

SYBASE®

参考：构件块、表和过程

Sybase IQ

15.2

文档 ID: DC01134-01-1520-01

最后修订日期: 2010 年 4 月

版权所有 © 2010 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协议提供, 其使用或复制必须符合许可条款。

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、传播或翻译本手册的任何部分。

可在位于 <http://www.sybase.com/detail?id=1011207> 的“Sybase 商标页” (Sybase trademarks page) 中查看 Sybase 商标。Sybase 和所列标记均为 Sybase, Inc. 的商标。® 表示已在美国注册。

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

关于本手册	xix	
第 1 章	文件位置和安装设置	1
	安装目录结构	1
	Sybase IQ 如何定位文件	2
	简单文件搜索	3
	广泛文件搜索	4
	环境变量	4
	设置环境变量	5
	IQDIR15 环境变量	7
	IQPORT 环境变量	7
	IQLOGDIR15 环境变量	8
	IQTIMEOUT 环境变量	8
	IQTMP15 环境变量	9
	JAVA_HOME 环境变量	10
	LIBPATH 或 LD_LIBRARY_PATH 环境变量	10
	PATH 环境变量	10
	SACHARSET 环境变量	11
	SALANG 环境变量	11
	SQLCONNECT 环境变量	11
	SYBASE 环境变量	12
	\$SYBASE_JRE6_32、\$SYBASE_JRE6_64 和 \$SYBASE_JRE5_64 环境变量	12
	SYBASE_OCS 环境变量	13
	注册表条目	13
	当前用户和本地计算机设置	13
	注册表结构	14
	安装时的注册表设置	14

第 2 章	SQL 语言元素	15
	关键字	15
	保留字	16
	标识符	18
	字符串	20
	表达式	21
	表达式中的常量	22
	表达式中的列名	22
	表达式中的子查询	22
	SQL 运算符	23
	IF 表达式	26
	CASE 表达式	26
	表达式的兼容性	27
	搜索条件	30
	比较条件	31
	搜索条件中的子查询	32
	ALL 或 ANY 条件	36
	BETWEEN 条件	37
	LIKE 条件	37
	IN 条件	41
	CONTAINS 条件	41
	EXISTS 条件	42
	IS NULL 条件	42
	带有逻辑运算符的条件	42
	NOT 条件	43
	真值条件	43
	三值逻辑	43
	用户提供的条件提示	44
	特殊值	49
	CURRENT DATABASE 特殊值	49
	CURRENT DATE 特殊值	49
	CURRENT PUBLISHER 特殊值	49
	CURRENT TIME 特殊值	50
	CURRENT TIMESTAMP 特殊值	50
	CURRENT USER 特殊值	50
	LAST USER 特殊值	51
	SQLCODE 特殊值	51
	SQLSTATE 特殊值	51
	TIMESTAMP 特殊值	52
	USER 特殊值	52
	变量	53
	局部变量	53
	连接级变量	55
	全局变量	55
	注释	59
	NULL 值	60

第 3 章	SQL 数据类型	63
	字符数据类型	63
	数值数据类型	67
	二进制数据类型	71
	Bit 数据类型	76
	日期和时间数据类型	76
	向数据库发送日期和时间	78
	从数据库中检索日期和时间	79
	比较日期和时间	79
	使用明确的日期和时间	80
	域	81
	数据类型转换	83
第 4 章	SQL 函数	89
	概述	89
	集合函数	90
	分析函数	92
	数据类型转换函数	96
	日期和时间函数	97
	日期分量	100
	HTTP 函数	101
	数值函数	102
	字符串函数	103
	系统函数	105
	连接属性	107
	可用于服务器的属性	108
	可用于每个数据库的属性	108
	SQL 和 Java 用户定义的函数	109
	时序和预测函数	110
	杂项函数	110
	按字母顺序排列的函数列表	111
	ABS 函数 [数值]	111
	ACOS 函数 [数值]	111
	ARGN 函数 [杂项]	112
	ASCII 函数 [字符串]	112
	ASIN 函数 [数值]	113
	ATAN 函数 [数值]	113
	ATAN2 函数 [数值]	114
	AVG 函数 [集合]	114
	BFILE 函数 [数据提取]	115
	BIGINTTOHEX 函数 [数据类型转换]	115
	BIT_LENGTH 函数 [字符串]	116
	BYTE_LENGTH 函数 [字符串]	116
	BYTE_LENGTH64 函数	117

BYTE_SUBSTR64 和 BYTE_SUBSTR 函数	117
CAST 函数 [数据类型转换]	117
CEIL 函数 [数值]	118
CEILING 函数 [数值]	119
CHAR 函数 [字符串]	119
CHAR_LENGTH 函数 [字符串]	120
CHAR_LENGTH64 函数	120
CHARINDEX 函数 [字符串]	121
COALESCE 函数 [杂项]	122
COL_LENGTH 函数 [系统]	122
COL_NAME 函数 [系统]	122
CONNECTION_PROPERTY 函数 [系统]	123
CONVERT 函数 [数据类型转换]	124
CORR 函数 [集合]	127
COS 函数 [数值]	128
COT 函数 [数值]	128
COVAR_POP 函数 [集合]	129
COVAR_SAMP 函数 [集合]	130
COUNT 函数 [集合]	131
CUME_DIST 函数 [分析]	131
DATALLENGTH 函数 [系统]	132
DATE 函数 [日期和时间]	133
DATEADD 函数 [日期和时间]	133
DATECEILING 函数 [日期和时间]	134
DATEDIFF 函数 [日期和时间]	137
DATEFLOOR 函数 [日期和时间]	139
DATEFORMAT 函数 [日期和时间]	141
DATENAME 函数 [日期和时间]	142
DATEPART 函数 [日期和时间]	143
DATEROUND 函数 [日期和时间]	144
DATETIME 函数 [日期和时间]	146
DAY 函数 [日期和时间]	146
DAYNAME 函数 [日期和时间]	147
DAYS 函数 [日期和时间]	147
DB_ID 函数 [系统]	148
DB_NAME 函数 [系统]	148
DB_PROPERTY 函数 [系统]	149
DEGREES 函数 [数值]	150
DENSE_RANK 函数 [分析]	150
DIFFERENCE 函数 [字符串]	151
DOW 函数 [日期和时间]	152
ERRORMSG 函数 [杂项]	152
EVENT_CONDITION 函数 [系统]	153
EVENT_CONDITION_NAME 函数 [系统]	154

EVENT_PARAMETER 函数 [系统]	155
EXP 函数 [数值]	156
EXP_WEIGHTED_AVG 函数 [集合]	156
FIRST_VALUE 函数 [集合]	157
FLOOR 函数 [数值]	159
GETDATE 函数 [日期和时间]	159
GRAPHICAL_PLAN 函数 [字符串]	160
GROUPING 函数 [集合]	162
GROUP_MEMBER 函数 [系统]	162
HEXTOBIGINT 函数 [数据类型转换]	163
HEXTOINT 函数 [数据类型转换]	164
HOUR 函数 [日期和时间]	165
HOURS 函数 [日期和时间]	165
HTML_DECODE 函数 [HTTP]	166
HTML_ENCODE 函数 [HTTP]	166
HTML_PLAN 函数 [字符串]	167
HTTP_DECODE 函数 [HTTP]	169
HTTP_ENCODE 函数 [HTTP]	169
HTTP_HEADER 函数 [HTTP]	169
HTTP_VARIABLE 函数 [HTTP]	170
IFNULL 函数 [杂项]	170
INDEX_COL 函数 [系统]	171
INSERTSTR 函数 [字符串]	171
INTTOHEX 函数 [数据类型转换]	172
ISDATE 函数 [日期和时间]	173
ISNULL 函数 [杂项]	174
ISNUMERIC 函数 [杂项]	174
LAG 函数 [分析]	175
LAST_VALUE 函数 [集合]	176
LCASE 函数 [字符串]	178
LEAD 函数 [分析]	179
LEFT 函数 [字符串]	180
LEN 函数 [字符串]	181
LENGTH 函数 [字符串]	181
LN 函数 [数值]	182
LOCATE 函数 [字符串]	182
LOG 函数 [数值]	184
LOG10 函数 [数值]	184
LOWER 函数 [字符串]	184
LTRIM 函数 [字符串]	185
MAX 函数 [集合]	186
MEDIAN 函数 [集合]	186
MIN 函数 [集合]	187
MINUTE 函数 [日期和时间]	188

MINUTES 函数 [日期和时间]	188
MOD 函数 [数值]	189
MONTH 函数 [日期和时间]	190
MONTHNAME 函数 [日期和时间]	190
MONTHS 函数 [日期和时间]	191
NEWID 函数 [杂项]	192
NEXT_CONNECTION 函数 [系统]	193
NEXT_DATABASE 函数 [系统]	194
NEXT_HTTP_HEADER 函数 [HTTP]	195
NEXT_HTTP_VARIABLE 函数 [HTTP]	195
NOW 函数 [日期和时间]	195
NTILE 函数 [分析]	196
NULLIF 函数 [杂项]	197
NUMBER 函数 [杂项]	198
OBJECT_ID 函数 [系统]	199
OBJECT_NAME 函数 [系统]	199
OCTET_LENGTH 函数 [字符串]	200
PATINDEX 函数 [字符串]	200
PERCENT_RANK 函数 [分析]	201
PERCENTILE_CONT 函数 [分析]	203
PERCENTILE_DISC 函数 [分析]	205
PI 函数 [数值]	207
POWER 函数 [数值]	207
PROPERTY 函数 [系统]	207
PROPERTY_DESCRIPTION 函数 [系统]	208
PROPERTY_NAME 函数 [系统]	209
PROPERTY_NUMBER 函数 [系统]	209
QUARTER 函数 [日期和时间]	210
RADIANS 函数 [数值]	210
RAND 函数 [数值]	211
RANK 函数 [分析]	211
REGR_AVGX 函数 [集合]	213
REGR_AVGY 函数 [集合]	214
REGR_COUNT 函数 [集合]	215
REGR_INTERCEPT 函数 [集合]	215
REGR_R2 函数 [集合]	217
REGR_SLOPE 函数 [集合]	218
REGR_SXX 函数 [集合]	219
REGR_SXY 函数 [集合]	220
REGR_SYY 函数 [集合]	221
REMAINDER 函数 [数值]	222
REPEAT 函数 [字符串]	222
REPLACE 函数 [字符串]	223
REPLICATE 函数 [字符串]	224

REVERSE 函数 [字符串]	225
RIGHT 函数 [字符串]	225
ROUND 函数 [数值]	226
ROW_NUMBER 函数 [分析]	227
ROWID 函数 [杂项]	228
RTRIM 函数 [字符串]	229
SECOND 函数 [日期和时间]	229
SECONDS 函数 [日期和时间]	230
SIGN 函数 [数值]	231
SIMILAR 函数 [字符串]	231
SIN 函数 [数值]	232
SORTKEY 函数 [字符串]	232
SOUNDEX 函数 [字符串]	236
SPACE 函数 [字符串]	236
SQLFLAGGER 函数 [杂项]	237
SQRT 函数 [数值]	238
SQUARE 函数 [数值]	238
STDDEV 函数 [集合]	238
STDDEV_POP 函数 [集合]	240
STDDEV_SAMP 函数 [集合]	241
STR 函数 [字符串]	242
STR_REPLACE 函数 [字符串]	242
STRING 函数 [字符串]	244
STRTOUUID 函数 [字符串]	244
STUFF 函数 [字符串]	245
SUBSTRING 函数 [字符串]	246
SUBSTRING64 函数	247
SUM 函数 [集合]	247
SUSER_ID 函数 [系统]	248
SUSER_NAME 函数 [系统]	248
TAN 函数 [数值]	249
TODAY 函数 [日期和时间]	249
TRIM 函数 [字符串]	249
TRUNCNUM 函数 [数值]	250
TS_ARMA_AR 函数 [时序]	251
TS_ARMA_CONST 函数 [时序]	251
TS_ARMA_MA 函数 [时序]	251
TS_AUTOCORRELATION 函数 [时序]	252
TS_AUTO_ARIMA 函数 [时序]	252
TS_AUTO_ARIMA_OUTLIER 函数 [时序]	252
TS_AUTO_ARIMA_RESULT_AIC 函数 [时序]	253
TS_AUTO_ARIMA_RESULT_AICC 函数 [时序]	253
TS_AUTO_ARIMA_RESULT_BIC 函数 [时序]	253

TS_AUTO_ARIMA_RESULT_FORECAST_VALUE 函数 [时序]	254
TS_AUTO_ARIMA_RESULT_FORECAST_ERROR 函数 [时序]	254
TS_AUTO_ARIMA_RESULT_MODEL_D 函数 [时序]	254
TS_AUTO_ARIMA_RESULT_MODEL_P 函数 [时序]	255
TS_AUTO_ARIMA_RESULT_MODEL_Q [时序]	255
TS_AUTO_ARIMA_RESULT_MODEL_S 函数 [时序]	255
TS_AUTO_ARIMA_RESULT_RESIDUAL_SIGMA [时序]	256
TS_AUTO_UNI_AR 函数 [时序]	256
TS_BOX_COX_XFORM 函数 [时序]	256
TS_DIFFERENCE 函数 [时序]	257
TS_DOUBLE_ARRAY [时序]	257
TS_ESTIMATE_MISSING 函数 [时序]	257
TS_GARCH 函数 [时序]	258
TS_GARCH_RESULT_A 函数 [时序]	258
TS_GARCH_RESULT_AIC 函数 [时序]	258
TS_GARCH_RESULT_USER [时序]	259
TS_INT_ARRAY [时序]	259
TS_LACK_OF_FIT 函数 [时序]	259
TS_LACK_OF_FIT_P 函数 [时序]	260
TS_MAX_ARMA_AR 函数 [时序]	260
TS_MAX_ARMA_CONST 函数 [时序]	260
TS_MAX_ARMA_LIKELIHOOD 函数 [时序]	261
TS_MAX_ARMA_MA 函数 [时序]	261
TS_OUTLIER_IDENTIFICATION 函数 [时序]	261
TS_PARTIAL_AUTOCORRELATION 函数 [时序]	262
TS_VWAP 函数 [时序]	262
UCASE 函数 [字符串]	262
UPPER 函数 [字符串]	263
USER_ID 函数 [系统]	264
USER_NAME 函数 [系统]	264
UIDTOSTR 函数 [字符串]	265
VAR_POP 函数 [集合]	265
VAR_SAMP 函数 [集合]	266
VARIANCE 函数 [集合]	267
WEEKS 函数 [日期和时间]	269
WEIGHTED_AVG 函数 [集合]	270
WIDTH_BUCKET 函数 [数值]	271
YEAR 函数 [日期和时间]	273
YEARS 函数 [日期和时间]	273
YMD 函数 [日期和时间]	275

第 5 章	与其它 SQL 术语的区别	277
	Sybase IQ 功能	277
第 6 章	物理限制	281
	大小和数量限制	281
第 7 章	系统过程	283
	系统过程概述	283
	存储过程的语法规则	284
	了解存储过程报告的统计信息	284
	系统存储过程	285
	sa_char_terms 系统过程	285
	sa_dependent_views 过程	285
	sa_external_library_unload 过程	285
	sa_list_external_library 过程	286
	sa_nchar_terms 系统过程	286
	sa_text_index_vocab 系统过程	286
	sa_verify_password 过程	286
	sa_get_user_status 过程	286
	sp_expireallpasswords 过程	287
	sp_iqaddlogin 过程	287
	sp_iqbackupdetails 过程	288
	sp_iqbackupsummary 过程	290
	sp_iqcardinality_analysis 过程	291
	sp_iqcheckdb 过程	293
	sp_iqcheckoptions 过程	300
	sp_iqclient_lookup 过程	302
	sp_iqcolumn 过程	303
	sp_iqcolumnuse 过程	305
	sp_iqconnection 过程	306
	sp_iqconstraint 过程	310
	sp_iqcontext 过程	311
	sp_iqcopyloginpolicy 过程	313
	sp_iqcursorinfo 过程	313
	sp_iqdatatype 过程	316
	sp_iqdbsize 过程	318
	sp_iqdbspace 过程	320
	sp_iqdbspaceinfo 过程	322
	sp_iqdbspaceobjectinfo 过程	325
	sp_iqdbstatistics 过程	329
	sp_iqdroplogin 过程	330
	sp_iqemptyfile 过程	331
	sp_iqestjoin 过程	332

sp_iqestdbspaces 过程	333
sp_iqestspace 过程	335
sp_iqevent 过程	335
sp_iqfile 过程	338
sp_iqhelp 过程	340
sp_iqindex 和 sp_iqindex_alt 过程	346
sp_iqindexadvice 过程	348
sp_iqindexfragmentation 过程	349
sp_iqindexinfo 过程	351
sp_iqindexmetadata 过程	352
sp_iqindexsize 过程	354
sp_iqindexuse 过程	355
sp_iqjoinindex 过程	356
sp_iqjoinindexsize 过程	359
sp_iqlmconfig 过程	360
sp_iqllocks 过程	361
sp_iqmodifyadmin 过程	363
sp_iqmodifylogin 过程	364
sp_iqmpxinconnpoolinfo 过程	364
sp_iqmpxinheartbeatinfo 过程	365
sp_iqmpxinfo 过程	365
sp_iqmpxvalidate 过程	365
sp_iqmpxversioninfo 过程	365
sp_iqobjectinfo 过程	365
sp_iqpassword 过程	368
sp_iqpkeys 过程	368
sp_iqprocedure 过程	370
sp_iqprocparm 过程	372
sp_iqrebuildindex 过程	375
sp_iqrename 过程	377
sp_iq_reset_identity 过程	379
sp_iqrestoreaction 过程	379
sp_iqrowdensity 过程	380
sp_iqshowpsexex 过程	381
sp_iqspaceinfo 过程	383
sp_iqspaceused 过程	384
sp_iqstatistics 过程	385
sp_iqstatus 过程	387
sp_iqsysmon 过程	389
sp_iqtable 过程	395
sp_iqtablesize 过程	397
sp_iqtableuse 过程	399
sp_iqtransaction 过程	399
sp_iqunusedcolumn 过程	402

sp_iqunusedindex 过程	403
sp_iqunusedtable 过程	405
sp_iqversionuse 过程	406
sp_iqview 过程	407
sp_iqwho 过程	408
sp_iqworkmon 过程	411
目录存储过程	413
sa_ansi_standard_packages 过程	413
sa_audit_string 系统过程	413
sa_checkpoint_execute 系统过程	413
sa_conn_activity 系统过程	414
sa_conn_info 系统过程	414
sa_conn_properties 系统过程	415
sa_db_info 系统过程	415
sa_db_properties 系统过程	415
sa_disable_auditing_type 系统过程	415
sa_enable_auditing_type 系统过程	416
sa_eng_properties 系统过程	416
sa_flush_cache 系统过程	416
sa_make_object 系统过程	416
sa_rowgenerator 系统过程	417
sa_server_option 系统过程	417
sa_set_http_header 系统过程	417
sa_set_http_option 系统过程	418
sa_table_page_usage 系统过程	418
sa_validate 系统过程	418
sa_verify_password 系统过程	418
sp_login_environment 系统过程	419
sp_remote_columns 系统过程	419
sp_remote_exported_keys 系统过程	419
sp_remote_imported_keys 系统过程	420
sp_remote_primary_keys 系统过程	420
sp_remote_tables 系统过程	420
sp_servercaps 系统过程	421
sp_tsql_environment 系统过程	421
Adaptive Server Enterprise 系统和目录存储过程	421
Adaptive Server Enterprise 系统过程	422
Adaptive Server Enterprise 目录存储过程	423
SQL Anywhere 支持的过程	424

第 8 章	系统表和视图	425
	系统表	429
	系统表列表	429
	DUMMY 系统表	432
	系统视图	432
	SYSARTICLE 系统视图	433
	SYSARTICLECOL 系统视图	433
	SYSARTICLECOLS 统一视图	433
	SYSARTICLES 统一视图	433
	SYSCAPABILITIES 统一视图	434
	SYSCAPABILITY 系统视图	434
	SYSCAPABILITYNAME 系统视图	434
	SYSCATALOG 统一视图	434
	SYSCHECK 系统视图	435
	SYSCOLAUTH 统一视图	435
	SYSCOLLATION 兼容性视图（不建议使用）	435
	SYSCOLLATIONMAPPINGS 兼容性视图（不建议使用）	435
	SYSCOLPERM 系统视图	436
	SYSCOLSTAT 系统视图	436
	SYSCOLSTATS 统一视图	436
	SYSCOLUMN 兼容性视图（不建议使用）	437
	SYSCOLUMNS 统一视图	437
	SYSCOLUMNS ASE 兼容性视图	437
	SYSCOMMENTS ASE 兼容性视图	437
	SYSCONSTRAINT 系统视图	438
	SYSDBFIELD 系统视图	438
	SYSDBSpace 系统视图	438
	SYSDBSpacePERM 系统视图	439
	SYSDEPENDENCY 系统视图	439
	SYSDOMAIN 系统视图	439
	SYSEVENT 系统视图	440
	SYSEVENTTYPE 系统视图	440
	SYSEXTERNENV 系统视图	440
	SYSEXTERNENVOBJECT 系统视图	441
	SYSEXTERNLOGIN 系统视图	441
	SYSFILE 兼容性视图（不建议使用）	441
	SYSFKCOL 兼容性视图（不建议使用）	442
	SYSFKKEY 系统视图	442
	SYSFOREIGNKEY 兼容性视图（不建议使用）	442
	SYSFOREIGNKEYS 统一视图	442
	SYSGROUP 系统视图	443
	SYSGROUPS 统一视图	443
	SYSHISTORY 系统视图	443
	SYSIDX 系统视图	444

SYSIDXCOL 系统视图	444
SYSINDEX 兼容性视图（不建议使用）	444
SYSINDEXES 统一视图	445
SYSINDEXES ASE 兼容性视图	445
SYSINFO 兼容性视图（不建议使用）	445
SYSIQBACKUPHISTORY 系统视图	445
SYSIQBACKUPHISTORYDETAIL 系统视图	446
SYSIQCOLUMN 系统视图（不建议使用）	447
SYSIQDBFILE 系统视图	447
SYSIQDBSPACE 系统视图	448
SYSIQFILE 系统视图（不建议使用）	449
SYSIQIDX 系统视图	449
SYSIQINFO 系统视图	450
SYSIQJOINIDX 系统视图	451
SYSIQJOININDEX 系统视图（不建议使用）	452
SYSIQJOINIXCOLUMN 系统视图	452
SYSIQJOINIXTABLE 系统视图	453
SYSIQMPXLOGINPOLICYOPTION 系统视图	453
SYSIQMPXSERVER 系统视图	453
SYSIQOBJECTS ASE 兼容性视图	454
SYSIQPARTITIONCOLUMN 系统视图	454
SYSIQTAB 系统视图	454
SYSIQTABCOL 系统视图	455
SYSIQTABLE 系统视图（不建议使用）	456
SYSIQVINDEX ASE 兼容性视图	456
SYSIXCOL 兼容性视图（不建议使用）	456
SYSJAR 系统视图	457
SYSJARCOMPONENT 系统视图	457
SYSJAVACLASS 系统视图	457
SYSLOGINMAP 系统视图	457
SYSLOGINPOLICY 系统视图	458
SYSLOGINPOLICYOPTION 系统视图	458
SYSLOGINS ASE 兼容性视图	458
SYSMVOPTION 系统视图	458
SYSMVOPTIONNAME 系统视图	459
SYSOBJECT 系统视图	459
SYSOBJECTS ASE 兼容性视图	459
SYSOPTION 系统视图	459
SYSOPTIONS 统一视图	460
SYSOPTSTAT 系统视图	460
SYSPARTITION 系统视图	460
SYSPARTITIONKEY 系统视图	461
SYSPARTITIONSCHEME 系统视图	461
SYSPHYSIDX 系统视图	462

SYSPROCAUTH 统一视图	462
SYSPROCEDURE 系统视图	462
SYSPROCPARM 系统视图	463
SYSPROCPARMS 统一视图	463
SYSPROCPERM 系统视图	463
SYSPROCS 统一视图	463
SYSPROXYTAB 系统视图	464
SYSPUBLICATION 系统视图	464
SYSPUBLICATIONS 统一视图	464
SYSREMARK 系统视图	464
SYSREMOTEOPTION 系统视图	465
SYSREMOTEOPTION2 统一视图	465
SYSREMOTEOPTIONS 统一视图	465
SYSREMOTEOPTIONTYPE 系统视图	466
SYSREMOTETYPE 系统视图	466
SYSREMOTETYPES 统一视图	466
SYSREMOTEUSER 系统视图	466
SYSREMOTEUSERS 统一视图	467
SYSSCHEDULE 系统视图	467
SYSSERVER 系统视图	467
SYSSOURCE 系统视图	467
SYSSQLSERVERTYPE 系统视图	468
SYSSUBPARTITIONKEY 系统视图	468
SYSSUBSCRIPTION 系统视图	468
SYSSUBSCRIPTIONS 统一视图	468
SYSSYNC 系统视图	469
SYSSYNC2 统一视图	469
SYSSYNCPUBLICATIONDEFAULTS 统一视图	469
SYSSYNCS 统一视图	469
SYSSYNCSCRIPT 系统视图	470
SYSSYNCSCRIPTS 统一视图	470
SYSSYNCSUBSCRIPTIONS 统一视图	470
SYSSYNCSUSERS 统一视图	470
SYSTAB 系统视图	471
SYSTABLE 兼容性视图（不建议使用）	471
SYSTABAUTH 统一视图	471
SYSTABCOL 系统视图	471
SYSTABLEPERM 系统视图	472
SYSTEXTCONFIG 系统视图	472
SYSTEXTIDX 系统视图	472
SYSTEXTIDXTAB 系统视图	473
SYSTRIGGER 系统视图	473
SYSTRIGGERS 统一视图	473
SYSTYPEMAP 系统视图	473

SYSTYPES ASE 兼容性视图	474
SYSUSER 系统视图	474
SYSUSERAUTH 兼容性视图（不建议使用）	474
SYSUSERAUTHORITY 系统视图	474
SYSUSERLIST 兼容性视图（不建议使用）	475
SYSUSERMESSAGE 系统视图	475
SYSUSEROPTIONS 统一视图	475
SYSUSERPERM 兼容性视图（不建议使用）	475
SYSUSERPERMS 兼容性视图（不建议使用）	476
SYSUSERTYPE 系统视图	476
SYSUSERS ASE 兼容性视图	476
SYSVIEW 系统视图	476
SYSVIEWS 统一视图	477
SYSWEBSERVICE 系统视图	477
Transact-SQL 兼容性视图	477
附录 A	
与其它 Sybase 数据库的兼容性	481
Transact-SQL 支持概述	482
Adaptive Server 体系结构	483
服务器和数据库	483
空间分配和设备管理	484
系统表、Catalog 存储和 IQ 存储	485
管理角色	485
数据类型	486
Bit 数据类型	487
字符数据类型	487
二进制数据类型	488
日期、时间、日期时间和时间戳数据类型	489
数值数据类型	490
Text 数据类型	490
Image 数据类型	490
Java 数据类型	491
数据定义语言	491
创建与 Transact-SQL 兼容的数据库	491
区分大小写	492
确保兼容的对象名称	493
CREATE TABLE 语句	493
CREATE DEFAULT、CREATE RULE 和 CREATE DOMAIN	
语句	496
CREATE TRIGGER 语句	496
CREATE INDEX 语句	497
用户、组和权限	498
装载格式	500
为实现 Transact-SQL 兼容性设置选项	500

数据操作语言	500
编写可移植 SQL 的一般原则	500
编写兼容的查询	501
子查询	502
GROUP BY 子句	502
COMPUTE 子句	503
WHERE 子句	503
连接	503
Null 比较	504
零长度字符串	505
HOLDLOCK、SHARED 和 FOR BROWSE	505
SQL 函数	505
OLAP 函数	507
系统函数	508
用户定义的函数	509
日期的算术表达式	509
SELECT INTO	509
可更新视图	510
UPDATE 和 DELETE 中的 FROM 子句	510
Transact-SQL 过程语言概述	510
Transact-SQL 存储过程概述	510
Transact-SQL 批处理语句概述	511
过程和批处理语句中的 SQL 语句	511
存储过程的自动转换	513
从 Transact-SQL 过程中返回结果集	513
Transact-SQL 过程中的变量	514
Transact-SQL 过程中的错误处理	515
在过程中使用 RAISERROR 语句	516
Watcom-SQL 术语中与 Transact-SQL 类似的错误处理	517
SQL Anywhere 和 Sybase IQ	517
服务器和数据库启动与管理	517
数据库选项	518
数据定义语言 (DDL)	518
数据操作语言 (DML)	519
存储过程	520
Adaptive Server Enterprise 和 Sybase IQ	520
存储过程	520
系统视图	521
索引	523

关于本手册

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》。

相关 Sybase IQ 文档

Sybase IQ 15.2 文档集包括：

- 适用于您的平台的《发行公告》— 包含未能及时写入手册的最新信息。
最新版本的发行公告可能已发布。要了解本产品 CD 发行以后增加的重要产品或文档信息，请使用 Sybase Product Manuals Web 站点。
- 适用于您的平台的《安装和配置指南》— 介绍 Sybase IQ 的安装、升级和一些配置过程。
- 《Sybase IQ 15.2 新增功能摘要》— 总结了当前版本的新增功能和行为更改。
- 《Sybase IQ 中的高级安全性》— 涉及如何在 Sybase IQ 数据存储库中使用用户加密列。需要有单独的许可证才能安装此产品选项。
- 《错误消息》— 列出了由 Sybase 错误代码、SQLCode 和 SQLState 引用的 Sybase IQ 错误消息以及 SQL 预处理器错误和警告。
- 《IMSL 数字库用户指南：C Stat 库卷 2（共二卷）》— 包含 IMSL C Stat Library 时序 C 函数的简要说明。本手册仅适用于 RAP — Trading Edition® Enterprise 用户。
- 《Sybase IQ 简介》— 包括针对不熟悉 Sybase IQ 或 Sybase Central™ 数据库管理工具的用户的练习。
- 《性能和调优指南》— 介绍有关超大型数据库的查询优化、设计和调优问题。
- 《快速入门》— 讨论如何生成并查询 Sybase IQ 提供的演示数据库以验证 Sybase IQ 软件安装。其中包括有关将演示数据库转换为 Multiplex 数据库的信息。

-
- 《参考手册》 — Sybase IQ 的参考指南：
 - 《参考：构件块、表和过程》 — 介绍 Sybase IQ 支持的 SQL、存储过程、数据类型和系统表。
 - 《参考：语句和选项》 — 介绍 Sybase IQ 支持的 SQL 语句和选项。
 - 《系统管理指南》 — 包括：
 - 《系统管理指南，卷 1》 — 介绍启动、连接、数据库创建、填充和编制索引、版本控制、归类、系统备份和恢复、故障排除和数据库修复。
 - 《系统管理指南，卷 2》 — 介绍如何编写和运行过程和批处理、使用 OLAP 编程、访问远程数据、将 IQ 设置为 Open Server。本手册还讨论调度和事件处理、XML 编程和调试。
 - 《时序指南》 — 介绍用于时序预测和分析的 SQL 函数。需要 RAP — Trading Edition Enterprise 才能使用此产品选件。
 - 《Sybase IQ 中的非结构化数据分析》 — 说明如何在 Sybase IQ 数据库中存储和检索非结构化数据。需要有单独的许可证才能安装此产品选件。
 - 《用户定义的函数指南》 — 提供有关用户定义的函数、这些函数的参数以及可能的使用情形的信息。
 - 《使用 Sybase IQ Multiplex》 — 说明如何使用可管理跨多个节点的大型查询负载的 Multiplex 功能。
 - 《实用程序指南》 — 提供 Sybase IQ 实用程序参考资料，如可用的语法、参数和选项。

可以在以下位置通过联机方式获得 Sybase IQ 15.2 文档集：位于 <http://sybooks.sybase.com> 上的“产品手册” (Product Manuals)。

相关 SQL Anywhere 文档

由于 Sybase IQ 与 SQL Anywhere® Server (SQL Anywhere 软件包的组件) 共享许多相同组件，因此 Sybase IQ 支持许多与 SQL Anywhere Server 相同的功能。IQ 文档集会提示您参考相应的 SQL Anywhere 文档。

SQL Anywhere 的文档包括：

- 《SQL Anywhere Server — 数据库管理》介绍如何运行、管理和配置 SQL Anywhere 数据库。它介绍数据库连接、数据库服务器、数据库文件、备份过程、安全性、高可用性和使用 Replication Server® 进行复制以及管理实用程序和选项。

- 《SQL Anywhere Server — 编程》介绍如何使用 C、C++、Java、PHP、Perl、Python 和 .NET 编程语言（如 Visual Basic 和 Visual C#）生成和部署数据库应用程序。本手册还介绍诸如 ADO.NET 和 ODBC 之类的各种编程接口。
- 《SQL Anywhere Server — SQL 参考》提供系统过程的参考信息和目录（系统表和视图）。它还提供 SQL 语言的 SQL Anywhere 实现的说明（搜索条件、语法、数据类型和函数）。
- 《SQL Anywhere Server — SQL 用法》介绍如何设计和创建数据库；如何导入、导出和修改数据；如何检索数据以及如何生成存储过程和触发器。

此外，您还可以参阅位于 <http://sybooks.sybase.com> 的“产品手册” (Product Manuals) 的 SQL Anywhere 11.0.1 集中和位于 http://dcx.sybase.com/dcx_home.php 的 DocCommentXchange 站点内的 SQL Anywhere 文档。

语法定约

本文档在语法说明中使用以下约定：

- **关键字** SQL 关键字以大写字母显示。但是，SQL 关键字不区分大小写，所以可以大写或小写输入关键字：SELECT、Select 和 select 是相同的。
- **占位符** 必须以相应的标识符或表达式替代的项显示为 *斜体*。
- **续行符** 以省略号 (...) 开头的行为上一行语句的继续。
- **重复项** 重复项列表以相应列表元素后跟省略号 (...) 的形式表示。允许指定一个或多个列表元素。如果指定了多个元素，则必须用逗号将它们隔开。

- **可选部分** 语句的可选部分以中括号括起。例如：

```
RELEASE SAVEPOINT [ savepoint-name ]
```

中括号表示 *savepoint-name* 是可选的。不要键入括号。

- **选项** 如果在一个项列表中只能选择其中的一项或者不能选择任何一项，则这些项以竖线分隔，且整个列表以中括号括起。例如：

```
[ ASC | DESC ]
```

中括号表示可以选择 ASC 或 DESC，或两者都不能选。不要键入括号。

- **替代方法** 如果必须选择多个选项中的一个选项，则将备选选项以大括号括起。例如：

```
QUOTES { ON | OFF }
```

大括号表示必须包括 ON 或 OFF。不要键入括号。

排版约定

此表列出了本文档中采用的排版约定。

表 1: 排版约定

项目	说明
代码	SQL 和程序代码以等宽（固定宽度）字体显示。
用户输入	用户输入的文本以等宽（固定宽度）字体显示。
<i>文件名</i>	文件名以斜体显示。
数据库对象	数据库对象（如表和过程）的名称在印刷版本中以 sans serif 字型显示，在联机版本中以斜体显示。

关于本章

本章描述 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。

目录

主题	页码
安装目录结构	1
Sybase IQ 如何定位文件	2
环境变量	4
注册表条目	13

安装目录结构

安装 Sybase IQ 时，可能创建多个目录。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。本节说明了目录的结构。

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。

Sybase IQ 目录

缺省情况下，Sybase IQ 目录名称为 `IQ-15_2`。`IQ-15_2` 的位置因 Sybase IQ 的安装位置而异。`IQ-15_2` 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：

- *演示目录* (`%ALLUSERSPROFILE%/SybaseIQ/demo`)— 包含建立 `iqdemo` 数据库所需的工具。`iqdemo` 数据库文件包括 `iqdemo.db`、`iqdemo.iq`、`iqdemo.iqmsg` 和 `iqdemo.iqtmp`。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/mkiqdemo.bat` 或 UNIX 上的 `demo/mkiqdemo.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。

- **脚本目录 (IQ-15_2/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。
- **示例目录** — 示例目录包含 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` 目录包含说明如何将 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。`%ALLUSERSPROFILE%/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、帮助文件等文件。

在 UNIX 上，可执行文件子目录包括 `IQ-15_2` 子目录 `/bin64`、`/lib64`、`/logfiles`、`/res` 和 `/tix`。在 Windows 上，可执行文件子目录包括 `IQ-15_2` 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。为了确保使用的是正确文件，了解 Sybase IQ 如何查找这些文件是很重要的。系统上可能驻留有一些同名的目录或文件。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。

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.nnm` 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。

- 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，卷 1》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_2/logfiles/yourservername.0006.srvlog*。数据库日志（例如，*%ALLUSERSPROFILE%/SybaseIQ/IQ-15_2/demo/iqdemo.log*）将在连接到数据库时创建。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，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（比如 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）
- 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，比如 Windows 上的 Windows 目录和 Windows\system 目录。
- 5 CLASSPATH 目录 — 对于 Java 文件，将搜索 CLASSPATH 环境变量中列出的目录，以查找文件。
- 6 PATH 目录 — 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 — 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。

设置环境变量

必需的环境变量在 UNIX 上由环境源文件设置，在 Windows 上由 Sybase IQ 安装程序设置。

❖ 运行 UNIX 环境源文件

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_2/IQ-15_2.sh
```

- 2 对于 C shell:

```
source $SYBASE/IQ-15_2/IQ-15_2.csh;  
rehash
```

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，请根据您的操作系统选择以下过程之一。

❖ 在 Windows 上设置环境变量

- 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。

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

❖ 在 UNIX 上设置环境变量

- 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 在一个启动文件（`.cshrc`、`.shrc`、`.login`）中，添加一个用来设置变量的行。

在某些外壳程序（如 `sh`、`bash` 和 `ksh`）中，该行如下所示：

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

在其它外壳程序（如 `csh` 和 `tsch`）中，该行如下所示：

```
setenv VARIABLE "value"
```

有关 Sybase IQ 使用的变量的详细信息，请参见：

- [第 7 页的 “IQDIR15 环境变量”](#)
- [第 7 页的 “IQPORT 环境变量”](#)
- [第 8 页的 “IQLOGDIR15 环境变量”](#)
- [第 8 页的 “IQTIMEOUT 环境变量”](#)
- [第 9 页的 “IQTMP15 环境变量”](#)
- [第 10 页的 “JAVA_HOME 环境变量”](#)
- [第 10 页的 “LIBPATH 或 LD_LIBRARY_PATH 环境变量”](#)
- [第 10 页的 “PATH 环境变量”](#)
- [第 11 页的 “SCHARSET 环境变量”](#)
- [第 11 页的 “SALANG 环境变量”](#)
- [第 11 页的 “SQLCONNECT 环境变量”](#)
- [第 12 页的 “SYBASE 环境变量”](#)
- [第 12 页的 “\\$SYBASE_JRE6_32、\\$SYBASE_JRE6_64 和 \\$SYBASE_JRE5_64 环境变量”](#)
- [第 13 页的 “SYBASE_OCS 环境变量”](#)

IQDIR15 环境变量

设置	<code>IQDIR15 = \${SYBASE}/IQ-15_2</code>
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。
说明	<p>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：</p> <ul style="list-style-type: none">• <code>IQDIR15/bin[64]/util_db.ini</code> 包含实用程序数据库 <code>utility_db</code> 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。• <code>IQDIR15/logfiles</code> 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 <code>IQLOGDIR15</code> 环境变量，可以覆盖此缺省值。• <code>IQDIR15/demo</code> 是 <code>iqdemo</code> 数据库文件所在的位置。

IQPORT 环境变量

设置	<code>IQPORT = 5556</code>
操作系统	可选。如果用户未在环境源文件中指定 <code>IQPORT</code> ，则端口号缺省为 1099。可以在插件启动之前更改此缺省值。可以按照第 5 页的“ 设置环境变量 ”中的描述设置此变量，也可以通过在启动 Sybase Central 时向 <code>scjview</code> 提供 <code>-DIQPORT</code> 参数来进行设置。例如：
说明	<pre>scjview -DIQPORT=3345</pre> <p>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。</p>

注释 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。

IQLOGDIR15 环境变量

设置 `IQLOGDIR15 = path`

操作系统 可选。

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

- 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*。

IQTIMEOUT 环境变量

设置 `IQTIMEOUT = nnn`

操作系统 可选，但在 Multiplex 环境中推荐使用。有关 Multiplex 功能的信息，请参见《使用 Sybase IQ Multiplex》。

说明 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。

IQTIMEOUT 将覆盖缺省等待时间（五分钟），而参数 *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
```

注释 请在调用代理启动选项之前进行 IQTIMEOUT 设置。请参见《安装和配置指南》中的“安装之前”和“启动 Sybase IQ 代理”以及《Sybase IQ 简介》中的“运行 Sybase IQ 代理”。

IQTMP15 环境变量

设置

```
IQTMP15 = temp_directory
```

操作系统

在 UNIX 上可选。在 Windows 平台上不使用。

说明

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。

注释 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。有关 Sybase IQ 如何确定临时存储位置的信息，请参见《参考：语句和选项》的[第1章“SQL 语句”](#)中的 [CREATE DATABASE 语句](#)。

如果未显式设置 IQTMP15，或者它已设为 \$SYBASE 或 \$IQDIR15，则 Sybase IQ 代理会将 IQTMP15 设为 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
```

JAVA_HOME 环境变量

设置	<code>JAVA_HOME = Sybase/shared/JRE<version></code>
操作系统	必需。
说明	<p>指向包含 bin/java 的目录的 JRE 主目录。如果没有在 <code>\$\$SYBASE_JRE6_32</code>、<code>\$\$SYBASE_JRE6_64</code> 或 <code>\$\$SYBASE_JRE5_64</code> 环境变量中设置 Java VM 的位置，则使用此变量。</p> <p><code>JAVA_HOME</code> 通常是在安装 VM 时创建的。</p> <p>在 UNIX 上，运行 <code>SYBASE.csh</code> (C shell) 或 <code>SYBASE.sh</code> (Bourne 或 Korn shell) 环境源文件可查找并启动适用于 IQ 引擎的 JRE。在 <code>JAVA_HOME</code> 中指定的 Java VM 位置优先于 <code>SYBASE.csh</code> 或 <code>SYBASE.sh</code> 返回的位置。如果 <code>JAVA_HOME</code>、<code>SYBASE.csh</code> 或 <code>SYBASE.sh</code> 脚本均没有找到 Java VM，IQ 将不装载 Java VM。</p>

LIBPATH 或 LD_LIBRARY_PATH 环境变量

设置	<pre>For AIX: LIBPATH = installation_path/lib For all other UNIX/LINUX platforms: LD_LIBRARY_PATH = installation_path/lib</pre>
操作系统	必需。变量名因平台而异。仅限于 UNIX。
说明	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。

PATH 环境变量

设置	<code>PATH = installation_path</code>
操作系统	必需。
说明	<p><code>PATH</code> 环境变量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 <code>PATH</code>。在 UNIX 上，可以运行环境源文件以纳入必需的目录。</p> <p>在 Windows 上，<code>PATH</code> 取代了 <code>LIBRARY_PATH</code> 变量的位置，因此使用 <code>PATH</code> 变量可以找到可执行文件和 DLL。</p>

SACHARSET 环境变量

设置	SACHARSET=charset
说明	<p><i>Charset</i> 是字符集名称。例如，设置 SACHARSET=cp1252 可将缺省字符集设为 cp1252。</p> <p>以下值中，第一个设定的值将决定缺省的字符集。</p> <ul style="list-style-type: none">• SACHARSET 环境变量• 查询操作系统 <p>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。</p>

SALANG 环境变量

设置	SALANG=language_code
操作系统	可选，但在非英语环境中推荐使用。
说明	<p><i>Language_code</i> 是代表一种语言的两个字母的组合。例如，设置 SALANG=DE 可将缺省语言设为德语。</p> <p>以下值中，第一个设定的值将决定缺省的语言。</p> <ul style="list-style-type: none">• SALANG 环境变量• 由安装程序或 <i>dblank.exe</i> 设置的注册表（仅限 Windows）• 查询操作系统 <p>如果未设置任何语言信息，则英语将是缺省语言。</p>

SQLCONNECT 环境变量

设置	SQLCONNECT = parameter#value ; ...
操作系统	可选。
说明	<p>SQLCONNECT 环境变量是可选的，它不是由安装程序设置的。</p> <p>SQLCONNECT 用于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 <i>dbisql</i>、<i>dbinfo</i> 和 <i>dbstop</i>）使用的连接参数。此字符串是一个参数设置的列表，其格式为 <i>parameter=value</i>，以分号分隔。</p>

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。

注释 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 `dbisql` 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。

另请参见

请参见《系统管理指南，卷 1》的 [第 4 章“连接和通信参数”](#) 中的“[连接参数](#)”。

SYBASE 环境变量

设置

```
SYBASE = path
```

操作系统

必需。

说明

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。

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

设置

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

说明

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/LinuxIBM，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。

SYBASE_OCS 环境变量

设置	<code>SYBASE_OCS = "OCS-15_2"</code>
操作系统	必需。
说明	SYBASE_OCS 变量指定 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。

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（比如 Windows）具有两个级别的系统设置。某些设置是特定于单个用户的，仅当该用户登录时才会使用；这些设置称为当前用户设置。某些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；这些设置被称为本地计算机设置。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。

何时需要本地计算机设置

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。

若要启动该编辑器，请选择“开始”|“运行”，并在“打开”框中键入内容。

```
regedt32
```

注释 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 15.2
```

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_2
```

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。

关于本章

本章提供 Sybase IQ SQL 的语言元素和约定的详细说明。

目录

主题	页码
关键字	15
标识符	18
字符串	20
表达式	21
搜索条件	30
特殊值	49
变量	53
注释	59
NULL 值	60

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，但在这些手册中，关键字都以大写字母显示。

例如，在下面的语句中，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，必须使用以下语法检索名为 **SELECT** 的表的内容。

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

由于 SQL 的关键字不区分大小写，因此表 2-1 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。

如果使用嵌入式 SQL，则可以利用数据库函数 `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。

表 2-1 列出了 Sybase IQ 中的 SQL 保留字。

表 2-1: SQL 保留字

active	add	all	algorithm
alter	and	any	append
as	asc	auto	backup
begin	between	bigint	binary
bit	bottom	break	by
calibrate	calibration	call	cancel
capability	cascade	case	cast
certificate	char	char_convert	character
check	checkpoint	checksum	clientport
close	column	comment	commit
committed	comparison	compute	conflict
connect	constraint	contain	continue
convert	create	cross	cube
current	current_timestamp	current_user	cursor
date	dbspace	dbspacename	deallocate
debug	dec	decimal	declare
decoupled	decrypted	default	delay
delete	deleting	density	desc
deterministic	disable	distinct	do
double	drop	dynamic	elements
else	elseif	enable	encapsulated
encrypted	end	endif	escape

except	exception	exclude	exec
execute	existing	exists	explicit
express	externlogin	fastfirstrow	fetch
first	float	following	for
force	foreign	forward	from
full	gb	goto	grant
group	grouping	having	hidden
history	holdlock	identified	if
in	inactive	index	index_lparen
inner	inout	input	insensitive
insert	inserting	install	instead
int	integer	integrated	intersect
into	iq	is	isolation
jdk	join	kb	key
lateral	left	like	lock
logging	login	long	mb
match	membership	message	mode
modify	namespace	natural	new
no	noholdlock	nolock	not
notify	null	numeric	of
off	on	open	optimization
option	options	or	order
others	out	outer	over
pages	paglock	partial	partition
passthrough	password	plan	preceding
precision	prepare	primary	print
privileges	proc	procedure	proxy
publication	raiserror	range	raw
readcommitted	readonly	readpast	readtext
readuncommitted	readwrite	real	recursive
reference	references	release	relocate
remote	remove	rename	reorganize
repeatable	repeatableread	reserve	resizing
resource	restore	restrict	return
revoke	right	rollback	rollup
root	row	rowlock	rows
save	savepoint	schedule	scroll
secure	select	sensitive	serializable

service	session	set	setuser
share	smallint	soapaction	some
space	sqlcode	sqlstate	start
stop	subtrans	subtransaction	synchronize
syntax_error	table	tablock	tablockx
tb	temporary	then	ties
time	timestamp	tinyint	to
top	tran	transaction	transactional
transfer	tries	trigger	truncate
tsequal	unbounded	uncommitted	union
unique	uniqueidentifier	unknown	unsigned
update	updating	updlock	url
user	utc	using	validate
values	varbinary	varchar	variable
varying	virtual	view	wait
waitfor	web	when	where
while	window	with	withauto
with_cube	with_lparen	with_rollup	within
word	work	writeserver	writetext
xlock	xml		

标识符

作用

标识符是数据库中对象（如用户 ID、表和列）的名称。

说明

标识符的最大长度是 128 个字节。当以下任一条件成立时，标识符需要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。

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）。
- 数据库名不能包含双引号、单引号和分号。
- 用户名和数据库名不能以空格开头或结尾。
- `dbspace` 名称在 `CASE RESPECT` 数据库中不区分大小写。

如果将 `QUOTED_IDENTIFIER` 数据库选项设置为 `OFF`，则双引号用于分隔 SQL 字符串，而不能用于标识符。但不管 `QUOTED_IDENTIFIER` 如何设置，您都可以用中括号来界定标识符。

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`。

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

有关保留字的完整列表，请参见第 16 页的“保留字”。

有关 `QUOTED_IDENTIFIER` 选项的信息，请参见第 28 页的“`quoted_identifier` 选项”。

有关服务器和数据库名的其它限制，请参见《实用程序指南》的“`start iq 服务器选项`”中的 `-n` 服务器选项，或第 1 章“运行数据库服务器”中的 `-n` 数据库选项。

字符串

字符串具有以下类型：

- 字符串
- 具有 CHAR 或 VARCHAR 数据类型的表达式。

具有 CHAR 数据类型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。

有关表达式的详细信息，请参见第 21 页的“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：

```
'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。

表达式

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
说明	表达式由若干不同种类的元素构成，这些元素将在以下几节中进行论述。
兼容性	<ul style="list-style-type: none">• 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。

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。有关相关名的详细信息，请参见《参考：语句和选项》中的 [FROM 子句](#)。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"
```

请参见第 18 页的“标识符”。

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;
```

有关子查询的其它用法，请参见第 32 页的“搜索条件中的子查询”。

SQL 运算符

本节介绍 Sybase IQ 中提供的算术运算符、字符串运算符和逐位运算符。有关比较运算符的信息，请参见第 30 页的“搜索条件”。

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。

算术运算符

expression + expression 加法。如果其中任一表达式为空值，则结果为空值。

expression - expression 减法。如果其中任一表达式为空值，则结果为空值。

- expression 取非。如果表达式为空值，则结果为空值。

expression * expression 乘法。如果其中任一表达式为空值，则结果为空值。

expression / expression 除法。如果其中任一表达式为空值或第二个表达式为 0，则结果为空值。

expression % expression 两个整数相除之后，模取整数余数。例如，因为 21 除以 11 等于 1，余数为 10，所以 $21 \% 11 = 10$ 。

字符串运算符

expression || expression 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 + expression 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，您必须具有一个非结构化数据分析选项许可证，或者使用 CAST 并将 LEFT 设置为正确的数据类型和大小。

有关信息和用法，请参见第 225 页的“REVERSE 函数 [字符串]”。

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。

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运算符。

运算符	说明
&	与
	或
^	异或
~	非

与运算符 (&)

位 1	位 2	位 1 与 位 2
0	0	0
0	1	0
1	0	0
1	1	1

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。

逐位或 (|)

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

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。

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
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** 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 函数对数据类型进行显式转换。

注释 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

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，且没有 *expression2*，则 IF 表达式返回 NULL。
- 如果 *condition* 为 NULL，则 IF 表达式返回 NULL。

有关 TRUE、FALSE 和 UNKNOWN 条件的详细信息，请参见第 60 页的“NULL 值”和第 30 页的“搜索条件”。

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。

请参见《参考：语句和选项》中的 IF 语句。

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 语句后的表达式与 WHEN 语句后的表达式相同，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 "GROUPO".Products

```

替代语法如下：

```

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

```

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

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

```

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

```

用于简写 CASE 表达式的 NULLIF 函数

NULLIF 函数提供了一种以简写形式编写某些 CASE 语句的方法。NULLIF 的语法如下：

```

NULLIF ( expression-1, expression-2 )

```

NULLIF 比较两个表达式的值。如果第一个表达式的值与第二个表达式的值相等，则 NULLIF 返回 NULL。如果它们不相等，则 NULLIF 返回第一个表达式。

表达式的兼容性

表 2-2 和表 2-3 说明了 Adaptive Server Enterprise (ASE) 与 Sybase IQ 之间的表达式和常量的兼容性。这些表只是一个指南，而且，两者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。

在表 2-2 中，*expr* 表示表达式，*op* 表示运算符。

表 2-2: ASE 与 Sybase IQ 之间的表达式兼容性

表达式	支持情况
常量	两者
列名	两者
变量名	两者
函数 (expr)	两者
- expr	两者
expr op expr	两者
(expr)	两者
(subquery)	两者
if 表达式	仅 Sybase IQ

表 2-3: ASE 与 Sybase IQ 之间的常量兼容性

常量	支持情况
integer	两者
number	两者
'string'	两者
special-constant	两者
host-variable	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。

有关保留字的完整列表，请参见第 16 页的表 2-1。

设置该选项

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。

在 Sybase IQ 或 Adaptive Server Enterprise 中，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。

在 Sybase IQ 或 Adaptive Server Enterprise 中，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（Sybase IQ 中的缺省设置）时进行操作，则对于两种类型的 DBMS，以下包含 SQL 关键字 `user` 的语句均有效。

```
CREATE TABLE "user" (  
    coll char(5)  
);  
INSERT "user" ( coll )  
VALUES ( 'abcde' );
```

如果在 `quoted_identifier` 选项为 Off（Adaptive Server Enterprise 中的缺省设置）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"
```

搜索条件

作用	用于指定 WHERE 子句、HAVING 子句、CHECK 子句、JOIN 子句或 IF 表达式的搜索条件。
语法	<pre> { expression compare expression expression compare { ANY SOME ALL } (subquery) expression IS [NOT] NULL 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 } </pre>
参数	<pre> compare: { = > < >= <= <> != !< !> } </pre>
用法	任何位置
授权	必须连接到数据库。
示例	<p>例如，下面的查询可检索年龄最大雇员的姓名和出生年份：</p> <pre> SELECT Surname, BirthDate FROM Employees WHERE BirthDate <= ALL (SELECT BirthDate FROM Employees); </pre> <p>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。</p>
副作用	无
另请参见	第 21 页的“表达式” 。
说明	<p>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。</p> <p>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。有关显示三值逻辑中逻辑运算符的组合方式的表，请参见第 43 页的“三值逻辑”。</p>

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。有关详细信息，请参见第 60 页的“NULL 值”。

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。有关详细信息，请参见第 32 页的“搜索条件中的子查询”。

以下几节将论述不同类型的搜索条件。

比较条件

比较条件的语法如下：

expression compare expression

其中 *compare* 是比较运算符。表 2-4 列出了 Sybase IQ 中提供的比较运算符。

表 2-4: Sybase IQ 中提供的比较运算符

运算符	说明
=	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。

注释 所有字符串比较:

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

有关不区分大小写的数据库中的比较结果的详细信息，请参见《参考：语句和选项》中的 [CREATE DATABASE 语句](#)。

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。

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项列表项。

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。

返回一列以及任意行数的子查询可用于 IN 条件、ANY 条件、ALL 条件或 EXISTS 条件。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，以下查询包含两个通过 OR 连接的子查询：

```
CREATE VARIABLE @ln int;

SELECT @ln = 1;select count(*) FROM lineitem

WHERE l_shipdate IN (select l_shipdate FROM lineitem
WHERE l_orderkey IN (2,4,6))

OR l_shipdate IN (select l_shipdate FROM lineitem WHERE
```

```
l_orderkey IN (1,3,5))
OR l_linenumber = @ln;
```

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

请参见第 31 页的“比较条件”。

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（包含对在外表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》中的 [SUBQUERY_CACHING_PREFERENCE 选项](#)。

示例

示例 1 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');
```

示例 2 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');
```

示例 3 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);
```

示例 4 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);
```

示例 5 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 合并最后结果。

下面的查询说明了一个合并查询，该查询获取的结果与第 35 页的“示例 5”中查询的结果相同。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

ANY 条件的语法如下：

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 NULL 且不等于子查询中的任何一列，则 ANY 条件为 FALSE：

expression = ANY (subquery)

除非子查询结果不包含任何行（在这种情况下，该条件始终为 FALSE），否则在 *expression* 为空值的情况下，ANY 条件为 UNKNOWN。

您可以使用关键字 SOME 而非 ANY。

ALL 条件的语法为：

expression compare ALL (subquery)

其中，*compare* 为比较运算符。

限制

如果定量比较谓词的某一侧有多个表达式，则会返回错误消息。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

在 Adaptive Server Enterprise 与 Sybase IQ 之间，ANY 和 ALL 子查询是兼容的。只有 Sybase IQ 支持将 SOME 用作 ANY 的同义词。

BETWEEN 条件

BETWEEN 条件的语法如下所示：

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，则条件的值为 TRUE。NOT 关键字使条件的含义相反，但保留 UNKNOWN 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 与 Adaptive Server Enterprise 之间是兼容的。

LIKE 条件

LIKE 条件的语法如下：

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，如果有可用的 HG 或 LF 索引，会提高该谓词的执行速度。

如果提供了 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，则条件的值为 TRUE。如果 *expression* 或 *pattern* 的值为空值，则此条件为 UNKNOWN。NOT 关键字使条件的含义相反，但保留 UNKNOWN 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

介于 127 到 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
'12345678901234567890123456789012345678901234567890123
456789012345678901234567890123456789012345678901234567
89012345678901234567890' ;
```

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 会生成错误。

介于 127 到 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
'12345678901234567890123456789012345678901234567890123
456789012345678901234567890123456789012345678901234567
8901234567890123456_____ ' ;
```

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 会生成错误。

介于 127 到 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
'12345678901234567890123456789012345678901234567890123
456789012345678901234567890123456789012345678901234567
8901234567890123456[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（对于括号中的 1）+ 1（对于括号中的 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 会生成错误。

搜索集合中的某个字符

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'
```

搜索一定范围内的某个字符

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，卷 1》的第 11 章“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'
```

括号 [a-mpqs-z] 解释为“正好一个字符，该字符或者处于 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，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，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。[^] 仅匹配脱字符，[%] 仅匹配百分号（在此环境中百分号字符不充当通配符），[_] 仅匹配下划线字符。同样，[[] 仅匹配字符 [。

其它特殊情况包括：

- 表达式 [a-] 匹配字符 a 或 -。
- 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] 查找 a、% 或 b 之一。
- 除非作为括号内的首个字符，否则脱字符不能用于否定范围。表达式 [a^b] 查找 a、^ 或 b 之一。

兼容性

只有 Sybase IQ 支持 ESCAPE 子句。

另请参见

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》中的第 4 章“非结构化数据查询”。

用户必须经过专门授权才能使用大对象数据类型 LONG BINARY 和 LONG VARCHAR。有关非结构化数据分析选件的详细信息，请参见《Sybase IQ 中的非结构化数据分析》。

IN 条件

IN 条件的语法如下：

```
{ 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* 的值为空值，则条件为 UNKNOWN；否则，条件为 FALSE。NOT 关键字使条件的含义相反，但保留 UNKNOWN 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 与 Sybase IQ 之间是兼容的。

CONTAINS 条件

对于包含 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，而且必须具有一个 WD 索引。*word1*、*word2* 和 *word3* 表达式必须是长度不超过 255 个字节的字符串常量，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* 的值为空值，则条件为 UNKNOWN；否则，条件为 FALSE。NOT 关键字对这些值求反，但 UNKNOWN 保持不变。

例如，以下搜索条件：

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

如果 *varchar_col* 的值为 The cat is on the mat，则条件为 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

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；如果子查询结果不包含任何行，则条件为 FALSE。EXISTS 条件不能为 UNKNOWN。

兼容性

EXISTS 条件在 Adaptive Server Enterprise 与 Sybase IQ 之间是兼容的。

IS NULL 条件

IS NULL 条件的语法如下：

***expression* IS [NOT] NULL**

在没有 NOT 关键字的情况下，如果表达式的值为空值，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 与 Sybase IQ 之间是兼容的。

带有逻辑运算符的条件

可以使用 AND、OR 和 NOT 将搜索条件组合起来。

使用 AND 组合条件的方式如下：

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*），则不保证任何顺序。

请参见第 33 页的“分离子查询谓词”。

兼容性

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

NOT 条件

NOT 条件的语法如下：

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

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* 的值等于提供的 *truth-value*（必须是 TRUE、FALSE 或 UNKNOWN 之一），则条件的值为 TRUE。否则，值为 FALSE。NOT 关键字使条件的含义相反，但保留 UNKNOWN 不变。

兼容性

只有 Sybase IQ 支持真值条件。

三值逻辑

下表显示了 SQL 的 AND、OR、NOT 和 IS 逻辑运算符在三值逻辑中如何运算。

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

用户提供的条件提示

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。

注释 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。

用户提供的条件提示字符串

除支持用户提供的选择性估计外，Sybase IQ 还允许用户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。

有四种不同的提示类型：

- [选择性提示](#)（等同于用户提供的选择性估计）
- [索引首选项提示](#)
- [执行阶段提示](#)
- [有用性提示](#)

选择性提示

可以在提示字符串中出现的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 44 页的“用户提供的条件选择性”中的第二个示例完全等效。

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

索引首选项提示

下一种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。有关 -10 和 10 之间整数的具体含义，请参见《参考：语句和选项》的第 2 章“数据库选项”中的“INDEX_PREFERENCE 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
```

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）的说明如下：

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。

CURRENT DATABASE 特殊值

作用	CURRENT DATABASE 返回当前数据库的名称。
数据类型	STRING
另请参见	第 21 页的“表达式”

CURRENT DATE 特殊值

作用	当前年份、月份和日期。
数据类型	DATE
另请参见	第 21 页的“表达式” 第 76 页的“日期和时间数据类型”

CURRENT PUBLISHER 特殊值

作用	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
另请参见	CURRENT PUBLISHER 可在具有字符数据类型的列中用作缺省值。 第 21 页的“表达式”

CURRENT TIME 特殊值

作用	返回当前小时、分钟、秒和秒的小数值。
数据类型	TIME
说明	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	第 21 页的“表达式” 第 76 页的“日期和时间数据类型”

CURRENT TIMESTAMP 特殊值

作用	组合 CURRENT DATE 和 CURRENT TIME，以构成包含年月日、小时、分钟、秒和秒的小数值的 TIMESTAMP 值。与 CURRENT TIME 一样，秒的小数值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	第 21 页的“表达式” 第 76 页的“日期和时间数据类型”

CURRENT USER 特殊值

作用	CURRENT USER 返回一个字符串，其中包含当前连接的用户 ID。
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。
说明	在 UPDATE 时，不会更改缺省值为 CURRENT USER 的列。
另请参见	第 21 页的“表达式”

LAST USER 特殊值

作用	返回上次修改行的用户的名称。
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。
说明	在执行 INSERT 和 LOAD 操作时，此常量与 CURRENT USER 效果相同。在执行 UPDATE 操作时，如果未显式修改缺省值为 LAST USER 的列，该列将更改为当前用户的名称。 与 DEFAULT TIMESTAMP 组合使用时，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。
另请参见	第 50 页的“CURRENT USER 特殊值” 第 50 页的“CURRENT TIMESTAMP 特殊值” 《参考：语句和选项》中的 CREATE TABLE 语句

SQLCODE 特殊值

作用	返回当前 SQLCODE 值。
数据类型	STRING
说明	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。
另请参见	第 21 页的“表达式” 《系统管理指南，卷 2》中的第 1 章“SQL 语句”

SQLSTATE 特殊值

作用	返回当前 SQLSTATE 值。
数据类型	STRING
说明	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。
另请参见	第 21 页的“表达式” 《系统管理指南，卷 2》中的第 1 章“SQL 语句”

TIMESTAMP 特殊值

作用	TIMESTAMP 指示表中各行上次修改的时间。
数据类型	TIMESTAMP
说明	<p>当使用 DEFAULT TIMESTAMP 声明列时，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。</p> <p>执行 INSERT 和 LOAD 操作时，DEFAULT TIMESTAMP 与 CURRENT TIMESTAMP 效果相同。在执行 UPDATE 操作时，如果未显式修改缺省值为 TIMESTAMP 的列，则该列的值将变为当前日期和时间。</p>

注释 Sybase IQ 不支持 UTC TIMESTAMP 或 CURRENT UTC TIMESTAMP 的 DEFAULT 值，也不支持数据库选项 DEFAULT_TIMESTAMP_INCREMENT。每次用户尝试为类型为 UTC TIMESTAMP 或 CURRENT UTC TIMESTAMP 的列插入或更新 DEFAULT 值时，Sybase IQ 都会生成错误。

另请参见 [第 76 页的“日期和时间数据类型”](#)

USER 特殊值

作用	USER 返回一个字符串，其中包含当前连接的用户 ID。
数据类型	STRING
说明	<p>USER 可在具有字符数据类型的列中用作缺省值。</p> <p>在 UPDATE 时，不会更改缺省值为 USER 的列。</p>

另请参见 [第 21 页的“表达式”](#)
[第 50 页的“CURRENT USER 特殊值”](#)
[第 51 页的“LAST USER 特殊值”](#)

变量

Sybase IQ 支持以下三个级别的变量：

- **局部变量** 这些是使用 `DECLARE` 语句在过程或批处理中的复合语句内定义的变量。它们只存在于复合语句中。
- **连接级变量** 这些是使用 `CREATE VARIABLE` 语句定义的变量。它们属于当前连接，当您从数据库断开连接或使用 `DROP VARIABLE` 语句时，它们就会消失。
- **全局变量** 这些是具有系统提供值的变量。

局部变量和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。全局变量是系统提供的变量，用于提供系统提供的值。所有全局变量的名称都以两个 `@` 符号开头。例如，全局变量 `@@version` 所具有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

局部变量是使用 `DECLARE` 语句声明的，只能在复合语句（即两端带有关键字 `BEGIN` 和 `END` 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

- 下面的批处理语句说明如何使用局部变量：

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

从 ISQL 运行此批处理语句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 ;
```

- 以下示例说明如何使用带有 `INTO` 子句的 `SELECT` 语句设置局部变量的值：

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

从 `ISQL` 运行此批处理语句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` 语句，作用域将仅限于该复合语句。
- **声明** 在 `Sybase IQ` 中，每个 `DECLARE` 语句只能声明一个变量。而在 `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` 的语法如下：

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

变量创建时最初设置为 `NULL`。设置连接级变量的值的方法与局部变量相同，即使用 `SET` 语句或使用带有 `INTO` 子句的 `SELECT` 语句。

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 运行此批处理语句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

全局变量具有 Sybase IQ 设置的值。例如，全局变量 `@@version` 所具有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`）保存连接特定的信息，因此具有连接特定的值。其它变量（如 `@@connections`）则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十分相似。

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。以下过程用参数 *ver* 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

表 2-5 列出了 Sybase IQ 中提供的全局变量。

表 2-5: Sybase IQ 全局变量

变量名	含义
<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。如果发生错误，类似“if @@error != 0 return”的语句会导致退出。每个 SQL 语句都会重置 @@error，因此在成功状态不确定的语句之后必须立即执行状态检查。
<i>@@fetch_status</i>	包含从上一 fetch 语句所生成的状态信息。@@fetch_status 可包含以下值 <ul style="list-style-type: none"> • 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 @@sqlstatus 相同，只不过它返回的值不同。这是为了与 Microsoft SQL Server 兼容。
<i>@@identity</i>	通过插入、装载或更新语句插入标识 / 自动增量列的最后一个值。每次向表中插入一行时，@@identity 都会被重置。如果一个语句插入多行，@@identity 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，则会将 @@identity 设置为 0。@@identity 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。@@identity 保留插入标识 / 自动增量列的最后一个值，即使插入该值的语句未能提交也是如此。
<i>@@isolation</i>	当前隔离级别。@@isolation 采用活动级别的值。
<i>@@procid</i>	当前执行的过程的存储过程 ID。
<i>@@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。

兼容性

表 2-6 列出了 Sybase IQ 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。与表 2-5 相反，此列表包括返回值的所有全局变量，包括值固定为 NULL、1、-1 或 0 以及可能无意义的全局变量。

表 2-6: Sybase IQ 中支持的 ASE 全局变量

全局变量	返回值
@@char_convert	返回 0。
@@client_csname	在 Adaptive Server Enterprise 中，返回客户端的字符集名称。如果客户端字符集从未初始化，则设置为 NULL；否则包含最近使用的字符集的名称。在 Sybase IQ 中，返回 NULL。
@@client_csid	在 Adaptive Server Enterprise 中，返回客户端的字符集 ID。如果客户端字符集从未初始化，则设置为 -1；否则包含 syscharsets 中最近使用的客户端字符集 ID。在 Sybase IQ 中返回 -1。
@@connections	自服务器上次启动后的登录数。
@@cpu_busy	在 Adaptive Server Enterprise 中，返回自 Adaptive Server Enterprise 上次启动后 CPU 在 Adaptive Server Enterprise 运行上花费的时间（以时钟周期为单位）。在 Sybase IQ 中，返回 0。
@@error	<p>常用于检查最近执行的语句的错误状态（成功或失败）。如果上一事务成功，则包含 0，否则包含系统生成的最后一个错误号。如果发生类似如下的错误：</p> <pre>if @@error != 0 return</pre> <p>这样的语句会导致退出。每个语句（包括 PRINT 语句或 IF 测试）都会重置 @@error，因此在成功状态不确定的语句之后必须立即执行状态检查。</p>
@@identity	在 Adaptive Server Enterprise 中，返回通过 INSERT、LOAD 或 SELECT INTO 语句插入 IDENTITY 列的最后一个值。每次向表中插入一行时，@@identity 都会被重置。如果一个语句插入多行，@@identity 将反映最后插入的行的 IDENTITY 值。如果受影响的表不包含 IDENTITY 列，则会将 @@identity 设置为 0。@@identity 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。@@identity 保留插入 IDENTITY 列的最后一个值，即使插入该值的语句未能提交也是如此。
@@idle	在 Adaptive Server Enterprise 中，返回自服务器上次启动后 Adaptive Server Enterprise 的空闲时间（以时钟周期为单位）。在 Sybase IQ 中，返回 0。
@@io_busy	在 Adaptive Server Enterprise 中，返回自服务器上次启动后 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。

全局变量	返回值
<code>@@isolation</code>	当前的连接隔离级别。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 缺省字符集中的最大字符长度（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）。每次一个存储过程或触发器调用另一个存储过程或触发器时，嵌套级别都会递增。在 Sybase IQ 中，返回 -1。
<code>@@pack_received</code>	在 Adaptive Server Enterprise 中，返回自服务器上次启动后 Adaptive Server Enterprise 读取的输入数据包数。在 Sybase IQ 中，返回 0。
<code>@@pack_sent</code>	在 Adaptive Server Enterprise 中，返回自服务器上次启动后 Adaptive Server Enterprise 写入的输出数据包数。在 Sybase IQ 中，返回 0。
<code>@@packet_errors</code>	在 Adaptive Server Enterprise 中，返回在 Adaptive Server Enterprise 发送和接收数据包过程中发生的错误数。在 Sybase IQ 中，返回 0。
<code>@@procid</code>	当前执行的过程的存储过程 ID。
<code>@@servername</code>	本地 Adaptive Server Enterprise 或 Sybase IQ 服务器的名称。
<code>@@spid</code>	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 <code>sa_conn_info</code> 过程显示的值相同。
<code>@@sqlstatus</code>	包含从上一 FETCH 语句所生成的状态信息。 <code>@@sqlstatus</code> 可包含以下值： <ul style="list-style-type: none"> • 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。
<code>@@thresh_hysteresis</code>	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。

全局变量	返回值
<code>@@timeticks</code>	在 Adaptive Server Enterprise 中，返回每个时钟周期的微秒数。每个时钟周期的时间量与计算机相关。在 Sybase IQ 中，返回 0。
<code>@@total_errors</code>	在 Adaptive Server Enterprise 中，返回在 Adaptive Server Enterprise 读取或写入过程中发生的错误数。在 Sybase IQ 中，返回 0。
<code>@@total_read</code>	在 Adaptive Server Enterprise 中，返回自服务器上次启动后 Adaptive Server Enterprise 读取的磁盘数。在 Sybase IQ 中，返回 0。
<code>@@total_write</code>	在 Adaptive Server Enterprise 中，返回自服务器上次启动后 Adaptive Server Enterprise 写入的磁盘数。在 Sybase IQ 中，返回 0。
<code>@@tranchained</code>	Transact-SQL 程序的当前事务模式。 <code>@@tranchained</code> 返回 0（非链式）或 1（链式）。
<code>@@trancount</code>	事务的嵌套级别。批处理中的每个 <code>BEGIN TRANSACTION</code> 都会增加事务计数。
<code>@@transtate</code>	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。
<code>@@version</code>	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 中提供了几种注释指示符：

- **--（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。

注释 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 "GROUPO".Employees
WHERE DepartmentID = 200
```

NULL 值

作用	指定未知或不适用的值。
语法	NULL
用法	任何位置
权限	必须连接到数据库。
副作用	无

说明

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。空值用于表示缺少或不适用的信息。以下是使用 NULL 的两种截然不同的情况：

情况	说明
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。

NULL 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 中的一个值）选择包含 NULL 值的行。有关此子句的说明，请参见第 30 页的“搜索条件”。

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 NULL 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 NULL 值视为 0，则必须使用 ISNULL(*expression*, 0) 函数。请参见第 4 章“SQL 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。有关组合搜索条件时三值逻辑的效果的说明，请参见第 30 页的“搜索条件”。

示例

以下 INSERT 语句将一个 NULL 值插入表 Borrowed_book 的 date_returned 列中。

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

关于本章

本章介绍了 Sybase IQ 支持的数据类型。

目录

主题	页码
字符数据类型	63
数值数据类型	67
二进制数据类型	71
Bit 数据类型	76
日期和时间数据类型	76
向数据库发送日期和时间	78
从数据库中检索日期和时间	79
比较日期和时间	79
使用明确的日期和时间	80
域	81
数据类型转换	83

字符数据类型

说明

用于存储包含字母、数字和符号的字符串。

语法

```

CHAR [ ( max-length ) ]
CHARACTER [ ( max-length ) ]
CHARACTER VARYING [ ( max-length ) ]
VARCHAR [ ( max-length ) ]
UNIQUEIDENTIFIERSTR

```

用法

CHAR 最大长度为 *max-length* 个字节的字符数据。如果省略 *max-length*，则缺省值为 1。允许的最大大小为 32KB - 1。有关对超过 255 个字节的 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，均会用空白填充所有 **CHAR** 值以达到 *max-length*。当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。

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 这是用户定义的数据类型。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 个字节的 **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 4 GB 乘以数据库页大小。请参见《Sybase IQ 中的非结构化数据分析》。

存储大小

表 3-1 列出了字符数据的存储大小。

表 3-1: 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) 个字节的宽度	(32K - 1) 个字节	(32K - 1) 个字节
VARCHAR、CHARACTER VARYING	(32K - 1) 个字节的宽度	(32K - 1) 个字节	(32K - 1) 个字节

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 个字节的 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

使用 INSERT、UPDATE 或 LOAD TABLE 插入的数据可以采用下列格式：

- 加引号
- 不加引号
- 二进制

对于数据类型为 VARCHAR 的列，按如下方式处理插入的数据中的尾随空白：

- 对于加引号的数据，永远不会剪裁尾随空白。
- 对于未加引号的数据：
 - 在插入和更新时始终剪裁尾随空白。
 - 对于 LOAD 语句，可使用 STRIP RTRIM/OFF LOAD 选项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，则使用 CHAR 列，而非 VARCHAR 列。

对于超过 255 个字节的 CHAR 和 VARCHAR 数据的限制

超过 255 个字节的 CHAR 和 VARCHAR 列仅支持缺省索引以及 WD、TEXT 和 CMP 这三种索引类型。不能为这些列创建 LF、HG、HNG、DATE、TIME 或 DTTM 索引。

兼容性

- 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 支持为 CHAR 和 VARCHAR 数据类型使用 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

SQL Anywhere 将 CHAR、VARCHAR 和 LONG VARCHAR 列全部视为同一类型。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。

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

说明	存储数值数据。
语法	<pre>[UNSIGNED] BIGINT [UNSIGNED] { INT INTEGER } SMALLINT TINYINT DECIMAL [(precision [, scale])] NUMERIC [(precision [, scale])] DOUBLE FLOAT [(precision)] REAL</pre>
用法	<p>BIGINT 带符号的 64 位整数，需要 8 个字节的存储空间。</p> <p>您可以将整数指定为 UNSIGNED。缺省情况下，该数据类型是带符号的。它的范围介于 -9223372036854775808 到 9223372036854775807（带符号），或者是 0 到 18446744073709551615（不带符号）之间。</p> <p>INT 或 INTEGER 带符号的 32 位整数，值的范围介于 -2147483648 到 2147483647 之间，需要 4 个字节的存储空间。</p> <p>INTEGER 数据类型是精确数值数据类型；其精度在算术运算后不变。</p> <p>您可以将整数指定为 UNSIGNED 形式。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。</p> <p>SMALLINT 16 位有符号整数，范围介于 -32768 到 32767 之间，需要 2 个字节的存储空间。</p> <p>SMALLINT 数据类型是精确数值数据类型；其精度在算术运算后不变。</p> <p>TINYINT 不带符号的 8 位整数，范围介于 0 到 255 之间，需要 1 个字节的存储空间。</p> <p>TINYINT 数据类型是精确数值数据类型；其精度在算术运算后不变。</p> <p>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，将根据列的实际数据类型计算结果，但您可对返回应用程序的结果的最大标度进行设置。有关详细信息，请参见《参考：语句和选项》中第 386 页的“MAX_CLIENT_NUMERIC_SCALE 选项”和 SET OPTION 语句。</p>

表 3-2 列出了十进制数字所需的存储空间。

表 3-2: 十进制数字的存储大小

精度	存储
1 到 4	2 字节
5 到 9	4 字节
10 到 18	8 字节
19 到 126	请参见下文

对于精度大于 18 的十进制值的存储空间要求（以字节为单位），可以使用以下公式计算：

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

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 NULL 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
```

说明为：

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

注释 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：“The result datatype for function ' _funcname' exceeds the maximum supported numeric precision of 255. Please set the proper value for precision in numeric function, 'location'”（“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”）

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）。绝对非零值的范围是 $2.2250738585072014e-308$ 到 $1.797693134862315708e+308$ 。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 数据类型是一种近似数值数据类型；它在算术运算后容易产生舍入误差。

FLOAT 如果不提供 *precision*，则 FLOAT 数据类型与 REAL 数据类型相同。如果提供了 *precision*，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 数据类型是一种近似数值数据类型；它在算术运算后容易产生舍入误差。

REAL 占用 4 个字节存储空间单精度浮点数（带符号）。绝对非零值的范围是 $1.175494351e-38$ 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。

注意

- 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。

索引

- 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。

兼容性

- 在嵌入式 SQL 中，将 TINYINT 列提取到 2 个字节或 4 个字节的整数列中。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
- 如果是负值，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 数据类型的同义词，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 包括两种用户定义的数据类型，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。

二进制数据类型

说明 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) 个字节。UNIQUEIDENTIFIER 数据类型用于存储 UUID（也称为 GUID）值。

语法

```
BINARY [( length )]
VARBINARY [( max-length )]
UNIQUEIDENTIFIER
```

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（即 1 个字节）。每个字节可以存储 2 个十六进制数字。虽然缺省长度为 1 个字节，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。

BINARY 长度为 *length* 个字节的二进制数据。如果省略 *length*，则缺省值为 1 个字节。允许的最大大小为 255 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 *length*，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 *max-length* 个字节的二进制数据。如果省略 *max-length*，则缺省值为 1 个字节。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。

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）的二进制大对象 (BLOB) 数据（对于大小为 128KB 的 IQ 页），或零 (0) 到 2PB（千万亿字节）的二进制大对象 (BLOB) 数据（对于大小为 512KB 的 IQ 页）。最大长度为 4 GB 乘以数据库页大小。请参见《Sybase IQ 中的非结构化数据分析》。

有关 LONG BINARY 和 IMAGE 数据类型的信息，请参见附录“与其它 Sybase 数据库的兼容性”中的第 488 页的“二进制数据类型”。

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，均会截断尾随零。

以下示例创建了一个表，该表中含有使用 NULL 和 NOT NULL 定义的 BINARY 和 VARBINARY 数据类型的全部四个变体。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
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” 的形式存储在变长二进制列 (VARBINARY) 中。在定长二进制列 (BINARY) 中，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

注释 在上面的示例中，确保将 string_truncation 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));
```

从平面文件装载 ASCII 数据

从平面文件装载到二进制类型列（BINARY 或 VARBINARY）的任何 ASCII 数据都会存储为半字节。例如，如果将 0x1234 或 1234 从平面文件读取到二进制列中， 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

表 3-3 列出二进制数据的存储大小。

表 3-3：二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) 个字节的宽度	(32K - 1) 个字节的二进制数据	(32K - 1) 个字节
VARBINARY	(32K - 1) 个字节的宽度	(64K - 2) 个字节的 ASCII 数据	(32K - 1) 个字节
BINARY	255 个字节的宽度	255 个字节的二进制数据	255 字节
BINARY	255 个字节的宽度	510 个字节的 ASCII 数据	255 字节

平台相关性 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 INTOHEX 和 HEXTOINT 函数，而不是特定于平台的 CONVERT 函数。有关详细信息，请参见第 96 页的“数据类型转换函数”部分。

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

下列限制适用于包含 BINARY 和 VARBINARY 数据的列：

- 您不能将集合函数 SUM、AVG、STDDEV 或 VARIANCE 用于二进制数据类型。集合函数 MIN、MAX 和 COUNT 确实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 个字节的 VARBINARY 仅支持缺省索引和 CMP 索引类型。
- 长度等于或小于 8 个字节的 BINARY 和 VARBINARY 数据支持位操作。

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 3-4：尾随零的处理

数据类型	Sybase IQ	SQL Anywhere	ASE
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，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 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）。

```
UPDATE mytab
    SET uid_col = NEWID()
    WHERE uid_col IS NULL
```

如果您执行以下语句，

```
SELECT NEWID()
```

会返回唯一标识符 BINARY(16)。例如，其值可以是 0xd3749fe09cf446e399913bc6434f1f08。您可以使用 UIDTOSTR() 函数将此字符串转换为可读的格式。

UUID 值又称 GUID（全局唯一标识符）。

STRTOUUID 和 UIDTOSTR 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 值太大，所以使用 UNSIGNED BIGINT 或 UNSIGNED INT 标识列来替代 UNIQUEIDENTIFIER 并不会更加有效（如果您不需要跨数据库唯一标识符）。

UNIQUEIDENTIFIER 的标准和兼容性

- **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，通过 CIS 函数补偿来支持 STRTOUUID、UIDTOSTR 和 NEWID 函数。在 15.2 版及更高版本中，STRTOUUID、UIDTOSTR 和 NEWID 函数是本地 Sybase IQ 函数。

另请参见

有关 UNIQUEIDENTIFIER 的详细信息：

- [第 192 页的“NEWID 函数 \[杂项 \]”](#)
- [第 265 页的“UIDTOSTR 函数 \[字符串 \]”](#)
- [第 244 页的“STRTOUUID 函数 \[字符串 \]”](#)

Bit 数据类型

说明

存储布尔值。

数据类型	值	支持情况
BIT	0 或 1	Sybase IQ 和 Enterprise

用法

BIT 只存储值 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。

兼容性

Adaptive Server Enterprise BIT 数据类型仅允许值 0 或 1。

日期和时间数据类型

说明

存储日期和时间。

语法

DATE
DATETIME
SMALLDATETIME
TIME
TIMESTAMP

用法

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 个字节的存储空间。

DATETIME 一个以 TIMESTAMP 的形式实现的域。之所以提供 DATETIME，主要是为了与 Adaptive Server Enterprise 兼容。有关例外，请参见第 83 页的“字符串到日期时间转换的兼容性”。

SMALLDATETIME 一个以 TIMESTAMP 的形式实现的域。之所以提供 SMALLDATETIME，主要是为了与 Adaptive Server Enterprise 兼容。有关例外，请参见第 83 页的“字符串到日期时间转换的兼容性”。

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 个字节的存储空间。（ODBC 标准将 TIME 数据类型限制为精确到秒。因此，不要在要求的精度比秒更高的 WHERE 子句比较中使用 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 个字节的存储空间。

TIMESTAMP 数据类型的有效范围为 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'
```

注意

日期和时间数据支持下列索引类型:

- 所有日期和时间数据类型都支持 CMP、HG、HNG 和 LF 索引类型；WD 索引类型不受支持。
- 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。

向数据库发送日期和时间

说明

您可以通过下列方式之一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`），但可以显示在字符串中的任意位置。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：

- 以 `yyyy/mm/dd` 或 `yyyy-mm-dd` 格式的字符串形式，其格式由数据库明确解释
- 以按照 `DATE_ORDER` 数据库选项解释的字符串形式

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。

从数据库中检索日期和时间

说明

您可以通过下列方式之一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

将日期或时间作为字符串进行检索时，应使用由数据库选项 `DATE_FORMAT`、`TIME_FORMAT` 和 `TIMESTAMP_FORMAT` 指定的格式检索它。有关这些选项的说明，请参见《参考：语句和选项》中的 [SET OPTION 语句](#)。

有关用于处理日期和时间的函数的信息，请参见第 76 页的“日期和时间数据类型”。日期中允许使用下列运算符：

- **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** 创建组合给定日期和时间的日期戳。

比较日期和时间

说明

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。例如：

```
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;
```

使用明确的日期和时间

说明

格式为 `yyyy/mm/dd` 或 `yyyy-mm-dd` 的日期总是被视为日期（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防止根据用户的 `DATE_ORDER` 设置对日期做出错误的解释。

格式为 `hh:mm:ss.sss` 的字符串还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。解释取决于数据库选项 `DATE_ORDER` 的设置。

`DATE_ORDER` 数据库选项可以拥有值 “MDY”、“YMD” 或 “DMY”。请参见《参考：语句和选项》中的“[SET OPTION 语句](#)”。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”。无论何时进行连接，ODBC 驱动程序都将 `DATE_ORDER` 选项设置为 “YMD”。使用 `SET OPTION` 语句可更改该值。

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。选项 `NEAREST_CENTURY` 的值会影响对 2 位数年份的解释：在小于 `NEAREST_CENTURY` 的值中添加 2000，在所有其它值中添加 1900。此选项的缺省值为 50。因此，缺省情况下，会将 50 解释为 1950，将 49 解释为 2049。请参见《参考：语句和选项》中的“[NEAREST_CENTURY 选项 \[TSQL\]](#)”。

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。

如果使用适当的 DATE_ORDER 设置，下列字符串均为有效日期：

```
99-05-23 21:35
99/5/23
1999/05/23
May 23 1999
23-May-1999
Tuesday May 23, 1999 10:00pm
```

如果字符串只包含部分日期说明，则使用缺省值填写日期。使用下列缺省值：

年 1900

月 没有缺省值

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

说明

域是内置数据类型的别名，包括适用的精度和标度值。

域，也称为用户定义的数据类型，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。请参见《参考：语句和选项》中的“[CREATE DOMAIN 语句](#)”。

以下语句将创建名为 street_address 的数据类型，而该数据类型是一个含 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。创建数据类型后，执行 CREATE DOMAIN 语句的用户 ID 会成为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与任何其它数据类型完全相同。例如，以下表包含两个列，它的第二个列为 `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。

用户定义的数据类型的约束和缺省值

用户定义的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取代 `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）。

兼容性

- **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 `NULL` 或 `NOT NULL` 条件）创建用户定义的数据类型。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，则必须使用 `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

说明

类型转换自动进行，或者您可以使用 `CAST` 或 `CONVERT` 函数显式请求类型转换。

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，并删除所有小时、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。

有关 `CAST` 和 `CONVERT` 函数的信息，请参见第 96 页的“数据类型转换函数”。

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 都使用缺省值 January 1, 1900。而 SQL Anywhere 使用当前日期。

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 解释值的方式是相同的。

示例

- 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

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，卷 1》中的第 7 章“将数据移入和移出数据库”。

从 BIT 到 BINARY 数据类型的转换

Sybase IQ 支持从 BIT 到 BINARY 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。Sybase IQ 为比较运算符、算术运算以及 INSERT 和 UPDATE 语句将 BIT 显式转换为 BINARY 并将 BIT 转换为 VARBINARY 数据类型。

对于从 BIT 到 BINARY 的转换，将位值“b”复制到二进制字符串的第一个字节，其余字节用零填充。例如，BIT 值 1 转换为 BINARY(n) 字符串“0x0100...00”（拥有 2^n 个半元组）。BIT 值 0 转换为 BINARY 字符串 0x00...00。

对于从 BIT 到 VARBINARY 的转换，将 BIT 值“b”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

从 BIT 到 BINARY 以及从 BIT 到 VARBINARY 的转换示例 这些示例说明了从 BIT 到 BINARY 以及从 BIT 到 VARBINARY 数据类型的隐式和显式转换。

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)
```

BIT 到 BINARY 的隐式转换：

```
INSERT tbin SELECT c2 FROM tbar

c1
---
0x010000000000000000    (18 nibbles)
0x000000000000000000    (18 nibbles)
```

BIT 到 VARBINARY 的隐式转换：

```
INSERT tvarbin SELECT c2 FROM tbar

c2
---
0x01
0x00
```

BIT 到 BINARY 的显式转换：

```
INSERT tbin SELECT CONVERT (BINARY(9), c2) FROM tbar

c1
---
0x010000000000000000    (18 nibbles)
0x000000000000000000    (18 nibbles)
```

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

Sybase IQ 支持针对比较运算符、算术运算以及 INSERT 和 UPDATE 语句在 BIT 和 CHAR 以及 BIT 和 VARCHAR 数据类型之间进行隐式转换。

BIT 到 VARCHAR、CHAR 到 BIT 以及 VARCHAR 到 BIT 的转换示例 以下示例说明 BIT 和 CHAR 以及 BIT 和 VARCHAR 数据类型之间的隐式和显式转换。

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)
```

BIT 到 VARCHAR/VARCHAR 到 BIT 的隐式转换和 BIT 到 VARCHAR 的隐式转换：

```
INSERT tvarchar SELECT c2 FROM tbar
SELECT c2, char_length(c2) FROM tvarchar

c2,char_length(tvarchar.c2)
-----
'1',1
'0',1
```

VARCHAR 到 BIT 的隐式转换：

```
INSERT tbit SELECT c2 FROM tvarchar
SELECT c2 FROM tbit

c2
--
0
1
```

BIT 到 CHAR/CHAR 到 BIT 的显式转换和 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
```

CHAR 到 BIT 的显式转换:

```
INSERT tbit SELECT CONVERT (BIT, c1) FROM tchar
SELECT c2 FROM tbit
```

```
c2
--
0
1
```

BIT 到 VARCHAR/VARCHAR 到 BIT 的显式转换和 BIT 到 VARCHAR 的显式转换:

```
INSERT tvarchar SELECT CONVERT (VARCHAR(9), c2)
FROM tbar
SELECT c2, char_length(c2) FROM tvarchar
```

```
c2,char_length(tvarchar.c2)
-----
'1',1
'0',1
```

VARCHAR 到 BIT 的显式转换:

```
INSERT tbit SELECT CONVERT (BIT, c2) FROM tvarchar
SELECT c2 FROM tbit
```

```
c2
--
0
1
```


关于本章

本章介绍了 Sybase IQ 支持的内置函数。

目录

主题	页码
概述	89
集合函数	90
分析函数	92
数据类型转换函数	96
日期和时间函数	97
HTTP 函数	101
数值函数	102
字符串函数	103
系统函数	105
SQL 和 Java 用户定义的函数	109
杂项函数	110
按字母顺序排列的函数列表	111

概述

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。

对 Sybase IQ 使用函数时：

- 除非另外指定，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。

集合函数

作用

集合函数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。

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的一组行中的数据。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。
- 使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。请参见第 92 页的“分析函数”。

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

- 用于进行时序分析的 ISO/ANSI SQL:2008 OLAP 函数包括：CORR()、COVAR_POP()、COVAR_SAMP()、CUME_DIST()、FIRST_VALUE()、LAST_VALUE()、REGR_AVGX()、REGR_AVGY()、REGR_COUNT()、REGR_INTERCEPT()、REGR_R2()、REGR_SLOPE()、REGR_SXX()、REGR_SXY() 和 REGR_SYY()。
- 在数据库行业使用的非 ISO/ANSI SQL:2008 OLAP 集合函数扩展包括 FIRST_VALUE()、MEDIAN() 和 LAST_VALUE()。
- 用于计算加权移动平均值的加权 OLAP 集合函数包括 EXP_WEIGHTED_AVG() 和 WEIGHTED_AVG()。

专门为财务时序预测和分析而设计的时序函数的名称以 “TS_” 开头。有关这些时序函数的信息，请参见《时序指南》。

表 4-1: 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)
MAX	([DISTINCT] { <i>column-name</i> <i>numeric-expr</i> })
MEDIAN	(expression)
MIN	([DISTINCT] { <i>column-name</i> <i>numeric-expr</i> })
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] <i>expression</i>)
SUM	([DISTINCT] { <i>column-name</i> <i>numeric-expr</i> })
VARIANCE	([ALL] <i>expression</i>)
WEIGHTED_AVG	(<i>expression</i> , <i>period-expression</i>)

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

有关使用每个函数的特定详细信息，请参见本章中的各个分析函数的说明。

有关使用 OLAP 函数的详细信息，请参见《系统管理指南，卷 2》中的第 2 章“使用 OLAP”。

有关支持 LONG BINARY 和 LONG VARCHAR 数据类型的集合函数的信息，请参见《Sybase IQ 中的非结构化数据分析》中的第 9 章“函数支持”。

分析函数

作用

分析函数包括：

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

注释 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。

表 4-2 列出了分析函数及其参数。不同于表 4-1 中的某些集合函数，您不能在窗口函数中指定 DISTINCT。

表 4-2: 分析函数

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

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

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。您可以对窗口使用三类窗口函数：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...)
```

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 `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` 子句仅在 `OVER` 子句中使用，并且不是用于 `SELECT` 的 `ORDER BY`。不允许排序查询 `ROW` 中的任何集合函数指定 `DISTINCT`。

注释 `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。升序顺序是缺省值。

`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。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。在 Sybase IQ 15.2 中，允许在一个查询中使用多个 `rank` 分析函数。

统计集合分析函数的用法

汇总数据库的一组行中的数据。这些组是使用 `SELECT` 语句的 `GROUP BY` 子句构成的。只能在选择列表中以及 `SELECT` 语句的 `HAVING` 和 `ORDER BY` 子句中使用集合函数。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，并对在 `WITHIN GROUP` 子句中指定的一组数据进行操作，或者对整个数据集进行操作。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）。

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。升序顺序是缺省值。

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 函数提供对表或分区中位于 CURRENT ROW 前面且与之相距给定物理偏移量的行的访问。LEAD 函数提供对表或分区中位于 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，卷 2》的第 2 章 “使用 OLAP” 中的 “行间函数用法”。

行间函数需要使用 OVER (ORDER_BY) 子句。请参见第 93 页的 “窗口化集合函数的用法”。

兼容性

Adaptive Server Enterprise 不支持排名函数和逆分布分析函数。

另请参见

有关使用每个函数的特定详细信息，请参见本章中的各个分析函数的说明。

请参见《系统管理指南，卷 2》中的第 2 章 “使用 OLAP”。

有关支持 LONG BINARY 和 LONG VARCHAR 数据类型的分析函数的信息，请参见《Sybase IQ 中的非结构化数据分析》中的第 9 章 “函数支持”。

数据类型转换函数

作用

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。

表 4-3 列出了数据类型转换函数及其参数。

表 4-3: 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 AS data type</i>)
CONVERT	(<i>data type, expression [,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>)

说明

第 97 页的“日期和时间函数”中列出了 DATE、DATETIME、DATEFORMAT 和 YMD 函数，这些函数根据日期格式将表达式转换为日期、时间戳或字符串。第 103 页的“字符串函数”一节讨论了 STRING 函数，该函数将表达式转换为字符串。

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。有关 Sybase IQ 执行的数据类型自动转换的详细信息，请参见第 83 页的“数据类型转换”。

日期和时间函数

作用

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。

表 4-4 和表 4-5 列出了日期和时间函数及其参数。

语法 1

表 4-4：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>)
HOUR	(<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>)
MINUTES	(<i>datetime-expr</i>)

日期和时间函数	参数
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 4-5: 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	()

说明

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。

表 4-4 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）。

表 4-5 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。如果您发出不包含 FROM 子句的 SELECT 语句，则该语句会传递给 SQL Anywhere。下面的语句由 SQL Anywhere 以独占方式处理：

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

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

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

请考虑另一示例。MONTHS 函数返回自“任意开始日期”以来的月数。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。

注释 创建只含一个列和一个行的虚拟表。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根据日期分量生成的日期。表 4-6 显示了 *date-part* 的允许值。

表 4-6: 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	整数。一周开始的年份。包含一年中的前几天的周可以是上一年中最后一周的一部分，具体取决于它开始于哪一天。如果新年的开始日期为星期四到星期六，则它的第一周开始于上一年的最后一个星期日。如果新年的开始日期为星期日到星期三，则它的任何一天都不是上一年的一部分。
Calweekofyear	cwk	一个从 1 到 54 的整数，表示年中包含指定日期的周编号。
Caldayofweek	cdw	周中的日期编号 (星期日 = 1, 星期六 = 7)。

注释 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》中的 [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。

HTTP 函数

作用

使用 HTTP 函数可便于处理 Web 服务中的 HTTP 请求。

表 4-7 列出了所有 HTTP 函数及其参数。

表 4-7: 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>])
NEXT_HTTP_HEADER	(<i>header-name</i>)
NEXT_HTTP_VARIABLE	(<i>var-name</i>)

数值函数

作用

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。

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。

注释 请考虑创建一个虚拟表以在这种情况下使用。

表 4-8 列出了数值函数及其参数。

表 4-8: 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>)
LOG	(<i>numeric-expr</i>)
LOG10	(<i>numeric-expr</i>)
MOD	(<i>dividend</i> , <i>divisor</i>)
PI	(*)
POWER	(<i>numeric-expr1</i> , <i>numeric-expr2</i>)
RADIANS	(<i>numeric-expr</i>)
RAND	([<i>integer-expr</i>])
REMAINDER	(<i>numeric-expr</i> , <i>numeric-expr</i>)
ROUND	(<i>numeric-expr</i> , <i>integer-expr</i>)
SIGN	(<i>numeric-expr</i>)
SIN	(<i>numeric-expr</i>)

数值函数	参数
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

作用

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。请参见第 281 页的表 6-1 中的“Field Size”列。

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。

表 4-9 列出了字符串函数及其参数。

表 4-9: 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>)
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>])

字符串函数	参数
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》中的第 9 章“函数支持”。

系统函数

作用

系统函数返回系统信息。

表 4-10 列出了系统函数及其参数。

表 4-10: 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>])
DATALLENGTH	(<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> })

系统函数	参数
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>])

说明 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 表 4-11 显示了 Sybase IQ 中的 Adaptive Server Enterprise 系统函数及其状态：

表 4-11: 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

函数	状态
host_id	未实现
host_name	未实现
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

注意

- 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 函数作为内置函数来实现。

连接属性

连接属性适用于单个连接。本节介绍了如何检索特定连接属性的值或所有连接属性的值。有关所有连接属性的说明，请参见 SQL Anywhere 文档中的“连接属性”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — 数据库管理” > “配置数据库” > “连接、数据库和数据库服务器属性”。连接属性 QueryBypassedCosted、QueryBypassedOptimized、QueryDescribedOptimizer 和 StatementPostAnnotatesSimple 适用于 SQL Anywhere 对象，但不适用于 IQ 表。

示例

❖ 检索连接属性的值

- 使用 connection_property 系统函数。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' )
```

❖ 检索所有连接属性的值

- 使用 sa_conn_properties 系统过程：

```
call sa_conn_properties
```

将针对每个属性为每个连接显示一个单独的行。

可用于服务器的属性

服务器属性作为整体应用于整个服务器中。本节介绍了如何检索特定服务器属性的值或所有服务器属性的值。有关所有服务器属性的说明，请参见 SQL Anywhere 文档中的“数据库服务器属性”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — 数据库管理” > “配置数据库” > “连接、数据库和数据库服务器属性”。

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。请参见第 360 页的“`sp_iqlmconfig` 过程”。

示例

❖ 检索服务器属性的值

- 使用 `property` 系统函数。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
```

❖ 检索所有服务器属性的值

- 使用 `sa_eng_properties` 系统过程：

```
call sa_eng_properties
```

可用于每个数据库的属性

数据库属性应用于整个数据库。本节介绍了如何检索特定数据库属性的值或所有数据库属性的值。有关所有数据库属性的说明，请参见 SQL Anywhere 文档中的“数据库服务器属性”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — 数据库管理” > “配置数据库” > “连接、数据库和数据库服务器属性”。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

❖ 检索数据库属性的值

- 使用 `db_property` 系统函数。下面的语句返回当前数据库的页大小：

```
select db_property ( 'PageSize') from iq_dummy
```

❖ 检索所有数据库属性的值

- 使用 `sa_db_properties` 系统过程：

```
call sa_db_properties
```

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。

注释 用户定义的函数由 SQL Anywhere 来处理。它们不能利用 Sybase IQ 的性能特征的优势。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 数据类型视为截然不同的数据类型，而 SQL Anywhere 将 CHAR 数据视为 VARCHAR。

SQL 中用户定义的函数

您可以在 SQL 中使用 CREATE FUNCTION 语句实现您自己的函数。CREATE FUNCTION 语句内的 RETURN 语句确定函数的数据类型。

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。

注释 避免在具有用户定义的函数的视图中使用 CONTAINS 谓词，因为系统会忽略 CONTAINS 条件。请改用 LIKE 谓词，或发出查询时不针对视图。

有关创建 SQL 函数的详细信息，请参见《系统管理指南，卷 2》中的第 1 章“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 类以及类方法的详细信息，请参见 SQL Anywhere 文档中的“SQL Anywhere 中的 Java 支持”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — 编程” > “数据库中的 Java”。

时序和预测函数

作用 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。

注释 时序和预测功能仅可用于 RAP — Trading Edition Enterprise。

杂项函数

作用 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。

表 4-12 列出了杂项函数及其参数。

表 4-12: 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]

作用	返回数值表达式的绝对值。
语法	ABS (<i>numeric-expression</i>)
参数	numeric-expression 要返回其绝对值的数字。
示例	下面的语句返回值 66: <pre>SELECT ABS(-66) FROM iq_dummy</pre>
标准和兼容性	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 与 Adaptive Server Enterprise 兼容。

ACOS 函数 [数值]

作用	返回数值表达式的反余弦值（以弧度表示）。
语法	ACOS (<i>numeric-expression</i>)
参数	numeric-expression 角度的余弦值。
示例	下面的语句返回值 1.023945: <pre>SELECT ACOS(0.52) FROM iq_dummy</pre>
标准和兼容性	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 与 Adaptive Server Enterprise 兼容。

另请参见	第 113 页的“ASIN 函数 [数值]” 第 113 页的“ATAN 函数 [数值]” 第 114 页的“ATAN2 函数 [数值]” 第 128 页的“COS 函数 [数值]”
------	--

ARGN 函数 [杂项]

作用	从参数列表中返回所选参数。
语法	ARGN (<i>integer-expression</i> , <i>expression</i> [, ...])
参数	integer-expression 表达式列表中的参数位置。 expression 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。
示例	下面的语句返回值 6: <pre>SELECT ARGN(6, 1,2,3,4,5,6) FROM iq_dummy</pre>
用法	使用 <i>integer-expression</i> 的值作为 <i>n</i> ，从其余参数的列表中返回第 <i>n</i> 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	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 不受 Adaptive Server Enterprise 支持。

ASCII 函数 [字符串]

作用	返回字符串表达式中第一个字节的整数 ASCII 值。
语法	ASCII (<i>string-expression</i>)
参数	string-expression 字符串
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: <pre>SELECT ASCII('Z') FROM iq_dummy</pre>
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	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 与 Adaptive Server Enterprise 兼容。

ASIN 函数 [数值]

作用 返回一个数字的反正弦值（以弧度表示）。

语法 **ASIN** (*numeric-expression*)

参数 **numeric-expression** 角度的正弦值

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

[第 111 页的“ACOS 函数 \[数值 \]”](#)

[第 113 页的“ATAN 函数 \[数值 \]”](#)

[第 114 页的“ATAN2 函数 \[数值 \]”](#)

[第 232 页的“SIN 函数 \[数值 \]”](#)

ATAN 函数 [数值]

作用 返回一个数字的反正切值（以弧度表示）。

语法 **ATAN** (*numeric-expression*)

参数 **numeric-expression** 角度的正切值

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

[第 111 页的“ACOS 函数 \[数值 \]”](#)

[第 113 页的“ASIN 函数 \[数值 \]”](#)

[第 114 页的“ATAN2 函数 \[数值 \]”](#)

[第 249 页的“TAN 函数 \[数值 \]”](#)

ATAN2 函数 [数值]

作用 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 **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。

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 [第 111 页的“ACOS 函数 \[数值 \]”](#)
[第 113 页的“ASIN 函数 \[数值 \]”](#)
[第 113 页的“ATAN 函数 \[数值 \]”](#)
[第 249 页的“TAN 函数 \[数值 \]”](#)

AVG 函数 [集合]

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

示例 下面的语句返回值 49988.6:

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

用法 该平均值不包括 *numeric-expression* 的值为空的行。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 [第 131 页的“COUNT 函数 \[集合 \]”](#)
[第 247 页的“SUM 函数 \[集合 \]”](#)
《系统管理指南，卷 2》中的第 2 章“使用 OLAP”

BFILE 函数 [数据提取]

- 作用** 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 数据提取功能包含 BFILE 函数，可用于将各 LONG BINARY 和 LONG VARCHAR 单元提取到服务器上的单独操作系统文件中。BFILE 函数不一定要与数据提取功能一起使用。
- 请参见《Sybase IQ 中的非结构化数据分析》中的第 9 章“函数支持”。
- 用户必须经过专门授权才能使用大对象数据类型 LONG BINARY 和 LONG VARCHAR。有关非结构化数据分析选件的详细信息，请参见《Sybase IQ 中的非结构化数据分析》。

BIGINTTOHEX 函数 [数据类型转换]

- 作用** 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** **integer-expression** 要转换成十六进制的整数。
- 示例** 下面的语句返回值 0000000000000009:
- ```
SELECT BIGINTTOHEX(9) FROM iq_dummy
```
- 下面的语句返回值 FFFFFFFFFFFFFFFF7:
- ```
SELECT BIGINTTOHEX (-9) FROM iq_dummy
```
- 用法** BIGINTTOHEX 函数接受计算结果为 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，结果为空。
- 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** 《参考：语句和选项》中的 [CONVERSION_ERROR 选项 \[TSQL\]](#)
[第 163 页的“HEXTOBIGINT 函数 \[数据类型转换\]”](#)
[第 164 页的“HEXTOINT 函数 \[数据类型转换\]”](#)
[第 172 页的“INTTOHEX 函数 \[数据类型转换\]”](#)

BIT_LENGTH 函数 [字符串]

作用	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 (<i>column-name</i>)
参数	column-name 列的名称
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。
标准和兼容性	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 不受 SQL Anywhere 或 Adaptive Server Enterprise 支持。
另请参见	第 200 页的 “OCTET_LENGTH 函数 [字符串]” 《Sybase IQ 中的非结构化数据分析》中的第 9 章 “函数支持”

BYTE_LENGTH 函数 [字符串]

作用	返回字符串中的字节数。
语法	BYTE_LENGTH (<i>string-expression</i>)
参数	string-expression 要计算其长度的字符串
示例	下面的语句返回值 12: <pre>SELECT BYTE_LENGTH('Test Message') FROM iq_dummy</pre>
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。 如果字符串在多字节字符集中，则 BYTE_LENGTH 值不同于 CHAR_LENGTH 返回的字符数。
标准和兼容性	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 不受 Adaptive Server Enterprise 支持。
另请参见	第 120 页的 “CHAR_LENGTH 函数 [字符串]” 第 132 页的 “DATALENGTH 函数 [系统]” 第 181 页的 “LENGTH 函数 [字符串]” 《Sybase IQ 中的非结构化数据分析》中的第 9 章 “函数支持”

BYTE_LENGTH64 函数

- 作用** BYTE_LENGTH64 函数返回无符号的 64 位值，该值包含 LONG BINARY 列参数的字节长度。
- 用法** BYTE_LENGTH64 还支持 LONG VARCHAR 数据类型以及任意数据大小的 LONG BINARY 和 LONG VARCHAR 变量。
- 请参见《Sybase IQ 中的非结构化数据分析》中的第 9 章“函数支持”。
- 用户必须经过专门授权才能使用大对象数据类型 LONG BINARY 和 LONG VARCHAR。有关非结构化数据分析选件的详细信息，请参见《Sybase IQ 中的非结构化数据分析》。

BYTE_SUBSTR64 和 BYTE_SUBSTR 函数

- 作用** BYTE_SUBSTR64 和 BYTE_SUBSTR 函数返回 LONG BINARY 列参数的长二进制字节子字符串。
- 用法** BYTE_SUBSTR64 和 BYTE_SUBSTR 函数还支持 LONG VARCHAR 数据类型以及任意数据大小的 LONG BINARY 和 LONG VARCHAR 变量。
- 请参见《Sybase IQ 中的非结构化数据分析》中的第 9 章“函数支持”。
- 用户必须经过专门授权才能使用大对象数据类型 LONG BINARY 和 LONG VARCHAR。有关非结构化数据分析选件的详细信息，请参见《Sybase IQ 中的非结构化数据分析》。

CAST 函数 [数据类型转换]

- 作用** 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** **expression** 要转换的表达式
data type 目标数据类型
- 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：

```
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 NULL 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
```

说明为：

```
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

[第 124 页的“CONVERT 函数 \[数据类型转换\]”](#)

CEIL 函数 [数值]

作用

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

expression 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 会生成错误。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。

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

[第 119 页的“CEILING 函数 \[数值\]”](#)。

《系统管理指南，卷 1》中的[第 411 页的“有关国际语言和字符集的任务”](#)。

CEILING 函数 [数值]

作用 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 **numeric-expression** 要计算其上限的数字

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

[第 159 页的“FLOOR 函数 \[数值 \]”](#)

CHAR 函数 [字符串]

作用 返回含数字所表示的 ASCII 值的字符。

语法 **CHAR** (*integer-expression*)

参数 **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。

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]

作用 返回字符串中的字符数。

语法 **CHAR_LENGTH** (*string-expression*)

参数 **string-expression** 要计算其长度的字符串

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。

如果字符串在多字节字符集中，则 CHAR_LENGTH 值可能会小于 BYTE_LENGTH 值。

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 [第 116 页的“BYTE_LENGTH 函数 \[字符串 \]”](#)

《Sybase IQ 中的非结构化数据分析》中的 [第 9 章“函数支持”](#)

CHAR_LENGTH64 函数

作用 CHAR_LENGTH64 函数返回无符号的 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 还支持任意数据大小的 LONG VARCHAR 变量。

请参见《Sybase IQ 中的非结构化数据分析》中的 [第 9 章“函数支持”](#)。

用户必须经过专门授权才能使用大对象数据类型 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》。

CHARINDEX 函数 [字符串]

作用 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 **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。

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	Moira

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。

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

[第 246 页的“SUBSTRING 函数 \[字符串 \]”](#)

《Sybase IQ 中的非结构化数据分析》中的 [第 9 章“函数支持”](#)

COALESCE 函数 [杂项]

作用 返回列表中的第一个非空表达式。

语法 **COALESCE** (*expression*, *expression* [, ...])

参数 **expression** 任意表达式

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。

COL_LENGTH 函数 [系统]

作用 返回定义的列长度。

语法 **COL_LENGTH** (*table-name*, *column-name*)

参数 **table-name** 表名

column-name 列名

示例 下面的语句返回列长度 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

[第 132 页的 “DATALENGTH 函数 \[系统 \]”](#)

COL_NAME 函数 [系统]

作用 返回列名。

语法 **COL_NAME** (*table-id*, *column-id* [, *database-id*])

参数 **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

[第 148 页的“DB_ID 函数 \[系统 \]”](#)

[第 199 页的“OBJECT_ID 函数 \[系统 \]”](#)

CONNECTION_PROPERTY 函数 [系统]

作用 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 ] )
```

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 3 章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

参数

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。如果您提供 *integer-expression1*，则它将是连接属性 ID。可以使用 `PROPERTY_NUMBER` 函数确定此值。

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。

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

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

另请参见

[第 107 页的“连接属性”](#)

[第 209 页的“PROPERTY_NUMBER 函数 \[系统 \]”](#)

CONVERT 函数 [数据类型转换]

作用 返回转换成提供的数据类型的表达式。

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

参数 **data-type** 表达式将转换成的数据类型

expression 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。

表 4-13 列出了 *format-style* 参数值的含义。

如果未提供任何 *format-style* 参数，将使用数据库选项设置。

表 4-13: 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)
10	110	mm-dd-yy[yy]
11	111	[yy]yy/mm/dd
12	112	[yy]yymmdd
-	13 或 113	dd mmm yyyy hh:nn:ss:sss (24 小时制, 欧洲缺省时间 + 毫秒, 4 位数年份)
14	114	hh:nn:ss (24 小时制)
-	20 或 120	yyyy-mm-dd hh:nn:ss (24 小时制, ODBC 规范, 4 位数年份)
-	21 或 121	yyyy-mm-dd hh:nn:ss:sss (24 小时制, ODBC 规范加毫秒, 4 位数年份)
36	136	hh:nn:ss.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

不含世纪 (yy)	含世纪 (yyyy)	输出
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
ssssss	microsecond	0 - 999999
mmm	month	一月到十二月
dd	day	1 - 31
yyyy	year	0001 - 9999
mm	month	1 - 12

示例

下列语句说明了格式样式的使用法:

```
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。如果您在 SELECT INTO 语句中使用 CONVERT，则您必须具有一个非结构化数据分析选件许可证，或者使用 CAST 并将 CONVERT 设置为正确的数据类型和大小。

有关详细信息，请参见 [“REPLACE 函数 \[字符串 \]”](#)。

- 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 [第 117 页的“CAST 函数 \[数据类型转换\]”](#)

CORR 函数 [集合]

作用 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*)

window-spec: 请参见下面“用法”部分中“语法 2”的说明。

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

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，采用双精度浮点执行计算并返回 DOUBLE 作为结果。如果将 CORR 应用于空集，它将返回 NULL。

dependent-expression 和 *independent-expression* 都是数值。在消除 *dependent-expression* 或 *independent-expression* 为 NULL 的对之后，会将函数应用于 *dependent-expression* 和 *independent-expression* 对的集合。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*。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见 [第 92 页的“分析函数”](#)。

- 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。

COS 函数 [数值]

作用 返回一个数字的余弦值（以弧度表示）。

语法 **COS** (*numeric-expression*)

参数 **numeric-expression** 角度（以弧度为单位）。

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

[第 111 页的“ACOS 函数 \[数值 \]”](#)

[第 128 页的“COT 函数 \[数值 \]”](#)

[第 232 页的“SIN 函数 \[数值 \]”](#)

[第 249 页的“TAN 函数 \[数值 \]”](#)

COT 函数 [数值]

作用 返回一个数字的余切值（以弧度表示）。

语法 **COT** (*numeric-expression*)

参数 **numeric-expression** 角度（以弧度为单位）。

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

[第 128 页的“COS 函数 \[数值 \]”](#)

[第 232 页的“SIN 函数 \[数值 \]”](#)

[第 249 页的“TAN 函数 \[数值 \]”](#)

COVAR_POP 函数 [集合]

作用 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)`

window-spec: 请参见下面“用法”部分中“语法 2”的说明。

参数 **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 NULL。

dependent-expression 和 *independent-expression* 都是数值。在消除 *dependent-expression* 或 *independent-expression* 为 NULL 的对之后，会将函数应用于 *dependent-expression* 和 *independent-expression* 对的集合。将进行以下计算：

$$(\text{SUM}(x*y) - \text{SUM}(x) * \text{SUM}(y) / n) / n$$

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

请参见 SQL Anywhere 文档中的“集合函数的数学公式”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

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]

作用 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*)

window-spec: 请参见下面“用法”部分中“语法 2”的说明。

参数 **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 NULL。

dependent-expression 和 *independent-expression* 都是数值。在消除 *dependent-expression* 或 *independent-expression* 为 NULL 的对之后，会将函数应用于 *dependent-expression* 和 *independent-expression* 对的集合。

$(\text{SUM}(x*y) - \text{SUM}(x) * \text{SUM}(y) / n) / (n-1)$

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

请参见 SQL Anywhere 文档中的“集合函数的数学公式”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

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]

作用 根据指定的参数计算组中的行数。

语法 `COUNT (* | expression | DISTINCT column-name)`

参数 * 返回每组中的行数。

注释 当显示查询结果时，* 不会显示在列标题中，而是显示为：

Count ()

expression 返回每一组中 *expression* 值不为空值的行数。

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。

示例 下列语句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

[第 114 页的 “AVG 函数 \[集合 \]”](#)

[第 247 页的 “SUM 函数 \[集合 \]”](#)

《系统管理指南，卷 2》中的 [第 2 章 “使用 OLAP”](#)

CUME_DIST 函数 [分析]

作用 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)`

window-spec: 请参见下面的 “用法” 一节。

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 DepartmentID
ORDER BY Salary DESC) "Rank"
FROM Employees
WHERE State IN ('CA');
```

返回的结果集为：

表 4-14: 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，CUME_DIST 函数中不允许使用排序键组合。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，不能包含 ROWS 或 RANGE 子句。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

注释 不支持 DISTINCT。

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]

作用

返回表达式的长度（以字节为单位）。

语法

DATALENGTH (*expression*)

参数

expression 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。

用法

表 4-15 列出了 DATALENGTH 的返回值。

表 4-15: 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 下面的语句返回值 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 [第 120 页的“CHAR_LENGTH 函数 \[字符串 \]”](#)
[第 122 页的“COL_LENGTH 函数 \[系统 \]”](#)

DATE 函数 [日期和时间]

作用 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 **expression** 要转换成日期格式的值。表达式通常是字符串。

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]

作用 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 **date part** 要添加到日期中的日期分量。

有关允许的日期分量的完整列表，请参见[第 100 页的“日期分量”](#)。

numeric-expression 要添加到日期中的日期分量的数量。*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 mins 47 secs 483647 mcs)。

date-expression 要修改的日期。

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。

DATECEILING 函数 [日期和时间]

作用

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

date-part 要添加到日期中的日期分量。

下面的日期分量与 DATECEILING 不兼容:

- DayofYear
- WeekDay
- CalYearofWeek
- CalWeekofYear
- CalDayofWeek

有关日期分量的完整列表，请参见第 100 页的“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 NULL，则该函数的结果也为 NULL。

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。

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** 不受 Adaptive Server Enterprise 或 SQL Anywhere 支持。

DATEDIFF 函数 [日期和时间]

作用 返回两个日期之间的时间间隔。

语法 `DATEDIFF (date-part, date-expression1, date-expression2)`

参数 **date-part** 指定要测量其时间间隔的日期分量。

有关允许的日期分量的完整列表，请参见第 100 页的“日期分量”。

date-expression1 时间间隔的开始日期。从 *date-expression2* 中减去该值，以返回两个参数之间日期分量的数量。

date-expression2 时间间隔的结束日期。从此值中减去 *date-expression1* 以返回两个参数间的日期分量的数量。

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：在日期分量中，带符号的整数值 = (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 会返回两个指定的时间之间（包括第二个日期但不包括第一个日期）的午夜数。例如，下面的语句返回值 5。第一天 2003/08/03 的午夜未包括在结果中。第二天的午夜包括在内，即使指定的时间在午夜之前也是如此。

```
SELECT DATEDIFF( DAY, '2003/08/03 14:00', '2003/08/08  
14:00' ) FROM iq_dummy
```

当使用 **month** 作为日期分量时，DATEDIFF 会返回两个日期之间（包括第二个日期但不包括第一个日期）出现的月的第一天的数目。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。

使用 **week** 作为日期分量时，DATEDIFF 会返回两个日期（包括第二个日期但不包括第一个日期）之间的星期日的数量。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。

DATEFLOOR 函数 [日期和时间]

作用 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 **date part** 要添加到日期中的日期分量。

下面的日期分量与 DATEFLOOR 不兼容：

- DayofYear
- WeekDay
- CalYearofWeek
- CalWeekofYear
- CalDayofWeek

有关日期分量的完整列表，请参见第 100 页的“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 NULL，则该函数的结果也为 NULL。

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。

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** 不受 Adaptive Server Enterprise 或 SQL Anywhere 支持。

DATEFORMAT 函数 [日期和时间]

作用 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 **datetime-expression** 要转换的日期 / 时间。必须是日期、时间、时间戳或字符串。

string-expression 转换后的日期格式。

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，但是也可以是 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;
```

标准和兼容性

- **SQL** ISO/ANSI SQL 语法的供应商扩展。
- **Sybase** I 与 SQL Anywhere 兼容。

另请参见

《参考：语句和选项》中的 [DATE_FORMAT 选项](#)

DATENAME 函数 [日期和时间]

作用 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 **date-part** 要指定的日期分量。

有关允许的日期分量的完整列表，请参见第 100 页的“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*。

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。

DATEPART 函数 [日期和时间]

作用	返回日期 / 时间值的指定部分的整数值。
语法	DATEPART (<i>date-part</i> , <i>date-expression</i>)
参数	date-part 要返回的日期分量。 有关允许的日期分量的完整列表, 请参见第 100 页的“日期分量”。 date-expression 要返回其日期分量的日期。日期必须包含 <i>date-part</i> 字段。
示例	下面的语句返回值 5: <pre>SELECT DATEPART(MONTH, '1987/05/02') FROM iq_dummy</pre> 下面的语句返回值 722,001: <pre>SELECT DATEPART(MICROSECOND, '2009-11-10 14:57:52.722001') FROM iq_dummy</pre> 下面的语句返回值 777,777: <pre>SELECT DATEPART(MICROSECOND, '2000/07/07 07:07:07.777777') FROM iq_dummy</pre> 下面的语句返回值 33,189: <pre>SELECT DATEPART(MCS, '2009-11-03 11:10:42.033189') FROM iq_dummy</pre>
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	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的 Transact-SQL 扩展。• Sybase 与 Adaptive Server Enterprise 兼容。

DATEROUND 函数 [日期和时间]

作用 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 **date part** 要添加到日期中的日期分量。

下面的日期分量与 `DATEROUND` 不兼容：

- `DayofYear`
- `WeekDay`
- `CalYearofWeek`
- `CalWeekofYear`
- `CalDayofWeek`

有关日期分量的完整列表，请参见第 100 页的“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` 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 `NULL`，则该函数的结果也为 `NULL`。

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

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

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

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

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

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

下面的语句返回值 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，则此函数将日期和时间舍入为指定粒度的最接近的指定倍数。

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。

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** 不受 Adaptive Server Enterprise 或 SQL Anywhere 支持。

DATETIME 函数 [日期和时间]

作用 将表达式转换成时间戳。

语法 **DATETIME** (*expression*)

参数 **expression** 要转换的 *expression*。表达式通常是字符串。可能会报告转换错误。

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** 不受 Adaptive Server Enterprise 支持。

DAY 函数 [日期和时间]

作用 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 **date-expression** 日期。

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]

作用 返回指定日期中的星期几的名称。

语法 `DAYNAME (date-expression)`

参数 `date-expression` 日期。

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]

作用 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。

语法 `DAYS (datetime-expression)`
`| (datetime-expression, datetime-expression)`
`| (datetime-expression, integer-expression)`

参数 `datetime-expression` 日期和时间。

integer-expression 要添加到 *datetime-expression* 中的天数。如果 *integer-expression* 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 *datetime-expression* 显式转换为日期。

有关转换数据类型的信息，请参见第 117 页的“CAST 函数 [数据类型转换]”。

DAYS 忽略小时、分钟和秒。

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。

DB_ID 函数 [系统]

作用 返回数据库 ID 号。

语法 **DB_ID** ([*database-name*])

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 3 章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

参数 **database-name** 包含数据库名的字符串表达式。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。

示例 下面的语句在 *iqdemo* 是唯一正在运行的数据库时返回值 0。

```
SELECT DB_ID( 'iqdemo' ) FROM iq_dummy
```

下面的语句在对唯一正在运行的数据库执行时返回值 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

[第 148 页的“DB_NAME 函数 \[系统 \]”](#)

[第 199 页的“OBJECT_ID 函数 \[系统 \]”](#)

DB_NAME 函数 [系统]

作用 返回数据库名。

语法 **DB_NAME** ([*database-id*])

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 3 章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

参数 **database-id** 数据库的 ID。 *database-id* 必须是数值表达式。

示例 下面的语句在对演示数据库执行时返回数据库名 *iqdemo*:

```
SELECT DB_NAME( 0 ) FROM iq_dummy
```

用法 如果未提供 *database-id*，则会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
- 第 122 页的“COL_NAME 函数 [系统]”
 - 第 148 页的“DB_ID 函数 [系统]”
 - 第 199 页的“OBJECT_NAME 函数 [系统]”

DB_PROPERTY 函数 [系统]

作用 返回给定属性的值。

语法 **DB_PROPERTY** ({ *property-id* | *property-name* }
[, { *database-id* | *database-name* }])

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 3 章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

- 参数
- 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。
 - database-name** 数据库的名称，由 DB_NAME 返回。

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
- 第 108 页的“可用于每个数据库的属性”
 - 第 148 页的“DB_ID 函数 [系统]”
 - 第 148 页的“DB_NAME 函数 [系统]”

DEGREES 函数 [数值]

作用 将数字从弧度转换为度数。

语法 **DEGREES** (*numeric-expression*)

参数 **numeric-expression** 以弧度表示的角度。

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]

作用 排列组中的项目。

语法 **DENSE_RANK** () **OVER** (**ORDER BY** *expression* [**ASC** | **DESC**])

参数 **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。

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。存在并列排名时 **RANK** 会留空位。

DENSE_RANK 需要 **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** 子句仅在 **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。升序顺序是缺省值。

DENSE_RANK 只能在 SELECT 或 INSERT 语句的选择列表中或者在 SELECT 语句的 ORDER BY 子句中使用。DENSE_RANK 可以在视图中或联合中使用。您不能在子查询中、HAVING 子句中，或者 UPDATE 或 DELETE 语句的选择列表中使用 DENSE_RANK 函数。每个查询仅允许使用一个 rank 分析函数。

标准和兼容性

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

另请参见

[第 92 页的“分析函数”](#)

[第 211 页的“RANK 函数 \[分析\]”](#)

《系统管理指南，卷 2》中的 [第 2 章“使用 OLAP”](#)

DIFFERENCE 函数 [字符串]

作用

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

string-expression1 要比较的第一个字符串。

string-expression2 要比较的第二个字符串。

示例

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
```

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

[第 236 页的“SOUNDEX 函数 \[字符串 \]”](#)

DOW 函数 [日期和时间]

作用

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

date-expression 日期。

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》中的 [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]

作用

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

sqlstate 要返回其错误消息的 SQLSTATE 值。

sqlcode 要返回其错误消息的 SQLCODE 值。

示例

下面的语句返回 SQLCODE -813 的错误消息:

```
select errmsg( -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	<ul style="list-style-type: none"> • SQL ISO/ANSI SQL 语法的供应商扩展。 • Sybase 不受 Adaptive Server Enterprise 支持。
另请参见	《系统管理指南，卷 2》

EVENT_CONDITION 函数 [系统]

作用 指定何时触发事件处理程序。

语法 **EVENT_CONDITION** (*condition-name*)

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 3 章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

参数 **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。每个条件仅对于某些事件类型有效。表 4-16 列出了各种条件以及这些条件针对哪些事件有效。

表 4-16: 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	自上次执行处理程序以来的时间。
ErrorNumber	不适用	RAISERROR	
IdleTime	秒	ServerIdle	
Interval	秒	All	
LogFreePercent	不适用	LogDiskSpace	
LogFreeSpace	兆字节	LogDiskSpace	
LogSize	兆字节	GrowLog	

条件名称	单位	适用于	注释
RemainingValues	整数	GlobalAutoincrement	剩余值的数目。 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。
TempFreePercent	不适用	TempDiskSpace	
TempFreeSpace	兆字节	TempDiskSpace	
TempSize	兆字节	GrowTemp	

示例

下面的事件定义使用了 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

《参考：语句和选项》中的 [CREATE EVENT 语句](#)

EVENT_CONDITION_NAME 函数 [系统]

作用

可用于列出 EVENT_CONDITION 的可能参数。

语法

```
EVENT_CONDITION_NAME ( integer )
```

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的 [第 3 章“优化查询和删除”](#) 中的 [“导致 SQL Anywhere 进行处理的条件”](#)。

参数

integer 必须大于或等于零。

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。

另请参见

《参考：语句和选项》中的 [CREATE EVENT 语句](#)

EVENT_PARAMETER 函数 [系统]

作用 提供事件处理程序的上下文信息。

语法 **EVENT_PARAMETER** (*context-name*)

```

context-name:
'ConnectionID'
| 'User'
| 'EventName'
| 'Executions'
| 'IQDBMainSpaceName'
| 'NumActive'
| 'TableName'
| condition-name

```

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第3章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

参数 **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，可以从 EVENT_PARAMETER 函数访问 EVENT_CONDITION 函数的任何有效的 *condition-name* 参数。

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。

另请参见 [第 153 页的“EVENT_CONDITION 函数 \[系统 \]”](#)
 《参考：语句和选项》中的 [CREATE EVENT 语句](#)

EXP 函数 [数值]

作用 返回指数函数，即 e 的指定数字次乘方。

语法 **EXP** (*numeric-expression*)

参数 **numeric-expression** 指数。

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]

作用 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*)

window-spec: 请参见下面的“用法”部分。

参数 **expression** 计算其加权值的数值表达式。

period-expression 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 的权重按算数级数减小，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，不能包含架构规范。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

注释 GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

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 4-17: 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。

FIRST_VALUE 函数 [集合]

作用 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 **expression** 要对其确定有序集合中第一个值的表达式。

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。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

注释 不支持 DISTINCT。

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 4-18: FIRST_VALUE 结果集

DepartmentID	EmployeeID	Percentage
500	1,658	100.0000000000000000000000
500	1,570	138.842709713689113761394
500	1,615	110.428462434244870095972
500	1,013	109.585190539292454724330
500	750	137.734409508894510701521
500	921	167.449704854836766654619
500	868	113.239368750752921334778
500	703	222.867927558928643135365
500	191	119.664297474199895594908
400	1,684	100.0000000000000000000000
400	1,740	76.128652163477274215016
400	1,751	76.353400685155687446813
400	1,607	133.758100765890593292456
400	1,507	77.996465120338650199655
400	1,576	150.428767810774836893669

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。

FLOOR 函数 [数值]

作用 返回一个数字的下限（不大于的最大整数）。

语法 **FLOOR** (*numeric-expression*)

参数 **numeric-expression** 数字，通常是浮点数。

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

[第 119 页的“CEILING 函数 \[数值 \]”](#)

GETDATE 函数 [日期和时间]

作用 返回当前的日期和时间。

语法 **GETDATE** ()

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]

作用 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*)

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的 [第 3 章“优化查询和删除”](#) 中的 [“导致 SQL Anywhere 进行处理的条件”](#)。

参数 **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。

如果未向 GRAPHICAL_PLAN 函数提供参数，则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 函数的行为通过数据库选项

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。

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，则会生成带有优化程序估计的查询计划（NOEXEC 选项为 ON 的查询计划），并在 dbisql 客户端计划窗口中显示该查询计划。

当用户请求尚未执行的查询计划时，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但无 QUERY_PLAN_AFTER_RUN 统计信息的查询计划。

无法使用 GRAPHICAL_PLAN 函数访问存储过程的查询计划。

用户可以查看针对 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。若要获取最近打开的游标的查询计划，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 文档中的“在 Interactive SQL 中查看图形计划”，具体位置为：“SQL Anywhere 11.0.1” > “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 Interactive SQL 计划查看器查看计划”。

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。

注释 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` 对包含格式字符（如制表符或回车）的文本非常有用。

```
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** 不受 Adaptive Server Enterprise 支持。

另请参见

第 167 页的“[HTML_PLAN 函数 \[字符串 \]](#)”。

《参考：构件块、表和过程》中的“[OUTPUT 语句 \[DBISQL\]](#)”。

《参考：语句和选项》中的“[NOEXEC 选项](#)”、“[QUERY_PLAN_AFTER_RUN 选项](#)”、“[QUERY_PLAN_AS_HTML 选项](#)”、“[QUERY_PLAN_TEXT_ACCESS 选项](#)”和“[QUERY_PLAN_TEXT_CACHING 选项](#)”。

《SQL Anywhere Server - SQL 参考》中的 [PLAN 函数 \[杂项 \]](#)、[EXPLANATION 函数 \[杂项 \]](#)、[GRAPHICAL_ULPLAN 函数 \[杂项 \]](#)、[LONG_ULPLAN 函数 \[杂项 \]](#) 和 [SHORT_ULPLAN 函数 \[杂项 \]](#)。

GROUPING 函数 [集合]

作用	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。
语法	GROUPING (<i>group-by-expression</i>)
参数	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	<ul style="list-style-type: none">• 1 表示 <i>group-by-expression</i> 由于为小计行的一部分而为空。该列不是该行的前缀列。• 0 表示 <i>group-by-expression</i> 为小计行的前缀列。
标准和兼容性	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 不受 Adaptive Server Enterprise 支持。
另请参见	《参考：语句和选项》中的 SELECT 语句 《系统管理指南，卷 2》中的 第 2 章 “使用 OLAP”

GROUP_MEMBER 函数 [系统]

作用	标识用户是否属于指定的组。
语法	GROUP_MEMBER (<i>group-name-string-expression</i> [, <i>user-name-string-expression</i>])
参数	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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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]

作用	返回十六进制字符串的 BIGINT 等效项。
语法	HEXTOBIGINT (<i>hexadecimal-string</i>)
参数	<p>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：</p> <pre> 0x<i>hex-string</i> 0X<i>hex-string</i> <i>hex-string</i> </pre>
示例	<p>以下语句返回值 4294967287：</p> <pre> SELECT HEXTOBIGINT ('0xffffffff7') FROM iq_dummy SELECT HEXTOBIGINT ('0Xffffffff7') FROM iq_dummy SELECT HEXTOBIGINT ('ffffffff7') FROM iq_dummy </pre>
用法	<p>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。</p> <p>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。</p> <p>对于少于 16 位数的输入，假定将用零对其左侧进行填充。</p> <p>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将返回 NULL。</p> <p>在以下情况下会返回错误：如果 BINARY 或 VARBINARY 值超过 8 个字节，并且 CHAR 或 VARCHAR 值超过 16 个字符，后面附加了 “0x” 的值例外。</p>
标准和兼容性	<ul style="list-style-type: none"> • SQL ISO/ANSI SQL 语法的供应商扩展。 • Sybase 与 Adaptive Server Enterprise 兼容。
另请参见	<p>《参考：语句和选项》中的 CONVERSION_ERROR 选项 [TSQL]</p> <p>第 115 页的 “BIGINTTOHEX 函数 [数据类型转换]”</p> <p>第 164 页的 “HEXTOINT 函数 [数据类型转换]”</p> <p>第 172 页的 “INTTOHEX 函数 [数据类型转换]”</p>

HEXTOINT 函数 [数据类型转换]

作用 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 **hexadecimal-string** 要转换为整数的字符串。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：

```
0xhex-string
0Xhex-string
hex-string
```

示例 以下语句返回值 420：

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

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将返回 **NULL**。

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。当 **ASE_FUNCTION_BEHAVIOR** 选项为 **ON** 时：

- Sybase IQ **HEXTOINT** 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，则会用零填充字符串左侧。
- Sybase IQ **HEXTOINT** 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。
- 假设 Sybase IQ **HEXTOINT** 函数的输出的数据类型为 32 位带符号整数。
- Sybase IQ **HEXTOINT** 接受 32 位十六进制整数作为带符号的表示形式。
- 对于 8 个以上的十六进制字符， Sybase IQ **HEXTOINT** 只考虑相关字符。
- **SQL** ISO/ANSI SQL 语法的供应商扩展。
- **Sybase** 与 **Adaptive Server Enterprise** 兼容。

标准和兼容性

另请参见

《参考：语句和选项》中的 [ASE_FUNCTION_BEHAVIOR 选项](#)

《参考：语句和选项》中的 [CONVERSION_ERROR 选项 \[TSQL\]](#)

第 172 页的“[INTTOHEX 函数 \[数据类型转换\]](#)”

HOUR 函数 [日期和时间]

作用 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)`

参数 `datetime-expression` 日期 / 时间。

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]

作用 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 `datetime-expression` 日期和时间。

integer-expression 要添加到 `datetime-expression` 中的小时数。如果 `integer-expression` 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 `datetime-expression` 显式转换为 `datetime` 数据类型。

有关转换数据类型的信息，请参见第 117 页的“[CAST 函数 \[数据类型转换 \]](#)”。

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。

HTML_DECODE 函数 [HTTP]

作用

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*)

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 4 章“SQL 函数”中的“导致 SQL Anywhere 进行处理的条件”。

HTML_DECODE 函数是一个 SQL Anywhere 函数。请参见 SQL Anywhere 文档中的“HTML_DECODE 函数 [HTTP]”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “使用 SQL” > “SQL 函数” > “SQL 函数 (E-O)”。

HTML_ENCODE 函数 [HTTP]

作用

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*)

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 4 章“SQL 函数”中的“导致 SQL Anywhere 进行处理的条件”。

HTML_ENCODE 函数是一个 SQL Anywhere 函数。请参见 SQL Anywhere 文档中的“HTML_ENCODE 函数 [HTTP]”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “使用 SQL” > “SQL 函数” > “SQL 函数 (E-O)”。

HTML_PLAN 函数 [字符串]

作用 以 HTML 格式字符串的形式返回查询计划。

语法 `HTML_PLAN (string-expression)`

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第4章“SQL 函数”中的“导致 SQL Anywhere 进行处理的条件”。

参数 **string-expression** 生成计划所使用的 SQL 语句。该语句主要为 SELECT 语句，但是也可以为 UPDATE 或 DELETE 语句。

如果未向 HTML_PLAN 函数提供参数，则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，则会显示以下消息：

```
No plan available
```

HTML_PLAN 函数的行为通过数据库选项 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，则将查询计划从高速缓存返回给您。

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。

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 文档中的“使用 Interactive SQL 计划查看器查看计划”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

当针对 SQL Anywhere 查询或 OMNI/CIS 分解查询请求 HTML_PLAN 时，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 对包含格式设置字符（如制表符或回车）的文本非常有用。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** 不受 Adaptive Server Enterprise 支持。

另请参见

[第 160 页的“GRAPHICAL_PLAN 函数 \[字符串\]”](#)。

《参考：构件块、表和过程》中的“[OUTPUT 语句 \[DBISQL\]](#)”。

《参考：语句和选项》中的“[NOEXEC 选项](#)”、“[QUERY_PLAN_AFTER_RUN 选项](#)”、“[QUERY_PLAN_AS_HTML 选项](#)”、“[QUERY_PLAN_TEXT_ACCESS 选项](#)”和“[QUERY_PLAN_TEXT_CACHING 选项](#)”。

《SQL Anywhere Server - SQL 参考》中的 [PLAN 函数 \[杂项\]](#)、[EXPLANATION 函数 \[杂项\]](#)、[GRAPHICAL_ULPLAN 函数 \[杂项\]](#)、[LONG_ULPLAN 函数 \[杂项\]](#) 和 [SHORT_ULPLAN 函数 \[杂项\]](#)。

HTTP_DECODE 函数 [HTTP]

作用 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)`

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第4章“SQL 函数”中的“导致 SQL Anywhere 进行处理的条件”。

HTTP_DECODE 函数是一个 SQL Anywhere 函数。请参见 SQL Anywhere 文档中的“HTTP_DECODE 函数 [HTTP]”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “使用 SQL” > “SQL 函数” > “SQL 函数 (E-O)”。

HTTP_ENCODE 函数 [HTTP]

作用 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)`

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第4章“SQL 函数”中的“导致 SQL Anywhere 进行处理的条件”。

HTTP_ENCODE 函数是一个 SQL Anywhere 函数。请参见 SQL Anywhere 文档中的“HTTP_ENCODE 函数 [HTTP]”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “使用 SQL” > “SQL 函数” > “SQL 函数 (E-O)”。

HTTP_HEADER 函数 [HTTP]

作用 获取 HTTP 标头的值。

语法 `HTTP_HEADER (field-name)`

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第4章“SQL 函数”中的“导致 SQL Anywhere 进行处理的条件”。

HTTP_HEADER 函数是一个 SQL Anywhere 函数。请参见 SQL Anywhere 文档中的“HTTP_HEADER 函数 [HTTP]”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “使用 SQL” > “SQL 函数” > “SQL 函数 (E-O)”。

HTTP_VARIABLE 函数 [HTTP]

作用 获取一个 HTTP 变量的值。

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

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 4 章“SQL 函数”中的“导致 SQL Anywhere 进行处理的条件”。

HTTP_VARIABLE 函数是一个 SQL Anywhere 函数。请参见 SQL Anywhere 文档中的“HTTP_VARIABLE 函数 [HTTP]”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “使用 SQL” > “SQL 函数” > “SQL 函数 (E-O)”。

IFNULL 函数 [杂项]

作用 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 **expression1** 要计算其值的表达式。它的值决定是返回 *expression2* 还是 *expression3*。

expression2 当 *expression1* 为空值时的返回值。

expression3 当 *expression1* 不为空值时的返回值。

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]

作用 返回索引列的名称。

语法 `INDEX_COL (table-name, index-id, key_# [, user-id])`

参数 **table-name** 表名。

index-id *table-name* 的索引的索引 ID。

key_# *index-id* 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，*key_#* 等于 0。如果是多列索引，对于第一列，*key_#* 等于 0，对于第二列，它等于 1，依此类推。

user-id *table-name* 的所有者的用户 ID。如果未指定 *user-id*，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 [第 199 页的“OBJECT_ID 函数 \[系统 \]”](#)

INSERTSTR 函数 [字符串]

作用 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 **numeric-expression** 要在其后插入 *string-expression2* 的位置。使用零可将字符串插入到开头。

string-expression1 要在其中插入 *string-expression2* 的字符串。

string-expression2 要插入的字符串。

注释 INSERTSTR 函数的结果数据类型为 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。

请参见 [“REPLACE 函数 \[字符串 \]”](#)。

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** 不受 Adaptive Server Enterprise 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。

另请参见 [第 245 页的“STUFF 函数 \[字符串 \]”](#)

INTTOHEX 函数 [数据类型转换]

作用 返回十进制整数的十六进制等效数字。

语法 **INTTOHEX** (*integer-expression*)

参数 **integer-expression** 要转换成十六进制的整数。

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，但如果 CONVERSION_ERROR 选项设置为 OFF 则例外。在这种情况下，结果为空。

ASE_FUNCTION_BEHAVIOR 选项 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。

当禁用 ASE_FUNCTION_BEHAVIOR 选项时（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 的输出不具有“0x”或“0X”前缀。
- INTTOHEX 的输入是一个最多为 64 位的整数。

当启用 ASE_FUNCTION_BEHAVIOR 选项时（值为 ON）:

- INTTOHEX 的输出与 ASE 兼容。
- INTTOHEX 的输出始终是 8 位数字，并且用零填充左侧；返回数据类型为 VARCHAR。
- INTTOHEX 的输出不具有“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
```

返回值 3B9ACA00。但是语句:

```
SELECT INTTOHEX( 10000000000 ) 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
- 《参考：语句和选项》中的 [CONVERSION_ERROR 选项 \[TSQL\]](#)
- 《参考：语句和选项》中的 [ASE_FUNCTION_BEHAVIOR 选项](#)
- 第 164 页的“[HEXTOINT 函数 \[数据类型转换\]](#)”

ISDATE 函数 [日期和时间]

作用 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 **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）。

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** 不受 Adaptive Server Enterprise 支持。

ISNULL 函数 [杂项]

作用	返回参数列表中的第一个非空表达式的值。
语法	ISNULL (<i>expression</i> , <i>expression</i> [..., <i>expression</i>])
参数	expression 要测试是否为空值的表达式。 必须向此函数至少传递两个表达式。
示例	下面的语句返回值 -66: <pre>SELECT ISNULL(NULL , -66, 55, 45, NULL, 16) FROM iq_dummy</pre>
用法	ISNULL 函数与 COALESCE 函数相同。
标准和兼容性	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的 Transact-SQL 扩展。• Sybase 不受 Adaptive Server Enterprise 支持。
另请参见	第 122 页的“COALESCE 函数 [杂项]”

ISNUMERIC 函数 [杂项]

作用	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 (<i>string</i>)
<hr/> 注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 4 章“SQL 函数”中的“ 导致 SQL Anywhere 进行处理的条件 ”。 <hr/>	
参数	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）。
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 <code>height_in_cms</code> 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 <code>int</code> 格式返回。 <pre>data height_in_cms ----- asde asde 180</pre>

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** 不受 Adaptive Server Enterprise 支持。

LAG 函数 [分析]

作用

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

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 2^{31} 。

default *offset* 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。可以选择在 **OVER (ORDER BY)** 子句中使用窗口分区子句。**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，您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 的缺省值为 NULL。*default* 的数据类型必须可隐式转换为 *value_expr* 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。
- [第 179 页的“LEAD 函数 \[分析\]”](#)
- 《系统管理指南，卷 2》中的[第 2 章“使用 OLAP”](#)

另请参见

LAST_VALUE 函数 [集合]

作用

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

expression 用于确定有序集合中最后一个值的表达式。

用法

LAST_VALUE 返回一组值中的最后一个值，该值通常是一个有序集合。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。如果指定了 IGNORE NULLS，LAST_VALUE 将返回这组值中的最后一个非空值，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。

不能将 `LAST_VALUE` 或任何其它分析函数用作 `expression`。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。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

注释 不支持 `DISTINCT`。

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 4-19: 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

employee_name	Salary	DepartmentID	highest_paid
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。

LCASE 函数 [字符串]

作用

将字符串中的所有字符转换成小写形式。

语法

LCASE (*string-expression*)

参数

string-expression 要转换成小写形式的字符串。

注释 LCASE 函数的结果数据类型为 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。

有关详细信息，请参见 [“REPLACE 函数 \[字符串 \]”](#)。

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** Adaptive Server Enterprise 不支持 LCASE；使用 LOWER 可以实现相同的功能。

另请参见

[第 184 页的 “LOWER 函数 \[字符串 \]”](#)

[第 262 页的 “UCASE 函数 \[字符串 \]”](#)

[第 263 页的 “UPPER 函数 \[字符串 \]”](#)

LEAD 函数 [分析]

- 作用** 返回表或表分区中的后续行中的属性值的行间函数。
- 语法** `LEAD (value_expr) [, offset [, default]] OVER ([PARTITION BY window partition] ORDER BY window ordering)`
- 参数**
- value_expr** 定义要从表返回的偏移量数据的表列或表达式。
 - offset** 当前行下方的行数，表示为一个准确的非负数字文本，或表示为一个包含准确数值数据的 SQL 变量。允许的范围为 0 到 2^{31} 。
 - default** *offset* 值超过表或分区的范围时要返回的值。
 - window partition** (可选) 一个或多个用逗号分隔的值表达式，表示您希望如何分隔一组结果行。
 - 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**
- LEAD 函数需要 OVER (ORDER_BY) 窗口规范。可以选择在 OVER (ORDER_BY) 子句中使用窗口分区子句。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，您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* 的缺省值为 NULL。*default* 的数据类型必须可隐式转换为 *value_expr* 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

- [第 175 页的“LAG 函数 \[分析 \]”](#)
- 《系统管理指南，卷 2》中的 [第 2 章“使用 OLAP”](#)

LEFT 函数 [字符串]

作用

从字符串开头返回指定数量的字符。

语法

LEFT (*string-expression*, *numeric-expression*)

参数

string-expression 字符串。

numeric-expression 要返回的字符数。

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。

注释 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。

有关详细信息，请参见 [“REPLACE 函数 \[字符串 \]”](#)。

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

- [第 225 页的“RIGHT 函数 \[字符串 \]”](#)
- 《系统管理指南，卷 1》中的 [第 11 章“国际语言和字符集”](#)

LEN 函数 [字符串]

作用 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 **string_expr** 是要求值的字符串表达式。

示例 下面的示例返回值 3152:

```
select len(Photo) from Products
where ID = 500
```

用法 此函数是 CHAR_LENGTH (*string_expression*) 的等效函数。

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 **数据类型** CHAR、NCHAR、VARCHAR、NVARCHAR。

请参见第 3 章“SQL 数据类型”。

作用 第 120 页的“CHAR_LENGTH 函数 [字符串]”和第 242 页的“STR_REPLACE 函数 [字符串]”。

有关字符串函数的常规信息，请参见第 103 页的“字符串函数”。

LENGTH 函数 [字符串]

作用 返回指定字符串中的字符数。

语法 `LENGTH (string-expression)`

参数 **string-expression** 字符串。

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。改用 CHAR_LENGTH 函数。

另请参见 [第 116 页的“BYTE_LENGTH 函数 \[字符串 \]”](#)
[第 120 页的“CHAR_LENGTH 函数 \[字符串 \]”](#)
《系统管理指南，卷 1》中的 [第 11 章“国际语言和字符集”](#)

LN 函数 [数值]

作用 返回指定表达式的自然对数。

语法 **LN** (*numeric-expression*)

参数 **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 函数会生成错误。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。改用 LOG 函数。

另请参见 [第 184 页的“LOG 函数 \[数值 \]”](#)
《系统管理指南，卷 1》中的 [第 11 章“国际语言和字符集”](#)

LOCATE 函数 [字符串]

作用 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 **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。排除的字节数的计算公式为 (-1 * 偏移量) - 1。

对于 CHAR、VARCHAR 和 BINARY 列，*numeric-expression* 是 32 位有符号整数。

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。

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

《Sybase IQ 中的非结构化数据分析》中的第 9 章“函数支持”

LOG 函数 [数值]

作用	返回一个数字的自然对数。 LN 是 LOG 的别名。
语法	LOG (<i>numeric-expression</i>)
参数	numeric-expression 数字。
示例	下面的语句返回值 3.912023: <pre>SELECT LOG(50) FROM iq_dummy</pre>
标准和兼容性	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 与 Adaptive Server Enterprise 兼容。
另请参见	第 184 页的“LOG10 函数 [数值]”

LOG10 函数 [数值]

作用	返回一个数字的以 10 为底的对数。
语法	LOG10 (<i>numeric-expression</i>)
参数	numeric-expression 数字。
示例	下面的语句返回值 1.698970。 <pre>SELECT LOG10(50) FROM iq_dummy</pre>
标准和兼容性	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 与 Adaptive Server Enterprise 兼容。
另请参见	第 184 页的“LOG 函数 [数值]”

LOWER 函数 [字符串]

作用	将字符串中的所有字符转换成小写形式。
语法	LOWER (<i>string-expression</i>)

参数	string-expression 要转换的字符串。
<hr/>	
	注释 LOWER 函数的结果数据类型为 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。 有关详细信息，请参见 “REPLACE 函数 [字符串]” 。
<hr/>	
示例	下面的语句返回值 “lower case”： <pre>SELECT LOWER('LOWER CasE') FROM iq_dummy</pre>
标准和兼容性	<ul style="list-style-type: none"> • SQL 符合 ISO/ANSI SQL 标准。 • Sybase 与 Adaptive Server Enterprise 兼容。
另请参见	第 178 页的 “LCASE 函数 [字符串]” 第 262 页的 “UCASE 函数 [字符串]” 第 263 页的 “UPPER 函数 [字符串]”

LTRIM 函数 [字符串]

作用	删除字符串中的前导空白。
语法	LTRIM (<i>string-expression</i>)
参数	string-expression 要剪裁的字符串。
<hr/>	
	注释 LTRIM 函数的结果数据类型为 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。 有关详细信息，请参见 第 223 页的 “REPLACE 函数 [字符串]” 。
<hr/>	
示例	下面的语句返回删除了所有前导空白的值 “Test Message”： <pre>SELECT LTRIM(' Test Message') FROM iq_dummy</pre>
标准和兼容性	<ul style="list-style-type: none"> • SQL ISO/ANSI SQL 语法的供应商扩展。 • Sybase 与 Adaptive Server Enterprise 兼容。
另请参见	第 229 页的 “RTRIM 函数 [字符串]” 第 249 页的 “TRIM 函数 [字符串]”

MAX 函数 [集合]

作用	返回在每一组行中找到的最大 <i>expression</i> 值。
语法	MAX (<i>expression</i> DISTINCT <i>column-name</i>)
参数	expression 要计算其最大值的表达式。这通常是列名称。 DISTINCT column-name 与 MAX (<i>expression</i>) 返回相同的值，为了完整起见，才将它列了出来。
示例	下面的语句返回值 138948.000，该值表示 Employees 表中的最高工资： <pre>SELECT MAX (Salary) FROM Employees</pre>
用法	忽略 <i>expression</i> 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	<ul style="list-style-type: none">• SQL 符合 ISO/ANSI SQL 标准。• Sybase 与 Adaptive Server Enterprise 兼容。
另请参见	第 187 页的“MIN 函数 [集合]” 《系统管理指南，卷 2》中的 第 2 章“使用 OLAP”

MEDIAN 函数 [集合]

作用	返回表达式的中值。
语法 1	MEDIAN ([ALL DISTINCT] <i>expression</i>)
语法 2	MEDIAN ([ALL DISTINCT] <i>expression</i>) OVER (<i>window-spec</i>)
	<i>window-spec</i> : 请参见下面“用法”部分中“语法 2”的说明。
参数	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。缺省值为 ALL ，将对所有行执行操作。
	<hr/> 注释 使用语法 1 时， GROUP BY 子句不支持 ROLLUP 和 CUBE 。 <hr/>

语法 2 表示在 `SELECT` 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

注释 *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 4-20: 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。

MIN 函数 [集合]

作用

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

expression 要计算其最小值的表达式。这通常是列名称。

DISTINCT column-name 与 `MIN (expression)` 返回相同的值，为了完整起见，才将它列了出来。

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

[第 186 页的“MAX 函数 \[集合\]”](#)

《系统管理指南，卷 2》中的 [第 2 章“使用 OLAP”](#)

MINUTE 函数 [日期和时间]

作用 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

datetime-expression 日期 / 时间值。

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]

作用 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

datetime-expression 日期和时间。

integer-expression 要添加到 *datetime-expression* 中的分钟数。如果 *integer-expression* 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 *datetime-expression* 显式转换为 *datetime* 数据类型。

有关转换数据类型的信息，请参见 [第 117 页的“CAST 函数 \[数据类型转换\]”](#)。

示例

下面的语句返回值 1051125487:

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

下面的语句返回值 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
```

下面的语句返回日期时间值 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。

MOD 函数 [数值]

作用

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

dividend 被除数, 即除法的分子。

divisor 除数, 即除法的分母。

示例

下面的语句返回值 2:

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

用法

被除数 为负数时除法结果为负数或零。*除数* 的符号没有任何影响。

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。% 运算符在 Adaptive Server Enterprise 中作为模运算符使用。

另请参见

[第 222 页的 “REMAINDER 函数 \[数值 \]”](#)

MONTH 函数 [日期和时间]

作用 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 **date-expression** 日期 / 时间值。

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]

作用 返回指定日期表达式中的月份名称。

语法 **MONTHNAME** (*date-expression*)

参数 **date-expression** 日期时间值。

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
```

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

《参考：语句和选项》中的 [DATE_ORDER 选项](#)

MONTHS 函数 [日期和时间]

作用 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 *date-expression* 日期和时间。

integer-expression 要添加到 *date-expression* 中的月数。如果 *integer-expression* 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 *date-expression* 显式转换为 *datetime* 数据类型。

有关转换数据类型的信息，请参见第 117 页的“CAST 函数 [数据类型转换]”部分。

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同一月中。

```
MONTHS( invoice_sent ) = MONTHS( payment_received )
```

比较 MONTH 函数时通常会错误地包括发票发出后的 12 个月内进行的付款。

第二种语法返回从第一个日期到第二个日期之间的月数。数值可能为负。该值是根据两个日期之间一个月的第一天的数目计算出的。会忽略小时、分钟和秒钟。

第三种语法将 *integer-expression* 个月添加到给定日期中。如果新日期超出月末（如 MONTHS('1992-01-31', 1)），则结果设置为该月的最后一天。如果 *integer-expression* 是负数，则从日期中减去相应的月数。会忽略小时、分钟和秒钟。

标准和兼容性

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

NEWID 函数 [杂项]

作用 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。

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。

```
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()
```

会返回唯一标识符 **BINARY(16)**。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。

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** 不受 Adaptive Server Enterprise 支持。

另请参见 [第 244 页的“STRTOUUID 函数 \[字符串 \]”](#)

[第 265 页的“UUIDTOSTR 函数 \[字符串 \]”](#)

[第 71 页的“二进制数据类型”](#)

[第 63 页的“字符数据类型”](#)

NEXT_CONNECTION 函数 [系统]

作用 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*})

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第3章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

返回值 INT

参数

connection-id 一个整数，通常从上一个 NEXT_CONNECTION 调用返回。如果 *connection-id* 为 NULL，则 NEXT_CONNECTION 返回最近连接 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 *database-id*，则使用当前数据库。如果提供 NULL，则 NEXT_CONNECTION 返回下一连接，而不考虑数据库。

用法 NEXT_CONNECTION 可用于枚举数据库的连接。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。

标准和兼容性

- **SQL** ISO/ANSI SQL 语法的供应商扩展。

备注 NEXT_CONNECTION 可用于枚举与数据库的连接。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 NULL 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。但是，因为 NEXT_CONNECTION 按相反顺序返回连接 ID，所以它不会返回在启动函数后创建的连接。如果要确保断开所有连接，可阻止在运行 NEXT_CONNECTION 之前创建新连接。

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]

作用 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* })

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 3 章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

参数 **database-id** 一个整数，它指定数据库的 ID 号。

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 **NEXT_DATABASE** 可用于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 [第 199 页的“OBJECT_ID 函数 \[系统 \]”](#)

NEXT_HTTP_HEADER 函数 [HTTP]

作用 获得下一个 HTTP 标头名称。

语法 **NEXT_HTTP_HEADER** (*header-name*)

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第3章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

NEXT_HTTP_HEADER 函数是一个 SQL Anywhere 函数。请参见 SQL Anywhere 文档中的“NEXT_HTTP_HEADER 函数 [HTTP]”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “使用 SQL” > “SQL 函数” > “SQL 函数 (E-O)”。

NEXT_HTTP_VARIABLE 函数 [HTTP]

作用 获得下一个 HTTP 变量的名称。

语法 **NEXT_HTTP_VARIABLE** (*var-name*)

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第3章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

NEXT_HTTP_VARIABLE 函数是一个 SQL Anywhere 函数。请参见 SQL Anywhere 文档中的“NEXT_HTTP_VARIABLE 函数 [HTTP]”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “使用 SQL” > “SQL 函数” > “SQL 函数 (E-O)”。

NOW 函数 [日期和时间]

作用 返回当前的日期和时间。这是 CURRENT_TIMESTAMP 的传统语法。

语法 **NOW** (*)

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]

作用 将查询结果分发到指定数量的“表元”中，并将表元号分派到表元中的每一行。

语法 `NTILE (expression1)
OVER (ORDER BY expression2 [ASC | DESC])`

参数 **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

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

- 用法** NTILE 为一个秩分析函数，可将查询结果分配到指定数量的表元，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 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 子句仅在 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。升序顺序是缺省值。
- NTILE 只能在 SELECT 或 INSERT 语句的选择列表中或者在 SELECT 语句的 ORDER BY 子句中使用。NTILE 可以在视图中或联合中使用。您不能在子查询中、HAVING 子句中，或者 UPDATE 或 DELETE 语句的选择列表中使用 NTILE 函数。每个查询仅允许使用一个 NTILE 函数。
- 标准和兼容性**
- **SQL** ISO/ANSI SQL 语法的供应商扩展。
 - **Sybase** 不受 Adaptive Server Enterprise 或 SQL Anywhere 支持。
- 另请参见** [第 92 页的“分析函数”](#)
《系统管理指南，卷 2》中的 [第 2 章“使用 OLAP”](#)

NULLIF 函数 [杂项]

- 作用** 通过比较表达式提供缩写的 CASE 表达式。
- 语法** `NULLIF (expression1, expression2)`
- 参数** **expression1** 要比较的表达式。
expression2 要比较的表达式。
- 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。
如果第一个表达式的值与第二个表达式的值相等，则 NULLIF 返回 NULL。
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 函数提供了编写某些 CASE 表达式的简便方法。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** [第 26 页的“CASE 表达式”](#)

NUMBER 函数 [杂项]

- 作用** 为查询结果中的各个连续行生成从 1 开始的编号。
- 语法** **NUMBER (*)**
- 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
```

- 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`。

注释 如果在以下位置使用 `NUMBER`，则会生成语法错误：`DELETE` 语句、`WHERE` 子句、`HAVING` 子句、`ORDER BY` 子句、子查询、涉及集合的查询、任意约束、`GROUP BY`、`DISTINCT`、包含 `UNION ALL` 的查询，或派生表。

标准和兼容性

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

OBJECT_ID 函数 [系统]

作用 返回对象 ID。

语法 `OBJECT_ID (object-name)`

参数 **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

[第 122 页的“COL_NAME 函数 \[系统 \]”](#)

[第 148 页的“DB_ID 函数 \[系统 \]”](#)

[第 199 页的“OBJECT_NAME 函数 \[系统 \]”](#)

OBJECT_NAME 函数 [系统]

作用 返回对象名称。

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

参数 **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
- 第 122 页的 “COL_NAME 函数 [系统]”
 - 第 148 页的 “DB_NAME 函数 [系统]”
 - 第 199 页的 “OBJECT_ID 函数 [系统]”

OCTET_LENGTH 函数 [字符串]

- 作用 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 **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。
- 标准和兼容性 **Sybase** 不受 SQL Anywhere 或 Adaptive Server Enterprise 支持。
- 另请参见 第 116 页的 “BIT_LENGTH 函数 [字符串]”
- 《Sybase IQ 中的非结构化数据分析》中的第 9 章 “函数支持”

PATINDEX 函数 [字符串]

- 作用 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 **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，那么，如果模式出现在字符串起始处，PATINDEX 返回一 (1)；否则返回零。如果 *pattern* 以百分比通配符开头，则将两个前导百分比通配符视为一个。
- 模式使用的通配符与 LIKE 比较相同。表 4-21 列出了模式通配符。

表 4-21: 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

不带通配符（百分比 % 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。

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。

如果未找到模式，**PATINDEX** 将返回零 (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 位无符号的整数位置。

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

[第 37 页的“LIKE 条件”](#)

[第 182 页的“LOCATE 函数 \[字符串 \]”](#)

《Sybase IQ 中的非结构化数据分析》中的 [第 9 章“函数支持”](#)

PERCENT_RANK 函数 [分析]

作用

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

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。

示例

下面的语句说明了 **PERCENT_RANK** 函数的用法:

```
SELECT s_suppkey, SUM(s_acctBal) AS sum_acctBal,
PERCENT_RANK() OVER ( ORDER BY SUM(s_acctBal) DESC )
```

```
AS percent_rank_all FROM supplier GROUP BY s_suppkey;
```

s_suppkey	sum_acctBal	percent_rank_all
supplier#011	200000	0
supplier#002	200000	0
supplier#013	123000	0.3333
supplier#004	110000	0.5
supplier#035	110000	0.5
supplier#006	50000	0.8333
supplier#021	10000	1

用法

PERCENT_RANK 是秩分析函数。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

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

其中 R_x 是组中行的秩位置， $N_{totalRow}$ 是 OVER 子句指定的组中的总行数。

PERCENT_RANK 需要 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 子句仅在 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。升序顺序是缺省值。

PERCENT_RANK 只能在 SELECT 或 INSERT 语句的选择列表中或者在 SELECT 语句的 ORDER BY 子句中使用。PERCENT_RANK 可以在视图或联合中使用。您不能在子查询中、HAVING 子句中，或者 UPDATE 或 DELETE 语句的选择列表中使用 PERCENT_RANK 函数。每个查询仅允许使用一个 rank 分析函数。

标准和兼容性

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

另请参见

[第 92 页的“分析函数”](#)

《系统管理指南，卷 2》中的 [第 2 章“使用 OLAP”](#)

PERCENTILE_CONT 函数 [分析]

作用 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。

注释 如果仅需要计算百分点，可改用 NTILE 函数，将值设为 100。

语法

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

参数

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 个百分点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% 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 ORDER BY 项的数据类型相同。PERCENTILE_CONT 的 ORDER BY 表达式必须属于数字数据类型。

PERCENTILE_CONT 需要 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, 此表达式必须属于数字数据类型。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, 将报告错误。

ASC 或 DESC 参数用于指定升序或降序排序序列。升序顺序是缺省值。

子查询、HAVING 子句、视图或联合中允许使用 PERCENTILE_CONT 函数。PERCENTILE_CONT 可在使用简单非分析集合函数的任意位置使用。PERCENTILE_CONT 函数忽略数据集中的空值。

标准和兼容性

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

另请参见

[第 92 页的“分析函数”](#)

[第 196 页的“NTILE 函数 \[分析 \]”](#)

[第 205 页的“PERCENTILE_DISC 函数 \[分析 \]”](#)

《系统管理指南, 卷 2》中的第 2 章 “使用 OLAP”

PERCENTILE_DISC 函数 [分析]

作用 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。

注释 如果仅需要计算百分点，可改用 NTILE 函数，将值设为 100。

语法

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

参数

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% 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对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 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 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, 将报告错误。

ASC 或 DESC 参数用于指定升序或降序排序序列。升序顺序是缺省值。

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 NULL 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 或 SQL Anywhere 支持。

另请参见

[第 92 页的“分析函数”](#)

[第 196 页的“NTILE 函数 \[分析 \]”](#)

[第 203 页的“PERCENTILE_CONT 函数 \[分析 \]”](#)

《系统管理指南, 卷 2》中的第 2 章 “使用 OLAP”

PI 函数 [数值]

作用 返回数值 PI。

语法 **PI (*)**

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 支持 PI() 函数，但不支持 PI(*)。

POWER 函数 [数值]

作用 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 **numeric-expression1** 底数。

numeric-expression2 指数。

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-expression2*。

标准和兼容性

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

PROPERTY 函数 [系统]

作用 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 })**

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第 3 章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

参数 **property-id** 一个整数，该整数表示服务器级属性的属性号。此数值可通过 PROPERTY_NUMBER 函数确定。当在一组属性间循环时，常使用 *property-id*。

property-name 给出属性名称的字符串。有关服务器属性名称的列表，请参见第 108 页的“可用于服务器的属性”。

示例	下面的语句返回当前数据库服务器的名称： <pre>SELECT PROPERTY('Name') FROM iq_dummy</pre>
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	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 不受 Adaptive Server Enterprise 支持。
另请参见	第 108 页的“可用于服务器的属性”

PROPERTY_DESCRIPTION 函数 [系统]

作用	返回有关属性的说明。
语法	<pre>PROPERTY_DESCRIPTION ({ <i>property-id</i> <i>property-name</i> })</pre>
<hr/> 注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的 第 3 章“优化查询和删除” 中的 “导致 SQL Anywhere 进行处理的条件” 。 <hr/>	
参数	<p>property-id 一个整数，该整数表示属性的属性号。此数值可通过 PROPERTY_NUMBER 函数确定。当在一组属性间循环时，常使用 <i>property-id</i>。</p> <p>property-name 给出属性名称的字符串。有关属性名称，请参见 第 107 页的“连接属性”、第 108 页的“可用于服务器的属性” 和 第 108 页的“可用于每个数据库的属性” 中的列表。</p>
示例	下面的语句返回说明“索引插入数”： <pre>SELECT PROPERTY_DESCRIPTION('IndAdd') FROM iq_dummy</pre>
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	<ul style="list-style-type: none">• SQL ISO/ANSI SQL 语法的供应商扩展。• Sybase 不受 Adaptive Server Enterprise 支持。
另请参见	第 107 页的“连接属性” 第 108 页的“可用于服务器的属性” 第 108 页的“可用于每个数据库的属性”

PROPERTY_NAME 函数 [系统]

作用 返回具有所提供属性号的属性的名称。

语法 `PROPERTY_NAME (property-id)`

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第3章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

参数 `property-id` 属性的属性号。

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

第 107 页的“连接属性”

第 108 页的“可用于服务器的属性”

第 108 页的“可用于每个数据库的属性”

PROPERTY_NUMBER 函数 [系统]

作用 返回具有所提供属性名称的属性的属性号。

语法 `PROPERTY_NUMBER (property-name)`

注释 需要考虑 CIS 函数补偿性能方面的一些事宜。请参见《性能和调优指南》的第3章“优化查询和删除”中的“导致 SQL Anywhere 进行处理的条件”。

参数 `property-name` 属性名称。有关属性名称，请参见第 107 页的“连接属性”、第 108 页的“可用于服务器的属性”和第 108 页的“可用于每个数据库的属性”中的列表。

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
- 第 107 页的“连接属性”
 - 第 108 页的“可用于服务器的属性”
 - 第 108 页的“可用于每个数据库的属性”

QUARTER 函数 [日期和时间]

作用 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 `date-expression` 日期。

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
```

用法 表 4-22 列出一年的各个季度中的日期。

表 4-22: 年的季度值

季度	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。

另请参见 《参考: 语句和选项》中的 [DATE_ORDER 选项](#)。

RADIANS 函数 [数值]

作用 将数字由角度转换成弧度。

语法 `RADIANS (numeric-expression)`

参数 `numeric-expression` 数字 (以角度为单位)。该角度将转换成弧度。

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** 与 Adaptive Server Enterprise 兼容。

RAND 函数 [数值]

作用 返回一个 DOUBLE 精度的随机数字 x，其中 $0 \leq x < 1$ ，使用可选种子。

语法 `RAND ([integer-expression])`

参数 **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。

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。查询优化程序不会高速缓存 RAND 函数的结果。

注释 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。

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_tvcs) 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
```

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]

作用 排列组中的项目。

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

参数 **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。

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 是秩分析函数。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。存在并列排名时 **RANK** 会留空位。

RANK 需要 **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** 子句仅在 **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**。

可以选择在 **OVER (ORDER BY)** 子句中使用 **PARTITION BY** 窗口分区子句。请参见《系统管理指南，卷 2》的 [第 2 章 “使用 OLAP”](#) 中的 [“窗口分区”](#)。

ASC 或 **DESC** 参数用于指定升序或降序排序序列。升序顺序是缺省值。

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在 **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** 不受 Adaptive Server Enterprise 或 SQL Anywhere 支持。

另请参见

[第 92 页的 “分析函数”](#)

[第 150 页的 “DENSE_RANK 函数 \[分析 \]”](#)

[《系统管理指南，卷 2》中的第 2 章 “使用 OLAP”](#)

REGR_AVGX 函数 [集合]

作用 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*)

window-spec: 请参见下面“用法”部分中“语法 2”的说明。

参数 **dependent-expression** 受独立变量影响的变量。

independent-expression 影响结果的变量。

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 NULL。

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 文档中的“集合函数的数学公式”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

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。

REGR_AVGY 函数 [集合]

作用 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*)

window-spec: 请参见下面 “用法” 部分中 “语法 2” 的说明。

参数 **dependent-expression** 受独立变量影响的变量。

independent-expression 影响结果的变量。

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 NULL。

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 文档中的 “集合函数的数学公式”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的 “分析函数”。

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]

作用 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)`

window-spec: 请参见下面“用法”部分中“语法 2”的说明。

参数 **dependent-expression** 受独立变量影响的变量。

independent-expression 影响结果的变量。

用法 此函数返回一个 UNSIGNED BIGINT 值作为结果。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

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。以下函数返回值 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。

REGR_INTERCEPT 函数 [集合]

作用 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)`

window-spec: 请参见下面“用法”部分中“语法 2”的说明。

参数	dependent-expression 受独立变量影响的变量。 independent-expression 影响结果的变量。
用法	<p>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 NULL。</p> <p>在消除所有 <i>dependent-expression</i> 或 <i>independent-expression</i> 为 NULL 的值对之后，会将此函数应用于 <i>dependent-expression</i> 和 <i>independent-expression</i> 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 <i>dependent-expression</i>，x 表示 <i>independent-expression</i>：</p> $\text{AVG}(y) - \text{REGR_SLOPE}(y, x) * \text{AVG}(x)$ <p>请参见 SQL Anywhere 文档中的“集合函数的数学公式”，具体位置为：“SQL Anywhere 11.0.1” > “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”。</p>
	<hr/> 注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。 <hr/>
	<p>语法 2 表示在 SELECT 语句中作为窗口函数的用法。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 <i>window-spec</i> 的元素。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。</p>
示例	<p>下面的示例返回值 1874.5805688517603：</p> <pre>SELECT REGR_INTERCEPT(Salary, (YEAR(NOW()) - YEAR(BirthDate))) FROM Employees;</pre>
标准和兼容性	<ul style="list-style-type: none">• SQL 符合 ISO/ANSI SQL 标准。SQL/OLAP 功能 T612。• Sybase 与 SQL Anywhere 兼容。

REGR_R2 函数 [集合]

作用 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 `REGR_R2(dependent-expression, independent-expression)`

语法 2 `REGR_R2(dependent-expression, independent-expression)
OVER (window-spec)`

window-spec: 请参见下面“用法”部分中“语法 2”的说明。

参数 **dependent-expression** 受独立变量影响的变量。

independent-expression 影响结果的变量。

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 NULL。

在消除所有 *dependent-expression* 或 *independent-expression* 为 NULL 的值对之后，会将 REGR_R2 应用于 *dependent-expression* 和 *independent-expression* 对的集合。Sybase IQ 随后会应用于下面的算法：

- 如果 $VAR_POP(x) = 0$ ，REGR_R2 将计算 $VAR_POP(x)$ 并返回空值；如果 $VAR_POP(y) = 0$ ，则会计算 $VAR_POP(y)$ 并返回值 1。
- 如果 $VAR_POP(x)$ 和 $VAR_POP(y)$ 都不是 0，返回值为 $POWER(CORR(y,x),2)$

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

请参见 SQL Anywhere 文档中的“集合函数的数学公式”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

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

REGR_SLOPE 函数 [集合]

作用 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*)

window-spec: 请参见下面 “用法” 部分中 “语法 2” 的说明。

参数 **dependent-expression** 受独立变量影响的变量。

independent-expression 影响结果的变量。

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 NULL 的值对之后，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，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 文档中的 “集合函数的数学公式”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的 “分析函数”。

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

REGR_SXX 函数 [集合]

作用 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)`

window-spec: 请参见下面“用法”部分中“语法 2”的说明。

参数 **dependent-expression** 受独立变量影响的变量。

independent-expression 影响结果的变量。

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 NULL。

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*，*x* 表示 *independent-expression*：

REGR_COUNT(y, x) * VAR_POP(x)

请参见 SQL Anywhere 文档中的“集合函数的数学公式”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

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

REGR_SXY 函数 [集合]

作用 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)`

window-spec: 请参见下面 “用法” 部分中 “语法 2” 的说明。

参数 **dependent-expression** 受独立变量影响的变量。

independent-expression 影响结果的变量。

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 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*，*x* 表示 *independent-expression*：

REGR_COUNT(x, y) * COVAR_POP(x, y)

请参见 SQL Anywhere 文档中的 “集合函数的数学公式”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的 “分析函数”。

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。

REGR_SYY 函数 [集合]

作用 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)`

window-spec: 请参见下面“用法”部分中“语法 2”的说明。

参数 **dependent-expression** 受独立变量影响的变量。

independent-expression 影响结果的变量。

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 NULL。

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*，*x* 表示 *independent-expression*：

REGR_COUNT(x, y) * VAR_POP(y)

请参见 SQL Anywhere 文档中的“集合函数的数学公式”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

注释 使用语法 1 时，GROUP BY 子句不支持 ROLLUP 和 CUBE。不支持 DISTINCT。

语法 2 表示在 SELECT 语句中作为窗口函数的用法。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。有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

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。

REMAINDER 函数 [数值]

作用 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 **dividend** 被除数，即除法的分子。

divisor 除数，即除法的分母。

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 [第 189 页的“MOD 函数 \[数值 \]”](#)

REPEAT 函数 [字符串]

作用 以指定的次数连接字符串。

语法 **REPEAT** (*string-expression*, *integer-expression*)

参数 **string-expression** 要重复的字符串。

integer-expression 字符串的重复次数。如果 *integer-expression* 是负数，则返回空字符串。

注释 REPEAT 函数的结果数据类型为 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项许可证，或使用 CAST 并将 REPEAT 设置为正确的数据类型和大小。

有关详细信息，请参见 [“REPLACE 函数 \[字符串 \]”](#)。

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** Adaptive Server Enterprise 不支持该函数，但 REPLICATE 可提供相同的功能。

另请参见 [第 224 页的“REPLICATE 函数 \[字符串 \]”](#)

REPLACE 函数 [字符串]

作用 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。

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 *replace-string* 替换的字符串。该字符串限制为 255 个字节。如果 *search-string* 是空字符串，则按原样返回原始字符串。

replace-string 替换字符串，该字符串用于替换 *search-string*。该字符串可为任意长度。如果 *replace-string* 是空字符串，则删除出现的所有 *search-string*。

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;
```

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	<p>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。</p> <p>有两种方法可以解决这一问题：</p> <ul style="list-style-type: none"> • 声明一个本地临时表，然后执行 INSERT： <pre>DECLARE local temporary table #mytable (name_column char(10)) on commit preserve rows; INSERT INTO #mytable SELECT REPLACE(name,'0','1') FROM dummy_table01;</pre> • 使用 CAST： <pre>SELECT CAST(replace(name, '0', '1') AS Char(10)) into #mytable from dummy_table01;</pre>
标准和兼容性	<ul style="list-style-type: none"> • SQL ISO/ANSI SQL 语法的供应商扩展。 • Sybase 与 Adaptive Server Enterprise 兼容。
另请参见	第 246 页的“SUBSTRING 函数 [字符串]”

REPLICATE 函数 [字符串]

作用	以指定的次数连接字符串。
语法	REPLICATE (<i>string-expression</i> , <i>integer-expression</i>)
参数	<p>string-expression 要重复的字符串。</p> <p>integer-expression 字符串的重复次数。</p>
示例	<p>下面的语句返回值 “repeatrepeatrepeat”：</p> <pre>SELECT REPLICATE('repeat', 3) FROM iq_dummy</pre>
用法	REPLICATE 与 REPEAT 函数相同。
	<hr/> <p>注释 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。</p> <p>有关详细信息，请参见第 223 页的“REPLACE 函数 [字符串]”。</p> <hr/>

标准和兼容性	<ul style="list-style-type: none"> • SQL ISO/ANSI SQL 语法的供应商扩展。 • Sybase 与 Adaptive Server Enterprise 兼容。
另请参见	第 222 页的“REPEAT 函数 [字符串]”

REVERSE 函数 [字符串]

作用	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 (<i>expression</i> <i>uchar_expr</i>)
参数	expression 是字符或二进制类型列名、变量或常量表达式（属于 CHAR 、 VARCHAR 、 NCHAR 、 NVARCHAR 、 BINARY 或 VARBINARY 类型）。
示例 1	<pre>select reverse("abcd") ----- dcba</pre>
示例 2	<pre>select reverse(0x12345000) ----- 0x00503412</pre>
用法	<ul style="list-style-type: none"> • REVERSE，一个字符串函数，它返回表达式的逆序形式。 • 如果表达式为空，则逆序返回空。 • 代理对被视为不可拆分的，因此不能逆序。
权限	任何用户都可以执行 REVERSE 。
标准和兼容性	<ul style="list-style-type: none"> • SQL ISO/ANSI SQL 语法的 Transact-SQL 扩展。
另请参见	<p>作用 第 184 页的“LOWER 函数 [字符串]” 和 第 263 页的“UPPER 函数 [字符串]”。</p> <p>有关字符串函数的常规信息，请参见 第 103 页的“字符串函数”。</p>

RIGHT 函数 [字符串]

作用	返回字符串中最右边的字符。
语法	RIGHT (<i>string-expression</i> , <i>numeric-expression</i>)
参数	<p>string-expression 要从左边截断的字符串。</p> <p>numeric-expression 要返回的从字符串结尾处开始的字符数。</p>
示例	<p>下面的语句返回值 “olate”：</p> <pre>SELECT RIGHT('chocolate', 5) FROM iq_dummy</pre>

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。

注释 RIGHT 函数的结果数据类型为 LONG VARCHAR。如果在 SELECT INTO 语句中使用 RIGHT，您必须具有非结构化数据分析选项许可证，或使用 CAST 并将 RIGHT 设置为正确的数据类型和大小。

有关详细信息，请参见第 223 页的“REPLACE 函数 [字符串]”。

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 第 180 页的“LEFT 函数 [字符串]”
《系统管理指南，卷 1》中的第 11 章“国际语言和字符集”

ROUND 函数 [数值]

作用 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 **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。

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)
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 第 250 页的“TRUNCNUM 函数 [数值]”

ROW_NUMBER 函数 [分析]

作用 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 **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。可以选择在 OVER (ORDER_BY) 子句中使用窗口分区子句。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。

另请参见

- 《系统管理指南，卷 2》中的第 2 章 “使用 OLAP”

ROWID 函数 [杂项]

作用 返回表的每一行的内部行 ID 值。

语法 **ROWID** (*table-name*) ...**FROM** *table-name*

参数 **table-name** 表的名称。指定括号内不带引号或带双引号的表名。

示例 下面的语句返回行 ID 值 1 到 10:

```
SELECT ROWID ( "PRODUCTS" ) FROM PRODUCTS

rowid(Products)
              1
              2
              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。

您必须指定 **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]

作用 返回尾随空白已删除的字符串。

语法 **RTRIM** (*string-expression*)

参数 **string-expression** 要剪裁的字符串。

注释 RTRIM 函数的结果数据类型为 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。

有关详细信息，请参见第 223 页的“REPLACE 函数 [字符串]”。

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

第 185 页的“LTRIM 函数 [字符串]”

SECOND 函数 [日期和时间]

作用 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 **datetime-expression** 日期 / 时间值。

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。

SECONDS 函数 [日期和时间]

作用 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 **datetime-expression** 日期和时间。
integer-expression 要添加到 *datetime-expression* 中的秒数。如果 *integer-expression* 是负数，则从日期 / 时间值中减去相应的秒数。如果提供整数表达式，则必须将 *datetime-expression* 显式转换为 `datetime` 数据类型。

有关转换数据类型的信息，请参见第 117 页的“[CAST 函数 \[数据类型转换 \]](#)”。

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]

- 作用** 返回一个数字的符号。
- 语法** **SIGN** (*numeric-expression*)
- 参数** **numeric-expression** 要返回其符号的数字。
- 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 ]

- 作用** 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** **string-expression1** 要比较的第一个字符串。  
**string-expression2** 要比较的第二个字符串。
- 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]

作用 返回一个数字的正弦值，以弧度表示。

语法 **SIN** (*numeric-expression*)

参数 **numeric-expression** 角度（以弧度为单位）。

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

- 第 113 页的 “[ASIN 函数 \[数值 \]](#)”
- 第 128 页的 “[COS 函数 \[数值 \]](#)”
- 第 128 页的 “[COT 函数 \[数值 \]](#)”
- 第 249 页的 “[TAN 函数 \[数值 \]](#)”

SORTKEY 函数 [字符串]

作用 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 **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，**SORTKEY** 函数将返回空值。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。

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 -1`。
- 下表列出了 Adaptive Server Enterprise 归类。

说明	归类名	归类 ID
缺省 Unicode 多语言	default	0
CP 850 方案：没有变音	altnoacc	39
CP 850 方案：小写优先	altdict	45
CP 850 西欧：没有大小写优先级	altnocsp	46
CP 850 斯堪的纳维亚语字典	scandict	47
CP 850 斯堪的纳维亚文：没有大小写优先级	scannocp	48
GB 拼音	gbpinyin	不适用
二进制排序	binary	50
拉丁语 -1 英语、法语、德语字典	dict	51
拉丁语 -1 英语、法语、德语，不区分大小写	nocase	52
拉丁语 -1 英语、法语、德语，不区分大小写，优先	nocasep	53
拉丁语 -1 英语、法语、德语，无变音	noaccent	54
Latin-1 西班牙语字典	espdict	55
拉丁语 -1 西班牙语，不区分大小写	espnoacs	56
拉丁语 -1 西班牙语，无变音	espnoac	57
ISO 8859-5 俄语字典	rusdict	58
ISO 8859-5 俄语，不区分大小写	rusnoacs	59
ISO 8859-5 西里尔语字典	cyrdict	63
ISO 8859-5 西里尔语，不区分大小写	cyrnoacs	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，无变音	hunnoacs	70
ISO 8859-2 匈牙利语，不区分大小写	hunnoacs	71
ISO 8859-5 土耳其语字典	turdict	72
ISO 8859-5 土耳其语，无变音	turnoacs	73
ISO 8859-5 土耳其语，不区分大小写	turnoacs	74
CP 874 (TIS 620) 皇家泰语字典	thaidict	1
ISO 14651 排序标准	14651	22

说明	归类名	归类 ID
Shift-JIS 二进制顺序	sjisbin	179
Unicode UTF-8 二进制排序	utf8bin	24
EUC JIS 二进制顺序	ucjisbn	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

collation-tailoring-string（可选）指定归类定制选项 (*collation-tailoring-string*) 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》第 1 章“SQL 语句”的“CREATE DATABASE 语句”中的“*collation-tailoring-string*”。

注释 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。

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。

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 )
```

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' )
```

有关对 SQL Anywhere 数据库使用归类定制和 **SORTKEY** 函数的信息，请参见《SQL Anywhere Server - SQL 参考》中的“SQL 函数”。

注释 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 及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問題，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

另请参见

- **SQL** ISO/ANSI SQL 语法的供应商扩展。

第 2 章“数据库选项”中的“**SORT_COLLATION** 选项”

第 103 页的“字符串函数”

《系统管理指南，卷 1》中的第 11 章“国际语言和字符集”

SOUNDEX 函数 [字符串]

作用 返回表示字符串声音的数字。

语法 **SOUNDEX** (*string-expression*)

参数 **string-expression** 字符串。

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 函数忽略多字节字符。

尽管不很完善，但 SOUNDEX 对于那些听起来相似和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

SPACE 函数 [字符串]

作用 返回指定数目的空格。

语法 **SPACE** (*integer-expression*)

参数 **integer-expression** 要返回的空格数。

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]

作用 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 **sql-standard-string** 测试遵从性时依据的标准级别。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-statement-string 用于检查一致性的 SQL 语句。

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' );
```

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

第 413 页的 “[sa_ansi_standard_packages 过程](#)”。

第 417 页的 “[SQL_FLAGGER_ERROR_LEVEL 选项 \[TSQL\]](#)”。

《实用程序指南》中的第 5 章 “[SQL 预处理器](#)”。

SQL Anywhere 文档中的 “使用 SQL 标志程序测试 SQL 一致性”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 用法” > “SQL 术语和兼容性” > “SQL 术语”。

SQRT 函数 [数值]

- 作用 返回一个数字的平方根。
- 语法 **SQRT** (*numeric-expression*)
- 参数 **numeric-expression** 要计算其平方根的数字。
- 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 ]

- 作用 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 **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 函数会生成错误。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 ]

- 作用 返回一组数字的标准偏差。
- 语法 **STDDEV** ( [ ALL ] *expression* )
- 参数 **expression** 任何数值数据类型（FLOAT、REAL 或 DOUBLE 精度）的表达式。
- 示例 假定以下数据:

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

```
Salary
51432.000
57090.000
42300.000
```

| Salary     |
|------------|
| 43700.00   |
| 36500.000  |
| 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'
```

| Name      | 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:

$$stddev = \sqrt{\text{方差}}$$

STDDEV 返回数据类型 DOUBLE 精度浮点的结果。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

第 241 页的“[STDDEV\\_SAMP 函数 \[ 集合 \]](#)”

第 267 页的“[VARIANCE 函数 \[ 集合 \]](#)”

《系统管理指南，卷 2》中的第 2 章“[使用 OLAP](#)”

## STDDEV\_POP 函数 [ 集合 ]

- 作用** 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** **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）。
- 示例** 下面的语句列出不同时间段每个订单的项目数的平均值和方差：

```
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;
```

| 年份   | 季度  | 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** [第 92 页的“分析函数”](#)

《系统管理指南，卷 2》中的 [第 2 章“使用 OLAP”](#)

## STDDEV\_SAMP 函数 [ 集合 ]

**作用** 计算由数值表达式组成的样本的标准偏差，类型为 DOUBLE。

**注释** STDDEV\_SAMP 是 STDDEV 的别名。

**语法** **STDDEV\_SAMP** ( [ ALL ] *expression* )

**参数** **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）。

**示例** 下面的语句列出不同时间段每个订单的项目数的平均值和方差：

```
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;
```

| 年份   | 季度  | 平均值       | 方差      |
|------|-----|-----------|---------|
| 2000 | 1   | 25.775148 | 14.3218 |
| 2000 | 2   | 27.050847 | 15.0696 |
| ...  | ... | ...       | ...     |

**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**

[第 92 页的“分析函数”](#)

[第 238 页的“STDDEV 函数 \[ 集合 \]”](#)

《系统管理指南，卷 2》中的 [第 2 章“使用 OLAP”](#)

## STR 函数 [ 字符串 ]

|        |                                                                                                                                                                                                    |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作用     | 返回一个数字的等效字符串。                                                                                                                                                                                      |
| 语法     | <b>STR</b> ( <i>numeric-expression</i> [ , <i>length</i> [ , <i>decimal</i> ] ] )                                                                                                                  |
| 参数     | <p><b>numeric-expression</b> 任何近似数值类型（FLOAT、REAL 或 DOUBLE 精度）的表达式。</p> <p><b>length</b> 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。</p> <p><b>decimal</b> 要返回的小数点右边的位数。缺省值为 0。</p> |
| 示例     | <p>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：</p> <pre>SELECT STR( 1234.56 ) FROM iq_dummy</pre> <p>下面的语句返回结果 1234.5：</p> <pre>SELECT STR( 1234.56, 6, 1 ) FROM iq_dummy</pre>                        |
| 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：                                                                                                                                                              |
|        | <pre>SELECT STR( 1234.56, 3 ) FROM iq_dummy</pre>                                                                                                                                                  |
| 标准和兼容性 | <ul style="list-style-type: none"> <li>• <b>SQL</b> ISO/ANSI SQL 语法的供应商扩展。</li> <li>• <b>Sybase</b> 与 Adaptive Server Enterprise 兼容。</li> </ul>                                                    |

## STR\_REPLACE 函数 [ 字符串 ]

|    |                                                                                                                                                                                                                                                                                                                                                            |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作用 | 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 ( <i>string_expr1</i> ) 中出现的第二个字符串表达式 ( <i>string_expr2</i> ) 的所有实例替换为第三个表达式 ( <i>string_expr3</i> )。                                                                                                                                                                                                           |
|    | STR_REPLACE 是 REPLACE 函数的别名                                                                                                                                                                                                                                                                                                                                |
| 语法 | <b>REPLACE</b> ( <i>string_expr1</i> , <i>string_expr2</i> , <i>string_expr3</i> )                                                                                                                                                                                                                                                                         |
| 参数 | <p><b>string_expr1</b> 是源字符串或要搜索的字符串表达式，表示形式为：CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY 或 BINARY 数据类型。</p> <p><b>string_expr2</b> 是模式字符串，或要在第一个表达式 (<i>string_expr1</i>) 中搜索的字符串表达式，表示形式为 CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY 或 BINARY 数据类型。</p> <p><b>string_expr3</b> 是替换字符串表达式，表示形式为 CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY 或 BINARY 数据类型。</p> |

## 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

- 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 **数据类型** CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY 或 BINARY。请参见第 3 章 “SQL 数据类型”。

**作用** 第 181 页的 “LENGTH 函数 [ 字符串 ]”。

有关字符串函数的常规信息，请参见第 103 页的 “字符串函数”。

## STRING 函数 [ 字符串 ]

**作用** 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** **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。

**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。STRING 函数还可用于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 ]

**作用** 将字符串值转换为唯一标识符 (UUID 或 GUID) 值。

**语法** **STRTOUUID** ( *string-expression* )

**参数** **string-expression** 字符串，格式为 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

**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     | <p>将一个格式为 <code>xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</code>（其中 <i>x</i> 是十六进制数字）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。</p> <p>可以使用 <code>STRTOUUID</code> 将 UUID 值插入 Sybase IQ 数据库。</p> |
| 标准和兼容性 | <ul style="list-style-type: none"> <li>• <b>SQL</b> ISO/ANSI SQL 语法的 Transact-SQL 扩展。</li> <li>• <b>Sybase</b> 不受 Adaptive Server Enterprise 支持。</li> </ul>                                  |
| 另请参见   | <p><a href="#">第 192 页的“NEWID 函数 [ 杂项 ]”</a></p> <p><a href="#">第 265 页的“UIDTOSTR 函数 [ 字符串 ]”</a></p> <p><a href="#">第 71 页的“二进制数据类型”中的 UNIQUEIDENTIFIER</a></p>                             |

## STUFF 函数 [ 字符串 ]

|        |                                                                                                                                                                                                                                                                                                                  |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作用     | 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     | <b>STUFF</b> ( <i>string-expression1</i> , <i>start</i> , <i>length</i> , <i>string-expression2</i> )                                                                                                                                                                                                            |
| 参数     | <p><b>string-expression1</b> 要用 STUFF 函数修改的字符串。</p> <p><b>start</b> 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。</p> <p><b>length</b> 要删除的字符数。</p> <p><b>string-expression2</b> 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。</p>                                                                                               |
| 示例     | <p>下面的语句返回值 “chocolate pie”：</p> <pre>SELECT STUFF( 'chocolate cake', 11, 4, 'pie' ) FROM iq_dummy</pre>                                                                                                                                                                                                         |
| 用法     | <p>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。要使用 STUFF 插入字符串，请将长度设为零。</p> <p>在以下情况下，STUFF 函数将返回 NULL 结果：</p> <ul style="list-style-type: none"> <li>• 前三个参数中的任一参数为 NULL 值。</li> <li>• <code>start</code> 或 <code>length</code> 参数为负值。</li> <li>• <code>start</code> 参数大于 <code>string-expression1</code> 的长度。</li> </ul> |
| 标准和兼容性 | <ul style="list-style-type: none"> <li>• <b>SQL</b> ISO/ANSI SQL 语法的供应商扩展。</li> <li>• <b>Sybase</b> 与 Adaptive Server Enterprise 兼容。</li> </ul>                                                                                                                                                                  |
| 另请参见   | <a href="#">第 171 页的“INSERTSTR 函数 [ 字符串 ]”</a>                                                                                                                                                                                                                                                                   |

## SUBSTRING 函数 [ 字符串 ]

**作用** 返回字符串的子字符串。

**语法** { **SUBSTRING** | **SUBSTR** } ( *string-expression*, *start* [ , *length* ] )

**参数** **string-expression** 要从中返回子字符串的字符串。

**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 *length* 表示子字符串在起始位置的右侧 *length* 个字符处结束，而负 *length* 指定子字符串在起始位置的左侧 *length* 个字符处结束。

**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。

**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。改用 SUBSTRING。

**另请参见**

《Sybase IQ 中的非结构化数据分析》中的第 9 章 “函数支持”

## SUBSTRING64 函数

- 作用** 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。
- 请参见《Sybase IQ 中的非结构化数据分析》中的第 9 章“函数支持”。
- 用户必须经过专门授权才能使用大对象数据类型 LONG BINARY 和 LONG VARCHAR。有关非结构化数据分析选件的详细信息，请参见《Sybase IQ 中的非结构化数据分析》。

## SUM 函数 [ 集合 ]

- 作用** 返回每一组行的指定表达式总数。
- 语法** `SUM ( expression | DISTINCT column-name )`
- 参数** **expression** 要总计的对象。这通常是列名称。  
**DISTINCT column-name** 计算每组行的 *column-name* 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。
- 示例** 下面的语句返回 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** [第 114 页的“AVG 函数 \[集合 \]”](#)
[第 131 页的“COUNT 函数 \[集合 \]”](#)
《系统管理指南，卷 2》中的第 2 章“使用 OLAP”

SUSER_ID 函数 [系统]

作用 返回整数用户标识号。

语法 **SUSER_ID** ([*user-name*])

参数 **user-name** 用户名。

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

[第 248 页的 “SUSER_NAME 函数 \[系统 \]”](#)

[第 264 页的 “USER_ID 函数 \[系统 \]”](#)

SUSER_NAME 函数 [系统]

作用 返回用户名。

语法 **SUSER_NAME** ([*user-id*])

参数 **user-id** 用户标识号。

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 Adaptive Server Enterprise 函数。在 Adapter Server Enterprise 中，SUSER_NAME 返回服务器用户名。

另请参见

[第 248 页的 “SUSER_ID 函数 \[系统 \]”](#)

[第 264 页的 “USER_NAME 函数 \[系统 \]”](#)

TAN 函数 [数值]

作用	返回一个数字的正切值。
语法	TAN (<i>numeric-expression</i>)
参数	numeric-expression 角度（以弧度为单位）。
示例	下面的语句返回值 0.572561： <pre>SELECT TAN(0.52) FROM iq_dummy</pre>
标准和兼容性	<ul style="list-style-type: none"> • SQL ISO/ANSI SQL 语法的供应商扩展。 • Sybase 与 Adaptive Server Enterprise 兼容。
另请参见	第 128 页的“COS 函数 [数值]” 第 232 页的“SIN 函数 [数值]”

TODAY 函数 [日期和时间]

作用	返回当前日期。这是 CURRENT DATE 的传统语法。
语法	TODAY (*)
示例	下列语句根据系统时钟返回当前日期。 <pre>SELECT TODAY(*) FROM iq_dummy</pre>
标准和兼容性	<ul style="list-style-type: none"> • 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]

作用	删除字符串中的前导和尾随空白。
语法	TRIM (<i>string-expression</i>)
参数	string-expression 要剪裁的字符串。

注释 TRIM 函数的结果数据类型为 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。

有关详细信息，请参见[第 223 页的“REPLACE 函数 \[字符串 \]”](#)。

- 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。改为使用 LTRIM 和 RTRIM。
- 另请参见                [第 185 页的 “LTRIM 函数 \[ 字符串 \]”](#)  
[第 229 页的 “RTRIM 函数 \[ 字符串 \]”](#)

## TRUNCNUM 函数 [ 数值 ]

- 作用                    在小数点后的指定位数截断数字。
- 语法                    **TRUNCNUM** ( *numeric-expression*, *integer-expression* )
- 参数                    **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。
- 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。  
您可通过将 ROUND、FLOOR 和 CEILING 组合使用来提供类似功能。
- 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                [第 226 页的 “ROUND 函数 \[ 数值 \]”](#)

---

## TS\_ARMA\_AR 函数 [ 时序 ]

**作用** 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)
```

---

**注释** 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 ]

**作用** 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)
```

---

**注释** 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 ]

**作用** 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)
```

---

**注释** 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 ]

作用 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*)

---

**注释** 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 ]

作用 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*)

---

**注释** 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 ]

作用 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*)

---

**注释** 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 ]

**作用** 第 252 页的“TS\_AUTO\_ARIMA 函数 [ 时序 ]”的支持函数。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)`

---

**注释** 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 ]

**作用** 第 253 页的“TS\_AUTO\_ARIMA\_RESULT\_AICC 函数 [ 时序 ]”的支持函数。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)`

---

**注释** 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 ]

**作用** 第 252 页的“TS\_AUTO\_ARIMA 函数 [ 时序 ]”的支持函数。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)`

---

**注释** 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 ]

**作用** 第 252 页的“TS\_AUTO\_ARIMA 函数 [ 时序 ]”的支持函数。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)`

---

**注释** 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 ]

**作用** 第 252 页的“TS\_AUTO\_ARIMA 函数 [ 时序 ]”的支持函数。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)`

---

**注释** 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 ]

**作用** 第 252 页的“TS\_AUTO\_ARIMA 函数 [ 时序 ]”的支持函数。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)`

---

**注释** 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 ]

**作用** 第 252 页的“TS\_AUTO\_ARIMA 函数 [ 时序 ]”的支持函数。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)`

---

**注释** 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 [ 时序 ]

**作用** 第 252 页的“TS\_AUTO\_ARIMA 函数 [ 时序 ]”的支持函数。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)`

---

**注释** 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 ]

**作用** 第 252 页的“TS\_AUTO\_ARIMA 函数 [ 时序 ]”的支持函数。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)`

---

**注释** 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 [ 时序 ]

**作用** 第 252 页的“TS\_AUTO\_ARIMA 函数 [ 时序 ]”的支持函数。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)`

---

**注释** 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 ]

**作用** 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)`

---

**注释** 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 ]

**作用** 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)`

---

**注释** 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 ]

作用 区分季节性或非季节性时序。

语法 **TS\_DIFFERENCE** (*timeseries\_expression*, *period1* [, *period2* [, ...*period10*]]) **OVER** (*window-spec*)

---

**注释** 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 [ 时序 ]

作用 第 258 页的“TS\_GARCH 函数 [ 时序 ]”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* ...]])

---

**注释** 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 ]

作用 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*)

---

**注释** 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 ]

作用 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)

---

**注释** 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 ]

作用 第 258 页的“TS\_GARCH 函数 [ 时序 ]”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)

---

**注释** 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 ]

作用 第 258 页的“TS\_GARCH 函数 [ 时序 ]”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)

---

**注释** 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 ]

**作用** 第 258 页的“TS\_GARCH 函数 [ 时序 ]”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)`

---

**注释** 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 ]

**作用** 第 252 页的“TS\_AUTO\_ARIMA 函数 [ 时序 ]”和第 252 页的“TS\_AUTO\_ARIMA\_OUTLIER 函数 [ 时序 ]”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] ... ])`

---

**注释** 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 ]

**作用** 在给定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)`

---

**注释** 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 ]

**作用** 对单变量时序执行失拟检验。此函数与第 259 页的“TS\_LACK\_OF\_FIT 函数 [ 时序 ]”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)`

---

**注释** 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 ]

**作用** 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)`

---

**注释** 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 ]

**作用** 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)`

---

**注释** 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 ]

**作用** 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)
```

---

**注释** 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 ]

**作用** 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)
```

---

**注释** 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 ]

**作用** 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)
```

---

**注释** 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 ]

**作用** 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*)

---

**注释** 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 ]

**作用** VWAP 表示加权平均价。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。

与其他时序函数不同，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*)

---

**注释** 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 ]

**作用** 将字符串中的所有字符转换成大写形式。

**语法** **UCASE** ( *string-expression* )

**参数** **string-expression** 要转换成大写形式的字符串。

有关详细信息，请参见第 223 页的“REPLACE 函数 [ 字符串 ]”。

**示例** 下面的语句返回值 “CHOCOLATE”：

```
SELECT UCASE('ChocoLate') FROM iq_dummy
```

|        |                                                                                                                                                                          |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 用法     | UCASE 函数的结果数据类型为 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。                                                            |
| 标准和兼容性 | <ul style="list-style-type: none"><li>• <b>SQL</b> ISO/ANSI SQL 语法的供应商扩展。</li><li>• <b>Sybase</b> Adaptive Server Enterprise 不支持 UCASE，但 UPPER 可通过兼容方式提供相同的功能。</li></ul> |
| 另请参见   | <a href="#">第 178 页的“LCASE 函数 [ 字符串 ]”</a><br><a href="#">第 263 页的“UPPER 函数 [ 字符串 ]”</a>                                                                                 |

## UPPER 函数 [ 字符串 ]

|        |                                                                                                                                              |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 作用     | 将字符串中的所有字符转换成大写形式。                                                                                                                           |
| 语法     | <b>UPPER</b> ( <i>string-expression</i> )                                                                                                    |
| 参数     | <b>string-expression</b> 要转换成大写形式的字符串。<br>有关详细信息，请参见 <a href="#">“REPLACE 函数 [ 字符串 ]”</a> 。                                                  |
| 示例     | 下面的语句返回值 “CHOCOLATE”：<br><pre>SELECT UPPER( 'ChocoLate' ) FROM iq_dummy</pre>                                                                |
| 用法     | UPPER 函数的结果数据类型为 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。                                |
| 标准和兼容性 | <ul style="list-style-type: none"><li>• <b>SQL</b> ISO/ANSI SQL 语法的供应商扩展。</li><li>• <b>Sybase</b> 与 Adaptive Server Enterprise 兼容。</li></ul> |
| 另请参见   | <a href="#">第 178 页的“LCASE 函数 [ 字符串 ]”</a><br><a href="#">第 184 页的“LOWER 函数 [ 字符串 ]”</a><br><a href="#">第 262 页的“UCASE 函数 [ 字符串 ]”</a>       |

## USER\_ID 函数 [ 系统 ]

作用 返回整数用户标识号。

语法 **USER\_ID** ( [ *user-name* ] )

参数 **user-name** 用户名。

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

[第 248 页的 “SUSER\\_ID 函数 \[ 系统 \]”](#)

[第 264 页的 “USER\\_NAME 函数 \[ 系统 \]”](#)

## USER\_NAME 函数 [ 系统 ]

作用 返回用户名。

语法 **USER\_NAME** ( [ *user-id* ] )

参数 **user-id** 用户标识号。

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 Adapter Server Enterprise 中，**USER\_NAME** 返回用户名，而不是服务器用户名。

另请参见

[第 248 页的 “SUSER\\_NAME 函数 \[ 系统 \]”](#)

[第 264 页的 “USER\\_ID 函数 \[ 系统 \]”](#)

## UUIDTOSTR 函数 [ 字符串 ]

- 作用** 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** `uuid-expression` 唯一标识符值。
- 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格式为 `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`（其中的 x 是十六进制数字）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** [第 192 页的“NEWID 函数 \[杂项 \]”](#)
[第 244 页的“STRTOUUID 函数 \[字符串 \]”](#)
[第 71 页的“二进制数据类型”中的 UNIQUEIDENTIFIER](#)

VAR_POP 函数 [集合]

- 作用** 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** `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）。
- 示例** 下面的语句列出不同时间段每个订单的项目数的平均值和方差：

```
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
```

年份	季度	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

[第 92 页的“分析函数”](#)

《系统管理指南，卷 2》中的 [第 2 章“使用 OLAP”](#)

VAR_SAMP 函数 [集合]

作用 计算由数值表达式组成的样本的统计方差，类型为 DOUBLE。

注释 VAR_SAMP 是 VARIANCE 的别名。

语法

VAR_SAMP ([ALL] *expression*)

参数

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）。

示例

下面的语句列出不同时间段每个订单的项目数的平均值和方差：

```
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
```

年份	季度	平均值	方差
2000	1	25.775148	205.1158
2000	2	27.050847	227.0939
...

用法 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。

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** 不受 Adaptive Server Enterprise 支持。

另请参见

第 92 页的“分析函数”

第 267 页的“VARIANCE 函数 [集合]”

《系统管理指南，卷 2》中的第 2 章“使用 OLAP”

VARIANCE 函数 [集合]

作用 返回一组数字的方差。

语法 **VARIANCE** ([ALL] *expression*)

参数 **expression** 任何数字数据类型（FLOAT、REAL 或 DOUBLE）表达式。

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
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 返回双精度浮点型数据类型的结果。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

[第 238 页的“STDDEV 函数 \[集合\]”](#)

[第 266 页的“VAR_SAMP 函数 \[集合\]”](#)

《系统管理指南，卷 2》中的第 2 章“使用 OLAP”

WEEKS 函数 [日期和时间]

作用 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 *datetime-expression* 日期和时间。

integer-expression 要添加到 *datetime-expression* 中的周数。如果 *integer-expression* 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 *datetime-expression* 显式转换为 DATETIME 数据类型。

有关转换数据类型的信息，请参见第 117 页的“CAST 函数 [数据类型转换]”。

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，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。

WEIGHTED_AVG 函数 [集合]

作用 计算算术（或线性）加权平均值。加权平均值是一个平均值，对于该平均值，将为每个要平均的数量分配一个权重。加权确定组成平均值的每个数量的相对重要性。

语法 **WEIGHTED_AVG** (*expression*)
OVER (*window-spec*)

window-spec: 请参见下面的“用法”部分。

参数 **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 4-1: WEIGHTED_AVG 计算

$$WMA_M = \frac{np_M + (n-1)p_{M-1} + \cdots + 2p_{M-n+2} + p_{M-n+1}}{n + (n-1) + \cdots + 2 + 1}$$

若要夸大加权，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。

有关如何指定窗口的信息，请参见第 92 页的“分析函数”。

示例

下面的示例按部门返回佛罗里达州员工的加权平均薪水，其中最近雇用员工的薪水对平均值贡献了最多加权：

```
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 4-23: 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。

WIDTH_BUCKET 函数 [数值]

作用

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

expression 是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 *expr* 的计算结果为 Null，则表达式返回 Null。

min_value 解析为 *expr* 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 *expr* 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。
- 另请参见 [第 196 页的“NTILE 函数 \[分析 \]”](#)，它可创建等高直方图。
《系统管理指南，卷 2》中的 [第 2 章“使用 OLAP”](#)

YEAR 函数 [日期和时间]

- 作用 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 **datetime-expression** 日期和时间。
- 示例 下面的语句返回 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 [第 273 页的“YEARS 函数 \[ 日期和时间 \]”](#)

## YEARS 函数 [ 日期和时间 ]

- 作用 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 **datetime-expression** 日期和时间。
- integer-expression** 要添加到 *datetime-expression* 中的年数。如果 *integer-expression* 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 *datetime-expression* 显式转换为 DATETIME 数据类型。
- 有关转换数据类型的信息，请参见 [第 117 页的“CAST 函数 \[ 数据类型转换 \]”](#)。

示例

下面的语句返回值 1998:

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

下面的语句返回值 2, 表示两个日期之间的差值。

```
SELECT YEARS('1997-07-13 06:07:12',
 '1999-09-13 10:07:12') FROM iq_dummy
```

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

[第 273 页的“YEAR 函数 \[日期和时间\]”](#)

## YMD 函数 [ 日期和时间 ]

|        |                                                                                                                                                                                                                                                                                                                                                            |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作用     | 返回对应于给定年、月、日的日期值。                                                                                                                                                                                                                                                                                                                                          |
| 语法     | <b>YMD</b> ( <i>integer-expression1</i> , <i>integer-expression2</i> , <i>integer-expression3</i> )                                                                                                                                                                                                                                                        |
| 参数     | <b>integer-expression1</b> 年。<br><b>integer-expression2</b> 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。<br><b>integer-expression3</b> 表示日的数字。日可以是任意整数，将相应地调整日期。                                                                                                                                                                                                     |
| 示例     | <p>下面的语句返回值 1998-06-12:</p> <pre>SELECT YMD( 1998, 06, 12 ) FROM iq_dummy</pre> <p>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:</p> <pre>SELECT YMD( 1992, 15, 1 ) FROM iq_dummy</pre> <p>下面的语句返回值 1993-02-28:</p> <pre>SELECT YMD ( 1992, 15, 1-1 ) FROM iq_dummy</pre> <p>下面的语句返回值 1992-02-29:</p> <pre>SELECT YMD ( 1992, 3, 1-1 ) FROM iq_dummy</pre> |
| 标准和兼容性 | <ul style="list-style-type: none"><li>• <b>SQL</b> ISO/ANSI SQL 语法的供应商扩展。</li><li>• <b>Sybase</b> 与 Adaptive Server Enterprise 兼容。</li></ul>                                                                                                                                                                                                               |



## 关于本章

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。

本章介绍 Sybase IQ 的一些特有功能，这些功能在其它 SQL 实现中并不常见。

## 目录

| 主题                           | 页码  |
|------------------------------|-----|
| <a href="#">Sybase IQ 功能</a> | 277 |

## Sybase IQ 功能

以下 Sybase IQ 功能是一些其它 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：

- **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。有关这些函数的说明，请参见第 4 章“SQL 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。Adaptive Server IQ 随后将强制实现主键的唯一性，并确保主键中的列不包含空值。

**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 允许在表之间使用自动连接。除其它实现中支持的 **NATURAL** 和 **OUTER** 连接运算符外，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。

## 变更表

对 **ALTER TABLE** 进行了扩展。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。请参见《参考：语句和选项》的“**SQL 语句**”中的“**ALTER TABLE 语句**”。

**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。有关可用函数的完整列表，请参见第 4 章“SQL 函数”。

**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。



关于本章

本章介绍对 Sybase IQ 数据库中对象的大小和数量的限制。有关仅适用于一个平台的限制，请参见该平台特定的文档。

目录

| 主题      | 页码  |
|---------|-----|
| 大小和数量限制 | 281 |

## 大小和数量限制

表 6-1 列出了对 Sybase IQ 数据库中对象的大小和数量的限制。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。

**表 6-1: Sybase IQ 数据库对象的大小和数量限制**

| 项目           | 限制                                                                                                                                                         |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Catalog 文件大小 | 除采用 Fat 32 文件系统的 Windows 系统（具有 4GB 的限制）外，对于所有平台最大值均为 1TB。带 NTFS 的 Windows 系统支持 1TB 的最大大小。Sybase IQ 不支持在 NAS（网络附加存储）设备上创建 dbspace。                          |
| 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。<br>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。另请参见“dbfile 的最大数目”。                                                                                                                            |
| dbspace 大小   | 裸设备：最大大小为 4TB。<br>操作系统文件：操作系统支持的最大大小。                                                                                                                      |
| 字段大小         | 对于 BINARY 为 255 字节，对于 VARBINARY 为 32,767 字节。<br>对于 CHAR、VARCHAR 为 32,767。<br>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。<br>对于多列索引为 5300 字节。                                                                                                                        |
| 字符串文字的最大长度   | 32KB                                                                                                                                                       |

| 项目                        | 限制                                                                                                                                                                                                                                                                                     |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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。<br>如果 SQL 语句较长，可使用 <code>-gss</code> 增加目录堆栈大小，并使用 <code>-c</code> 或 <code>-ch</code> 加 <code>-cl</code> 的组合增加目录内存高速缓存大小。<br>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 <code>-gp</code> 设置启动服务器，不过通常 Sybase 建议使用缺省值 <code>-gp 4096</code> 。 |
| 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）。<br>200（Windows 上的 32 位和 64 位平台）。                                                                                                                                                                                                              |
| 临时提取文件的最大大小               | 通过 <code>TEMP_EXTRACT_SIZE</code> n 选项设置。平台限制如下：<br>AIX & HP-UX: 0 — 64GB<br>Sun Solaris: 0 — 512GB<br>Windows: 0 — 128GB<br>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 <code>NOFILE</code> 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。                                                                                                                                                                                                                                                                               |
| 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。                                                                                                                                                                                                                                                                             |

## 关于本章

本章说明 Sybase IQ 数据库中系统提供的可用于检索系统信息的各存储过程。

## 目录

| 主题                                                   | 页码  |
|------------------------------------------------------|-----|
| <a href="#">系统过程概述</a>                               | 283 |
| <a href="#">系统存储过程</a>                               | 285 |
| <a href="#">目录存储过程</a>                               | 413 |
| <a href="#">Adaptive Server Enterprise 系统和目录存储过程</a> | 421 |
| <a href="#">SQL Anywhere 支持的过程</a>                   | 424 |

## 系统过程概述

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》的附录 A “Multiplex 引用” 中的 “系统过程”。
- Transact-SQL 系统和目录存储过程。有关这些系统过程的列表，请参见第 421 页的 “Adaptive Server Enterprise 系统和目录存储过程”。

本章介绍各系统过程。

《Sybase IQ 中的非结构化数据分析》的第 5 章 “存储过程支持” 中介绍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 中所能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。

有关语法变化形式的说明，请参见表 7-1。

**表 7-1: 存储过程语法变化形式**

| 语法                                                              | 语法类型                     | 说明                                                 |
|-----------------------------------------------------------------|--------------------------|----------------------------------------------------|
| <i>procedure_name</i> (' <i>param</i> ')                        | Sybase IQ                | 如果将参数包含在括号中，则需要使用引号。                               |
| <i>procedure_name</i> ' <i>param</i> '                          | Sybase IQ                | 如果将参数包含在引号中，则括号是可选项。                               |
| <i>procedure_name param</i>                                     | Transact-SQL             | 如果省略参数两侧的引号，则还必须省略括号。                              |
| <i>procedure_name</i>                                           | Sybase IQ 或 Transact-SQL | 如果在 <i>dbisql</i> 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。 |
| <i>call procedure_name</i><br>( <i>param</i> =' <i>value</i> ') | 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

系统存储过程由用户 ID dbo 拥有。本节中的系统过程用于执行 IQ 主存储区中的系统管理员任务。

---

**注释** 缺省情况下，由 dbisqlc 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 dbisql 菜单中选择“命令”>“选项”，然后为“限制显示列”(Limit Display Columns)和/或“限制输出列”(Limit Output Columns)选择并输入较大的值。

---

### sa\_char\_terms 系统过程

**作用** 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》的第 5 章“存储过程支持”中的“sa\_char\_terms 系统过程”。

### sa\_dependent\_views 过程

**作用** 返回某个给定表或视图的所有相关视图的列表。

请参见 SQL Anywhere 文档中的“sa\_dependent\_views 系统过程”，具体位置为：“SQL Anywhere 11.0.1”>“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”。

### sa\_external\_library\_unload 过程

**作用** 卸载外部库。

请参见《Sybase IQ 中的非结构化数据分析》的第 5 章“存储过程支持”中的“sa\_external\_library\_unload 过程”。

## sa\_list\_external\_library 过程

作用

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》的第 5 章“存储过程支持”中的“sa\_list\_external\_library 过程”。

## sa\_nchar\_terms 系统过程

作用

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》的第 5 章“存储过程支持”中的“sa\_nchar\_terms 系统过程”。

## sa\_text\_index\_vocab 系统过程

作用

列出 TEXT 索引中出现的所有术语，以及每个术语所在的索引值的总数。

请参见《Sybase IQ 中的非结构化数据分析》的第 5 章“存储过程支持”中的“sa\_text\_index\_vocab 系统过程”。

## sa\_verify\_password 过程

作用

检验当前用户的口令。

请参见 SQL Anywhere 文档中的“sa\_verify\_password 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_get\_user\_status 过程

作用

允许确定用户的当前状态。

请参见 SQL Anywhere 文档中的“sa\_get\_user\_status 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

权限

查看所有用户的状态时需要具有 DBA 或 USER ADMIN 权限。没有 DBA 或 USER ADMIN 权限的用户可以查看他们自己的状态。

## sp\_expireallpasswords 过程

|      |                                                                             |
|------|-----------------------------------------------------------------------------|
| 作用   | 导致所有用户口令立即过期。                                                               |
| 语法 1 | <b>call sp_expireallpasswords</b>                                           |
| 语法 2 | <b>sp_expireallpasswords</b>                                                |
| 另请参见 | 第 368 页的 “sp_iqpassword 过程”<br>《参考：语句和选项》的第 1 章 “SQL 语句” 中的 CREATE USER 语句。 |
| 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：<br><pre>call sp_expireallpasswords</pre>                      |

## sp\_iqaddlogin 过程

|      |                                                                                                                                                                                                                                                                                                                                                                        |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作用   | 向指定的登录策略添加一个新的 Sybase IQ 用户帐户。                                                                                                                                                                                                                                                                                                                                         |
| 语法 1 | <b>call sp_iqaddlogin</b> ('username_in', 'pwd',<br>[ 'password_expiry_on_next_login' ] [, 'policy_name' ])                                                                                                                                                                                                                                                            |
| 语法 2 | <b>sp_iqaddlogin</b> 'username_in', 'pwd',<br>[ 'password_expiry_on_next_login' ] [, 'policy_name' ]                                                                                                                                                                                                                                                                   |
| 语法 3 | <b>sp_iqaddlogin</b> username_in, pwd, [ password_expiry_on_next_login ] [,<br>policy_name ]                                                                                                                                                                                                                                                                           |
| 用法   | <b>username_in</b> 用户的登录名。登录名必须符合标识符规则。<br><b>pwd</b> 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。<br><b>password_expiry_on_next_login</b> （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）。<br><b>policy_name</b> （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。<br>使用 sp_iqaddlogin 创建并设置为在一天后过期的 <b>username_in/pwd</b> 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。                                                                                                                                                                                                                                                                                                                                                           |
| 另请参见 | 第 368 页的 “sp_iqpassword 过程”<br>《参考：语句和选项》的第 1 章 “SQL 语句” 中的 CREATE USER 语句。                                                                                                                                                                                                                                                                                            |

**说明** 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。

---

**注释** 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。

**示例** 这些调用按照名为 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'
```

## sp\_iqbackupdetails 过程

**作用** 显示特定备份中包括的所有 dbfile。

**语法** `sp_iqbackupdetails backup_id`

**参数** `backup_id` 指定备份操作的事务标识符。

---

**注释** 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。

说明 `sp_iqbackupdetails` 返回以下结果：

**表 7-2: `sp_iqbackupdetails` 列**

| 列名                               | 说明                                                                                                                                                                                                  |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>backup_id</code>           | 备份事务的标识符。                                                                                                                                                                                           |
| <code>backup_time</code>         | 备份的时间。                                                                                                                                                                                              |
| <code>backup_type</code>         | 备份的类型：“Full”、“Incremental since incremental”或“Incremental since full”。                                                                                                                              |
| <code>selective_type</code>      | 备份的子类型：“All inclusive”、“All RW files in RW dbspaces”、“Set of RO dbspace/file”。                                                                                                                      |
| <code>depends_on_id</code>       | 备份所依赖的上次备份的标识符。                                                                                                                                                                                     |
| <code>dbspace_id</code>          | 正在备份的 <code>dbspace</code> 的标识符。                                                                                                                                                                    |
| <code>dbspace_name</code>        | <code>SYSIQBACKUPHISTORYDETAIL</code> 的 <code>dbspace</code> 的名称。如果该 <code>dbspace</code> 名称与给定 <code>dbspace_id</code> 的 <code>SYSDBSPACE</code> 中的 <code>dbspace</code> 名称匹配，则该名称存在。否则为“Null”。    |
| <code>dbspace_rwstatus</code>    | “读写”或“只读”。                                                                                                                                                                                          |
| <code>dbspace_createid</code>    | <code>Dbspace</code> 创建的事务标识符。                                                                                                                                                                      |
| <code>dbspace_alterid</code>     | 变更 <code>DBSPACE</code> 读写模式的事务标识符。                                                                                                                                                                 |
| <code>dbspace_online</code>      | 状态。值为“Online”或“Offline”。                                                                                                                                                                            |
| <code>dbspace_size</code>        | 备份时 <code>dbspace</code> 的大小 (KB)。                                                                                                                                                                  |
| <code>dbspace_backup_size</code> | 在 <code>dbspace</code> 中备份的数据的大小 (KB)。                                                                                                                                                              |
| <code>dbfile_id</code>           | 正在备份的 <code>dbfile</code> 的标识符。                                                                                                                                                                     |
| <code>dbfile_name</code>         | 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”。                                                                                                                                                                 |
| <code>dbfile_rwstatus</code>     | “读写”或“只读”。                                                                                                                                                                                          |
| <code>dbfile_createid</code>     | <code>Dbfile</code> 创建的事务标识符。                                                                                                                                                                       |
| <code>dbfile_alterid</code>      | 变更 <code>FILE</code> 读写模式的变更 <code>DBSPACE</code> 的事务标识符。                                                                                                                                           |
| <code>dbfile_size in MB</code>   | <code>dbfile</code> 的大小 (KB)。                                                                                                                                                                       |
| <code>dbfile_backup_size</code>  | <code>dbfile</code> 备份的大小 (KB)。                                                                                                                                                                     |
| <code>dbfile_path</code>         | <code>SYSBACKUPDETAIL</code> 的 <code>dbfile</code> 路径，如果该路径与给定 <code>dbspace_id</code> 和 <code>dbfile_id</code> 的 <code>SYSDBFILE</code> 中的物理文件路径 (“ <code>file_name</code> ”) 匹配，则该路径存在。否则为“Null”。 |

示例 `sp_iqbackupdetails` 的输出样本：

```

backup_id backup_time backup_type selective_type 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 [第 445 页的“SYSIQBACKUPHISTORY 系统视图”](#)

## sp\_iqbackupsummary 过程

- 作用** 总结执行的备份操作。
- 语法** `sp_iqbackupsummary [ timestamp or backup_id ]`
- 参数** *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。
- 说明** sp\_iqbackupsummary 返回以下结果：

**表 7-3: sp\_iqbackupsummary 列**

| 列名             | 说明                                                                            |
|----------------|-------------------------------------------------------------------------------|
| 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” |
| 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 | 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<br>'c:\\\\temp\\\\b1' |

## sp\_iqcardinality\_analysis 过程

**作用** 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 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
- 单字节 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
- WORD 索引

**语法** `sp_iqcardinality_analysis ([ 'table_name' ], [ 'table_owner' ], [ 'script' ])`

参数

**table\_name** 表的名称。

**table\_owner** 表所有者的姓名。如果未指定此参数，该过程将查找当前用户所拥有的表。

**script** 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 7-4: 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。<br>— 可以创建单字节 FP 索引。<br>— 调用存储过程：<br>sp_iqrebuildindex<br>'onebytefp','column c1 255'          |
| onebytefp  | DBA         | c1          | 10          | LF         | — “c1” 列不具有 LF 索引并且基数小于 1000。<br>— 可以使用 CREATE INDEX 语句创建 LF 索引：<br>CREATE LF INEDX<br>IQ_T400_c1_LF ON<br>DBA.onebytefp (c1) |

示例 2

```
sp_iqcardinality_analysis 'onebytefp', 'DBA', 'script'
```

表 7-5: 主控台输出示例 2

**索引建议**

- “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

作用

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,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。

---

**注释** `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

**database** 如果目标是数据库，则所有 *dbspace* 都必须处于联机状态。

**index-type** 以下索引类型之一：FP、CMP、LF、HG、HNG、WD、DATE、TIME、DTTM、TEXT。

如果指定的 *index-type* 在目标中不存在，则返回错误消息。如果指定多个索引类型，并且目标仅包含这些索引类型中的一部分，则 **sp\_iqcheckdb** 将处理现有索引类型。

**index-name** 可能包含所有者和表限定符：[*owner.*] *table-name.* *index-name*

如果未指定 *owner*，则将以该顺序替代当前用户及数据库所有者 (*dbo*)。如果不指定 *table*，则 *index-name* 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 *owner*，则将以该顺序替代当前用户及数据库所有者 (*dbo*)。*table-name* 不能是临时表或预连接表。

---

**注释** 如果表名称或索引名称包含空格，请将 *table-name* 或 *index-name* 参数放在双引号内：

```
sp_iqcheckdb 'check index "dbo.sstab.i2" resources 75'
```

---

**partition-name** *partition-name* 参数不包含限定符。如果它包含空格，请将其放在双引号内。

**dbspace-name** *dbspace-name* 参数不包含限定符。如果它包含空格，请将其放在双引号内。

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` 目标检查属于该 `dbspace` 的数据库页的子集。`dbspace` 必须联机。当表只有一个 `dbspace` 时，`dbspace` 和数据库目标在语义上是等同的。

**resource-percent** 输入参数 *resource-percent* 必须是大于零的整数。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 *resource-percent* = 100（缺省值），则每个 CPU 创建一个线程。如果 *resource-percent* > 100，则有比 CPU 数更多的线程，对于某些计算机配置，这可能会提高性能。线程数的最小值为 1。

---

**注释** `sp_iqcheckdb` 参数字符串必须放在单引号内，并且长度不能超过 255 个字节。

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。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 说明 | <p><code>sp_iqcheckdb</code> 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 <code>sp_iqdbstatistics</code> 过程。<code>sp_iqdbstatistics</code> 显示最近一次执行 <code>sp_iqcheckdb</code> 的最新结果。</p> <p><code>sp_iqcheckdb</code> 可以执行几种不同的功能，具体取决于指定的参数。用于检查和修复数据库一致性的模式是：</p> <p><b>分配模式</b> 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 <code>dbspace</code> 的块映射信息检查分配情况。不检查索引一致性。</p> <p>检测重复块（两个或多个对象声明对其的所有权的块）或额外块（对象所拥有的未分配块）。</p> <p>检测数据库或 <code>dbspace</code> 目标的泄漏块（在指定目标中任何对象都未声明的已分配块）。</p> <p>当目标是分区表时，分配模式将：</p> <ul style="list-style-type: none"> <li>• 检查所有表分区分配位图的元数据</li> <li>• 检查表分配位图的元数据</li> <li>• 检验块映射条目与表的分配位图是否一致</li> </ul> |

- 检验表的分区分配位图没有重叠
- 检查在表的分区分配位图中定义的行是否构成表的存在位图的超集
- 检查在表的分区分配位图中定义的行是否构成表的分配位图的超集

---

**注释** 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。

---

以分配模式运行：

- 检测重复块或无主块（使用数据库或特定表或索引作为目标）
- 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。在与 Sybase IQ 技术支持联系之前，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配位图。

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 `dbspace` 的非 FP 索引及其相应 FP 索引的内容。如果指定的目标包含 FP 及相应的非 FP 索引的所有数据页，则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。

---

**注释** `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》。

以下示例演示 `sp_iqcheckdb` 过程的使用。

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，如以下命令所示：

```
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 以下示例演示 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：

- 无主 LVC 单元
- 重复的 LVC 单元行
- 未分配的 LVC 单元行

这些消息表明与 VARCHAR、VARBINARY、LONG BINARY (BLOB) 或 LONG VARCHAR (CLOB) 列不一致。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。

---

**注释** 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 dbspace 目标数。

表 7-6 总结了四种 `sp_iqcheckdb` 模式的操作和输出。

**表 7-6: `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<br>大部分索引错误 | 所有可用统计信息  | 每小时 60 GB |
| 检验 | 分配错误<br>所有索引错误  | 所有可用统计信息  | 每小时 15 GB |
| 沙漏 | 分配错误            | 仅输出分配统计信息 | 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

- 作用                    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。
- 说明                    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 7-7: *sp\_iqcheckoptions* 列

| 列名            | 说明                                                                    |
|---------------|-----------------------------------------------------------------------|
| User_name     | 已设置其选项的用户或组的名称。在创建数据库时，将为 public 组设置所有选项。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，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。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;<br>Application=DBTools;Signa |                 | Temporary   |
| DBA       | Login_procedure    | DBA.sp_iq_proce                              | sp_login_envir  | Permanent   |
| PUBLIC    | Login_procedure    | DBA.sp_iq_proce                              | sp_login_envir  | Permanent   |
| 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   | Option_type |
|-----------|--------------------|----------------------------------------------|-----------------|-------------|
| joel      | Ansi_update_constr | CURSORS                                      | Off             | Permanent   |
| PUBLIC    | Ansi_update_constr | Cursors                                      | Off             | Permanent   |
| joel      | Checkpoint_time    | 20                                           | 60              | Temporary   |
| joel      | Connection_authent | Company=MyComp;<br>Application=DBTools;Signa |                 | Temporary   |
| joel      | Login_procedure    | DBA.sp_iq_proce                              | sp_login_envir  | Permanent   |
| PUBLIC    | Login_procedure    | DBA.sp_iq_proce                              | sp_login_envir  | Permanent   |
| joel      | Max_Warnings       | 55                                           | 281474976710655 | Temporary   |
| joel      | Thread_count       | 25                                           | 0               | Temporary   |

## sp\_iqclient\_lookup 过程

**作用** 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** **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。

**说明** 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。

---

**注释** 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
```

---

**注释** 当用户指定了不正确的参数时，此结果为空。

---

## sp\_iqcolumn 过程

**作用** 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 1** 如果指定 `table_name`，而不指定 `table_owner`，则必须用 `NULL` 代替 `table_name`。例如，`sp_iqcolumn NULL, DBA`。

**语法 2** 参数可以任何顺序进行指定。将 `'table_name'` 和 `'table_owner'` 放在单引号内。

**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。

说明

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 7-8: sp\_iqcolumn 列**

| 列名              | 说明                                                                                                   |
|-----------------|------------------------------------------------------------------------------------------------------|
| table_name      | 表的名称                                                                                                 |
| table_owner     | 表的所有者                                                                                                |
| column_name     | 列的名称                                                                                                 |
| domain_name     | 数据类型                                                                                                 |
| width           | 具有精度和标度的数值数据类型的精度，或不具有标度的数值数据类型的存储宽度；字符数据类型的宽度                                                       |
| scale           | 数值数据类型的标度                                                                                            |
| nulls           | 如果列可以包含 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 |
| location        | 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                                                                                              |

示例

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

作用

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。

说明

不会报告在 SYSTEM 中创建的表的列。

**表 7-9: sp\_iqcolumnuse 列**

| 列名         | 说明           |
|------------|--------------|
| TableName  | 表名           |
| ColumnName | 列名           |
| 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）。

示例 `sp_iqcolumnuse` 过程的输出样本:

| TableName | ColumnName     | Owner | UID            | LastDT                  | NRef |
|-----------|----------------|-------|----------------|-------------------------|------|
| orders    | o_orderdate    | DBA   |                | 151 20070917 22:41:22.. | 13   |
| orders    | o_shippriority | DBA   |                | 154 20070917 22:41:22.. | 13   |
| lineitem  | l_orderkey     | DBA   |                | 186 20070917 22:41:22.. | 13   |
| lineitem  | l_extendedp..  | DBA   |                | 191 20070917 22:41:22.. | 13   |
| lineitem  | l_discount     | DBA   |                | 192 20070917 22:41:22.. | 13   |
| lineitem  | l_shipdate     | DBA   |                | 196 20070917 22:41:22.. | 13   |
| #tmp1     | expression     | DBA   | 10000000001218 | 20070917 22:57:36..     | 1    |
| #tmp1     | expression     | DBA   | 10000000001222 | 20070917 22:41:58..     | 1    |
| ...       |                |       |                |                         |      |

**注释** 上述示例中的较长编号为临时 ID。

另请参见

《性能和调优指南》的第 3 章“优化查询和删除”中的“监控负载”。  
第 320 页的“`sp_iqdbspace` 过程”、第 348 页的“`sp_iqindexadvice` 过程”、第 355 页的“`sp_iqindexuse` 过程”、第 399 页的“`sp_iqtableuse` 过程”、第 402 页的“`sp_iqunusedcolumn` 过程”、第 403 页的“`sp_iqunusedindex` 过程”、第 405 页的“`sp_iqunusedtable` 过程”和第 411 页的“`sp_iqworkmon` 过程”。

《参考：语句和选项》的第 2 章“数据库选项”中的“`INDEX_ADVISOR` 选项”。

## sp\_iqconnection 过程

作用

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。

说明

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，如表 7-10 所示：

**表 7-10: 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   | <ul style="list-style-type: none"> <li>• NULL 用于本地 / 用户连接。</li> <li>• 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。</li> </ul> |
| Multiplex 辅助服务器 | <ul style="list-style-type: none"> <li>• NULL 用于本地 / 用户连接。</li> <li>• 包含事务协调器服务器名 (连接源) 的值。</li> </ul>                             |

有关 Multiplex 功能的信息，请参见《使用 Sybase IQ Multiplex》。

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");
```

有关使用 RemotePWD 参数的详细信息，请参见 SQL Anywhere 文档中的“指定服务器上的数据库”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — 编程” > “SQL Anywhere 数据访问 API” > “SQL Anywhere JDBC 驱动程序” > “使用 jConnect JDBC 驱动程序” > “向驱动程序提供 URL”。

**表 7-11: sp\_iqconnection 列**

| 列名          | 说明                |
|-------------|-------------------|
| ConnHandle  | 连接的 ID 号。         |
| Name        | 服务器的名称。           |
| Userid      | 连接的用户 ID。         |
| 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。    |

| 列名                  | 说明                                                                                                                                                                                                                                                       |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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       | 该连接上最后一个 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。可能分配了一些线程，但是它们处于空闲状态。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     | 此连接在处理诸如排序、散列和临时位图这样的工作空间时所用的 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 10 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 端的交叉数。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 端的交叉数。您可能发现该列在确定连接活动时偶尔有用。                                                                                                                                                                                  |
| 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。                                                                                                                                                                             |

示例 `sp_iqconnection` 输出示例:

| ConnHandle | Name               | Userid | LastReqTime               | ReqType |
|------------|--------------------|--------|---------------------------|---------|
| 9          | 'IQ_MPX_SERVER_H'  | 'dbo'  | '2008-11-18 13:15:00.035' | 'EXEC'  |
| 11         | 'IQ_MPX_SERVER_H'  | 'dbo'  | '2008-11-18 13:15:00.046' | 'EXEC'  |
| 13         | 'IQ_MPX_SERVER_H'  | 'dbo'  | '2008-11-18 14:52:55.003' | 'EXEC'  |
| 15         | 'IQ_MPX_SERVER_H'  | 'dbo'  | '2008-11-18 14:53:25.005' | 'EXEC'  |
| 17         | 'SQL_DBC_49450e8'  | 'DBA'  | '2008-11-18 14:59:45.680' | 'OPEN'  |
| 44         | 'Sybase Central 1' | 'DBA'  | '2008-11-18 14:59:45.023' | 'CLOSE' |

| IQCmdType             | LastIQCmdTime         | IQCursors | LowestIQCursorState |
|-----------------------|-----------------------|-----------|---------------------|
| 'NONE'                | 2008-11-18 13:15:00.0 | 0         | 'NONE'              |
| 'NONE'                | 2008-11-18 13:15:00.0 | 0         | 'NONE'              |
| 'NONE'                | 2008-11-18 14:52:55.0 | 0         | 'NONE'              |
| 'NONE'                | 2008-11-18 14:53:25.0 | 0         | 'NONE'              |
| 'IQUTILITYOPENCURSOR' | 2008-11-18 14:59:45.0 | 0         | 'NONE'              |
| 'NONE'                | 2008-11-18 14:43:33.0 | 0         | 'NONE'              |

| IQthreads | TxnID | ConnCreateTime        | TempTableSpaceKB | TempWorkSpaceKB |
|-----------|-------|-----------------------|------------------|-----------------|
| 1         | 0     | 2008-11-18 13:14:09.0 | 0                | 0               |
| 1         | 0     | 2008-11-18 13:14:34.0 | 0                | 0               |
| 1         | 0     | 2008-11-18 13:14:55.0 | 0                | 0               |
| 1         | 0     | 2008-11-18 13:15:25.0 | 0                | 0               |
| 1         | 50024 | 2008-11-18 13:28:08.0 | 0                | 0               |
| 1         | 50545 | 2008-11-18 14:03:50.0 | 0                | 0               |

| IQconnID | satoiq_count | iqtosas_count | CommLink | NodeAddr       | LastIdle | MPXServerName |
|----------|--------------|---------------|----------|----------------|----------|---------------|
| 23198    | 28           | 12            | 'local'  | ''             | 2977     | 'mpx0631_r1'  |
| 23202    | 28           | 12            | 'local'  | ''             | 1503     | 'mpx0631_r2'  |
| 23207    | 127          | 12            | 'local'  | ''             | 10000    | 'mpx0631_w1'  |
| 23212    | 127          | 12            | 'local'  | ''             | 10000    | 'mpx0631_w2'  |
| 23267    | 658          | 66            | 'TCP/IP' | '10.18.60.181' | 9375     |               |
| 23443    | 510          | 54            | 'local'  | ''             | 1238     |               |

## sp\_iqconstraint 过程

**作用** 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。

**说明** 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,ck2

PTAB2 bob ASIQ_IDX_T27_HG unique ck1,ck2 selftab bob CK6FK4 Y
ASIQ_IDX_T206_I42_HG ck1,ck2

selftab bob ASIQ_IDX_T26_HG unique ck1,ck2 selftab bob CK3FK1 Y
ASIQ_IDX_T206_I42_HG ck1,ck2
```

显示的列分别是：

- 主强制表
- 表所有者
- 候选键索引
- 主键或唯一约束
- 主键列
- 外部表
- 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”）

## sp\_iqcontext 过程

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作用 | 按照连接跟踪和显示有关当前执行的语句的信息。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 语法 | <b>sp_iqcontext</b> [ <i>connhandle</i> ]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 用法 | 输入参数 <i>connhandle</i> 相当于 <b>Number</b> 连接属性并且是连接的 ID 号。<br>当用有效输入参数 <i>connhandle</i> 调用时， <b>sp_iqcontext</b> 仅返回该连接的信息。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 权限 | 需要具有 <b>DBA</b> 权限。必须向没有 <b>DBA</b> 权限的用户授予 <b>EXECUTE</b> 权限才能运行存储过程。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 说明 | <p><b>DBA</b> 使用 <b>sp_iqcontext</b> 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：</p> <ul style="list-style-type: none"><li>• 将语句文本与 <b>sp_iqconnection</b> 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</li><li>• 将语句文本与当 <b>-zr</b> 服务器选项设置为 <b>ALL</b> 或 <b>SQL</b> 时创建的 <b>SQL</b> 日志中的等效行匹配</li><li>• 使用链接信息将 <b>sp_iqcontext</b> 中的语句文本与 <b>.iqmsg</b> 文件中的等效行匹配，后者在当 <b>Sybase IQ</b> 可以收集相应信息时包含相应的查询计划</li><li>• 将语句文本与可能产生的 <b>IQ</b> 堆栈跟踪 (<i>stktrc-yyyymmdd-hhnnss_#.iq</i>) 匹配</li><li>• 将此信息与可能产生的操作系统堆栈跟踪（例如 <b>Sun Solaris</b> 上的 <b>pstack</b>）进行比较</li></ul> <p>所收集的语句文本的最大大小是 <b>Catalog</b> 存储的页大小。</p> |

**表 7-12: sp\_iqcontext 列**

| 列名                  | 说明                                                                                                                                                                          |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ConnOrCursor        | CONNECTION 或 CURSOR。                                                                                                                                                        |
| ConnHandle          | 连接的 ID 号。                                                                                                                                                                   |
| Name                | 服务器的名称。                                                                                                                                                                     |
| Userid              | 连接或游标的用户 ID。                                                                                                                                                                |
| numIQCursors        | 如果第 1 列是 CONNECTION，则是在此连接上打开的游标数。<br>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。                                                                                                 |
| 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。                                                                                                                                       |
| TxnID               | 当前事务的事务 ID。                                                                                                                                                                 |
| ConnOrCurCreateTime | 创建此连接或游标的时间。                                                                                                                                                                |
| IQConnID            | 作为 .iqmsg 文件中所有消息的一部分显示的 10 位数连接 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。请参见《性能和调优指南》的第 3 章“优化查询和删除”中的“设置查询优先级”。 |
| CmdLine             | 执行的用户命令的前 4096 个字符。                                                                                                                                                         |

**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。

```

CONNECTION 701773517 dba7 DBA 6 1 1324 2009-06-04 09:24:17.000 4 NO COMMAND
CURSOR 701773517 dba7 DBA 1 0 1324 2009-06-04 09:24:46.000 4 2 select * from fool
CURSOR 701773517 dba7 DBA 2 0 1324 2009-06-04 09:24:47.000 4 2 select a from fool
...
CURSOR 701773517 dba7 DBA 6 0 1324 2009-06-04 09:24:47.000 4 2 select e from fool
CONNECTION 1271624950 dba7 DBA 0 12 1377 2009-06-04 09:24:12.000 3 2 sp_iqcheckdb
CONNECTION 1841476383 dba7 DBA 10 1 1337 2009-06-04 09:24:19.000 5 2 call sp_iqcontext()
CURSOR 1841476383 dba7 DBA 1 0 1337 2009-06-04 09:24:47.000 5 2 select * from foo
...
CURSOR 1841476383 dba7 DBA 10 0 1337 2009-06-04 09:24:48.000 5 2 select i from foo

```

输出的第一行显示连接 701773517（IQ 连接 ID 4）。此连接在服务器 dba7 上，用户是 DBA。它有六个活动游标和一个 IQ 线程，并且是从事务 1324 创建的。当发出 sp\_iqcontext 时，此连接没有执行命令。后面六行输出列出了此连接使用的游标（这里仅显示三个）。

两个连接在运行存储过程。连接 1271624950 直接从 dbisql 运行 `sp_iqcheckdb`，它没有活动游标，但在使用 12 个 IQ 线程。连接 1841476383 已将 `sp_iqcontext` 作为过程调用，它仅使用 1 个 IQ 线程，并有 10 个活动游标（这里仅显示第一个和最后一个）。在这两个示例中，显示的都是存储过程的名称，而非在其中执行的代码行。

连接句柄（对于此示例中的第一个连接为 701773517）标识 `-zr` 日志中的结果。IQ 连接 ID（对此示例中的第一个连接为 4）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。用户命令片段之前的 2 表示这是中优先级查询。

## sp\_iqcopyloginpolicy 过程

|      |                                                                                                                                      |
|------|--------------------------------------------------------------------------------------------------------------------------------------|
| 作用   | 通过复制现有登录策略创建新的登录策略。                                                                                                                  |
| 语法 1 | <code>call sp_iqcopyloginpolicy ('existing-policy-name', 'new-policy-name')</code>                                                   |
| 语法 2 | <code>sp_iqcopyloginpolicy 'existing-policy-name', 'new-policy-name'</code>                                                          |
| 语法 3 | <code>sp_iqcopyloginpolicy existing-policy-name, new-policy-name<br/>policy_name ]</code>                                            |
| 用法   | <b>existing policy name</b> 要复制的登录策略。<br><b>new policy name</b> 要创建的新登录策略的名称 (CHAR(128))。                                            |
| 权限   | 需要具有 DBA 权限。                                                                                                                         |
| 另请参见 | <a href="#">第 368 页的“sp_iqpassword 过程”</a><br>《参考：语句和选项》中第 1 章“SQL 语句”中的 CREATE USER 语句                                              |
| 示例   | 以下存储过程通过从名为 <i>root</i> 的现有登录策略复制登录策略选项值创建名为 <i>lockeduser</i> 的新登录策略。<br><pre>call sp_iqcopyloginpolicy ('root','lockeduser')</pre> |

## sp\_iqcursorinfo 过程

|    |                                                               |
|----|---------------------------------------------------------------|
| 作用 | 显示有关当前在服务器上打开的游标的详细信息。                                        |
| 语法 | <code>sp_iqcursorinfo [ cursor-name ] [, conn-handle ]</code> |
| 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

**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。

**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。

可以在不带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 7-13: `sp_iqcursorinfo` 用法示例**

| 语法                                        | 输出                                          |
|-------------------------------------------|---------------------------------------------|
| <code>sp_iqcursorinfo</code>              | 显示有关当前在服务器上打开的所有游标的信息                       |
| <code>sp_iqcursorinfo 'cursor1'</code>    | 显示有关在所有连接中名为 <code>cursor1</code> 的所有游标的信息  |
| <code>sp_iqcursorinfo NULL, 3</code>      | 显示有关在连接 3 中的所有游标的信息                         |
| <code>sp_iqcursorinfo 'cursor2', 4</code> | 显示有关在连接 4 中名为 <code>cursor2</code> 的所有游标的信息 |

说明

`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` 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 过程返回以下列信息：

**表 7-14: sp\_iqcursorinfo 列**

| 列名         | 说明                                                    |
|------------|-------------------------------------------------------|
| Name       | 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。                                         |
| NumFetch   | 游标提取行的次数。同一行可以多次提取。                                   |
| NumUpdate  | 游标更新行的次数（如果游标可更新）。同一行可以多次更新。                          |
| NumDelete  | 游标删除行的次数（如果游标可更新）。                                    |
| NumInsert  | 游标插入行的次数（如果游标可更新）。                                    |
| RWTabOwner | 由游标在 RW 模式下打开的表的所有者。                                  |
| RWTabName  | 由游标在 RW 模式下打开的表的名称。                                   |
| CmdLine    | 用户执行的命令的前 4096 个字符                                    |

#### 示例

显示有关当前在服务器上打开的所有游标的信息：

```
sp_iqcursorinfo
```

```

Name ConnHandle IsUpd IsHold IQConnID UserID

crsr1 1 Y N 118 DBA
crsr2 3 N N 118 DBA

CreateTime CurrentRow NumFetch NumUpdate

2009-06-26 15:24:36.000 19 100000000 200000000
2009-06-26 15:38:38.000 20000 200000000

NumDelete NumInsert RWTabOwner RWTabName CmdLine

20000000 3000000000 DBA test1 call procl()
 call proc2()
```

另请参见

《参考：语句和选项》：[DECLARE CURSOR 语句 \[ESQL\] \[SP\]](#)、[DECLARE CURSOR 语句 \[T-SQL\]](#)、[UPDATE（定位）语句 \[ESQL\] \[SP\]](#) 和 [DELETE（定位）语句 \[ESQL\] \[SP\]](#)、[FORCE\\_NO\\_SCROLL\\_CURSORS 选项](#) 以及 [FORCE\\_UPDATABLE\\_CURSORS 选项](#)

《系统管理指南，卷 2》的第 1 章 “使用过程和批处理” 中的 “在过程中使用游标”。

《系统管理指南，卷 1》的第 10 章 “事务和版本控制” 中的 “事务中的游标”。

## sp\_iqdatatype 过程

作用

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

**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

可以在不带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的数据类型（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 7-15: *sp\_iqdatatype* 用法示例

| 语法                                             | 输出                                                                                               |
|------------------------------------------------|--------------------------------------------------------------------------------------------------|
| <code>sp_iqdatatype</code>                     | 显示数据库中所有用户定义的数据类型的相关信息                                                                           |
| <code>sp_iqdatatype country_t</code>           | 显示名为 <code>country_t</code> 的用户定义数据类型的相关信息                                                       |
| <code>sp_iqdatatype non_existing_type</code>   | 不返回任何行，因为数据类型 <code>non_existing_type</code> 不存在                                                 |
| <code>sp_iqdatatype NULL, DBA</code>           | 显示 <code>DBA</code> 拥有的所有用户定义数据类型的相关信息                                                           |
| <code>sp_iqdatatype country_t, DBA</code>      | 显示 <code>DBA</code> 拥有的数据类型 <code>country_t</code> 的相关信息                                         |
| <code>sp_iqdatatype rowid</code>               | <code>rowid</code> 是系统定义的数据类型。如果也没有名为 <code>rowid</code> 的用户定义数据类型，则不返回任何行。（缺省情况下，仅返回用户定义的数据类型。） |
| <code>sp_iqdatatype rowid, SYS</code>          | 不返回任何行，因为数据类型 <code>rowid</code> 不是用户定义的数据类型（缺省情况下，仅返回用户定义的数据类型）                                 |
| <code>sp_iqdatatype NULL, NULL, SYSTEM</code>  | 显示所有系统定义的数据类型（由 <code>dbo</code> 或 <code>SYS</code> 拥有）的相关信息                                     |
| <code>sp_iqdatatype rowid, NULL, SYSTEM</code> | 显示有关系统数据类型 <code>rowid</code> 的信息                                                                |
| <code>sp_iqdatatype NULL, NULL, 'ALL'</code>   | 显示有关用户定义数据类型和系统数据类型的信息                                                                           |

## 说明

`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。用户定义的数据类型也称为域。`sp_iqdatatype` 输出中不包括预定义域名。

如果指定一个或多个参数，则将根据指定参数对 `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。如果未指定任何参数，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 过程返回以下列信息：

**表 7-16: sp\_iqdatatype 列**

| 列名        | 说明                                                 |
|-----------|----------------------------------------------------|
| type_name | 数据类型的名称。                                           |
| creator   | 数据类型的所有者。                                          |
| nulls     | Y 表示用户定义的数据类型允许空值； 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。                  |
| scale     | 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" "check"
country_t DBA U 15 0 (NULL) (NULL)

```

另请参见

《参考：语句和选项》中的 [CREATE DOMAIN 语句](#)  
[第 3 章 “SQL 数据类型”](#)

## sp\_iqdbsize 过程

作用

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。

说明

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 7-17: sp\_iqdbsize 列

| 列名               | 说明                                   |
|------------------|--------------------------------------|
| Database         | 数据库文件的路径名称。                          |
| Physical Blocks  | 数据库总大小（以块为单位）。                       |
| KBytes           | 数据库总大小（以千字节为单位）。                     |
| Pages            | 总 IQ 页数。                             |
| Compressed Pages | 压缩的总 IQ 页数（在磁盘上）。Pages 的子集。          |
| NBlocks          | 用于存储表和连接索引中的数据的总 IQ 块大小。             |
| Catalog Blocks   | 用于存储表和连接索引的元数据的总 IQ 块大小。NBlocks 的子集。 |

sp\_iqdbsize 列的说明:

**Database** 当前数据库文件的路径名称。

**Physical Blocks** 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。

有关 IQ 页大小和块大小之间的对应关系，请参见《性能和调优指南》中的第 4 章“管理系统资源”。

**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。

**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于磁盘上所需的总 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。

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_2/demo/iqdemo.db
 1280 522 688 257 1119 18

```

另请参见 《性能和调优指南》的第 4 章“管理系统资源”中的“内存使用概述”中的“指定页大小”。

《系统管理指南，卷 1》的第 5 章“使用数据库对象”中的“使用数据库对象”。

《使用 Sybase IQ Multiplex》附录 A “Multiplex 引用”中的“sp\_iqdbsize 过程”。

## sp\_iqdbspace 过程

**作用** 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。

**说明** 可以使用 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 显示此信息：

**表 7-18: sp\_iqdbspace 列**

| 列名          | 说明                                                                             |
|-------------|--------------------------------------------------------------------------------|
| DBSpaceName | 在 CREATE DBSPACE 语句中指定的 dbspace 的名称。对于使用 CASE RESPECT 创建的数据库，Dbspace 名称不区分大小写。 |
| DBSpaceType | dbspace 的类型（仅限于 MAIN 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（请参见表 7-19）。                                                |
| OkToDrop    | “Y”表示 dbspace 可以删除；否则为“N”。                                                     |

表 7-19 列出了块类型标识符的值。

**表 7-19: 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      |
| 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》。

示例

以下输出显示有关 dbspace 的信息。

```
sp_iqdbspace;
```

**注释** 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 dbspace 示例，您的数据库中可能不包括该示例。

| DB Space Name  | DB Space Type | Writable | Online | Usage | Total Size | Reserve | Num Files | RW Files | Striping | Stripe Size | Blk Types                     | OkTo Drop |
|----------------|---------------|----------|--------|-------|------------|---------|-----------|----------|----------|-------------|-------------------------------|-----------|
| IQ_MAIN        | MAIN          | T        | T      | 55    | 75M        | 200M    | 1         | 1        | T        | 1K          | 1H,<br>5169A,<br>190          | N         |
| IQ_SYSTEM_MAIN | MAIN          | T        | T      | 21    | 300M       | 50M     | 1         | 1        | F        | 8K          | 1H,<br>7648F,<br>32D,<br>128M | N         |

| DB Space Name  | DB Space Type | Writable | Online | Usage | Total Size | Reserve | NumF iles | RW Files | Striping on | Stripe Size | Blk Types    | OkTo Drop |
|----------------|---------------|----------|--------|-------|------------|---------|-----------|----------|-------------|-------------|--------------|-----------|
| IQ_SYSTEM_TEMP | TEMPORARY     | T        | T      | 1     | 100M       | 50M     | 1         | 1        | F           | 8K          | 1H, 64F, 32A | N         |

另请参见

《使用 Sybase IQ Multiplex》附录 A “Multiplex 引用” 中的 “sp\_iqdbspace 过程”。

第 322 页的 “sp\_iqdbspaceinfo 过程”

第 325 页的 “sp\_iqdbspaceobjectinfo 过程”

第 351 页的 “sp\_iqindexinfo 过程”

## sp\_iqdbspaceinfo 过程

作用

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

**dbspace\_name** 如果指定，则 sp\_iqdbspaceinfo 显示在指定的 dbspace 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 显示有关数据库中所有用户的表和连接索引的信息。

**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 存储过程支持用于解释 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 dbspace 的信息。

说明

`sp_iqdbspaceinfo` 为 DBA 显示每个 `dbspace` 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。输出将按 `dbspace_name`、`object_name` 和 `owner_name` 排序。

表 7-20: `sp_iqdbspaceinfo` 列

| 列名                             | 说明                                                                       |
|--------------------------------|--------------------------------------------------------------------------|
| <code>dbspace_name</code>      | <code>dbspace</code> 的名称。                                                |
| <code>object_type</code>       | 对象的类型（仅限于 <code>table</code> 或 <code>joinindex</code> ）。                 |
| <code>owner</code>             | 对象所有者的名称。                                                                |
| <code>object_name</code>       | 位于 <code>dbspace</code> 上的对象（仅限于表和连接索引的对象）的名称。                           |
| <code>object_id</code>         | 对象的全局对象 ID。                                                              |
| <code>id</code>                | 对象的表 ID 或连接索引 ID。                                                        |
| <code>column</code>            | 给定 <code>dbspace</code> 上的列存储空间的大小。                                      |
| <code>index</code>             | 给定 <code>dbspace</code> 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）。 |
| <code>metadata</code>          | 给定 <code>dbspace</code> 上元数据对象的存储空间大小。                                   |
| <code>primary_key</code>       | 给定 <code>dbspace</code> 上主键相关对象的存储空间大小。                                  |
| <code>unique_constraint</code> | 给定 <code>dbspace</code> 上唯一约束相关对象的存储空间大小。                                |
| <code>foreign_key</code>       | 给定 <code>dbspace</code> 上外键相关对象的存储空间大小。                                  |
| <code>dbspace_online</code>    | 表示 <code>dbspace</code> 处于联机状态 (Y) 还是脱机状态 (N)。                           |

**注释**

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: 只读数据库不允许修改。应出现此行为。在其它存储过程（例如 sp_iqdbspace、sp_iqfile、sp_iqdbspaceobjectinfo 或 sp_iqobjectinfo）上不发生此错误。`

示例

显示数据库的所有 **dbspace** 内所有表中的所有对象和子对象的大小：

**注释** 下面的三个示例演示 **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 **dbspace** 示例，您的数据库中可能不包括该示例。

sp\_iqdbspaceinfo

| dbspace_name | object_type | owner  | object_name     | object_id | id  | columns |
|--------------|-------------|--------|-----------------|-----------|-----|---------|
| iq_main      | table       | DBA    | empl            | 3689      | 741 | 96K     |
| iq_main      | table       | DBA    | iq_dummy        | 3686      | 740 | 24K     |
| iq_main      | table       | DBA    | sale            | 3698      | 742 | 96K     |
| iq_main      | table       | GROUPO | Contacts        | 3538      | 732 | 288K    |
| iq_main      | table       | GROUPO | Customers       | 3515      | 731 | 240K    |
| iq_main      | table       | GROUPO | Departments     | 3632      | 738 | 72K     |
| iq_main      | table       | GROUPO | Employees       | 3641      | 739 | 408K    |
| iq_main      | table       | GROUPO | FinancialCodes  | 3612      | 736 | 72K     |
| iq_main      | table       | GROUPO | FinancialData   | 3621      | 737 | 96K     |
| iq_main      | table       | GROUPO | Products        | 3593      | 735 | 272K    |
| iq_main      | table       | GROUPO | SalesOrderItems | 3580      | 734 | 120K    |
| iq_main      | table       | GROUPO | 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 **dbspace** 中指定用户拥有的所有对象和子对象的大小：

sp\_iqdbspaceinfo iq\_main,GROUPO

| dbspace_name | object_type | owner  | object_name     | object_id | id  | columns |
|--------------|-------------|--------|-----------------|-----------|-----|---------|
| iq_main      | table       | GROUPO | Contacts        | 3538      | 732 | 288K    |
| iq_main      | table       | GROUPO | Customers       | 3515      | 731 | 240K    |
| iq_main      | table       | GROUPO | Departments     | 3632      | 738 | 72K     |
| iq_main      | table       | GROUPO | Employees       | 3641      | 739 | 408K    |
| iq_main      | table       | GROUPO | FinancialCodes  | 3612      | 736 | 72K     |
| iq_main      | table       | GROUPO | FinancialData   | 3621      | 737 | 96K     |
| iq_main      | table       | GROUPO | Products        | 3593      | 735 | 272K    |
| iq_main      | table       | GROUPO | SalesOrderItems | 3580      | 734 | 120K    |

|         |          |             |                   |             |                |     |      |
|---------|----------|-------------|-------------------|-------------|----------------|-----|------|
| iq_main | table    |             | GROUPO            | SalesOrders | 3565           | 733 | 144K |
| indexes | metadata | primary_key | unique_constraint | foreign_key | dbspace_online |     |      |
| 0B      | 5.45M    | 24K         | 0B                | 48K         | Y              |     |      |
| 48K     | 4.63M    | 24K         | 0B                | 0B          | Y              |     |      |
| 0B      | 1.78M    | 24K         | 0B                | 48K         | Y              |     |      |
| 0B      | 8.03M    | 24K         | 0B                | 48K         | Y              |     |      |
| 0B      | 1.53M    | 24K         | 0B                | 0B          | Y              |     |      |
| 0B      | 2.19M    | 24K         | 0B                | 48K         | Y              |     |      |
| 192K    | 4.67M    | 24K         | 0B                | 0B          | Y              |     |      |
| 0B      | 2.7M     | 24K         | 0B                | 104K        | Y              |     |      |
| 0B      | 3.35M    | 24K         | 0B                | 144K        | Y              |     |      |

显示数据库的指定 `dbspace` 中指定用户拥有的指定对象及其子对象的大小：

```
sp_iqdbspaceinfo iq_main, GROUPO, Departments
```

|              |             |             |                   |             |                |         |
|--------------|-------------|-------------|-------------------|-------------|----------------|---------|
| dbspace_name | object_type | owner       | object_name       | object_id   | id             | columns |
| iq_main      | table       | GROUPO      | Departments       | 3632        | 738            | 72K     |
| indexes      | metadata    | primary_key | unique_constraint | foreign_key | dbspace_online |         |
| 0B           | 1.78M       | 24K         | 0B                | 48K         | Y              |         |

另请参见

[第 320 页的“sp\\_iqdbspace 过程”](#)和 [第 351 页的“sp\\_iqindexinfo 过程”](#)。

《使用 Sybase IQ Multiplex》附录 A “Multiplex 引用”中的“[sp\\_iqfile 过程](#)”。

## sp\_iqdbspaceobjectinfo 过程

**作用** 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** **dbspace-name** 如果指定，则 `sp_iqdbspaceobjectinfo` 仅显示指定 `dbspace` 的输出。否则，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_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信息。

**object-name** 表或连接索引的名称。如果未指定，则 `sp_iqdbspaceobjectinfo` 显示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，并可能提供独立于其它参数的值的任何参数。

`sp_iqdbspaceobjectinfo` 存储过程支持用于解释 `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 `dbspace` 的信息。

说明

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 7-21: `sp_iqdbspaceobjectinfo` 列**

| 列名                             | 说明                                                                                                                                                                                                           |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>dbspace_name</code>      | <code>dbspace</code> 的名称。                                                                                                                                                                                    |
| <code>dbspace_id</code>        | <code>dbspace</code> 的标识符。                                                                                                                                                                                   |
| <code>object_type</code>       | 表或连接索引。                                                                                                                                                                                                      |
| <code>owner</code>             | 对象所有者的名称。                                                                                                                                                                                                    |
| <code>object_name</code>       | 位于 <code>dbspace</code> 上的对象（仅限于表和连接索引的对象）的名称。                                                                                                                                                               |
| <code>object_id</code>         | 对象的全局对象 ID。                                                                                                                                                                                                  |
| <code>id</code>                | 对象的表 ID 或连接索引 ID。                                                                                                                                                                                            |
| <code>column</code>            | 位于给定 <code>dbspace</code> 上的表列的数量。如果一列或列分区之一位于 <code>dbspace</code> 上，则认为该 <code>dbspace</code> 上存在该列或列分区。结果以 <code>n/N</code> （超出表中总列数 <code>N</code> 之外的 <code>n</code> 在给定的 <code>dbspace</code> 上）的格式显示。 |
| <code>index</code>             | 位于给定 <code>dbspace</code> 上的表中用户定义的索引数。以 <code>n/N</code> （超出表中总索引数 <code>N</code> 之外的 <code>n</code> 在给定的 <code>dbspace</code> 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 <code>FP</code> 索引和 <code>HG</code> 索引。       |
| <code>metadata</code>          | 布尔字段 (Y/N)，表示是否子对象的元数据信息也位于此 <code>dbspace</code> 上。                                                                                                                                                         |
| <code>primary_key</code>       | 布尔字段 (1/0)，表示是否表的主键（如果有）也位于此 <code>dbspace</code> 上。                                                                                                                                                         |
| <code>unique_constraint</code> | 位于给定 <code>dbspace</code> 上的表中唯一约束的数量。以 <code>n/N</code> （超出表中总唯一约束数 <code>N</code> 之外的 <code>n</code> 在给定的 <code>dbspace</code> 上）的格式显示。                                                                    |
| <code>foreign_key</code>       | 位于给定 <code>dbspace</code> 上的表中外键的数量。以 <code>n/N</code> （超出表中总外键数 <code>N</code> 之外的 <code>n</code> 在给定的 <code>dbspace</code> 上）的格式显示。                                                                        |
| <code>partition</code>         | 位于给定 <code>dbspace</code> 上的表中分区区的数量。以 <code>n/N</code> （超出表中总分区数 <code>N</code> 之外的 <code>n</code> 在给定的 <code>dbspace</code> 上）的格式显示。                                                                       |

示例

**注释** 下面的两个示例演示 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 dbspace 示例，您的数据库中可能不包括该示例。

---

显示有关数据库中某个特定 **dbspace** 的信息:

sp\_iqdbspaceobjectinfo iq\_main

| dbspace_name | dbspace_id | object_type | owner  | object_name     | object_id | id  | columns |
|--------------|------------|-------------|--------|-----------------|-----------|-----|---------|
| iq_main      | 16387      | table       | DBA    | empl            | 3689      | 741 | 4/4     |
| iq_main      | 16387      | table       | DBA    | iq_dummy        | 3686      | 740 | 1/1     |
| iq_main      | 16387      | table       | DBA    | sale            | 3698      | 742 | 4/4     |
| iq_main      | 16387      | table       | GROUPO | Contacts        | 3538      | 732 | 12/12   |
| iq_main      | 16387      | table       | GROUPO | Customers       | 3515      | 731 | 10/10   |
| iq_main      | 16387      | table       | GROUPO | Departments     | 3632      | 738 | 3/3     |
| iq_main      | 16387      | table       | GROUPO | Employees       | 3641      | 739 | 21/21   |
| iq_main      | 16387      | table       | GROUPO | FinancialCodes  | 3612      | 736 | 3/3     |
| iq_main      | 16387      | table       | GROUPO | FinancialData   | 3621      | 737 | 4/4     |
| iq_main      | 16387      | table       | GROUPO | Products        | 3593      | 735 | 8/8     |
| iq_main      | 16387      | table       | GROUPO | SalesOrderItems | 3580      | 734 | 5/5     |
| iq_main      | 16387      | table       | GROUPO | SalesOrders     | 3565      | 733 | 6/6     |

| indexes | metadata | primary_key | unique_constraint | foreign_key | partitions |
|---------|----------|-------------|-------------------|-------------|------------|
| 0/0     | Y        | 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 **dbspace** 中指定用户拥有的对象的信息:

sp\_iqdbspaceobjectinfo iq\_main,GROUPO

| dbspace_name | dbspace_id | object_type | owner  | object_name     | object_id | id  | columns |
|--------------|------------|-------------|--------|-----------------|-----------|-----|---------|
| iq_main      | 16387      | table       | GROUPO | Contacts        | 3538      | 732 | 2/12    |
| iq_main      | 16387      | table       | GROUPO | Customers       | 3515      | 731 | 10/10   |
| iq_main      | 16387      | table       | GROUPO | Departments     | 3632      | 738 | 3/3     |
| iq_main      | 16387      | table       | GROUPO | Employees       | 3641      | 739 | 21/21   |
| iq_main      | 16387      | table       | GROUPO | FinancialCodes  | 3612      | 736 | 3/3     |
| iq_main      | 16387      | table       | GROUPO | FinancialData   | 3621      | 737 | 4/4     |
| iq_main      | 16387      | table       | GROUPO | Products        | 3593      | 735 | 8/8     |
| iq_main      | 16387      | table       | GROUPO | SalesOrderItems | 3580      | 734 | 5/5     |
| iq_main      | 16387      | table       | GROUPO | SalesOrders     | 3565      | 733 | 6/6     |

| indexes | metadata | primary_key | unique_constraint | foreign_key | partitions |
|---------|----------|-------------|-------------------|-------------|------------|
| 0/0     | Y        | 1           | 0/0               | 1/1         | 0/0        |
| 1/1     | Y        | 1           | 0/0               | 0/0         | 0/0        |
| 0/0     | Y        | 1           | 0/0               | 1/1         | 0/0        |

|     |   |   |     |     |     |
|-----|---|---|-----|-----|-----|
| 0/0 | Y | 1 | 0/0 | 1/1 | 0/0 |
| 0/0 | Y | 1 | 0/0 | 0/0 | 0/0 |
| 0/0 | Y | 1 | 0/0 | 1/1 | 0/0 |
| 4/4 | Y | 1 | 0/0 | 0/0 | 0/0 |
| 0/0 | Y | 1 | 0/0 | 2/2 | 0/0 |
| 0/0 | Y | 1 | 0/0 | 3/3 | 0/0 |

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;
```

另请参见

第 320 页的“`sp_iqdbspace` 过程”和第 351 页的“`sp_iqindexinfo` 过程”

## sp\_iqdbstatistics 过程

**作用** 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。

**说明** 显示在最近一次执行 `sp_iqcheckdb` 时收集的数据库统计信息。

**示例** 下面的示例显示 `sp_iqdbstatistics` 产生的输出。对于此示例，最近一次执行 `sp_iqcheckdb` 的形式是 `sp_iqcheckdb 'allocation database'` 命令。

| DB Statistics               | Value           | Flags |
|-----------------------------|-----------------|-------|
| =====                       | =====           | ===== |
| DBCC Allocation Mode Report |                 |       |
| =====                       | =====           | ===== |
| ** DBCC Status              | 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 | |
=====|=====|=====

```

另请参见 [有关 sp\\_iqcheckdb 的使用和 sp\\_iqcheckdb 输出解释的详细信息](#)，请参见《系统管理指南，卷 1》中的第 13 章“系统恢复与数据库修复”。

### sp\_iqdroplogin 过程

作用 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   | <b>userid</b> 要删除的用户的用户 ID。                                                                                                                  |
| 说明   | <b>sp_iqdroplogin</b> 删除指定用户。                                                                                                                |
| 示例   | 以下存储过程调用将删除用户 <b>rose</b> 。<br><pre>sp_iqdroplogin 'rose' sp_iqdroplogin rose call sp_iqdroplogin ('rose')</pre>                             |
| 另请参见 | 第 287 页的“ <a href="#">sp_iqaddlogin 过程</a> ”<br>《参考：语句和选项》中的 <a href="#">REVOKE 语句</a><br>《系统管理指南，卷 1》中的第 8 章“ <a href="#">管理用户 ID 和权限</a> ” |

## sp\_iqemptyfile 过程

|    |                                                                                                                                                                                            |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作用 | 清空 <b>dbspace</b> 文件并将该文件中的对象移动到另一个可用的读写 <b>dbpace</b> 文件。                                                                                                                                 |
| 语法 | <b>sp_iqemptyfile</b> ( <i>logical-file--name</i> )                                                                                                                                        |
| 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。                                                                                                                                          |
| 说明 | <b>sp_iqemptyfile</b> 清空 <b>dbspace</b> 文件。 <b>dbspace</b> 必须是只读的才可以执行 <b>sp_iqemptyfile</b> 过程。该过程将文件中的对象移动到另一个可用的读写 <b>dbspace</b> 文件中。如果没有其它读写 <b>dbspace</b> 文件可用，则 Sybase IQ 将显示错误消息。 |
|    | <hr/> <b>注释</b> 在 Multiplex 环境中，只能在事务协调器上运行 <b>sp_iqemptyfile</b> 。<br>必须有一个读写 <b>dbspace</b> 可用才能成功执行该过程。 <hr/>                                                                           |
| 示例 | 以下示例清空 <b>dbspace dbspace1</b> ：<br><pre>sp_iqemptyfile 'dbspace1'</pre>                                                                                                                   |

## sp\_iqestjoin 过程

- 作用                                 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。
- 说明                                 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 7-22 列出了 sp\_iqestjoin 的参数。

**表 7-22: sp\_iqestjoin 的参数**

| 名称                  | 数据类型      | 说明                                                                |
|---------------------|-----------|-------------------------------------------------------------------|
| <i>table1_name</i>  | char(256) | 连接中第一个表的名称。                                                       |
| <i>table1_row_#</i> | int       | 参与连接的第一个表中的行数。                                                    |
| <i>table2_name</i>  | char(256) | 连接中第二个表的名称。                                                       |
| <i>table2_row_#</i> | int       | 参与连接的第二个表中的行数。                                                    |
| <i>relation</i>     | 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”。 |
| <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 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
Columns:
8
Width:
223
Table2:SalesOrders
Rows:15000000
Columns:
9
```

| Cases       | Indexsize | Create time | Msg |
|-------------|-----------|-------------|-----|
| Width:      |           |             |     |
| 134         |           |             |     |
| IQpagesize: |           |             |     |
| 65536       |           |             |     |
| Min Case    | 48001024  | 3h0m/CPU    |     |
| Max Case    | 95449088  | 9h6m/CPU    |     |
| Avg Case    | 70496256  | 5h53m/CPU   |     |

## sp\_iqestdbspaces 过程

**作用** 估计给定索引总大小所需的 dbspace 的数量和大小。

**语法** `sp_iqestdbspaces ( 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。

**说明** 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 7-23 列出了 sp\_iqestdbspaces 的参数。

**表 7-23: sp\_iqestdbspaces 的参数**

| 名称                      | 数据类型        | 说明                                                            |
|-------------------------|-------------|---------------------------------------------------------------|
| <i>db_size_in_bytes</i> | decimal(16) | 数据库的大小（以字节为单位）。                                               |
| <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）。 |
| <i>min_#_of_bytes</i>   | int         | 每个 dbspace 段的最小字节数。缺省值为 20,000,000 (20MB)。                    |
| <i>max_#_of_bytes</i>   | int         | 每个 dbspace 段的最大字节数。缺省值为 2,146,304,000 (2.146GB)。              |

**用法** sp\_iqestdbspaces 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：

**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 dbspace 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 dbspace 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 dspace 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 dspace 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。

❖ **将 sp\_iquestdbspaces 与其它系统存储过程一起使用**

- 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\_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 dspace 段，因为这样可以在稍后删除未使用的段。

示例

```
sp_iquestdbspaces 12000000000, 65536, 500000000, 2146304000
```

| dbspace files | Type  | Size       | Msg |
|---------------|-------|------------|-----|
| 1             | min   | 2146304000 |     |
| 2             | min   | 2146304000 |     |
| 3             | min   | 507392000  |     |
| 4             | avg   | 2146304000 |     |
| 5             | max   | 2053697536 |     |
| 6             | spare | 1200001024 |     |

本例估计的是一个 12GB 数据库所需的 dspace 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 dspace 段。

## sp\_iqestspace 过程

|    |                                                                                                            |
|----|------------------------------------------------------------------------------------------------------------|
| 作用 | 根据表中的行数估计创建索引所需的空间大小。                                                                                      |
| 语法 | <b>sp_iqestspace</b> ( <i>table_name</i> , #_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。                                                          |
| 说明 | 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）。表 7-24 列出了 sp_iqestspace 的参数。 |

**表 7-24: sp\_iqestspace 的参数**

| 名称                  | 数据类型      | 说明                                                           |
|---------------------|-----------|--------------------------------------------------------------|
| <i>table_name</i>   | char(256) | 表的名称                                                         |
| <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

|    |                                                                                        |
|----|----------------------------------------------------------------------------------------|
| 作用 | 显示有关系统事件和用户定义的事件的信息。                                                                   |
| 语法 | <b>sp_iqevent</b> [ <i>event-name</i> ], [ <i>event-owner</i> ], [ <i>event-type</i> ] |
| 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 | <p><b>event-name</b> 事件的名称。</p> <p><b>event-owner</b> 事件的所有者。</p> <p><b>event-type</b> 事件的类型。允许的值如下：</p> |
|----|----------------------------------------------------------------------------------------------------------|

- **SYSTEM**: 只显示有关系统事件（由用户 SYS 或 dbo 拥有的事件）的信息
- **ALL**: 显示有关用户事件和系统事件的信息
- 其它任何值: 显示有关用户事件的信息

可以在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（由 dbo 或 SYS 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 7-25: sp\_iqevent 用法示例**

| 语法                                                  | 输出                                                                                                              |
|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <code>sp_iqevent</code>                             | 显示有关数据库中所有用户事件的信息                                                                                               |
| <code>sp_iqevent e1</code>                          | 显示与事件 <code>e1</code> 有关的信息                                                                                     |
| <code>sp_iqevent non_existing_event</code>          | 不返回任何行，因为事件 <code>non_existing_event</code> 不存在                                                                 |
| <code>sp_iqevent NULL, DBA</code>                   | 显示有关 <code>DBA</code> 拥有的所有事件的信息                                                                                |
| <code>sp_iqevent e1, DBA</code>                     | 显示有关 <code>DBA</code> 拥有的事件 <code>e1</code> 的信息                                                                 |
| <code>sp_iqevent ev_iqbegintxn</code>               | <code>ev_iqbegintxn</code> 是系统定义的事件。如果没有同样名为 <code>ev_iqbegintxn</code> 的任何用户定义的事件，则不会返回任何行。（缺省情况下，只返回用户定义的事件。） |
| <code>sp_iqevent ev_iqbegintxn, dbo</code>          | 不返回任何行，因为事件 <code>ev_iqbegintxn</code> 不是用户事件（缺省情况下，只返回用户事件）                                                    |
| <code>sp_iqevent NULL, NULL, SYSTEM</code>          | 显示有关所有系统事件（由 <code>dbo</code> 或 <code>SYS</code> 拥有的事件）的信息                                                      |
| <code>sp_iqevent ev_iqbegintxn, NULL, SYSTEM</code> | 显示有关系统事件 <code>ev_iqbegintxn</code> 的信息                                                                         |
| <code>sp_iqevent ev_iqbegintxn, dbo, ALL</code>     | 显示有关 <code>dbo</code> 拥有的系统事件 <code>ev_iqbegintxn</code> 的信息                                                    |

说明

`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，如果指定了 `event-name`，则只显示有关指定事件的信息。如果指定了 `event-owner`，`sp_iqevent` 将只返回有关指定所有者拥有的事件的信息。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 7-26: sp\_iqevent 列**

| 列名                       | 说明                                             |
|--------------------------|------------------------------------------------|
| <code>event_name</code>  | 事件的名称                                          |
| <code>event_owner</code> | 事件的所有者                                         |
| <code>event_type</code>  | 对于系统事件，是 <code>SYSEVENTTYPE</code> 系统表中列出的事件类型 |
| <code>enabled</code>     | 指明是否允许触发事件 (Y/N)                               |
| <code>action</code>      | 事件处理程序定义                                       |
| <code>condition</code>   | 用于控制事件处理程序触发的 <code>WHERE</code> 条件            |

| 列名       | 说明                                                                                                        |
|----------|-----------------------------------------------------------------------------------------------------------|
| location | 允许事件触发的位置： <ul style="list-style-type: none"> <li>• C = 统一</li> <li>• R = 远程</li> <li>• A = 全部</li> </ul> |
| 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<br>dbo.sp_iqlog... |
| ev_iqmpxcompact | dbo         | (NULL)         | N       | begin Declare<br>_Catalog...  |

  

| condition | location | remarks |
|-----------|----------|---------|
| (NULL)    | A        | (NULL)  |
| (NULL)    | A        | (NULL)  |

## 另请参见

《参考：语句和选项》中的 [CREATE EVENT 语句](#)

《系统管理指南，卷 2》中的第 6 章 “使用日程表和事件自动完成任务”

## sp\_iqfile 过程

- 作用                                    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。
- 说明                                    sp\_iqfile 显示 dbspace 中每个 dbfile 中的数据的用法、属性和类型。可以使用这些信息确定是否必须移动数据，以及对于已移动的数据是否已释放旧版本。

sp\_iqfile 将显示以下信息：

**表 7-27: sp\_iqfile 列**

| 列名          | 说明                                                                             |
|-------------|--------------------------------------------------------------------------------|
| DBSpaceName | 在 CREATE DBSPACE 语句中指定的 dbspace 的名称。对于使用 CASE RESPECT 创建的数据库，Dbspace 名称不区分大小写。 |
| DBFileName  | 逻辑文件名。                                                                         |
| Path        | 物理文件或裸分区的位置。                                                                   |
| SegmentType | dbspace 的类型（MAIN 或 TEMPORARY）。                                                 |
| RWMode      | dbspace 的模式：只写 (RW) 或只读 (RO)。                                                  |
| Online      | T（联机）或 F（脱机）。                                                                  |
| Usage       | 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（有关标识符的值，请参见表 7-28）。                                        |
| 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”。                                                            |

表 7-28 列出了块类型标识符的值。

**表 7-28: 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 dbspace 中文件的信息：

```
sp_iqfile;
```

| DBSpace Name   | DBFile Name    | Path                             | Segment Type | RWMode |
|----------------|----------------|----------------------------------|--------------|--------|
| IQ_SYSTEM_MAIN | IQ_SYSTEM_MAIN | /sunopt/users/user1/iqdemo.iq    | MAIN         | RW     |
| IQ_SYSTEM_TEMP | IQ_SYSTEM_TEMP | /sunopt/users/user1/iqdemo.iqtmp | TEMPORARY    | RW     |

  

| Online | Usage | DBFileSize | Reserve | Stripesize | BlkTypes              | FirstBlk | LastBlk | OkToDrop |
|--------|-------|------------|---------|------------|-----------------------|----------|---------|----------|
| T      | 21    | 300M       | 50M     | 8K         | 1H,7648F,<br>32D,128M | 1        | 38400   | N        |
| T      | 1     | 100M       | 50M     | 8K         | 1H,64F,16A            | 1        | 12800   | N        |

另请参见

第 322 页的 “sp\_iqdbspaceinfo 过程” 和第 351 页的 “sp\_iqindexinfo 过程”

《Sybase IQ 系统管理指南》中的第 5 章 “使用数据库对象”

## sp\_iqhelp 过程

- 作用                    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                    **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。

**表 7-29: sp\_iqhelp 的 obj-category 参数值**

| object-type 参数 | 指定                 |
|----------------|--------------------|
| “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 |

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。

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”。

为 obj-category 参数加上单引号或双引号（值为 NULL 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 (未找到给定说明的对象)” 错误。

**表 7-30: sp\_iqhelp 用法示例**

| 语法                                              | 输出                                                                                   |
|-------------------------------------------------|--------------------------------------------------------------------------------------|
| <code>sp_iqhelp</code>                          | 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                                                  |
| <code>sp_iqhelp t1, u1, "table"</code>          | 显示用户 <code>u1</code> 拥有的表 <code>t1</code> 以及与 <code>t1</code> 关联的列、索引和约束的相关信息        |
| <code>sp_iqhelp NULL, u1, "view"</code>         | 显示用户 <code>u1</code> 拥有的视图 <code>v1</code> 以及与 <code>v1</code> 关联的列的相关信息             |
| <code>sp_iqhelp sp2</code>                      | 显示过程 <code>sp2</code> 及 <code>sp2</code> 的参数的相关信息                                    |
| <code>sp_iqhelp e1</code>                       | 显示与事件 <code>e1</code> 有关的信息                                                          |
| <code>sp_iqhelp dt1</code>                      | 显示有关数据类型 <code>dt1</code> 的信息                                                        |
| <code>sp_iqhelp NULL, NULL, NULL, SYSTEM</code> | 显示有关所有系统对象（由 <code>dbo</code> 或 <code>SYS</code> 拥有的对象）的摘要信息                         |
| <code>sp_iqhelp non_existing_obj</code>         | 返回错误 “Object ‘non_existing_obj’ not found”，因为对象 <code>non_existing_obj</code> 不存在    |
| <code>sp_iqhelp NULL, non_existing_user</code>  | 返回错误 “User ‘non_existing_user’ not found”，因为用户 <code>non_existing_user</code> 不存在    |
| <code>sp_iqhelp t1, NULL, "apple"</code>        | 返回错误 “Invalid object category ‘apple’ ”，因为 “apple” 不是 <code>obj-category</code> 的允许值 |
| <code>sp_iqhelp t1, NULL, NULL, "USER"</code>   | 返回错误 “Invalid object type 'USER'”，因为 “USER” 不是 <code>obj-type</code> 的允许值            |

#### 说明

`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，如果指定了 `obj-name`，则只显示有关指定对象的信息。如果指定了 `obj-owner`，`sp_iqhelp` 将只返回有关指定所有者拥有的对象的信息。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 7-31: sp\_iqhelp 摘要信息**

| 对象类型            | 显示的列                                                                                          |
|-----------------|-----------------------------------------------------------------------------------------------|
| 基表              | table_name、table_owner、server_type、location、table_constraints、remarks                         |
| 视图              | view_name、view_creator、view_def、server_type、location、remarks                                  |
| 存储过程            | proc_name、proc_creator、proc_defn、replicate、srvid、remarks                                      |
| 函数              | proc_name、proc_creator、proc_defn、replicate、remarks                                            |
| 事件              | event_name、event_creator、enabled、location、event_type、action、external_action、condition、remarks |
| 系统数据类型和用户定义数据类型 | type_name、creator、nulls、width、scale、default、check                                             |

如果有一个对象与指定的 sp\_iqhelp 参数匹配，sp\_iqhelp 将显示有关该对象的详细信息。

**表 7-32: sp\_iqhelp 详细信息**

| 对象类型 | 说明                                      | 列                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 表    | 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 | <ul style="list-style-type: none"> <li>• 表列: table_name、table_owner、server_type、location、table_constraints、remarks</li> <li>• 列列: column_name、domain_name、width、scale、nulls、default、check、pkey、user_type、cardinality、est_cardinality、remarks</li> <li>• 索引列: index_name、column_name、index_type、unique_index、location、remarks</li> <li>• 约束列: constraint_name（角色）、column_name、index_name、constraint_type、foreigntable_name、foreigntable_owner、foreigncolumn_name、foreignindex_name、location</li> <li>• 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</li> </ul> |
| 视图   | 显示有关指定视图及其列的信息                          | <ul style="list-style-type: none"> <li>• 视图列: view_name、view_creator、view_def、server_type、location、remarks</li> <li>• 列列: column_name、domain_name、width、scale、nulls、default、check、pkey、user_type、cardinality、est_cardinality、remarks</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                        |

| 对象类型 | 说明              | 列                                                                                                                                                                         |
|------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 存储过程 | 显示有关指定过程及其参数的信息 | <ul style="list-style-type: none"> <li>过程列: proc_name、proc_creator、proc_defn、replicate、svid、remarks</li> <li>参数列: parameter_name、type、width、scale、default、mode</li> </ul> |
| 函数   | 显示有关指定函数及其参数的信息 | <ul style="list-style-type: none"> <li>函数列: proc_name、proc_creator、proc_defn、replicate、svid、remarks</li> <li>参数列: parameter_name、type、width、scale、default、mode</li> </ul> |
| 事件   | 显示有关指定事件的信息     | <ul style="list-style-type: none"> <li>事件列: event_name、event_creator、enabled、location、event_type、action、external_action、condition、remarks</li> </ul>                      |
| 数据类型 | 显示有关指定数据类型的信息   | <ul style="list-style-type: none"> <li>数据类型列: type_name、creator、nulls、width、scale、default、check</li> </ul>                                                                |

---

**注释** 系统过程的过程定义 (proc-defn) 在视图中被加密和隐藏。

---

有关表 7-32 中列出的各个输出列的说明，请参见以下存储过程的说明：

- 表：第 395 页的“[sp\\_iqtable 过程](#)”
- 列：第 303 页的“[sp\\_iqcolumn 过程](#)”
- 索引：第 346 页的“[sp\\_iqindex 和 sp\\_iqindex\\_alt 过程](#)”
- 约束：第 310 页的“[sp\\_iqconstraint 过程](#)”
- 连接索引：第 356 页的“[sp\\_iqjoinindex 过程](#)”
- 视图：第 407 页的“[sp\\_iqview 过程](#)”以及 Adaptive Server Enterprise 目录存储过程 `sp_columns`（用于视图列）
- 存储过程和函数：第 370 页的“[sp\\_iqprocedure 过程](#)”和第 372 页的“[sp\\_iqprocparm 过程](#)”（用于过程参数）
- 事件：第 335 页的“[sp\\_iqevent 过程](#)”
- 数据类型：第 316 页的“[sp\\_iqdatatype 过程](#)”

**Adaptive Server Enterprise 兼容性** Sybase IQ 的 `sp_iqhelp` 存储过程与 Adaptive Server Enterprise 的 `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 在体系结构方面存在一些差异。在 Adaptive Server 中，所有对象（表、视图、存储过程、日志、规则、缺省值、触发器、检查约束、参照约束和临时对象）都存储在 SYSOBJECTS 系统表中，并且位于同一命名空间中。 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 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 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。

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
```

| Table_name | Table_owner | Server_type | Location | dbspace_id | isPartitioned |
|------------|-------------|-------------|----------|------------|---------------|
| sale       | DBA         | IQ          | Main     | 16387      | N             |

```
Remarks table_constraints
=====
(NULL) (NULL)
```

```
column_name domain_name width scale nulls default cardinality
=====
```

| 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
=====
```

| 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
=====
```

| index_name          | column_name | index_type | unique_index | location |
|---------------------|-------------|------------|--------------|----------|
| ASIQ_IDX_T463_C2_FP | month_num   | FP         | N            | Main     |

|                     |         |    |   |      |
|---------------------|---------|----|---|------|
| ASIQ_IDX_T463_C1_FP | prod_id | FP | N | Main |
| ASIQ_IDX_T463_C3_FP | rep_id  | FP | N | Main |
| ASIQ_IDX_T463_C4_FP | sales   | FP | N | Main |

```

remarks
=====
(NULL)
(NULL)
(NULL)
(NULL)

```

显示有关过程 `sp_customer_list` 的详细信息:

```

sp_iquhelp sp_customer_list
proc_name proc_owner proc_defn
===== =====
sp_customer_list DBA create procedure DBA.sp_customer_list()
 result(id integer company_name char(35))
 begin
 select id company_name from Customers
 end

```

```

replicate srvid remarks
===== =====
N (NULL) (NULL)

```

| parm_name    | parm_type | parm_mode | domain_name | width | scale |
|--------------|-----------|-----------|-------------|-------|-------|
| =====        | =====     | =====     | =====       | ===== | ===== |
| id           | result    | out       | integer     | 4     | 0     |
| company_name | result    | out       | char        | 35    | 0     |

```

default
=====
(NULL)

```

## sp\_iqindex 和 sp\_iqindex\_alt 过程

|      |                                                                                                                                                                                                                                                                        |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作用   | 列出有关索引的信息。                                                                                                                                                                                                                                                             |
| 语法 1 | <code>sp_iqindex ( [ table_name ],[column_name ],[table_owner ] )</code>                                                                                                                                                                                               |
| 语法 2 | <code>sp_iqindex [table_name='tablename'],<br/>[column_name='columnname'],[table_owner='tableowner']</code>                                                                                                                                                            |
| 语法 3 | <code>sp_iqindex_alt ( [ table_name ],[column_name ],[table_owner ] )</code>                                                                                                                                                                                           |
| 语法 4 | <code>sp_iqindex_alt [table_name='tablename'],<br/>[column_name='columnname'],[table_owner='tableowner']</code>                                                                                                                                                        |
| 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   | <p><b>语法 1</b> 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，<code>sp_iqindex NULL,NULL,DBA</code> 和 <code>sp_iqindex Departments,NULL,DBA</code>。</p> <p><b>语法 2</b> 您可以按任意顺序指定参数。将其用单引号引起来。</p> <p><b>语法 3 和 4</b> 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。</p> |
| 说明   | 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 7-33: sp\_iqindex 和 sp\_iqindex\_alt 列**

| 列名           | 说明                                              |
|--------------|-------------------------------------------------|
| 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 总是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。

示例

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 上的多列索引。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      | State       | 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 | location | 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)  |

另请参见

《参考：语句和选项》的第 2 章 “数据库选项” 中的 “FP\_LOOKUP\_SIZE 选项”、“INDEX\_ADVISOR 选项” 和 “MINIMIZE\_STORAGE 选项”。

《系统管理指南，卷 1》中的第 6 章 “使用 Sybase IQ 索引”。

## sp\_iqindexadvice 过程

|    |                                                                           |
|----|---------------------------------------------------------------------------|
| 作用 | 显示存储的索引建议消息。可以选择清除建议存储。                                                   |
| 语法 | <code>sp_iqindexadvice ([ resetflag ])</code>                             |
| 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 | <b>resetflag</b> 允许调用者清除索引建议存储。如果 <i>resetflag</i> 不为零，则会在检索完最后一行后删除所有建议。 |
| 说明 | 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。                       |

表 7-34 列出了 INDEX\_ADVISOR 列。

**表 7-34: sp\_iqindexadvice 列**

| 列名     | 说明             |
|--------|----------------|
| 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

表 7-35 显示了 `sp_iqindexadvice` 过程的输出样本。

**表 7-35: `sp_iqindexadvice` 输出**

| Advice                                                                 | NInst | LastDT                  |
|------------------------------------------------------------------------|-------|-------------------------|
| Add a CMP index on DBA.tb (c2, c3)<br>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<br>have mismatched data types | 911   | 2009-02-25 20:59:01.000 |

另请参见

第 305 页的“`sp_iqcolumnuse` 过程”、第 320 页的“`sp_iqdbspace` 过程”、第 355 页的“`sp_iqindexuse` 过程”、第 399 页的“`sp_iqtableuse` 过程”、第 402 页的“`sp_iqunusedcolumn` 过程”、第 403 页的“`sp_iqunusedindex` 过程”、第 405 页的“`sp_iqunusedtable` 过程”和第 411 页的“`sp_iqworkmon` 过程”。

《参考：语句和选项》的第 2 章“数据库选项”中的“`FP_LOOKUP_SIZE` 选项”、“`INDEX_ADVISOR` 选项”和“`MINIMIZE_STORAGE` 选项”。

《系统管理指南，卷 1》中的第 6 章“使用 Sybase IQ 索引”。

## sp\_iqindexfragmentation 过程

作用

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

**table-name** 目标 table *table-name* 用于报告指定表中的所有非缺省索引。

**index-name** 目标 index *index-name* 用于报告指定索引。每个 *index-name* 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')
```

| 索引                  | 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。这些类型包括 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

《参考：构件块、表和过程》的第 2 章“数据库选项”中的“GARRAY\_FILL\_FACTOR\_PERCENT 选项”和“GARRAY\_PAGE\_SPLIT\_PAD\_PERCENT 选项”。

《参考：语句和选项》的第 2 章“数据库选项”中的“FP\_LOOKUP\_SIZE 选项”、“INDEX\_ADVISOR 选项”和“MINIMIZE\_STORAGE 选项”。

《系统管理指南，卷 1》中的第 6 章“使用 Sybase IQ 索引”。

## sp\_iqindexinfo 过程

**作用** 显示某个给定对象在每个主 dbspace 的每个索引中所用的块数。如果对象驻留在多个 dbspace 中，则 sp\_iqindexinfo 返回在所有 dbspace 中使用的空间，如下例所示。

**语法**

```
sp_iqindexinfo '{ database
| [table table-name | index index-name] [...] }
| resources resource-percent '
```

**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 显示辅助服务器上有关共享的 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。通过资源百分比，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。

**说明** sp\_iqindexinfo 会向 DBA 显示给定对象驻留在哪些 dbspace 上。DBA 可用该信息来确定哪些 dbspace 必须具有 relocate 模式才能重新分配对象。

sp\_iqindexinfo 的结果将依据运行该命令的事务所见到的版本来显示。不显示由其它版本使用的块。

**表 7-36: sp\_iqindexinfo 列**

| 列名           | 说明                                                            |
|--------------|---------------------------------------------------------------|
| Object       | 表、索引或连接索引的名称                                                  |
| Dbspace_name | dbspace 的名称                                                   |
| ObjSize      | 此 dbspace 中此对象的数据大小                                           |
| DBSpPct      | 该对象所使用的 dbspace 的百分比                                          |
| MinBlk       | 由此 dbspace 中的此对象使用的第一个块                                       |
| MaxBlk       | 由此 dbspace 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 dbspace 调整为较小的大小 |

示例 显示关于 t2 的索引信息：

```
sp_iqindexinfo 'table t2';
```

| Object                     | dbspace_name   | ObjSize | DBSpPct | MinBlk  | MaxBlk  |
|----------------------------|----------------|---------|---------|---------|---------|
| t2                         | IQ_SYSTEM_MAIN | 32K     | 1       | 84      | 107     |
| 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 [第 320 页的“sp\\_iqdbspace 过程”](#)、[第 322 页的“sp\\_iqdbspaceinfo 过程”](#)、[第 383 页的“sp\\_iqspaceinfo 过程”](#)。

《系统管理指南，卷 1》中的第 5 章“使用数据库对象”。

《使用 Sybase IQ Multiplex》附录 A “Multiplex 引用”中的“sp\_iqindexinfo 过程”。

## sp\_iqindexmetadata 过程

**作用** 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。

**说明** 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 7-37: `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<ld>, Version, Distinct Keys                                                                                                                    |
| LF                 | Type, Version, IndexStatus, NumberOfBlockmaps, BitsPerBlockmap, Distinct Keys                                                                                |
| WD                 | Type, Version, KeySize, Delimiters, DelimiterCount, MaxKeyWordLength, PermitEmptyWord                                                                        |

示例

以下命令显示了关于 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

第 346 页的“`sp_iqindex` 和 `sp_iqindex_alt` 过程”、第 349 页的“`sp_iqindexfragmentation` 过程”、第 351 页的“`sp_iqindexinfo` 过程”和第 354 页的“`sp_iqindexsize` 过程”。

《系统管理指南, 卷 1》的《系统管理指南, 卷 1》中的第 5 章“使用数据库对象”和第 6 章“使用 Sybase IQ 索引”。

《参考: 语句和选项》的第 2 章“数据库选项”中的“`FP_LOOKUP_SIZE` 选项”和“`MINIMIZE_STORAGE` 选项”。

《系统管理指南, 卷 1》中的第 6 章“使用 Sybase IQ 索引”。

## sp\_iqindexsize 过程

- 作用                                    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。
- 说明                                    **表 7-38: sp\_iqindexsize 列**

| 列名               | 说明                                                                                                                                                          |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Username         | 索引所有者。                                                                                                                                                      |
| Indexname        | 返回其结果的索引，包括表名。                                                                                                                                              |
| Type             | 索引类型。                                                                                                                                                       |
| Info             | 报告其 KBytes、Pages 和 Compressed Pages 的 IQ 索引的组件。组件因索引类型而异。例如，缺省的 (FP) 索引包括 BARRAY (barray) 和 Bitmap (bm) 组件。Low_Fast (LF) 索引包括 B-tree (bt) 和 Bitmap (bm) 组件。 |
| KBytes           | 以 KB 表示的物理对象大小。                                                                                                                                             |
| Pages            | 将对象保存在内存中所需的 IQ 页数。                                                                                                                                         |
| Compressed Pages | 对象在磁盘上压缩后的 IQ 页数。                                                                                                                                           |

以字节或千字节为单位返回索引的总大小，Info 列则描述会报告其 KBytes、Pages 和 Compressed Pages 的 IQ 索引的组件。描述的组件因素索引类型而异。例如，缺省的 (FP) 索引包括 BARRAY (barray) 和 Bitmap (bm) 组件。Low\_Fast (LF) 索引包括 B-tree (bt) 和 Bitmap (bm) 组件。

此外，还返回将对象保存在内存中所需的页数，以及将索引在磁盘上压缩后的 IQ 页数。

对于此过程，必须指定 *index\_name* 参数。若要将结果限制为单个表中的该索引名，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                      | Type | Info     | KBytes | Pages | Compressed Pages |
|----------|--------------------------------|------|----------|--------|-------|------------------|
| DBA      | Employees.ASIQ_IDX_T452_C19_FP | FP   | Total    | 288    | 4     | 2                |
| DBA      | Employees.ASIQ_IDX_T452_C19_FP | FP   | vdo      | 0      | 0     | 0                |
| 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           | Type | Info     | KBytes | Pages | Compressed Pages |
|----------|---------------------|------|----------|--------|-------|------------------|
| GROUPO   | GROUPO.Employees.ti | TEXT | Total    | 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                |
| 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

《系统管理指南，卷 1》中的第 6 章“使用 Sybase IQ 索引”。

《参考：语句和选项》的第 2 章“数据库选项”中的“FP\_LOOKUP\_SIZE 选项”和“MINIMIZE\_STORAGE 选项”。

## sp\_iqindexuse 过程

作用

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。

说明

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。

不会报告在 SYSTEM 中创建的表的索引。

**表 7-39: sp\_iqindexuse 列**

| 列名        | 说明        |
|-----------|-----------|
| IndexName | 索引名称      |
| TableName | 表名        |
| Owner     | 索引所有者的用户名 |
| UID**     | 索引唯一标识符   |
| Type      | 索引类型      |

| 列名          | 说明              |
|-------------|-----------------|
| 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

《性能和调优指南》的第 3 章“优化查询和删除”中的“监控负载”。

第 305 页的“sp\_iqcolumnuse 过程”、第 320 页的“sp\_iqdbspace 过程”、第 348 页的“sp\_iqindexadvice 过程”、第 399 页的“sp\_iqtableuse 过程”、第 402 页的“sp\_iqunusedcolumn 过程”、第 403 页的“sp\_iqunusedindex 过程”、第 405 页的“sp\_iqunusedtable 过程”和第 411 页的“sp\_iqworkmon 过程”。

《参考：语句和选项》的第 2 章“数据库选项”中的“INDEX\_ADVISOR 选项”。

## sp\_iqjoinindex 过程

作用

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

**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。

`sp_iqjoinindex` 过程可以不带任何参数进行调用。如果未指定参数，`sp_iqjoinindex` 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。

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 7-40: `sp_iqjoinindex` 用法示例**

| 语法                                                           | 输出                                                                                                                    |
|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>sp_iqjoinindex</code>                                  | 显示与所有连接索引有关的信息。                                                                                                       |
| <code>sp_iqjoinindex t1, NULL, DBA</code>                    | 显示 DBA 所拥有的 <code>t1</code> 构成操作左侧的所有连接索引的信息。                                                                         |
| <code>sp_iqjoinindex t2, n1, DBA</code>                      | 显示以 DBA 所拥有的表 <code>t2</code> 的列 <code>n1</code> 作为连接左侧的连接索引信息。                                                       |
| <code>sp_iqjoinindex NULL, NULL, DBA, NULL, NULL, DBA</code> | 显示其左侧和右侧表皆为 DBA 所有的连接索引的信息。                                                                                           |
| <code>sp_iqjoinindex NULL, NULL, NULL, t2, NULL, NULL</code> | 显示表 <code>t2</code> 位于连接操作右侧的所有连接索引的信息。                                                                               |
| <code>sp_iqjoinindex t1, n1, DBA, t2, n1, DBA</code>         | 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 <code>t1</code> 的列 <code>n1</code> ，右侧为 DBA 所拥有的表 <code>t2</code> 的列 <code>n1</code> 。 |
| <code>sp_iqjoinindex non_existing_table</code>               | 不返回任何行，因为表 <code>non_existing_table</code> 不存在。                                                                       |
| <code>sp_iqjoinindex NULL, NULL, non_existing_user</code>    | 不返回任何行，因为用户 <code>non_existing_user</code> 不存在。                                                                       |

另请参见

《参考：语句和选项》中的 [CREATE JOIN INDEX 语句](#)。

《系统管理指南，卷 1》中的第 6 章“使用 Sybase IQ 索引”。

说明

`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。如果未指定参数，`sp_iqjoinindex` 显示数据库中所有连接索引的信息。

sp\_iqjoinindex 过程使用以下列返回信息：

**表 7-41: sp\_iqjoinindex 列**

| 列名                | 说明                                                                                                                          |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------|
| 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。                                                                                                             |
| key_type          | 定义键上的连接类型： <ul style="list-style-type: none"> <li>• NATURAL：自然连接</li> <li>• KEY：键连接</li> <li>• ON：左外 / 右外 / 完全连接</li> </ul> |
| valid             | 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

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 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

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 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

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 dbspace_id remarks
t1_t2_t3_join DBA t2 DBA name NATURAL
= t3 DBA name NATURAL
Y 16387 ((NULL))

```

## sp\_iqjoinindexsize 过程

**作用** 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。

**说明** 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 7-42: sp\_iqjoinindexsize 列**

| 列名               | 说明                 |
|------------------|--------------------|
| 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

作用 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 7-43: “edition” 参数的摘要信息**

| 主题   | 值                                         |
|------|-------------------------------------------|
| 缺省值  | 'EE' (企业版)                                |
| 值的范围 | 'SE' (小型商业版)<br>'SA' (单应用版)<br>'EE' (企业版) |
| 状态   | Static                                    |

语法 2 `sp_iqlmconfig 'license type', { 'CP' | 'DT' | 'SF' | 'AC' | 'BC' | 'CH' | 'DH' | 'SH' | 'AH' | 'BH' }`

**表 7-44: “license type” 参数的摘要信息**

| 主题   | 值                                                                                                                                                                                                                 |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 缺省值  | 'DT' (开发和测试)                                                                                                                                                                                                      |
| 值的范围 | 'AC' (OEM CPU 许可)<br>'AH' (OEM 芯片 CPU 许可)<br>'BC' (OEM 备用许可)<br>'BH' (OEM 芯片备用许可)<br>'CP' (CPU 许可)<br>'CH' (芯片 CPU 许可)<br>'DH' (芯片开发和测试许可)<br>'DT' (开发和测试)<br>'EV' (评测)<br>'SF' (备用 CPU 许可)<br>'SH' (备用芯片 CPU 许可) |
| 状态   | Static                                                                                                                                                                                                            |

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> | '`

**host name** 指定用于电子邮件通知的 SMTP 主机。

语法 5 `sp_iqlmconfig 'email sender', '<email address> |`

**<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> |`

**<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 | <code>sp_iqlmconfig</code>                                                                                                                                                                                                                                                                                                                                                                           |
| 权限   | 需要 DBA 授权。                                                                                                                                                                                                                                                                                                                                                                                           |
| 用法   | <p>启动时，<code>sp_iqlmconfig</code> 检查指定许可证的版本类型和许可证类型。</p> <ul style="list-style-type: none"> <li>• 如果未找到指定许可证，则服务器处于宽限模式。</li> <li>• 只有指定了非空的版本值时，指定许可证类型才有效。</li> <li>• 如果不带任何参数调用 <code>sp_iqlmconfig</code>（语法 3），将显示上述所有信息，以及其它信息，例如： <ul style="list-style-type: none"> <li>• 产品版本和许可证类型</li> <li>• 正在使用哪些可选许可证</li> <li>• 许可证计数</li> <li>• 电子邮件通知</li> <li>• 有关许可证的常规信息</li> </ul> </li> </ul> |

## sp\_iqlocks 过程

|    |                                                                                     |
|----|-------------------------------------------------------------------------------------|
| 作用 | 显示与数据库中 IQ 存储和 Catalog 存储中的锁有关的信息。                                                  |
| 语法 | <code>sp_iqlocks</code> ([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 | 表 7-45 列出了您可以指定的用来限制结果的可选 <code>sp_iqlocks</code> 参数。                               |

**表 7-45: 可选 `sp_iqlocks` 参数**

| 名称                      | 数据类型      | 说明                                                                                                  |
|-------------------------|-----------|-----------------------------------------------------------------------------------------------------|
| <i>connection</i>       | integer   | 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 <code>owner</code> ，系统则假定该过程的调用者拥有该表。 |
| <i>max_locks</i>        | integer   | 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。                                                                      |

| 名称                | 数据类型    | 说明                                                                                                                   |
|-------------------|---------|----------------------------------------------------------------------------------------------------------------------|
| <i>sort_order</i> | char(1) | 信息返回的顺序：<br><ul style="list-style-type: none"> <li>• C 表示按连接排序（缺省值）</li> <li>• T 表示按 <i>table_name</i> 排序</li> </ul> |

说明

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，并按参数 *sort\_order* 中指定的顺序进行排序：

**表 7-46: *sp\_iqlocks* 列**

| 列                     | 数据类型            | 说明                                                                                                                                                                                                                                                                                                                   |
|-----------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>conn_name</i>      | VARCHAR(128)    | 当前连接的名称。                                                                                                                                                                                                                                                                                                             |
| <i>conn_id</i>        | INTEGER         | 拥有锁的连接 ID。                                                                                                                                                                                                                                                                                                           |
| <i>user_id</i>        | CHAR(128)       | 与该连接 ID 相关联的用户。                                                                                                                                                                                                                                                                                                      |
| <i>table_type</i>     | 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 目录存储中的 SQL Anywhere 表支持物化视图。                                                                                                                                                                                                                                 |
| <i>creator</i>        | VARCHAR(128)    | 表的所有者。                                                                                                                                                                                                                                                                                                               |
| <i>table_name</i>     | VARCHAR(128)    | 持有锁的表。                                                                                                                                                                                                                                                                                                               |
| <i>index_id</i>       | INTEGER         | 索引 ID 或空值。                                                                                                                                                                                                                                                                                                           |
| <i>lock_class</i>     | CHAR(8)         | 指示锁类型的字符串：<br>S — 共享。<br>SW — 共享且可写。<br>EW — 排它且可写。<br>E — 排它。<br>P — 幻像。<br>A — 反幻像。<br>W — 可写。<br>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 *：<br>T — 该锁针对有序扫描。<br>* — 该锁针对所有扫描。<br><i>nnn</i> — 索引号；该锁针对特定索引。<br>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。 |
| <i>lock_duration</i>  | CHAR(11)        | 锁持续时间。Transaction、Position 或 Connection 之一。                                                                                                                                                                                                                                                                          |
| <i>lock_type</i>      | CHAR(9)         | 标识锁的值（取决于锁类）                                                                                                                                                                                                                                                                                                         |
| <i>row_identifier</i> | 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，就会以 0（零）来表示该连接 ID、以 `User unavailable` 来表示用户名。

**注释** 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_identifier
===== ===== ===== ===== =====
ASIQ_IDX_T452_C19_FP Table Position Table 1
```

## sp\_iqmodifyadmin 过程

#### 作用

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

**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。

另请参见 [第 368 页的“sp\\_iqpassword 过程”](#)。  
《参考：语句和选项》中 [第 1 章“SQL 语句”](#) 中的 ALTER LOGIN POLICY 语句。

示例 **示例 1** 将名为 *lockeduser* 的策略的登录选项 *locked* 设置为 ON。

```
call sp_iqmodifyadmin ('locked','on','lockeduser')
```

**示例 2** 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')
```

## sp\_iqmodifylogin 过程

作用 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 **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 **示例 1** 为用户 *joe* 分配名为 *expired\_password* 的登录策略：

```
sp_iqmodifylogin 'joe', 'expired_password'
```

**示例 2** 为用户 *joe* 分配 *root* 登录策略：

```
call sp_iqmodifylogin ('joe')
```

另请参见 [第 363 页的“sp\\_iqmodifyadmin 过程”](#)。

## sp\_iqmpxinconnpoolinfo 过程

作用 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。若要了解 `sp_iqmpxinconnpoolinfo` 语法和完整说明，请参见《使用 Sybase IQ Multiplex》。

## sp\_iqmpxinheartbeatinfo 过程

**作用** 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。若要了解 sp\_iqmpxinheartbeatinfo 语法和完整说明，请参见《使用 Sybase IQ Multiplex》。

## sp\_iqmpxinfo 过程

**作用** 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。若要了解 sp\_iqmpxinfo 语法和完整说明，请参见《使用 Sybase IQ Multiplex》。

## sp\_iqmpxvalidate 过程

**作用** 检查 Multiplex 配置中是否存在不一致情况。若要了解 sp\_iqmpxvalidate 语法和完整说明，请参见《使用 Sybase IQ Multiplex》。

## sp\_iqmpxversioninfo 过程

**作用** 显示此服务器的当前版本信息。若要了解 sp\_iqmpxversioninfo 语法和完整说明，请参见《使用 Sybase IQ Multiplex》。

## sp\_iqobjectinfo 过程

**作用** 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** **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 显示数据库中所有表和连接索引的相关信息。

**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** 一同使用；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** 存储过程支持用于解释 *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 **dbspace** 的信息。

说明

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 7-47: sp\_iqobjectinfo 列**

| 列名              | 说明                                         |
|-----------------|--------------------------------------------|
| owner           | 对象所有者的名称。                                  |
| object_name     | 位于 <b>dbspace</b> 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 <b>dbspace</b> 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。                          |

| 列名             | 说明                                                                            |
|----------------|-------------------------------------------------------------------------------|
| 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

**注释** 下面的两个示例演示 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 dbspace 示例，您的数据库中可能不包括该示例。

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 dbspace 赋值的信息：

```
sp_iqobjectinfo GROUPO,Departments
```

| owner  | object_name | sub_object_name                | object_type | object_id | id  |
|--------|-------------|--------------------------------|-------------|-----------|-----|
| GROUPO | Departments | (NULL)                         | table       | 3632      | 738 |
| GROUPO | Departments | DepartmentID                   | column      | 3633      | 738 |
| GROUPO | Departments | DepartmentName                 | column      | 3634      | 738 |
| GROUPO | Departments | DepartmentHeadID               | column      | 3635      | 738 |
| GROUPO | Departments | DepartmentsKey                 | primary key | 83        | 738 |
| GROUPO | Departments | FK_DepartmentHeadID_EmployeeID | foreign key | 92        | 738 |

| dbspace_name | partition_name |
|--------------|----------------|
| 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 dbspace 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)         |

## sp\_iqpassword 过程

|      |                                                                                                                                                                                                                                                                   |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作用   | 更改用户口令。创建用户的首选方式是使用 CREATE USER 语句。请参见《参考：语句和选项》的第 1 章“SQL 语句”中的“CREATE USER 语句”。                                                                                                                                                                                 |
| 语法 1 | <pre>call sp_iqpassword ('caller_password', 'new_password' [, 'user_name'])</pre>                                                                                                                                                                                 |
| 语法 2 | <pre>sp_iqpassword 'caller_password', 'new_password' [, 'user_name']</pre>                                                                                                                                                                                        |
| 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   | <p><b>caller_password</b> 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。</p> <p><b>new_password</b> 用户的新口令，或 loginname 的新口令。</p> <p><b>user_name</b> 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。</p> |
| 说明   | 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。                                                                                                                                                                                                |
| 示例   | <p><b>示例 1</b> 将已登录用户的口令从 irk103 更改为 exP984:</p> <pre>sp_iqpassword 'irk103', 'exP984'</pre> <p><b>示例 2</b> 仅当已登录用户具有 DBA or PERMS ADMIN 权限或用户是 joe 本人时，才会将用户 joe 的口令从 epr45 更改为 pdi032:</p> <pre>call sp_iqpassword ('epr45', 'pdi932', 'joe')</pre>             |

## sp\_iqkeys 过程

|    |                                                                           |
|----|---------------------------------------------------------------------------|
| 作用 | 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 | <pre>sp_iqkeys { [ table-name ], [ column-name ], [ table-owner ] }</pre> |
| 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 | <b>table-name</b> 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 7-48: sp\_iqpkeys 用法示例**

| 语法                              | 输出                                         |
|---------------------------------|--------------------------------------------|
| sp_iqpkeys sales                | 显示表 sales 上定义的主键的信息                        |
| sp_iqpkeys sales, NULL, DBA     | 显示在 DBA 所拥有的表 sales 上定义的主键的信息              |
| sp_iqpkeys sales, store_id, DBA | 显示在 DBA 所拥有的表 sales 中的列 store_id 上定义的主键的信息 |
| sp_iqpkeys NULL, NULL, DBA      | 显示在 DBA 拥有的所有表上定义的主键的信息                    |

说明

sp\_iqpkeys 存储过程显示有关数据库中基表和全局临时表的主键的下列信息：

**表 7-49: sp\_iqpkeys 列**

| 列名              | 说明        |
|-----------------|-----------|
| table_name      | 表的名称      |
| table_owner     | 表的所有者     |
| column_name     | 定义主键的列的名称 |
| column_id       | 列 ID      |
| constraint_name | 主键约束的名称   |
| constraint_id   | 主键约束 ID   |

**注释** sp\_iqpkeys 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_iqpkeys sales1
```

```
table_name table_owner column_name column_id constraint_name constraint_id
sales1 DBA store_id 1 MA114 114
```

显示在表 sales2 的列上定义的主键：

```
sp_iqpkkeys sales2
```

```
table_name table_owner column_name column_id constraint_name constraint_id
sales2 DBA store_id, 1,2 MA115 115
 order_num
```

显示在表 **sales2** 的列 **store\_id** 上定义的主键:

```
sp_iqpkkeys sales2, store_id
```

```
table_name table_owner column_name column_id constraint_name constraint_id
sales2 DBA store_id 1 MA115 115
```

另请参见 [第 346 页的 “sp\\_iqindex 和 sp\\_iqindex\\_alt 过程”](#)。  
[第 303 页的 “sp\\_iqcolumn 过程”](#)。

## sp\_iqprocedure 过程

**作用** 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** **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:** 仅显示有关系统过程（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

**sp\_iqprocedure** 过程可以不带任何参数进行调用。如果未指定任何参数，则缺省情况下仅显示与用户定义过程（不由 dbo 或 SYS 所拥有的过程）有关的信息。

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 7-50: *sp\_iqprocedure* 用法示例

| 语法                                              | 输出                                                                                          |
|-------------------------------------------------|---------------------------------------------------------------------------------------------|
| <i>sp_iqprocedure</i>                           | 显示不由 <i>dbo</i> 或 <i>SYS</i> 拥有的数据库中的所有过程的信息                                                |
| <i>sp_iqprocedure sp_test</i>                   | 显示有关过程 <i>sp_test</i> 的信息                                                                   |
| <i>sp_iqprocedure non_existing_proc</i>         | 不返回任何行，因为过程 <i>non_existing_proc</i> 不存在                                                    |
| <i>sp_iqprocedure NULL, DBA</i>                 | 显示有关由 <i>DBA</i> 所拥有的所有过程的信息                                                                |
| <i>sp_iqprocedure sp_test, DBA</i>              | 显示有关由 <i>DBA</i> 所拥有的过程 <i>sp_test</i> 的信息                                                  |
| <i>sp_iqprocedure sp_iqtable</i>                | 过程 <i>sp_iqtable</i> 不是系统过程。如果没有任何用户定义的过程也命名为 <i>sp_iqtable</i> ，则不会返回任何行。（缺省情况下仅返回用户定义过程。） |
| <i>sp_iqprocedure sp_iqtable, dbo</i>           | 不返回任何行，因为过程 <i>sp_iqtable</i> 不是用户过程（缺省情况下仅返回用户过程）                                          |
| <i>sp_iqprocedure NULL, NULL, SYSTEM</i>        | 显示与所有系统过程（由 <i>dbo</i> 或 <i>SYS</i> 所拥有）有关的信息                                               |
| <i>sp_iqprocedure sp_iqtable, NULL, 'SYSTEM</i> | 显示与系统过程 <i>sp_iqtable</i> 有关的信息                                                             |
| <i>sp_iqprocedure sp_iqtable, dbo, ALL</i>      | 显示有关由 <i>dbo</i> 所拥有的系统过程 <i>sp_iqtable</i> 的信息                                             |

## 说明

*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，如果指定了 *proc-name*，将仅显示与该指定过程有关的信息。如果指定 *proc-owner*，*sp\_iqprocedure* 仅返回属于该指定所有者的过程的信息。如果未指定任何参数，*sp\_iqprocedure* 显示数据库中所有用户定义过程的信息。

*sp\_iqprocedure* 过程使用以下列返回信息：

表 7-51: *sp\_iqprocedure* 列

| 列名                | 说明                                              |
|-------------------|-------------------------------------------------|
| <i>proc_name</i>  | 过程的名称                                           |
| <i>proc_owner</i> | 过程的所有者                                          |
| <i>proc_defn</i>  | 用于创建过程的命令。对于隐藏过程，将显示关键字“HIDDEN”。                |
| <i>replicate</i>  | 如果过程是 Replication Server 安装中的主数据源，将显示 Y；否则显示 N。 |
| <i>srvid</i>      | 如果过程在远程数据库服务器上，则指示该远程服务器                        |
| <i>remarks</i>    | 注释字符串                                           |

示例 显示用户定义过程 `sp_test` 的信息：

```
sp_iqprocedure sp_test
```

| proc_name | proc_owner | proc_defn                                                                      | replicate | srvid  | remarks |
|-----------|------------|--------------------------------------------------------------------------------|-----------|--------|---------|
| sp_test   | DBA        | create procedure<br>DBA.sp_test(in n1<br>integer)<br>begin message'sp_test'end | N         | (NULL) | (NULL)  |

显示由用户 `DBA` 所拥有的所有过程的信息：

```
sp_iqprocedure NULL, DBA
```

| proc_name | proc_owner | proc_defn                                                                        | replicate | srvid  | remarks |
|-----------|------------|----------------------------------------------------------------------------------|-----------|--------|---------|
| sp_test   | DBA        | create procedure<br>DBA.sp_test(in n1<br>integer)<br>begin message 'sp_test' end | N         | (NULL) | (NULL)  |
| sp_dept   | DBA        | create procedure<br>DBA.sp_dept() begin end                                      | N         | (NULL) | (NULL)  |

另请参见 《参考：语句和选项》的第 1 章 “SQL 语句” 中的 “CREATE USER 语句”。

## sp\_iqprocparm 过程

**作用** 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** **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**: 仅显示有关系统过程（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，则显示用户定义过程（不由 `dbo` 或 `SYS` 所拥有的过程）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 7-52: `sp_iqprocparm` 用法示例**

| 语法                                                  | 输出                                                                                                   |
|-----------------------------------------------------|------------------------------------------------------------------------------------------------------|
| <code>sp_iqprocparm</code>                          | 显示不由 <code>dbo</code> 或 <code>SYS</code> 所拥有的数据库中的所有过程的参数。                                           |
| <code>sp_iqprocparm sp_test</code>                  | 显示有关过程 <code>sp_test</code> 的信息。                                                                     |
| <code>sp_iqprocparm non_existing_proc</code>        | 不返回任何行，因为过程 <code>non_existing_proc</code> 不存在。                                                      |
| <code>sp_iqprocparm NULL, DBA</code>                | 显示由 <code>DBA</code> 所拥有的所有过程的参数。                                                                    |
| <code>sp_iqprocparm sp_test, DBA</code>             | 显示由 <code>DBA</code> 所拥有的过程 <code>sp_test</code> 的参数。                                                |
| <code>sp_iqprocparm sp_iqtable</code>               | <code>sp_iqtable</code> 是系统过程。如果没有任何用户定义的过程也命名为 <code>sp_iqtable</code> ，则不会返回任何行。（缺省情况下，仅返回用户定义过程。） |
| <code>sp_iqprocparm sp_iqtable, dbo</code>          | 不返回任何行，因为过程 <code>sp_iqtable</code> 不是用户过程。（缺省情况下，仅返回用户过程。）                                          |
| <code>sp_iqprocparm NULL, NULL, SYSTEM</code>       | 显示所有系统过程（由 <code>dbo</code> 或 <code>SYS</code> 所拥有的）的参数。                                             |
| <code>sp_iqprocparm sp_iqtable, NULL, SYSTEM</code> | 显示系统过程 <code>sp_iqtable</code> 的参数。                                                                  |
| <code>sp_iqprocparm sp_iqtable, dbo, ALL</code>     | 显示由 <code>dbo</code> 所拥有的系统过程 <code>sp_iqtable</code> 的参数。                                           |

#### 说明

`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`，`sp_iqprocparm` 仅返回属于该指定所有者的过程的参数相关信息。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 过程使用以下列返回信息：

**表 7-53: sp\_iqprocparm 列**

| 列名          | 说明                                                                                                                                                                                                                     |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| proc_name   | 过程的名称                                                                                                                                                                                                                  |
| proc_owner  | 过程的所有者                                                                                                                                                                                                                 |
| parm_name   | 参数名                                                                                                                                                                                                                    |
| parm_type   | 参数类型为下列值之一： <ul style="list-style-type: none"> <li>• 普通参数（变量）</li> <li>• 结果变量：用于返回结果集的过程</li> <li>• SQLSTATE 错误值</li> <li>• SQLCODE 错误值</li> </ul>                                                                     |
| parm_mode   | 参数模式：参数为过程提供一个值；参数返回一个值；参数既提供一个值又返回一个值；参数既不提供值也不返回值。参数模式为以下模式之一： <ul style="list-style-type: none"> <li>• in：参数为过程提供一个值</li> <li>• out：参数返回一个值</li> <li>• inout：参数既提供一个值又返回一个值</li> <li>• NULL：参数既不提供值也不返回值</li> </ul> |
| domain_name | 参数的数据类型名称，如 SYSDOMAIN 系统表中所列                                                                                                                                                                                           |
| width       | 字符串参数的长度、数值参数的精度以及所有其它数据类型的存储字节数                                                                                                                                                                                       |
| scale       | 数值数据类型参数的小数点后的位数，对于所有其它数据类型该值为零                                                                                                                                                                                        |
| default     | 参数的缺省值，保留为字符串                                                                                                                                                                                                          |

示例

显示有关用户定义过程 sp\_test 的参数的信息：

```
sp_iqprocparm sp_test

proc_name proc_owner parm_name parm_type parm_mode domain_name width scale default
sp_test DBA ID normal in integer 4 0 (NULL)
```

显示有关系统过程 sp\_iqshowcompression 的参数的信息：

```
sp_iqprocparm sp_iqshowcompression, dbo, system

proc_name proc_owner parm_name parm_type parm_mode
domain_name width scale default
sp_iqshowcompression dbo @owner_name normal in
char 128 0 (NULL)
sp_iqshowcompression dbo @table_name normal in
char 128 0 (NULL)
```

```

sp_iqshowcompression dbo @column_name normal in
char 128 0 (NULL)
sp_iqshowcompression dbo Column result out
char 128 0 (NULL)
sp_iqshowcompression dbo Compression result out
char 3 0 (NULL)

```

另请参见 [《参考：语句和选项》的第1章“SQL 语句”中的 CREATE PROCEDURE 语句。](#)

## sp\_iqrebuildindex 过程

**作用** 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** 只有拥有表的 INSERT 权限，才能对该表重建索引。

**用法** **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 column_name [count]`

`[dbspace dbspace_name]`

`index index_name`

`[dbspace dbspace_name]`

每个 `column_name` 或 `index_name` 必须引用指定表中的某列或某个索引。如果多次指定 `column_name` 或 `index_name`，该过程就会返回一条错误，并且不重建任何索引。

`count` 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非重复值的估计数量。不同的值的数量会影响查询速度和存储要求。有关详细信息，请参见 [《系统管理指南，卷1》第5章“使用数据库对象”中的“优化存储和查询性能”](#)。

必须指定关键字 `column` 和 `index`。关键字 `dbspace` 是可选的。这些关键字是不区分大小写的。

除非指定了 `dbspace dbspace-name`，否则 Sybase IQ 在原始列或索引驻留的同一 `dbspace` 中重建列或索引。

---

**注释** 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。

---

另请参见

第 349 页的“[sp\\_iqindexfragmentation 过程](#)”和第 380 页的“[sp\\_iqrowdensity 过程](#)”。

《系统管理指南，卷 1》中的第 6 章“使用 Sybase IQ 索引”。

《参考：语句和选项》的第 2 章“数据库选项”中的“[FP\\_LOOKUP\\_SIZE 选项](#)”和“[MINIMIZE\\_STORAGE 选项](#)”。

说明

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`，0（零）会用作缺省值。

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，就会用列的基数值来更新 `SYS.SYSIQCOLUMN` 中的 `approx_unique_count` 列。

如果列中有大于 255 字节的数据类型 `VARCHAR` 或 `VARBINARY`，`sp_iqrebuildindex` 将不会重建缺省索引。

`sp_iqrebuildindex` 将对一系列数据类型 `LONG VARCHAR` 重建一个 `WD` 索引 (`CLOB`)。

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`，就会使用该数据类型的缺省值来重建索引。有关详细信息，请参见《系统管理指南，卷 1》中的“[Sybase IQ 索引类型](#)”和第 7 章“[系统过程](#)”。

示例

重建列 *Surname* 的缺省索引:

```
sp_iqrebuildindex 'empl', 'column dept_id'
```

或:

```
call sp_iqrebuildindex ('empl', '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')
```

另请参见

第 349 页的“[sp\\_iqindexfragmentation 过程](#)”和第 380 页的“[sp\\_iqrowdensity 过程](#)”。

《参考: 语句和选项》的第 2 章“[数据库选项](#)”中的“[FP\\_LOOKUP\\_SIZE 选项](#)”和“[MINIMIZE\\_STORAGE 选项](#)”。

《系统管理指南, 卷 1》中的第 6 章“[使用 Sybase IQ 索引](#)”。

## sp\_iqrename 过程

作用

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

**object-name** 用户创建的对象的原名称。

或者, *owner-name* 可以指定为 *object-name* 的一部分, 即 *owner-name.object-name*, 其中 *owner-name* 是要重命名的对象的所有者名称。如果未指定 *owner-name*, 调用 **sp\_iqrename** 的用户就会被假定为对象的所有者。只有调用 **sp\_iqrename** 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, *object-name* 的形式可以为 *table-name.object-name* 或 *owner-name.table-name.object-name*。

**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 *object-name* 的类型。*object-type* 参数可用大写或小写字母进行指定。

**表 7-54: sp\_iqrename object-type 参数值**

| object-type 参数         | 指定                                |
|------------------------|-----------------------------------|
| 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 <i>object-type</i> | 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。

说明

`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 “Item already exists”。

`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 “Feature not supported”。

示例

将属于用户 `shweta` 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
```

将针对表 `books` 的主键约束 `prim_id` 重命名为 `prim_isbn`：

```
sp_iqrename books.prim_id, prim_isbn, constraint
```

另请参见

《参考：语句和选项》中的 [ALTER TABLE 语句](#) 的 `RENAME` 子句和 [ALTER INDEX 语句](#) 的 `RENAME` 子句。

## sp\_iq\_reset\_identity 过程

|      |                                                                                                                        |
|------|------------------------------------------------------------------------------------------------------------------------|
| 作用   | 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   | <b>sp_iq_reset_identity</b> ( <i>table_name</i> , <i>table_owner</i> , <i>value</i> )                                  |
| 用法   | <b>语法</b> 必须指定 <i>table_name</i> 、 <i>table owner</i> 和 <i>value</i> 。                                                 |
| 权限   | 不需要。                                                                                                                   |
| 说明   | 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。             |
| 示例   | 下面的示例创建起始种子为 50 的标识列： <pre>CREATE TABLE mytable(c1 INT identity) call sp_iq_reset_identity('mytable', 'dba', 50)</pre> |
| 另请参见 | 第 303 页的 “ <a href="#">sp_iqcolumn 过程</a> ”。                                                                           |

## sp\_iqrestoreaction 过程

|    |                                                                                                                                                                                                                                                                                                                                                            |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 作用 | 显示在给定期日之后要使数据库达到一致状态所需要的恢复操作。                                                                                                                                                                                                                                                                                                                              |
| 语法 | <b>sp_iqrestoreaction</b> [ <i>timestamp</i> ]                                                                                                                                                                                                                                                                                                             |
| 参数 | <b>timestamp</b> 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。                                                                                                                                                                                                                                                                                                          |
| 说明 | <p>如果不能使数据库达到时间戳的一致状态，<b>sp_iqrestoreaction</b> 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。</p> <p>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。</p> <p>根据备份时间，输出可能不完全是升序顺序。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）。</p> <p>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。</p> |

sp\_iqrestoreaction 将返回:

**表 7-55: sp\_iqrestoreaction 列**

| 列名                  | 说明                                                |
|---------------------|---------------------------------------------------|
| 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” |
| 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
1 1192 c:\\\\temp\\\\b1 2008-09-23 14:47:40.0
2 1201 c:\\\\temp\\\\b2.inc 2008-09-23 14:48:05.01
3 1208 c:\\\\temp\\\\b3.inc 2008-09-23 14:48:13.0

virtual_type restore_dbSPACE restore_dbfile backup_comment
Nonvirtual
Nonvirtual
Nonvirtual

```

## sp\_iqrowdensity 过程

作用

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

**table-name** 目标表 *table-name* 用于报告指定表中的所有列。

**column-name** 目标列 *column-name* 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。

**说明** `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  $0 < density \leq 1$  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** 《参考：语句和选项》的第 2 章“数据库选项”中的“[FP\\_LOOKUP\\_SIZE 选项](#)”和“[MINIMIZE\\_STORAGE 选项](#)”。  
《系统管理指南，卷 1》中的第 6 章“使用 Sybase IQ 索引”。

## sp\_iqshowpsexex 过程

**作用** 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** `connection-id` 表示连接 ID 的整数。

如果指定 `connection-id`，则 `sp_iqshowpsexex` 仅返回与指定连接相关的信息。如果不指定 `connection-id`，则 `sp_iqshowpsexex` 返回所有连接的相关信息。

如果指定的 `connection-id` 不存在，则 `sp_iqshowpsexex` 不返回任何行。

说明

sp\_iqshowpsexex 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十分有用。

**表 7-56: sp\_iqshowpsexex 列**

| 列名                     | 说明                                                                                                                              |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| connectionid           | 连接 ID                                                                                                                           |
| application            | 打开连接的客户端应用程序的相关信息。包括 ApplInfo 连接属性信息：<br>HOST: 客户端计算机的主机名<br>EXE: 客户端可执行文件的名称（仅限 Windows）<br>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 dbspace。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。                                                    |
| 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。                                             |

**注释** ApplInfo 属性在 Open Client 或 jConnect 应用程序（例如 Interactive SQL (dbisql) 或 Sybase Central）中可能不可用。如果 ApplInfo 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=GOODGUY-XP;EXE=C:\\Program Files\\Sybase\\
 IQ-15_2\\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 在第 7 章“系统过程”中：第 306 页的“[sp\\_iqconnection 过程](#)”、第 311 页的“[sp\\_iqcontext 过程](#)”和第 414 页的“[sa\\_conn\\_info 系统过程](#)”。

第 123 页的“[CONNECTION\\_PROPERTY 函数 \[系统\]](#)”。

在《参考：语句和选项》中的第 2 章“数据库选项”中：[IQGOVERN\\_MAX\\_PRIORITY 选项](#)、[IQGOVERN\\_PRIORITY 选项](#)、[IQGOVERN\\_PRIORITY\\_TIME 选项](#)、[MAX\\_QUERY\\_TIME 选项](#)、[QUERY\\_ROWS\\_RETURNED\\_LIMIT 选项](#)、[QUERY\\_TEMP\\_SPACE\\_LIMIT 选项](#)、[MAX\\_CURSOR\\_COUNT 选项](#)和 [MAX\\_STATEMENT\\_COUNT 选项](#)。

《系统管理指南，卷 1》的第 4 章“连接和通信参数”中的“[AppInfo 连接参数 \[App\]](#)”。

## sp\_iqspaceinfo 过程

作用 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。

说明 针对当前数据库显示对象名、每个对象使用的块数和 dbspace 的名称。  
**sp\_iqspaceinfo** 不需要任何参数。

**sp\_iqspaceinfo** 返回的信息对于管理 dbspace 很有帮助。

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 [第 351 页的 “sp\\_iqindexinfo 过程”](#)、[第 320 页的 “sp\\_iqdbspace 过程”](#) 和 [第 322 页的 “sp\\_iqdbspaceinfo 过程”](#)。

《系统管理指南，卷 1》中的 [第 5 章 “使用数据库对象”](#)。

《使用 Sybase IQ Multiplex》的附录 A [“Multiplex 引用”](#) 中的 [“sp\\_iqspaceinfo 过程”](#)。

## sp\_iqspaceused 过程

作用 显示 IQ 存储和 IQ 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)

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 参数返回四个值。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。

说明 `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 7-57: `sp_iqspaceused` 列**

| 列名         | 说明                                |
|------------|-----------------------------------|
| 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）。 |
| tempKB     | IQ 临时存储空间的总大小（以 KB 为单位）。          |
| tempKBUsed | 数据库使用的 IQ 临时存储空间的大小（以 KB 为单位）。    |

示例 `sp_iqspaceused` 需要四个输出参数。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四个输出参数，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;
 call sp_iqspaceused(mt,mu,tt,tu);
 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;
end
```

若要显示 `sp_iqspaceused` 的输出，请运行过程 `myspace`：

```
myspace
```

## sp\_iqstatistics 过程

作用 返回每项可用统计或指定统计的序列号、名称、说明、值和单位指示符。

语法 **sp\_iqstatistics** [ *stat\_name* ]

参数 **stat\_name** （可选） VARCHAR 参数指定统计的名称。

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。

**说明** 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             | 说明     |
|-------------------------|------------------|--------|
| <code>stat_num</code>   | UNSIGNED INTEGER | 统计的序列号 |
| <code>stat_name</code>  | VARCHAR(255)     | 统计的名称  |
| <code>stat_desc</code>  | VARCHAR(255)     | 统计的说明  |
| <code>stat_value</code> | LONG VARCHAR     | 统计的值   |
| <code>stat_unit</code>  | VARCHAR(128)     | 单位指示符  |

可能返回以下统计：

| <code>stat_num</code> | <code>stat_name</code>                      | <code>stat_desc</code>               | <code>stat_unit</code> |
|-----------------------|---------------------------------------------|--------------------------------------|------------------------|
| 0                     | <code>CpuTotalTime</code>                   | 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）  | 秒                      |
| 1                     | <code>CpuUserTime</code>                    | 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） | 秒                      |
| 2                     | <code>CpuSystemTime</code>                  | 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） | 秒                      |
| 3                     | <code>ThreadsFree</code>                    | 可用 IQ 线程数                            | 不适用                    |
| 4                     | <code>ThreadsInUse</code>                   | 正在使用的 IQ 线程数                         | 不适用                    |
| 5                     | <code>MemoryAllocated</code>                | 分配的内存（以 MB 为单位）                      | MB                     |
| 6                     | <code>MemoryMaxAllocated</code>             | 分配的最大内存（以 MB 为单位）                    | MB                     |
| 7                     | <code>MainCacheCurrentSize</code>           | main 高速缓存当前大小（以 MB 为单位）              | MB                     |
| 8                     | <code>MainCacheFinds</code>                 | main 高速缓存查找请求总数                      | 不适用                    |
| 9                     | <code>MainCacheHits</code>                  | main 高速缓存命中总数                        | 不适用                    |
| 10                    | <code>MainCachePagesPinned</code>           | main 高速缓存已固定页数                       | 页码                     |
| 11                    | <code>MainCachePagesPinnedPercentage</code> | main 高速缓存已固定页的百分比                    | %                      |
| 12                    | <code>MainCachePagesDirtyPercentage</code>  | main 高速缓存脏页的百分比                      | %                      |
| 13                    | <code>MainCachePagesInUsePercentage</code>  | main 高速缓存正在使用的页的百分比                  | %                      |
| 14                    | <code>TempCacheCurrentSize</code>           | 临时高速缓存当前大小（以 MB 为单位）                 | MB                     |
| 15                    | <code>TempCacheFinds</code>                 | 临时高速缓存查找请求总数                         | 不适用                    |
| 16                    | <code>TempCacheHits</code>                  | 临时高速缓存命中总数                           | 不适用                    |
| 17                    | <code>TempCachePagesPinned</code>           | 临时高速缓存已固定页数                          | 页码                     |
| 18                    | <code>TempCachePagesPinnedPercentage</code> | 临时高速缓存已固定页的百分比                       | %                      |
| 19                    | <code>TempCachePagesDirtyPercentage</code>  | 临时高速缓存脏页的百分比                         | %                      |
| 20                    | <code>TempCachePagesInUsePercentage</code>  | 临时高速缓存正在使用的页的百分比                     | %                      |
| 21                    | <code>MainStoreDiskReads</code>             | 从 main 存储读取的字节数（以 KB 为单位）            | KB                     |

| stat_num | stat_name                           | stat_desc                  | stat_unit |
|----------|-------------------------------------|----------------------------|-----------|
| 22       | MainStoreDiskWrites                 | 写入 main 存储的字节数 (以 KB 为单位)  | KB        |
| 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

**示例 1** 显示单个统计的总 CPU 时间:

```
sp_iqstatistics 'CPUTotalTime'
```

**示例 2** 显示 MainCache% 的所有统计:

```
SELECT * from sp_iqstatistics() WHERE stat_name LIKE 'MainCache%'
```

## sp\_iqstatus 过程

作用

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。

说明

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 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。请参见第 384 页的“sp\_iqspaceused 过程”。

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。

**示例**

**注释** 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 dbspace 示例，您的数据库中可能不包括该示例。

以下输出来自 sp\_iqstatus 存储过程：

```

Sybase IQ (TM) Copyright (c) 1992-2009 by Sybase, Inc.
 All rights reserved.
Version: 15.1.0/090416/P/MS/Windows/2003/
 32bit/2009-04-16 02:11:41
Time Now: 2009-04-21 13:48:22.319
Build Time: 2009-04-16 02:15:39
File Format: 23 on 03/18/1999
Server mode: IQ Server
Catalog Format: 2
Stored Procedure Revision: 1
Page Size: 131072/8192blksz/16bpp
Number of Main DB Files : 2
Main Store Out Of Space: N
Number of Temp DB Files : 1
Temp Store Out Of Space: N
DB Blocks: 1-3200 IQ_SYSTEM_MAIN
DB Blocks: 1045440-1055039 iq_main
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
Temporary IQ Blocks Used: 65 of 800, 8%=0Mb, Max Block#: 0
Main Reserved Blocks Available: 1600 of 1600, 100%=6Mb
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: 压缩率

另请参见

[sp\\_iqtransaction](#) 过程和 [sp\\_iqversionuse](#) 过程。

《使用 Sybase IQ Multiplex》附录 A “Multiplex 引用”中的 “[sp\\_iqstatus](#) 过程”。

## sp\_iqsysmon 过程

作用

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 itime-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 **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。请参见表 7-58。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 7-58: 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        | memory        |
| 线程管理        | threads       |
| CPU 利用率     | cpu           |
| 事务管理        | txn           |
| 服务器上下文统计信息  | server        |
| 目录统计信息      | Catalog       |

**注释** 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

**start\_monitor** 开始监控。

**stop\_monitor** 停止监控并将剩余输出写入日志文件。

**filemode** 指定 `sp_iqsysmon` 在文件模式下运行。在文件模式下，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 `dbname.connid-iqmon` 的日志文件中。使用 `file_suffix` 选项可更改输出文件的后缀。有关 `file_suffix` 选项的说明，请参见 `monitor_options` 参数。

**monitor\_options** `monitor_options` 字符串可以包括以下一个或多个选项：

- **-interval seconds**  
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 `-interval` 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。

- **-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 或 -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。

- **-section section(s)**  
指定要写入监控器日志文件的一个或多个部分的缩写。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。有关缩写的列表，请参见表 7-58。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 7-59: sp\_iqsysmon 用法示例**

| 语法                                                                                                               | 结果                                          |
|------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
| sp_iqsysmon start_monitor<br>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<br>sp_iqsysmon stop_monitor<br>“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,<br>‘filemode’, “-interval 5 -sections<br>mbufpool memory”<br>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。 |

说明

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。

批处理模式下的 sp\_iqsysmon 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，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
RemovedFromWash 0 0 0 0 0 0 0 0 0
RemovedInScanMode 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 %)

```

另请参见

《参考：语句和选项》的第1章“SQL语句”中的 [IQ UTILITIES](#) 语句。  
《性能和调优指南》中的第5章“监控和调优性能”。

## sp\_iqtable 过程

作用

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`。

---

**注释** 在语法 1 中，*table\_type* 值 ALL 和 VIEW 必须用单引号引起来。

---

**语法 2** 参数可以任何顺序进行指定。将其用单引号引起来。

表 7-60 列出允许的 *table\_type* 参数值：

**表 7-60: sp\_iqtable 的 table\_type 值**

| <b>table_type 值</b> | <b>显示的信息</b> |
|---------------------|--------------|
| 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         |

说明

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 7-61: sp\_iqtable 列**

| 列名                | 说明                                                                                                                                         |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| table_name        | 表的名称。                                                                                                                                      |
| table_type        | BASE — 基表。<br>MAT VIEW — 物化视图。（仅 SA 表）<br>GBL TEMP — 全局临时表。<br>PARTITION — 表分区（该表仅供内部使用，不能被 Sybase IQ 用户使用）。<br>VIEW — 视图。<br>JVT — 连接虚拟表。 |
| table_owner       | 表的所有者                                                                                                                                      |
| server_type       | IQ — 在 IQ 存储中创建的对象。<br>SA — 在 SA 存储中创建的对象。<br>所有视图都是在 SA 存储中创建的。                                                                           |
| location          | TEMP — IQ 临时存储。<br>MAIN — IQ 存储。<br>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。                                                                                                                                     |

示例

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       | location |
|-------------|---------------|-------------|-------------------|----------|
| 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属于表所有者 GROUPO 的所有表：

```
sp_iqtable NULL, GROUPO
sp_iqtable table_owner='GROUPO'
```

| Table_name      | Table_type | Table_owner | Server_type | location |
|-----------------|------------|-------------|-------------|----------|
| Contacts        | BASE       | GROUPO      | IQ          | 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)            |
| 16387      | N             | (NULL)  | (NULL)            |

## sp\_iqtablesize 过程

**作用** 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。

**说明** 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。必须随此过程指定 `table_name` 参数。如果您是 `table_name` 的所有者，则不必指定 `table_owner` 参数。

**表 7-62: sp\_iqtablesiz 列**

| 列名              | 说明                   |
|-----------------|----------------------|
| 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。

**注释** 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_iqtablesiz ('dba.emp1')`

| Ownername | Tablename | Columns | KBytes | Pages | CompressedPages | NBlocks |
|-----------|-----------|---------|--------|-------|-----------------|---------|
| DBA       | emp1      | 4       | 1504   | 24    | 14              | 188     |

## sp\_iqtableuse 过程

|    |                                                   |
|----|---------------------------------------------------|
| 作用 | 报告负载访问的表的详细使用信息。                                  |
| 语法 | <b>sp_iqtableuse</b>                              |
| 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。 |
| 说明 | 不会报告在 SYSTEM 中创建的表。                               |

**表 7-63: sp\_iqtableuse 列**

| 列名        | 说明           |
|-----------|--------------|
| TableName | 表名           |
| 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 《性能和调优指南》中的“[监控负载](#)”第 3 章“[优化查询和删除](#)”。

第 305 页的“[sp\\_iqcolumnuse 过程](#)”、第 320 页的“[sp\\_iqdbspace 过程](#)”、第 348 页的“[sp\\_iqindexadvice 过程](#)”、第 355 页的“[sp\\_iqindexuse 过程](#)”、第 402 页的“[sp\\_iqunusedcolumn 过程](#)”、第 403 页的“[sp\\_iqunusedindex 过程](#)”、第 405 页的“[sp\\_iqunusedtable 过程](#)”和第 411 页的“[sp\\_iqworkmon 过程](#)”。

《参考：语句和选项》的第 2 章“[数据库选项](#)”中的“[INDEX\\_ADVISOR 选项](#)”。

## sp\_iqtransaction 过程

|    |                                                                                                                       |
|----|-----------------------------------------------------------------------------------------------------------------------|
| 作用 | 显示事务和版本的相关信息。                                                                                                         |
| 语法 | <b>sp_iqtransaction</b>                                                                                               |
| 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。                                                                     |
| 说明 | 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。

**注释** 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\_iqllocks 是实现此目的的更好选择。

**表 7-64: sp\_iqtransaction 列**

| 列名              | 说明                                                                                                                                                                                                                                                     |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name            | 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。                                                                                                                                                                                                                            |
| VersionID       | 在 Simplex 数据库中，VersionID 与 TxnID 相同。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。           |
| State           | 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。                                                                                                                         |
| 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。<br>对于 ACTIVE 事务，该列与 sp_iqconnection 的 TempWorkSpaceKB 列相同。 |

| 列名            | 说明                                                                                                               |
|---------------|------------------------------------------------------------------------------------------------------------------|
| 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           |

另请参见 [《使用 Sybase IQ Multiplex》附录 A “Multiplex 引用”](#) 中的“[sp\\_iqtransaction 过程](#)”。

### sp\_iqunusedcolumn 过程

- 作用** 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。
- 说明** 不会报告在 SYSTEM 中创建的表和本地临时表的列。

**表 7-65: sp\_iqunusedcolumn 列**

| 列名         | 说明       |
|------------|----------|
| TableName  | 表名       |
| ColumnName | 列名       |
| 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                  | GROUPO |
| SalesOrders     | CustomerID          | GROUPO |
| SalesOrders     | OrderDate           | GROUPO |
| SalesOrders     | FinancialCode       | GROUPO |
| SalesOrders     | Region              | GROUPO |
| SalesOrders     | SalesRepresentative | GROUPO |
| SalesOrderItems | ID                  | GROUPO |
| SalesOrderItems | LineID              | GROUPO |
| SalesOrderItems | ProductID           | GROUPO |
| SalesOrderItems | Quantity            | GROUPO |
| SalesOrderItems | ShipDate            | GROUPO |
| Contacts        | ID                  | GROUPO |
| Contacts        | Surname             | GROUPO |
| Contacts        | GivenName           | GROUPO |
| ...             |                     |        |

另请参见

《性能和调优指南》中的“监控负载”第3章“优化查询和删除”。

第305页的“sp\_iqcolumnuse 过程”、第320页的“sp\_iqdbspace 过程”、第348页的“sp\_iqindexadvice 过程”、第355页的“sp\_iqindexuse 过程”、第399页的“sp\_iqtableuse 过程”、第403页的“sp\_iqunusedindex 过程”、第405页的“sp\_iqunusedtable 过程”和第411页的“sp\_iqworkmon 过程”。

《参考：语句和选项》的第2章“数据库选项”中的“INDEX\_ADVISOR 选项”。

## sp\_iqunusedindex 过程

作用

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。

说明

不会报告在 SYSTEM 中创建的表或本地临时表的索引。

**表 7-66: sp\_iqunusedindex 列**

| 列名        | 说明   |
|-----------|------|
| IndexName | 索引名称 |
| TableName | 表名   |

| 列名        | 说明        |
|-----------|-----------|
| 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     | GROUPO | HG        |
| ASIQ_IDX_T450_C6_HG  | SalesOrders     | GROUPO | HG        |
| ASIQ_IDX_T450_C4_HG  | SalesOrders     | GROUPO | HG        |
| ASIQ_IDX_T450_C2_HG  | SalesOrders     | GROUPO | HG        |
| ASIQ_IDX_T451_I6_HG  | SalesOrderItems | GROUPO | HG        |
| ASIQ_IDX_T451_C3_HG  | SalesOrderItems | GROUPO | HG        |
| ASIQ_IDX_T451_C1_HG  | SalesOrderItems | GROUPO | HG        |
| ASIQ_IDX_T452_I11_HG | Contacts        | GROUPO | HG        |
| ASIQ_IDX_T453_I10_HG | Contacts        | GROUPO | HG        |
| ASIQ_IDX_T454_I4_HG  | FinancialCodes  | GROUPO | HG        |
| ASIQ_IDX_T455_I5_HG  | FinancialData   | GROUPO | HG        |
| ASIQ_IDX_T455_C3_HG  | FinancialData   | GROUPO | HG        |
| ASIQ_IDX_T456_I8_HG  | Products        | GROUPO | HG        |
| ASIQ_IDX_T457_I4_HG  | Departments     | GROUPO | HG        |
| ASIQ_IDX_T457_C3_HG  | Departments     | GROUPO | HG        |
| ASIQ_IDX_T458_I21_HG | Departments     | GROUPO | HG        |
| ASIQ_IDX_T458_C5_HG  | Departments     | GROUPO | HG        |

另请参见

《性能和调优指南》中的“[监控负载](#)”第3章“[优化查询和删除](#)”。

第305页的“[sp\\_iqcolumnuse](#)过程”、第320页的“[sp\\_iqdbspace](#)过程”、第348页的“[sp\\_iqindexadvice](#)过程”、第355页的“[sp\\_iqindexuse](#)过程”、第399页的“[sp\\_iqtableuse](#)过程”、第402页的“[sp\\_iqunusedcolumn](#)过程”、第405页的“[sp\\_iqunusedtable](#)过程”和第411页的“[sp\\_iqworkmon](#)过程”。

《参考：语句和选项》的第2章“[数据库选项](#)”中的“[INDEX\\_ADVISOR](#)选项”。

## sp\_iqunusedtable 过程

|    |                                                   |
|----|---------------------------------------------------|
| 作用 | 报告负载未引用的 IQ 表。                                    |
| 语法 | <b>sp_iqunusedtable</b>                           |
| 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。 |
| 说明 | 不会报告在 SYSTEM 中创建的表和本地临时表。                         |

**表 7-67: sp\_iqunusedtable 列**

| 列名        | 说明       |
|-----------|----------|
| TableName | 表名       |
| 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 |
| empl            | DBA    |
| SalesOrders     | GROUP0 |
| FinancialData   | GROUP0 |
| Departments     | GROUP0 |
| SalesOrderItems | GROUP0 |
| Products        | GROUP  |
| iq_dummy        | DBA    |
| Customers       | GROUP0 |
| sale            | DBA    |

另请参见 《性能和调优指南》中的“[监控负载](#)”第 3 章“[优化查询和删除](#)”。  
[第 305 页的“sp\\_iqcolumnuse 过程”](#)、[第 320 页的“sp\\_iqdbspace 过程”](#)、[第 348 页的“sp\\_iqindexadvice 过程”](#)、[第 355 页的“sp\\_iqindexuse 过程”](#)、[第 399 页的“sp\\_iqtableuse 过程”](#)、[第 402 页的“sp\\_iqunusedcolumn 过程”](#)、[第 403 页的“sp\\_iqunusedindex 过程”](#)和 [第 411 页的“sp\\_iqworkmon 过程”](#)。

《参考：语句和选项》的第 2 章“[数据库选项](#)”中的“[INDEX\\_ADVISOR 选项](#)”。

## sp\_iqversionuse 过程

- 作用** 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。
- 说明** `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。
- 如果出现空间不足的情况，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。有关 Multiplex 功能的信息，请参见《使用 Sybase IQ Multiplex》。
- 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 SQL Remote 将版本使用情况信息复制回事务协调器，`WasReported` 就更改为 1。如果 `WasReported` 在很长时间内是 0，则 SQL Remote 可能已停止。

**注释** `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 7-68: sp\_iqversionuse 列**

| 列名           | 说明                   |
|--------------|----------------------|
| VersionID    | 版本标识符                |
| Server       | 该版本的用户所连接的服务器        |
| IQConnID     | 使用该版本的连接 ID          |
| WasReported  | 指示服务器是否已收到该版本的使用情况信息 |
| MinKBRelease | 该版本不再使用时返回的最小空间量     |
| MaxKBRelease | 该版本不再使用时返回的最大空间量     |

示例

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

《使用 Sybase IQ Multiplex》附录 A “Multiplex 引用” 中的 “`sp_iqversionuse` 过程”。

## sp\_iqview 过程

作用

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，则必须使用 NULL 替代省略的参数。例如：`sp_iqview NULL,NULL,SYSTEM` 和 `sp_iqview deptview,NULL,'ALL'`。

---

**注释** 在语法 1 中，`view_type` 值 ALL 必须用单引号引起来。

---

**语法 2** 参数可以任何顺序进行指定。将其用单引号引起来。

表 7-69 列出允许的 `view_type` 参数值。

**表 7-69: 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    |

**说明** 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 7-70: sp\_iqview 列**

| 列名         | 说明                       |
|------------|--------------------------|
| 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。

**示例** 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 ,<br>ViewSalesOrders( ID, LineID,<br>ProductID, Quantity,<br>OrderDate, ShipDate, Region,<br>SalesRepresentativeName | (NULL)  |

## sp\_iqwho 过程

**作用** 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。

**说明** 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 7-71: sp\_iqwho 列

| 列名               | 说明                                                                                    |
|------------------|---------------------------------------------------------------------------------------|
| 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。                                                             |
| 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。               |
| Q Cursors        | 连接中的活动游标数；如果没有游标则为 0。                                                                 |
| QThreads         | 连接的线程数。连接打开后至少会立即启动一个线程，因此 QThreads 的最小值为 1。                                          |
| TempTableSpaceKB | 临时表空间的大小（以 KB 为单位）；如果没有使用临时表空间则为 0。                                                   |
| TempWorkSpaceKB  | 临时工作空间的大小（以 KB 为单位）；如果没有使用临时工作空间则为 0。                                                 |

**Adaptive Server Enterprise 兼容性** Sybase IQ sp\_iqwho 存储过程包含由 Adaptive Server Enterprise sp\_who 过程显示的列的 Sybase IQ 等效项。省略了一些 Adaptive Server Enterprise 列，因为它们不适用于 Sybase IQ。表 7-72 指出 Adaptive Server Enterprise sp\_who 列与 sp\_iqwho 显示的列的对应关系。

**表 7-72: 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**

**connhandle** 表示连接 ID 的整数。如果指定该参数, 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。如果指定该参数, 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 user-name does not exist”

**arg-type** *arg-type* 参数是可选参数, 仅在指定了第一个参数后才可指定该参数。*arg-type* 的值只能是 “user”。如果将 *arg-type* 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。如果将 “user” 以外的任何值指定给 *arg-type*, sp\_iqwho 会返回错误

“Invalid parameter.”

请用双引号将 *arg-type* 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"
```

如果将 *arg-type* 的值指定为 “user”，`sp_iqwho` 会将第一个参数解释为用户名，而不是连接 ID。如果数据库中包含名为 1 的用户，`sp_iqwho` 将显示有关由用户 1 打开的连接的信息。

**表 7-73: `sp_iqwho` 用法示例**

| 语法                                      | 输出                                                                 |
|-----------------------------------------|--------------------------------------------------------------------|
| <code>sp_iqwho</code>                   | 显示所有活动连接                                                           |
| <code>sp_iqwho 3</code>                 | 显示连接 3 的相关信息                                                       |
| <code>sp_iqwho “DBA”</code>             | 显示由用户 DBA 打开的连接                                                    |
| <code>sp_iqwho 3, “user”</code>         | 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” |
| <code>sp_iqwho non-existing-user</code> | 返回错误 “User non-existing-user does not exist”                       |
| <code>sp_iqwho 3, “xyz”</code>          | 返回错误 “Invalid parameter: xyz”                                      |

示例

显示所有活动连接：

```

ConnHandle IQConnID Userid ReqType IQCmdType BlockedOn BlockUserid IQCu
rsors
IQThreads IQIdle SAIdle TempTableSpaceKB TempWorkSpaceKB

12 118 DBA CURSOR_OPEN IQUTILITYOPENCURSOR 0 (NULL) 0
1 1 0 0 0
13 119 shweta DO_NOTHING NONE 0 (NULL) 0
1 16238757 470 0 0

```

另请参见

[第 306 页的 “`sp\_iqconnection` 过程”。](#)

[第 414 页的 “`sa\_conn\_info` 系统过程”。](#)

## `sp_iqworkmon` 过程

作用

控制对负载监控器使用信息的收集，并报告监控收集状态。

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 'start', 'all'

```

**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** **action** 指定要应用的控制操作。 *start* 值针对指定模式立即启动监控。 *stop* 值立即停止监控。 *status* 值（缺省值）显示当前状态，而不会更改状态。

始终会保留统计信息，除非使用 *reset* 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启后自动恢复，需要使用 *start* 重启该集合。

**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 7-74: sp\_iqworkmon 列**

| 列名       | 说明      |
|----------|---------|
| MonMode  | 表、索引或列  |
| Status   | 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 《性能和调优指南》中的“[监控负载](#)”第 3 章“[优化查询和删除](#)”。

第 305 页的“[sp\\_iqcolumnuse 过程](#)”、第 348 页的“[sp\\_iqindexadvice 过程](#)”、第 320 页的“[sp\\_iqdbspace 过程](#)”、第 355 页的“[sp\\_iqindexuse 过程](#)”、第 399 页的“[sp\\_iqtableuse 过程](#)”、第 402 页的“[sp\\_iqunusedcolumn 过程](#)”、第 403 页的“[sp\\_iqunusedindex 过程](#)”和第 405 页的“[sp\\_iqunusedtable 过程](#)”。

《参考：语句和选项》的第 2 章“[数据库选项](#)”中的“[INDEX\\_ADVISOR 选项](#)”。

## 目录存储过程

以下目录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 过程

**作用** 返回某个给定表或视图的所有相关视图的列表。

**语法** `sa_ansi_standard_packages ( sql-standard-string, sql-statement-string )`

请参见 SQL Anywhere 文档中的“sa\_ansi\_standard\_packages 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

### sa\_audit\_string 系统过程

**作用** 将字符串添加到事务日志中。

**语法** `sa_audit_string ( string )`

请参见 SQL Anywhere 文档中的“sa\_audit\_string 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

### sa\_checkpoint\_execute 系统过程

**作用** 允许在检查点操作期间执行 shell 命令。

**语法** `sa_checkpoint_execute 'shell_commands'`

**参数** **shell\_commands** 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。

**说明** 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，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

作用

返回指向服务器上数据库的每个连接的最近预准备 SQL 语句。

语法

```
sa_conn_activity ([connidparm])
```

请参见 SQL Anywhere 文档中的“`sa_conn_activity` 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_conn\_info 系统过程

作用

报告连接属性信息。

语法

```
sa_conn_info ([connidparm])
```

请参见 SQL Anywhere 文档中的“`sa_conn_info` 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_conn\_properties 系统过程

作用 报告连接属性信息。

语法 **sa\_conn\_properties** ( [ *connidparm* ] )

请参见 SQL Anywhere 文档中的 “sa\_conn\_info 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_db\_info 系统过程

作用 报告数据库属性信息。

语法 **sa\_db\_info** ( [ *dbidparm* ] )

请参见 SQL Anywhere 文档中的 “sa\_db\_info 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_db\_properties 系统过程

作用 报告数据库属性信息。

语法 **sa\_db\_properties** ( [ *dbidparm* ] )

请参见 SQL Anywhere 文档中的 “sa\_db\_properties 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_disable\_auditing\_type 系统过程

作用 禁用特定事件的审计。

语法 **sa\_disable\_auditing\_type**( [ *types* ] )

请参见 SQL Anywhere 文档中的 “sa\_disable\_auditing\_type 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_enable\_auditing\_type 系统过程

作用 启用审计并指定要审计哪些事件。

语法 **sa\_enable\_auditing\_type**( [types] )

请参见 SQL Anywhere 文档中的“sa\_enable\_auditing\_type 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_eng\_properties 系统过程

作用 报告数据库服务器属性信息。

语法 **sa\_eng\_properties**( )

请参见 SQL Anywhere 文档中的“sa\_eng\_properties 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_flush\_cache 系统过程

作用 清空数据库服务器高速缓存中当前数据库的所有页。

语法 **sa\_flush\_cache**( )

请参见 SQL Anywhere 文档中的“sa\_flush\_cache 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_make\_object 系统过程

作用 确保在执行 ALTER 语句之前对象的框架实例已经存在。

语法 **sa\_make\_object** ( objtype, objname [, owner [, tablename ] ] )  
*object-type*: 'procedure' | 'function' | 'view' | 'trigger' | 'service' | 'event'

请参见 SQL Anywhere 文档中的“sa\_make\_object 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_rowgenerator 系统过程

作用 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* ] ] ] )

请参见 SQL Anywhere 文档中的“sa\_rowgenerator 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_server\_option 系统过程

作用 当服务器运行时覆盖服务器选项。

语法 **sa\_server\_option** ( *opt*, *val* )

说明 请参见 SQL Anywhere 文档中的“sa\_server\_option 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

OptionWatchList 选项在 Sybase IQ 中具有唯一定义：

**OptionWatchList** 指定希望获得相关通知的数据库选项的逗号分隔列表，或者在设置这些选项时让数据库服务器返回错误。字符串长度限制为 128 个字节。缺省情况下为空字符串。

```
CALL dbo.sa_server_option('OptionWatchList')
```

可以通过执行以下查询来查看此属性的当前设置：

```
SELECT DB_PROPERTY('OptionWatchList');
```

## sa\_set\_http\_header 系统过程

作用 允许 Web 服务设置结果中的 HTTP 标头。

语法 **sa\_set\_http\_header** ( *fld-name*, *val* )

请参见 SQL Anywhere 文档中的“sa\_set\_http\_header 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_set\_http\_option 系统过程

作用 允许 Web 服务设置结果中的 HTTP 选项。

语法 **sa\_set\_http\_option** ( *option-name*, *value* )

请参见 SQL Anywhere 文档中的 “sa\_set\_http\_option 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_table\_page\_usage 系统过程

作用 报告有关数据库表的页面使用情况的信息。

语法 **sa\_table\_page\_usage**( )

请参见 SQL Anywhere 文档中的 “sa\_table\_page\_usage 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_validate 系统过程

作用 检验 Catalog 存储中的所有表。有关检验的详细信息，请参见《实用程序指南》的第 4 章 “数据库管理实用程序” 中的 “验证实用程序 (dbvalid)”。

语法 **sa\_validate** [[ *owner\_name*. ] [ *tbl\_name* | *owner\_name* ]

请参见 SQL Anywhere 文档中的 “sa\_validate 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sa\_verify\_password 系统过程

作用 检验当前用户的口令。

语法 **sa\_verify\_password** ( *curr\_pswd* )

请参见 SQL Anywhere 文档中的 “sa\_verify\_password 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sp\_login\_environment 系统过程

**作用** 在用户登录时设置连接选项。缺省情况下，sp\_login\_environment 系统过程会对 TDS 连接调用 sp\_tsql\_environment 系统过程。

**语法** `sp_login_environment( )`

请参见 SQL Anywhere 文档中的“sp\_login\_environment 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sp\_remote\_columns 系统过程

**作用** 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 文档中的“sa\_verify\_password 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

**另请参见** 《系统管理指南，卷 2》中的第 5 章“用于进行远程数据访问的服务器类”和第 4 章“访问远程数据”

《参考：语句和选项》中的 [CREATE SERVER 语句](#)

## sp\_remote\_exported\_keys 系统过程

**作用** 提供其外键在指定主键表上的表的相关信息。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 ]])`

请参见 SQL Anywhere 文档中的“sp\_remote\_exported\_keys 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

**另请参见** 《系统管理指南，卷 2》中的第 4 章“访问远程数据”和第 5 章“用于进行远程数据访问的服务器类”

《参考：语句和选项》中的 [CREATE SERVER 语句](#)

## sp\_remote\_imported\_keys 系统过程

**作用** 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 ]])`

请参见 SQL Anywhere 文档中的“sp\_remote\_imported\_keys 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

**另请参见** 《系统管理指南，卷 2》中的第 4 章“访问远程数据”和第 5 章“用于进行远程数据访问的服务器类”。

《参考：语句和选项》中的 CREATE SERVER 语句

## sp\_remote\_primary\_keys 系统过程

**作用** 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 文档中的“sp\_remote\_imported\_keys 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

## sp\_remote\_tables 系统过程

**作用** 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 ( @servername [, @tablename [, @table_owner [, @table_qualifier [, @with_table_type ]]])`

请参见 SQL Anywhere 文档中的“sp\_remote\_tables 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

**另请参见** 《参考：语句和选项》中的 CREATE SERVER 语句。

《系统管理指南，卷 2》中的第 4 章“访问远程数据”和第 5 章“用于进行远程数据访问的服务器类”。

## sp\_servercaps 系统过程

**作用** 显示有关远程服务器的功能的信息。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 )`

请参见 SQL Anywhere 文档中的“sp\_servercaps 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

**另请参见** 《参考：语句和选项》中的 [CREATE SERVER 语句](#)。

《系统管理指南，卷 2》中的第 4 章“访问远程数据”和第 5 章“用于进行远程数据访问的服务器类”。

## sp\_tsq\_environment 系统过程

**作用** 设置用户从 jConnect 或 Open Client 应用程序连接时的连接选项。

**语法** `sp_tsq_environment( )`

请参见 SQL Anywhere 文档中的“sp\_tsq\_environment 系统过程”，具体位置为：“SQL Anywhere 11.0.1” > “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”。

**另请参见** 《参考：语句和选项》的第 2 章“数据库选项”中的 [LOGIN\\_PROCEDURE 选项](#)。

## Adaptive Server Enterprise 系统和目录存储过程

Adaptive Server Enterprise 提供了系统过程和目录存储过程来执行许多管理功能并获得系统信息。Sybase IQ 已实现对这些过程中的某些过程的支持。

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。目录存储过程以表格形式从系统表中检索信息。

**注释** 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`。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'
```

## Adaptive Server Enterprise 系统过程

表 7-75 说明了在 Sybase IQ 中提供的 Adaptive Server Enterprise 系统过程。

**表 7-75: Sybase IQ 中提供的 ASE 系统过程**

| 系统过程                                                                            | 说明                                                       | 权限                                                                                       |
|---------------------------------------------------------------------------------|----------------------------------------------------------|------------------------------------------------------------------------------------------|
| <code>sp_addgroup group-name</code>                                             | 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。       |
| <code>sp_addlogin userid, password[, defdb [, deflanguage [, fullname]]]</code> | 将新用户帐户添加到数据库中                                            | 需要具有 DBA 或 USER ADMIN 权限。                                                                |
| <code>sp_addmessage message-num, message_text [, language]</code>               | 向 SYSUSERMESSAGES 添加用户定义消息，以供存储过程 PRINT 和 RAISERROR 调用使用 | 需要具有 DBA 或 RESOURCE 权限。                                                                  |
| <code>sp_addtype typename, datatype, [, "identity"   nulltype]</code>           | 创建用户定义的数据类型。Sybase IQ 不支持 IDENTITY 列。                    | 需要具有 DBA 或 RESOURCE 权限。                                                                  |
| <code>sp_adduser userid [, name_in_db [, grpname]]</code>                       | 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。 |
| <code>sp_changegroup new-group-name, userid</code>                              | 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。                                                               |
| <code>sp_dboption [dbname, optname, {true   false}]</code>                      | 显示或更改数据库选项                                               | 不需要。                                                                                     |
| <code>sp_dropgroup group-name</code>                                            | 从数据库中删除组                                                 | 需要具有 DBA 或 PERMS ADMIN 权限。                                                               |
| <code>sp_droplogin userid</code>                                                | 从数据库中删除用户                                                | 需要具有 DBA 或 USER ADMIN 权限。                                                                |

| 系统过程                                                                 | 说明                                                      | 权限                                                                          |
|----------------------------------------------------------------------|---------------------------------------------------------|-----------------------------------------------------------------------------|
| <code>sp_dropmessage message-number [, language]</code>              | 删除用户定义的消息                                               | 需要具有 DBA 或 RESOURCE 权限。                                                     |
| <code>sp_droptype typename</code>                                    | 删除用户定义的数据类型                                             | 需要具有 DBA 或 RESOURCE 权限。                                                     |
| <code>sp_dropuser userid</code>                                      | 从数据库中删除用户                                               | 需要具有 DBA 或 USER ADMIN 权限。                                                   |
| <code>sp_getmessage message-num, @msg-var output [, language]</code> | 从 SYSUSERMESSAGES 中检索存储消息字符串, 供 PRINT 和 RAISERROR 语句使用。 | 不需要。                                                                        |
| <code>sp_helptext 'owner.object-name'</code>                         | 显示系统过程或视图的文本                                            | 不需要。                                                                        |
| <code>sp_password caller_passwd, new_passwd [, userid]</code>        | 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。 |

**注释** 像 `sp_dropuser` 这样的过程提供与 Adaptive Server Enterprise 存储过程的最小兼容性。如果习惯于 Adaptive Server Enterprise (或 Sybase IQ 11.x) 存储过程, 则在 `dbisql` 中使用 Sybase IQ 12 过程之前, 请比较两者的文本。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, 所有者始终是 `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'
```

## Adaptive Server Enterprise 目录存储过程

除了 `sp_column_privileges` 过程以外, Sybase IQ 实现大多数 Adaptive Server Enterprise 目录存储过程。表 7-76 中说明了所实现的目录存储过程。对于某些 Adaptive Server 目录存储过程, Sybase IQ 还具有相似的自定义存储过程。

**表 7-76: Sybase IQ 中实现的 ASE 目录存储过程**

| ASE 目录存储过程                                                                                                                          | 说明         | IQ 过程 |
|-------------------------------------------------------------------------------------------------------------------------------------|------------|-------|
| <code>sp_columns table-name [, table-owner] [, table-qualifier] [, column-name]</code>                                              | 返回指定列的数据类型 |       |
| <code>sp_fkeys phtable_name [, phtable-owner] [, phtable-qualifier] [, fktable-name] [, fktable_owner] [, fktable-qualifier]</code> | 返回指定表的外键信息 |       |

| ASE 目录存储过程                                                                                         | 说明                   | IQ 过程                       |
|----------------------------------------------------------------------------------------------------|----------------------|-----------------------------|
| <code>sp_pkeys table-name [, table_owner ]<br/>[, table_qualifier ]</code>                         | 返回单个表的主键信息           | <code>sp_iqpkeys</code>     |
| <code>sp_special_columns table_name [, table-owner ]<br/>[, table-qualifier ] [, col-type ]</code> | 返回唯一标识表中某一行的最佳列集     |                             |
| <code>sp_sproc_columns proc-name [, proc_owner ]<br/>[, proc-qualifier ] [, column-name ]</code>   | 返回有关存储过程的输入和返回参数的信息  | <code>sp_iqprocparm</code>  |
| <code>sp_stored_procedures [ sp-name ] [, sp-owner ]<br/>[, sp-qualifier ]</code>                  | 返回有关一个或多个存储过程的信息     | <code>sp_iqprocedure</code> |
| <code>sp_tables table-name [, table-owner ]<br/>[, table-qualifier ] [, table-type ]</code>        | 返回可出现在 FROM 子句中的对象列表 |                             |

不支持以下 Adaptive Server Enterprise 目录存储过程：

- `sp_column_privileges`
- `sp_databases`
- `sp_datatype_info`
- `sp_server_info`

## SQL Anywhere 支持的过程

Sybase IQ 支持 SQL Anywhere 文档中“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “系统过程”中“按字母顺序排列的系统过程列表”中列出的 SQL Anywhere 过程。

`sa_get_table_definition` 过程仅受 SQL Anywhere 表支持。如果对 IQ 表运行此过程，则此过程会返回错误 `not implemented for IQ tables`。

**注释** 在《SQL Anywhere Server - SQL 参考》的“系统扩展存储过程”一节中，规定用户必须被授予 `EXECUTE` 权限或拥有 `DBA` 授权。但是，多个后续过程显示列为“无”的权限。

以下过程需要具有 `DBA` 权限。

- `xp_cmdshell` — 允许数据库服务器执行外部 shell 命令的系统扩展过程

## 关于本章

本章介绍系统表、系统视图、统一视图、兼容性视图以及 Sybase IQ 支持的 ASE T-SQL 兼容性视图。

## 目录

| 主题                                | 页码  |
|-----------------------------------|-----|
| 系统表                               | 429 |
| 系统表列表                             | 429 |
| DUMMY 系统表                         | 432 |
| 系统视图                              | 432 |
| SYSARTICLE 系统视图                   | 433 |
| SYSARTICLECOL 系统视图                | 433 |
| SYSARTICLECOLS 统一视图               | 433 |
| SYSARTICLES 统一视图                  | 433 |
| SYSCAPABILITIES 统一视图              | 434 |
| SYSCAPABILITY 系统视图                | 434 |
| SYSCAPABILITYNAME 系统视图            | 434 |
| SYSCATALOG 统一视图                   | 434 |
| SYSCHECK 系统视图                     | 435 |
| SYSCOLAUTH 统一视图                   | 435 |
| SYSCOLLATION 兼容性视图（不建议使用）         | 435 |
| SYSCOLLATIONMAPPINGS 兼容性视图（不建议使用） | 435 |
| SYSCOLPERM 系统视图                   | 436 |
| SYSCOLSTAT 系统视图                   | 436 |
| SYSCOLSTATS 统一视图                  | 436 |
| SYSCOLUMN 兼容性视图（不建议使用）            | 437 |
| SYSCOLUMNS 统一视图                   | 437 |
| SYSCOLUMNS ASE 兼容性视图              | 437 |
| SYSCOMMENTS ASE 兼容性视图             | 437 |
| SYSCONSTRAINT 系统视图                | 438 |
| SYSDBFILE 系统视图                    | 438 |
| SYSDBSPACE 系统视图                   | 438 |
| SYSDBSPACEPERM 系统视图               | 439 |
| SYSDEPENDENCY 系统视图                | 439 |



| 主题                        | 页码  |
|---------------------------|-----|
| SYSIQTABLE 系统视图（不建议使用）    | 456 |
| SYSIQVINDEX ASE 兼容性视图     | 456 |
| SYSIXCOL 兼容性视图（不建议使用）     | 456 |
| SYSJAR 系统视图               | 457 |
| SYSJARCOMPONENT 系统视图      | 457 |
| SYSJAVACLASS 系统视图         | 457 |
| SYSLOGINMAP 系统视图          | 457 |
| SYSLOGINPOLICY 系统视图       | 458 |
| SYSLOGINPOLICYOPTION 系统视图 | 458 |
| SYSLOGINS ASE 兼容性视图       | 458 |
| SYSMVOPTION 系统视图          | 458 |
| SYSMVOPTIONNAME 系统视图      | 459 |
| SYSOBJECT 系统视图            | 459 |
| SYSOBJECTS ASE 兼容性视图      | 459 |
| SYSOPTION 系统视图            | 459 |
| SYSOPTIONS 统一视图           | 460 |
| SYSOPTSTAT 系统视图           | 460 |
| SYSPARTITION 系统视图         | 460 |
| SYSPARTITIONKEY 系统视图      | 461 |
| SYSPARTITIONScheme 系统视图   | 461 |
| SYSPHYSIDX 系统视图           | 462 |
| SYSPROCAUTH 统一视图          | 462 |
| SYSPROCEDURE 系统视图         | 462 |
| SYSROCPARM 系统视图           | 463 |
| SYSROCPARMS 统一视图          | 463 |
| SYSROCPERM 系统视图           | 463 |
| SYSROCS 统一视图              | 463 |
| SYSROXYTAB 系统视图           | 464 |
| SYSROBICATION 系统视图        | 464 |
| SYSROBICATIONS 统一视图       | 464 |
| SYSREMARK 系统视图            | 464 |
| SYSREMOPTION 系统视图         | 465 |
| SYSREMOPTION2 统一视图        | 465 |
| SYSREMOPTIONS 统一视图        | 465 |
| SYSREMOPTIONTYPE 系统视图     | 466 |
| SYSREMOTOTYPE 系统视图        | 466 |
| SYSREMOTOTYPEs 统一视图       | 466 |

| 主题                              | 页码  |
|---------------------------------|-----|
| SYSREMOTEUSER 系统视图              | 466 |
| SYSREMOTEUSERS 统一视图             | 467 |
| SYSSCHEDULE 系统视图                | 467 |
| SYSSERVER 系统视图                  | 467 |
| SYSSOURCE 系统视图                  | 467 |
| SYSSQLSERVERTYPE 系统视图           | 468 |
| SYSSUBPARTITIONKEY 系统视图         | 468 |
| SYSSUBSCRIPTION 系统视图            | 468 |
| SYSSUBSCRIPTIONS 统一视图           | 468 |
| SYSSYNC 系统视图                    | 469 |
| SYSSYNC2 统一视图                   | 469 |
| SYSSYNCPUBLICATIONDEFAULTS 统一视图 | 469 |
| SYSSYNCS 统一视图                   | 469 |
| SYSSYNCSCRIPT 系统视图              | 470 |
| SYSSYNCSCRIPTS 统一视图             | 470 |
| SYSSYNCSUBSCRIPTIONS 统一视图       | 470 |
| SYSSYNCSUSERS 统一视图              | 470 |
| SYSTAB 系统视图                     | 471 |
| SYSTABLE 兼容性视图（不建议使用）           | 471 |
| SYSTABAUTH 统一视图                 | 471 |
| SYSTABCOL 系统视图                  | 471 |
| SYSTABLEPERM 系统视图               | 472 |
| SYSTEMTEXTCONFIG 系统视图           | 472 |
| SYSTEMTEXTIDX 系统视图              | 472 |
| SYSTEMTEXTIDXTAB 系统视图           | 473 |
| SYSTRIGGER 系统视图                 | 473 |
| SYSTRIGGERS 统一视图                | 473 |
| SYSTYPEMAP 系统视图                 | 473 |
| SYSTYPES ASE 兼容性视图              | 474 |
| SYSUSER 系统视图                    | 474 |
| SYSUSERAUTH 兼容性视图（不建议使用）        | 474 |
| SYSUSERAUTHORITY 系统视图           | 474 |
| SYSUSERLIST 兼容性视图（不建议使用）        | 475 |
| SYSUSERMESSAGE 系统视图             | 475 |
| SYSUSEROPTIONS 统一视图             | 475 |
| SYSUSERPERM 兼容性视图（不建议使用）        | 475 |
| SYSUSERPERMS 兼容性视图（不建议使用）       | 476 |

| 主题                                 | 页码  |
|------------------------------------|-----|
| <a href="#">SYSUSERTYPE 系统视图</a>   | 476 |
| <a href="#">SYSUSERS ASE 兼容性视图</a> | 476 |
| <a href="#">SYSVIEW 系统视图</a>       | 476 |
| <a href="#">SYSVIEWS 统一视图</a>      | 477 |
| <a href="#">SYSWEBSERVICE 系统视图</a> | 477 |
| <a href="#">Transact-SQL 兼容性视图</a> | 477 |

## 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。

## 系统表列表

表 8-1 列出所有 Sybase IQ 系统表。

**表 8-1：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       |

| 系统表                        | 仅在内部使用? |
|----------------------------|---------|
| ISYSEXTERNENV              | 是       |
| ISYSEXTERNENVOBJECT        | 是       |
| ISYSEXTERNLOGIN            | 是       |
| ISYSFKEY                   | 是       |
| ISYSGROUP                  | 是       |
| ISYSHISTORY                | 是       |
| ISYSIDX                    | 是       |
| ISYSIDXCOLUMN              | 是       |
| ISYSIQBACKUPHISTORY        | 是       |
| ISYSIQBACKUPHISTORYDETAIL  | 是       |
| ISYSIQDBFILE               | 是       |
| ISYSIQDBSPACE              | 是       |
| ISYSIQIDX                  | 是       |
| ISYSIQINFO                 | 是       |
| ISYSIQJOINIDX              | 是       |
| ISYSIQJOINIXCOLUMN         | 是       |
| ISYSIQMPXLOGINPOLICYOPTION | 是       |
| ISYSIQMPXSERVER            | 是       |
| ISYSIQPARTITIONCOLUMN      | 是       |
| ISYSIQTAB                  | 是       |
| ISYSIQTABCOL               | 是       |
| ISYSJAR                    | 是       |
| ISYSJARCOMPONENT           | 是       |
| ISYSJAVACLASS              | 是       |
| ISYSLOGINMAP               | 是       |
| ISYSLOGINPOLICY            | 是       |
| ISYSLOGINPOLICYOPTION      | 是       |
| ISYSMVOPTION               | 是       |
| ISYSMVOPTIONNAME           | 是       |
| ISYSOBJECT                 | 是       |
| ISYSOPTION                 | 是       |
| ISYSOPTSTAT                | 是       |
| ISYSPARTITION              | 是       |
| ISYSPARTITIONKEY           | 是       |
| ISYSPARTITIONSCHEME        | 是       |
| ISYSPHYSIDX                | 是       |
| ISYSPROCEDURE              | 是       |

| 系统表                  | 仅在内部使用? |
|----------------------|---------|
| ISYSPROCPARM         | 是       |
| ISYSPROCPERM         | 是       |
| ISYSPROXYTAB         | 是       |
| ISYSPUBLICATION      | 是       |
| ISYSREMARK           | 是       |
| ISYSREMOTEOPTION     | 是       |
| ISYSREMOTEOPTIONTYPE | 是       |
| ISYSREMOTETYPE       | 是       |
| ISYSREMOTEUSER       | 是       |
| ISYSSCHEDULE         | 是       |
| ISYSSEVER            | 是       |
| ISYSSOURCE           | 是       |
| ISYSSQLSERVERTYPE    | 是       |
| ISYSSUBPARTITIONKEY  | 是       |
| ISYSSUBSCRIPTION     | 是       |
| ISYSSYNC             | 是       |
| 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       |

## DUMMY 系统表

DUMMY 表作为一个始终只有一行的表提供。它对于从数据库中提取信息很有用，如下例所示，该示例从数据库中获得当前用户 ID 和当前日期。

```
SELECT USER, today(*) FROM SYS.DUMMY
```

DUMMY 表是 SQL Anywhere 系统表。有关详细信息（包括指向其 SQL Anywhere 系统视图的链接），请参见 SQL Anywhere 文档中的“DUMMY 系统表”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “表” > “系统表”。

## 系统视图

若要查看系统表的内容，请使用系统视图。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。

### 关于统一视图

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。有关这些视图和其他统一视图的定义，请参见 SQL Anywhere 文档中的“统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图”。

### 关于兼容性视图

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 文档中的“兼容性视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图”。

### 关于 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。请参见第 477 页的“[Transact-SQL 兼容性视图](#)”。

## SYSARTICLE 系统视图

SYSARTICLE 系统视图的每一行均描述了发布中的一个项目。此视图的基础系统表为 ISYSARTICLE。

SYSARTICLE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSARTICLE 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSARTICLECOL 系统视图

SYSARTICLECOL 系统视图的每一行均标识项目中的一列。此视图的基础系统表为 ISYSARTICLECOL。

SYSARTICLECOL 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSARTICLECOL 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSARTICLECOLS 统一视图

该视图中的每一行均标识项目中的一列。

SYSARTICLECOLS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSARTICLECOLS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSARTICLES 统一视图

SYSARTICLES 视图中的每一行均描述了发布中的一个项目。

SYSARTICLES 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSARTICLES 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSCAPABILITIES 统一视图

SYSCAPABILITIES 视图中的每一行均描述一种功能。此视图从 ISYSCAPABILITY 和 ISYSCAPABILITYNAME 系统表获取数据。

SYSCAPABILITIES 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSCAPABILITIES 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSCAPABILITY 系统视图

SYSCAPABILITY 系统视图的每一行均标识远程服务器的一种功能。此视图的基础系统表为 ISYSCAPABILITY。

SYSCAPABILITY 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSCAPABILITY 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSCAPABILITYNAME 系统视图

SYSCAPABILITYNAME 系统视图的每一行均指定一种在 SYSCAPABILITY 系统视图中定义的功能。

SYSCAPABILITYNAME 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSCAPABILITYNAME 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSCATALOG 统一视图

SYSCATALOG 视图的每一行均描述一个系统表。

SYSCATALOG 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSCATALOG 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSCHECK 系统视图

SYSCHECK 系统视图的每一行均提供表中的一个命名检查约束的定义。此视图的基础系统表为 ISYSCHECK。

SYSCHECK 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSCHECK 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSCOLAUTH 统一视图

SYSCOLAUTH 视图中的每一行均描述了被授予的对列的一组权限（UPDATE、SELECT 或 REFERENCES）。借助 SYSCOLAUTH 视图，可以很方便地显示 SYSCOLPERM 系统视图中的数据。

SYSCOLAUTH 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSCOLAUTH 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSCOLLATION 兼容性视图（不建议使用）

SYSCOLLATION 兼容性视图包含相应数据库的归类序列信息。此视图可以通过内置函数获得，且不会保留在目录中。

SYSCOLLATION 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSCOLLATION 兼容性视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “视图” > “兼容性视图”。

## SYSCOLLATIONMAPPINGS 兼容性视图（不建议使用）

SYSCOLLATIONMAPPINGS 兼容性视图只包含一个具有数据库归类映射的行。此视图可以通过内置函数获得，且不会保留在目录中。

SYSCOLLATIONMAPPINGS 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSCOLLATIONMAPPINGS 兼容性视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “兼容性视图”。

## SYSCOLPERM 系统视图

GRANT 语句可以向表中的各个列分别授予 UPDATE、SELECT 或 REFERENCES 权限。具有 UPDATE、SELECT 或 REFERENCES 权限的每一列均记录在 SYSCOLPERM 系统视图的一行中。此视图的基础系统表为 ISYSCOLPERM。

SYSCOLPERM 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSCOLPERM 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSCOLSTAT 系统视图

SYSCOLSTAT 系统视图包含优化程序所使用的列统计信息（包括直方图）。使用 sa\_get\_histogram 存储过程或直方图实用程序可以有效地检索此视图的内容。此视图的基础系统表为 ISYSCOLSTAT。

此视图的基础系统表始终是加密的，以保护数据免受未经授权的访问。

SYSCOLSTAT 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSCOLSTAT 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSCOLSTATS 统一视图

SYSCOLSTATS 视图包含存储为直方图、由优化程序使用的列统计信息。

SYSCOLSTATS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSCOLSTATS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## 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 系统表的较旧版本的 Sybase IQ 兼容。但是，以前的 SYSCOLUMN 表已由 ISYSTABCOL 系统表及其相应的 SYSTABCOL 系统视图取代，您应改用它们。

SYSCOLUMN 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSCOLUMN 兼容性视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “兼容性视图”。

## SYSCOLUMNS 统一视图

SYSCOLUMNS 视图中的每一行均描述了目录中的每个表和视图的一列。

SYSCOLUMNS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSCOLUMNS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## 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。请参见第 478 页的表 8-2。

## SYSCOMMENTS ASE 兼容性视图

此视图由用户 DBO 所有。syscomments 包含与每个视图、规则、缺省值、触发器、表约束、分区、过程、计算列、基于函数的索引键以及其它形式的已编译对象相对应的条目。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。每个对象最多可以占用 65,025 行。请参见第 478 页的表 8-2。

## SYSCONSTRAINT 系统视图

SYSCONSTRAINT 系统视图中的每一行均描述了数据库中的一个命名约束。此视图的基础系统表为 ISYSCONSTRAINT。

SYSCONSTRAINT 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSCONSTRAINT 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSDBFIL 系统视图

SYSDBFIL 系统视图的每一行均描述了一个 dbspace 文件。此视图的基础系统表为 ISYSDBFIL。

---

**注释** 此视图将取代不建议使用的 SYSDBFIL 系统视图。

---

SYSDBFIL 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSDBFIL 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSDBSpace 系统视图

SYSDBSpace 系统视图中的每一行均描述了一个 dbspace 文件。此视图的基础系统表为 ISYSDBSpace。

---

**注释** 此视图将取代不建议使用的 SYSDBSpace 系统视图。

---

SYSDBSpace 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSDBSpace 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSDATABASEPERM 系统视图

SYSDATABASEPERM 系统视图中的每一行均描述了一种对 dbspace 文件的权限。此视图的基础系统表为 ISYSDATABASEPERM。

SYSDATABASEPERM 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSDATABASEFILE 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSDEPENDENCY 系统视图

SYSDEPENDENCY 系统视图中的每一行均描述了两个数据库对象之间的依赖性。此视图的基础系统表为 ISYSDEPENDENCY。

如果一个数据库对象在其定义中引用另一个数据库对象，则这两个对象之间存在依赖性。例如，如果某视图的查询规范引用某个表，则认为该视图依赖该表。数据库服务器跟踪视图对表、视图、物化视图和列的依赖性。

SYSDEPENDENCY 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSDEPENDENCY 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## 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。此视图的内容在正常操作期间不会发生更改。此视图的基础系统表为 ISYSDOMAIN。

SYSDOMAIN 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSDOMAIN 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSEVENT 系统视图

SYSEVENT 系统视图中的每一行均描述了一个用 CREATE EVENT 创建的事件。此视图的基础系统表为 ISYSEVENT。

SYSEVENT 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSEVENT 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSEVENTTYPE 系统视图

SYSEVENTTYPE 系统视图定义可以由 CREATE EVENT 引用的系统事件类型。

SYSEVENTTYPE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSEVENT 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSEXTERNENV 系统视图

Sybase IQ 包括对六个外部运行时环境的支持。这些运行时环境包括以 C/C++ 编写的嵌入式 SQL 和 ODBC 应用程序，以及以 Java、Perl、PHP 或诸如 C# 和 Visual Basic 之类的基于 Microsoft .NET Framework 公共语言运行库 (CLR) 的语言编写的应用程序。

SYSEXTERNENV 系统视图中的每一行均描述了标识和启动每个外部环境所需的信息。此视图的基础系统表为 ISYSEXTERNENV。

SYSEXTERNENV 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSEXTERNENV 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSEXTERNENVOBJECT 系统视图

Sybase IQ 包括对六个外部运行时环境的支持。这些运行时环境包括以 C/C++ 编写的嵌入式 SQL 和 ODBC 应用程序，以及以 Java、Perl、PHP 或诸如 C# 和 Visual Basic 之类的基于 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均描述了一个已安装的外部对象。此视图的基础系统表为 ISYSEXTERNENVOBJECT。

SYSEXTERNENVOBJECT 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSEXTERNENVOBJECT 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSEXTERNLOGIN 系统视图

SYSEXTERNLOGIN 系统视图中的每一行均描述了一个用于远程数据访问的外部登录名。此视图的基础系统表为 ISYSEXTERNLOGIN。

SYSEXTERNLOGIN 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSEXTERNLOGIN 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSFILE 兼容性视图（不建议使用）

SYSFILE 系统视图中的每一行均描述了数据库的一个 dbspace。每个数据库都包含一个或多个 dbspace；每个 dbspace 都对应于一个操作系统文件。

SYSFILE 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSFILE 兼容性视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSFKCOL 兼容性视图（不建议使用）

SYSFKCOL 中的每一行均描述了该关系的外表中的外部列与主表中的主列之间的关联。不建议使用此视图；请改用 SYSIDX 和 SYSIDXCOL 系统视图。

SYSFKCOL 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSFKCOL 兼容性视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSFKEY 系统视图

SYSFKEY 系统视图中的每一行均描述了系统中的一个外键约束。此视图的基础系统表为 ISYSFKEY。

SYSFKEY 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSFKEY 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## 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 系统表的较旧版本的 Sybase IQ 兼容。但是，以前的 SYSFOREIGNKEY 系统表已由 ISYSFKEY 系统表及其相应的 SYSFKEY 系统视图取代，您应改用它们。

SYSFOREIGNKEY 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSFOREIGNKEY 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSFOREIGNKEYS 统一视图

SYSFOREIGNKEYS 视图中的每一行均描述了目录中每个表的一个外键。

SYSFOREIGNKEYS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSFOREIGNKEYS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSGROUP 系统视图

每个组的每个成员在 SYSGROUP 系统视图中都有对应的一行。此视图描述了组与成员之间的多对多关系。一个组可能有很多成员，一个用户可能是很多组的成员。此视图的基础系统表为 ISYSGROUP。

SYSGROUP 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSGROUP 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSGROUPS 统一视图

每个组的每个成员在 SYSGROUPS 视图中都有对应的一行。此视图描述了组与成员之间的多对多关系。一个组可能有很多成员，一个用户可能是很多组的成员。

SYSGROUPS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSGROUPS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSHISTORY 系统视图

SYSHISTORY 系统视图中的每一行均记录了对数据库的一项系统操作，例如，数据库启动、数据库校准等。此视图的基础系统表为 ISYSHISTORY。

SYSHISTORY 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSHISTORY 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSIDX 系统视图

SYSIDX 系统视图中的每一行均定义了数据库中的一个逻辑索引。此视图的基础系统表为 ISYSIDX。

---

**注释** 此视图将取代不建议使用的 SYSINSDX 系统视图。

---

SYSIDX 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSIDX 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSIDXCOL 系统视图

SYSIDXCOL 系统视图中的每一行均描述了 SYSIDX 系统视图中描述的索引的一个列。此视图的基础系统表为 ISYSIDXCOL。

SYSIDXCOL 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSIDXCOL 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## 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 系统表的较旧版本的 Sybase IQ 兼容。但是，SYSINDEX 系统表已由 ISYSIDX 系统表及其相应的 SYSIDX 系统视图取代，您应改用它们。

SYSINDEX 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSINDEX 兼容性视图（不建议使用）”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “兼容性视图”。

## SYSINDEXES 统一视图

SYSINDEXES 视图中的每一行均描述了数据库中的一个索引。除了此视图，您还可以使用 SYSIDX 和 SYSIDXCOL 系统视图。

SYSINDEXES 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSINDEXES 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSINDEXES ASE 兼容性视图

此视图由用户 DBO 所有。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。请参见第 478 页的表 8-2。

## SYSINFO 兼容性视图（不建议使用）

SYSINFO 视图表示在创建数据库时定义的数据库特征。它始终只包含一行。此视图可以通过内置函数获得，且不会保留在目录中。

SYSINFO 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSINFO 兼容性视图（不建议使用）”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “兼容性视图”。

## 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      | 说明                                                                 |
|---------|-----------------|----------|--------------------------------------------------------------------|
| 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：<br>0 = FULL<br>1 = INCREMENTAL<br>2 = INCREMENTAL SINCE FULL |

| 列名             | 列类型             | 列约束      | 说明                                                                                            |
|----------------|-----------------|----------|-----------------------------------------------------------------------------------------------|
| selective_type | tinyint         | NOT NULL | 备份子类型：<br>0 = ALL （备份所有 dbfile）<br>1 = READ/WRITE ONLY （备份所有读写文件）<br>2 = READ ONLY （备份特定只读文件） |
| virtual_type   | tinyint         | NOT NULL | 备份虚拟类型：<br>0 = NONE<br>1 = DECOUPLED<br>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)

**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             | 说明                                    |
|------------------|-----------------|---------------------------------------|
| 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                   |

| 列名                   | 列类型             | 说明            |
|----------------------|-----------------|---------------|
| 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。请参见第 455 页的“SYSIQTABCOL 系统视图”。

## SYSIQDBFILE 系统视图

以可读格式显示 ISYSIQDBFILE 中的组信息。

**注释** 此视图将取代不建议使用的 SYSIQFILE 系统视图。

| 列名            | 列类型          | 说明                                   |
|---------------|--------------|--------------------------------------|
| 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                               |

| 列名            | 列类型          | 说明                                           |
|---------------|--------------|----------------------------------------------|
| create_txn_id | xact_id      | 创建 dbfile 的事务 ID                             |
| alter_txn_id  | xact_id      | 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)

**SYSIQDBSPACE 系统视图**

以可读格式显示 ISYSIQDBSPACE 中的组信息。

| 列名             | 列类型          | 说明                                              |
|----------------|--------------|-------------------------------------------------|
| 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                              |
| create_txn_id  | xact_id      | 创建 dbspace 的事务 ID                               |
| alter_txn_id   | xact_id      | 上次修改读写状态的事务 ID                                  |
| striping_on    | char(1)      | “T” — 打开磁盘条带化;<br>“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。请参见第 447 页的“SYSIQDBFILE 系统视图”。

## 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。

**注释** 此视图将取代不建议使用的 SYSIQINDEX 系统视图。

| 列名                     | 列类型           | 说明                                           |
|------------------------|---------------|----------------------------------------------|
| 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                                        |
| 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                   | 说明                                                                                                                                                                               |
|---------------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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       | xact_id               | 供内部使用                                                                                                                                                                            |
| rd_commit_txn_id    | xact_id               | 供内部使用                                                                                                                                                                            |
| multiplex name      | CHAR(128) NULL        | 此数据库是其成员的 Multiplex 的名称                                                                                                                                                          |
| last_multiplex_mode | TINYINT NULL          | <p>(Sybase IQ 15.2 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。</p> <ul style="list-style-type: none"> <li>• 0 — 单节点。</li> <li>• 1 — 读取器。</li> <li>• 2 — 事务协调器。</li> <li>• 3 — 写入器。</li> </ul> |

## 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。

**注释** 此视图将取代不建议使用的 SYSIQJOININDEX 系统视图。

| 列名                 | 列类型                 | 说明                                  |
|--------------------|---------------------|-------------------------------------|
| 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      | xact_id             | 供内部使用。                              |
| txn_id             | xact_id             | 供内部使用。                              |
| valid              | 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)

## SYSIQJOININDEX 系统视图（不建议使用）

SYSIQJOININDEX 已由 SYSIQJOINIDX 系统视图取代。请参见第 451 页的“SYSIQJOINIDX 系统视图”。

## SYSIQJOINIXCOLUMN 系统视图

```
ALTER VIEW "SYS"."SYSIQJOINIXCOLUMN"
as select * from SYS.ISYSIQJOINIXCOLUMN
```

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。

| 列名              | 列类型             | 说明                                                     |
|-----------------|-----------------|--------------------------------------------------------|
| 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。 |
| coalesce        | char(1)         | 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

```
ALTER VIEW "SYS"."SYSIQJOINIXTABLE"
as select * from SYS.ISYSIQJOINIXTABLE
```

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。

| 列名           | 列类型             | 说明                        |
|--------------|-----------------|---------------------------|
| 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

## SYSIQMPXLOGINPOLICYOPTION 系统视图

显示便于阅读版的 ISYSIQMPXLOGINPOLICYOPTION 表中的组信息，该表存储的信息允许用户针对每台 Multiplex 服务器覆盖某个特定登录策略的两个登录策略选项 (lock 和 max\_connections)。

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》的附录 A “Multiplex 引用” 中的“SYSIQMPXLOGINPOLICYOPTION 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》的附录 A “Multiplex 引用” 中的“SYSIQMPXSERVER 系统视图”。

## 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。请参见第 478 页的表 8-2。

## SYSIQPARTITIONCOLUMN 系统视图

```
ALTER VIEW "SYS"."SYSIQPARTITIONCOLUMN"
as select * from SYS.ISYSIQPARTITIONCOLUMN
```

以可读格式显示 ISYSIQPARTITIONCOLUMN 中的组信息。SYSIQPARTITIONCOLUMN 视图中的每一行均描述了 SYSPARTITIONSCHEME 视图中描述的分区的 SYSIQPARTITION 视图中描述的分区的列。SYSIQPARTITIONCOLUMN 仅描述未存储在分区的 dbspace 上的列的分区的。

| 列名                    | 列类型             | 说明                             |
|-----------------------|-----------------|--------------------------------|
| 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

```
ALTER VIEW "SYS"."SYSIQTAB"
as select * from SYS.ISYSIQTAB
```

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。

**注释** 此视图将取代不建议使用的 SYSIQTAB 系统视图。

| 列名                 | 列类型                 | 说明                       |
|--------------------|---------------------|--------------------------|
| 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。                   |
| vdo_id_size        | unsigned int        | 供内部使用。                   |
| info_location      | hs_vdorecid         | 未使用。始终为 0。               |
| info_recid_size    | unsigned int        | 未使用。始终为 0。               |
| info_location_size | unsigned int        | 未使用。始终为 0。               |
| commit_txn_id      | xact_id             | 供内部使用。                   |
| txn_id             | xact_id             | 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)

## SYSIQTABCOL 系统视图

```
ALTER VIEW "SYS"."SYSIQTABCOL"
as select * from SYS.ISYSIQTABCOL
```

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。

**注释** 此视图将取代不建议使用的 SYSIQCOLUMN 系统视图。

| 列名             | 列类型          | 说明                                                                  |
|----------------|--------------|---------------------------------------------------------------------|
| 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 <code>select * from table</code> 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。                                                              |

| 列名                  | 列类型          | 说明                              |
|---------------------|--------------|---------------------------------|
| 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)

## SYSIQTABLE 系统视图（不建议使用）

SYSIQTABLE 已由 SYSIQTAB 系统视图取代。请参见第 454 页的“SYSIQTAB 系统视图”。

## 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。请参见第 478 页的表 8-2。

## 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 系统表的较旧版本的 Sybase IQ 兼容。但是，SYSIXCOL 系统表已由 ISYSIDXCOLD 系统表及其相应的 SYSIDXCOLD 系统视图取代。

SYSIXCOL 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSIXCOL 兼容性视图（不建议使用）”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “兼容性视图”。

## SYSJAR 系统视图

SYSJAR 系统视图中的每一行均定义了一个存储在数据库中的 JAR 文件。此视图的基础系统表为 ISYSJAR。

SYSJAR 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSJAR 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSJARCOMPONENT 系统视图

SYSJARCOMPONENT 系统视图中的每一行均定义了一个 JAR 文件组件。此视图的基础系统表为 ISYSJARCOMPONENT。

SYSJARCOMPONENT 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“ISYSJARCOMPONENT 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSJAVACLASS 系统视图

SYSJAVACLASS 系统视图中的每一行均描述了一个存储在数据库中的 Java 类。此视图的基础系统表为 ISYSJAVACLASS。

SYSJAVACLASS 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSJAVACLASS 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSLOGINMAP 系统视图

可以使用集成登录或 Kerberos 登录连接到数据库的每个用户在 SYSLOGINMAP 系统视图中都有对应的一行。作为安全措施，只有具有 DBA 授权的用户才能查看该视图的内容。此视图的基础系统表为 ISYSLOGINMAP。

SYSLOGINMAP 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSLOGINMAP 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSLOGINPOLICY 系统视图

此视图的基础系统表为 ISYSLOGINPOLICY。

SYSLOGINPOLICY 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSLOGINPOLICY 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSLOGINPOLICYOPTION 系统视图

此视图的基础系统表为 ISYSLOGINPOLICYOPTION。

SYSLOGINPOLICYOPTION 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSLOGINPOLICYOPTION 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSLOGINS ASE 兼容性视图

此视图由用户 DBO 所有。每个有效的 Adaptive Server 用户帐户在 syslogins 中都有对应的一行。请参见第 479 页的表 8-3。

## SYSMVOPTION 系统视图

SYSMVOPTION 系统视图中的每一行均描述了物化视图的一个选项值在创建时的设置。但是，该描述没有包含选项名称。此视图的基础系统表为 ISYSMVOPTION。

---

**注释** 仅 IQ 目录存储中的 SQL Anywhere 表支持物化视图。

---

SYSMVOPTION 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSMVOPTION 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSMVOPTIONNAME 系统视图

SYSMVOPTIONNAME 系统视图的每一行均包含 SYSMVOPTION 系统视图中定义的一个选项的名称。此视图的基础系统表为 ISYSMVOPTIONNAME。

SYSMVOPTIONNAME 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSMVOPTIONNAME 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSOBJECT 系统视图

SYSOBJECT 系统视图中的每一行均描述了一个数据库对象。此视图的基础系统表为 ISYSOBJECT。

SYSOBJECT 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSOBJECT 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSOBJECTS ASE 兼容性视图

此视图由用户 DBO 所有。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已编译对象在 sysobjects 中都有对应的一行。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。请参见第 478 页的表 8-2。

## SYSOPTION 系统视图

数据库中存储的每个选项设置在 SYSOPTION 系统视图中都有对应的一行。对于给定选项，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，以供那些没有自己的设置的用户使用。此视图的基础系统表为 ISYSOPTION。

SYSOPTION 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSOPTION 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSOPTIONS 统一视图

SYSOPTIONS 视图中的每一行均描述了一个使用 SET 创建的选项。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，以供那些没有自己的设置的用户使用。

SYSOPTIONS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSOPTIONS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSOPTSTAT 系统视图

SYSOPTSTAT 系统视图存储 ALTER DATABASE CALIBRATE 语句计算的成本模型校准信息。此视图的内容仅供内部使用，可以通过 sa\_get\_dtt 系统过程实现最佳访问。此视图的基础系统表为 ISYSOPTSTAT。

SYSOPTSTAT 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSOPTSTAT 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSPARTITION 系统视图

```
ALTER VIEW "SYS"."SYSPARTITION"
as select * from SYS.ISYSPARTITION
```

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）。

| 列名                    | 列类型             | 说明                                 |
|-----------------------|-----------------|------------------------------------|
| 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。                        |
| position              | 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**

```
ALTER VIEW "SYS"."SYSPARTITIONKEY"
as select * from SYS.ISYSPARTITIONKEY
```

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）。

| 列名                    | 列类型             | 说明                                     |
|-----------------------|-----------------|----------------------------------------|
| 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。                    |
| position              | 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**

```
ALTER VIEW "SYS"."SYSPARTITIONSCHEME"
as select * from SYS.ISYSPARTITIONSCHEME
```

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）。

| 列名                    | 列类型             | 说明                    |
|-----------------------|-----------------|-----------------------|
| partitioned_object_id | unsigned bigint | 每个分区对象（表）都被指派了一个唯一编号。 |

| 列名                  | 列类型     | 说明                                           |
|---------------------|---------|----------------------------------------------|
| partition_method    | tinyint | 此表的分区方法。有效值：<br>1—表示范围<br>Sybase IQ 仅支持范围分区。 |
| subpartition_method | tinyint | 留待将来使用。                                      |

#### 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 系统视图中的每一行均定义了数据库中的一个物理索引。此视图的基础系统表为 ISYSPHYSIDX。

SYSPHYSIDX 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSPHYSIDX 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSPROCAUTH 统一视图

SYSPROCAUTH 视图中的每一行均描述了被授予的对过程的一组权限。此外，您还可以使用 SYSPROCPERM 系统视图。

SYSPROCAUTH 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSPROCAUTH 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSPROCEDURE 系统视图

SYSPROCEDURE 系统视图中的每一行均描述了数据库中的一个过程。此视图的基础系统表为 ISYSPROCEDURE。

SYSPROCEDURE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSPROCEDURE 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSROCPARM 系统视图

SYSROCPARM 系统视图中的每一行均描述了数据库中过程的一个参数。此视图的基础系统表为 ISYSROCPARM。

SYSROCPARM 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSROCPARM 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSROCPARMS 统一视图

SYSROCPARMS 视图中的每一行均描述了数据库中过程的一个参数。

SYSROCPARMS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSROCPARMS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSROCPERM 系统视图

SYSROCPERM 系统视图的每一行均描述了被授予执行过程的权限的一个用户。只有得到授权的用户才能执行过程。此视图的基础系统表为 ISYSROCPERM。

SYSROCPERM 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSROCPERM 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSROCS 统一视图

SYSROCS 视图显示过程或函数名称、过程或函数创建者的名称以及记录的任何过程或函数注释。

SYSROCS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSROCS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## **SYSPROXYTAB 系统视图**

SYSPROXYTAB 系统视图的每一行均描述了一个代理表的远程参数。此视图的基础系统表为 ISYSPROXYTAB。

SYSPROXYTAB 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSPROXYTAB 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## **SYSPUBLICATION 系统视图**

SYSPUBLICATION 系统视图中的每一行均描述了一个 SQL Remote 或 MobiLink 发布。此视图的基础系统表为 ISYSPUBLICATION。

SYSPUBLICATION 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSPUBLICATION 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## **SYSPUBLICATIONS 统一视图**

SYSPUBLICATIONS 视图中的每一行均描述了一个 SQL Remote 或 MobiLink 发布。

SYSPUBLICATIONS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSPUBLICATIONS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## **SYSREMARK 系统视图**

SYSREMARK 系统视图中的每一行均描述了对对象的一个注释。此视图的基础系统表为 ISYSREMARK。

SYSREMARK 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSREMARK 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSREMOTEOPTION 系统视图

SYSREMOTEOPTION 系统视图中的每一行均描述了一个 SQL Remote 消息链接参数的值。此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 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSREMOTEOPTION 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSREMOTEOPTION2 统一视图

以可读格式显示 SYSREMOTEOPTION 和 SYSREMOTEOPTIONTYPE 中不包含敏感数据的列。

SYSREMOTEOPTION2 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSREMOTEOPTION2 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSREMOTEOPTIONS 统一视图

SYSREMOTEOPTIONS 视图中的每一行均描述了一个 SQL Remote 消息链接参数的值。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 视图提供了对不敏感数据的公共访问权限。

SYSREMOTEOPTIONS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSREMOTEOPTIONS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## **SYSREMOTEOPTIONTYPE 系统视图**

SYSREMOTEOPTIONTYPE 系统视图中的每一行均描述了一个 SQL Remote 消息链接参数。此视图的基础系统表为 ISYSREMOTEOPTIONTYPE。

SYSREMOTEOPTIONTYPE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSREMOTEOPTIONTYPE 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## **SYSREMOTETYPE 系统视图**

SYSREMOTETYPE 系统视图包含有关 SQL Remote 的信息。此视图的基础系统表为 ISYSREMOTETYPE。

SYSREMOTETYPE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSREMOTETYPE 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## **SYSREMOTETYPES 统一视图**

SYSREMOTETYPES 视图的每一行均描述了一个 SQL Remote 消息类型，包括发布者地址。

SYSREMOTETYPES 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSREMOTETYPES 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## **SYSREMOTEUSER 系统视图**

SYSREMOTEUSER 系统视图中的每一行均描述了一个具有 REMOTE 权限的用户 ID（预订者），以及发送给该用户和来自该用户的 SQL Remote 消息的状态。此视图的基础系统表为 ISYSREMOTEUSER。

SYSREMOTEUSER 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSREMOTEUSER 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSREMOTEUSERS 统一视图

SYSREMOTEUSERS 视图的每一行均描述了一个具有 REMOTE 权限的用户 ID（预订者），以及发送给该用户和来自该用户的 SQL Remote 消息的状态。

SYSREMOTEUSERS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSREMOTEUSERS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSSCHEDULE 系统视图

SYSSCHEDULE 系统视图中的每一行均描述了一个触发事件的时间，该时间由 CREATE EVENT 的 SCHEDULE 子句指定。此视图的基础系统表为 ISYSSCHEDULE。

SYSSCHEDULE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSREMOTEUSER 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSSERVER 系统视图

SYSSERVER 系统视图中的每一行均描述了一个远程服务器。此视图的基础系统表为 ISYSSERVER。

SYSSERVER 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSREMOTEUSER 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSSOURCE 系统视图

SYSSOURCE 系统视图中的每一行均包含 SYSOBJECT 系统视图中所列对象的相应源代码（如果有）。

SYSSOURCE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSSOURCE 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSSQLSERVERTYPE 系统视图

SYSSQLSERVERTYPE 系统视图包含与 Adaptive Server Enterprise 兼容性相关的信息。此视图的基础系统表为 ISYSSQLSERVERTYPE。

SYSSQLSERVERTYPE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSSQLSERVERTYPE 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSSUBPARTITIONKEY 系统视图

此视图留待将来使用。Sybase IQ 15.2 不支持子分区。

## SYSSUBSCRIPTION 系统视图

SYSSUBSCRIPTION 系统视图中的每一行均描述了一个用户 ID（必须有 REMOTE 权限）对一个发布的预定。此视图的基础系统表为 ISYSSUBSCRIPTION。

SYSSUBSCRIPTION 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSSUBSCRIPTION 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSSUBSCRIPTIONS 统一视图

每一行描述一个用户 ID（必须有 REMOTE 权限）对一个发布的预订。

SYSSUBSCRIPTIONS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSSUBSCRIPTIONS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSSYNC 系统视图

SYSSYNC 系统视图包含与 MobiLink 同步相关的信息。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。SYSSYNC2 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。此视图的基础系统表为 ISYSSYNC。

SYSSYNC 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSSYNC 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSSYNC2 统一视图

SYSSYNC2 视图提供了对在 SYSSYNC 系统视图中找到的数据（与 MobiLink 同步相关的信息）的公共访问权限，但不公开潜在敏感数据。

SYSSYNC2 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSSYNC2 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSSYNCPUBLICATIONDEFAULTS 统一视图

SYSSYNCPUBLICATIONDEFAULTS 视图提供了与 MobiLink 同步中涉及的发布相关的缺省同步设置。

SYSSYNCPUBLICATIONDEFAULTS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSSYNCPUBLICATIONDEFAULTS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSSYNCS 统一视图

SYSSYNCS 视图包含与 MobiLink 同步相关的信息。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。

SYSSYNCS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSSYNCS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSSYNCSCRIPT 系统视图

SYSSYNCSCRIPT 系统视图中的每一行均标识了 MobiLink 脚本式上载的一个存储过程。此视图几乎与 SYSSYNCSCRIPTS 视图完全相同，只不过此视图中的值采用其原始格式。若要以可读格式查看这些数据，请参见 [“SYSSYNCSCRIPTS 统一视图”](#)。

SYSSYNCSCRIPT 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的 [“SYSSYNCSCRIPT 系统视图”](#)，具体位置为：[“SQL Anywhere 11.0.1”](#) > [“SQL Anywhere Server — SQL 参考”](#) > [“系统对象”](#) > [“视图”](#) > [“系统视图”](#)。

## SYSSYNCSCRIPTS 统一视图

SYSSYNCSCRIPTS 视图中的每一行均标识了 MobiLink 脚本式上载的一个存储过程。此视图几乎与 SYSSYNCSCRIPT 系统视图完全相同，只不过此视图中的值采用可读格式（与原始数据相对）。

SYSSYNCSCRIPTS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的 [“SYSSYNCSCRIPTS 统一视图”](#)，具体位置为：[“SQL Anywhere 11.0.1”](#) > [“SQL Anywhere Server — SQL 参考”](#) > [“系统对象”](#) > [“视图”](#) > [“统一视图”](#)。

## SYSSYNCSUBSCRIPTIONS 统一视图

SYSSYNCSUBSCRIPTIONS 视图包含与 MobiLink 同步预订关联的同步设置。

SYSSYNCSUBSCRIPTIONS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的 [“SYSSYNCSUBSCRIPTIONS 统一视图”](#)，具体位置为：[“SQL Anywhere 11.0.1”](#) > [“SQL Anywhere Server — SQL 参考”](#) > [“系统对象”](#) > [“视图”](#) > [“统一视图”](#)。

## SYSSYNCUSERS 统一视图

与 MobiLink 同步用户关联的同步设置的视图。

SYSSYNCUSERS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的 [“SYSSYNCUSERS 统一视图”](#)，具体位置为：[“SQL Anywhere 11.0.1”](#) > [“SQL Anywhere Server — SQL 参考”](#) > [“系统对象”](#) > [“视图”](#) > [“系统视图”](#)。

## SYSTAB 系统视图

SYSTAB 系统视图中的每一行均描述了数据库中的一个表或视图。可以在 SYSVIEW 系统视图中找到视图的其它信息。此视图的基础系统表为 ISYSTAB。

SYSTAB 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSTAB 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## 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 系统表的较旧版本的 Sybase IQ 兼容。但是，SYSTABLE 系统表已由 ISYSTAB 系统表及其相应的 SYSTAB 系统视图取代。

SYSTABLE 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSTABLE 兼容性视图（不建议使用）”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “兼容性视图”。

## SYSTABAUTH 统一视图

SYSTABAUTH 视图包含 SYSTABLEPERM 系统视图中的信息（但仅限可读格式的信息）。

SYSTABAUTH 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSTABAUTH 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSTABCOL 系统视图

数据库中的每个表和视图的每一列在 SYSTABCOL 系统视图中都有对应的一行。此视图的基础系统表为 ISYSTABCOL。

SYSTABCOL 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSTABCOL 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSTABLEPERM 系统视图

通过 GRANT 语句授予的权限存储在 SYSTABLEPERM 系统视图中。此视图中的每一行对应于一个表、一个授予权限的用户 ID（授予者）和一个被授予权限的用户 ID（被授予者）。此视图的基础系统表为 ISYSTABLEPERM。

SYSTABLEPERM 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSTABLEPERM 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSTEXTCONFIG 系统视图

SYSTEXTCONFIG 系统视图中的每一行均描述了一个文本配置对象，以便与全文本搜索功能结合使用。此视图的基础系统表为 ISYSTEXTCONFIG。

ISYSTEXTCONFIG 保存有关入口点以及用于令牌和预过滤的外部库的信息。ISYSTEXTCONFIG 中的预过滤列的数据类型是 long varchar，external\_term\_breaker 列的数据类型也是 long varchar，用来保存外部函数名和外部术语中断程序库名称。

SYSTEXTCONFIG 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSTEXTCONFIG 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSTEXTIDX 系统视图

SYSTEXTIDX 系统视图中的每一行均描述了一个 TEXT 索引。此视图的基础系统表为 ISYSTEXTIDX。

SYSTEXTIDX 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSTEXTIDX 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSTEXTIDXTAB 系统视图

SYSTEXTIDXTAB 系统视图中的每一行均描述了一个作为 TEXT 索引的一部分的生成表。此视图的基础系统表为 ISYSTEXTIDXTAB。

SYSTEXTIDXTAB 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSTEXTIDXTAB 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSTRIGGER 系统视图

SYSTRIGGER 系统视图中的每一行均描述了数据库中的一个触发器。此视图还包含为具有参照触发操作（例如 ON DELETE CASCADE）的外键定义自动创建的触发器。此视图的基础系统表为 ISYSTRIGGER。

SYSTRIGGER 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSTRIGGER 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSTRIGGERS 统一视图

SYSTRIGGERS 视图中的每一行均描述了数据库中的一个触发器。此视图还包含为具有参照触发操作（例如 ON DELETE CASCADE）的外键定义自动创建的触发器。

SYSTRIGGERS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSTRIGGERS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSTYPEMAP 系统视图

SYSTYPEMAP 系统视图包含 SYSSQLSERVERTYPE 系统视图中的条目的兼容性映射值。此视图的基础系统表为 ISYSTYPEMAP。

SYSTYPEMAP 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSTYPEMAP 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSTYPES ASE 兼容性视图

此视图由用户 DBO 所有。每个系统提供的数据类型和用户定义的数据类型在 `systypes` 中都有对应的一行。指定了域（由规则定义）和缺省值（如果存在）。无法改变描述系统提供的数据类型的行。请参见第 478 页的表 8-2。

## SYSUSER 系统视图

SYSUSER 系统视图中的每一行均描述了系统中的一个用户。此视图的基础系统表为 `ISYSUSER`。

SYSUSER 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSUSER 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSUSERAUTH 兼容性视图（不建议使用）

提供 SYSUSERAUTH 视图的目的是为了与较旧版本的 Sybase IQ 兼容。请改用 SYSUSERAUTHORITY 系统视图。

SYSUSERAUTH 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSUSERAUTH 兼容性视图（不建议使用）”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “兼容性视图”。

## SYSUSERAUTHORITY 系统视图

SYSUSERAUTHORITY 系统视图的每一行均描述了为一个用户 ID 授予的权限。此视图的基础系统表为 `ISYSUSERAUTHORITY`。

SYSUSERAUTHORITY 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSUSERAUTHORITY 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSUSERLIST 兼容性视图（不建议使用）

提供 SYSUSERLIST 视图的目的是为了与较旧版本的 Sybase IQ 兼容。SYSUSERLIST 视图的每一行均描述了一个用户，而不公开其用户 ID 和口令。每个用户由其用户名标识。

SYSUSERLIST 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSUSERLIST 兼容性视图（不建议使用）”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “兼容性视图”。

## SYSUSERMESSAGE 系统视图

SYSUSERMESSAGE 系统视图中的每一行均保存一个有关错误条件的用户定义消息。此视图的基础系统表为 ISYSUSERMESSAGE。

SYSUSERMESSAGE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSUSERMESSAGE 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSUSEROPTIONS 统一视图

SYSUSEROPTIONS 视图包含每个用户的有效选项设置。如果用户没有某选项的设置，此视图将显示该选项的公共设置。

SYSUSEROPTIONS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSUSEROPTIONS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSUSERPERM 兼容性视图（不建议使用）

不建议使用此视图，因为它只显示以前版本中可用的权限。将应用程序更改为使用 SYSUSERAUTHORITY 系统视图。SYSUSERPERM 视图的每一行均描述了一个用户 ID。

SYSUSERPERM 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSUSERPERM 兼容性视图（不建议使用）”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “兼容性视图”。

## SYSUSERPERMS 兼容性视图（不建议使用）

不建议使用此视图，因为它只显示以前版本中可用的权限。将应用程序更改为使用 SYSUSERAUTHORITY 系统视图。与 SYSUSERPERM 视图类似，SYSUSERPERMS 视图中的每一行均描述了一个用户 ID。但是，未包括口令信息。允许所有用户从此视图中读取内容。

SYSUSERPERMS 视图是 SQL Anywhere 兼容性视图。请参见 SQL Anywhere 文档中的“SYSUSERPERMS 兼容性视图（不建议使用）”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “兼容性视图”。

## SYSUSERTYPE 系统视图

SYSUSERTYPE 系统视图中的每一行均描述了一种用户定义的数据类型。此视图的基础系统表为 ISYSUSERTYPE。

SYSUSERTYPE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSUSERTYPE 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSUSERS ASE 兼容性视图

此视图由用户 DBO 所有。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。请参见第 478 页的表 8-2。

## SYSVIEW 系统视图

SYSVIEW 系统视图中的每一行均描述了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。

---

**注释** 仅 IQ 目录存储中的 SQL Anywhere 表支持物化视图。

---

SYSVIEW 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSVIEW 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## SYSVIEWS 统一视图

SYSVIEWS 视图的每一行均描述了一个视图，包括其视图定义。

SYSVIEWS 视图是 SQL Anywhere 统一视图。请参见 SQL Anywhere 文档中的“SYSVIEWS 统一视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “统一视图”。

## SYSWEBSERVICE 系统视图

SYSWEBSERVICE 系统视图中的每一行均描述了一种 Web 服务。

SYSWEBSERVICE 视图是 SQL Anywhere 系统视图。请参见 SQL Anywhere 文档中的“SYSWEBSERVICE 系统视图”，具体位置为：“SQL Anywhere 11.0.1” > “SQL Anywhere Server — SQL 参考” > “系统对象” > “视图” > “系统视图”。

## 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的系统信息，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 与 Sybase IQ 系统目录不同。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。

表 8-2、表 8-3 和表 8-4 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。

每个 Adaptive Server Enterprise 数据库中的表

**表 8-2: 每个 ASE 数据库中的表**

| 表名             | 说明                                                                                               | 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占一行，过程中的每个参数占一行。<br>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          |
| 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。<br>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、外键或公用键占一行；由用户设置（Adaptive Server Enterprise 不维护）。                                            | 否       | 否          |
| syslogs        | 事务日志。                                                                                            | 否       | 否          |
| sysobjects     | 每个表、视图、过程、规则、缺省值、日志和临时对象（仅在 tempdb 中）占一行。                                                        | 仅包含兼容数据 | 是          |
| sysprocedures  | 每个视图、规则、缺省值和过程占一行，以给出内部定义。                                                                       | 否       | 否          |
| sysprotects    | 用户权限信息。                                                                                          | 否       | 否          |
| sysreferences  | 在表或列上声明的每个参照完整性约束占一行。                                                                            | 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    | 每一段（指定的磁盘区段集合）占一行。                                                                               | 否       | 否          |
| systhresholds  | 为数据库定义的每个阈值占一行。                                                                                  | 否       | 否          |

| 表名       | 说明                    | 是否包含数据? | 是否受 IQ 支持? |
|----------|-----------------------|---------|------------|
| systypes | 每个系统提供的和用户定义的数据类型占一行。 | 是       | 是          |
| sysusers | 数据库允许的每个用户占一行。        | 是       | 是          |

Adaptive Server  
Enterprise master  
数据库中的表

**表 8-3: ASE master 数据库表**

| 表名              | 说明                                            | 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          |
| syssservers     | 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

**表 8-4: ASE sybsecurity 数据库表**

| 表名              | 说明          | 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

## 关于本附录

本附录旨在简化从其它 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。本附录首先概述了 Transact-SQL，并就从迁移到 Sybase IQ 时需要注意的多个方面对这些数据库进行比较：

- 体系结构
- 数据类型
- 数据定义语言
- 数据操作语言
- 存储过程语言

Sybase IQ 的每个新版本中都涉及了兼容性功能。本附录将 Sybase IQ 15.2 与 Adaptive Server Enterprise 15.0.3（及早期版本）和 SQL Anywhere 11.0.1 进行了比较。

## 主题

| 主题                                                     | 页码  |
|--------------------------------------------------------|-----|
| <a href="#">Transact-SQL 支持概述</a>                      | 482 |
| <a href="#">Adaptive Server 体系结构</a>                   | 483 |
| <a href="#">数据类型</a>                                   | 486 |
| <a href="#">数据定义语言</a>                                 | 491 |
| <a href="#">数据操作语言</a>                                 | 500 |
| <a href="#">Transact-SQL 过程语言概述</a>                    | 510 |
| <a href="#">存储过程的自动转换</a>                              | 513 |
| <a href="#">从 Transact-SQL 过程中返回结果集</a>                | 513 |
| <a href="#">Transact-SQL 过程中的变量</a>                    | 514 |
| <a href="#">Transact-SQL 过程中的错误处理</a>                  | 515 |
| <a href="#">SQL Anywhere 和 Sybase IQ</a>               | 517 |
| <a href="#">Adaptive Server Enterprise 和 Sybase IQ</a> | 520 |

本书中其它位置的兼容性信息

您可在以下章节中找到其它兼容性信息：

- 《参考：语句和选项》中的第 2 章“数据库选项”。请参见 [Transact-SQL 兼容性选项](#)。
- 第 3 章“SQL 数据类型”中，请参见各种数据类型的兼容性信息；另请参见第 83 页的“数据类型转换”。
- 《参考：语句和选项》。参见各个命令的兼容性信息。

有关 SQL Anywhere 的注意事项

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。除了 Sybase IQ 文档中让您参考 SQL Anywhere 文档的主题外，还始终要参考紧跟在每本 Sybase IQ 书籍目录后面“关于本手册”中所列的“Sybase IQ 文档”。

## Transact-SQL 支持概述

像 SQL Anywhere 一样，Sybase IQ 支持 **Transact-SQL** 中相当大的的一部分，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：

- 在 Sybase IQ 和 Adaptive Server Enterprise 之间，大多数 SQL 语句都是兼容的。
- 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。

- 过程或批处理语句以 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，每种术语都具有不同的流控制语句。

#### 相似和差异之处

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 体系结构

Adaptive Server Enterprise、SQL Anywhere 和 Sybase IQ 是互为补充的产品，其体系结构设计用于满足它们不同的目标。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中，而每个服务器也都可以包含多个数据库。用户可以具有服务器的登录权限并且可以连接到服务器。然后，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。

有关如何使用 `isql` 连接到包含多个数据库的服务器上的特定数据库的详细信息，请参见《系统管理指南，卷 2》中的第 3 章“[Sybase IQ 作为数据服务器](#)”。

## 空间分配和设备管理

所有这三种产品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，目的是反应产品的不同用法。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` 或者使用阈值自动添加更多空间。
- 在 Sybase IQ 中，最初通过在 `CREATE DATABASE` 语句中列出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警告 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。通过 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。请参见《实用程序指南》的第 4 章“[数据库管理实用程序](#)”中的“[初始化实用程序 \(iqinit\)](#)”。

有关磁盘管理的信息，请参见《系统管理指南，卷 1》。

## 系统表、Catalog 存储和 IQ 存储

IQ 数据库是包含如下部分的连接数据存储区：

- 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 表的集。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。有关列表及各项的说明，请参见第 8 章“系统表和视图”。有关 SQL Anywhere 兼容性视图的完整列表，请参见《SQL Anywhere Server - SQL 参考》。
- 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。

---

**注释** 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。

在 Adaptive Server Enterprise 中，不应重复的角色包括：

- **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全部权限。

在 SQL Anywhere 和 Sybase IQ 中，以下全数据库范围的权限具有管理角色：

- **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

本节讨论了各种数据类型的兼容性信息。有关 Sybase IQ 数据类型的详细信息，请参见第 3 章“SQL 数据类型”。

---

**注释** 本节中未包括的数据类型当前受全部三种产品支持。

---

## Bit 数据类型

所有这三种产品都支持 BIT 数据类型，但具有以下区别：

- SQL Anywhere 只允许 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

所有这三种产品都允许使用 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（定长）和 VARCHAR（变长）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。请参见第 3 章“SQL 数据类型”中的字符数据类型。

向 CHAR 或 VARCHAR 中插入时：

- 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，逻辑页大小可以是 2K、4K、8K 和 16K。例如：
  - 2K 页大小允许列为单个行那么大，大约是 1962 字节。
  - 4K 页大小允许的列大小约为 4010 字节。
- SQL Anywhere 支持最多 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，不支持名称 LONG VARCHAR。

- Sybase IQ 支持最多 32K-1 字节的 CHAR 和 VARCHAR 数据。  
Sybase IQ 也支持最多 512TB（IQ 页大小为 128KB）和 2PB（IQ 页大小为 512KB）的 LONG VARCHAR 数据。有关 Sybase IQ 中的 LONG VARCHAR 数据类型的信息，请参见《Sybase IQ 中的大对象管理》。
- 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 数据类型类型的 Unicode，而不支持作为单独数据类型类型的 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。

## 二进制数据类型

下表总结了二进制数据类型支持。

**表 A-1：二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* | 不支持                        | 2GB - 1      | 512TB（IQ 页大小为 128KB）<br>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 中的大对象管理》。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 同时支持 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”。

## 日期、时间、日期时间和时间戳数据类型

虽然所有这三种产品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` 类型和作为以二进制形式 (8) 实现的用户定义的数据类型 (域) 的时间戳。
- 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。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变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 转换时间的方式都相同。

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。

当使用 `INSERT...LOCATION` 将从 Adaptive Server Enterprise 数据库检索的 `TIME` 和 `DATETIME` 值插入 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”。

## 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。

## Text 数据类型

对 TEXT 数据的支持当前按如下方式实现：

- Adaptive Server Enterprise 和 SQL Anywhere 支持最多 2GB 的 LONG VARBINARY 和 TEXT 数据。
- Sybase IQ 支持最多 32KB - 1 的 VARCHAR 数据。Sybase IQ 还支持最多 512TB（IQ 页大小为 128KB）和 2PB（IQ 页大小为 512KB）的 LONG VARCHAR 数据。有关 Sybase IQ 中的 LONG VARCHAR 数据类型的信息，请参见《Sybase IQ 中的大对象管理》。

## Image 数据类型

对 IMAGE 数据的支持当前按如下方式实现：

- Adaptive Server Enterprise 和 SQL Anywhere 支持最多 2GB 的 IMAGE 数据。
- Sybase IQ 支持最多 512TB（IQ 页大小为 128KB）和 2PB（IQ 页大小为 512KB）的 LONG BINARY 数据。有关 Sybase IQ 中 LONG BINARY 数据类型的信息，请参见《Sybase IQ 中的大对象管理》。

## 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。

## 数据定义语言

本节讨论了用于创建数据库对象的兼容性信息。有关详细信息，另请参见第 485 页的“系统表、Catalog 存储和 IQ 存储”和第 484 页的“空间分配和设备管理”。

## 创建与 Transact-SQL 兼容的数据库

本节介绍了您在创建或重建数据库时必须做出的选择。

下面是创建与 Transact-SQL 兼容的数据库时所需执行的基本步骤。本节的其余部分将介绍您需要设置的选项。

### ❖ 从 Sybase Central 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 将字符串比较的区分大小写设置为 ON。这将模拟 Adaptive Server Enterprise。
- 3 将密码的区分大小写设置为 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

- 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** 数据的区分大小写特性会反映在索引、查询结果等内容中。
- **标识符** 标识符包括表名、列名、用户 ID 等。
- **口令** 口令的区分大小写的处理方式与其它标识符有所不同。

### 数据的区分大小写

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，可更改单字节字符集的区分大小写特性。

### 标识符的区分大小写

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，用户定义的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

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：

- 如果数据库最初是在不区分大小写的数据库中输入的，则口令会保持不区分大小写。
- 如果口令最初是在区分大小写的数据库中输入的，则大写和小写混合的口令会保持区分大小写。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。

## 确保兼容的对象名称

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，表名在数据库中和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。

## CREATE TABLE 语句

创建表以实现兼容性时，请注意以下各项。

### 列中的 NULL

若要实现 NULL 的兼容处理：

- SQL Anywhere 和 Sybase IQ 假设列可以为 NULL，但如果列在定义中指定 NOT NULL 则除外。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，以及用户定义的数据类型执行检查约束。用户可以记录检查完整性约束违规，并指定可在 LOAD 语句回滚前发生的冲突数量。

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，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 支持用户定义的数据类型，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，卷 1》中的第 9 章“确保数据完整性”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：SET NULL、DEFAULT。

---

**注释** 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/DEFAULT AUTOINCREMENT 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 视图中添加或删除 AUTOINCREMENT 列。
- Adaptive Server Enterprise 支持每个表一个标识列。ASE 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。ASE 用户不能显式设置为标识列选择的下一个值。ASE 视图不能包含 IDENTITY/AUTOINCREMENT 列。在某些条件下使用 SELECT INTO 时，ASE 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** 语句中在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。有关 Sybase IQ 中的已声明临时表的信息，请参见《参考：语句和选项》中的 **DECLARE LOCAL TEMPORARY TABLE** 语句。

请参见《参考：语句和选项》中的 **CREATE TABLE** 语句。

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。

## CREATE DEFAULT、CREATE RULE 和 CREATE DOMAIN 语句

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。

## CREATE TRIGGER 语句

触发器支持具有如下区别：

- SQL Anywhere 支持行级和语句级触发器。
- Adaptive Server Enterprise 只支持语句级触发器。
- Sybase IQ 不支持触发器。

---

**注释** 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。

---

## CREATE INDEX 语句

在这三种产品中，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 允许但忽略关键字 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，但如果在 CREATE INDEX 语句中指定了索引类型则除外：

```
CREATE [UNIQUE] [type] INDEX name
ON table (column,...)
```

---

**注释** 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。

---

请参见《系统管理指南，卷 1》中的第 6 章“使用 Sybase IQ 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。

### 用户如何连接

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
```

该语句向用户 **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 还支持针对其它项执行 **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 授权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。

#### 全数据库范围的权限

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授权。
- Adaptive Server Enterprise 中的系统管理员对服务器上的所有数据库拥有此权限。但是，对 Sybase IQ 数据库的 DBA 授权不同于 Adaptive Server Enterprise 数据库所有者的权限，Adaptive Server Enterprise 数据库所有者必须使用 Adaptive Server Enterprise 的 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。

请参见《系统管理指南，卷 1》中的第 8 章“管理用户 ID 和权限”，以及《系统管理指南，卷 2》中的第 3 章“Sybase IQ 作为数据服务器”。

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。有关详细信息，请参见第 7 章“系统过程”，包括第 422 页的“Adaptive Server Enterprise 系统过程”。

## 装载格式

这三种产品中的装载格式支持如下：

- 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 通过 BCP 处理 ASCII 和 BINARY 装载格式。

---

**注释** 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。

---

## 为实现 Transact-SQL 兼容性设置选项

使用 SET OPTION 语句设置 Sybase IQ 数据库选项。有关兼容行为所需的选项设置的列表，请参见《参考：语句和选项》中“[数据库选项](#)”的“[Transact-SQL 兼容性选项](#)”。

## 数据操作语言

本节提供了编写可移植查询的一些一般原则，然后讨论了特定的查询要求。

### 编写可移植 SQL 的一般原则

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。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：

- 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。

## 编写兼容的查询

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"
```

有关 **SELECT** 语句的完整说明，请参见《参考：语句和选项》中的 [SELECT 语句](#)。

下面各节详细说明了编写兼容查询时需要注意的一些项。

## 子查询

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 则不支持。

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

GROUP BY ALL 支持如下所示:

- 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 支持在 GROUP BY 子句中使用 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。

有关将 GROUP BY 与 OLAP 函数配合使用的信息，请参见《系统管理指南，卷 2》中的第 2 章“使用 OLAP”。

## COMPUTE 子句

COMPUTE 子句支持如下所示：

- 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

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()。

## 连接

本节讨论了对 Transact-SQL 外连接和 ANSI 连接的支持。

### Transact-SQL 外连接

外连接的支持语法可概括如下：

- 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。

---

**注释** SQL Anywhere 和 Sybase IQ 不支持 Transact-SQL 外连接语法。

---

## ANSI 连接

ANSI 连接的支持语法可概括如下：

- 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。

FULL OUTER JOIN 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。

## 有关外连接的详细信息

有关 Transact-SQL 外连接的详细信息，包括 ANSI 语法替代项，请参见白皮书“Transact-SQL 外连接的语义和兼容性” (Semantics and Compatibility of Transact-SQL Outer Joins)，网址为 <http://www.sybase.com/support/> 中的 Sybase。。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。

## Null 比较

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。

---

**注释** 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

按如下方式处理零长度字符串：

- 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

对此语法的支持如下所示：

- 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

Sybase IQ 支持 SQL Anywhere 和 Adaptive Server Enterprise 所支持的大多数函数，但具有以下区别：

- Adaptive Server Enterprise 支持在 PatIndex() 中使用 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 Len() 作为 Length() 的替代语法；SQL Anywhere 则不支持此替代语法。
- 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 的用于数据类型转换的 Convert() 以外， SQL Anywhere 和 Sybase IQ 还支持 Cast()。

---

**注释** Cast() 是符合 ANSI 的名称。

---

- SQL Anywhere 和 Sybase IQ 支持 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，但需要注意 CIS 函数补偿性能方面的一些事宜。
- SQL Anywhere 支持 NEWID、 STRTOUID 和 UIDTOSTR 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。

---

**注释** 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

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。

---

**注释** 对 OLAP 函数的支持是 Sybase 产品发展的一个快速发展的领域。请参见第 4 章 “SQL 函数”。另请参见《系统管理指南，卷 2》中的第 2 章 “使用 OLAP”。

---

## 系统函数

SQL Anywhere 和 Sybase IQ 不支持下列 Adaptive Server Enterprise 系统函数：

- curunreservedpgs() — dbspace 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

用户定义的函数 (UDF) 支持按如下方式改变:

- 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。

## 日期的算术表达式

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” 会利用两个项创建一个时间戳。

## SELECT INTO

语句中允许的表类型存在类似下面的区别:

```
select into table1 from table2
```

- Adaptive Server Enterprise 允许 `table1` 成为永久表、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。仅当选择到表中并指定多个列时，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。

## 可更新视图

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。

## UPDATE 和 DELETE 中的 FROM 子句

所有这三种产品均支持在 UPDATE 和 DELETE 中对多个表执行 FROM 子句。

## Transact-SQL 过程语言概述

存储过程语言是在存储过程和批处理语句中使用的 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 批处理语句概述

在 Transact-SQL 中，批处理语句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。

有关参数的信息，请参见《参考：语句和选项》中的 [PARAMETERS 语句 \[DBISQL\]](#)。

## 过程和批处理语句中的 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：

- 可以在一个批处理语句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 过程或批处理语句的一部分。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_il) from a) then set
 var4 = 100;
end if;
end;
```

## CASE 语句

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。

有关 Sybase IQ 和 Adaptive Server Enterprise 中的 CASE 表达式支持的详细比较，请参见第 21 页的“表达式”。

## 行级游标操作

所有这三种产品都支持按如下所示将游标与 UPDATE 和 DELETE 配合使用：

```
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

PRINT 命令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 dbisql 在 Sybase IQ 中运行。

---

**注释** Sybase IQ 用户可能会首选带 JDBC 的 dbisql，而不是 iAdaptive Server Anywhere JDBC 驱动程序（以前称为 JDBC-ODBC Bridge）。

---

## 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：

- **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')
```

## 从 Transact-SQL 过程中返回结果集

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。

Transact-SQL 过程的示例

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
```

### Watcom-SQL 过程的示例

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
```

### 多个结果集

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差别:

- `isql` 在一个流中显示所有结果。
- `dbisql` 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 `dbisql`。
- `dbisqlc` 提供 `RESUME` 命令以显示每个连续的结果集。

有关过程和结果的详细信息，请参见《系统管理指南，卷 2》中的第 1 章“使用过程和批处理”。

## Transact-SQL 过程中的变量

SQL Anywhere 和 Sybase IQ 使用 `SET` 语句来向过程中的变量赋值。在 Transact-SQL 中，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。

## 变量的顺序和保留

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）。变量仅在复合语句的持续时间内保留。

有关使用 SELECT 语句给变量赋值的详细信息，请参见第 501 页的“编写兼容的查询”。有关使用 SET 语句给变量赋值的详细信息，请参见《参考：语句和选项》中的 SET 语句 [ESQL]。

## 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'
```

表 A-2 描述了内置过程返回值及其含义：

**表 A-2：内置过程返回值**

| 值   | 含义          |
|-----|-------------|
| 0   | 过程执行时没有发生错误 |
| -1  | 缺少对象        |
| -2  | 数据类型错误      |
| -3  | 进程被选作死锁牺牲品  |
| -4  | 权限错误        |
| -5  | 语法错误        |
| -6  | 杂类用户错误      |
| -7  | 资源错误，如空间不足  |
| -8  | 非致命内部问题     |
| -9  | 达到系统限制      |
| -10 | 致命的内部不一致性   |
| -11 | 致命的内部不一致性   |
| -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。

有关 RAISERROR 语句的说明，请参见《参考：语句和选项》中的 [RAISERROR 语句 \[T-SQL\]](#)。

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 ON\_TSQL\_ERROR 数据库选项设置为 CONTINUE，RAISERROR 语句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，则将返回错误信息并丢失 RAISERROR 信息。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

您可以通过向 CREATE PROCEDURE 语句提供 ON EXCEPTION RESUME 子句来使 Watcom-SQL 术语过程以类似于 Transact-SQL 的方式处理错误：

```
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

以上各节强调与 Transact-SQL 的兼容性，还阐明了 Sybase IQ 和 SQL Anywhere 之间的许多区别。

本节指出了 Sybase IQ 和 SQL Anywhere 之间的其它区别和共享的功能。

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。

## 服务器和数据库启动与管理

请注意启动和管理数据库和服务器的下列区别：

- 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，但并不支持全部。Sybase IQ 支持其它服务器选项，但 SQL Anywhere 则不支持。
- Sybase IQ 提供 `stop_iq` 实用程序来关闭服务器。
- 在 Sybase IQ 和 SQL Anywhere 中，允许在 BACKUP 和 RESTORE 语句中使用的子句不同。
- 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 不支持下列 SQL Anywhere 实用程序：backup、compression、console、initialization、license、log transfer、log translation、rebuild、spawn、一些 transaction log 选项（-g、-il、-ir、-n、-x、-z）、uncompression、unload、upgrade 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。

## 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 和 TIMESTAMP\_FORMAT。

Sybase IQ 还包括许多不受 SQL Anywhere 支持的选项。请参见《参考：语句和选项》中的第 2 章“数据库选项”。

## 数据定义语言 (DDL)

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 和 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 CREATE ENCRYPTED DATABASE 和 CREATE DECRYPTED DATABASE 命令，而 SQL Anywhere 支持这些命令。请参见《Sybase IQ 中的高级安全性》。

## 数据操作语言 (DML)

- 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 和 VALIDATE TABLE。

---

**注释** 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，卷 1》的第 7 章“将数据移入和移出数据库”中的“数据提取选项”一节。

---

- Sybase IQ 支持 INSERT...LOCATION 语法，而 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。请参见《系统管理指南，卷 1》中的第 10 章“事务和版本控制”。
- 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。请参见《性能和调优指南》中的“导致 SQL Anywhere 进行处理的条件”。
- Sybase IQ 忽略 Adaptive Server Enterprise SELECT 语句中完全限定名称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`。

## 存储过程

请参见第 424 页的“[SQL Anywhere 支持的过程](#)”。

## Adaptive Server Enterprise 和 Sybase IQ

本节指出了 Sybase IQ 和 Adaptive Server Enterprise 之间的其它区别和共享的功能。

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。

## 存储过程

Sybase IQ 不再支持以下的 Adaptive Server 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`

## 系统视图

Sybase IQ 不再支持以下的 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
- syssservers
- sysssrvroles

- systhresholds
- sysusages

列名差异

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

## 英文

- ABS 函数 111
- ACOS 函数 111
- Adaptive Server Enterprise
  - 兼容性 481
- ALL
  - 条件 36
- AND 条件 42
- ANY
  - 条件 36
- ARGN 函数 112
- ASCII 函数 112
- ASCII 值 112, 119
- ASIN 函数 113
- ATAN 函数 113
- ATAN2 函数 114
- AVG 函数 114
- BETWEEN 条件 37
- BFILE 函数
  - 说明 115
- BIGINTTOHEX 函数 115
- BINARY 数据类型 71
- BIT 数据类型
  - Transact-SQL 76
  - 兼容性 487
- BIT\_LENGTH 函数 116
- BLOB 数据类型
  - LIKE 条件 40
- BYTE\_LENGTH 函数 116
- BYTE\_LENGTH64 函数
  - 说明 117
- BYTE\_SUBSTR 函数
  - 说明 117
- BYTE\_SUBSTR64 函数
  - 说明 117
- CASE 表达式 26
  - NULLIF 函数 197
- CAST 函数 83, 117
- Catalog
  - Adaptive Server Enterprise 兼容性 485
  - 系统表 429
  - Catalog 存储
    - IQ 485
    - 监控 389
    - 检验 418
  - CEIL 函数 118
  - CEILING 函数 119
  - char
    - 拆分为术语 285
  - CHAR 函数 119
  - CHAR 数据类型
    - 关于 63
  - CHAR\_LENGTH 函数 120
  - CHAR\_LENGTH64 函数
    - 说明 120
  - CHARACTER VARYING 数据类型
    - 关于 64
    - 剪裁尾随空白 65, 487
  - CHARACTER 数据类型
    - 关于 64
  - CHARINDEX 函数 121
  - CHECK 条件
    - Transact-SQL 494
  - CHECKPOINT 语句
    - 在检查点操作期间备份 413
  - CLOB 数据类型 40
    - LIKE 条件 40
  - COALESCE 函数 122
  - COL\_LENGTH 函数 122
  - COL\_NAME 函数 122
  - column
    - SYSCOLUMNS 系统视图 437
    - 和用户定义的数据类型 82
    - 命名 22
  - comparison
    - 关于 30

- COMPUTE 子句
  - Transact-SQL 503
- CONNECTION\_PROPERTY 函数 123
- CONTAINS 条件
  - 具有 TEXT 索引 41
  - 具有 WD 索引 41
- CONVERT 函数 83, 124
  - 日期到整数的转换 126
  - 日期到字符串的转换 126
  - 整数到日期的转换 126
  - 字符串到日期的转换 126
- CORR 函数 127
- COS 函数 128
- COT 函数 128
- COUNT 函数 131
- COVAR\_POP 函数 129
- COVAR\_SAMP 函数 130
- CPU 利用率
  - 数据库一致性检查程序 295
- CREATE DECRYPTED DATABASE 语句 519
- CREATE DEFAULT 语句
  - 不支持 496
- CREATE DOMAIN 语句
  - Transact-SQL 兼容性 496
  - 使用 81
- CREATE ENCRYPTED DATABASE 语句 519
- CREATE INDEX 语句
  - IQ 497
  - Transact-SQL 497
- CREATE RULE 语句
  - 不支持 496
- CREATE TABLE 语句
  - Transact-SQL 493
- CREATE TRIGGER
  - 不支持 496
- CUBE 运算
  - GROUPING 函数 162
- CUME\_DIST 函数 131
- CURRENT DATABASE
  - 特殊值 49
- CURRENT DATE
  - default 49
  - 特殊值 49
- CURRENT PUBLISHER
  - default 49
  - 特殊值 49
- CURRENT TIME
  - default 50
  - 特殊值 50
- CURRENT TIMESTAMP
  - default 50
  - 特殊值 50
- CURRENT USER
  - default 50
  - 特殊值 50
- data
  - 区分大小写 492
- DATALength 函数 132
- DATE 函数 133
- DATE 数据类型 76
- DATE\_ORDER 选项 80
- DATEADD 函数 133
- DATECEILING 函数 134
- DATEDIFF 函数 137
- DATEFLOOR 函数 139
- DATEFORMAT 函数 141
- DATENAME 函数 142
- DATEPART 函数 143
- DATEROUND 函数 144
- DATETIME 函数 146
- DAY 函数 146
- DAYNAME 函数 147
- DAYS 函数 147
- DB\_ID 函数 148
- DB\_NAME 函数 148
- DB\_PROPERTY 函数 149
- DBCC
  - 输出 299
  - 数据库检验 293
  - 性能 298
  - 运行时间 298
- dbcc
  - 线程使用情况 295
- DBCC\_LOG\_PROGRESS 选项 299
- dbinit 实用程序 484
- dbspace
  - 管理 484
  - 阻止读写操作 297
- DDL
  - SQL Anywhere 518
- DECIMAL 数据类型 67

- DEGREES 函数 150
- DENSE\_RANK 函数 150
- DIFFERENCE 函数 151
- DISK 语句
  - 不支持 484
- DML
  - SQL Anywhere 519
- DOUBLE 数据类型 69
- DOW 函数 152
- DUMMY 表 432
- ELSE
  - IF 表达式 26
- ENDIF
  - IF 表达式 26
- ERRORMSG 函数
  - SQL 语法 152
- EVENT\_CONDITION 函数 153
- EVENT\_CONDITION\_NAME 函数 154
- EVENT\_PARAMETER 函数 155
- EXISTS 条件 42
- EXP 函数 156
- EXP\_WEIGHTED\_AVG 函数 156
- FIRST\_VALUE 函数 157
- FLOAT 数据类型 69
- FLOOR 函数 159
- FOR BROWSE 语法
  - Transact-SQL 505
- FP 索引
  - 检验 296
- FROM 子句 102
  - UPDATE 和 DELETE 510
- GETDATE 函数 159
- GRAPHICAL\_PLAN 函数 160
- GROUP BY
  - 兼容性 502
- GROUP\_MEMBER 函数
  - SQL 语法 162
- GROUPING 函数 162
- GUID
  - NEWID 函数的 SQL 语法 192
  - STRTOUUID 函数的 SQL 语法 244
  - UIDTOSTR 函数的 SQL 语法 265
- HEXTOBIGINT 函数 163
- HEXTOINT 函数 164
  - ASE\_FUNCTION\_BEHAVIOR 选项 164
- HOLDLOCK 语法
  - Transact-SQL 505
- HOUR 函数 165
- HOURS 函数 165
- HTML\_DECODE 函数 166
- HTML\_ENCODE 函数 166
- HTML\_PLAN 167
- HTML\_PLAN 函数 167
- HTTP
  - 设置标头 417
  - 设置选项 418
- HTTP 函数 101
  - HTML\_DECODE 166
  - HTML\_ENCODE 166
  - HTTP\_DECODE 169
  - HTTP\_ENCODE 169
  - HTTP\_HEADER 169
  - HTTP\_VARIABLE 170
  - NEXT\_HTTP\_HEADER 195
  - NEXT\_HTTP\_VARIABLE 195
- HTTP\_DECODE 函数 169
- HTTP\_ENCODE 函数 169
- HTTP\_HEADER 函数 169
- HTTP\_VARIABLE 函数 170
- IF 表达式 26
- IFNULL 函数 170
- IMAGE 数据类型 71, 488
  - 兼容性 490
- IN 条件 41
- INDEX\_COL 函数 171
- INSERTSTR 函数 171
- INTEGER 数据类型 67
- INTTOHEX 函数 172
  - ASE\_FUNCTION\_BEHAVIOR 选项 172
- IQ Agent
  - port 7
  - 等待时间 8
- IQ 存储 485
- iq\_dummy 表 89
- IQDIR15 环境变量 7
- iqinit 实用程序 484
- IQLOGDIR15 环境变量 8
- IQPORT 环境变量 7
- IQTIMEOUT 环境变量
  - 指定 IQ 代理的等待时间 8
- IQTMP15 环境变量 9

- IS NULL 条件 42
- ISDATE 函数
  - SQL 语法 173
- ISNULL 函数 174
- ISNUMERIC 函数
  - SQL 语法 174
- Java
  - 用户定义的函数 109
- Java 数据类型
  - 兼容性 491
- Java 运行时环境
  - 设置 12
- JAVA\_HOME 环境变量 10
- LASTUSER
  - 特殊值 51
- LAST\_VALUE 函数 176
- LCASE 函数 178
- LD\_LIBRARY\_PATH 环境变量 10
- LEFT 函数 180
- LEN 函数
  - SQL 语法 181
- LENGTH 函数 181, 182
- LIBPATH 环境变量 10
- LIKE 条件 37, 40
  - LONG BINARY 数据 40
  - LONG VARCHAR 数据 40
  - 大对象数据 40
- LOB 数据类型
  - LIKE 条件 40
- LOCATE 函数 182
- LOG 函数 184
- LOG10 函数 184
- logarithm (base 10) 184
- LONG BINARY 数据类型 71, 488, 490
  - LIKE 条件 40
- LONG VARCHAR 数据类型 40
  - LIKE 条件 40
- LOWER 函数 184
- LTRIM 函数 185
- LVC 单元 298
- master 数据库
  - 不支持 483
- MAX 函数 186
- MEDIAN 函数 186
- memory
  - 使用 sp\_iqsysmon 监控 389
- MIN 函数 187
- MINUTE 函数 188
- MINUTES 函数 188
- MOD 函数 189
- MONEY 数据类型 70
- MONTH 函数 190
- MONTHNAME 函数 190
- MONTHS 函数 191
- MPXServerName 列 307
- Multiplex
  - 同步查询服务器 8
  - 系统过程 306
- nchar
  - 拆分为术语 286
- NEWID 函数
  - SQL 语法 192
- NEXT\_CONNECTION 函数 193
- NEXT\_DATABASE 函数 194
- NEXT\_HTTP\_HEADER 函数 195
- NEXT\_HTTP\_VARIABLE 函数 195
- NOT 条件 43
- NOW 函数 195
- NTILE 函数 196
- Null 比较
  - Transact-SQL 504
- NULL 值
  - 关于 60
- NULLIF 函数 27, 197
- NUMBER 函数 198
- NUMERIC 69
- OBJECT\_ID 函数 199
- OBJECT\_NAME 函数 199
- OCTET\_LENGTH 函数 200
- OLAP
  - DENSE\_RANK 函数 150
  - GROUPING 函数 162
  - NTILE 函数 196
  - PERCENT\_RANK 函数 201
  - PERCENTILE\_CONT 函数 203
  - PERCENTILE\_DISC 函数 205
  - RANK 函数 211
  - STDDEV 函数 238
  - VARIANCE 函数 267
  - 窗口规范 94
  - 窗口函数 94

- 窗口函数类型 94
- 窗口集合函数 92
- 窗口名称 94
- 分布函数 92
- 排名函数 92
- 数值函数 92
- 统计函数 92
- 行比较函数 92
- OLAP OVER 子句 94
- OLAP 函数
  - 兼容性 507
- ON EXCEPTION RESUME 子句
  - Transact-SQL 517
- Open Client 设置 13
- OR 关键字 42
- OVER 子句 94
- PATH 环境变量 10
- PATINDEX 函数 200
- PERCENT\_RANK 函数 201
- PERCENTILE\_CONT 函数 203
- PERCENTILE\_DISC 函数 205
- PI 函数 207
- POWER 函数 207
- PRINT 命令
  - Transact-SQL 512
- PROPERTY 函数 207
- PROPERTY\_DESCRIPTION 函数 208
- PROPERTY\_NAME 函数 209
- PROPERTY\_NUMBER 函数 209
- QUARTER 函数 210
- QUOTED\_IDENTIFIER 选项 28
- RADIANS 函数 210
- RAISERROR 语句
  - ON EXCEPTION RESUME 517
  - Transact-SQL 516
- RAND 函数 211
- RANK 函数 211
- REGR\_AVGX 函数 213
- REGR\_AVGY 函数 214
- REGR\_COUNT 函数 215
- REGR\_INTERCEPT 函数 215
- REGR\_R2 函数 217
- REGR\_SLOPE 函数 218
- REGR\_SXX 函数 219
- REGR\_SXY 函数 220
- REGR\_SYY 函数 221
- REMAINDER 函数 222
- REPEAT 函数 222
- REPLACE 函数 223
  - 在 SELECT INTO 语句中 23, 126, 171, 178, 180, 185, 222, 224, 226, 229, 249, 263
- REPLICATE 函数 224
- request\_level\_debugging
  - 关于 417
- resetclocks
  - sp\_iqcheckdb 选项 296
- REVERSE 函数
  - SQL 语法 225
- RIGHT 函数 225
- ROLLUP 运算
  - GROUPING 函数 162
- ROUND 函数 226
- ROWID 函数 228
- RTRIM 函数 229
- sa\_ansi\_standard\_packages 系统过程 413
- sa\_audit\_string 系统过程 413
- sa\_char\_terms 存储过程 285
- sa\_checkpoint\_execute 系统过程 413
- sa\_conn\_activity 系统过程
  - 语法 414
- sa\_conn\_info 系统过程 414
- sa\_conn\_properties 系统过程 415
- sa\_db\_info 系统过程 415
- sa\_db\_properties 系统过程 415
- sa\_dependent\_views 系统过程 285
- sa\_disable\_auditing\_type 系统过程 415
- sa\_enable\_auditing\_type 系统过程 416
- sa\_eng\_properties 系统过程 416
- sa\_external\_library\_unload 存储过程 285
- sa\_flush\_cache 系统过程 416
- sa\_get\_table\_definition 系统过程 424
- sa\_list\_external\_library 存储过程 286
- sa\_make\_object 系统过程 416
- sa\_nchar\_terms 存储过程 286
- sa\_rowgenerator 系统过程
  - 语法 417
- sa\_server\_option 系统过程 417
- sa\_set\_http\_header 系统过程 417
- sa\_set\_http\_option 系统过程 418
- sa\_table\_page\_usage 系统过程 418
- sa\_text\_index\_vocab 存储过程 286

- sa\_validate 系统过程
  - 语法 418
- sa\_verify\_password 系统过程 286
- SACHARSET 环境变量 11
- SALANG 环境变量 11
- SECOND 函数 229
- SECONDS 函数 230
- SELECT INTO
  - Transact-SQL 509
  - 使用 REPLACE 函数 23, 126, 171, 178, 180, 185, 222, 224, 226, 229, 249, 263
- SELECT 语句
  - Transact-SQL 501
- SET OPTION 语句
  - Transact-SQL 500
- SHARED 语法
  - Transact-SQL 505
- SIGN 函数 231
- SIGNAL 语句
  - Transact-SQL 516
- SIMILAR 函数 231
- SIN 函数 232
- SMALLDATETIME 数据类型 76
- SMALLINT 数据类型 67
- SMALLMONEY 数据类型 70
- SOME 条件 36
- SORTKEY 函数 232
- SOUNDEX 函数 236
- sp\_expireallpasswords 系统过程 287
- sp\_iq\_reset\_identity 系统过程 379
- sp\_iqaddlogin 系统过程 287
- sp\_iqbackupdetails 存储过程 288
- sp\_iqbackupsummary 存储过程 290
- sp\_iqbrestoreaction 存储过程 379
- sp\_iqcardinality\_analysis 系统过程 291
- sp\_iqcheckdb
  - DBCC\_LOG\_PROGRESS 选项 299
  - resetclocks 选项 296
  - 分配模式 295
  - 检查模式 296
  - 检验模式 296
  - 沙漏模式 297
  - 输出 299
  - 输出样本 299
  - 性能 298
  - 语法 293
  - 运行时间 298
- sp\_iqcheckdb 系统过程 293
- sp\_iqcheckoptions 系统过程 300
- sp\_iqcolumn 系统过程 303
- sp\_iqcolumnuse 系统过程 305
- sp\_iqconnection 系统过程 306
- sp\_iqcontext 系统过程 311
- sp\_iqcopyloginpolicy 系统过程 313, 363
- sp\_iqcursorinfo 系统过程 313
- sp\_iqdatatype 系统过程 316
- sp\_iqdbsize 系统过程 318
- sp\_iqdbspace 系统过程 320
- sp\_iqdbspaceinfo 系统过程 322
- sp\_iqdbspaceobjectinfo 系统过程 325
- sp\_iqdbstatistics 系统过程 329
- sp\_iqdroplogin 系统过程 330
- sp\_iqemptyfile 系统过程 331
- sp\_iquestdbspaces 系统过程 333
- sp\_iquestjoin 系统过程 332
- sp\_iquestspace 系统过程 335
- sp\_iqevent 系统过程 335
- sp\_iqfile 系统过程 338
- sp\_iqhelp 系统过程 340
- sp\_iqindex 系统过程 346
- sp\_iqindex\_alt 系统过程 346
- sp\_iqindexadvice 系统过程 348
- sp\_iqindexfragmentation 系统过程 349
- sp\_iqindexinfo
  - 显示索引信息 352, 353
- sp\_iqindexinfo 系统过程 351
- sp\_iqindexmetadata 系统过程 352
- sp\_iqindexsize 系统过程 354
- sp\_iqindexuse 系统过程 355
- sp\_iqjoinindex 系统过程 356
- sp\_iqjoinindexsize 系统过程 359
- sp\_iqlocks 系统过程 361
- sp\_iqmodifylogin 364
- sp\_iqmodifylogin 系统过程 364
- sp\_iqmpxinconnpoolinfo 存储过程 364
- sp\_iqmpxincheartbeatinfo 存储过程 365
- sp\_iqmpxinfo 存储过程 365
- sp\_iqmpxvalidate 存储过程 365
- sp\_iqobjectinfo 系统过程 365
- sp\_iqpassword 系统过程 368
- sp\_iqpkkeys 系统过程 368

- sp\_iqprocedure 系统过程 370
- sp\_iqprocparm 系统过程 372
- sp\_iqrebuildindex 系统过程 375, 380
- sp\_iqrename 系统过程 377
- sp\_iqsetcompression 系统过程 283
- sp\_iqshowcompression 系统过程 283
- sp\_iqshowpsexex 系统过程 381
- sp\_iqspaceinfo 系统过程 383
  - 输出样本 384
- sp\_iqspaceused 系统过程 384
- sp\_iqstatistics 系统过程 385
- sp\_iqstatus 系统过程 387
  - 输出样本 388
- sp\_iqsysmon 系统过程 389
- sp\_iqtable 系统过程 395
- sp\_iqtablesize 系统过程 397
- sp\_iqtableuse 系统过程 399
- sp\_iqtransaction 系统过程 399
- sp\_iqunusedcolumn 系统过程 402
- sp\_iqunusedindex 系统过程 403
- sp\_iqunusedtable 系统过程 405
- sp\_iqversionuse 系统过程 406
- sp\_iqview 系统过程 407
- sp\_iqwho 系统过程 408
- sp\_iqworkmon 系统过程 411
- sp\_login\_environment 系统过程 419
- sp\_remote\_columns 系统过程 419
- sp\_remote\_exported\_keys 系统过程 419, 420
- sp\_remote\_primary\_keys 系统过程
  - 语法 420
- sp\_remote\_tables 系统过程 420
- sp\_servercaps 系统过程 421
- sp\_tsq1\_environment 系统过程 421
- SPACE 函数 236
- SQL
  - IQ 术语区别 277
  - 用户定义的函数 109
- SQL Anywhere 482
  - 参照完整性约束 494
  - 管理角色 485
- SQL 函数
  - ERRORMSG 函数语法 152
  - GRAPHICAL\_PLAN 函数语法 160
  - GROUP\_MEMBER 函数语法 162
  - GROUPING 函数语法 162
  - HTML\_PLAN 函数语法 167
  - ISDATE 函数语法 173
  - ISNUMERIC 函数语法 174
  - LEN 函数的语法 181
  - NEWID 函数语法 192
  - REVERSE 函数的语法 225
  - STR\_REPLACE 函数语法 242
  - STRTOUUID 函数语法 244
  - UUIDTOSTR 函数语法 265
  - 兼容性 505
- SQL 语法
  - CURRENT DATABASE 特殊值 49
  - CURRENT PUBLISHER 特殊值 49
  - CURRENT USER 特殊值 50
  - LAST USER 特殊值 51
  - TIMESTAMP 特殊值 52
  - USER 特殊值 52
  - 标识符 18
- SQL 语句
  - CREATE DECRYPTED DATABASE 519
  - CREATE ENCRYPTED DATABASE 519
- SQL92 一致性 277
- SQLCODE
  - 特殊值 51
- SQLCONNECT 环境变量 11
- SQLFLAGGER 函数 237
- SQLSTATE
  - 特殊值 51
- SQRT 函数 238
- SQUARE 函数 238
- STDDEV 函数 238
- STDDEV\_POP 函数 240
- STDDEV\_SAMP 函数 241
- STR 函数 242
- STR\_REPLACE 函数
  - SQL 语法 242
- string
  - 长度 120
- STRING 函数 244
- STRTOUUID 函数
  - SQL 语法 244
- STUFF 函数 245
- SUBSTR 函数 246
- SUBSTRING 函数 246

- SUBSTRING64 函数
  - 说明 247
- SUM 函数 247
- SUSER\_ID 函数 248
- SUSER\_NAME 函数 248
- Sybase IQ 用户管理
  - sp\_iqdroplogin 330
- SYBASE 环境变量 12
- SYBASE\_JRE 环境变量 12
- SYBASE\_OCS 环境变量 13
- SYSIQBACKUPHISTORY 系统视图 445
- SYSIQBACKUPHISTORYDETAIL 系统视图 446
- SYSIQDBFILE 系统视图 447
- SYSIQDBSPACE 系统视图 448
- SYSIQIDX 系统视图 449
- SYSIQJOINIDX 系统视图 451
- SYSIQJOINIXCOLUMN 系统视图 452
- SYSIQJOINIXTABLE 系统视图 453
- SYSIQPARTITIONCOLUMN 系统视图 454
- SYSIQTAB 系统视图 454
- SYSPARTITION 系统视图 460
- SYSPARTITIONKEY 系统视图 461
- SYSPARTITIONSCHEME 系统视图 461
- SYSSUBPARTITIONKEY 系统视图 468
- TAN 函数 249
- TDS 连接 419
- TEXT 数据类型 64, 487
  - 兼容性 487, 490
- TEXT 索引
  - 统计信息 286
- THEN
  - IF 表达式 26
- threads
  - dbcc 295
- TIME 数据类型 76
- TIMESTAMP
  - 数据类型 76
  - 数据类型兼容性 489, 490
  - 特殊值 52
  - 转换为表达式 146
- TINYINT 数据类型 67
- TODAY 函数 249, 432
- Transact-SQL
  - 比较条件 32
  - 编写可移植 SQL 500
  - 变量 514
  - 表达式 27
  - 参照完整性约束 494
  - 常量 27
  - 创建兼容的数据库 491
  - 概述 482
  - 关于 481
  - 过程 510
  - 过程语言概述 510
  - 结果集 513
  - 局部变量 54
  - 连接 503
  - 批处理 511
  - 外连接运算符 25
  - 系统 Catalog 477
  - 用户定义的数据类型 82
  - 逐位运算符 24
  - 字符串 28
- Transact-SQL 兼容性
  - 数据库 492
- TRIM 函数 249
- TRUNCNUM 函数 250
- TS\_ARMA\_AR 函数 251
- TS\_ARMA\_CONST 函数 251
- TS\_ARMA\_MA 函数 251
- TS\_AUTO\_ARIMA 函数 252
- TS\_AUTO\_ARIMA\_OUTLIER 函数 252
- TS\_AUTO\_ARIMA\_RESULT\_AIC 函数 253
- TS\_AUTO\_ARIMA\_RESULT\_AICC 函数 253
- TS\_AUTO\_ARIMA\_RESULT\_BIC 函数 253
- TS\_AUTO\_ARIMA\_RESULT\_FORECAST\_ERROR
  - 函数 254
- TS\_AUTO\_ARIMA\_RESULT\_FORECAST\_VALUE
  - 函数 254
- TS\_AUTO\_ARIMA\_RESULT\_MODEL\_D 函数 254
- TS\_AUTO\_ARIMA\_RESULT\_MODEL\_P 函数 255
- TS\_AUTO\_ARIMA\_RESULT\_MODEL\_S 函数 255
- TS\_AUTO\_ARIMA\_RESULT\_RESIDUAL\_SIGMA
  - 函数 256
- TS\_AUTO\_UNI\_AR 函数 256
- TS\_AUTOCORRELATION 函数 252
- TS\_BOX\_COX\_XFORM 函数 256
- TS\_DIFFERENCE 函数 257
- TS\_DOUBLE\_ARRAY 函数 257

TS\_ESTIMATE\_MISSING 函数 257  
 TS\_GARCH 函数 258  
 TS\_GARCH\_RESULT\_A 函数 258  
 TS\_GARCH\_RESULT\_AIC 函数 258  
 TS\_GARCH\_RESULT\_USER 函数 259  
 TS\_INT\_ARRAY 函数 259  
 TS\_LACK OF FIT 函数 259  
 TS\_LACK OF FIT\_P 函数 260  
 TS\_MAX\_ARMA\_AR 函数 260  
 TS\_MAX\_ARMA\_CONST 函数 260  
 TS\_MAX\_ARMA\_LIKELIHOOD 函数 261  
 TS\_MAX\_ARMA\_MA 函数 261  
 TS\_OUTLIER\_IDENTIFICATION 函数 261  
 TS\_PARTIAL\_AUTOCORRELATION 函数 262  
 TS\_VWAP 函数 262  
 UCASE 函数 262  
 UNION  
     在子查询中 502  
 UNIQUEIDENTIFIER 数据类型 75  
 UNIQUEIDENTIFIERSTR 数据类型  
     关于 63  
 UPPER 函数 263  
 USER  
     特殊常量 432  
     特殊值 52  
 USER\_ID 函数 264  
 USER\_NAME 函数 264  
 UUID  
     NEWID 函数的 SQL 语法 192  
     STRTOUUID 函数的 SQL 语法 244  
     UIDTOSTR 函数的 SQL 语法 265  
 UIDTOSTR 函数  
     SQL 语法 265  
 VAR\_POP 函数 265  
 VAR\_SAMP 函数 266  
 VARBINARY 数据类型 71  
 VARCHAR 数据类型  
     关于 63, 64  
     剪裁尾随空白 65, 487  
 VARIANCE 函数 267  
 WEEKS 函数 269  
 WEIGHTED\_AVG 函数 270  
 WHERE 子句  
     Transact-SQL 503  
 WIDTH\_BUCKET 函数 271

WITHIN GROUP 子句 95  
 xp\_cmdshell 系统过程  
     语法 424  
 YEAR 函数 273  
 YEARS 函数 273  
 YMD 函数 275

## A

安装目录  
     关于 1

## B

百分点  
     使用 NTILE 函数计算 196  
 版式  
     文档 xxi  
 保留字 16  
     列表 16  
 备份  
     活动较少时 413  
     检查点操作期间 413  
 备份操作  
     摘要 290  
 备份历史记录文件  
     位置 8  
 本地计算机  
     环境设置 13  
 比较日期和时间 79  
 变量  
     Transact-SQL 514  
     本地 53  
     关于 53  
     连接级 55  
     全局 53, 55  
 标识符  
     SQL Anywhere 中的最大长度 18  
     SQL 语法 18  
     关于 18  
     区分大小写 492  
     唯一性 493

## 标识列

- 兼容性 495
- 支持作为缺省值 495

## 标准偏差

- 函数 238
- 样本函数 241
- 总体函数 240

## 标准偏差函数 237

## 表

- iq\_dummy 89
- Transact-SQL 493
- 每个连接索引的表数 282
- 显示信息 340

## 表达式 21

- CASE 26
- Transact-SQL 27
- 长度（以字节为单位） 132
- 转换为时间戳 146

## 表达式子查询

- IF 语句中 512

## 并置字符串 23

**C**

## 参数选择 112

## 参照完整性约束

- CASCADE 不受支持 494
- 兼容性 494

## 插入

- SQL Anywhere 519

## 查询

- Transact-SQL 501

## 查询服务器

- 同期 8

## 拆分

- 术语 285, 286

## 常量

- Transact-SQL 27
- 在表达式中 22

## 重命名对象

- sp\_iqrename 过程 377

## 触发器

- 不支持 496

## 窗口规范 94

## 窗口函数

- 窗口分区 93, 94
- 窗口函数类型 93
- 窗口名称或规范 93

## 窗口函数, 定义 94

## 窗口集合函数 92

## 窗口类型 94

## 窗口名称 94

## 创建

- 数据类型 81

## 存储过程

- Adaptive Server Enterprise 520
- sa\_ansi\_standard\_packages 413
- sa\_char\_terms 285
- sa\_dependent\_views 285
- sa\_external\_library\_unload 285
- sa\_list\_external\_library 286
- sa\_nchar\_terms 286
- sa\_rowgenerator 417
- sa\_text\_index\_vocab 286
- sa\_verify\_password 286, 418
- sp\_iqbackupdetails 288
- sp\_iqbackupsummary 290
- sp\_iqclient\_lookup 302
- sp\_iqmpxinconnpoolinfo 364
- sp\_iqmpxinheartbeatinfo 365
- sp\_iqmpxinfo 365
- sp\_iqmpxvalidate 365
- sp\_iqrestoreaction 379
- SQL Anywhere 520
- xp\_cmdshell 424

## 存储过程语言

- 概述 510

## 错误

- Transact-SQL 515, 517

## 错误消息

- ERRORMSG 函数 152
- 获取文本 152

**D**

- 大对象数据
  - LIKE 条件 40
- 代码页
  - 和数据存储 64
- 当前用户
  - 环境设置 13
- 登录策略
  - 复制 313, 363
  - 将用户指派给 364
- 登录管理
  - sp\_expireallpasswords 287
  - sp\_iqaddlogin 287
  - sp\_iqcopyloginpolicy 313, 363
- 定义窗口 94
- 对象
  - 重命名 377
  - 定义 416
  - 确定 ID 199
  - 确定名称 199
  - 显示信息 340

**E**

- 二进制数据
  - 兼容性 488

**F**

- 发布者
  - SQL Remote 49
- 反斜杠
  - SQL 标识符中不允许 18
- 反余弦值 111
- 反正切值 113
- 反正切值比率 114
- 反正弦值 113
- 返回值
  - 过程 516
- 分布函数 92
- 分隔 SQL 字符串 18
- 分离查询谓词 33

- 分配映射
  - 重置 297
- 分区
  - 一致性检查 295
- 分区表
  - 检验 297
- 分析函数 92
  - DENSE\_RANK 150
  - NTILE 196
  - PERCENT\_RANK 201
  - PERCENTILE\_CONT 203
  - PERCENTILE\_DISC 205
  - RANK 211
- 服务
  - 注册表条目 14
- 服务器
  - 属性 108
- 服务器管理
  - SQL Anywhere 和 IQ 517

**G**

- 高速缓存
  - 刷新 416
- 估计
  - 优化程序 44
- 故障排除
  - request\_level\_logging 417
- 关键字
  - SQL 15
  - 列表 16
- 规则
  - Transact-SQL 496
- 归类
  - SORTKEY 函数 232
- 过程
  - Transact-SQL 513
  - Transact-SQL 概述 510
  - 错误处理 515, 517
  - 返回值 516
  - 显示参数信息 372
  - 显示信息 340, 370
  - 转换 513
- 过程语言
  - 概述 510

## H

- 函数 89
  - ABS 函数 111
  - ACOS 函数 111
  - Adaptive Server Enterprise 系统函数 106
  - ARGN 函数 112
  - ASCII 函数 112
  - ASIN 函数 113
  - ATAN 函数 113
  - ATAN2 函数 114
  - AVG 函数 114
  - BFILE 函数 115
  - BIGINTTOHEX 函数 115
  - BIT\_LENGTH 函数 116
  - BYTE\_LENGTH 函数 116
  - BYTE\_LENGTH64 函数 117
  - BYTE\_SUBSTR 函数 117
  - BYTE\_SUBSTR64 函数 117
  - CAST 函数 117
  - CEIL 函数 118
  - CEILING 函数 119
  - CHAR 函数 119
  - CHAR\_LENGTH 函数 120
  - CHAR\_LENGTH64 函数 120
  - CHARINDEX 函数 121
  - COALESCE 函数 122
  - COL\_LENGTH 函数 122
  - COL\_NAME 函数 122
  - CONNECTION\_PROPERTY 函数 123
  - CONVERT 函数 124
  - CORR 函数 127
  - COS 函数 128
  - COT 函数 128
  - COUNT 函数 131
  - COVAR\_POP 函数 129
  - COVAR\_SAMP 函数 130
  - CUME\_DIST 函数 131
  - DATALENGTH 函数 132
  - DATE 函数 133
  - DATEADD 函数 133
  - DATECEILING 函数 134
  - DATEDIFF 函数 137
  - DATEFLOOR 函数 139
  - DATEFORMAT 函数 141
  - DATENAME 函数 142
  - DATEPART 函数 143
  - DATEROUND 函数 144
  - DATETIME 函数 146
  - DAY 函数 146
  - DAYNAME 函数 147
  - DAYS 函数 147
  - DB\_ID 函数 148
  - DB\_NAME 函数 148
  - DB\_PROPERTY 函数 149
  - DEGREES 函数 150
  - DENSE\_RANK 函数 150
  - DIFFERENCE 函数 151
  - DOW 函数 152
  - ERRORMSG 函数 SQL 语法 152
  - EVENT\_CONDITION 函数 153
  - EVENT\_CONDITION\_NAME 函数 154
  - EVENT\_PARAMETER 函数 155
  - EXP 函数 156
  - EXP\_WEIGHTED\_AVG 函数 156
  - FIRST\_VALUE 函数 157
  - FLOOR 函数 159
  - GETDATE 函数 159
  - GRAPHICAL\_PLAN 160
  - GROUP\_MEMBER 函数 SQL 语法 162
  - GROUPING 函数 SQL 语法 162
  - HEXTOBIGINT 函数 163
  - HEXTOINT 函数 164
  - HOUR 函数 165
  - HOURS 函数 165
  - HTML\_DECODE 函数 166
  - HTML\_ENCODE 函数 166
  - HTML\_PLAN 函数 167
  - HTTP 101
  - HTTP\_DECODE 函数 169
  - HTTP\_ENCODE 函数 169
  - HTTP\_HEADER 函数 169
  - HTTP\_VARIABLE 函数 170
  - IFNULL 函数 170

- INDEX\_COL 函数 171
- INSERTSTR 函数 171
- INTTOHEX 函数 172
- IQ 扩展 279
- ISDATE 函数 SQL 语法 173
- ISNULL 函数 174
- ISNUMERIC 函数 SQL 语法 174
- LAST\_VALUE 176
- LCASE 函数 178
- LEFT 函数 180
- LEN 函数的 SQL 语法 181
- LENGTH 函数 181, 182
- LOCATE 函数 182
- LOG 函数 184
- LOG10 函数 184
- LOWER 函数 184
- LTRIM 函数 185
- MAX 函数 186
- MEDIAN 函数 186
- MIN 函数 187
- MINUTE 函数 188
- MINUTES 函数 188
- MOD 函数 189
- MONTH 函数 190
- MONTHNAME 函数 190
- MONTHS 函数 191
- NEWID 函数 SQL 语法 192
- NEXT\_CONNECTION 函数 193
- NEXT\_DATABASE 函数 194
- NEXT\_HTTP\_HEADER 函数 195
- NEXT\_HTTP\_VARIABLE 函数 195
- NOW 函数 195
- NTILE 函数 196
- NULLIF 函数 197
- NUMBER 函数 198
- OBJECT\_ID 函数 199
- OBJECT\_NAME 函数 199
- OCTET\_LENGTH 函数 200
- PATINDEX 函数 200
- PERCENT\_RANK 函数 201
- PERCENTILE\_CONT 函数 203
- PERCENTILE\_DISC 函数 205
- PI 函数 207
- POWER 函数 207
- PROPERTY 函数 207
- PROPERTY\_DESCRIPTION 函数 208
- PROPERTY\_NAME 函数 209
- PROPERTY\_NUMBER 函数 209
- QUARTER 函数 210
- RADIANS 函数 210
- RAND 函数 211
- RANK 函数 211
- REGR\_AVGX 函数 213
- REGR\_AVGY 函数 214
- REGR\_COUNT 函数 215
- REGR\_INTERCEPT 函数 215
- REGR\_R2 函数 217
- REGR\_SLOPE 函数 218
- REGR\_SXX 函数 219
- REGR\_SXY 函数 220
- REGR\_SYY 函数 221
- REMAINDER 函数 222
- REPEAT 函数 222
- REPLACE 函数 223
- REPLICATE 函数 224
- REVERSE 函数的 SQL 语法 225
- RIGHT 函数 225
- ROUND 函数 226
- ROWID 函数 228
- RTRIM 函数 229
- SECOND 函数 229
- SECONDS 函数 230
- SIGN 函数 231
- SIMILAR 函数 231
- SIN 函数 232
- SORTKEY 函数 232
- SOUNDEX 函数 236
- SPACE 函数 236
- SQLFLAGGER 函数 237
- SQRT 函数 238
- SQUARE 函数 238
- STDDEV 函数 238
- STDDEV\_POP 函数 240
- STDDEV\_SAMP 函数 241

- STR 函数 242
- STR\_REPLACE 函数 SQL 语法 242
- STRING 函数 244
- STRTOUUID 函数 SQL 语法 244
- STUFF 函数 245
- SUBSTR 函数 246
- SUBSTRING 函数 246
- SUBSTRING64 函数 247
- SUM 函数 247
- SUSER\_ID 函数 248
- SUSER\_NAME 函数 248
- TAN 函数 249
- today 432
- TODAY 函数 249
- Transact-SQL 505
- TRIM 函数 249
- TRUNCNUM 函数 250
- TS\_ARMA\_AR 函数 251
- TS\_ARMA\_CONST 函数 251
- TS\_ARMA\_MA 函数 251
- TS\_AUTO\_ARIMA 函数 252
- TS\_AUTO\_ARIMA\_OUTLIER 函数 252
- TS\_AUTO\_ARIMA\_RESULT\_AIC 函数 253
- TS\_AUTO\_ARIMA\_RESULT\_AICC 函数 253
- TS\_AUTO\_ARIMA\_RESULT\_BIC 函数 253
- TS\_AUTO\_ARIMA\_RESULT\_FORECAST\_ERROR 函数 254
- TS\_AUTO\_ARIMA\_RESULT\_FORECAST\_VALUE 函数 254
- TS\_AUTO\_ARIMA\_RESULT\_MODEL\_D 函数 254
- TS\_AUTO\_ARIMA\_RESULT\_MODEL\_P 函数 255
- TS\_AUTO\_ARIMA\_RESULT\_MODEL\_S 函数 255
- TS\_AUTO\_ARIMA\_RESULT\_RESIDUAL\_SIGMA 函数 256
- TS\_AUTO\_UNI\_AR 函数 256
- TS\_AUTOCORRELATION 函数 252
- TS\_BOX\_COX\_XFORM 函数 256
- TS\_DIFFERENCE 函数 257
- TS\_DOUBLE\_ARRAY 函数 257
- TS\_ESTIMATE\_MISSING 函数 257
- TS\_GARCH 函数 258
- TS\_GARCH\_RESULT\_A 函数 258
- TS\_GARCH\_RESULT\_AIC 函数 258
- TS\_GARCH\_RESULT\_USER 函数 259
- TS\_INT\_ARRAY 函数 259
- TS\_LACK OF FIT 函数 259
- TS\_LACK OF FIT\_P 函数 260
- TS\_MAX\_ARMA\_AR 函数 260
- TS\_MAX\_ARMA\_CONST 函数 260
- TS\_MAX\_ARMA\_LIKELIHOOD 函数 261
- TS\_MAX\_ARMA\_MA 函数 261
- TS\_OUTLIER\_IDENTIFICATION 函数 261
- TS\_PARTIAL\_AUTOCORRELATION 函数 262
- TS\_VWAP 函数 262
- UCASE 函数 262
- UPPER 函数 263
- USER\_ID 函数 264
- USER\_NAME 函数 264
- UUIDTOSTR 函数 SQL 语法 265
- VAR\_POP 函数 265
- VAR\_SAMP 函数 266
- VARIANCE 函数 267
- WEEKS 函数 269
- WEIGHTED\_AVG 函数 270
- WIDTH\_BUCKET 函数 271
- YEAR 函数 273
- YEARS 函数 273
- YMD 函数 275
- 按字母顺序排列的列表 111
- 窗口化集合 92
- 分布 92
- 分析 92
- 集合 90
- 排名 92
- 日期和时间 97
- 时序 110
- 数据类型转换 96
- 数值 92, 102
- 统计 92
- 一致的结果 102
- 用户定义的 109
- 杂项 110
- 字符串 103

- 函数, 集合
  - GROUPING 162
- 函数, 数据类型转换
  - ISDATE 173
- 函数, 杂项
  - ERRORMSG 152
  - ISNUMERIC 174
  - NEWID 192
- 函数, 字符串 167, 181, 225, 242
  - STRTUUUID 244
  - UIDTOSTR 265
- 环境变量
  - IQLOGDIR15 8
  - IQPORT 7
  - IQTIMEOUT 8
  - PATH 10
  - SQLCONNECT 11
  - SYBASE 12
  - SYBASE\_JRE 12
  - SYBASE\_OCS 13
  - 关于 4
- 缓冲区高速缓存
  - 使用 sp\_iqsysmon 监控 389
- 恢复操作
  - 一致状态 379
- 活动超时
  - 数据库服务器 417

## J

- 集合函数 90
  - AVG 114
  - COUNT 131
  - MAX 186
  - MIN 187
  - STDDEV 238
  - STDDEV\_POP 240
  - STDDEV\_SAMP 241
  - SUM 247
  - VAR\_POP 265
  - VAR\_SAMP 266
  - VARIANCE 267
- 计算列
  - 不支持 495
- 监控器
  - sp\_iqsysmon 过程 389

- 兼容性
  - Adaptive Server Enterprise 481
  - 参照完整性约束 494
- 检查约束 494
  - Transact-SQL 兼容性 494
  - 强制 494
- 检验
  - Catalog 存储 418
  - 分区表 297
  - 键 296
  - 口令 286, 418
  - 索引 296
- 剪裁尾随空白 65, 487
- 键
  - 检验 296
  - 显示信息 368
- 建议
  - 存储 348
  - 清除 348
  - 显示 348
- 角色
  - Adaptive Server Enterprise 485
- 结果集
  - Transact-SQL 513
- 局部变量
  - 关于 53
- 绝对值 111

## K

- 可移植 SQL 500
- 空白
  - 剪裁尾随 65, 487
- 空值
  - Transact-SQL 兼容性 493
- 口令
  - sa\_verify\_password 系统过程 286, 418
  - 到期 287
  - 区分大小写 492
  - 添加或修改 368
- 括号
  - SQL 标识符 18
  - 数据库对象 18

## L

- 类型
  - 关于数据类型 63
- 类型转换 83
- 立即刷新 286
- 连接
  - Transact-SQL 503
  - 确定 ID 号 193
  - 属性 107
  - 数量 306
  - 外部运算符 25
  - 显示信息 408
  - 自动 278
- 连接等同性条件 48
- 连接级变量
  - 关于 55
- 连接属性值 123
- 连接数
  - 确定 306
- 连接索引
  - 表数
    - 查询
      - 每个块的表数 282
    - 显示信息 356
- 连接信息
  - sp\_iqcontext 311
- 连接运算符
  - ANSI 504
  - Transact-SQL 503
- 链式外连接 503
- 列表
  - 外部库 286
- 列长度 122
- 列名 122
- 列缺省值
  - 不支持 495
- 临时表
  - Transact-SQL 496
- 零长度字符串
  - Transact-SQL 505

## M

- 命令行选项
  - 替换 417
- 命名空间
  - 索引 493
- 模式匹配
  - 关于 37
  - 和归类 39
  - 限制 38
- 目录结构 1

## P

- 排版
  - 约定 xxii
- 排名函数 92
- 批处理
  - Transact-SQL 概述 511
  - 编写 511
- 撇号
  - 字符串中 20
- 平方根函数 238
- 平均值 114

## Q

- 嵌套外连接 503
- 请求级别的日志记录
  - 关于 417
- 区分大小写
  - Transact-SQL 兼容性 492
  - 比较条件 31
  - 标识符 492
  - 和模式匹配 39
  - 口令 492
  - 数据 492
  - 数据库 492
  - 用户 ID 492
  - 用户定义的数据类型 492

权限  
 Adaptive Server Enterprise 498  
 SYSCOLAUTH 系统视图 435  
 系统视图 471  
 全局变量  
 关于 53, 55  
 兼容性 57  
 列表 56  
 全局唯一标识符  
 NEWID 函数的 SQL 语法 192  
 缺省值  
 CURRENT DATABASE 49  
 CURRENT DATE 49  
 CURRENT PUBLISHER 49  
 CURRENT TIME 50  
 CURRENT TIMESTAMP 50  
 CURRENT USER 50  
 LAST USER 51  
 TIMESTAMP 52  
 Transact-SQL 496  
 USER 52  
 不支持 495

## R

日期  
 查询 79  
 将字符串解释为日期 80  
 决定当前的 195, 249  
 算术表达式 509  
 日期和时间函数 97  
 DATE 133  
 DATEADD 133  
 DATECEILING 134  
 DATEDIFF 137  
 DATEFLOOR 139  
 DATEFORMAT 141  
 DATENAME 142  
 DATEPART 143  
 DATEROUND 144  
 DATETIME 146  
 DAY 146  
 DAYNAME 147  
 DAYS 147

DOW 152  
 GETDATE 159  
 HOUR 165  
 HOURS 165  
 IQ 功能 277  
 MINUTE 188  
 MINUTES 188  
 MONTH 190  
 MONTHNAME 190  
 MONTHS 191  
 NOW 195  
 QUARTER 210  
 SECOND 229  
 SECONDS 230  
 TODAY 249  
 WEEKS 269  
 YEAR 273  
 YEARS 273  
 YMD 275  
 获取一致的结果 99  
 日期和时间数据类型  
 兼容性 489  
 日期时间和时间数据类型  
 兼容性 490

## S

三值逻辑  
 NULL 值 61  
 关于 43  
 沙漏模式 297  
 设备  
 管理 484  
 审计  
 禁用 415  
 启用 416  
 添加注释 413  
 时间  
 查询 79  
 时序函数 110  
 实例  
 外部库 286  
 实用程序  
 SQL Anywhere 517

## 索引

### 识别

外部库 286

### 事件

EVENT\_CONDITION 函数 153

EVENT\_CONDITION\_NAME 函数 154

EVENT\_PARAMETER 函数 155

显示信息 335, 340

### 事务管理

使用 sp\_iqsysmon 监控 389

### 事务日志

添加字符串 413

### 视图

可更新 510

系统视图 477

显示信息 340

### 属性

ID 说明 208

服务器 108

服务器级 207

连接 107

确定编号 209

确定名称 209

数据库 108

### 术语

拆分 285, 286

行位置 285, 286

### 数据库

检验 Catalog 存储 418

区分大小写 492

确定 ID 号 148, 194

确定名称 148

属性 108

属性值 149

系统表 429

系统过程 283

### 数据库对象

标识 18

确定 ID 199

确定名称 199

### 数据库服务器

覆盖命令行选项 417

### 数据库管理员

角色 486

### 数据库选项

DATE\_ORDER 80

QUOTED\_IDENTIFIER 28

### 数据类型

Adaptive Server Enterprise 486

IMAGE 71, 488, 490

IQ 486

LONG BINARY 71, 488, 490

numeric 67

SQL Anywhere 486

TEXT 64, 487, 490

UNIQUEIDENTIFIERSTR 63

二进制 71

关于 63

和兼容性 83

和舍入误差 69

日期和时间 76

显示信息 316, 340

用户定义的 81

字符 63

### 数据类型兼容性

bit 数据 487

IMAGE 数据 490

Java 数据 491

TEXT 数据 490

二进制数据 488

日期和时间数据 489

日期时间和时间数据 490

数值数据 490

字符数据 487

### 数据类型转换

BIT 到 CHAR 86

BIT 到 VARCHAR 86

CHAR 到 BIT 86

VARCHAR 到 BIT 86

从 BIT 到 BINARY 84

从 BIT 到 VARBINARY 84

关于 83

函数 96

### 数据类型转换函数

BIGINTTOHEX 115

CAST 117

CONVERT 124

HEXTOBIGINT 163

- HEXTOINT 164
  - INTTOHEX 172
  - 数学表达式 23
  - 数值函数 92, 102
    - ABS 111
    - ACOS 111
    - ASIN 113
    - ATAN 113
    - ATAN2 114
    - CEIL 118
    - CEILING 119
    - COS 128
    - COT 128
    - DEGREES 150
    - EXP 156
    - FLOOR 159
    - LOG 184
    - LOG10 184
    - MOD 189
    - PI 207
    - POWER 207
    - RADIANS 210
    - RAND 211
    - REMAINDER 222
    - ROUND 226
    - SIGN 231
    - SIN 232
    - SQRT 238
    - SQUARE 238
    - TAN 249
    - TRUNCNUM 250
    - WIDTH\_BUCKET 271
  - 一致的结果 102
  - 数值数据类型
    - 兼容性 490
  - 数字 22
  - 数字的对数 184
  - 刷新
    - 立即 286
  - 双引号
    - SQL 标识符中不允许 18
    - 数据库对象 18
  - 搜索条件
    - ALL 或 ANY 条件 36
    - BETWEEN 条件 37
    - CONTAINS 条件 41
    - EXISTS 条件 42
    - IN 条件 41
    - IS NULL 条件 42
    - LEADING SUBSTRING SEARCH 条件 37
    - LIKE 条件 37
    - NOT 条件 43
    - 比较条件 31
    - 关于 30
    - 三值逻辑 43
    - 真值条件 43
    - 子查询 32
  - 算术表达式 23
    - 关于日期 509
  - 索引
    - Adaptive Server Enterprise 497
    - IQ 497
    - SQL Anywhere 497
    - Transact-SQL 493
    - 系统视图 445
  - 索引首选项提示 46
  - 锁
    - 显示 361
- ## T
- 特殊值
    - CURRENT DATABASE 49
    - CURRENT DATE 49
    - CURRENT PUBLISHER 49
    - CURRENT TIME 50
    - CURRENT TIMESTAMP 50
    - CURRENT USER 50
    - LAST USER 51
    - SQLCODE 51
    - SQLSTATE 51
    - TIMESTAMP 52
    - USER 52
  - 特殊字符
    - 字符串中 20
  - 提示
    - 索引首选项 46
    - 执行阶段 47, 48
  - 体系结构
    - Adaptive Server 483

- 条件
  - 用户提供的用于查询的条件 44
- 条件提示字符串 45
- 通用唯一标识符
  - NEWID 函数的 SQL 语法 192
- 统计函数 92
- 统计信息
  - TEXT 索引 286
- 退出时间
  - 数据库服务器 417

## W

- 外部库
  - 列表 286
  - 卸载 285
- 外键
  - 系统视图 442
- 外连接
  - Transact-SQL 503
  - 和子查询 22
  - 链式 503
  - 嵌套的 503
  - 运算符 25
- 尾随空白
  - 剪裁 65, 487
- 位长度 116
- 谓词
  - 分离 33
  - 关于 30
- 文档
  - SQL Anywhere xx
  - Sybase IQ xix
  - 约定 xxi, xxii
- 文件
  - 位置 2
- 文字
  - 最大长度 281

## X

- 系统 Catalog 434
  - Adaptive Server Enterprise 兼容性 485
  - Transact-SQL 477
- 系统安全员
  - Adaptive Server Enterprise 485
- 系统变量 55
- 系统表
  - Adaptive Server Enterprise 兼容性 485
  - DUMMY 432
  - 关于 429
  - 显示信息 340
- 系统调用
  - xp\_cmdshell 系统过程 424
  - 从存储过程 424
- 系统管理员
  - Adaptive Server Enterprise 485
- 系统过程
  - sa\_ansi\_standard\_packages 413
  - sa\_audit\_string 413
  - sa\_checkpoint\_execute 413
  - sa\_conn\_activity 414
  - sa\_conn\_info 414
  - sa\_conn\_properties 415
  - sa\_db\_info 415
  - sa\_db\_properties 415
  - sa\_dependent\_views 285
  - sa\_disable\_auditing\_type 415
  - sa\_enable\_auditing\_type 416
  - sa\_eng\_properties 416
  - sa\_flush\_cache 416
  - sa\_get\_table\_definition 424
  - sa\_make\_object 416
  - sa\_rowgenerator 417
  - sa\_server\_option 417
  - sa\_set\_http\_header 417
  - sa\_set\_http\_option 418
  - sa\_table\_page\_usage 418
  - sa\_validate 418
  - sa\_verify\_password 286, 418

- sp\_expireallpasswords 287
- sp\_iqaddlogin 287
- sp\_iqbackupdetails 288
- sp\_iqbackupsummary 290
- sp\_iqcardinality\_analysis 291
- sp\_iqcheckdb 293
- sp\_iqcheckoptions 300
- sp\_iqclient\_lookup 302
- sp\_iqcolumn 303
- sp\_iqcolumnuse 305
- sp\_iqconnection 306
- sp\_iqcontext 311
- sp\_iqcopyloginpolicy 313, 363
- sp\_iqcursorinfo 313
- sp\_iqdatatype 316
- sp\_iqdbsize 318
- sp\_iqdbspaceobjectinfo 325
- sp\_iqdbstatistics 329
- sp\_iqdroplogin 330
- sp\_iqemptyfile 331
- sp\_iqestdbspaces 333
- sp\_iqestjoin 332
- sp\_iqestspace 335
- sp\_iqevent 335
- sp\_iqfile 338
- sp\_iqhelp 340
- sp\_iqindex 346
- sp\_iqindex\_alt 346
- sp\_iqindexadvice 348
- sp\_iqindexsize 354
- sp\_iqindexuse 355
- sp\_iqjoinindex 356
- sp\_iqjoinindexsize 359
- sp\_iqmodifylogin 364
- sp\_iqmpxinconnpoolinfo 364
- sp\_iqmpxinheartbeatinfo 365
- sp\_iqmpxinfo 365
- sp\_iqmpxvalidate 365
- sp\_iqobjectinfo 365
- sp\_iqpassword 368
- sp\_iqpkeys 368
- sp\_iqprocedure 370
- sp\_iqprocparm 372
- sp\_iqrename 377
- sp\_iqrestoreaction 379
- sp\_iqsetcompression 283
- sp\_iqshowcompression 283
- sp\_iqshowpsexec 381
- sp\_iqspaceinfo 383
- sp\_iqspaceused 384
- sp\_iqstatistics 385
- sp\_iqstatus 387
- sp\_iqsysmon 389
- sp\_iqtable 395
- sp\_iqtablesize 397
- sp\_iqtableuse 399
- sp\_iqtransaction 399
- sp\_iqunusedcolumn 402
- sp\_iqunusedindex 403
- sp\_iqunusedtable 405
- sp\_iqversionuse 406
- sp\_iqview 407
- sp\_iqwho 408
- sp\_iqworkmon 411
- sp\_login\_environment 419
- sp\_remote\_columns 419
- sp\_remote\_exported\_keys 419, 420
- sp\_remote\_primary\_keys 420
- sp\_remote\_tables 420
- sp\_servercaps 421
- sp\_tsq\_environment 421
- xp\_cmdshell 424
- 关于 283
- 显示信息 340
- 系统函数 105
  - COL\_LENGTH 122
  - COL\_NAME 122
  - CONNECTION\_PROPERTY 123
  - DATALENGTH 132
  - DB\_ID 148
  - DB\_NAME 148
  - DB\_PROPERTY 149
  - EVENT\_CONDITION 153
  - EVENT\_CONDITION\_NAME 154
  - EVENT\_PARAMETER 155
  - INDEX\_COL 171
  - NEXT\_CONNECTION 193
  - NEXT\_DATABASE 194
  - OBJECT\_ID 199
  - OBJECT\_NAME 199
  - PROPERTY 207
  - PROPERTY\_DESCRIPTION 208

- PROPERTY\_NAME 209  
 PROPERTY\_NUMBER 209  
 SUSER\_ID 248  
 SUSER\_NAME 248  
 Transact-SQL 508  
 USER\_ID 264  
 USER\_NAME 264  
 系统视图  
   Adaptive Server Enterprise 521  
   SYSARTICLE 433  
   SYSARTICLECOL 433  
   SYSARTICLECOLS 433  
   SYSARTICLES 433  
   SYSCAPABILITIES 434  
   SYSCAPABILITY 434  
   SYSCAPABILITYNAME 434  
   SYSCATALOG 434  
   SYSCHECK 435  
   SYSCOLAUTH 435  
   SYSCOLLATION 435  
   SYSCOLLATIONMAPPINGS 435  
   SYSCOLPERM 436  
   SYSCOLSTAT 436  
   SYSCOLSTATS 436  
   SYSCOLUMN 437  
   SYSCOLUMNS 437  
   SYSCOLUMNS ASE 兼容性视图 437  
   SYSCONSTRAINT 438  
   SYSDBFILE 438  
   SYSDBSPACE 438  
   SYSDBSPACEPERM 439  
   SYSDEPENDENCY 439  
   SYSDOMAIN 439  
   SYSEVENT 440  
   SYSEVENTTYPE 440  
   SYSEXTERNALLOGIN 441  
   SYSEXTERNENV 440  
   SYSEXTERNENVOBJECT 441  
   SYSFILE 441  
   SYSFKCOL 442  
   SYSFKKEY 442  
   SYSFORIGNKEY 442  
   SYSFORIGNKEYS 442  
   SYSGROUP 443  
   SYSGROUPS 443  
   SYSHISTORY 443  
   SYSIDX 444  
   SYSIDXCOLUMN 444  
   SYSINDEX 444  
   SYSINDEXES 445  
   SYSINDEXES ASE 兼容性视图 445  
   SYSINFO 445  
   SYSIQBACKUPHISTORY 445  
   SYSIQBACKUPHISTORYDETAIL 446  
   SYSIQCOLUMN 447  
   SYSIQDBFILE 447  
   SYSIQDBSPACE 448  
   SYSIQFILE 449  
   SYSIQIDX 449  
   SYSIQINFO 450  
   SYSIQITAB 454  
   SYSIQJOINIDX 451  
   SYSIQJOININDEX 452  
   SYSIQJOINIXCOLUMN 452  
   SYSIQJOINIXTABLE 453  
   SYSIQMPXLOGINPOLICYOPTION 453  
   SYSIQMPXSERVER 453  
   SYSIQOBJECTS ASE 兼容性视图 454  
   SYSIQPARTITIONCOLUMN 454  
   SYSIQTABCOL 455  
   SYSIQTABLE 系统视图 456  
   SYSIQVINDEXTABLE ASE 兼容性视图 456  
   SYSIXCOL 兼容性视图 456  
   SYSJAR 457  
   SYSJARCOMPONENT 457  
   SYSJAVACLASS 457  
   SYSLOGINMAP 457  
   SYSLOGINPOLICY 458  
   SYSLOGINPOLICYOPTION 458  
   SYSLOGINS ASE 兼容性视图 458  
   SYSMVOPTION 458  
   SYSOBJECTS ASE 兼容性视图 459  
   SYSOPTIONS 460  
   SYSPARTITION 460  
   SYSPARTITIONKEY 461  
   SYSPARTITIONSCHEME 461  
   SYSPHYSIDX 462  
   SYSPROCAUTH 462  
   SYSPROCEDURE 462  
   SYSPROCPARMS 463  
   SYSPROCPERM 463  
   SYSPROCS 463

- SYSROXYTAB 464
  - SYSPUBLICATION 464
  - SYSPUBLICATIONS 464
  - SYSREMARK 464
  - SYSREMOTEOPTION 465
  - SYSREMOTEOPTION2 465
  - SYSREMOTEOPTIONS 465
  - SYSREMOTEOPTIONTYPE 466
  - SYSREMOTETYPE 466
  - SYSREMOTETYPES 466
  - SYSREMOTEUSER 466
  - SYSREMOTEUSERS 467
  - SYSSCHEDULE 467
  - SYSSOURCE 467
  - SYSSQLSERVERTYPE 468
  - SYSSUBPARTITIONKEY 468
  - SYSSUBSCRIPTION 468
  - SYSSUBSCRIPTIONS 468
  - SYSSYNC 469
  - SYSSYNCPROFILE 469
  - SYSSYNCS 469
  - SYSSYNCSUBSCRIPTIONS 470
  - SYSSYNCSUSERS 470
  - SYSTAB 471
  - SYSTABAUTH 471
  - SYSTABCOL 471
  - SYSTABLE 471
  - SYSTABLEPERM 472
  - SYSTEXTCONFIG 472
  - SYSTEXTIDX 472
  - SYSTEXTIDXTAB 473
  - SYSTRIGGER 473
  - SYSTRIGGERS 473
  - SYSTYPEMAP 473
  - SYSTYPES ASE 兼容性视图 474
  - SYSUSER 474
  - SYSUSERAUTH 474
  - SYSUSERAUTHORITY 474
  - SYSUSERLIST 475
  - SYSUSERMESSAGE 475
  - SYSUSEROPTIONS 475
  - SYSUSERPERM 475
  - SYSUSERPERMS 476
  - SYSUSERS ASE 兼容性视图 476
  - SYSUSERTYPE 476
  - SYSVIEW 476
  - SYSVIEWS 477
  - SYSWEBSERVICE 477
  - 统一 432
  - 卸载
    - 外部库 285
  - 星期几 (DOW) 152
  - 行
    - 计数 131
  - 性能
    - sp\_iqshowpsexec 连接信息 381
    - sp\_iqsysmon 过程 389
    - 监控 389
  - 虚拟 IQ 表 89
    - 获取一致的结果 99
  - 选项
    - DBCC\_LOG\_PROGRESS 299
    - QUOTED\_IDENTIFIER 28
    - SQL Anywhere 518
    - 系统视图 460, 475
  - 选择性
    - 提示 46
    - 显式 44
    - 用户提供的条件 44
- ## Y
- 样本方差函数 266
  - 一致性检查
    - 分区 295
  - 一致状态
    - 恢复 379
  - 引号
    - SQL 标识符 18
    - 数据库对象 18
    - 字符串 20
  - 用户
    - 连接数 306
    - 删除 330
    - 添加 287
    - 显示信息 408
    - 修改 364

- 用户 ID
    - Adaptive Server Enterprise 498
    - 区分大小写 492
    - 通过用户名确定 248, 264
  - 用户定义的函数 109
    - 兼容性 509
  - 用户定义的数据类型
    - Transact-SQL 82
    - 关于 81
    - 区分大小写 492
  - 用户管理。请参见登录管理
  - 用户名
    - 通过用户 ID 确定 248, 264
  - 用户提供的关于连接等同性条件的提示 48
  - 用户提供的条件
    - 用于查询 44
  - 用户提供的条件提示、准则和用法 49
  - 用户提供的条件提示字符串 45
  - 用户提供的条件选择性 44
  - 优化程序
    - 估计 44
    - 用户定义的选择性 44
  - 游标
    - IQ 中的行级 512
    - Transact-SQL 512
    - 显示信息 313
  - 有用性提示 48
  - 余切值 128
  - 余弦值 128
  - 语法
    - CURRENT DATABASE 特殊值 49
    - CURRENT PUBLISHER 特殊值 49
    - CURRENT USER 特殊值 50
    - LAST USER 特殊值 51
    - SQL 标识符 18
    - TIMESTAMP 特殊值 52
    - USER 特殊值 52
    - 文档约定 xxi
  - 语句
    - CREATE DEFAULT 496
    - CREATE DOMAIN 496
    - CREATE RULE 496
    - CREATE TABLE 493
    - DISK INIT 484
    - DISK MIRROR 484
    - DISK REFIT 484
    - DISK REINIT 484
    - DISK REMIRROR 484
    - DISK UNMIRROR 484
    - RAISERROR 516, 517
    - SELECT 501
    - SIGNAL 516
  - 语言
    - 指定 11
  - 域
    - 关于 81
  - 预取
    - 使用 sp\_iqsysmon 监控 389
  - 远程表
    - 列 419, 420
    - 列表 420
  - 远程服务器
    - 功能 421
  - 约定
    - 排版 xxii
    - 文档 xxi, xxii
    - 语法 xxi
  - 运算符
    - 比较运算符 31
    - 优先级 25
  - 运算符的优先级 25
- ## Z
- 杂项函数
    - ARGN 112
    - COALESCE 122
    - IFNULL 170
    - ISNULL 174
    - NULLIF 197
    - NUMBER 198
    - ROWID 228
    - SQLFLAGGER 237
  - 摘要 288, 364, 365
  - 正切值 249
  - 执行阶段提示 47
  - 指数函数 156

- 中括号
  - SQL 标识符 18
  - 数据库对象 18
- 逐位运算符 24
- 主键
  - UUID 和 GUID 192
  - 生成唯一值 192
  - 使用 UUID 生成唯一值 192
  - 显示信息 368
- 注册表条目
  - 关于 13
- 注释
  - 注释指示符 59
- 装载格式
  - Transact-SQL 和 SQL Anywhere 500
- 子查询
  - Adaptive Server Enterprise 502
  - IQ 502
  - IQ 实现 279
  - SQL Anywhere 502
  - 分离 33
  - 在表达式中 22
  - 在搜索条件中 22
- 子查询, 分离 34
- 子查询的分离 34
- 子句
  - ON EXCEPTION RESUME 517
- 自动连接
  - 和外键 278
- 字符串 20, 22
  - insert 171
  - SOUNDEX 函数 236
  - Transact-SQL 28
  - 并置 23, 224, 244
  - 并置运算符 23
  - 常量 20, 22
  - 长度 116
  - 返回子字符串 246
  - 分隔符 28
  - 关于 20
  - 决定长度 181, 182
  - 确定相似性 231
  - 删除空白 249
  - 删除前导空白 185
  - 删除尾随空白 229
  - 特殊字符 20
  - 替换子串 223
  - 位置 121
  - 转换为大写形式 262
  - 转换为小写 178, 184
  - 字符串 20
- 字符串函数 103
  - ASCII 112
  - BFILE 115
  - BIT\_LENGTH 116
  - BYTE\_LENGTH 116
  - BYTE\_LENGTH64 117
  - BYTE\_SUBSTR 117
  - BYTE\_SUBSTR64 117
  - CHAR 119
  - CHAR\_LENGTH 120
  - CHAR\_LENGTH64 120
  - CHARINDEX 121
  - DIFFERENCE 151
  - INSERTSTR 171
  - LCASE 178
  - LEFT 180
  - LENGTH 181, 182
  - LOCATE 182
  - LOWER 184
  - LTRIM 185
  - OCTET\_LENGTH 200
  - PATINDEX 200
  - REPEAT 222
  - REPLACE 223
  - REPLICATE 224
  - RIGHT 225
  - RTRIM 229
  - SIMILAR 231
  - SORTKEY 232
  - SOUNDEX 236
  - SPACE 236
  - STR 242
  - STRING 244
  - STUFF 245
  - SUBSTR 246
  - SUBSTRING 246
  - SUBSTRING64 247
  - TRIM 249
  - UCASE 262
  - UPPER 263

## 索引

- 字符串文字
  - 最大长度 281
- 字符集
  - 指定 11
- 字符数据
  - 兼容性 487
- 字节长度 200
- 字母字符
  - 已定义 18
- 总体方差函数 265
- 组
  - Adaptive Server Enterprise 498