冲区管理器。如果指定 “mbufman tbufman” 或 “bufman”，则会同时对主存储缓冲区管理器和临时存储缓冲区管理器进行监控。

表 188. sp_iqsysmon 报告部分缩写

报告部分或 IQ 组件	缩写
缓冲区管理器	(m/t)bufman
缓冲池	(m/t)bufpool
预取管理	(m/t)prefetch
可用空间列表管理	(m/t)freelist
缓冲区分配	(m/t)bufalloc
内存管理	内存
线程管理	线程
CPU 利用率	cpu

报告部分或 IQ 组件	缩写
事务管理	txn
服务器上下文统计信息	服务器
目录统计信息	Catalog

注意： `sp_iqsysmon` 当前不支持 Sybase IQ 组件磁盘 I/O 和锁管理程序。

文件模式用法

表 189. 参数

参数	描述
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 <code>sp_iqsysmon</code> 在文件模式下运行。在文件模式下，为监控周期中的每一间隔显示统计信息样本。缺省情况下，输出写入到名为 <code>dbname.connid-iqmon</code> 的日志文件中。使用 <code>file_suffix</code> 选项可更改输出文件的后缀。有关 <code>file_suffix</code> 选项的说明，请参见 <code>monitor_options</code> 参数。
monitor_options	<code>monitor_options</code> 字符串

`monitor_options` 字符串可以包含一个或多个选项：

表 190. `monitor_options` 字符串选项

<code>monitor_options</code> 字符串选项	描述
<code>-intervalseconds</code>	指定报告间隔（以秒为单位）。在经过每一时间间隔之后将监控器统计信息的输出样本到日志文件中。如果未指定 <code>-interval</code> 选项，则缺省设置为每 60 秒输出一次。最小报告时间间隔为 2 秒。如果为该选项指定的间隔无效或小于 2 秒，则系统会将间隔设置为 2 秒。 初次显示时显示自服务器启动以来的计数器信息。后续显示则显示与前一显示不同的地方。在运行有性能问题的查询期间或在出现性能问题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 秒通常就足够了。

monitor_options 字符串选项	描述
-file_suffix suffix	创建名为 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 iqmon。如果指定 -file_suffix 选项，但未提供后缀或提供空字符串作为后缀，则不使用后缀。
-append or -truncate	指示 sp_iqsysmon 是添加到现有输出文件还是截断现有文件。Truncate 是缺省值。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。
-sectionsection(s)	指定要写入监控器日志文件的一个或多个部分的缩写。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。如果指定的部分多于一个，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。忽略无效部分缩写，并在 IQ 消息文件中显示警告。

用法语法示例

表 191. sp_iqsysmon 用法示例

语法	结果
sp_iqsysmon start_monitor sp_iqsysmon stop_monitor	在批处理模式下启动监控器，并显示 main 存储和临时存储的所有部分。
sp_iqsysmon start_monitor sp_iqsysmon stop_monitor “mbufman mbufpool”	在批处理模式下启动监控器，并显示 main 存储的缓冲区管理器和缓冲池统计信息。
sp_iqsysmon “00:00:10” , “mbufpool memory”	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 start_monitor, ‘filemode’ , “-interval 5 -sections mbufpool memory” sp_iqsysmon stop_monitor	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每 5 秒写入日志文件一次。

描述

sp_iqsysmon 存储过程监控 Sybase IQ 的多个组件，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

在批处理模式下，**sp_iqsysmon** 收集监控器启动到停止这个时间段内的监控器统计信息，或者收集 *time-period* 参数指定的时间段内的统计信息。监控周期结束时，**sp_iqsysmon** 显示合并统计信息的列表。

sp_iqsysmon 在批处理模式下 类似于 Adaptive Server Enterprise 过程 **sp_sysmon**。

- 文件模式

在文件模式下，**sp_iqsysmon** 将监控器启动和停止的每个时间间隔的样本统计信息写入日志文件。

在文件模式下初次显示时，系统将显示自服务器启动以来的计数器信息。后续显示则显示与前一显示不同的地方。

文件模式下的 **sp_iqsysmon** 类似于 **IQ UTILITIES** 命令 **START MONITOR** 和 **STOP MONITOR** 接口。

批处理模式示例

在 10 分钟后输出监控信息：

```
sp_iqsysmon "00:10:00"
```

在 5 分钟后仅输出 **sp_iqsysmon** 报告的内存管理器部分：

```
sp_iqsysmon "00:05:00", memory
```

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
```

在 20 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"
```

文件模式示例

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
```

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

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"
===== Buffer Pool (Main)
===== STATS-
NAME          TOTAL  NONE  BTREEV  BTREEF  BV  VDO  DBEXT  DBID  SORT
MovedToMRU           0      0      0      0      0      0      0      0      0
MovedToWash          0      0      0      0      0      0      0      0      0
```

RemovedFromLRU	0	0	0	0	0	0	0	0	0	0
RemovedFromWash	0	0	0	0	0	0	0	0	0	0
RemovedInScanMode	0	0	0	0	0	0	0	0	0	0
STORE	GARRAY	BARRAY	BLKMAP	HASH	CKPT	BM	TEST	CMID	RIDCA	LOB
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
STATS-NAME	VALUE			Pages						127 (
100.0 %) InUse				4		(3.1 %)				
Dirty				1		(0.8 %)				
Pinned				0		(0.0 %)				
Flushes				0		FlushedBufferCount				0
GetPageFrame				0		GetPageFrameFailure				0
GotEmptyFrame				0		Washed				0
TimesSweepersWoken				0						
washTeamSize	0			WashMaxSize		26 (20.5				
%) washNBuffers				4		(3.1 %)				
washNDirtyBuffers				1		(0.8				
%)										
	washSignalThreshold			3		(2.4 %)				
washNActiveSweepers	0			washIntensity		1				
=====						Memory Manager				
=====						STATS-NAME				VALUE
MemAllocated	43616536			(42594 KB)						
MemAllocatedMax	43735080			(42710 KB)						
MemAllocatedEver	0			(0 KB)						
MemNAllocated	67079			MemNAllocatedEver		0				
MemNTimesLocked	0			MemNTimesWaited						
0	(0.0 %)									

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 ] )
```

语法 2

```
sp_iqtable [table_name='tablename'],  
[table_owner='tableowner' ],[table_type='tabletype' ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法：语法 1

如果未指定前两个参数中的任何一个参数，但指定了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。

用法：语法 2

参数可以任何顺序进行指定。将其用单引号引起来。

允许的 table_type 参数值：

表 192. sp_iqtable 的 table_type 值

table_type 值	显示的信息
SYSTEM	系统表
TEMP	全局临时表
VIEW	视图
ALL	IQ 表、系统表及视图
其它任何值	IQ 表

描述

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。如果不指定参数，则将返回数据库中的所有 Sybase IQ 表。无法返回本地临时表的名称。

表 193. sp_iqtable 列

列名	描述
table_name	表的名称。
table_type	BASE - 基表。 MAT VIEW - 物化视图。（仅 SA 表） GBL TEMP - 全局临时表。 PARTITION - 表分区（该表仅供内部使用，不能被 Sybase IQ 用户使用）。 VIEW - 视图。 JVT - 连接虚拟表。
table_owner	表的所有者

列名	描述
server_type	IQ - 在 IQ 存储中创建的对象。 SA - 在 SA 存储中创建的对象。 所有视图都是在 SA 存储中创建的。
位置	TEMP - IQ 临时存储。 MAIN - IQ 存储。 SYSTEM - 目录存储。
dbspace_id	表所驻留的 dbspace 的名称。
isPartitioned	如果列属于分区表并有一个或多个分区的 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 COMMENT 语句添加的用户注释。
table_constraints	对表的约束。

另请参见

- sp_iqcolumn 过程（第 356 页）
- sp_iqconstraint 过程（第 362 页）
- sp_iqdatatype 过程（第 369 页）
- sp_iqevent 过程（第 390 页）
- sp_iqhelp 过程（第 395 页）
- sp_iqindex 和 sp_iqindex_alt 过程（第 401 页）
- sp_iqjoinindex 过程（第 415 页）
- sp_iqpkeys 过程（第 431 页）
- sp_iqprocparm 过程（第 436 页）
- sp_iq_reset_identity 过程（第 443 页）
- sp_iqview 过程（第 474 页）

sp_iqtable 过程示例

使用示例作为 **sp_iqtable** 用法的参考。

以下两种不同形式的语法都会返回表 **Departments** 的相关信息：

```
sp_iqtable ('Departments')
```

```
sp_iqtable table_name='Departments'
```

Table_name	Table_type	Table_owner	Server_type	位置
Departments	BASE	GROUPO	IQ	Main

dbspace_id	isPartitioned	Remarks	table_constraints
16387	N	(NULL)	(NULL)

以下两种不同形式的语法都会返回表所有者 GROUPO 拥有的所有表：

```
sp_iqtable NULL,GROUPO
```

```
sp_iqtable table_owner='GROUPO'
```

Table_name	Table_type	Table_owner	Server_type	位置
Contacts	BASE	GROUPO	IQ	Main
Customers	BASE	GROUPO	IQ	Main
Departments	BASE	GROUPO	IQ	Main
Employees	BASE	GROUPO	IQ	Main
FinancialCodes	BASE	GROUPO	IQ	Main
FinancialData	BASE	GROUPO	IQ	Main
Products	BASE	GROUPO	IQ	Main
SalesOrders	BASE	GROUPO	IQ	Main
SalesOrderItems	BASE	GROUPO	IQ	Main

dbspace_id	isPartitioned	Remarks	table_constraints
16387	N	(NULL)	(NULL)
16387	N	(NULL)	(NULL)
16387	N	(NULL)	(NULL)
16387	N	(NULL)	(NULL)
16387	N	(NULL)	(NULL)
16387	N	(NULL)	(NULL)
16387	N	(NULL)	(NULL)
16387	N	(NULL)	(NULL)

dbspace_id	isPartitioned	Remarks	table_constraints
16387	N	(NULL)	(NULL)

sp_iqtablesize 过程

返回指定表的大小。

语法

```
sp_iqtablesize ( table_owner.table_name )
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

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。

表 194. sp_iqtablesize 列

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 (以 KB 为单位)
Pages	将表保存在内存中所需的 IQ 页数
CompressedPages	当压缩表时 (在磁盘上) 压缩的 IQ 页数
NBlocks	IQ 块数

Pages 是表的总 IQ 页数。页的测量单位是 IQ 页大小。所有内存内缓冲区 (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。

如果用 KBytes 值除以页大小，就会得到磁盘上页的平均大小。

注意：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 中读取和写入 88K。一般页 按因数 2 到 3 进行压缩。

NBlocks 是 Kbytes 除以 IQ 块大小后得到的数值。

CompressedPages 是压缩的 页数。例如，如果 Pages 是 1000 且 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%，因为大部分页都压缩。空白页不会压缩，因为 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.emp1')
```

Ownername	Tablename	Columns	KBytes	Pages	CompressedPages	NBlocks
DBA	emp1	4	1504	24	14	188

sp_iqtableuse 过程

报告负载访问的表的详细使用信息。

语法

```
sp_iqtableuse
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

不会报告在 SYSTEM 中创建的表。

表 195. 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 358 页）
- sp_iqindexadvice 过程 （第 404 页）

- sp_iqindexuse 过程（第 413 页）
- sp_iqunusedcolumn 过程（第 469 页）
- sp_iqunusedindex 过程（第 470 页）
- sp_iqunusedtable 过程（第 472 页）
- sp_iqworkmon 过程（第 479 页）

sp_iqtransaction 过程

显示事务和版本的相关信息。

语法

sp_iqtransaction

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

sp_iqtransaction 为 Sybase IQ 事务管理器中 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 输出中不包含未启动事务的连接的行。若要查看所有连接，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写入表的用户，但 **sp_iqlocks** 是实现此目的的更好选择。

可以在 Multiplex 环境中使用此过程。请参见《使用 Sybase IQ Multiplex》。

表 196. sp_iqtransaction 列

列名	描述
名称	应用程序的名称。
Userid	连接的用户 ID。
TxnID	该事务控制块的事务 ID。在开始事务时分配事务 ID。该事务 ID 与 .iqmsg 文件中的 BeginTxn、CmtTxn 和 PostCmtTxn 消息以及打开数据库时记录的 Txn ID Seq 中显示的事务 ID 相同。
CmtID	提交事务时事务管理器分配的 ID。对于活动事务则为零。

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。它由 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。该变量反映内部 Sybase IQ 实现的详细信息，将来可能会发生更改。在编写本书时，事务状态有 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 指示事务已完成并且正在等待 APPLIED，此时，对任何事务都不可见的版本将进行碎片收集。 一旦事务状态变为 ROLLED_BACK、COMMITTED 或 APPLIED，该事务将不再拥有打开的游标所持有的锁以外的任何锁。
ConnHandle	连接的 ID 号。
IQConnID	作为 .iqmsg 文件中所有消息的一部分显示的十位数连接 ID。这是服务器会话内唯一单调增加的整数。
MainTableKBCr	该事务创建的 IQ 存储区空间的大小（以 KB 为单位）。
MainTableKBDr	该事务删除但继续存在于磁盘上存储区中的 IQ 存储区空间（因为相应空间在该事务的其它数据库版本或其它保存点中可见）的大小（以 KB 为单位）。
TempTableKBCr	该事务创建的用于存储 IQ 临时表数据的 IQ 临时存储区空间的大小（以 KB 为单位）。
TempTableKBDr	该事务删除但继续存在于磁盘上 IQ 临时存储区中的 IQ 临时表空间（因为相应空间可由 IQ 游标看到或由该事务的其它保存点所有）的大小（以 KB 为单位）。
TempWorkSpaceKB	对于 ACTIVE 事务，这是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 sp_iqtransaction 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 sp_iqtransaction，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。 对于 ACTIVE 事务，该列与 sp_iqconnection 的 TempWorkSpaceKB 列相同。

列名	描述
TxnCreateTime	事务的开始时间。所有 Sybase IQ 事务 都在活动连接建立后马上隐式开始或在前一事务提交 或回退时隐式开始。
CursorCount	引用该事务控制块的 打开的 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，因此可能反映不出用户创建的保存点。
MPXServerName	该值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	该值指示与当前事务关联的全局事务 ID。如果没有相关的全局事务，则为零。

示例

下面是 `sp_iqtransaction` 输出的示例：

Name	Userid	TxnID	CmtID	VersionID	State	ConnHandle	IQConnID
=====	=====	=====	=====	=====	=====	=====	=====
red2	DBA	10058	10700	10058	COMMITTED	419740283	14
blue1	DBA	10568	0	10568	ACTIVE	640038605	17
	DBA	10604	0	10604	ACTIVE	2094200996	18
fromSCJ	DBA	10619	0	10619	ACTIVE	954498130	20
blue2	DBA	10634	10677	10634	COMMITTED	167015670	21
ntJava2	DBA	10676	0	10676	ACTIVE	1779741471	24
blue2	DBA	10678	0	10678	ACTIVE	167015670	21
nt1	DBA	10699	0	10699	ACTIVE	710225777	28
red2	DBA	10701	0	10701	ACTIVE	419740283	14
	DBA	16687	0	16687	ACTIVE	1306718536	23
MainTableKBCr	MainTableKBDr		TempTableKBCr		TempTableKBDr		
=====	=====		=====		=====		
0	0		65824		0		
0	0		0		0		
0	0		0		0		
0	0		0		0		
3960	152		0		0		
0	0		0		0		
2400	1992		0		0		
0	0		0		0		
0	0		2912		22096		
0	0		0		0		

TempWorkspaceKB	TxnCreateTime		CursorCount	SpCount	
SpNumber					
=====	=====		=====	=====	=====
0	2009-06-26	13:17:27.612	1	3	2
102592	2009-06-26	13:27:28.491	1	1	0
0	2009-06-26	13:30:27.548	0	1	0
0	2009-06-26	13:31:27.151	0	24	262
0	2009-06-26	13:35:02.128	0	0	0
0	2009-06-26	13:43:58.805	0	39	408
128	2009-06-26	13:45:28.379	0	1	0
0	2009-06-26	14:05:15.759	0	42	413
680	2009-06-26	14:57:51.104	1	2	20
0	2009-06-26	15:09:30.319	0	1	0
MPXServerName	GlobalTxnID				
=====	=====				
(NULL)	0				
(NULL)	0				
(NULL)	0				
(NULL)	0				
(NULL)	0				
(NULL)	0				
(NULL)	0				
(NULL)	0				
(NULL)	0				
(NULL)	0				

另请参见

- `sp_iqstatus` 过程（第 453 页）
- `sp_iqversionuse` 过程（第 473 页）

sp_iqunusedcolumn 过程

报告负载未引用的 IQ 列。

语法

`sp_iqunusedcolumn`

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

不会报告在 SYSTEM 中创建的表和本地临时表的列。

表 197. 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 358 页）
- sp_iqindexadvice 过程（第 404 页）
- sp_iqindexuse 过程（第 413 页）
- sp_iqtableuse 过程（第 465 页）
- sp_iqunusedindex 过程（第 470 页）
- sp_iqunusedtable 过程（第 472 页）
- sp_iqworkmon 过程（第 479 页）

sp_iqunusedindex 过程

报告负载未引用的 IQ 辅助（非 FP）索引。

语法

```
sp_iqunusedindex
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

不会报告在 SYSTEM 中创建的表或本地临时表的索引。

表 198. 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	GROUP0	HG
ASIQ_IDX_T455_C3_HG	FinancialData	GROUP0	HG
ASIQ_IDX_T456_I8_HG	Products	GROUP0	HG
ASIQ_IDX_T457_I4_HG	Departments	GROUP0	HG
ASIQ_IDX_T457_C3_HG	Departments	GROUP0	HG
ASIQ_IDX_T458_I21_HG	Departments	GROUP0	HG
ASIQ_IDX_T458_C5_HG	Departments	GROUP0	HG

另请参见

- sp_iqcolumnuse 过程（第 358 页）
- sp_iqindexadvice 过程（第 404 页）
- sp_iqindexuse 过程（第 413 页）

- sp_iqtableuse 过程（第 465 页）
- sp_iqunusedcolumn 过程（第 469 页）
- sp_iqunusedtable 过程（第 472 页）
- sp_iqworkmon 过程（第 479 页）

sp_iqunusedtable 过程

报告负载未引用的 IQ 表。

语法

sp_iqunusedtable

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

不会报告在 SYSTEM 中创建的表和本地临时表。

表 199. 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	GROUP0
Contacts	GROUP0	Employees	GROUP0
emp1	DBA	SalesOrders	GROUP0
FinancialData	GROUP0	Departments	GROUP0
SalesOrderItems	GROUP0	Products	GROUP
iq_dummy	DBA	Customers	GROUP0
sale	DBA		

另请参见

- sp_iqcolumnuse 过程（第 358 页）
- sp_iqindexadvice 过程（第 404 页）
- sp_iqindexuse 过程（第 413 页）
- sp_iqtableuse 过程（第 465 页）
- sp_iqunusedcolumn 过程（第 469 页）

- `sp_iqunusedindex` 过程（第 470 页）
- `sp_iqworkmon` 过程（第 479 页）

`sp_iqversionuse` 过程

显示 IQ main 存储的版本使用情况。

语法

```
sp_iqversionuse
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

描述

`sp_iqversionuse` 系统存储过程可帮助解决由于存在多个表版本而导致的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 服务器上较高百分比的主块在使用时，请运行 `sp_iqversionuse`，以查明正在使用何种版本及通过释放该版本可以恢复的空间量。

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 指示版本使用情况信息 是否已从辅助服务器发送至事务协调器。新版本在事务协调器上的 WasReported 最初为 0。一旦数据库服务器将版本使用情况信息复制回事务协调器，WasReported 就更改为 1。

注意： WasReported 列用在 Multiplex 设置中。有关 Multiplex 的详细信息，请参见《使用 Sybase IQ Multiplex》。

表 200. `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。它由 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。
服务器	该版本的用户所连接的服务器
IQConnID	使用该版本的连接 ID
WasReported	指示服务器是否已收到该版本的使用情况信息

列名	描述
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

下表说明 `sp_iqversionuse` 系统过程的输出样本：

VersionID	Server	IQConnID	WasReported
=====	=====	=====	=====
0	ab2ab_iqdemo	9	0
MinKBRelease	MaxKBRelease		
=====	=====		
0	0		

另请参见

- `sp_iqstatus` 过程（第 453 页）
- `sp_iqtransaction` 过程（第 466 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])
```

语法 2

```
sp_iqview [view_name='viewname'],  
[view_owner='viewowner' ],[view_type='viewtype' ]
```

权限

需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法：语法 1

```
sp_iqview NULL,NULL,SYSTEM
```

如果未指定前两个参数中的任何一个参数，但指定了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` and `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。

用法：语法 2

参数可以任何顺序进行指定。将其用单引号引起来。

允许的 *view_type* 参数值:

表 201. *sp_iqview* 的 *view_type* 值

view_type 值	显示的信息
SYSTEM	系统视图
ALL	用户和系统视图
其它任何值	用户视图

描述

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 202. *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 CREATE VIEW 语句中指定的视图定义
remarks	使用 COMMENT 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 356 页)
- *sp_iqconstraint* 过程 (第 362 页)
- *sp_iqdatatype* 过程 (第 369 页)
- *sp_iqevent* 过程 (第 390 页)
- *sp_iqhelp* 过程 (第 395 页)
- *sp_iqindex* 和 *sp_iqindex_alt* 过程 (第 401 页)
- *sp_iqjoinindex* 过程 (第 415 页)
- *sp_iqpkeys* 过程 (第 431 页)
- *sp_iqprocparm* 过程 (第 436 页)
- *sp_iq_reset_identity* 过程 (第 443 页)
- *sp_iqtable* 过程 (第 460 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 ] ]
```

权限

需要具有 **DBA** 权限。必须向没有 **DBA** 权限的用户授予 **EXECUTE** 权限才能运行存储过程。

描述

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。

表 203. sp_iqwho 列

列名	描述
ConnHandle	SA 连接句柄
IQConnID	特定于 Sybase IQ 的连接 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

列名	描述
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 Sybase IQ 命令的类型；如果未发出任何命令则为 NONE
QIdle	自上个 Sybase IQ 命令通过连接发出以来所经历的时间（以秒为单位）；如果之前未有 Sybase IQ 命令，则显示自 2000 年 1 月 1 日起所经历的时间
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。
QCursors	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 204. sp_who 和 sp_iqwho 列的对应关系

sp_who 列	sp_iqwho 列
fid	块所属的系列；已省略，因为不适用于 Sybase IQ
spid	ConnHandle、IQConnID
status	QIdle、SAIdle
loginame	Userid
origname	用户别名；已省略，因为不适用于 Sybase IQ
hostname	运行服务器的主机的名称；当前不支持
blk_spid	BlockedOn
dbname	已省略，因为对于 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

用法

表 205. 参数

参数	描述
connhandle	表示连接 ID 的整数。如果指定此参数，则 sp_iqwho 将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， 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。如果指定的用户名在数据库中不存在， sp_iqwho 会返回错误消息 “User <i>user-name</i> does not exist”。
arg-type	<i>arg-type</i> 参数是可选的，并且只有在指定第一个参数以后才能指定该参数。 <i>arg-type</i> 的值只能是 “user”。如果将 <i>arg-type</i> 的值指定为 “user”， sp_iqwho 会将第一个参数解释为用户名，即使第一个参数是数字也是如此。如果为 <i>arg-type</i> 指定了 “user” 以外的任何值，则 sp_iqwho 将返回错误 “Invalid parameter” 请用双引号将 <i>arg-type</i> 值引起来。

如果未指定任何参数，**sp_iqwho** 会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。

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"
```

如果指定了 *arg-type* “user”，**sp_iqwho** 会将第一个参数解释为用户名，而不是连接 ID。如果数据库中存在名为 1 的用户，**sp_iqwho** 将显示有关用户 1 打开的连接的信息。

表 206. sp_iqwho 用法示例

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

语法	输出
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			
IQThreads	QIdle	SAIdle	TempTableSpaceKB	TempWorkSpaceKB	
12	118	DBA	CURSOR_OPEN	IQUTILITYOPENCURSOR	0
(NULL)	0	1	0	0	0
13	119	shweta	DO_NOTHING	NONE	0
(NULL)	0	1	16238757	470	0

sp_iqwho 与 Adaptive Server Enterprise 的兼容性

Sybase IQ**sp_iqwho** 存储过程包含由 Adaptive Server Enterprises**sp_who** 过程显示的列的 Sybase IQ 等效项。

某些 Adaptive Server Enterprise 列已省略，因为它们不适用于 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** 收集仅用于查询的信息（包含 **FROM** 子句的 SQL 语句）。您不能将 **sp_iqworkmon** 用于 **INSERT** 或 **LOAD** 语句。

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'
```

权限
需要具有 DBA 权限。必须向没有 DBA 权限的用户授予 EXECUTE 权限才能运行存储过程。

用法

表 207. 参数

参数	描述
action	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。 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。
mode	指定要控制的监控类型。INDEX、TABLE 和 COLUMN 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 ALL 关键字同时控制对所有使用情况监控功能的监控。

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。

仅针对包含 **FROM** 子句的 SQL 语句收集使用情况; 例如, **SELECT**、**UPDATE** 和 **DELETE**。

表 208. sp_iqworkmon 列

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 358 页)
 - sp_iqindexadvice 过程 (第 404 页)
 - sp_iqindexuse 过程 (第 413 页)
 - sp_iqtableuse 过程 (第 465 页)

- `sp_iqunusedcolumn` 过程 (第 469 页)
- `sp_iqunusedindex` 过程 (第 470 页)
- `sp_iqunusedtable` 过程 (第 472 页)

目录存储过程

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 组对它们拥有 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** - 该标准用于核心扩展。SQL:1999 或 SQL:2003 其中之一。
- **statement** - 要计算的 SQL 语句。

注释

如果存在用于语句的非核心扩展，则结果集为空。

权限

无

副作用

无

示例

以下是调用 `sa_ansi_standard_packages` 系统过程的示例：

```
CALL sa_ansi_standard_packages( 'SQL:2003',
'SELECT *
FROM ( SELECT o.SalesRepresentative,
o.Region,
SUM( s.Quantity * p.UnitPrice ) AS total_sales,
DENSE_RANK() OVER ( PARTITION BY o.Region,
GROUPING( o.SalesRepresentative )
ORDER BY total_sales DESC ) AS
sales_rank
FROM Product p, SalesOrderItems s, SalesOrders o
WHERE p.ID = s.ProductID AND s.ID = o.ID
GROUP BY GROUPING SETS( ( o.SalesRepresentative,
o.Region ), o.Region ) ) AS DT
```

```
WHERE sales_rank <= 3
ORDER BY Region, sales_rank');
```

此查询会生成以下结果集：

package_id	package_name
T612	Advanced OLAP operations
T611	Elementary OLAP operations
F591	Derived tables
T431	Extended grouping capabilities

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 200 个字节。

权限

DBA 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' );
```

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

参数	描述
shell_commands	要在系统 shell 中执行的一个或多个用户命令。shell 命令特定于系统 shell。命令用分号 (;) 分隔。

权限
无。

描述
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/'
```

sa_conn_activity 系统过程

为服务器上的每个指定数据库连接返回一个最近预准备 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。

结果集

列名	数据类型	说明
数字	INT	返回连接的 ID 编号。
名称	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: 。
用户 ID	VARCHAR(255)	返回连接的用户 ID。
DBNumber	INT	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

sa_conn_activity 系统过程为每个连接都返回一个包含最近预准备 SQL 语句的结果集（前提是已通知服务器搜集该信息）。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。

如果未指定 **connidparm**，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。如果 **connidparm** 小于零，则返回当前连接的选项值。

由于客户数据库隔离规则，当此系统过程在云中运行时，仅会返回有关当前客户数据库的信息。

权限

无

副作用

无

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。

结果集

列名	数据类型	说明
数字	INTEGER	返回连接的 ID 编号。
名称	VARCHAR(255)	返回连接的 ID 编号。 临时连接名在连接名前面附有 INT: 。
用户 ID	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缓存，则其 ReqType 值为 CONNECT_POOL_CACHE。
CommLink	VARCHAR(255)	返回连接的通信链接。这是 SQL Anywhere 所支持的网络协议之一，或者对于相同计算机连接为 local。
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。

列名	数据类型	说明
LockIndexID	INTEGER	返回锁定索引的标识符。 如果锁与 LockTable 中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，则返回与该锁相关联的表名称。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

如果指定连接 ID 号，则 sa_conn_info 系统过程为提供的连接返回一个包含连接属性的结果集。如果没有提供 connidparm，则此系统过程返回服务器上数据库的所有当前连接的信息。如果 connidparm 小于零，则返回当前连接的选项值。

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。

由于客户数据库隔离规则，当此系统过程在云中运行时，将仅返回有关当前客户数据库的详细。

权限
无

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( );
```

数字	名称	用户 ID	DBNumber	...
79		DBA	0	...
46	Sybase Central 1	DBA	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)

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* 使用此可选的 `INTEGER` 参数指定连接 ID 号。
- *dbidparm* 使用此可选的 `INTEGER` 参数指定数据库 ID 号。

结果集

列名	数据类型	描述
数字	<code>INTEGER</code>	连接 ID 号。

另请参见

“SQL Anywhere Server - SQL 参考” > “系统过程” > “按字母顺序排列的系统过程列表” > “`sa_conn_list` 系统过程”。

注意： 此参考指向 SQL Anywhere 文档。

sa_conn_properties 系统过程

报告连接属性信息。

语法

```
sa_conn_properties ( [ connidparm ] )
```

请参见“SQL Anywhere Server - SQL 参考” > “系统过程” > “按字母顺序排列的 系统过程列表” > “sa_conn_properties 系统过程”。

注意： 此参考指向 SQL Anywhere 文档。

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。

结果集

列名	数据类型	说明
数字	INTEGER	返回连接的 ID 编号。
别名	VARCHAR(255)	返回数据库名。
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。

如果未提供 dbidparm，则返回所有数据库的属性。

由于客户数据库隔离规则，当此系统过程在云中运行时，将仅返回有关当前客户数据库的详细。

权限
无

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
数字	0
别名	demo
文件	
ConnCount	1
PageSize	4096
LogName	C:\Documents and Settings\All Users\Program Data\SybaseIQ\demo\iqdemo.log

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。

结果集

列名	数据类型	说明
数字	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。

列名	数据类型	说明
值	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。

如果未指定 `dbidparm`，则返回所有数据库的属性。

由于客户数据库隔离规则，当此系统过程在云中运行时，将仅返回有关当前客户数据库的详细。

权限

无

副作用

无

示例

以下示例使用 `sa_db_properties` 系统过程为所有数据库返回一个汇总数据库属性信息的结果集。

```
CALL sa_db_properties( );
```

数字	PropNum	PropName	...
0	0	ConnCount	...
0	1	IdleCheck	...
0	2	IdleWrite	...
...	...	...	...

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

- types** - 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。
- **连接** – 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。
- **授权** – 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。

权限

DBA 特权

副作用

无

示例

禁用所有审计:

```
CALL sa_disable_auditing_type( 'all' );
```

sa_disk_free_space 系统过程

报告 dbspace、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_dbspace_name ] )
```

参数

- **p_dbspace_name** – 使用此 VARCHAR(128) 参数指定 dbspace 的名称、事务日志文件、事务日志镜像文件或临时文件。

如果有一个名为 `log`、`mirror` 或 `temp` 的 dbspace，则可以在关键字前面加上一个下划线。例如，若存在一个名为 `log` 的 dbspace，则使用 `_log` 可以获得有关日志文件的信息。

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 dbspace 名称、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	dbspace 所在驱动器的可用磁盘空间的总量。

注释

如果未指定 *p_dbspace_name* 参数或该参数为 NULL，则每个 dbspace 在结果集中都占一行，另外每个事务日志、事务日志镜像和临时文件（如果存在）也都占一行。如果指定了 *p_dbspace_name*，则正好返回一行或零行（如果不存在这样的 dbspace，或者指定了 log 或 mirror 但是没有日志文件或镜像文件，则返回零行）。

权限

DBA 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
系统	10952101888	21410402304
translog	10952101888	21410402304
临时	10952101888	21410402304

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

- **types** – 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。
- **连接** – 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。
- **选项** – 启用公共选项的审计。
- **授权** – 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。
- **触发器** – 启用触发事件之后的审计。

注释

sa_enable_auditing_type 与 PUBLIC.auditing 选项配合使用可启用指定类型信息的审计。

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。

权限

DBA 特权

副作用

无

示例

若仅启用选项审计：

```
CALL sa_enable_auditing_type( 'options' );
```

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。

列名	数据类型	说明
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。
值	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。

权限
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_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。

权限
DBA 特权

副作用
无

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 CURRENT USER。
- **tabname** - 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。

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。

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。

权限

创建或修改数据库对象需要有资源权限

副作用

自动提交

示例

以下语句可确保创建框架过程定义，可以定义过程，并可授予对其的权限。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_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。缺省值为 0。
- **rend** - 使用此可选的 INTEGER 参数指定大于或等于 **rstart** 的结束值。缺省值为 100。
- **rstep** - 使用此可选的 INTEGER 参数指定序列值的增量。缺省值为 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 );
```


权限
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。其中有些范围不包含雇员，将引起警告。

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) 参数服务器选项指定新值。

注释

使用此过程，数据库管理员可以临时替换某些数据库服务器选项，而不需要重新启动数据库服务器。

当服务器关闭时，使用此存储过程更改的选项值将重置为其缺省值。如果想在服务器每次启动时都更改选项值，则可以在数据库服务器启动时指定相应的数据库服务器选项（如果存在），下表中最右侧的列给出了这些选项。

以下选项设置可以更改：

选项名称	值	缺省值
AutoMultiProgrammingLevel	YES、NO	是
AutoMultiProgrammingLevelStatistics	YES、NO	否
CacheSizingStatistics	YES、NO	否
CollectStatistics	YES、NO	是
ConnsDisabled	YES、NO	否
ConnsDisabledForDB	YES、NO	否
ConsoleLogFile	filename	
ConsoleLogMaxSize	file-size，以字节为单位	
CurrentMultiProgrammingLevel	Integer	20

选项名称	值	缺省值
DatabaseCleaner	ON、OFF	ON
DeadlockLogging	ON、OFF、RESET、CLEAR	OFF
DebuggingInformation	YES、NO	否
DropBadStatistics	YES、NO	是
DropUnusedStatistics	YES、NO	是
IdleTimeout	整数（以分钟为单位）	240
IPAddressMonitorPeriod	整数（以秒为单位）	便携设备的缺省值为 120，其它设备为 0
LivenessTimeout	整数（以秒为单位）	120
MaxMultiProgrammingLevel	Integer	CurrentMultiProgrammingLevel 值的四倍
MessageCategoryLimit	Integer	400
MinMultiProgrammingLevel	Integer	-gtc 服务器选项的值和计算机上的逻辑 CPU 的数量的最小值
OptionWatchAction	MESSAGE、ERROR	MESSAGE
OptionWatchList	以逗号分隔的数据库选项列表	
ProcedureProfiling	YES、NO、RESET、CLEAR	否
ProfileFilterConn	<i>connection-id</i>	
ProfileFilterUser	<i>user-id</i>	
QuittingTime	有效的日期和时间	
RememberLastPlan	YES、NO	否
RememberLastStatement	YES、NO	否
RequestFilterConn	<i>connection-id</i> , -1	
RequestFilterDB	<i>database-id</i> , -1	
RequestLogFile	filename	

选项名称	值	缺省值
RequestLogging	SQL、HOSTVARS、 PLAN、 PROCEDURES、 TRIGGERS、OTHER、 BLOCKS、REPLACE、 ALL、YES、NONE、 NO	无
RequestLogMaxSize	<i>file-size</i> , 以字节为单位	
RequestLogNumFiles	Integer	
RequestTiming	YES、NO	否
SecureFeatures	<i>feature-list</i>	
StatisticsCleaner	ON、OFF	ON
WebClientLogFile	<i>filename</i>	
WebClientLogging	ON、OFF	OFF

- **AutoMultiProgrammingLevel** - 设置为 YES 时, 数据库服务器会自动调整其进程并发水平, 该属性控制某一时刻可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** - 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。
- **CacheSizingStatistics** - 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** - 当设置为 YES 时, 数据库服务器搜集性能监控器统计信息。
- **ConnsDisabled** - 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。
- **ConnsDisabledForDB** - 当设置为 YES 时, 不允许与当前数据库建立任何其它连接。
- **ConsoleLogFile** - 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。路径中的任何反斜线字符必须写两遍, 因为这是 SQL 字符串。
- **ConsoleLogMaxSize** - 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, 通过附加的扩展名 **.old**, 该文件被重命名 (如果已经存在具有相同名称的文件, 则将其替换)。然后, 该数据库服务器消息日志文件被重新启动。
- **CurrentMultiProgrammingLevel** - 设置数据库服务器的进程并发水平。

- **DatabaseCleaner** – 除非有 iAnywhere 技术支持的建议，否则不要更改此选项的设置。
- **DeadlockLogging** – 控制死锁记录。同时支持值 `deadlock_logging`。还可以在 Sybase Central 的 [数据库属性] 窗口中配置死锁记录选项。支持以下值：
 - **ON** – 启用死锁记录。
 - **OFF** – 禁用死锁记录，但使死锁数据可供查看。
 - **RESET** – 清除记录的死锁数据（如果存在的话），然后启用死锁记录。
 - **CLEAR** – 清除记录的死锁数据（如果存在的话），然后禁用死锁记录。

启用死锁记录后，可以使用 `sa_report_deadlocks` 系统过程从数据库检索死锁信息。

- **DebuggingInformation** – 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。
- **DropBadStatistics** – 允许对从数据库返回错误估计的删除统计信息进行自动统计信息管理。
- **DropUnusedStatistics** – 允许对从数据库返回的连续 90 天未使用的删除统计信息进行自动统计信息管理。
- **IdleTimeout** – 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保持锁定。
- **IPAddressMonitorPeriod** – 设置检查新 IP 地址的时间（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。
- **LivenessTimeout** – 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** – 设置数据库服务器进程并发水平的最大值。
- **MessageCategoryLimit** – 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** – 设置数据库服务器进程并发水平的最小值。
- **OptionWatchAction** – 指定数据库服务器在尝试设置列表中的选项时应采取的操作。支持的值为 `MESSAGE` 和 `ERROR`。当 `OptionWatchAction` 设置为 `MESSAGE`，并且设置了一个由 `OptionWatchList` 指定的选项时，数据库服务器消息窗口中将出现一条消息，指示正在设置的选项位于选项监视项目列表中。

当 `OptionWatchAction` 设置为 `ERROR` 时，将返回错误，指示无法设置选项，因为该选项在选项监视项目列表中。

可以通过执行以下的查询来查看此属性的当前设置：

```
SELECT DB_PROPERTY( 'OptionWatchAction' );
```

- **OptionWatchList** – 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 `automatic_timestamp`、`float_as_double` 和 `tsql_hex_constant` 选项添加到监视选项列表中。

```
CALL sa_server_option( 'OptionWatchList','automatic_timestamp,
float_as_double,tsql_hex_constant' )
```

可以通过执行以下的查询来查看此属性的当前设置:

```
SELECT DB_PROPERTY( 'OptionWatchList' );
```

- **ProcedureProfiling** – 控制存储过程、函数、事件和触发器的过程分析。过程分析向您显示存储过程、函数、事件和触发器的执行时间。还可以在 Sybase Central 的 [数据库属性] 窗口中设置过程分析选项

- **是** – 对您当前连接到的数据库启用过程分析。
- **否** – 禁用过程分析，但使分析数据可供查看。
- **RESET** – 将分析计数器恢复到零，并且不改变 YES 或 NO 设置。
- **CLEAR** – 将分析计数器恢复到零，并且禁用过程分析。

启用了分析后，可以使用 `sa_procedure_profile_summary` 和 `sa_procedure_profile` 系统过程从数据库中检索分析信息。

- **ProfileFilterConn** – 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，`SELECT DB_PROPERTY( 'ProfileFilterConn' )` 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。
- **ProfileFilterUser** – 指示数据库服务器捕获特定用户 ID 的分析信息。
- **QuittingTime** – 指示数据库服务器在指定的时间关闭。
- **RememberLastPlan** – 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。此设置还由 `-zp` 服务器选项控制。

当启用了 `RememberLastPlan` 时，可以通过查询 `LastPlanText` 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。

```
SELECT CONNECTION_PROPERTY( 'LastPlanText' );
```

- **RememberLastStatement** – 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 `RememberLastStatement` 启用时，可以通过查询 `LastStatement` 连接属性的值为连接获取 `LastStatement` 的当前值:

```
SELECT CONNECTION_PROPERTY( 'LastStatement' );
```

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 `RememberLastStatement` 打开时，下面的语句为指定连接返回最近预准备语句:

```
SELECT CONNECTION_PROPERTY( 'LastStatement', connection-id );
```

`sa_conn_activity` 系统过程为所有连接都返回此相同的信息。

警告！ 当指定 `-zl` 或启用 `RememberLastStatement` 服务器设置时，任何用户都可以调用 `sa_conn_activity` 系统过程或获取 `LastStatement` 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。

- **RequestFilterConn** – 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，这有助于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：

```
CALL sa_conn_info( );
```

一旦获得了连接 ID 后，即可执行以下语句来指定要记录日志信息的特定连接：

```
CALL sa_server_option( 'RequestFilterConn', connection-id );
```

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：

```
CALL sa_server_option( 'RequestFilterConn', -1 );
```

- **RequestFilterDB** – 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，这有助于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：

```
SELECT CONNECTION_PROPERTY( 'DBNumber' );
```

要指定仅记录特定数据库的信息，请执行以下语句：

```
CALL sa_server_option( 'RequestFilterDB', database-id );
```

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：

```
CALL sa_server_option( 'RequestFilterDB', -1 );
```

- **RequestLogFile** – 用来记录请求信息的文件的名称。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。路径中的任何反斜线字符必须写两遍，因为这是 `SQL` 字符串。

如果启用了客户端语句高速缓存，则应将 `max_client_statements_cached` 选项设为 0 以禁用客户端语句高速缓存。同时，如果会使用 `tracetime.pl` Perl 脚本对日志进行分析，还将捕获请求日志。

- **RequestLogging** – 此调用启用对发送到数据库服务器的各 `SQL` 语句的记录，以便于疑难解答，此选项设置同数据库服务器选项 `-zr` 和 `-zo` 一起使用。值可以是以下项的组合，这些项由加号 (+) 或逗号分隔：
 - **PLAN** – 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。
 - **HOSTVARS** – 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。
 - **PROCEDURES** – 启用从过程中执行的语句的记录。
 - **TRIGGERS** – 启用从触发器中执行的语句的记录。
 - **OTHER** – 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。

- **BLOCKS** - 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。
- **REPLACE** - 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。
- **ALL** - 记录所有支持的信息。这等效于指定 **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。
- **NO 或 NONE** - 禁用记录请求日志。

可以通过执行以下的查询来查看此属性的当前设置：

```
SELECT PROPERTY( 'RequestLogging' );
```

- **RequestLogMaxSize** - 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，则用相同名称替换现有的文件）。然后，重新启动请求日志文件。

- **RequestLogNumFiles** - 要保留的请求日志文件副本的数目。

如果在较长一段时间内启用了请求记录，则请求日志文件可能会变得很大。`-zn` 允许您指定要保留的请求日志文件副本的数量。

- **RequestTiming** - 指示数据库服务器维护每个连接的计时信息。缺省情况下，此功能关闭。此功能打开时，数据库服务器会为每个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中所花费的具体时间。可以使用 `sa_performance_diagnostics` 系统过程来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：

```
ReqCountUnscheduled
ReqTimeUnscheduled
ReqCountActive
ReqTimeActive
ReqCountBlockIO
ReqTimeBlockIO
ReqCountBlockLock
ReqTimeBlockLock
ReqCountBlockContention
ReqTimeBlockContention
```

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。

- **SecureFeatures** – 可用于启用或禁用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 (-)。

为启用或禁用功能而进行的任何更改都会立即对连接生效。这些设置不影响执行 `sa_server_option` 系统过程的连接；必须断开并重新连接才能使设置生效。

注意： 要使用 `sa_server_option` 系统过程启用或禁用功能，必须在启动数据库服务器时使用 `-sk` 选项指定一个密钥，然后将 `secure_feature_key` 数据库选项的值设置为 `-sk` 所指定的密钥（例如，`SET TEMPORARY OPTION secure_feature_key = 'j978kls12'`）。将 `secure_feature_key` 数据库选项设置为 `-sk` 值使您可以更改安全功能的设置。

例如，要禁用两个功能并启用第三个功能，应使用以下语法：

```
CALL sa_server_option('SecureFeatures',
'CONSOLE_LOG,WEBCLIENT_LOG,-REQUEST_LOG' );
```

执行了该语句后，`CONSOLE_LOG` 和 `WEBCLIENT_LOG` 被添加到安全功能列表中，而 `REQUEST_LOG` 将从列表中删除。

- **StatisticsCleaner** – 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** – Web 服务客户端日志文件的名称。每次使用 `-zoc` 服务器选项或 `WebClientLogFile` 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。路径中的任何反斜线字符都必须双写，因为这是字符串。
- **WebClientLogging** – 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写入记录，而指定 `OFF` 可停止向该文件写入记录。

权限

以下选项与应用程序分析或请求记录有关，需要具有 `DBA` 或 `PROFILE` 权限：

```
ProcedureProfiling
ProfileFilterConn
ProfileFilterUser
RequestFilterConn
RequestFilterDB
RequestLogFile
RequestLogging
RequestLogMaxSize
RequestLogNumFiles
```

所有其它选项都要求具有 `DBA` 权限。

副作用

无

示例

以下语句禁止与数据库服务器建立新连接:

```
CALL sa_server_option( 'ConnsDisabled', '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, 并指定要启动一个新请求日志:

```
CALL sa_server_option( 'RequestLogging', 'SQL+PROCEDURES+BLOCKS+PLAN+REPLACE' );
```

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 );
```

权限
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 服务器的首选项。可以按首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端没有交叉列表时，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。客户端的首选项顺序具有更高的优先级。如果不存在互斥的编码首选项，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。

请注意，如果 **CharsetConversion** 选项设置为 OFF，则不会执行 **AcceptCharset**。

- **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。当使用空值调用时，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。

在通过 web 服务调用的过程中调用 sa_set_http_option 时，如果此选项无效或包含无效值，将返回错误。

权限

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语字母表的 HTML 文档。

```
CREATE 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：

```

DECLARE sessionid VARCHAR(30);
DECLARE tm TIMESTAMP;
SET tm = NOW(*);
SET sessionid = 'MySessions_' ||
    CONVERT( VARCHAR, SECONDS(tm)*1000 + DATEPART(millisecond,tm));
CALL sa_set_http_option('SessionID', sessionid);

```

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。
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。

权限

DBA 特权

副作用

无

sa_validate 系统过程

校验全部或部分数据库。

语法

```
sa_validate(  
    [ tbl_name [, owner_name ] ]  
)
```

参数

- **tbl_name** – 此可选的 VARCHAR(128) 参数用于指定要校验的表或实例化视图的名称。
- **owner_name** – 此可选的 CHAR(128) 参数用于指定所有者。单独指定时，将校验所有者拥有的所有表和实例化视图。

权限

DBA 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>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>	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> 与 <i>tbl_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。

tbl_name 和 *owner_name* 的值是字符串并且必须用引号括起来。

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 **[No error detected]**。

警告！ 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 DBA 拥有的表和实例化视图执行校验：

```
CALL sa_validate( owner_name = 'DBA' );
```

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_pwsd** – 使用此 CHAR(128) 参数指定当前数据库用户的口令。

注释

此过程由 sp_password 使用。如果口令匹配，则接受。如果口令不匹配，则返回一条错误。

权限

无

副作用

无

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。

以下是 sp_login_environment 过程的文本：

```
CREATE PROCEDURE dbo.sp_login_environment( )
BEGIN
    IF connection_property( 'CommProtocol' ) = 'TDS' THEN
        CALL dbo.sp_tsql_environment( )
    END IF
END;
```

权限

无

副作用

无

sp_remote_columns 系统过程

生成远程表上列的列表及其数据类型的说明。

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 ( @servername , @tablename [, @table_owner ] [,
@table_qualifier ] ] )
```

请参见“SQL Anywhere Server - SQL 参考” > “系统过程” > “按字母顺序排列的系统过程列表” > “sp_remote_columns 系统 过程”。

注意： 此参考指向 SQL Anywhere 文档。

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
    , sp_name
    [, sp_owner
    [, sp_qualifier ] ]
)
```

参数

- **server_name** - 使用此 CHAR(128) 参数指定主表所在的服务器。
- **sp_name** - 使用此 CHAR(128) 参数指定包含主键的表。
- **sp_owner** - 使用此可选的 CHAR(128) 参数指定主表的所有者。
- **sp_qualifier** - 使用此可选的 CHAR(128) 参数指定包含主表的数据库。

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。

权限

无

副作用

无

示例

此示例在调用系统过程时使用已命名的参数，并返回包含 asetest 服务器上 production 数据库中 SYSOBJECTS 表上的外键的远程表信息：

```
CALL sp_remote_exported_keys(
    @server_name='asetest',
    @sp_name='sysobjects',
    @sp_qualifier='production' );
```

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
, sp_name
[, sp_owner
[, sp_qualifier ] ]
)
```

参数

- **server_name** – 使用此可选的 CHAR(128) 参数指定外键表所在的服务器。此参数需要值。
- **sp_name** – 使用此可选的 CHAR(128) 参数指定包含外键的表。此参数需要值。

- **sp_owner** – 使用此可选的 CHAR(128) 参数指定外键表的所有者。
- **sp_qualifier** – 使用此可选的 CHAR(128) 参数指定包含外键表的数据库。

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。

权限

无

副作用

无

示例

以下示例在调用过程时使用已命名的参数，并返回包含与 asetest 服务器 SYSOBJECTS 表中的外键相对应的主键的表：

```
CALL sp_remote_imported_keys(  
    @server_name='asetest',  
    @sp_name='sysobjects',  
    @sp_qualifier='production' );
```

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 ] ] ] )
```

请参见“SQL Anywhere Server - SQL 参考” > “系统过程” > “按字母顺序排列的系统过程列表” > “sp_remote_imported_keys 系统过程”。

注意： 此参考指向 SQL Anywhere 文档。

sp_remote_tables 系统过程

返回服务器上表的列表。

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_tables(  
    server_name  
    [, table_name  
    [, table_owner  
    [, table_qualifier  
    [, with_table_type ] ] ]  
)
```

参数

- **server_name** – 使用此 CHAR(128) 参数指定远程表所在的服务器。
- **table_name** – 使用此可选的 CHAR(128) 参数指定远程表。
- **table_owner** – 使用此可选的 CHAR(128) 参数指定远程表的所有者。
- **table_qualifier** – 使用此可选 CHAR(128) 参数指定 **table_name** 所在的数据库。
- **with_table_type** – 使用此可选的 BIT 参数指定远程表的类型。此参数是位类型，接受两个值：0（缺省值）和 1。如果希望结果集中包含列出表类型的列，则必须输入值 1。

结果集

列名	数据类型	说明
数据库	CHAR(128)	远程数据库的名称。
owner	CHAR(128)	远程数据库所有者的名称。
table-name	CHAR(128)	远程表。

列名	数据类型	说明
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
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。

标准和兼容性

- **Sybase** – 受 Open Client/Open Server 支持。

权限
无

副作用
无

示例
获取名为 excel 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( 'excel' );
```

获取 production 数据库（在名为 asetest 的 Adaptive Server Enterprise 服务器中）中由 fred 拥有的所有表的列表：

```
CALL sp_remote_tables( 'asetest', null, 'fred', 'production' );
```

sp_servercaps 系统过程

显示远程服务器的容量信息。

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_servercaps( sname )
```

- 参数*
- **sname** – 使用此 CHAR(64) 参数指定通过 CREATE SERVER 语句定义的服务器。
sname 必须与 CREATE SERVER 语句中使用的服务器名称相同。

注释

此过程显示有关远程服务器的容量的信息。使用此容量信息来确定可以向远程服务器转发多少 SQL 语句。直到第一次连接到远程服务器后，才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。

权限

无

副作用

无

示例

显示有关远程服务器 testasa 的信息：

```
CALL sp_servercaps( 'testasa' );
```

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。

如果希望改变缺省行为，建议创建新过程并改变 login_procedure 选项，使其指向这些新过程。

权限

无

副作用

无

示例

以下是 sp_tsql_environment 过程的文本：

```

CREATE PROCEDURE dbo.sp_tsql_environment()
BEGIN
    IF db_property( 'IQStore' ) = 'Off' THEN
        -- SQL Anywhere 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';
END

```

Adaptive Server Enterprise 系统和 Catalog 存储过程

Adaptive Server Enterprise 提供了系统过程和 Catalog 存储过程来执行许多管理功能并获得系统信息。Sybase IQ 支持这些过程中的某些过程。

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 Enterprise 和 Sybase IQ 12 版之前的版本中执行的功能相同，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。若要查看存储过程的文本，请运行：

```
sp_helptext 'owner.procedure_name'
```

对于所有由 Sybase 提供的系统存储过程来说，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，您必须指定该用户，例如：

```
sp_helptext 'myname.myprocedure'
```


Adaptive Server Enterprise 系统过程

Sybase IQ 中提供的 Adaptive Server Enterprise 系统过程：

表 209. Sybase IQ 中提供的 ASE 系统过程

系统过程	描述	权限
sp_addgroup <i>group-name</i>	将组添加到数据库中	需要 DBA 或 PERMS ADMIN 才能将现有用户更改为组。需要 DBA 或 USER ADMIN 和 PERMS ADMIN 才能创建新用户并将其更改为组。
sp_addlogin <i>userid, password[, defdb[, deflanguage[, fullname]]]</i>	将新用户帐户添加到数据库中	需要具有 DBA 或 USER ADMIN 权限。
sp_addmessage <i>message-num, 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	需要具有 DBA 或 RESOURCE 权限。
sp_addtype <i>typename, datatype[, "identity"/ nulltype]</i>	创建用户定义数据类型。Sybase IQ 不支持 IDENTITY 列。	需要具有 DBA 或 RESOURCE 权限。
sp_adduser <i>userid[, name_in_db[, grpname]]</i>	将新用户添加到数据库中	需要具有 DBA 或 USER ADMIN 权限才能创建新用户。需要具有 DBA 或 USER ADMIN 和 PERMS ADMIN 权限才能创建新用户并将其添加到指定组中。
sp_changegroup <i>new-group-name, userid</i>	更改用户所在组或将用户添加到组中	需要具有 DBA 或 PERMS ADMIN 权限。
sp_dboption <i>[dbname, optname, {true false}]</i>	显示或更改数据库选项	不需要。
sp_dropgroup <i>group-name</i>	从数据库中删除组	需要具有 DBA 或 PERMS ADMIN 权限。
sp_droplogin <i>userid</i>	从数据库中删除用户	需要具有 DBA 或 USER ADMIN 权限。
sp_dropmessage <i>message-number[, language]</i>	删除用户定义的消息	需要具有 DBA 或 RESOURCE 权限。
sp_droptype <i>typename</i>	删除用户定义数据类型	需要具有 DBA 或 RESOURCE 权限。

系统过程	描述	权限
sp_dropuser <i>userid</i>	从数据库中删除用户	需要具有 DBA 或 USER ADMIN 权限。
sp_getmessage <i>message-num, @msg-var output [, language]</i>	从 SYSUSERMESSAGES 中检索存储消息字符串以供 PRINT 和 RAISERROR 语句使用。	不需要。
sp_helptext 'owner.object-name'	显示系统过程或视图的文本	不需要。
sp_password <i>caller_passwd, new_passwd [, userid]</i>	添加或更改用户 ID 的口令	不具有 DBA 权限的用户需要具有 DBA 权限或 PERMS ADMIN 权限才能更改其他用户的口令。如果您不具有这些权限，则可以更改自己的口令。

注意： 像 **sp_dropuser** 这样的过程提供与 Adaptive Server Enterprise 存储过程的最小兼容性。如果您习惯使用 Adaptive Server Enterprise（或 Sybase IQ 11.x）存储过程，请在 **使用 Interactive SQL 中的过程之前**，将它们的文本与 Sybase IQ 12 过程进行比较。若要进行比较，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**。若要查看由不同用户拥有的同名存储过程的文本，您必须指定该用户，例如：

```
sp_helptext 'myname.myprocedure'
```

另请参见

- 用户、组和权限（第 626 页）

Adaptive Server Enterprise Catalog 存储过程

除了 **sp_column_privileges** 过程以外，Sybase IQ 实现大多数 Adaptive Server Enterprise Catalog 存储过程。

对于某些 Adaptive Server Enterprise Catalog 存储过程，Sybase IQ 还具有相似的自定义存储过程。

表 210. Sybase IQ 中实现的 ASE Catalog 存储过程

ASE 目录存储过程	描述	IQ 过程
sp_columns <i>table-name [, table-owner] [, table-qualifier] [, column-name]</i>	返回指定列的数据类型	

ASE 目录存储过程	描述	IQ 过程
sp_fkeys <i>pktable_name</i> [, <i>pktable-owner</i>][, <i>pktable-qualifier</i>][, <i>fktable_name</i>][, <i>fktable_owner</i>][, <i>fktable-qualifier</i>]	返回指定表的外键信息	
sp_pkeys <i>table-name</i> [, <i>table_owner</i>][, <i>table_qualifier</i>]	返回单个表的主键信息	sp_iqpkeys
sp_special_columns <i>table_name</i> [, <i>table-owner</i>][, <i>table-qualifier</i>][, <i>col-type</i>]	返回唯一标识表中某一行的最佳列集	
sp_sproc_columns <i>proc-name</i> [, <i>proc_owner</i>][, <i>proc-qualifier</i>][, <i>column-name</i>]	返回有关存储过程的输入和返回参数的信息	sp_iqprocparm
sp_stored_procedures [<i>sp-name</i>][, <i>sp-owner</i>][, <i>sp-qualifier</i>]	返回有关一个或多个存储过程的信息	sp_iqprocedure
sp_table <i>table-name</i> [, <i>table-owner</i>][, <i>table-qualifier</i>][, <i>table-type</i>]	返回可出现在 FROM 子句中的对象列表	

不支持以下 Adaptive Server Enterprise Catalog 存储过程：

- **sp_column_privileges**
- **sp_databases**
- **sp_datatype_info**
- **sp_server_info**

SQL Anywhere 支持的过程

Sybase IQ 支持 SQL Anywhere 系统过程。

请参见“SQL Anywhere Server - SQL 参考” > “系统对象” > “系统过程” > “按字母顺序排列的系统过程列表”。

只有 SQL Anywhere 表才支持 **sa_get_table_definition** 过程。如果对 IQ 表运行此过程，则此过程会返回错误 **not implemented for IQ tables**。

系统表和视图

以下主题介绍 Sybase IQ 支持的系统表、系统视图、统一视图、兼容性视图以及 ASE T-SQL 兼容性视图。

系统表

每个 Sybase IQ 数据库的结构都在很多系统表 中进行描述。系统表供内部 使用。

DUMMY 系统表是唯一允许您直接访问的系统表。对于所有其它系统表，可通过相应视图访问它们的基本数据。

表 211. 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
ISYSIQJOINIDX	是
ISYSIQJOINIXCOLUMN	是
ISYSIQLOGICALSERVER	是
ISYSIQLOGINPOLICYLSINFO	是
ISYSIQLSLOGINPOLICYOPTION	是
ISYSIQLSMEMBER	是
ISYSIQLSPOLICY	是
ISYSIQLSPOLICYOPTION	是
ISYSIQMPXSERVER	是
ISYSIQPARTITIONCOLUMN	是
ISYSIQTAB	是
ISYSIQTABCOL	是
ISYSJAR	是
ISYSJARCOMPONENT	是
ISYSJAVACLASS	是
ISYSLOGINMAP	是

系统表	仅在内部使用?
ISYSLOGINPOLICY	是
ISYSLOGINPOLICYOPTION	是
ISYSMVOPTION	是
ISYSMVOPTIONNAME	是
ISYSOBJECT	是
ISYSOPTION	是
ISYSOPTSTAT	是
ISYSPARTITION	是
ISYSPARTITIONKEY	是
ISYSPARTITIONSCHEME	是
ISYSPHYSIDX	是
ISYSPROCEDURE	是
ISYSPROCPARM	是
ISYSPROCPERM	是
ISYSPROXYTAB	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

系统表	仅在内部使用?
ISYSSYNCPROFILE	是
ISYSSYNCSRIPT	是
ISYSTAB	是
ISYSTABCOL	是
ISYSTABLEPERM	是
ISYTEXTCONFIG	是
ISYTEXTIDX	是
ISYTEXTIDXTAB	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
```

使用 DUMMY 表的查询是由 SQL Anywhere (Catalog 存储) 来运行的，而不是由 Sybase IQ 来运行的。可以在 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;
```

```
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。请注意，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

使用系统视图来查看系统表的内容。

我们提供了若干个预定义的系统视图，这些视图以可读格式显示系统表中的信息。

系统视图的定义及其说明也包括在内。这些定义中的某一些很复杂，但您并不需要理解它们便可加以使用。

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。

统一视图（如 SYSCATALOG 和 SYSINDEXES）是 Sybase IQ 和 SQL Anywhere 所共有的视图。有关这些视图和其它视图的定义，请参见“SQL Anywhere Server - SQL 参考”>“视图”>“统一视图”。

兼容性视图

兼容性视图是为了与早期版本的 SQL Anywhere 和 Sybase IQ 兼容而提供的 不建议使用的视图。

请尽可能使用系统视图和统一视图 而不使用兼容性视图，因为在 Sybase IQ 的未来版本中 可能不支持兼容性视图。

有关兼容性视图的详细信息，请参见“SQL Anywhere Server - SQL 参考”>“视图”>“兼容性视图”。

ASE T-SQL 兼容性视图

Sybase IQ 提供了一组由特殊用户 DBO 拥有的视图，它们对应于 Adaptive Server Enterprise 系统表 和视图。

另请参见

- Transact-SQL 兼容性视图 （第 608 页）

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)
```

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。

列名	数据类型	说明
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.dbospace_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.dbospace_id = dbs.dbospace_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)

```

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 权限赋予表中的单个列。SYSCOLPERM 系统视图的一行中记录了每个具有 UPDATE、SELECT 或 REFERENCES 权限的列。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	用户 ID 的用户号，该用户 ID 被赋予该列的权限。如果 grantee 是特殊 PUBLIC 用户 ID 的用户号，则将该权限赋予所有用户 ID。
grantor	UNSIGNED INT	授予权限的用户 ID 的用户号。
column_id	UNSIGNED INT	此列编号与 table_id 一同标识被授权权限的列。
privilege_type	SMALLINT	此列中的编号指示列权限的种类（16=REFERENCES、1=SELECT 或 8=UPDATE）。
is_grantable	CHAR(1)	表示是否将 WITH GRANT OPTION 授予了该列权限。

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。

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,s.update_time,s.density,s.max_steps,
s.actual_steps,s.step_values,s.frequencies
```

```

from SYS.ISYSCOLSTAT as s
  join SYS.ISYSTABCOL as c on(s.table_id = c.table_id
    and s.column_id = c.column_id)
  join SYS.ISYSTAB as t on(t.table_id = c.table_id)
  join SYS.ISYSUSER as 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 Enterprise 数据库中的表 (第 608 页)
- SYSCOMMENTS ASE 兼容性视图 (第 537 页)
- SYSINDEXES ASE 兼容性视图 (第 552 页)
- SYSIQOBJECTS ASE 兼容性视图 (第 562 页)
- SYSIQVINDEXT ASE 兼容性视图 (第 564 页)
- SYSOBJECTS ASE 兼容性视图 (第 569 页)
- SYSTYPES ASE 兼容性视图 (第 601 页)
- SYSUSERS ASE 兼容性视图 (第 606 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 Enterprise 数据库中的表 (第 608 页)
- SYSCOLUMNS ASE 兼容性视图 (第 537 页)
- SYSINDEXES ASE 兼容性视图 (第 552 页)
- SYSIQOBJECTS ASE 兼容性视图 (第 562 页)
- SYSIQVINDEXT ASE 兼容性视图 (第 564 页)
- SYSOBJECTS ASE 兼容性视图 (第 569 页)
- SYSTYPES ASE 兼容性视图 (第 601 页)
- SYSUSERS ASE 兼容性视图 (第 606 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： • C - 列检查约束 • T - 表约束 • P - 主键 • F - 外键 • U - 唯一约束
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)
```

SYSDBFILE 系统视图

SYSDBFILE 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 ISYSDBFILE。

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。对于 system 和 TEMPORARY 之外的 dbspace，可以用以下语句更改文件名： <code>ALTER DBSPACE dbspace RENAME 'new-filename';</code>
file_name	LONG VARCHAR	dbspace 的唯一名称。该名称用在 CREATE TABLE 命令中。

列名	数据类型	说明
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 的文件名。对于 SYSTEM dbspace，该值是创建数据库时的数据库文件的名称，它仅供参考且不能更改。对于其它 dbspace，可以用以下语句更改文件名： <code>ALTER DBSPACE dbspace RENAME 'new-filename';</code>
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 文件的权限。该视图的基础系统表为 ISYSDBSpacePerm。

SYSDBSpacePerm 视图为 SQL Anywhere 系统视图。请参见“SQL Anywhere Server - SQL 参考” > “视图” > “系统视图” > “SYSDbFile 系统视图”。

注意： 此参考指向 SQL Anywhere 文档。

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，则该视图就被认为是依赖于该表。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 CREATE TABLE 命令中使用的数据类型（如 CHAR 或 INTEGER）的名称。
type_id	SMALLINT	ODBC 数据类型。该值与 Transact-SQL 兼容性 dbo.SYSTYPES 表中的 data_type 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 NULL。

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。
已启用	CHAR(1)	指出是否允许触发事件。
位置	CHAR(1)	将要触发事件的位置： C = 统一 R = 远程 A = 全部
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。

SYSEVENTTYPE 视图为 SQL Anywhere 系统视图。请参见“SQL Anywhere Server - SQL 参考” > “视图” > “系统视图” > “SYSEVENT 系统视图”。

注意： 此参考指向 SQL Anywhere 文档。

SYSEXTERNENV 系统视图

支持许多外部运行时环境，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)	外部环境或语言的名称。
作用域	char(1)	标识外部环境是以 one-per-connection (C) 还是 one-per-database (D) 启动。
support_result_sets	char(1)	标识可以将结果集返回给用户的外部环境。
位置	long varchar	标识可以找到环境的主要可执行文件的位置。
选项	long varchar	标识启动外部环境的命令行所需的选项。
user_id	unsigned int	仅供内部使用。

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

支持许多外部运行时环境，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	标识外部对象的创建者/所有者。

列名	数据类型	说明
name	long varchar	标识在 INSTALL 语句中指定的外部对象的名称。
contents	long binary	外部对象的内容。
update_time	timestamp	标识上次修改（或安装）对象的时间。

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。

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自动创建 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.dbpace_id = 0 and b.dbfile_id = 0 then
    db_property('File')
  else
    if b.dbpace_id = 15 and b.dbfile_id = 15 then
      db_property('TempFileName')
    else
      b.file_name
    endif
  endif as file_name,
  a.dbpace_name,
  a.store_type,
  b.lob_map,
  b.dbpace_id
from SYS.ISYSDBSPACE as a
  join SYS.ISYSDBFILE as b on(a.dbpace_id = b.dbpace_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。
primary_table_id	UNSIGNED INT	主表的表号。
primary_index_id	UNSIGNED INT	主键的索引号。

列名	数据类型	说明
match_type	TINYINT	约束的匹配类型。匹配类型包括： • 0 - 使用缺省匹配 • 1 - SIMPLE • 2 - FULL • 129 - SIMPLE UNIQUE • 130 - FULL UNIQUE
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。注意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)
```

SYSFOREIGNKEY 兼容性视图 (不建议使用)

提供 SYSFOREIGNKEY 视图的目的是为了与提供 SYSFOREIGNKEY 系统表的旧版本软件兼容。但以前的 SYSFOREIGNKEY 系统表已被 ISYSFKEY 系统表所取代，并且您应当使用与该系统表对应的 ISYSFKEY 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 SYSFOREIGNKEY 中的一行和 SYSFKCOL 的一行或多行定义。SYSFOREIGNKEY 包含有关外键的一般信息，而 SYSFKCOL 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"."SYSFOREIGNKEY"  
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.ISYSINDEX as b  
on a.object_id = b.object_id and a.index_id = b.index_id  
left outer join SYS.ISYSINDEX as p  
on a.object_id = p.object_id and a.index_id = p.index_id  
left outer join SYS.ISYSINDEX as r  
on a.object_id = r.object_id and a.index_id = r.index_id)
```

```

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(
fk_col.table_id = fk_col.table_id
and fk_col.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 系统视图

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。该视图的基础系统表为 ISYSGROUP。

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 统一视图

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.ISYSGROUP,SYS.ISYSUSER as g,SYS.ISYSUSER as u  
    where ISYSGROUP.group_id = g.user_id and ISYSGROUP.group_member =  
u.user_id
```

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： • INIT - 有关何时创建数据库的信息。 • UPGRADE - 有关何时升级数据库的信息。 • START - 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。 • LAST_START - 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前存储在 LAST_START 行中的值不同，LAST_START 操作将会被转换成 START 操作。 • DTT - 有关倒数第二次对 dbspace 执行的磁盘传送时间 (Disk Transfer Time，简称 DTT) 校准操作的信息。即有关倒数第二次执行 ALTER DATABASE CALIBRATE 或 ALTER DATABASE RESTORE DEFAULT CALIBRATION 语句的信息。 • LAST_DTT - 有关最新对 dbspace 执行的 DTT 校准操作的信息。即有关最新执行 ALTER DATABASE CALIBRATE 或 ALTER DATABASE RESTORE DEFAULT CALIBRATION 语句的信息。 • LAST_BACKUP - 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。
object_id	UNSIGNED INT	对于 DTT 和 LAST_DTT 以外的任何操作，该列的值都将为 0。对于 DTT 和 LAST_DTT 操作，这是在 SYSDBSpace 系统视图中定义的 dbspace 的 dbspace_id 。
sub_operation	CHAR(128)	对于 DTT 和 LAST_DTT 以外的任何操作，该列的值将是一组空单引号 ("")。对于 DTT 和 LAST_DTT 操作，该列包含对 dbspace 执行的子操作的类型。值包括： • DTT_SET - 已设置 dbspace 校准。 • DTT_UNSET - 已将 dbspace 校准恢复为缺省设置。
版本	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。

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 SYSPHYSIDX 系统视图中物理索引的 object_id 相对应。
dbspace_id	SMALLINT	包含索引的文件的 ID。该值对应 SYSDBSpace 系统视图中的一个条目。
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。
index_category	TINYINT	索引的类型。值包括： • 1 - 主键 • 2 - 外键 • 3 - 次级索引（包括唯一约束） • 4 - 文本索引
"unique"	TINYINT	指出索引是唯一索引 (1)、非唯一索引 (4) 还是唯一约束 (2)。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。

列名	数据类型	说明
not_enforced	CHAR(1)	仅供系统使用。
file_id	SMALLINT	仅供系统使用。

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 SYSIDX 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 SYSCOLUMN 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 SYSINDEX 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 icc."order" = 'A' then ' ASC' else ' DESC' endif) order by
icc.table_id asc,icc.index_id asc,icc.sequence asc)
from SYS.ISYSIDXCOL as icc
join SYS.ISYSTABCOL as c on(
c.table_id = icc.table_id
and c.column_id = icc.column_id)
where icc.index_id = idx.index_id
and icc.table_id = idx.table_id),
0,0
from SYS.ISYSTAB as tab
join SYS.ISYSDBSPACE as dbs on(tab.dbpace_id = dbs.dbpace_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 Enterprise 数据库中的表 (第 608 页)
- SYSCOLUMNS ASE 兼容性视图 (第 537 页)
- SYS COMMENTS ASE 兼容性视图 (第 537 页)
- SYSIQOBJECTS ASE 兼容性视图 (第 562 页)
- SYSIQVINDEXT ASE 兼容性视图 (第 564 页)
- SYSOBJECTS ASE 兼容性视图 (第 569 页)
- SYSTYPES ASE 兼容性视图 (第 601 页)
- SYSUSERS ASE 兼容性视图 (第 606 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 340 页）

SYSIQBACKUPHISTORYDETAIL 系统视图

此视图描述备份时数据库中存在的所有 **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
bu_id	unsigned bigint	操作检查点的事务标识符。备份操作的备份 ID。
dbspace_id	smallint	与此 dbfile 记录关联的 dbspace ID。
dbfile_id	smallint	持续备份操作过程中，dbspace 中存在的 dbfile ID
dbspace_rwstatus	char(1)	T 表示读写
dbspace_createid	unsigned bigint	创建 dbspace 的事务的事务 ID
dbspace_alterid	unsigned bigint	将 dbspace 标记为 RO 的事务 ID。如果未标记，则为创建 ID
dbspace_online	char(1)	T 表示联机
dbfile_rwstatus	char(1)	T 表示读写
dbfile_createid	unsigned bigint	创建此 dbfile 的事务的事务 ID
dbfile_alterid	unsigned bigint	最后一个更改此 dbfile 的读写状态的事务的事务 ID
is_backed_up	char(1)	表示 dbfile 是在此备份中备份的
start_block	unsigned bigint	dbfile 的起始块
num_blocks	unsigned bigint	dbfile 中块的总数
num_blocks_backed_up	unsigned bigint	已备份的块的总数
dbspace_name	char(128)	dbspace 名称
dbfile_name	char(128)	dbfile 的逻辑文件名
dbfile_path	long varchar	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 563 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 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 服务器用来打开 IQ dbspace 的 IQ dbspace 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 556 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	为数据库中的每个 dbspace 分配一个唯一的编号 (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

列名	列类型	描述
create_txn_id	unsigned bigint	创建 dbspace 的事务 ID
alter_txn_id	unsigned bigint	上次修改读写状态的事务 ID
striping_on	char(1)	“T” - 打开磁盘条带化；“F” - 关闭磁盘条带化
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 KB 数

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 554 页）

SYSIQIDX 系统视图

以可读格式显示 ISYSIQIDX 中的组信息。SYSIQIDX 视图中的每一行均描述了一个 IQ 索引。

注意： 此视图将取代不建议使用的 SYSQINDEX 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
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_table_id	unsigned int	供内部使用
link_index_id	unsigned int	供内部使用

列名	列类型	描述
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。

ISYSIQINFO 系统表表示在使用 **CREATE DATABASE** 创建 Sybase IQ 数据库时所定义的数据库特征。它始终只包含一行。

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
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

列名	列类型	描述
multiplex name	CHAR(128) NULL	此数据库是其成员的 Multiplex 的名称
last_multiplex_mode	TINYINT NULL	(Sybase IQ 中未使用的列) 最后打开目录读写的 服务器的模式。 以下值之一。 • 0 - 单节点。 • 1 - 读取器。 • 2 - 事务协调器。 • 3 - 写入器。

SYSIQJOINIDX 系统视图

以可读格式显示 **ISYSIQJOINIDX** 中的组信息。**SYSIQJOINIDX** 视图中的每一行均描述了一个 IQ 连接索引。

注意： 此视图将取代不建议使用的 **SYSIQJOININDEX** 系统视图。

列名	列类型	描述
joinindex_id	unsigned int	每个连接索引都被指派了一个属于主键的唯一编号。
jvt_id	unsigned int	供内部使用。
dbspace_id	smallint	dbspace 的 ID。
joinindex_name	char(128)	定义连接索引的名称。
joinindex_type	char(12)	供内部使用。
creator	unsigned int	创建连接索引的用户号。
join_info_location	hs_vdorecid	供内部使用。
join_info_loc_size	unsigned int	供内部使用。
join_info_size	unsigned int	供内部使用。
block_map	hs_blockmapidentity	供内部使用。
block_map_size	unsigned int	供内部使用。
vdo	hs_vdoidentity	供内部使用。
vdo_size	unsigned_int	供内部使用。
commit_txn_id	unsigned bigint	供内部使用。
txn_id	unsigned bigint	供内部使用。
有效的	char(1)	指示是否需要同步此连接索引。Y 表示它不需要同步，N 表示它需要同步。

基础系统表的约束：

主键 (joinindex_id)

外键 (jvt_id) 引用 SYS.ISYSTAB

外键 (dbspace_id) 引用 SYS.ISYSDBSPACE

外键 (creator) 引用 SYS.ISYSUSER

唯一 (jvt_id、 commit_txn_id、 txn_id)

另请参见

- SYSIQJOININDEX 系统视图（不建议使用）（第 559 页）

SYSIQJOININDEX 系统视图（不建议使用）

SYSIQJOININDEX 已由 SYSIQJOINIDX 系统视图取代。

另请参见

- SYSIQJOINIDX 系统视图（第 558 页）

SYSIQJOINIXCOLUMN 系统视图

以可读格式显示 **ISYSIQJOINIXCOLUMN** 中的组信息。**SYSIQJOINIXCOLUMN** 视图中的每一行均描述了一个 IQ 连接索引。

```
ALTER VIEW "SYS"."SYSIQJOINIXCOLUMN"
as select * from SYS.ISYSIQJOINIXCOLUMN
```

列名	列类型	描述
joinindex_id	unsigned bigint	对应于 SYSIQJOINIDX 中的连接索引值。
left_table_id	unsigned int	对应于形成连接操作左侧的 SYSTAB 中的表值。
left_column_id	unsigned int	对应于作为连接左侧一部分的 SYSTABCOL 中的列值。
join_type	char(4)	当前唯一受支持的值是 “=”。
right_table_id	unsigned int	对应于形成连接操作右侧的 SYSTAB 中的表值。
right_column_id	unsigned int	对应于作为连接右侧一部分的 SYSTABCOL 中的列值。
order_num	unsigned int	供内部使用。
left_order_num	unsigned int	供内部使用。
right_order_num	unsigned int	供内部使用。
key_type	char(8)	定义键的连接类型。“NATURAL”是自然连接，“KEY”是键连接，“ON”是左外/右外/完整连接。

列名	列类型	描述
coalesce	char(1)	尚未使用。

基础系统表的约束：

主键 (joinindex_id、 left_table_id、 left_column_id、 right_table_id、 right_column_id)

外键 (joinindex_id) 引用 SYS.ISYSIQJOINIDX

外键 (left_table_id、 column_id) 引用 SYS.ISYSTABCOL

外键 (right_table_id、 column_id) 引用 SYS.ISYSTABCOL

SYSIQJOINIXTABLE 系统视图

以可读格式显示 ISYSIQJOINIXTABLE 中的组信息。SYSIQJOINIXTABLE 视图中的每一行均描述了一个 IQ 连接索引。

```
ALTER VIEW "SYS"."SYSIQJOINIXTABLE"
as select * from SYS.ISYSIQJOINIXTABLE
```

列名	列类型	描述
table_id	unsigned int	对应于包括在连接操作中的 SYSTAB 中的表值。
joinindex_id	unsigned bigint	对应于 SYSIQJOINIDX 中的连接索引值。
active	unsigned int	定义在连接索引中表的使用次数。

基础系统表的约束：

主键 (table_id、 joinindex_id)

外键 (table_id) 引用 SYS.ISYSTAB

外键 (joinindex_id) 引用 SYS.ISYSIQJOINIDX

SYSIQLOGICALSERVER 系统视图

显示 ISYSIQLOGICALSERVER 表中的信息，该表存储逻辑服务器以及逻辑服务器和关联的逻辑服务器策略信息之间的对应关系。

请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用” > “系统视图” > “SYSIQLOGICALSERVER 统一视图”。

SYSIQLOGINPOLICYLSINFO 系统视图

显示 ISYSIQLOGINPOLICYLSINFO 表中的组信息，该表存储登录策略逻辑服务器分配信息。

请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用” > “系统视图” > “SYSIQLOGINPOLICYLSINFO 统一视图”。

SYSIQLSLOGINPOLICYOPTION 系统视图

显示 ISYSIQLOGINPOLICYLSINFO 表中的组信息，该表存储登录策略逻辑服务器分配信息。

请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用”> “系统视图”> “SYSIQLSLOGINPOLICYOPTION 系统视图”。

SYSIQLSMEMBER 系统视图

显示 ISYSIQLSMEMBER 表中的组信息，该表存储逻辑服务器成员资格信息。

请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用”> “系统视图”> “SYSIQLSMEMBER 系统视图”。

SYSIQLSMEMBERS 统一视图

此视图描述了所有用户定义的逻辑服务器成员资格。

请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用”> “系统视图”> “SYSIQLSMEMBERS 统一视图”。

SYSIQLSLOGINPOLICIES 统一视图

此视图描述了基于登录策略的所有逻辑服务器的分配情况。

请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用”> “系统视图”> “SYSIQLSLOGINPOLICIES 统一视图”。

SYSIQLSPOLICY 系统视图

显示 ISYSIQLSPOLICY 表中的组信息，该表存储逻辑服务器策略。

请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用”> “系统视图”> “SYSIQLSPOLICY 系统视图”。

SYSIQLSPOLICYOPTION 系统视图

显示 ISYSIQLSPOLICYOPTION 表中的组信息，该表存储逻辑服务器策略选项。

请参见《使用 Sybase IQ Multiplex》> “Multiplex 引用”> “系统视图”> “SYSIQLSPOLICYOPTION 系统视图”。

SYSIQMPXSERVER 系统视图

显示便于阅读版的 ISYSIQMPXSERVER 表，该表存储给定 Multiplex 节点的成员资格属性和版本状态数据。

用户必须获得 Multiplex Grid Option 使用许可才能运行辅助节点。请参见《使用 Sybase IQ Multiplex》。

SYSIQOBJECTS ASE 兼容性视图

sysiqobjects 显示与每个系统表、用户表、视图、过程、触发器、事件、连接索引、约束、域 (sysdomain)、域 (sysusertype)、列和索引相对应的一行。此视图由用户 DBO 所有。

另请参见

- 每个 Adaptive Server Enterprise 数据库中的表 (第 608 页)
- SYSCOLUMNS ASE 兼容性视图 (第 537 页)
- SYS COMMENTS ASE 兼容性视图 (第 537 页)
- SYSINDEXES ASE 兼容性视图 (第 552 页)
- SYSIQVINDEXT ASE 兼容性视图 (第 564 页)
- SYSOBJECTS ASE 兼容性视图 (第 569 页)
- SYSTYPES ASE 兼容性视图 (第 601 页)
- SYSUSERS ASE 兼容性视图 (第 606 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

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	每个表都被指派了一个属于主键的唯一编号（表号）。
block_map	hs_blockmapidentity	供内部使用。
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 0。
info_recid_size	unsigned int	尚未使用。始终为 0。
info_location_size	unsigned int	尚未使用。始终为 0。
commit_txn_id	unsigned bigint	供内部使用。
txn_id	unsigned bigint	供内部使用。
join_id	unsigned int	供内部使用。
update_time	timestamp	最后一次修改 IQ 表的日期和时间。

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 564 页）

SYSIQTABCOL 系统视图

以可读格式显示 **ISYSIQTABCOL** 中的组信息。**SYSIQTABCOL** 视图中的每一行均描述了 IQ 表中的一列。

```
ALTER VIEW "SYS"."SYSIQTABCOL"
as select * from SYS.ISYSIQTABCOL
```

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
table_id	unsigned int	表号唯一标识此列所属的表。它对应于 SYSTAB 的 table_id 列。
column_id	unsigned int	每个表从 1 开始为列编号。列号的顺序决定了列在命令 select * from table 中的显示顺序。
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)。
has_original	char(1)	表示连接索引有原始数据 (T/F)。
original_not_null	char(1)	表示具有原始数据的连接索引列是 NOT NULL (T/F)。
original_unique	char(1)	表示具有原始数据的连接索引列是 UNIQUE (T/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 554 页）

SYSIQTABLE 系统视图（不建议使用）

SYSIQTABLE 已由 SYSIQTAB 系统视图取代。

另请参见

- SYSIQTAB 系统视图（第 563 页）

SYSIQVINDEXT 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 Enterprise 数据库中的表（第 608 页）
- SYSCOLUMNS ASE 兼容性视图（第 537 页）
- SYS COMMENTS ASE 兼容性视图（第 537 页）
- SYSINDEXES ASE 兼容性视图（第 552 页）

- SYSIQOBJECTS ASE 兼容性视图 (第 562 页)
- SYSOBJECTS ASE 兼容性视图 (第 569 页)
- SYSTYPES ASE 兼容性视图 (第 601 页)
- SYSUSERS ASE 兼容性视图 (第 606 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。
jar_name	LONG VARCHAR	JAR 文件的名称。
jar_file	LONG VARCHAR	数据库中 JAR 文件的外部文件名。
update_time	TIMESTAMP	上次更新 JAR 文件的时间。

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)	组件的类型。
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。
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	SYSJARCOMPONENT 系统视图中组件的 ID。
update_time	TIMESTAMP	类的上次更新时间。

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.ISSYSJARCOMPONENT
(component_id)
```

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。作为安全措施，只有具有 DBA 权限的用户才能查看该视图的内容。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.ISSYSUSER (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。

列名	数据类型	说明
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。每个有效的 Adaptive Server Enterprise 用户帐户在 syslogins 中都有对应的一行。

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。
状态	TINYINT	对象的状态。值包括： • 1 (有效) - 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]。 • 2 (无效) - 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。 • 4 (禁用) - 该对象已被用户显式禁用，例如使用 ALTER TABLE...DISABLE VIEW DEPENDENCIES 语句。
object_type	TINYINT	对象的类型。
creation_time	TIMESTAMP	对象创建的日期和时间。
object_type_str	CHAR(128)	对象的类型。

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 Enterprise 数据库中的表 (第 608 页)
- SYSCOLUMNS ASE 兼容性视图 (第 537 页)

- SYSCOMMENTS ASE 兼容性视图 （第 537 页）
- SYSINDEXES ASE 兼容性视图 （第 552 页）
- SYSIQOBJECTS ASE 兼容性视图 （第 562 页）
- SYSIQVINDEXT ASE 兼容性视图 （第 564 页）
- SYSTYPES ASE 兼容性视图 （第 601 页）
- SYSUSERS ASE 兼容性视图 （第 606 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 用户 ID 的设置定义了缺省设置，以用于没有自己的设置的用户。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。

列名	数据类型	说明
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 视图中的每一行均描述了数据库中的一个分区对象（表或索引）。

```
ALTER VIEW "SYS"."SYSPARTITION"  
as select * from SYS.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。
partition_values	long varchar	包含此范围分区的上限。
位置	unsigned int	分区的序号。
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

SYSPARTITIONSCHEME 系统视图

以可读格式显示 ISYSPARTITIONSCHEME 中的组信息。

SYSPARTITIONSCHEME 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"."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）都被指派了一个唯一编号
partition_method	tinyint	此表的分区方法。有效 值： 1 – 表示范围 Sybase IQ 仅支持范围分区。
subpartition_method	tinyint	留作将来使用。

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。

列名	数据类型	说明
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。
深度	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)

SYSROCAUTH 统一视图

SYSROCAUTH 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"."SYSROCAUTH"( grantee,
    creator,procname )
as select u1.user_name,u2.user_name,p.proc_name
    from SYS.ISYSROCPERM as p
        join SYS.ISYSROCPERM 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
```

SYSROCPERURE 系统视图

SYSROCPERURE 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 ISYSROCPERURE。

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。

列名	数据类型	说明
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.ISYSSERVER (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)

SYSPROCARM 系统视图

SYSPROCARM 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 ISYSPROCARM。

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。

列名	数据类型	说明
parm_type	SMALLINT	参数类型将为下列类型之一： • 0 - 普通参数（变量） • 1 - 结果变量 - 与返回结果集的过程一起使用 • 2 - SQLSTATE 错误值 • 3 - SQLCODE 错误值 • 4 - 函数的返回值
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, parmname, parm_id, parmtime, parmmode, parmdomain,
    length, scale, "default", user_type )
```

```
as select
up.user_name,p.proc_name,pp.parm_name,pp.parm_id,pp.parm_type,
  if pp.parm_mode_in = 'Y' and pp.parm_mode_out = 'N' then 'IN'
  else if pp.parm_mode_in = 'N' and pp.parm_mode_out = 'Y' then
'OUT'
    else 'INOUT'
  endif
endif,dom.domain_name,pp.width,pp.scale,pp."default",ut.type_name
from SYS.SYSPROCPARM 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
```

SYSPROCPERM 系统视图

SYSPROCPERM 系统视图的每行都描述一个被授权执行过程的用户。只有得到授权的用户才能执行过程。该视图的基础系统表为 ISYSPROCPERM。

列名	数据类型	说明
proc_id	UNSIGNED INT	过程号唯一标识被授予权限的过程。
grantee	UNSIGNED INT	接收权限的用户的用户号。

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)
```

SYSPROCS 统一视图

SYSPROCS 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"."SYSPROCS"( 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.ISYSSERVER (srvid)

SYSPUBLICATION 系统视图

SYSPUBLICATION 系统视图中的每一行都描述了一个发布。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。
类型	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： • 0 (日志扫描) - 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。 • 1 (脚本式上载) - 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 ISYSSYNCSRIPT 系统表中。 • 2 (仅下载) - 这是一个仅下载发布；不上载任何数据。

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了一个发布。

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)
```

SYSREMARK 系统视图

SYSREMARK 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。
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

SYSREMOTEOPTION 系统视图

SYSREMOTEOPTION 系统视图的每一行都描述一个 SQL Remote 消息链接参数的值。该视图的基础系统表为 ISYSREMOTEOPTION。

此视图中的某些列包含潜在敏感数据。因此，只有具有 DBA 授权的用户才能访问此视图。SYSREMOTEOPTION2 视图提供了对此视图中的数据（除潜在敏感的列外）的公共访问权限。

SYSREMOTEOPTION 视图为 SQL Anywhere 系统视图。请参见“SQL Anywhere Server - SQL 参考” > “视图” > “系统视图” > “SYSREMOTEOPTION 系统视图”。

注意： 此参考指向 SQL Anywhere 文档。

SYSREMOTEOPTION2 统一视图

以一种可读性更强的形式显示不包含敏感数据的 SYSREMOTEOPTION 和 SYSREMOTEOPTIONTYPE 中的列。

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"."SYSREMOTEOPTION2"
as select ISYSREMOTEOPTION.option_id,
        ISYSREMOTEOPTION.user_id,
        SYS.HIDE_FROM_NON_DBA(ISYSREMOTEOPTION.setting) as setting
from SYS.ISYSREMOTEOPTION
```

SYSREMOTOPTIONS 统一视图

SYSREMOTOPTIONS 视图的每一行都描述了消息链接参数的值。该视图中的一些列包含可能比较敏感的数据。为此，仅限具有 DBA 权限的用户能够访问该视图。SYSREMOTOPTION2 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"."SYSREMOTOPTIONS"
as select srt.type_name,
        sup.user_name,
        srot."option",
        SYS.HIDE_FROM_NON_DBA(sro.setting) as setting
from SYS.ISYSREMOTETYPE as srt
    ,SYS.ISYSREMOTOPTIONTYPE as srot
    ,SYS.ISYSREMOTOPTION 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
```

SYSREMOTOPTIONTYPE 系统视图

SYSREMOTOPTIONTYPE 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 ISYSREMOTOPTIONTYPE。

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.ISYSREMOTETYPE (type_id)
```

SYSREMOTETYPE 系统视图

SYSREMOTETYPE 系统视图包含有关 SQL Remote 的信息。该视图的基础系统表为 ISYSREMOTETYPE。

列名	数据类型	说明
type_id	SMALLINT	标识使用 SQL Remote 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。

列名	数据类型	说明
type_name	CHAR(128)	SQL Remote 所支持的消息系统的名称。
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)

SYSRE MOTETYPES 统一视图

SYSRE MOTETYPES 视图的每一行都描述了消息类型之一，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"."SYSRE MOTETYPES"
as select rt.type_id,rt.type_name,rt.publisher_address,rm.remarks
from SYS.ISYSRE MOTETYPE as rt
left outer join SYS.ISYSRE MARK as rm on(rt.object_id =
rm.object_id)
```

SYSRE MOTEUSER 系统视图

SYSRE MOTEUSER 系统视图的每一行都描述一个具有 REMOTE 权限的用户 ID（预订者），以及发送自/至该用户的消息状态。该视图的基础系统表为 ISYSRE MOTEUSER。

列名	数据类型	说明
user_id	UNSIGNED INT	具有 REMOTE 权限的用户的用户号。
consolidate	CHAR(1)	指出是授予了用户 CONSOLIDATE 权限 (Y) 还是 REMOTE 权限 (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。
频率	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。

列名	数据类型	说明
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时间。
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	已发送的消息数量。
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。

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.ISYSREMOETYPE (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 权限的用户 ID（预订者），以及发送自/至该用户的消息状态。

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(a.time_sent,60*hour(a.send_time)
+minute(seconds(a.send_time,59))))
from SYS.ISYSREMOTEUSER as a where a.frequency = 'P'
```

```

        and a.send_time = r.send_time)
    endif
    else if current date+r.send_time
        > coalesce(r.time_sent,current timestamp) then
        current date+r.send_time else current date+r.send_time+1
    endif
endif
    endif endif) as next_send,
    r.log_send,r.time_sent,r.log_sent,r.confirm_sent,r.send_count,
    r.resend_count,r.time_received,r.log_received,
    r.confirm_received,r.receive_count,r.rereceive_count
from SYS.ISYSREMOTEUSER as r
    join SYS.ISYSUSER as u on(u.user_id = r.user_id)
    join SYS.ISYSREMOETYPE as t on(t.type_id = r.type_id)

```

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。

SYSSCHEDULE 视图为 SQL Anywhere 系统视图。请参见“SQL Anywhere Server - SQL 参考” > “视图” > “系统视图” > “SYSSCHEDULE 系统视图”。

注意： 此参考指向 SQL Anywhere 文档。

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 SYSOBJECT 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.ISYSOBJECT (object_id) MATCH
UNIQUE FULL
```

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 SQL Anywhere 主域 ID。
primary_sa_user_type	SMALLINT	相应的 SQL Anywhere 主用户类型。

基础系统表上的约束

```
PRIMARY KEY (ss_user_type)
```

SYSSUBPARTITIONKEY 系统视图

此视图留待将来使用。Sybase IQ15.4 不支持 子分区。

SYSSUBSCRIPTION 系统视图

SYSSUBSCRIPTION 系统视图中的每一行都描述一个用户 ID（必须有 REMOTE 权限）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。

列名	数据类型	说明
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（必须有 REMOTE 权限）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，仅限具有 DBA 权限的用户能够访问该视图。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。
类型	CHAR(1)	此值始终为 D。
publication_id	UNSIGNED INT	在 SYSPUBLICATION 系统视图中找到的 publication_id。
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。

列名	数据类型	说明
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)	预定的名称。

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。

```
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 视图提供了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。

```

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.ISYPUBLICATION 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。该视图中的一些列包含可能比较敏感的数据。为此，仅限具有 DBA 权限的用户能够访问该视图。

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"."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.ISYPUBLICATION as p
on p.publication_id = s.publication_id

```

SYSSYNCSSCRIPT 系统视图

SYSSYNCSSCRIPT 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 SYSSYNCSSCRIPTS 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 ISYSSYNCSSCRIPT。

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。
类型	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 SYSSYNCSRIPT 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.ISYSSYNCSRIPT 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。

```
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.SYSSYNCUSERS as u
where s.site_name = u.site_name)

```

SYSSYNCUSERS 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。

```

ALTER VIEW "SYS"."SYSSYNCUSERS"
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。该数值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。

列名	数据类型	说明
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时间。该列仅在检查点时间更新。
last_modified_tsn	UNSIGNED BIGINT	分配给修改表的事务的序列号。
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_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。
table_type	TINYINT	表或视图的类型。值包括： • 1 - 基表 • 2 - 实例化视图 • 3 - 全局临时表 • 4 - 本地临时表 • 5 - 文本索引基表 • 6 - 文本索引全局临时表 • 21 - 视图
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： • 1 - 本地服务器 • 3 - 远程服务器
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。

列名	数据类型	说明
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。
table_type_str	CHAR(9)	table_type 的可读值。值包括: • BASE - 基表 • MAT VIEW - 实例化视图 • GBL TEMP - 全局临时表 • VIEW - 视图

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 )
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
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。
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	UNSIGNED INT	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。
列名	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。
列类型	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。

列名	数据类型	说明
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。
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。

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

GRANT 语句所赋予的权限存储在 SYSTABLEPERM 系统视图中。该视图中的每一行都对应于一个表、一个授予权限的用户 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	对其应用了权限的表或视图的表号。
grantee	UNSIGNED INT	接收权限的用户 ID 的用户号。
grantor	UNSIGNED INT	授予权限的用户 ID 的用户号。
selectauth	CHAR(1)	指出是否已授予 SELECT 权限。可能的值为 Y、N 或 G。有关这些值的含义的更多信息，请参见下面的注释区域。

列名	数据类型	说明
insertauth	CHAR(1)	指出是否已授予 INSERT 权限。可能的值为 Y、N 或 G。有关这些值的含义的更多信息，请参见下面的注释区域。
deleteauth	CHAR(1)	指出是否已授予 DELETE 权限。可能的值为 Y、N 或 G。有关这些值的含义的更多信息，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 权限。可能的值为 Y、N 或 G。有关这些值的含义的更多信息，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 权限。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 权限。
alterauth	CHAR(1)	指出是否已授予 ALTER 权限。可能的值为 Y、N 或 G。有关这些值的含义的更多信息，请参见下面的注释区域。
referenceauth	CHAR(1)	指出是否已授予 REFERENCE 权限。可能的值为 Y、N 或 G。有关这些值的含义的更多信息，请参见下面的注释区域。

注释

有多种可以授予的权限。每种权限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权限。
- **Y** - 是，授予者已授予被授予者该权限。
- **G** - 被授予者已被授予该权限并能将同样的权限授予其他用户。

注意： 另一个授予者可能已授予被授予者操作同一个表的权限。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。

列名	数据类型	说明
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 GENERIC ，值为 0；对于 NGRAM ，值为 1。 GENERIC 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。 NGRAM 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 min_term_length 的术语。 MINIMUM TERM LENGTH 设置仅对 GENERIC 术语断开器有意义。对于 NGRAM 文本索引，将忽略该设置。
max_term_length	TINYINT	对于 GENERIC 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 max_term_length 的术语。 对于 NGRAM 文本索引，该设置是用于断开术语的 n 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 CHAR 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 default_char 创建文本配置对象时使用此列。
nchar_stoplist	LONG NVARCHAR	在 NCHAR 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 default_nchar 创建文本配置对象时使用此列。
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。
状态	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： • 1 - MANUAL • 2 - AUTO • 3 - IMMEDIATE
refresh_interval	UNSIGNED INT	AUTO REFRESH 间隔（以分钟为单位）。
last_refresh	TIMESTAMP	上次刷新的时间。

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.ISYSTEXTCONFIG (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 ISYSTEXTIDXTAB。

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.ISYSTEXTIDX
(index_id, sequence)
FOREIGN KEY (table_id) references SYS.ISYSTAB (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 ISYSTRIGGER。

列名	数据类型	说明
trigger_id	UNSIGNED INT	SYSTRIGGER 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。
event	CHAR(1)	此操作将导致触发器的触发。 • A - INSERT, DELETE • B - INSERT, UPDATE • C - UPDATE COLUMNS • D - DELETE • E - DELETE, UPDATE • I - INSERT • M - INSERT, DELETE, UPDATE • U - UPDATE

列名	数据类型	说明
trigger_time	CHAR(1)	触发器相对于事件触发时的时间。 • A - AFTER (行级触发器) • B - BEFORE (行级触发器) • I - INSTEAD OF (行级触发器) • K - INSTEAD OF (语句级触发器) • R - RESOLVE • S - AFTER (语句级触发器)
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。 • C - CASCADE • D - SET DEFAULT • N - SET NULL • R - RESTRICT
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。
source	LONG VARCHAR	触发器的 SQL 源。该值存储在 ISYSSOURCE 系统表中。

基础系统表上的约束

PRIMARY KEY (trigger_id)

FOREIGN KEY (object_id) references SYS.ISYSOBJECT (object_id) MATCH UNIQUE FULL

FOREIGN KEY (table_id) references SYS.ISYSTAB (table_id)

FOREIGN KEY fkey_index (foreign_table_id, foreign_key_id) references SYS.ISYSIDX (table_id, index_id)

```
UNIQUE Index (table_id, event, trigger_time, trigger_order)
```

```
UNIQUE Index (trigger_name, table_id)
```

```
UNIQUE Index (table_id, foreign_table_id, foreign_key_id, event)
```

SYSTRIGGERS 统一视图

SYSTRIGGERS 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。

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"."SYSTRIGGERS"( owner,
    trigname,tname,event,trigtime,trigdefn )
as select u.user_name,trig.trigger_name,tab.table_name,
    if trig.event = 'I' then 'INSERT'
    else if trig.event = 'U' then 'UPDATE'
    else if trig.event = 'C' then 'UPDATE'
    else if trig.event = 'D' then 'DELETE'
    else if trig.event = 'A' then 'INSERT,DELETE'
    else if trig.event = 'B' then 'INSERT,UPDATE'
    else if trig.event = 'E' then 'DELETE,UPDATE'
    else 'INSERT,DELETE,UPDATE'
    endif
    endif
    endif
    endif
    endif
    endif,if trig.trigger_time = 'B' or trig.trigger_time = 'P' then
'BEFORE'
    else if trig.trigger_time = 'A' or trig.trigger_time = 'S' then
'AFter'
    else if trig.trigger_time = 'R' then 'RESOLVE'
    else 'INSTEAD OF'
    endif
    endif
    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
```

SYSTPEMAP 系统视图

SYSTPEMAP 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 ISYSTPEMAP。

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 SQL Anywhere domain_id。

列名	数据类型	说明
sa_user_type	SMALLINT	包含相应的 SQL Anywhere 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 Enterprise 数据库中的表（第 608 页）
- SYSCOLUMNS ASE 兼容性视图（第 537 页）
- SYS COMMENTS ASE 兼容性视图（第 537 页）
- SYSINDEXES ASE 兼容性视图（第 552 页）
- SYSIQOBJECTS ASE 兼容性视图（第 562 页）
- SYSIQVINDEXT ASE 兼容性视图（第 564 页）
- SYSOBJECTS ASE 兼容性视图（第 569 页）
- SYSUSERS ASE 兼容性视图（第 606 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。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。
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。
password_creation_time	TIMESTAMP	用户口令的创建时间。
failed_login_attempts	UNSIGNED INT	用户在帐户被锁定前可以登录失败的次数。

列名	数据类型	说明
last_login_time	TIMESTAMP	用户上次登录的时间。

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)

提供 SYSUSERAUTH 视图的目的是为了与旧版本的软件兼容。改用 SYSUSERAUTHORITY 系统视图。

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，因此该视图没有 PUBLIC 选择权限。

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"."SYSUSERAUTH"( name,
    password,resourceauth,dbaauth,scheduleauth,user_group )
as select
SYSUSERPERM.user_name,SYSUSERPERM.password,SYSUSERPERM.resourceauth
,SYSUSERPERM.dbaauth,SYSUSERPERM.scheduleauth,SYSUSERPERM.user_group
from SYS.SYSUSERPERM
```

SYSUSERAUTHORITY 系统视图

SYSUSERAUTHORITY 系统视图的每一行都描述授予一个用户 ID 的权限。该视图的基础系统表为 ISYSUSERAUTHORITY。

列名	数据类型	说明
user_id	UNSIGNED INT	权限所属的用户的 ID。
auth	VARCHAR(20)	授予用户的权限。

基础系统表上的约束

```
PRIMARY KEY (user_id, auth)

FOREIGN KEY (user_id) references SYS.ISYSUSER (user_id)
```

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的软件兼容。

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
错误	INTEGER	错误情况的唯一标识号。
uid	UNSIGNED INT	定义消息的用户号。
说明	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）

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。应更改您的应用程序改用 SYSUSERAUTHORITY 系统视图。

SYSUSERPERM 视图的每一行都描述一个用户 ID。

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 exists(select * from SYS.ISYSUSERAUTHORITY
        where ISYSUSERAUTHORITY.user_id = b.user_id and
        ISYSUSERAUTHORITY.auth = 'RESOURCE') then
        'Y' else 'N' endif as resourceauth,
        if exists(select * from SYS.ISYSUSERAUTHORITY
        where ISYSUSERAUTHORITY.user_id = b.user_id and
        ISYSUSERAUTHORITY.auth = 'DBA') then
        'Y' else 'N' endif as dbaauth,
        'N' as scheduleauth,
        if exists(select * from SYS.ISYSUSERAUTHORITY
        where ISYSUSERAUTHORITY.user_id = b.user_id and
        ISYSUSERAUTHORITY.auth = 'PUBLISH') then
        'Y' else 'N' endif as publishauth,
        if exists(select * from SYS.ISYSUSERAUTHORITY
        where ISYSUSERAUTHORITY.user_id = b.user_id and
        ISYSUSERAUTHORITY.auth = 'REMOTE DBA') then
        'Y' else 'N' endif as remotedbauth,
        if exists(select * from SYS.ISYSUSERAUTHORITY
        where ISYSUSERAUTHORITY.user_id = b.user_id and
        ISYSUSERAUTHORITY.auth = 'GROUP') 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)

```

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。应更改您的应用程序改用 SYSUSERAUTHORITY 和 SYSUSER 系统视图。

与 SYSUSERPERM 视图相似，SYSUSERPERMS 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 Enterprise 数据库中的表 (第 608 页)
- `SYSCOLUMNS ASE` 兼容性视图 (第 537 页)
- `SYSCOMMENTS ASE` 兼容性视图 (第 537 页)
- `SYSINDEXES ASE` 兼容性视图 (第 552 页)
- `SYSIQOBJECTS ASE` 兼容性视图 (第 562 页)
- `SYSIQVINDEXT ASE` 兼容性视图 (第 564 页)
- `SYSOBJECTS ASE` 兼容性视图 (第 569 页)
- `SYSTYPES ASE` 兼容性视图 (第 601 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。此视图的基础系统表为 `ISYSVIEW`。

您也可以对物化视图中的可读格式的信息使用 `sa_materialized_view_info` 系统过程。仅 `IQ Catalog` 存储中的 `SQL Anywhere` 表支持物化视图。

`SYSVIEW` 视图为 `SQL Anywhere` 系统视图。请参见“`SQL Anywhere Server - SQL 参考`”>“`视图`”>“`系统视图`”>“`SYSVIEW 系统视图`”。

注意： 此参考指向 `SQL Anywhere` 文档。

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）。
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。
已启用	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 Enterprise 和 Sybase IQ 有不同的系统目录，这反映了两个产品的不同用法。

在 Adaptive Server Enterprise 中，有一个单独的 master 数据库，其中包含的一组系统表含有应用于服务器上所有数据库的信息。master 数据库可能包括许多数据库，而且每个数据库都有另外的系统表与之关联。

在 Sybase IQ 中，每个数据库独立存在，并包含自己的系统表。数据库集合没有包含系统信息的 master 数据库。每台服务器都可以同时运行多个数据库，根据需要动态装载和卸载每个数据库。

Adaptive Server Enterprise 和 Sybase IQ 系统目录有所不同。Adaptive Server Enterprise 系统表和视图由特殊用户 dbo 拥有，并部分存在于 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；Sybase IQ 系统表和视图由特殊用户 SYS 拥有，并单独存在于每个数据库中。

为帮助准备兼容应用程序，Sybase IQ 提供了一组由特殊用户 dbo 拥有的视图，它们对应于 Adaptive Server Enterprise 系统表和视图。如果体系结构差异使得特定 Adaptive Server Enterprise 表或视图的内容在 Sybase IQ 上下文中无意义，则视图为空，仅包含列名称和数据类型。

这些主题列出了 Adaptive Server Enterprise 系统表及其在 Sybase IQ 系统目录中的实现。在每个 DBMS 中，dbo 是所有表的所有者。

另请参见

- ASE T-SQL 兼容性视图（第 530 页）

每个 Adaptive Server Enterprise 数据库中的表

并非所有 Adaptive Server Enterprise 系统表均在 Sybase IQ 系统目录中实现。

表 212. 每个 ASE 数据库中的表

表名称	描述	是否包含数据?	是否受 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 Sybase IQ 中，使用所有者名称 dbo（即 dbo.syscolumns）进行查询。	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

表名称	描述	是否包含数据?	是否受 IQ 支持?
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	每个聚簇或非聚簇索引占一行，每个没有索引的表占一行，每个包含文本或图像数据的表另占一行。在 Sybase IQ 中，使用所有者名称 dbo（即 dbo.sysindexes）进行查询。	是	是
sysiqobjects	每个系统表、用户表、视图、过程、触发器、事件、连接索引、约束、域 (sysdomain)、域 (sysusertype)、列和索引占一行。	是	是
sysiqvindex	每个非 fp iq 索引占一行。	是	是
syskeys	每个主键、外键或公用键在其中都有相应的一行；由用户设置（不是由 Adaptive Server Enterprise 维护的）。	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
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 537 页）
- SYS COMMENTS ASE 兼容性视图（第 537 页）
- SYSINDEXES ASE 兼容性视图（第 552 页）
- SYSIQOBJECTS ASE 兼容性视图（第 562 页）
- SYSIQVINDEXT ASE 兼容性视图（第 564 页）

- SYSOBJECTS ASE 兼容性视图 （第 569 页）
- SYSTYPES ASE 兼容性视图 （第 601 页）
- SYSUSERS ASE 兼容性视图 （第 606 页）

Adaptive Server Enterprise Master 数据库中的表
并非所有 Adaptive Server Enterprise master 数据库表 均在 Sybase IQ 系统目录中实现。

表 213. ASE master 数据库表

表名称	描述	是否包含数据?	是否受 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
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
sysservers	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 Enterprise Sybsecurity 数据库中的表

所有 Adaptive Server Enterprise sybsecurity 数据库表 均不在 Sybase IQ 系统目录中实现。

表 214. ASE sybsecurity 数据库表

表名称	描述	是否包含数据?	是否受 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

本节中的主题 旨在简化从其它 Sybase 数据库到 Sybase IQ 的迁移，并作为指南来指导创建 与 Adaptive Server Enterprise 或 SQL Anywhere 兼容的 Sybase IQ 应用程序。

Sybase IQ 的每个新版本中都涉及了兼容性功能。本附录将 Sybase IQ 与 Adaptive Server Enterprise 和 SQL Anywhere 进行了比较。

关于 SQL Anywhere

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，有一些重要区别。不要臆断 Sybase IQ 支持 SQL Anywhere 文档中 描述的功能。

Sybase IQ 文档集指出了许多（但不是全部）不同之处。*Sybase IQ 文档始终优先于 SQL Anywhere 文档。*

Transact-SQL 支持概述

像 SQL Anywhere 一样，Sybase IQ 支持 *Transact-SQL* 中相当大的一部分，Transact-SQL 是 Sybase Adaptive Server Enterprise 所支持的 SQL 术语。

在 Sybase IQ 中提供 Transact-SQL 支持的目标是 提供应用程序可移植性。可以编写许多应用程序、存储过程和批处理文件以与 Adaptive Server Enterprise 和 Sybase IQ 数据库配合使用。

目的是编写与 Adaptive Server Enterprise 和 Sybase IQ 配合使用的应用程序。现有 Adaptive Server Enterprise 应用程序 通常需要进行一些更改才能针对 SQL Anywhere 或 Sybase IQ 数据库运行。

Sybase IQ 中的 Transact-SQL 支持采用以下形式：

- 大多数 SQL 语句 在 Sybase IQ 和 Adaptive Server Enterprise 之间是兼容的。
- 对于一些语句，尤其是在过程和批处理语句中 使用的过程语言中，除早期版本的 Sybase IQ 支持的语法外，还支持单独的 Transact-SQL 语句。对于这些语句，SQL Anywhere 和 Sybase IQ 支持两种 SQL 术语。在本附录中，我们将这两种术语称为 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。

Sybase IQ 支持使用大部分 Transact-SQL 语言元素、函数和语句来处理现有数据。

此外, Sybase IQ 还支持大部分 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许多 (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, Sybase IQ 仍将 Transact-SQL 系统表 作为视图来提供 (在视图中, Sybase IQ 中没有任何意义的表将 不显示任何行)。此外, Sybase IQ 为某些更为常见的管理任务 提供了一组系统过程。

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 体系结构

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 是互补的产品, 其体系结构分别满足不同的需求。

Sybase IQ 是专门为数据仓库应用和分析处理而设计 的高性能决策支持服务器。SQL Anywhere 适于在工作组或 部门服务器需要的管理比较少时 作为个人数据库使用。Adaptive Server Enterprise 适合作为 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。还描述了 Sybase IQ 和 SQL Anywhere 为达到数据库管理的兼容性 而包含的类似于 Adaptive Server Enterprise 的工具。

服务器和数据库

Adaptive Server Enterprise 中的服务器和数据库之间的关系, 与 Sybase IQ 和 SQL Anywhere 中的是不同的。

在 Adaptive Server Enterprise 中, 每个数据库都存在于 服务器内, 且每个服务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, 用户如果拥有相应的权限, 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, 这些系统表包含服务器 上所有数据库共有的信息。

Sybase IQ 中不存在与 Adaptive Server Enterprise master 数据库对应的等效数据库。相反, 每个数据库都是一个独立的实体, 包含其自身的所有系统表。用户可以具有连接到数据库 (而不是服务器) 的权限。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 Sybase IQ 数据库服务器都可以动态地 启动和停止用户可独立连接到的 数据库。Sybase 极力建议您每台 服务器只运行一个 Sybase IQ 数据库。

SQL Anywhere 和 Sybase IQ 通过 Transact-SQL 支持和 Open Server 支持提供了一些工具, 可通过这些工具 采用与 Adaptive Server Enterprise 类似的方法执行一些任务。但是, 这些工具的 确切实现方式是存在一些区别的。

空间分配和设备管理

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 最初和后来 用于管理设备以及分配磁盘空间的模型都是不同的，目的是反映产品的不同用法。

例如：

- 在 Adaptive Server Enterprise 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一个数据库。您可以使用 **ALTER DATABASE** 添加更多空间，也可以使用阈值自动添加。
- 在 Sybase IQ 中，最初通过在 **CREATE DATABASE** 语句中列出的原始设备 来分配空间。您可以使用 **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。Sybase IQ 还可以使用文件系统空间。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 与 Sybase IQ 类似，只不过 初始 **CREATE DATABASE** 语句需要利用单个文件系统文件，而不是一系列原始设备。SQL Anywhere 可以使用名为 **dbinit** 的命令实用程序初始化它的数据库。Sybase IQ 提供用于初始化 IQ 数据库的此实用程序（名为 **iqinit**）的扩展版本。

有关磁盘管理的信息，请参见《系统管理指南第一卷》。

系统表、Catalog 存储和 IQ 存储

IQ 数据库是连接数据存储。

连接存储包括：

- Catalog 存储 包括系统表和存储过程，驻留在与 SQL Anywhere 兼容的一组表中。
- 永久 IQ 存储是 Sybase IQ 表集。表数据存储存储在索引中。
- 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。

Catalog 区分和兼容性功能包括：

- SQL Anywhere 和 Sybase IQ 对 Catalog（表、列 等等）使用与 Adaptive Server Enterprise 不同的 模式。
- SQL Anywhere 和 Sybase IQ 提供了一些兼容性视图，这些视图模拟 Adaptive Server Enterprise 系统表的相关部分，但使用它们时性能会受影响。
- 在 Adaptive Server Enterprise 中，数据库所有者（用户 ID **dbo**）拥有 Catalog 对象。
- 在 SQL Anywhere 和 Sybase IQ 中，系统所有者（用户 ID **SYS**）拥有 Catalog 对象。

注意： **dbo** 用户 ID 拥有 Sybase IQ 提供的与 Adaptive Server Enterprise 兼容的系统视图。

管理角色

与 SQL Anywhere 或 Sybase IQ 相比，Adaptive Server Enterprise 拥有一组更加复杂的管理角色。

虽然可以向 Adaptive Server Enterprise 上的多个登录帐户授予任意角色，而且一个帐户可以拥有多个角色，但 Adaptive Server Enterprise 中有一组不应重复的角色。

Adaptive Server Enterprise 角色	描述
系统管理员	负责与特定应用程序无关的常规管理任务；可以访问任何数据库对象。
系统安全员	负责 Adaptive Server Enterprise 中与安全相关的任务，但对数据库对象不具有特殊的权限。
数据库所有者	对自己拥有的数据库中的对象具有完全权限，可以将用户添加到数据库中并向其它用户授予在数据库中创建对象和执行命令的权限。
数据定义语句	可以向用户授予执行特定的数据定义语句（如 CREATE TABLE 或 CREATE VIEW ）的权限，使用户能够创建数据库对象。
对象所有者	每个数据库对象都有一个所有者，该所有者可以向其他用户授予访问该对象的权限。对象的所有者自动具有该对象的各种权限。

SQL Anywhere/ Sybase IQ 角色	描述
数据库管理员（DBA 授权）	与 Adaptive Server Enterprise 数据库所有者类似，数据库管理员（DBA 授权）对该数据库中的所有对象（SYS 所拥有的对象除外）具有完全权限，并且可以向其他用户授予在该数据库中创建对象和执行命令的权限。缺省的数据库管理员是用户 ID DBA。
RESOURCE 权限	允许用户在数据库中创建任意类型的对象。它取代 Adaptive Server Enterprise 将权限授予单独的 CREATE 语句的模式。
对象所有者	Sybase IQ 拥有对象所有者的方式与 Adaptive Server Enterprise 相同。对象所有者自动拥有对象的所有权限，包括授权权限。

为了无缝访问 Adaptive Server Enterprise 和 Sybase IQ 中保存的数据，应该在数据库中创建具有适当权限（在 Sybase IQ 中为 RESOURCE 权限，或者在 Adaptive Server Enterprise 中为执行单个 **CREATE** 语句的权限）的用户 ID，并利用该用户 ID 创建对象。如果在每个环境中都使用相同的用户 ID，则对象名和限定符可以在这两种数据库中完全相同，从而可确保能进行兼容访问。

数据类型

Adaptive Server Enterprise、SQL Anywhere 和 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 Enterprise、SQL Anywhere 和 Sybase IQ 支持 BIT 数据类型，但有一些区别。

不同之处在于：

- SQL Anywhere 只允许使用 0 或 1。
- Adaptive Server Enterprise 和 Sybase IQ 将整数数据类型隐式转换为 BIT。非零值存储为 1 (TRUE)。

字符数据类型

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 允许使用 CHAR 和 VARCHAR 数据，但每种产品均以不同的方式处理这些类型。

- SQL Anywhere 将所有字符串作为 VARCHAR 处理，即使在用空白填充的数据库中也是如此。
- Adaptive Server Enterprise 和 Sybase IQ 将 CHAR（固定长度）和 VARCHAR（可变长度）数据区别对待。

Adaptive Server Enterprise 剪裁 VARCHAR 值的尾随空白。Sybase IQ 根据数据格式和操作剪裁 VARCHAR 值的尾随空白。

插入到 CHAR 或 VARCHAR 中时：

- SQL Anywhere 允许将整数数据类型插入到 CHAR 或 VARCHAR 中（隐式转换）。
- Adaptive Server Enterprise 和 Sybase IQ 需要显式转换。

按如下方式确定列的最大大小：

- Adaptive Server Enterprise CHAR 和 VARCHAR 取决于逻辑页大小，它们可能是 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 支持最多 32K-1 的 CHAR 和 VARCHAR 数据，以及最多 2GB 的 LONG VARCHAR 数据。

- SQL Anywhere 支持 名称 **LONG VARCHAR** 及其同义词 **TEXT**，而 Adaptive Server Enterprise 仅支持 **TEXT** 名称，不支持 **LONG VARCHAR** 名称。
- Sybase IQ 最多支持 32K-1 字节的 **CHAR** 和 **VARCHAR** 数据。
Sybase IQ 也最多支持 512TB（对于大小为 128KB 的 IQ 页）和 2PB（对于大小为 512KB 的 IQ 页）的 **LONGVARCHAR** 数据。有关 Sybase IQ 中 **LONG VARCHAR** 数据类型 的信息，请参见《Sybase IQ 中的非结构化数据分析》。
- Adaptive Server Enterprise 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 Sybase IQ 支持 **CHAR** 和 **VARCHAR** 数据类型的 Unicode，而不 支持作为单独数据类型的 Unicode。
- 为了使 Sybase IQ 和 Adaptive Server Enterprise 相互兼容，务必指定字符数据类型 的长度。

另请参见

- 字符数据类型（第 65 页）

二进制数据类型

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间的二进制数据类型支持有 所不同。

表 215. 二进制数据类型支持的大小

数据类型	Adaptive Server Enterprise	SQL Anywhere	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 Sybase IQ 中 **LONG BINARY** 数据类型的信息，请参见《Sybase IQ 中的非结 构化数据分析》。此功能需要单独的许可证。

投影时，Adaptive Server Enterprise 和 SQL Anywhere 以不同的 方式显示二进制数据：

- Sybase IQ 支持 Adaptive Server Enterprise 和 SQL Anywhere 显示格式。
- 如果在 **BINARY** 字段中输入“123”，则 SQL Anywhere 显示格式为按字节显示， 即“123”；Adaptive Server Enterprise 显示格式为按半位元组显示，即 “0x616263”。

另请参见

- 二进制数据类型 (第 72 页)
- NEWID 函数 [杂项] (第 223 页)
- STRTOUUID 函数 [字符串] (第 290 页)
- UUIDTOSTR 函数 [字符串] (第 311 页)
- 字符数据类型 (第 65 页)

日期、时间、日期时间和时间戳数据类型

虽然 Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 都支持某种形式的日期和时间数据，但存在一些区别。

- SQL Anywhere 和 Sybase IQ 支持 4 字节的日期和时间数据类型。
- Adaptive Server Enterprise 支持 8 字节的 **datetime** 类型以及作为以二进制形式 (8) 实现的用户定义数据类型 (域) 的 **timestamp**。
- SQL Anywhere 和 Sybase IQ 支持 8 字节的 **timestamp** 类型和作为时间戳实现的 8 字节的 **datetime** 域。SQL Anywhere/Sybase IQ **datetime** 数据类型的精度为毫秒，这不同于 Adaptive Server Enterprise。

日期的显示格式具有不同的缺省值：

- Adaptive Server Enterprise 缺省为以 “MMM-DD-YYYY” 格式显示日期，但可以通过设置选项来对此格式进行更改。
- SQL Anywhere 和 Sybase IQ 缺省为 ISO “YYYY-MM-DD” 格式，但您可以通过设置选项来对此进行更改。

时间转换如下：

- Adaptive Server Enterprise 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 Sybase IQ 都会以相同的方式转换时间。

将时间插入到 DATETIME 列中时：

- Adaptive Server Enterprise 和 Sybase IQ 缺省提供 “1st January 1900”。
- SQL Anywhere 缺省提供当前日期。

使用 **INSERT...LOCATION** 将从 Adaptive Server Enterprise 数据库检索的 **TIME** 和 **DATETIME** 值插入到 Sybase IQ 表中的 **DATETIME** 列时，这些值会发生变化。**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 Enterprise 数据库的表列中：

```
2004-11-08 10:37:22.823
```

当使用 **INSERT...LOCATION** 在 Sybase IQ 表中检索并存储该值时，它会变为：

```
2004-11-08 10:37:22.823333
```

ASE 中的日期时间值和时间值的兼容性

使用 **INSERT...LOCATION** 从 Adaptive Server Enterprise 数据库检索的 DATETIME 或 TIME 值可能会由于 Open Client 的日期时间精度而不同。

例如，Adaptive Server Enterprise 数据库中的 DATETIME 值是 “2004-11-08 10:37:22.823”，而使用 **INSERT...LOCATION** 检索到的值是 “2004-11-08 10:37:22.823333”。

BIGTIME 和 BIGDATETIME 支持

Sybase IQ 支持 Adaptive Server Enterprise (ASE) 数据类型 BIGTIME 和 BIGDATETIME 用于组件集成服务 (CIS) 和 **INSERT...LOCATION**。

- 具有 Adaptive Server Enterprise 的组件集成服务 - aseodbc 服务器类代理表，这些表映射到包含数据类型为 BIGTIME 和 BIGDATETIME 的列的 Adaptive Server Enterprise 表。

创建映射到 Adaptive Server Enterprise 表的代理表时，如果未指定映射，则缺省情况下，BIGDATETIME 列将映射到 TIMESTAMP 列。缺省情况下，BIGTIME 列将映射到 TIME 列。

asejdbc 服务器类不支持 BIGTIME 和 BIGDATETIME 数据类型。

- **INSERT...LOCATION - INSERT...LOCATION** 命令，该命令将数据从包含数据类型为 BIGTIME 和 BIGDATETIME 的列的 Adaptive Server Enterprise 表装载到 Sybase IQ 表中。

Sybase IQ 将 Adaptive Server Enterprise 数据类型 BIGTIME 插入到 Sybase IQ 数据类型 TIME 中。

Sybase IQ 将 Adaptive Server Enterprise 数据类型 BIGDATETIME 插入到 Sybase IQ 数据类型 DATETIME、DATE、TIME 和 TIMESTAMP 中。

数值数据类型

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 具有不同的缺省精度和标度。

- 在 Adaptive Server Enterprise 中，缺省值是精度为 18，标度为 0。
- 在 SQL Anywhere 中，缺省值是精度为 30，标度为 6。
- 在 Sybase IQ 中，缺省值是精度为 126，标度为 38。由于这些缺省值对于 TDS 和一些客户端工具而言太大，请务必为 Sybase IQ 的精确数值类型指定精度和标度。

文本数据类型

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间对 TEXT 数据的支持有所不同。

- Adaptive Server Enterprise 支持最多 2GB 的 LONG VARBINARY（在 SQL Anywhere 中为 LONG BINARY）和 TEXT 数据。SQL Anywhere 不支持 LONG VARBINARY

作为列类型，而使用 **LONG BINARY** 来实现相同的目的。SQL Anywhere 支持最多 2GB 的 **LONG BINARY** 和 **TEXT** 数据。

- Sybase IQ 支持最多 32KB - 1 的 **VARCHAR** 数据。Sybase IQ 还支持最多 512TB（对于大小为 128KB 的 IQ 页）和 2PB（对于大小为 512KB 的 IQ 页）的 **LONGVARCHAR** 数据。有关 Sybase IQ 中 **LONG VARCHAR** 数据类型的信息，请参见《Sybase IQ 中的非结构化数据分析》。

图像数据类型

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间对 **IMAGE** 数据的支持有所不同。

- Adaptive Server Enterprise 和 SQL Anywhere 支持最多 2GB 的 **IMAGE** 数据。
- Sybase IQ 支持最多 512TB（对于大小为 128KB 的 IQ 页）和 2PB（对于大小为 512KB 的 IQ 页）的 **LONGBINARY** 数据。有关 Sybase IQ 中 **LONG BINARY** 数据类型的信息，请参见《Sybase IQ 中的非结构化数据分析》。

Java 数据类型

Adaptive Server Enterprise 允许数据库中使用 Java 数据类型，而 SQL Anywhere 和 Sybase IQ 则不允许。

Data Definition Language（数据定义语言）

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 在如何创建数据库和数据库对象方面存在差异。

从 Sybase Central 创建与 Transact-SQL 兼容的数据库

使用模拟 Adaptive Server Enterprise 的区分大小写设置来创建与 Transact-SQL 兼容的数据库。

1. 在创建数据库向导中，访问“指定归类序列”页。
2. 将“字符串比较区分大小写”设置为 ON。这样将模拟 Adaptive Server Enterprise。
3. 将“口令区分大小写”设置为 ON。

这些区分大小写设置将模拟 Adaptive Server Enterprise。

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。

在创建数据库时，应确定 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 Sybase IQ 数据库在比较时区分大小写。

Adaptive Server Enterprise 是否区分大小写取决于在 Adaptive Server Enterprise 系统上安装的排序顺序。通过重新配置 Adaptive Server Enterprise 的排序顺序，可以更改单字节字符集的区分大小写设置。

标识符的区分大小写

标识符包括表名、列名、用户 ID 等。

Sybase IQ 不支持区分大小写的标识符。在 Adaptive Server Enterprise 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 Enterprise 中，用户定义数据类型名称是区分大小写的。在 Sybase IQ 中，它们不区分大小写。

用户 ID 和口令的区分大小写

口令的区分大小写的处理方式与其它标识符有所不同。

在 Sybase IQ 和 SQL Anywhere 中，新建数据库中所有的口令都区分大小写，无论数据库本身是否区分大小写。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，Sybase IQ 和 SQL Anywhere 按照如下方式来确定口令的区分大小写特性：

- 如果口令最初是在不区分大小写的数据库中输入的，则口令仍然不区分大小写。
- 如果口令最初是在区分大小写的数据库中输入的，则大写和大小写混合的口令仍然区分大小写。如果口令是以全部小写的格式输入的，则该口令会变为不区分大小写。
- 对现有口令和新口令的更改都是区分大小写的。

在 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。与 SQL Anywhere 和 Sybase IQ 相比，Adaptive Server Enterprise 中的一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 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` 都可以创建 名为 `employee` 的表；唯一性 由完全限定名称 `user1.employee` 和 `user2.employee` 提供。
- 对于 Adaptive Server Enterprise，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，索引由创建它们时所在的表的所有者拥有。索引名称在给定表上必须是唯一的，但任何两个表都可以具有同名的索引，甚至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

在 Sybase IQ 中，可以使用 **ALTER INDEX** 语句重命名显式创建的索引、索引的外键角色名称，以及外键。在 SQL Anywhere 中，可以使用 **ALTER INDEX** 语句重命名索引、外键角色名称和外键。Adaptive Server Enterprise 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。

列中的空值

若要实现 NULL 的兼容处理：

- SQL Anywhere 和 Sybase IQ 假设 列可以为空值，但在列定义中指定非空值则例外。通过将数据库选项 **ALLOW_NULLS_BY_DEFAULT** 设置为与 Transact-SQL 兼容的设置 **OFF**，可以更改此行为。
- SQL Anywhere 仅假设 **BIT** 列不能为 **NULL**。
- Adaptive Server Enterprise 假设列不能为 **NULL**，但如果指定 **NULL** 则除外。

检查约束

Sybase IQ 强制对基表、全局临时表和本地临时表，以及用户定义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 可在 **LOAD** 语句回滚前 发生的违规数量。

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 Sybase IQ 表组成的检查约束。首次在 **LOAD**、**INSERT** 或 **UPDATE** 语句中 使用针对其定义检查约束的表时，系统将检测无法评估的约束。Sybase IQ 不允许包含以下项的检查约束：

- 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
- 集合函数
- 对非确定性函数或修改数据的函数的调用

Adaptive Server Enterprise 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。

Sybase IQ 支持用户定义数据类型，这些类型允许将约束封装在数据类型定义中。

参照完整性约束

Sybase IQ 按照《系统管理指南第一卷》中的描述强制参照完整性。

按照如下所示支持用于强制完整性的操作：

- SQL Anywhere 支持所有 ANSI 操作：SET NULL、CASCADE、DEFAULT、RESTRICT。
- Adaptive Server Enterprise 支持这些操作中的两个操作：SET NULL、DEFAULT。

注意：您可以通过使用触发器而不是参照完整性在 Adaptive Server Enterprise 中实现 CASCADE。

- Sybase IQ 仅支持 RESTRICT 操作。
- Sybase IQ 不支持 NOT NULL FOREIGN KEY。
- 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 Enterprise 和 SQL Anywhere 支持指定列的缺省值。
- 只有 SQL Anywhere 支持 DEFAULT UTC TIMESTAMP。
- Sybase IQ 支持指定列的缺省值（特殊值 DEFAULT UTC TIMESTAMP 和 DEFAULT CURRENT UTC TIMESTAMP 除外）。Sybase IQ 还忽略 DEFAULT_TIMESTAMP_INCREMENT 数据库选项的设置。

标识列

标识列支持有如下区别：

- Sybase IQ 支持 IDENTITY 或 DEFAULT AUTOINCREMENT 作为缺省值。Sybase IQ 支持精度和标度为 0 的任何 Numeric 类型的标识列，并且列可以是 NULL。Sybase IQ 标识列必须为正数，并受数据类型的范围限制。Sybase IQ 支持每个表一个标识列，并要求将数据库选项 IDENTITY_INSERT 设置为要显式插入和更新的表名。要删除含 IDENTITY 列的表，则不能为该表设置 IDENTITY_INSERT。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。Sybase IQ 视图不能包含 IDENTITY/DEFAULTAUTOINCREMENT 列。
- SQL Anywhere 支持 AUTOINCREMENT 缺省值。SQL Anywhere 支持标度和精度为任意允许值的任意 Numeric 类型的标识列。标识列值可以为正数、负数或零，这要受数据类型的范围限制。SQL Anywhere 支持每个表有任意数量的标识列，不

要求使用 `identity_insert` 来执行显式插入、删除和更新。添加标识列时，表必须为空。可以将 SQL Anywhere 标识列更改为非标识列，反之亦然。您可以在 SQL Anywhere 视图添加或删除 `AUTOINCREMENT` 列。

- Adaptive Server Enterprise 支持每个表一个标识列。Adaptive Server Enterprise 标识列只能包含标度为 0，最大精度为 38 的 Numeric 数据类型。它们必须为正数，它们受数据类型范围的限制，并且不能为 NULL。Adaptive Server Enterprise 要求使用 `identity_insert` 执行显式插入和删除，但不要求使用它执行标识列的更新。添加标识列时，该表可以包含数据。Adaptive Server Enterprise 用户不能显式设置为标识列选择的下一个值。Adaptive Server Enterprise 视图不能包含 `IDENTITY/AUTOINCREMENT` 列。在某些条件下使用 `SELECT INTO` 时，Adaptive Server Enterprise 允许结果表中具有标识/自动增量列（如果它们位于从中进行选择的表中）。

计算列

计算列支持有如下区别：

- SQL Anywhere 支持可以编制索引的计算列。
- Adaptive Server Enterprise 和 Sybase IQ 不支持。

临时表

您可以通过在 `CREATE TABLE` 语句中的表名前面放置不带所有者规范的井号 (#) 来创建临时表。这些临时表是 Sybase IQ 声明的临时表，仅在当前连接中可用。

定位表

Adaptive Server Enterprise 和 Sybase IQ 中以不同方式执行表的实际放置。Sybase IQ 支持 `ON segment-name` 子句，但 `segment-name` 引用 Sybase IQ dbspace。

使用 **CREATE DEFAULT**、**CREATE RULE** 和 **CREATE DOMAIN** 语句时的注意事项

Sybase IQ 提供了用于合并规则的替代方式。

- Adaptive Server Enterprise 支持使用 `Create Default` 和 `Create Rule` 语句来创建命名的缺省值。
- SQL Anywhere 和 Sybase IQ 支持使用 `CREATE DOMAIN` 语句来实现同一目标。

使用 **CREATE TRIGGER** 语句时的注意事项

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间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 Enterprise 仅支持语句级触发器。
- 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）。它还可用于编写审计追踪，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

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间的 **CREATE INDEX** 语法略有不同。

- Adaptive Server Enterprise 和 SQL Anywhere 使用以下语法 支持聚簇索引或非聚簇索引：

```
CREATE [UNIQUE] [CLUSTERED] INDEX name
ON table (column,...)
ON dbspace
```

Adaptive Server Enterprise 还允许使用 **NONCLUSTERED** 关键字，但对于这两种产品，缺省值都是 **NONCLUSTERED**。

- Adaptive Server Enterprise **CREATE INDEX** 语句在 SQL Anywhere 中运行，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实际执行查询。
- Sybase IQ 支持七种专用索引类型：**LF**、**HG**、**HNG**、**DATE**、**TIME**、**DTTM** 和 **WD**。Sybase IQ 还支持对两个具有相同数据类型、精度和标度的列之间的关系编制 **CMP** 索引。Sybase IQ 缺省创建 **HG** 索引，但在 **CREATE INDEX** 语句中指定了索引类型 则除外：

```
CREATE [UNIQUE] [type] INDEX name
ON table (column,...)
```

注意： Sybase IQ 还支持 **CREATE JOIN INDEX**，使用它可以针对经常在查询中以一致方式连接的一组特定列创建预连接的索引。

用户、组和权限

Adaptive Server Enterprise 与 SQL Anywhere 和 Sybase IQ 在用户和组的模型方面存在着一些差异。

在 Adaptive Server Enterprise 中，用户连接到某个服务器，而且每个用户都需要该服务器的登录 ID 和口令以及 该服务器上他们要访问的每个数据库的用户 ID。

SQL Anywhere 和 Sybase IQ 用户不需要服务器登录 ID。所有 SQL Anywhere 和 Sybase IQ 用户都会收到数据库的用户 ID 和口令。

用户组

所有这三种产品都支持用户组，所以您可以一次向许多用户授予权限。但是，组的具体细节存在区别：

- Adaptive Server Enterprise 只允许 每名用户成为一个组的成员。

- SQL Anywhere 和 Sybase IQ 允许用户成为多个组的成员，并且允许组层次。

所有这三种产品都拥有一个 **public** 组，该组用于定义缺省权限。每个用户都会自动成为该 **public** 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。

- 所有这三种产品都允许对数据库表和视图拥有 **SELECT**、**INSERT**、**DELETE**、**UPDATE** 和 **REFERENCES** 权限，以及对数据库表的选定列拥有 **UPDATE** 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** 组授予对 **TITLES** 表使用 **INSERT** 语句和 **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 Enterprise 对于其它项目 还支持 **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 Sybase IQ 要求用户具有 **RESOURCE** 权限才能创建数据库对象。（紧密对应的 Adaptive Server Enterprise 权限是 **GRANT ALL**，由数据库所有者使用。）
- 所有这三种产品都支持 **WITH GRANT OPTION** 子句，允许被授权者进而对它们授权，但 Sybase IQ 不允许 **WITH GRANT OPTION** 用在 **GRANT EXECUTE** 语句中。

数据库范围权限

Adaptive Server Enterprise 为数据库范围内的 用户权限使用不同的模型。

- SQL Anywhere 和 Sybase IQ 使用 **DBA** 权限来向用户授予对数据库的完全权限。
- Adaptive Server Enterprise 中的系统管理员享有服务器上所有数据库的这一权限。但是，对 Sybase IQ 数据库的 **DBA** 权限不同于 Adaptive Server Enterprise 数据库所有者的权限，该数据库所有者必须使用 Adaptive Server Enterprise **SETUSER** 语句获取对其它用户拥有的对象的 权限。

添加用户

Adaptive Server Enterprise 需要通过一个包含两个步骤的 过程来添加用户：首先是 **sp_addlogin**，然后是 **sp_add_user**。

SQL Anywhere 和 Sybase IQ 通过一个步骤添加用户。

虽然添加或删除用户不需要 Sybase IQ 登录管理存储过程，但该过程允许 **DBA** 添加或删除 Sybase IQ 用户 帐户。当启用 Sybase IQ 用户管理时，利用这些 Sybase IQ 用户 帐户，**DBA** 可以控制用户连接和口令有效期。

虽然 SQL Anywhere 和 Sybase IQ 允许使用 Adaptive Server Enterprise 系统过程管理用户和组，但在一些情况下，这些过程的确切语法和函数有所不同。

另请参见

- Adaptive Server Enterprise 系统过程（第 521 页）

装载格式

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间的装载格式支持有所不同。

- 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。
- Adaptive Server Enterprise 可通过 BCP 处理 ASCII 和 BINARY 装载格式。

注意： 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 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 Enterprise、SQL Anywhere 和 Sybase IQ 之间的查询要求有所不同。

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。
- 使用在变量名称前加 @ 符号这一 Transact-SQL 约定，以实现 Adaptive Server Enterprise 可移植性。
- 在 **BEGIN** 语句后立即声明过程和批处理语句中的变量和游标。Sybase IQ 需要这一点，尽管 Adaptive Server Enterprise 允许在过程或批处理语句中的任意位置进行声明。

- 不要在您的数据库中将 Adaptive Server Enterprise 或 Sybase IQ 中的保留字用作标识符。

编写兼容查询的标准

若要编写在 Sybase IQ 数据库和 Adaptive Server Enterprise 数据库上都能运行的查询，应遵循下面两条标准：

- 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。

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 639 页）

子查询支持

Sybase IQ 当前提供的子查询支持与 Adaptive Server Enterprise 和 SQL Anywhere 提供的子查询支持略有不同。

Adaptive Server Enterprise 和 SQL Anywhere 支持 **ON** 子句中的子查询；Sybase IQ 当前不支持。

按如下方式支持子查询中的 **UNION**：

- SQL Anywhere 支持相关子查询和非相关子查询中的 **UNION**。
- Sybase IQ 仅支持非相关子查询中的 **UNION**。
- Adaptive Server Enterprise 不支持任何子查询中的 **UNION**。

SQL Anywhere 支持许多其它位置的子查询，这些位置的标量值可以显示在语法中。关于可以在何处指定子查询，Adaptive Server Enterprise 和 Sybase IQ 都遵循 ANSI 标准。

GROUP BY 子句支持

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间的 **GROUP BY ALL** 支持有所不同。

- Adaptive Server Enterprise 支持 **GROUP BY ALL**，**GROUP BY ALL** 返回所有可能组，包括通过 **WHERE** 子句和 **HAVING** 子句删除的组。对于所有集合，这些子句均具有空值。
- 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**：

- Sybase IQ 和 SQL Anywhere 支持 **GROUP BY** 子句中的 **ROLLUP** 和 **CUBE**。
- Adaptive Server Enterprise 当前不支持 **ROLLUP** 和 **CUBE**。

Adaptive Server Enterprise 支持在 **SELECT** 子句中对非分组列进行投影。这称为按语义扩展的组，并返回一组值。Sybase IQ 支持按语义扩展的组，但 SQL Anywhere 不支持。只有 SQL Anywhere 支持 **List()** 集合返回值列表。

COMPUTE 子句支持

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间的 **COMPUTE** 支持有所不同。

- Adaptive Server Enterprise 支持 Transact-SQL **COMPUTE** 子句。
- SQL Anywhere 和 Sybase IQ 不支持 Transact-SQL **COMPUTE** 子句，原因是，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

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间的 WHERE 子句在对 Contains() 谓词的支持以及对 Like() 谓词中尾随空白的处理方面有所不同。

- Sybase IQ 支持使用 **Contains()** 谓词在字符数据中搜索单词（与 MS SQL Server 和 Verity 中的 Contains 类似）。Sybase IQ 使用 WORD 索引和 TEXT 索引优化这些搜索（如果可能）。
- Adaptive Server Enterprise 不支持 **Contains()**。

Transact-SQL 外连接支持

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间支持的外连接语法有所不同。

- Adaptive Server Enterprise 完全支持外连接的 *= 和 =* Transact-SQL 语法。
- SQL Anywhere 和 Sybase IQ 支持 Transact-SQL 外连接，但拒绝一些复杂的 Transact-SQL 外连接（具有潜在的不明确性）。
- 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使用 ANSI 语法。

注意： SQL Anywhere 和 Sybase IQ 不支持 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 MySybase 上的《Transact-SQL 外连接的语义和兼容性》白皮书。虽然本文档中的信息是针对 SQL Anywhere 而编写的，但也适用于 Sybase IQ。

ANSI 连接支持

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间的 ANSI 连接语法支持有所不同。

- Sybase IQ 当前不支持 **ON** 子句中的子查询。
- Adaptive Server Enterprise 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。

完全的外连接支持如下所示：

- SQL Anywhere 和 Sybase IQ 支持 **FULL OUTER JOIN**。
- Adaptive Server Enterprise 不支持 **FULL OUTER JOIN**。

空值比较支持

Adaptive Server Enterprise 拥有 Transact-SQL 扩展，这些扩展允许使用谓词比较空值。

例如，{col} = Null 表示 {col} Is Null。

SQL Anywhere 和 Sybase IQ 使用 ANSI 语义进行 Null 比较，但如果 ANSINULL 选项设置为 OFF 则除外，在这种情况下，这种比较是与 Adaptive Server Enterprise 兼容的。

注意：SQL Anywhere 8.0 和更高版本添加了 TDS_EMPTY_STRING_AS_NULL 支持，以在将空字符串映射到空值方面实现 Adaptive Server Enterprise 兼容性。

零长度字符串支持

零长度字符串在 Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 中的处理方式有所不同。

- Adaptive Server Enterprise 将零长度字符串 作为空值处理。
Adaptive Server Enterprise 用户存储单个空格 代表空字符串。
- SQL Anywhere 和 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 Null。

HOLDLOCK、SHARED 和 FOR BROWSE 支持

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 之间的 HOLDLOCK、SHARED 和 FOR BROWSE 语法有所不同。

- Adaptive Server Enterprise 支持 HOLDLOCK、SHARED 和 FOR BROWSE 语法。
- SQL Anywhere 支持 HOLDLOCK，但不支持 SHARED 或 FOR BROWSE。
- Sybase IQ 不支持这些关键字。

SQL 函数支持

Sybase IQ 支持 SQL Anywhere 和 Adaptive Server Enterprise 所支持的大多数函数，但具有某些区别。

- Adaptive Server Enterprise 支持 **PatIndex()** 中的 **USING CHARACTERS | USING BYTES** 语法；SQL Anywhere 和 Sybase IQ 则不支持。
- Adaptive Server Enterprise 支持 **Reverse()** 函数；SQL Anywhere 和 Sybase IQ 则不支持。
- Adaptive Server Enterprise 支持使用 **Len()** 作为 **Length()** 的替代语法；SQL Anywhere 则不支持该替代语法。
- Adaptive Server Enterprise 支持 **Square()** 和 **Str_Replace()** Microsoft 兼容性函数；SQL Anywhere 则不支持。
- Sybase IQ 支持 **Str_Replace()**。
- Adaptive Server Enterprise 和 SQL Anywhere 支持使用 **TSEQUAL()** 比较两个时间戳以确定修改时间；Sybase IQ 则不支持 **TSEQUAL()**。（**TSEQUAL** 在 Sybase IQ 表级版本控制模型中没有意义。）
- Sybase IQ 支持 **ROWID()**；Adaptive Server Enterprise 和 SQL Anywhere 则不支持。
- 除 Adaptive Server Enterprise Adaptive Server Enterprise 用于数据类型转换的 **Convert()** 外，SQL Anywhere 和 Sybase IQ 还支持 **Cast()**。

注意： **Cast()** 是符合 ANSI 标准的名称。

- SQL Anywhere 和 Sybase IQ 支持采用 **Lcase()** 和 **Ucase()** 作为 **Lower()** 和 **Upper()** 的同义词；Adaptive Server Enterprise 则不支持。
- SQL Anywhere 和 Sybase IQ 支持 **Locate()** 字符串函数；Adaptive Server Enterprise 则不支持。
- SQL Anywhere 支持使用 **IsDate()** 和 **IsNumeric()** 函数来测试将字符串转换为相应数据类型的功能；Adaptive Server Enterprise 则不支持。Sybase IQ 支持 **IsDate()**。您可以在 Sybase IQ 中使用 **IsNumeric**，但需要考虑 CIS 函数补偿性能方面的一些事宜。
- SQL Anywhere 支持 **NEWID**、**STRTOUID** 和 **UUIDTOSTR** 函数；Adaptive Server Enterprise 则不支持。这些是 Sybase IQ 中的本地函数，因此 CIS 功能补偿性能注意事项不适用。

注意： 一些 SQL 函数（包括 **SOUNDEX** 和 **DIFFERENCE** 字符串函数）和一些日期函数在 Sybase IQ 和 SQL Anywhere 中的运行方式不同。Sybase IQ 数据库选项 **ASE_FUNCTION_BEHAVIOR** 指定一些 Sybase IQ 数据类型转换函数（包括 **HEXTOINT** 和 **INTTOHEX**）的输出与 Adaptive Server Enterprise 函数的输出一致。

OLAP 函数支持

当前，Adaptive Server Enterprise 不支持 OLAP 函数。Sybase IQ 和 SQL Anywhere 则支持。

Sybase IQ 当前支持下列 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 Sybase IQ OLAP 函数。

当前，Adaptive Server Enterprise 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 产品开发的一个快速发展的领域。

系统函数支持

SQL Anywhere 和 SQL AnywhereSybase IQ 不支持某些 Adaptive Server Enterprise 系统函数。

SQL Anywhere 和 Sybase IQ 不支持以下 Adaptive Server Enterprise 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 Enterprise、SQL Anywhere 和 Sybase IQ 之间的用户定义的函数 (UDF) 支持有所不同：

- SQL Anywhere 支持使用 SQL、Java 和 C 编写的 UDF。

- Adaptive Server Enterprise 仅支持使用 Java 编写的 UDF。
- Sybase IQ 通过 CIS 查询分解来支持 UDF，但性能会受到影响。

解释日期的算术表达式的区别

SQL Anywhere 和 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 Enterprise 则不是这样。

- “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 Enterprise、SQL Anywhere 和 Sybase IQ 中的 SELECT INTO 语句所允许的表类型存在一些区别。

请考虑以下示例语句：

```
select into table1 from table2
```

- Adaptive Server Enterprise 允许 *table1* 为永久表、临时表或代理表。Adaptive Server Enterprise 还支持 **SELECT INTO EXISTING TABLE**。
- SQL Anywhere 和 Sybase IQ 允许 *table1* 成为永久表或临时表。仅当选择到 *table* 中并指定多个列时，系统才会创建永久表。**SELECT INTO #table**（不带所有者规范）始终创建临时表，无论指定的列数是多少。**SELECT INTO** 表（只含一个列）选择到宿主变量中。

可更新的视图支持

与 ANSI 的允许程度相比，Adaptive Server Enterprise 和 SQL Anywhere 更加自由，未请求 **WITH CHECK** 选项时，视图定义是可以更新的。

SQL Anywhere 提供 **ANSI_UPDATE_CONSTRAINTS** 选项来控制是将更新限制为 SQL92 支持的那些更新，还是一组更加自由的规则。

Sybase IQ 仅允许对可以展平的单表视图执行 **UPDATE**。Sybase IQ 不支持 **WITH CHECK**。

支持 UPDATE 和 DELETE 中的 FROM 子句

Sybase IQ、Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 都支持在 UPDATE 和 DELETE 中使用具有多个表的 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。

除了基于 SQL92 的 Watcom-SQL 术语之外，SQL Anywhere 和 Sybase IQ 还支持 Transact-SQL 存储过程语言中相当大的一部分。

Transact-SQL 存储过程概述

由于它是基于 ISO/ANSI 标准草案的，SQL Anywhere 和 Sybase IQ 存储过程语言在许多方面不同于 Transact-SQL 术语。

它们的许多概念和功能都相似，但语法却不同。SQL Anywhere 和 Sybase IQ 对 Transact-SQL 的支持是通过提供术语之间的自动转换来利用类似概念的。但是，只能使用两种术语中的一种来编写过程，而不能混用这两种术语。

SQL Anywhere 和 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。

批处理语句可以存储在命令文件中。SQL Anywhere 和 Sybase IQ 中的 ISQL 实用程序以及 Adaptive Server Enterprise 中的 `isql` 实用程序为交互地执行批处理语句提供了相似的功能。

过程中使用的控制语句也可以在批处理语句中使用。SQL Anywhere 和 Sybase IQ 支持在批处理语句中使用控制语句，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语句的结尾。

对于命令文件中存储的批处理语句，SQL Anywhere 和 Sybase IQ 支持在命令文件中使用参数。Adaptive Server Enterprise 不支持参数。

过程和批处理中的 SQL 语句

Sybase IQ 支持的 SQL 语句中有的是一种术语的一部分，有的则不是。

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。
- 可以在一个批处理语句或过程中同时包含 Adaptive Server Enterprise 不支持的语句以及两种服务器都支持的语句。
- 不能在一个批处理语句或过程中包含仅限 Transact-SQL 的语句以及仅限 Sybase IQ 的语句。

未用分号分隔的 SQL 语句是 Transact-SQL 过程或批处理的一部分。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 Enterprise 和 SQL Anywhere 支持在变量和由表达式子查询返回的标量值之间进行比较。

例如：

```
create procedure testIf () begin declare var4 int; set var4 =
10; if var4 = (select MIN (a_i1) from a) then set var4 = 100;
end if; end;
```

CASE 语句支持

在 Sybase IQ 和 SQL Anywhere 中，CASE 语句允许的用法不同。

Adaptive Server Enterprise 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 22 页）

行级游标操作支持

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 支持将游标与 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 Sybase IQ 中，可更新游标仅限敏感性游标、仅用于一个表以及仅限链式游标。禁止使用可更新持有游标。Sybase IQ 中的可更新游标获取一个表锁。

PRINT 命令支持

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 中的 PRINT 支持有所不同。

PRINT 的效果取决于客户端：

- Adaptive Server Enterprise**PRINT** 始终向客户端 发送消息。
- 在 SQL Anywhere 和 Sybase IQ 中， **PRINT** 将消息发送到客户端以建立 Open Client 和 JDBC 连接。
- 依赖于 **PRINT** 的 Adaptive Server Enterprise 存储过程 使用 Interactive SQL 在 Sybase IQ 中运行。

注意： Sybase IQ 用户可能会首选带 JDBC 的 Interactive SQL， 而不是 iAdaptive Server Anywhere JDBC 驱动程序（以前称为 JDBC-ODBC 桥）。

存储过程的自动转换

除了支持 Transact-SQL 替代语法之外， SQL Anywhere 和 Sybase IQ 还提供了用于在 Watcom-SQL 术语和 Transact-SQL 术语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 216. 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/Sybase IQ 过程和 Transact-SQL 过程以不同的方式返回结果集。

SQL Anywhere 和 Sybase IQ 使用 **RESULT** 子句来指定所返回的结果集。

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
```

下面是相应的 SQL Anywhere 或 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
```

三个 Sybase 客户端工具向客户端提供多个结果的方法有一些细微的区别：

- **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**。
- **Interactive SQL Classic**（不建议使用）提供 **RESUME** 以显示每个连续的结果集。

Transact-SQL 过程中的变量

SQL Anywhere 和 Sybase IQ 为过程中的变量赋值的方式不同于 Transact-SQL。

SQL Anywhere 和 Sybase IQ 使用 **SET** 语句来向过程中的变量赋值。

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 Enterprise** 中，您可以在存储过程正文中的任意位置声明变量。变量的保留时间为该过程的持续时间。
- 在 **SQL Anywhere** 和 **Sybase IQ** 中，必须在复合语句的开头声明变量（即紧跟在 **BEGIN...END** 对中的 **BEGIN** 后面）。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 629 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 217. 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

值	含义
-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 Sybase IQ 的区别 和共享的功能

Sybase IQ 和 SQL Anywhere 在启动和管理数据库和服务、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，请在使用产品时一定要参考 Sybase IQ 文档集。请在使用 SQL Anywhere 时参考 SQL Anywhere 文档集，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

Sybase IQ 和 SQL Anywhere 启动和管理数据库和服务的方式有所不同。

- 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。
- Sybase IQ 不支持个人服务器。
- Sybase IQ 支持许多 SQL Anywhere 服务器命令行选项，但并非支持全部。Sybase IQ 支持其它服务器选项，但 SQL Anywhere 则不支持。
- Sybase IQ 提供 **stop_iq** 实用程序 (UNIX) 来关闭服务器。
- 在 Sybase IQ 和 SQL Anywhere 中，**BACKUP** 和 **RESTORE** 语句允许的 子句有所不同。
- Sybase IQ 仅支持使用 SQL Remote 执行 Multiplex 操作。

Sybase IQ 支持许多 SQL Anywhere 数据库管理实用程序，但并非全部支持：

- Sybase IQ 不支持以下 SQL Anywhere 实用程序：**backup**、**compression**、**console**、**initialization**、**license**、**log transfer**、**log translation**、**rebuild**、**spawn**、某些 **transaction log** 选项（-g、-il、-ir、-n、-x、-z）、**uncompression**、**unload**、**upgrade** 和 **write file**。
- Sybase IQ 仅支持 Catalog 存储上的 SQL Anywhere **validation** 实用程序。要验证 IQ 存储，请使用 **sp_iqcheckdb**。

SQL Anywhere 数据库选项

包括 **DEFAULT_TIMESTAMP_INCREMENT** 在内的部分 SQL Anywhere 数据库选项不受 Sybase IQ 支持。

有些数据库选项仅应用于 Catalog 存储，包括：**FOR_XML_NULL_TREATMENT**、**ISOLATION_LEVEL**、**PREFETCH**、**PRECISION** 和 **SCALE**。

在行为、缺省值或允许值方面有区别的选项包括 **DELAYED_COMMITS**、**TIME_FORMAT**、**TIMESTAMP_FORMAT**。

Sybase IQ 还包括许多不受 SQL Anywhere 支持的选项。

SQL Anywhere 数据定义语言 (DDL) 区别

除了前面讨论的 DDL 区别之外：

- 在 **ALTER TABLE** 语句的 **DELETE/DROP** 或 **PRIMARY KEY** 子句中，Sybase IQ 执行 **RESTRICT** 操作（如果存在关联的外键，则报告错误）。SQL Anywhere 始终执行 **CASCADE** 操作。
- 同样，如果存在关联的外键约束，**DROP TABLE** 语句会在 Sybase IQ 中报告 错误。
- Sybase IQ 不支持下列 DDL 语句：**CREATE COMPRESSED DATABASE**、**CREATE TRIGGER**、**SETUSER**。
- 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。
- Sybase IQ 表不能拥有引用 SQL Anywhere（或 Catalog）表的外键，而 SQL Anywhere 表不能拥 有引用 Sybase IQ 表的外键。
- 在 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。Sybase IQ 的缺省值是 **CASE RESPECT** 和 **ISO_BINENG** 归类；对于 SQL Anywhere，缺省值是 **CASE IGNORE** 以及从操作系统的语言和 字符集推导出的归类。
- Sybase IQ 不 支持 SQL Anywhere 所支持的 **CREATE ENCRYPTED DATABASE** 和 **CREATE DECRYPTED DATABASE** 命令。请参见 《Sybase IQ 中的高级安全性》。

SQL Anywhere 数据操作语言 (DML) 区别

并不是所有的 SQL Anywhere DML 对象 和语法都受 Sybase IQ 支持。

- Sybase IQ 不支持下列 DML 和过程语句：**EXPLAIN**、**GET DATA**、**INPUT**、**PREPARE TO COMMIT**、**PUT**、**READTEXT**、**ROLLBACK TRIGGER**、**SYSTEM**、**UNLOAD TABLE**、**VALIDATE TABLE**。

注意：一组提取选项执行与 **UNLOAD TABLE** 类似的角色；有关详细信息，请参见《系统管理指南第一卷》中的数据提取选项。

- Sybase IQ 支持 **INSERT...LOCATION** 语法；SQL Anywhere 则 不支持。
- 在 Sybase IQ 和 SQL Anywhere 中，**LOAD TABLE** 选项是不同的。
- Sybase IQ 中的 **OPEN** 语句不 支持 **BLOCK** 和 **ISOLATION LEVEL** 子句。
- Sybase IQ 不支持触发器。
- 事务、隔离级别、检查点和自动生成的 **COMMIT** 以及游标支持的使用在 Sybase IQ 和 SQL Anywhere 中是不同的。
- 从 Sybase IQ 中的存储过程 执行 **SELECT** 时，需要注意 CIS 函数补偿性能方面的一些事宜。
- Sybase IQ 忽略 Adaptive Server Enterprise **SELECT** 语句中 完全限定名的数据库名称限定符，如带有 `<database name>.<owner>.<table name>` 的 **FROM** 子句。例如，Sybase IQ 将查询 **SELECT * FROM XXX..TEST** 解释为 **SELECT * FROM TEST**。

Adaptive Server Enterprise 和 Sybase IQ 的区别和共享的功能

Sybase IQ 和 Adaptive Server Enterprise 在存储过程支持和 视图支持方面存在一些区别。

有关其它信息，请在使用产品时一定要参考 Sybase IQ 文档集。请在使用 Adaptive Server Enterprise 时参考 Adaptive Server Enterprise 文档集，Sybase IQ 文档仅在涉及特定功能时才引用 Adaptive Server Enterprise 文档。

Adaptive Server Enterprise 存储过程

Sybase IQ 不支持某些存储过程。

Sybase IQ 不再支持 以下的 Adaptive Server Enterprise 存储过程：

- **sp_addserver**
- **sp_configure**
- **sp_estspace**
- **sp_help**
- **sp_helpuser**
- **sp_who**

Sybase IQ 不再支持以下的 Catalog 过程：

- **sp_column_privileges**
- **sp_databases**
- **sp_datatype_info**
- **sp_server_info**

Adaptive Server Enterprise 系统视图

Sybase IQ 不支持某些视图。

Sybase IQ 不再支持以下 Adaptive Server Enterprise 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
- syssrvroles
- systhresholds
- sysusages

列名区别

Adaptive Server Enterprise 视图 SYSTYPES 中所用的列名为 “allownulls”。Sybase IQ 视图 SYSTYPES 中所用的列名为 “allowsnulls”。

索引

A

ABS 函数 114
 ACOS 函数 114
 Adaptive Server Enterprise
 兼容性 613, 620
 ALL
 条件 37
 AND 条件 44
 ANY
 条件 37
 ARGN 函数 115
 ASCII 函数 116
 ASCII 值 116, 127
 ASIN 函数 117
 ATAN 函数 118
 ATAN2 函数 118
 AVG 函数 119
 安装目录
 关于 3

B

BETWEEN 条件 38
 BFILE 函数
 说明 120
 BIGDATETIME 数据类型
 兼容性 620
 BIGINTTOHEX 函数 121
 BIGTIME 数据类型
 兼容性 620
 BINARY 数据类型 73
 BIT 数据类型
 Transact-SQL 77
 兼容性 617
 BIT_LENGTH 函数 121
 BLOB 数据类型
 LIKE 条件 38
 BYTE_LENGTH 函数 122
 BYTE_LENGTH64 函数
 说明 124
 BYTE_SUBSTR 函数
 说明 124
 BYTE_SUBSTR64 函数
 说明 124

百分点
 使用 NTILE 函数计算 228
 保留字 17
 列表 17
 备份
 活动较少期间 482
 检查点操作期间 482
 备份操作
 摘要 342
 备份历史记录文件
 位置 10
 本地计算机
 环境设置 14
 比较
 关于 31
 比较日期和时间 81
 变量
 Transact-SQL 639
 关于 55
 局部 55
 连接级 57
 全局 55, 57
 标识
 外部库 336
 标识符
 SQL Anywhere 中的最大长度 20
 SQL 语法 20
 关于 20
 区分大小写 622
 唯一性 622
 标识列
 兼容性 624
 支持作为缺省值 624
 标准偏差
 函数 282
 样本函数 285
 总体函数 284
 标准偏差函数 280
 表
 iq_dummy 91
 Transact-SQL 623
 显示有关信息 395
 表达式 22
 CASE 28
 Transact-SQL 29

- 长度 (以字节为单位) 145
- 转换为时间戳 161
- 表达式子查询
 - IF 语句中 637
- 并置字符串 25

C

- CASE 表达式 28
 - NULLIF 函数 229
- CAST 函数 86, 124
- Catalog
 - Adaptive Server Enterprise 兼容性 615
- Catalog 存储
 - IQ 615
 - 监控 455
- CEIL 函数 125
- CEILING 函数 126
- char
 - 拆分为术语 335
- CHAR 函数 127
- CHAR 数据类型
 - 关于 65
- CHAR_LENGTH 函数 128
- CHAR_LENGTH64 函数
 - 说明 129
- CHARACTER VARYING 数据类型
 - 关于 65
 - 剪裁尾随空白 67, 617
- CHARACTER 数据类型
 - 关于 65
- CHARINDEX 函数 129
- CHECK 条件
 - Transact-SQL 623
- CHECKPOINT 语句
 - 在检查点操作期间备份 482
- CLOB 数据类型 38
- COALESCE 函数 130
- COL_LENGTH 函数 131
- COL_NAME 函数 132
- COMPUTE 子句
 - Transact-SQL 630
- CONNECTION_PROPERTY 函数 133
- CONTAINS 条件
 - 具有 TEXT 索引 43
 - 具有 WD 索引 42
- CONVERT 函数 86, 134
 - 日期到整数转换 137
 - 日期到字符串转换 137

- 整数到日期转换 137
- 字符串到日期转换 137
- CORR 函数 138
- COS 函数 139
- cosine 139
- COT 函数 140
- COUNT 函数 143
- COVAR_POP 函数 140
- COVAR_SAMP 函数 142
- CPU 利用率
 - 数据库一致性检查程序 348
- CREATE DECRYPTED DATABASE 语句 643
- CREATE DEFAULT 语句
 - 不支持 625
- CREATE DOMAIN 语句
 - Transact-SQL 兼容性 625
 - 使用 83
 - 语法 84
- CREATE ENCRYPTED DATABASE 语句 643
- CREATE INDEX 语句
 - IQ 626
 - Transact-SQL 626
- CREATE RULE 语句
 - 不支持 625
- CREATE TABLE 语句
 - Transact-SQL 623
- CREATE TRIGGER
 - 不受支持 625
- CUBE 操作
 - GROUPING 函数 182
- CUME_DIST 函数 144
- CURRENT DATABASE
 - 特殊值 52
- CURRENT DATE
 - 缺省值 52
 - 特殊值 52
- CURRENT PUBLISHER
 - 缺省值 52
 - 特殊值 52
- CURRENT TIME
 - 缺省值 52
 - 特殊值 52
- CURRENT TIMESTAMP
 - 缺省值 53
 - 特殊值 53
- CURRENT USER
 - 缺省值 53
 - 特殊值 53
- 参数选择 115

- 参照完整性约束
 - CASCADE 不受支持 624
 - 兼容性 624
- 插入
 - SQL Anywhere 643
- 查询
 - Transact-SQL 629
- 查询服务器
 - 同步 10
- 拆分
 - 术语 335, 336
- 常量
 - Transact-SQL 29
 - 表达式中 23
- 触发器
 - 不受支持 625
- 窗口规范 95
- 窗口函数
 - 窗口分区 95
 - 窗口函数类型 95
 - 窗口名称或规范 95
- 窗口函数, 定义 95
- 窗口集合函数 93
- 窗口类型 95
- 窗口名称 95
- 创建
 - 数据类型 83, 84
- 存储过程
 - Adaptive Server Enterprise 644
 - sa_char_terms 335
 - sa_external_library_unload 336
 - sa_list_external_library 336
 - sa_nchar_terms 336
 - sa_text_index_vocab 336
 - sp_iqbackupdetails 340
 - sp_iqbackupsummary 342
 - sp_iqclient_lookup 354
 - sp_iqmpxinconnpoolinfo 427
 - sp_iqmpxinheartbeatinfo 427
 - sp_iqmpxinfo 427
 - sp_iqmpxvalidate 427
 - sp_iqrestoreaction 444
- 存储过程语言
 - 概述 636
- 错误
 - Transact-SQL 640, 641
- 错误消息
 - ERRORMSG 函数 170
 - 获取文本 170

D

- DATALENGTH 函数 145
- DATE 函数 146
- DATE 数据类型 78
- DATE_ORDER 选项 81
- DATEADD 函数 146
- DATECEILING 函数 148
- DATEDIFF 函数 150
- DATEFLOOR 函数 152
- DATEFORMAT 函数 155
- DATENAME 函数 156
- DATEPART 函数 157
- DATEROUND 函数 158
- DATETIME 函数 161
- DAY 函数 161
- DAYNAME 函数 162
- DAYS 函数 163
- DB_ID 函数 164
- DB_NAME 函数 165
- DB_PROPERTY 函数 166
- dbcc
 - 线程使用情况 348
- DBCC
 - 输出 351
 - 数据库检验 346
 - 性能 351
 - 运行时间 351
- DBCC_LOG_PROGRESS 选项 351
- dbinit 实用程序 615
- DDL
 - SQL Anywhere 643
- DECIMAL data type 69
- DEGREES 函数 167
- DENSE_RANK 函数 167
- DIFFERENCE 函数 169
- DISK 语句
 - 不支持 615
- DML
 - SQL Anywhere 643
- DOW 函数 170
- DUMMY 表 528
- 大对象数据
 - LIKE 条件 38
- 代码页
 - 和数据存储 66
- 当前用户
 - 环境设置 14

索引

登录策略

- 分配用户 424
- 复制 366, 423

登录管理

- sp_expireallpasswords 338
- sp_iqaddlogin 338
- sp_iqcopyloginpolicy 366, 423

定义窗口 95

对象

- 确定 ID 232
- 确定名称 232
- 显示有关信息 395
- 重命名 441

E

ELSE

- IF 表达式 27

ENDIF

- IF 表达式 27

ERRORMSG 函数

- SQL 语法 170

EVENT_CONDITION 函数 171

EVENT_CONDITION_NAME 函数 173

EVENT_PARAMETER 函数 173

EXISTS 条件 43

EXP 函数 175

EXP_WEIGHTED_AVG 函数 175

二进制数据

- 兼容性 618

F

FIRST_VALUE 函数 177

FLOAT 数据类型 70

FLOOR 函数 179

FOR BROWSE 语法

- Transact-SQL 632

FP 索引

- 检验 349

FROM 子句 104

- UPDATE 和 DELETE 635

发布者

- SQL Remote 52

反斜杠

- 不允许在 SQL 标识符中使用 20

反余弦值 114

反正切值 118

反正切值比率 118

反正弦值 117

返回值

- 过程 640

分布函数 93

分隔 SQL 字符串 20

分离查询谓词 34

分配映射

- 重置 349

分区

- 一致性检查 347

分区表

- 检验 349

分析函数 93

- DENSE_RANK 167

- NTILE 228

- PERCENT_RANK 236

- PERCENTILE_CONT 237

- PERCENTILE_DISC 239

- RANK 247

服务

- 注册表条目 14

服务器

- 属性 111

服务器管理

- SQL Anywhere 和 IQ 642

G

GETDATE 函数 179

GRAPHICAL_PLAN 函数 180

GROUP BY

- 兼容性 630

GROUP_MEMBER 函数

- SQL 语法 183

GROUPING 函数 182

GUID

- NEWID 函数的 SQL 语法 223

- STRTOUUID 函数的 SQL 语法 290

- UUIDTOSTR 函数的 SQL 语法 311

估计值

- 优化程序 46

关键字

- SQL 17

- 列表 17

归类

- SORTKEY 函数 274

规则

- Transact-SQL 625

过程

- Transact-SQL 638
- Transact-SQL 概述 636
- 错误处理 640, 641
- 返回值 640
- 显示参数信息 436
- 显示有关信息 395, 434
- 转换 638

过程语言

- 概述 636

H

- HEXTOBIGINT 函数 184
- HEXTOINT 函数 185
 - ASE_FUNCTION_BEHAVIOR 选项 186
- HOLDLOCK 语法
 - Transact-SQL 632
- HOUR 函数 187
- HOURS 函数 188
- HTML_DECODE 函数 189
- HTML_ENCODE 函数 189
- HTML_PLAN 190
- HTML_PLAN 函数 190
- HTTP 函数 103
 - HTML_DECODE 189
 - HTML_ENCODE 189
 - HTTP_DECODE 191
 - HTTP_ENCODE 192
 - HTTP_HEADER 192
 - HTTP_VARIABLE 192
 - NEXT_HTTP_HEADER 227
 - NEXT_HTTP_VARIABLE 227
- HTTP_DECODE 函数 191
- HTTP_ENCODE 函数 192
- HTTP_HEADER 函数 192
- HTTP_VARIABLE 函数 192
- 函数
 - ABS 函数 114
 - ACOS 函数 114
 - Adaptive Server Enterprise 系统函数 109
 - ARGN 函数 115
 - ASCII 函数 116
 - ASIN 函数 117
 - ATAN 函数 118
 - ATAN2 函数 118
 - AVG 函数 119
 - BFILE 函数 120
 - BIGINTTOHEX 函数 121

- BIT_LENGTH 函数 121
- BYTE_LENGTH 函数 122
- BYTE_LENGTH64 函数 124
- BYTE_SUBSTR 函数 124
- BYTE_SUBSTR64 函数 124
- CAST 函数 124
- CEIL 函数 125
- CEILING 函数 126
- CHAR 函数 127
- CHAR_LENGTH 函数 128
- CHAR_LENGTH64 函数 129
- CHARINDEX 函数 129
- COALESCE 函数 130
- COL_LENGTH 函数 131
- COL_NAME 函数 132
- CONNECTION_PROPERTY 函数 133
- CONVERT 函数 134
- CORR 函数 138
- COS 函数 139
- COT 函数 140
- COUNT 函数 143
- COVAR_POP 函数 140
- COVAR_SAMP 函数 142
- CUME_DIST 函数 144
- DATALLENGTH 函数 145
- DATE 函数 146
- DATEADD 函数 146
- DATECEILING 函数 148
- DATEDIFF 函数 150
- DATEFLOOR 函数 152
- DATEFORMAT 函数 155
- DATENAME 函数 156
- DATEPART 函数 157
- DATEROUND 函数 158
- DATETIME 函数 161
- DAY 函数 161
- DAYNAME 函数 162
- DAYS 函数 163
- DB_ID 函数 164
- DB_NAME 函数 165
- DB_PROPERTY 函数 166
- DEGREES 函数 167
- DENSE_RANK 函数 167
- DIFFERENCE 函数 169
- DOW 函数 170
- ERRORMSG 函数 SQL 语法 170
- EVENT_CONDITION 函数 171
- EVENT_CONDITION_NAME 函数 173

EVENT_PARAMETER 函数 173
EXP 函数 175
EXP_WEIGHTED_AVG 函数 175
FIRST_VALUE 函数 177
FLOOR 函数 179
GETDATE 函数 179
GRAPHICAL_PLAN 180
GROUP_MEMBER 函数 SQL 语法 183
GROUPING 函数 SQL 语法 182
HEXTOBIGINT 函数 184
HEXTOINT 函数 185
HOUR 函数 187
HOURS 函数 188
HTML_DECODE 函数 189
HTML_ENCODE 函数 189
HTML_PLAN 函数 190
HTTP 103
HTTP_DECODE 函数 191
HTTP_ENCODE 函数 192
HTTP_HEADER 函数 192
HTTP_VARIABLE 函数 192
IFNULL 函数 193
INDEX_COL 函数 193
INSERTSTR 函数 194
INTTOHEX 函数 195
IQ 扩展 327
ISDATE 函数 SQL 语法 196
ISNULL 函数 197
ISNUMERIC 函数 SQL 语法 198
LAST_VALUE 201
LCASE 函数 203
LEFT 函数 205
LEN 函数 SQL 语法 206
LENGTH 函数 207, 209
LIST 函数 209
LOCATE 函数 210
LOG 函数 211
LOG10 函数 212
LOWER 函数 213
LTRIM 函数 214
MAX 函数 214
MEDIAN 函数 215
MIN 函数 217
MINUTE 函数 217
MINUTES 函数 218
MOD 函数 219
MONTH 函数 220
MONTHNAME 函数 221
MONTHS 函数 221
NEWID 函数 SQL 语法 223
NEXT_CONNECTION 函数 224
NEXT_DATABASE 函数 226
NEXT_HTTP_HEADER 函数 227
NEXT_HTTP_VARIABLE 函数 227
NOW 函数 227
NTILE 函数 228
NULLIF 函数 229
NUMBER 函数 231
OBJECT_ID 函数 232
OBJECT_NAME 函数 232
OCTET_LENGTH 函数 233
PATINDEX 函数 234
PERCENT_RANK 函数 236
PERCENTILE_CONT 函数 237
PERCENTILE_DISC 函数 239
PI 函数 241
POWER 函数 241
PROPERTY 函数 242
PROPERTY_DESCRIPTION 函数 243
PROPERTY_NAME 函数 243
PROPERTY_NUMBER 函数 244
QUARTER 函数 245
RADIANS 函数 246
RAND 函数 246
RANK 函数 247
REGR_AVGX 函数 249
REGR_AVGY 函数 250
REGR_COUNT 函数 251
REGR_INTERCEPT 函数 252
REGR_R2 函数 253
REGR_SLOPE 函数 254
REGR_SXX 函数 256
REGR_SXY 函数 257
REGR_SYY 函数 258
REMAINDER 函数 259
REPEAT 函数 260
REPLACE 函数 261
REPLICATE 函数 263
REVERSE 函数 SQL 语法 264
RIGHT 函数 265
ROUND 函数 266
ROWID 函数 268
RTRIM 函数 269
SECOND 函数 270
SECONDS 函数 271
SIGN 函数 272

- SIMILAR 函数 273
- SIN 函数 274
- SORTKEY 函数 274
- SOUNDEX 函数 278
- SPACE 函数 279
- SQLFLAGGER 函数 280
- SQRT 函数 281
- SQUARE 函数 281
- STDDEV 函数 282
- STDDEV_POP 函数 284
- STDDEV_SAMP 函数 285
- STR 函数 286
- STR_REPLACE 函数 SQL 语法 287
- STRING 函数 289
- STRTOUUID 函数 SQL 语法 290
- STUFF 函数 291
- SUBSTR 函数 292
- SUBSTRING 函数 292
- SUBSTRING64 函数 293
- SUM 函数 293
- SUSER_ID 函数 294
- SUSER_NAME 函数 295
- TAN 函数 296
- today 528
- TODAY 函数 297
- Transact-SQL 632
- TRIM 函数 297
- TRUNCNUM 函数 298
- TS_ARMA_AR 函数 299
- TS_ARMA_CONST 函数 299
- TS_ARMA_MA 函数 299
- TS_AUTO_ARIMA 函数 300
- TS_AUTO_ARIMA_OUTLIER 函数 300
- TS_AUTO_ARIMA_RESULT_AIC 函数 300
- TS_AUTO_ARIMA_RESULT_AICC 函数 301
- TS_AUTO_ARIMA_RESULT_BIC 函数 301
- TS_AUTO_ARIMA_RESULT_FORECAST_ERROR 函数 301
- TS_AUTO_ARIMA_RESULT_FORECAST_VALUE 函数 301
- TS_AUTO_ARIMA_RESULT_MODEL_D 函数 302
- TS_AUTO_ARIMA_RESULT_MODEL_P 函数 302
- TS_AUTO_ARIMA_RESULT_MODEL_S 函数 302
- TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA 函数 302
- TS_AUTO_UNI_AR 函数 303
- TS_AUTOCORRELATION 函数 299
- TS_BOX_COX_XFORM 函数 303
- TS_DIFFERENCE 函数 303
- TS_DOUBLE_ARRAY 函数 303
- TS_ESTIMATE_MISSING 函数 304
- TS_GARCH 函数 304
- TS_GARCH_RESULT_A 函数 304
- TS_GARCH_RESULT_AIC 函数 304
- TS_GARCH_RESULT_USER 函数 305
- TS_INT_ARRAY 函数 305
- TS_LACK OF FIT 函数 305
- TS_LACK OF FIT_P 函数 305
- TS_MAX_ARMA_AR 函数 306
- TS_MAX_ARMA_CONST 函数 306
- TS_MAX_ARMA_LIKELIHOOD 函数 306
- TS_MAX_ARMA_MA 函数 306
- TS_OUTLIER_IDENTIFICATION 函数 307
- TS_PARTIAL_AUTOCORRELATION 函数 307
- TS_VWAP 函数 307
- UCASE 函数 308
- UPPER 函数 309
- USER_ID 函数 310
- USER_NAME 函数 310
- UUIDTOSTR 函数 SQL 语法 311
- VAR_POP 函数 312
- VAR_SAMP 函数 314
- VARIANCE 函数 315
- WEEKS 函数 317
- WEIGHTED_AVG 函数 318
- WIDTH_BUCKET 函数 320
- YEAR 函数 321
- YEARS 函数 322
- YMD 函数 324
- 按字母顺序排列的列表 113
- 窗口化集合 93
- 分布 93
- 分析 93
- 集合 91
- 排名 93
- 日期和时间 99
- 时序 113
- 数据类型转换 98
- 数值 94, 104
- 统计 93

- 一致的结果 104
- 用户定义的 112
- 杂项 113
- 字符串 105
- 函数, 集合
 - GROUING 182
- 函数, 数据类型转换
 - ISDATE 196
- 函数, 杂项
 - ERRORMSG 170
 - ISNUMERIC 198
 - NEWID 223
- 函数, 字符串 190, 206, 264, 287
 - STRTOUUID 290
 - UIDTOSTR 311
- 环境变量
 - IQLOGDIR15 10
 - IQPORT 8
 - IQTIMEOUT 10
 - PATH 12
 - SQLCONNECT 13
 - SYBASE 13
 - SYBASE_JRE 13
 - SYBASE_OCS 14
 - 关于 6
- 缓冲区高速缓存
 - 使用 sp_iqsysmon 监控 455
- 恢复操作
 - 一致状态 444

I

- IF 表达式 27
- IFNULL 函数 193
- IMAGE 数据类型 618
 - 兼容性 621
- IN 条件 42
- INDEX_COL 函数 193
- INDEX_PREFERENCE 选项 48
- INSERTSTR 函数 194
- INTEGER 数据类型 69
- INTTOHEX 函数 195
 - ASE_FUNCTION_BEHAVIOR 选项 195
- IQ 存储 615
- IQ 代理
 - 等待时间 10
 - 端口 8
- iq_dummy 表 91, 528
- IQCHARSET 环境变量 7

- IQDIR15 环境变量 7
- iqinit 实用程序 615
- ISQLANG 9
- ISQLANG 环境变量 8
- IQLOGDIR15 环境变量 10
- IQPORT 环境变量 8
- IQTIMEOUT 环境变量
 - 指定 IQ 代理的等待时间 10
- IQTMP15 环境变量 11
- IS NULL 条件 44
- ISDATE 函数
 - SQL 语法 196
- ISNULL 函数 197
- ISNUMERIC 函数
 - SQL 语法 198

J

- Java
 - 用户定义的函数 112
- Java 数据类型
 - 兼容性 621
- Java 运行时环境
 - 设置 13
- JAVA_HOME 环境变量 11
- 集合函数 91
 - AVG 119
 - COUNT 143
 - LIST 209
 - MAX 214
 - MIN 217
 - STDDEV 282
 - STDDEV_POP 284
 - STDDEV_SAMP 285
 - SUM 293
 - VAR_POP 312
 - VAR_SAMP 314
 - VARIANCE 315
- 计算列
 - 不受支持 625
- 兼容性
 - Adaptive Server Enterprise 613
 - Adaptive Server Enterprise 数据类型 620
 - 参照完整性约束 624
- 监控
 - sp_iqsysmon 过程 455
- 剪裁尾随空白 67, 617
- 检查约束 623
 - Transact-SQL 兼容性 623

- 强制 623
- 检验
 - 分区表 349
 - 键 349
 - 索引 349
- 建议
 - 存储 404
 - 清除 404
 - 显示 404
- 键
 - 检验 349
 - 显示有关信息 431
- 角色
 - Adaptive Server Enterprise 616
- 结果集
 - Transact-SQL 638
- 局部变量
 - 关于 55
- 绝对值 114

K

- 可移植 SQL 628
- 空白
 - 剪裁尾随 617
 - 剪裁尾随空白 67
- 空值
 - Transact-SQL 兼容性 623
 - 关于 62
- 空值比较
 - Transact-SQL 631
- 口令
 - 到期 338
 - 区分大小写 622
 - 添加或修改 430

L

- language_code 9
- LAST USER
 - 特殊值 54
- LAST_VALUE 函数 201
- LCASE 函数 203
- LD_LIBRARY_PATH 环境变量 12
- LEFT 函数 205
- LEN 函数
 - SQL 语法 206
- LENGTH 函数 207, 209
- LIBPATH 环境变量 12

- LIKE 条件 38
 - LONG BINARY 数据 38
 - LONG VARCHAR 数据 38
 - 大对象数据 38
- LIST 函数 209
- LOB 数据类型
 - LIKE 条件 38
- LOCATE 函数 210
- LOG 函数 211
- LOG10 函数 212
- logarithm (base 10) 212
- LONG BINARY 数据类型 618, 621
 - LIKE 条件 38
- LONG VARCHAR 数据类型 38
- LOWER 函数 213
- LTRIM 函数 214
- LVC 单元 350
- 类型转换 86
- 立即刷新 336
- 连接
 - Transact-SQL 631
 - 确定 ID 号 224
 - 属性 110
 - 外连接运算符 27
 - 显示有关信息 476
 - 自动 326
 - 连接等同性条件 51
 - 连接级变量
 - 关于 57
 - 连接属性值 133
 - 连接索引
 - 显示有关信息 415
 - 连接信息
 - sp_iqcontext 364
 - 连接运算符
 - ANSI 631
 - Transact-SQL 631
 - 链式外连接 631
- 列
 - 和用户定义数据类型 83
 - 命名 23
- 列表
 - 外部库 336
- 列长度 131
- 列名称 132
- 列缺省值
 - 不受支持 624
- 临时表
 - Transact-SQL 625

索引

零长度字符串
Transact-SQL 632

M

MAX 函数 214
MEDIAN 函数 215
MIN 函数 217
MINUTE 函数 217
MINUTES 函数 218
MOD 函数 219
MONEY 数据类型 72
MONTH 函数 220
MONTHNAME 函数 221
MONTHS 函数 221
MPXServerName 列 360
Multiplex
 同步查询服务器 10
 系统过程 359
命名空间
 索引 622
模式匹配
 关于 38
 和归类 40
 限制 39
目录
 系统表 525
目录结构 3

N

nchar
 拆分为术语 336
NEWID 函数
 SQL 语法 223
NEXT_CONNECTION 函数 224
NEXT_DATABASE 函数 226
NEXT_HTTP_HEADER 函数 227
NEXT_HTTP_VARIABLE 函数 227
NOT 条件 45
NOW 函数 227
NTILE 函数 228
NULLIF 函数 29, 229
NUMBER 函数 231
内存
 使用 sp_iqsysmon 监控 455

O

OBJECT_ID 函数 232

OBJECT_NAME 函数 232
OCTET_LENGTH 函数 233
OLAP

 DENSE_RANK 函数 167
 GROUPING 函数 182
 NTILE 函数 228
 PERCENT_RANK 函数 236
 PERCENTILE_CONT 函数 237
 PERCENTILE_DISC 函数 239
 RANK 函数 247
 STDDEV 函数 282
 窗口规范 95
 窗口函数 95
 窗口函数类型 95
 窗口集合函数 93
 窗口名称 95
 分布函数 93
 排名函数 93
 数值函数 94
 统计函数 93
 行比较函数 94
OLAP OVER 子句 95
OLAP 函数
 兼容性 633
ON EXCEPTION RESUME 子句
 Transact-SQL 641
Open Client 设置 14
OR 关键字 44
OVER 子句 95

P

PATH 环境变量 12
PATINDEX 函数 234
PERCENT_RANK 函数 236
PERCENTILE_CONT 函数 237
PERCENTILE_DISC 函数 239
PI 函数 241
POWER 函数 241
PRINT 命令
 Transact-SQL 637
PROPERTY 函数 242
PROPERTY_DESCRIPTION 函数 243
PROPERTY_NAME 函数 243
PROPERTY_NUMBER 函数 244
排名函数 93
批处理
 Transact-SQL 概述 636
 编写 636

撇号

字符串中 21

平方根函数 281

平均 119

Q

QUARTER 函数 245

QUOTED_IDENTIFIER 选项 30

嵌套外连接 631

区分大小写

Transact-SQL 兼容性 622

比较条件 33

标识符 622

和模式匹配 40

口令 622

数据 622

数据库 622

用户 ID 622

用户定义数据类型 622

区域设置

设置 9

全局变量

关于 55, 57

兼容性 59

列表 58

全局唯一标识符

NEWID 函数的 SQL 语法 223

权限

Adaptive Server Enterprise 626

缺省值

CURRENT DATABASE 52

CURRENT DATE 52

CURRENT PUBLISHER 52

CURRENT TIME 52

CURRENT TIMESTAMP 53

CURRENT USER 53

LAST USER 54

TIMESTAMP 54

Transact-SQL 625

USER 55

不受支持 624

R

RADIANS 函数 246

RAISERROR 语句

ON EXCEPTION RESUME 641

Transact-SQL 641

RAND 函数 246

RANK 函数 247

REGR_AVGX 函数 249

REGR_AVGY 函数 250

REGR_COUNT 函数 251

REGR_INTERCEPT 函数 252

REGR_R2 函数 253

REGR_SLOPE 函数 254

REGR_SXX 函数 256

REGR_SXY 函数 257

REGR_SYY 函数 258

REMAINDER 函数 259

REPEAT 函数 260

REPLACE 函数 261

在 SELECT INTO 语句中 25, 137, 194, 203,
205, 213, 214, 260, 261, 263, 265,
270, 297, 308, 309

REPLICATE 函数 263

resetclocks

sp_iqcheckdb 选项 349

REVERSE 函数

SQL 语法 264

RIGHT 函数 265

ROLLUP 操作

GROUPING 函数 182

ROUND 函数 266

ROWID 函数 268

RTRIM 函数 269

日期

查询 80

将字符串解释为日期 81

确定当前日期 227, 297

算术表达式 635

日期和时间函数 99

DATE 146

DATEADD 146

DATECEILING 148

DATEDIFF 150

DATEFLOOR 152

DATEFORMAT 155

DATENAME 156

DATEPART 157

DATEROUND 158

DATETIME 161

DAY 161

DAYNAME 162

DAYS 163

DOW 170

GETDATE 179

- HOUR 187
- HOURS 188
- IQ 功能 325
- MINUTE 217
- MINUTES 218
- MONTH 220
- MONTHNAME 221
- MONTHS 221
- NOW 227
- QUARTER 245
- SECOND 270
- SECONDS 271
- TODAY 297
- WEEKS 317
- YEAR 321
- YEARS 322
- YMD 324
 - 获取一致的结果 99
- 日期和时间数据类型
 - 兼容性 619
- 日期时间和时间数据类型
 - 兼容性 620
- S**
 - sa_char_terms 存储过程 335
 - sa_checkpoint_execute 系统过程 482
 - sa_conn_list 系统过程 487
 - sa_conn_properties 系统过程 488
 - sa_external_library_unload 存储过程 336
 - sa_get_table_definition 系统过程 523
 - sa_list_external_library 存储过程 336
 - sa_nchar_terms 存储过程 336
 - sa_text_index_vocab 存储过程 336
 - SECOND 函数 270
 - SECONDS 函数 271
 - SELECT INTO
 - Transact-SQL 635
 - 使用 REPLACE 函数 25, 137, 194, 203, 205, 213, 214, 260, 261, 263, 265, 270, 297, 308, 309
 - SELECT 语句
 - Transact-SQL 629
 - SET OPTION 语句
 - Transact-SQL 628
 - SHARED 语法
 - Transact-SQL 632
 - SIGN 函数 272
 - SIGNAL 语句
 - Transact-SQL 641
 - SIMILAR 函数 273
 - SIN 函数 274
 - SMALLDATETIME 数据类型 78
 - SMALLMONEY 数据类型 72
 - SOME 条件 37
 - SORTKEY 函数 274
 - SOUNDEX 函数 278
 - sp_expireallpasswords 系统过程 338
 - sp_iq_reset_identity 系统过程 443
 - sp_iqaddlogin 系统过程 338
 - sp_iqbackupdetails 存储过程 340
 - sp_iqbackupsummary 存储过程 342
 - sp_iqbrestoreaction 存储过程 444
 - sp_iqcardinality_analysis 系统过程 343
 - sp_iqcheckdb
 - DBCC_LOG_PROGRESS 选项 351
 - resetclocks 选项 349
 - 分配模式 348
 - 检查模式 349
 - 检验模式 349
 - 沙漏模式 349
 - 输出 351
 - 输出样本 351
 - 性能 351
 - 语法 346
 - 运行时间 351
 - sp_iqcheckdb 系统过程 346
 - sp_iqcheckoptions 系统过程 352
 - sp_iqcolumn 系统过程 356
 - sp_iqcolumnuse 系统过程 358
 - sp_iqconnection 系统过程 359
 - sp_iqcontext 系统过程 364
 - sp_iqcopyloginpolicy 系统过程 366, 423
 - sp_iqcursorinfo 系统过程 367
 - sp_iqdatatype 系统过程 369
 - sp_iqdbsize 系统过程 372
 - sp_iqdbspace 系统过程 373
 - sp_iqdbspaceinfo 系统过程 376
 - sp_iqdbspaceobjectinfo 系统过程 379
 - sp_iqdbstatistics 系统过程 383
 - sp_iqdroplogin 系统过程 384
 - sp_iqemptyfile 系统过程 385
 - sp_iquestdbspaces 系统过程 387
 - sp_iquestjoin 系统过程 386
 - sp_iquestspace 系统过程 389
 - sp_iqevent 系统过程 390
 - sp_iqfile 系统过程 393
 - sp_iqhelp 系统过程 395

- sp_iqindex 系统过程 401
- sp_iqindex_alt 系统过程 401
- sp_iqindexadvice 系统过程 404
- sp_iqindexfragmentation 系统过程 406
- sp_iqindexinfo
 - 显示索引信息 408, 410
- sp_iqindexinfo 系统过程 407
- sp_iqindexmetadata 系统过程 409
- sp_iqindexsize 系统过程 410
- sp_iqindexuse 系统过程 413
- sp_iqjoinindex 系统过程 415
- sp_iqjoinindexsize 系统过程 418
- sp_iqlocks 系统过程 421
- sp_iqmodifylogin 424
- sp_iqmodifylogin 系统过程 424
- sp_iqmpxfilestatus 系统过程 427
- sp_iqmpxinconnpoolinfo 存储过程 427
- sp_iqmpxinheartbeatinfo 存储过程 427
- sp_iqmpxinfo 存储过程 427
- sp_iqmpxvalidate 存储过程 427
- sp_iqobjectinfo 系统过程 428
- sp_iqpassword 系统过程 430
- sp_iqpkeys 系统过程 431
- sp_iqprocedure 系统过程 434
- sp_iqprocparm 系统过程 436
- sp_iqrebuildindex 系统过程 439, 445
- sp_iqrename 系统过程 441
- sp_iqsetcompression 系统过程 333
- sp_iqsharedtempdistrib 过程 446
- sp_iqshowcompression 系统过程 333
- sp_iqshowpsex 系统过程 446
- sp_iqspaceinfo 系统过程 448
 - 输出样本 449
- sp_iqspaceused 系统过程 449
- sp_iqstatistics 系统过程 451
- sp_iqstatus 系统过程 453
 - 输出样本 454
- sp_iqsysmon 系统过程 455
- sp_iqtable 系统过程 460
- sp_iqtablesize 系统过程 464
- sp_iqtableuse 系统过程 465
- sp_iqtransaction 系统过程 466
- sp_iqunusedcolumn 系统过程 469
- sp_iqunusedindex 系统过程 470
- sp_iqunusedtable 系统过程 472
- sp_iqversionuse 系统过程 473
- sp_iqview 系统过程 474
- sp_iqwho 系统过程 476
- sp_iqworkmon 系统过程 479
- sp_remote_columns 系统过程 513
- sp_remote_primary_keys 系统过程
 - 语法 517
- SPACE 函数 279
- SQL
 - IQ 术语区别 325
 - 用户定义的函数 112
- SQL Anywhere 613
 - 参照完整性约束 624
 - 管理角色 616
- SQL 函数
 - ERRORMSG 函数语法 170
 - GRAPHICAL_PLAN 函数语法 180
 - GROUP_MEMBER 函数语法 183
 - GROUPING 函数语法 182
 - HTML_PLAN 函数语法 190
 - ISDATE 函数语法 196
 - ISNUMERIC 函数语法 198
 - LEN 函数语法 206
 - NEWID 函数语法 223
 - REVERSE 函数语法 264
 - STR_REPLACE 函数语法 287
 - STRTOUUID 函数语法 290
 - UUIDTOSTR 函数语法 311
 - 兼容性 632
- SQL 语法
 - CURRENT DATABASE 特殊值 52
 - CURRENT PUBLISHER 特殊值 52
 - CURRENT USER 特殊值 53
 - LAST USER 特殊值 54
 - TIMESTAMP 特殊值 54
 - USER 特殊值 55
 - 标识符 20
- SQL 语句
 - CREATE DECRYPTED DATABASE 643
 - CREATE ENCRYPTED DATABASE 643
- SQL92 一致性 325
- SQLCODE
 - 特殊值 54
- SQLCONNECT 环境变量 13
- SQLFLAGGER 函数 280
- SQLSTATE
 - 特殊值 54
- SQRT 函数 281
- SQUARE 函数 281
- STDDEV 函数 282
- STDDEV_POP 函数 284

- STDDEV_SAMP 函数 285
- STR 函数 286
- STR_REPLACE 函数
 - SQL 语法 287
- STRING 函数 289
- STRTOUUID 函数
 - SQL 语法 290
- STUFF 函数 291
- SUBSTR 函数 292
- SUBSTRING 函数 292
- SUBSTRING64 函数
 - 说明 293
- SUM 函数 293
- SUSER_ID 函数 294
- SUSER_NAME 函数 295
- Sybase IQ 用户管理
 - sp_iqdroplogin 384
- SYBASE 环境变量 13
- SYBASE_JRE 环境变量 13
- SYBASE_OCS 环境变量 14
- SYSIQBACKUPHISTORY 系统视图 553
- SYSIQBACKUPHISTORYDETAIL 系统视图 553
- SYSIQDBFILE 系统视图 554
- SYSIQDBSPACE 系统视图 555
- SYSIQIDX 系统视图 556
- SYSIQJOINIDX 系统视图 558
- SYSIQJOINIXCOLUMN 系统视图 559
- SYSIQJOINIXTABLE 系统视图 560
- SYSIQLOGICALSERVER 系统视图 560
- SYSIQLOGINPOLICYLSINFO 系统视图 560
- SYSIQLSLOGINPOLICYOPTION 系统视图 561
- SYSIQLSMEMBER 系统视图 561
- SYSIQLSPOLICY 系统视图 561
- SYSIQPARTITIONCOLUMN 系统视图 562
- SYSIQTAB 系统视图 563
- SYSPARTITION 系统视图 571
- SYSPARTITIONKEY 系统视图 571
- SYSPARTITIONSCHEME 系统视图 572
- SYSSUBPARTITIONKEY 系统视图 584
- 三值逻辑
 - 关于 45
 - 空值 62
- 沙漏模式 349
- 设备
 - 管理 615
- 时间
 - 查询 80
- 时序函数 113
- 实例
 - 外部库 336
- 实用程序
 - SQL Anywhere 642
- 事件
 - EVENT_CONDITION 函数 171
 - EVENT_CONDITION_NAME 函数 173
 - EVENT_PARAMETER 函数 173
 - 显示有关信息 390, 395
- 事务管理
 - 使用 sp_iqsysmon 监控 455
- 视图
 - 可更新的 635
 - 显示有关信息 395
- 属性
 - ID 说明 243
 - 服务器 111
 - 服务器级 242
 - 连接 110
 - 确定名称 243
 - 确定属性号 244
 - 数据库 111
- 术语
 - 拆分 335, 336
 - 行位置 335, 336
- 数据
 - 区分大小写 622
- 数据库
 - 区分大小写 622
 - 确定 ID 号 164, 226
 - 确定名称 165
 - 属性 111
 - 属性值 166
 - 系统表 525
 - 系统过程 333
- 数据库对象
 - 标识 20
 - 确定 ID 232
 - 确定名称 232
- 数据库管理员
 - 角色 616
- 数据库空间
 - 管理 615
 - 阻止读写操作 349
- 数据库选项
 - DATE_ORDER 81
 - QUOTED_IDENTIFIER 30

数据类型

- Adaptive Server Enterprise 617
- IMAGE 618, 621
- IQ 617
- LONG BINARY 618, 621
- SQL Anywhere 617
- TEXT 66, 618, 620
- UNIQUEIDENTIFIERSTR 65
- 创建 84
- 二进制 72
- 和兼容性 86
- 日期和时间 78
- 数值 68
- 显示有关信息 369, 395
- 用户定义 83
- 字符 65

数据类型兼容性

- IMAGE 数据 621
- Java 数据 621
- TEXT 数据 620
- 二进制数据 618
- 日期和时间数据 619
- 日期时间和时间数据 620
- 数值数据 620
- 位数据 617
- 字符数据 617

数据类型转换

- BIT 到 BINARY 87
- BIT 到 CHAR 88
- BIT 到 VARBINARY 87
- BIT 到 VARCHAR 88
- CHAR 到 BIT 88
- VARCHAR 到 BIT 88
- 关于 86
- 函数 98

数据类型转换函数 98

- BIGINTTOHEX 121
- CAST 124
- CONVERT 134
- HEXTOBIGINT 184
- HEXTOINT 185
- INTTOHEX 195

数学表达式 24

数值函数 94, 104

- ABS 114
- ACOS 114
- ASIN 117
- ATAN 118
- ATAN2 118

- CEIL 125
- CEILING 126
- COS 139
- COT 140
- DEGREES 167
- EXP 175
- FLOOR 179
- LOG 211
- LOG10 212
- MOD 219
- PI 241
- POWER 241
- RADIANS 246
- RAND 246
- REMAINDER 259
- ROUND 266
- SIGN 272
- SIN 274
- SQRT 281
- SQUARE 281
- TAN 296
- TRUNCNUM 298
- WIDTH_BUCKET 320
- 一致的结果 104

数值数据类型

- 兼容性 620

数字 23

- 数字的对数 211

刷新

- 立即 336

双引号

- 不允许在 SQL 标识符中使用 20
- 数据库对象 20

搜索条件

- ALL 或 ANY 条件 37
- BETWEEN 条件 38
- CONTAINS 条件 42
- EXISTS 条件 43
- IN 条件 42
- IS NULL 条件 44
- LEADING SUBSTRING SEARCH 条件 38
- LIKE 条件 38
- NOT 条件 45
- 比较条件 32
- 关于 31
- 三值逻辑 45
- 真值条件 45
- 子查询 34

索引

算术表达式 24

日期 635

索引

Adaptive Server Enterprise 626

IQ 626

SQL Anywhere 626

Transact-SQL 622

索引首选项提示 48

锁

显示 421

T

TAN 函数 296

tangent 296

TEXT 数据类型 66, 618

兼容性 618, 620

TEXT 索引

统计信息 336

THEN

IF 表达式 27

TIME 数据类型 78

TIMESTAMP

数据类型 79

数据类型兼容性 619, 620

特殊值 54

转换表达式 161

TINYINT 数据类型 70

TODAY 函数 297, 528

Transact-SQL

比较条件 33

编写可移植 SQL 628

变量 639

表达式 29

参照完整性约束 624

常量 29

创建兼容的数据库 621

概述 613

关于 613

过程 636

过程语言概述 636

结果集 638

局部变量 56

连接 631

批处理 636

外连接运算符 27

系统目录 608

用户定义数据类型 85

逐位运算符 25

字符串 30

Transact-SQL 兼容性

数据库 622

TRIM 函数 297

TRUNCNUM 函数 298

TS_ARMA_AR 函数 299

TS_ARMA_CONST 函数 299

TS_ARMA_MA 函数 299

TS_AUTO_ARIMA 函数 300

TS_AUTO_ARIMA_OUTLIER 函数 300

TS_AUTO_ARIMA_RESULT_AIC 函数 300

TS_AUTO_ARIMA_RESULT_AICC 函数 301

TS_AUTO_ARIMA_RESULT_BIC 函数 301

TS_AUTO_ARIMA_RESULT_FORECAST_ERR
OR 函数 301

TS_AUTO_ARIMA_RESULT_FORECAST_VAL
UE 函数 301

TS_AUTO_ARIMA_RESULT_MODEL_D 函数
302

TS_AUTO_ARIMA_RESULT_MODEL_P 函数
302

TS_AUTO_ARIMA_RESULT_MODEL_S 函数
302

TS_AUTO_ARIMA_RESULT_RESIDUAL_SIG
MA 函数 302

TS_AUTO_UNI_AR 函数 303

TS_AUTOCORRELATION 函数 299

TS_BOX_COX_XFORM 函数 303

TS_DIFFERENCE 函数 303

TS_DOUBLE_ARRAY 函数 303

TS_ESTIMATE_MISSING 函数 304

TS_GARCH 函数 304

TS_GARCH_RESULT_A 函数 304

TS_GARCH_RESULT_AIC 函数 304

TS_GARCH_RESULT_USER 函数 305

TS_INT_ARRAY 函数 305

TS_LACK OF FIT 函数 305

TS_LACK OF FIT_P 函数 305

TS_MAX_ARMA_AR 函数 306

TS_MAX_ARMA_CONST 函数 306

TS_MAX_ARMA_LIKELIHOOD 函数 306

TS_MAX_ARMA_MA 函数 306

TS_OUTLIER_IDENTIFICATION 函数 307

TS_PARTIAL_AUTOCORRELATION 函数 307

TS_VWAP 函数 307

特殊值

CURRENT DATABASE 52

CURRENT DATE 52

CURRENT PUBLISHER 52

- CURRENT TIME 52
- CURRENT TIMESTAMP 53
- CURRENT USER 53
- LAST USER 54
- SQLCODE 54
- SQLSTATE 54
- TIMESTAMP 54
- USER 55
- 特殊字符
 - 字符串中 21
- 提示
 - 索引首选项 48
 - 执行阶段 50, 51
- 体系结构
 - Adaptive Server Enterprise 614
 - SQL Anywhere 614
- 条件
 - 用户提供的用于查询的条件 46
- 条件提示字符串 47
- 通用唯一标识符
 - NEWID 函数的 SQL 语法 223
- 统计函数 93
- 统计信息
 - TEXT 索引 336

U

- UCASE 函数 308
- UNION
 - 在子查询中 630
- UNIQUEIDENTIFIER 数据类型 76
- UNIQUEIDENTIFIERSTR 数据类型
 - 关于 65
- UPPER 函数 309
- USER
 - 特殊常量 528
 - 特殊值 55
- USER_ID 函数 310
- USER_NAME 函数 310
- UUID
 - NEWID 函数的 SQL 语法 223
 - STRTOUUID 函数的 SQL 语法 290
 - UIDTOSTR 函数的 SQL 语法 311
- UIDTOSTR 函数
 - SQL 语法 311

V

- VAR_POP 函数 312

- VAR_SAMP 函数 314
- VARBINARY 数据类型 73
- VARCHAR 数据类型
 - 关于 65
 - 剪裁尾随空白 67, 617
- VARIANCE 函数 315

W

- WEEKS 函数 317
- WEIGHTED_AVG 函数 318
- WHERE 子句
 - Transact-SQL 631
- WIDTH_BUCKET 函数 320
- WITHIN GROUP 子句 97
- 外部库
 - 列表 336
 - 卸载 336
- 外连接
 - Transact-SQL 631
 - 和子查询 24
 - 链式 631
 - 嵌套 631
 - 运算符 27
- 尾随空白
 - 剪裁 67, 617
- 位长度 121
- 谓词
 - 分离 34
 - 关于 31
- 文件
 - 位置 4
- 文字
 - 最大长度 329

X

- 系统安全员
 - Adaptive Server Enterprise 616
- 系统变量 57
- 系统表
 - Adaptive Server Enterprise 兼容性 615
 - DUMMY 528
 - 关于 525
 - 显示信息 395
- 系统管理员
 - Adaptive Server Enterprise 616
- 系统过程
 - sa_checkpoint_execute 482

sa_conn_list 487
 sa_conn_properties 488
 sa_get_table_definition 523
 sp_expireallpasswords 338
 sp_iqaddlogin 338
 sp_iqbackupdetails 340
 sp_iqbackupsummary 342
 sp_iqcardinality_analysis 343
 sp_iqcheckdb 346
 sp_iqcheckoptions 352
 sp_iqclient_lookup 354
 sp_iqcolumn 356
 sp_iqcolumnuse 358
 sp_iqconnection 359
 sp_iqcontext 364
 sp_iqcopyloginpolicy 366, 423
 sp_iqcursinfo 367
 sp_iqdatatype 369
 sp_iqdbsize 372
 sp_iqdbspaceobjectinfo 379
 sp_iqdbstatistics 383
 sp_iqdroplogin 384
 sp_iqemptyfile 385
 sp_iquestdbspaces 387
 sp_iquestjoin 386
 sp_iquestspace 389
 sp_iqevent 390
 sp_iqfile 393
 sp_iqhelp 395
 sp_iqindex 401
 sp_iqindex_alt 401
 sp_iqindexadvice 404
 sp_iqindexsize 410
 sp_iqindexuse 413
 sp_iqjoinindex 415
 sp_iqjoinindexsize 418
 sp_iqmodifylogin 424
 sp_iqmpxfilestatus 427
 sp_iqmpxinconnpoolinfo 427
 sp_iqmpxinheartbeatinfo 427
 sp_iqmpxinfo 427
 sp_iqmpxvalidate 427
 sp_iqobjectinfo 428
 sp_iqpassword 430
 sp_iqpkeys 431
 sp_iqprocedure 434
 sp_iqprocparm 436
 sp_iqrename 441
 sp_iqrestoreaction 444

sp_iqsetcompression 333
 sp_iqsharedtempdistrib 446
 sp_iqshowcompression 333
 sp_iqshowpsex 446
 sp_iqspaceinfo 448
 sp_iqspaceused 449
 sp_iqstatistics 451
 sp_iqstatus 453
 sp_iqsysmon 455
 sp_iqtable 460
 sp_iqtablesize 464
 sp_iqtableuse 465
 sp_iqtransaction 466
 sp_iqunusedcolumn 469
 sp_iqunusedindex 470
 sp_iqunusedtable 472
 sp_iqversionuse 473
 sp_iqview 474
 sp_iqwho 476
 sp_iqworkmon 479
 sp_remote_columns 513
 sp_remote_primary_keys 517
 关于 333
 显示有关信息 395

系统函数 107

COL_LENGTH 131
 COL_NAME 132
 CONNECTION_PROPERTY 133
 DATALENGTH 145
 DB_ID 164
 DB_NAME 165
 DB_PROPERTY 166
 EVENT_CONDITION 171
 EVENT_CONDITION_NAME 173
 EVENT_PARAMETER 173
 INDEX_COL 193
 NEXT_CONNECTION 224
 NEXT_DATABASE 226
 OBJECT_ID 232
 OBJECT_NAME 232
 PROPERTY 242
 PROPERTY_DESCRIPTION 243
 PROPERTY_NAME 243
 PROPERTY_NUMBER 244
 USER_ID 294
 USER_NAME 295
 Transact-SQL 634
 USER_ID 310
 USER_NAME 310

系统目录

- Adaptive Server Enterprise 兼容性 615
- Transact-SQL 608

系统视图

- Adaptive Server Enterprise 644
- SYSCOLUMNS ASE 兼容性视图 537
- SYSDSPACEPERM 539
- SYSEVENTTYPE 541
- SYSINDEXES ASE 兼容性视图 552
- SYSIQBACKUPHISTORY 553
- SYSIQBACKUPHISTORYDETAIL 553
- SYSIQCOLUMN 554
- SYSIQDBFILE 554
- SYSIQDBSPACE 555
- SYSIQFILE 556
- SYSIQIDX 556
- SYSIQINFO 557
- SYSIQITAB 563
- SYSIQJOINIDX 558
- SYSIQJOININDEX 559
- SYSIQJOINIXCOLUMN 559
- SYSIQJOINIXTABLE 560
- SYSIQLLOGICALSERVER 系统视图 560
- SYSIQLLOGINPOLICYLSINFO 560
- SYSIQLSLOGINPOLICIES 561
- SYSIQLSLOGINPOLICYOPTION 561
- SYSIQLSMEMBER 561
- SYSIQLSMEMBERS 561
- SYSIQLSPOLICY 561
- SYSIQMPXSERVER 561
- SYSIQOBJECTS ASE 兼容性视图 562
- SYSIQPARTITIONCOLUMN 562
- SYSIQTABCOL 563
- SYSIQTABLE 系统视图 564
- SYSIQVINDEXT ASE 兼容性视图 564
- SYSLOGINS ASE 兼容性视图 568
- SYSOBJECTS ASE 兼容性视图 569
- SYSPARTITION 571
- SYSPARTITIONKEY 571
- SYSPARTITIONSCHEME 572
- SYSREMOPTION 579
- SYSSCHEDULE 583
- SYSSUBPARTITIONKEY 584
- SYSTYPES ASE 兼容性视图 601
- SYSUSERS ASE 兼容性视图 606
- SYSVIEW 606
- 统一 529

线程

- dbcc 348

卸载

- 外部库 336

- 星期几 (DOW) 170

行

- 计数 143

性能

- sp_iqshowpexe 连接信息 446
- sp_iqsysmon 过程 455
- 监控 455

虚拟 IQ 表 91

- 获取一致的结果 99

选项

- DBCC_LOG_PROGRESS 351
- QUOTED_IDENTIFIER 30
- SQL Anywhere 642

选择性

- 提示 47
- 显式 46
- 用户提供的条件 46

Y

- YEAR 函数 321

- YEARS 函数 322

- YMD 函数 324

- 样本方差函数 314

一致性检查

- 分区 347

一致状态

- 恢复 444

引号

- SQL 标识符 20
- 数据库对象 20
- 字符串 21

用户

- 删除 384
- 添加 338
- 显示有关信息 476
- 修改 424

用户 ID

- Adaptive Server Enterprise 626
- 区分大小写 622
- 通过用户名确定 294, 310

用户定义的函数 112

- 兼容性 634

用户定义数据类型

- CREATE DOMAIN 语句 84
- Transact-SQL 85
- 关于 83

索引

- 区分大小写 622
- 用户管理
 - 请参见 登录管理
- 用户名
 - 通过用户 ID 确定 295, 310
- 用户提供的关于连接等同性条件的提示 51
- 用户提供的条件
 - 查询 46
- 用户提供的条件提示、准则和用法 51
- 用户提供的条件提示字符串 47
- 用户提供的条件选择性 46
- 优化程序
 - 估计值 46
 - 用户定义的选择性 46
- 游标
 - IQ 中的行级 637
 - Transact-SQL 637
 - 显示有关信息 367
- 有用性提示 51
- 余切值 140
- 语法
 - CURRENT DATABASE 特殊值 52
 - CURRENT PUBLISHER 特殊值 52
 - CURRENT USER 特殊值 53
 - LAST USER 特殊值 54
 - SQL 标识符 20
 - TIMESTAMP 特殊值 54
 - USER 特殊值 55
- 语句
 - CREATE DEFAULT 625
 - CREATE DOMAIN 625
 - CREATE RULE 625
 - CREATE TABLE 623
 - DISK INIT 615
 - DISK MIRROR 615
 - DISK REFIT 615
 - DISK REINIT 615
 - DISK REMIRROR 615
 - DISK UNMIRROR 615
 - RAISERROR 641
 - SELECT 629
 - SIGNAL 641
- 语言
 - 指定 8
- 预取
 - 使用 sp_iqsysmon 监控 455
- 域 84
 - 关于 83

- 远程表
 - 列 513
- 运算符
 - 比较运算符 32
 - 优先级 27
- 运算符的优先级 27

Z

- 杂项函数 113
 - ARGN 115
 - COALESCE 130
 - IFNULL 193
 - ISNULL 197
 - NULLIF 229
 - NUMBER 231
 - ROWID 268
 - SQLFLAGGER 280
- 摘要 340, 427
- 执行阶段提示 50
- 指数函数 175
- 中括号
 - SQL 标识符 20
 - 数据库对象 20
- 重命名对象
 - sp_iqrename 过程 441
- 逐位运算符 25
- 主键
 - UUID 和 GUID 223
 - 生成唯一值 223
 - 使用 UUID 生成唯一值 223
 - 显示有关信息 431
- 主数据库
 - 不支持 614
- 注册表条目
 - 关于 14
- 注释
 - 注释指示符 61
- 装载格式
 - Transact-SQL 和 SQL Anywhere 628
- 子查询
 - Adaptive Server Enterprise 630
 - IQ 630
 - IQ 实现 326
 - SQL Anywhere 630
 - 表达式中 24
 - 分离 34
 - 搜索条件中的 34

子句

ON EXCEPTION RESUME 641

字符串 21, 23

SOUNDEX 函数 278

Transact-SQL 30

并置 25, 263, 289

并置运算符 25

插入 194

长度 122, 128

常量 21, 23

返回子字符串 292

分隔符 30

关于 21

决定长度 207, 209

确定相似性 273

删除空白 297

删除前导空白 214

删除尾随空白 269

特殊字符 21

替换子字符串 261

位置 129

文字字符串 21

转换为大写形式 308

转换为小写 203, 213

字符串函数 105

ASCII 116

BFILE 120

BIT_LENGTH 121

BYTE_LENGTH 122

BYTE_LENGTH64 124

BYTE_SUBSTR 124

BYTE_SUBSTR64 124

CHAR 127

CHAR_LENGTH 128

CHAR_LENGTH64 129

CHARINDEX 129

DIFFERENCE 169

INSERTSTR 194

LCASE 203

LEFT 205

LENGTH 207, 209

LOCATE 210

LOWER 213

LTRIM 214

OCTET_LENGTH 233

PATINDEX 234

REPEAT 260

REPLACE 261

REPLICATE 263

RIGHT 265

RTRIM 269

SIMILAR 273

SORTKEY 274

SOUNDEX 278

SPACE 279

STR 286

STRING 289

STUFF 291

SUBSTR 292

SUBSTRING 292

SUBSTRING64 293

TRIM 297

UCASE 308

UPPER 309

字符串文字

最大长度 329

字符集

指定 7

字符数据

兼容性 617

字节长度 233

字母字符

定义 20

自动连接

和外键 325

总体方差函数 312

组

Adaptive Server Enterprise 626

