



参考：语句和选项

Sybase IQ 15.4

文档 ID: DC01142-01-1540-01

最后修订日期: 2011 年 11 月

版权所有 © 2011 Sybase, Inc. 保留所有权利。

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

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

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

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

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

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

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

本书中提到的所有其它公司和产品名均可能是与之相关的相应公司的商标。

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

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

目录

读者	1
SQL 语句	3
SQL 语法中的常见元素	3
语法约定	4
语句适用性指示符	4
ALLOCATE DESCRIPTOR 语句 [ESQL]	5
ALTER DATABASE 语句	7
ALTER DBSPACE 语句	8
ALTER DOMAIN 语句	12
ALTER EVENT 语句	13
ALTER FUNCTION 语句	14
ALTER INDEX 语句	16
ALTER LOGICAL SERVER 语句	17
ALTER LOGIN POLICY 语句	19
逻辑服务器访问配置	20
登录策略选项配置	21
ALTER LS POLICY 语句	22
ALTER MULTIPLEX RENAME 语句	23
ALTER MULTIPLEX SERVER 语句	24
ALTER PROCEDURE 语句	25
ALTER SERVER 语句	27
ALTER SERVICE 语句	29
ALTER TABLE 语句	31
ALTER TEXT CONFIGURATION 语句	39
ALTER TEXT INDEX 语句	39
ALTER USER 语句	39
ALTER VIEW 语句	41
确定和修复无效的相关视图	43
BACKUP 语句	44
BEGIN·····END 语句	49
BEGIN PARALLEL IQ·····END PARALLEL IQ 语句 ...	52

BEGIN TRANSACTION 语句 [T-SQL]	53
CALL 语句	55
CASE 语句	57
CHECKPOINT 语句	58
CLEAR 语句 [Interactive SQL]	59
CLOSE 语句 [ESQL] [SP]	59
COMMENT 语句	61
COMMENT ON LOGICAL SERVER 语句	62
COMMIT 语句	63
CONFIGURE 语句 [Interactive SQL]	64
CONNECT 语句 [ESQL] [Interactive SQL]	65
CREATE DATABASE 语句	68
CREATE DBSPACE 语句	77
CREATE DOMAIN 语句	80
CREATE EVENT 语句	82
CREATE EXISTING TABLE 语句	87
CREATE EXTERNLOGIN 语句	89
CREATE FUNCTION 语句	90
CREATE FUNCTION 语句 (Java UDF)	96
CREATE INDEX 语句	97
CREATE JOIN INDEX 语句	104
CREATE LOGICAL SERVER 语句	107
CREATE LOGIN POLICY 语句	108
CREATE MESSAGE 语句 [T-SQL]	110
CREATE MULTIPLEX SERVER 语句	111
CREATE PROCEDURE 语句	112
在过程中引用临时表	117
CREATE PROCEDURE 语句 [T-SQL]	118
CREATE PROCEDURE 语句 (外部过程)	120
CREATE PROCEDURE 语句 (Java UDF)	127
CREATE PROCEDURE 语句 (表 UDF)	128
CREATE SCHEMA 语句	131
CREATE SERVER 语句	132
CREATE SERVICE 语句	133

CREATE TABLE 语句	136
CREATE TEXT CONFIGURATION 语句	147
CREATE TEXT INDEX 语句	147
CREATE USER 语句	147
CREATE VARIABLE 语句	149
CREATE VIEW 语句	151
DEALLOCATE DESCRIPTOR 语句 [ESQL]	152
声明部分 [ESQL]	153
DECLARE 语句	154
DECLARE CURSOR 语句 [ESQL] [SP]	155
DECLARE CURSOR 语句 [T-SQL]	160
DECLARE LOCAL TEMPORARY TABLE 语句	161
DELETE 语句	163
DELETE (定位) 语句 [ESQL] [SP]	165
DESCRIBE 语句 [ESQL]	166
DISCONNECT 语句 [Interactive SQL]	169
DROP 语句	170
DROP CONNECTION 语句	172
DROP DATABASE 语句	173
DROP EXTERNLOGIN 语句	174
DROP LOGIN POLICY 语句	175
DROP LOGICAL SERVER 语句	176
DROP MULTIPLEX SERVER 语句	176
DROP SERVER 语句	177
DROP SERVICE 语句	178
DROP STATEMENT 语句 [ESQL]	179
DROP TEXT CONFIGURATION 语句	180
DROP TEXT INDEX 语句	180
DROP USER 语句	180
DROP VARIABLE 语句	181
EXECUTE 语句 [ESQL]	182
EXECUTE 语句 [T-SQL]	184
EXECUTE IMMEDIATE 语句 [ESQL] [SP]	185
EXIT 语句 [Interactive SQL]	187

FETCH 语句 [ESQL] [SP]	188
FOR 语句	191
FORWARD TO 语句	193
FROM 子句	194
GET DESCRIPTOR 语句 [ESQL]	200
GOTO 语句 [T-SQL]	201
GRANT 语句	202
IF 语句	207
IF 语句 [T-SQL]	209
INCLUDE 语句 [ESQL]	210
INSERT 语句	211
INSTALL JAVA 语句	218
IQ UTILITIES 语句	220
LEAVE 语句	221
LOAD TABLE 语句	223
存储大小	238
LOCK TABLE 语句	239
LOOP 语句	241
MESSAGE 语句	242
OPEN 语句 [ESQL] [SP]	245
OUTPUT 语句 [Interactive SQL]	247
PARAMETERS 语句 [Interactive SQL]	250
PREPARE 语句 [ESQL]	251
PRINT 语句 [T-SQL]	253
PUT 语句 [ESQL]	255
RAISERROR 语句 [T-SQL]	256
READ 语句 [Interactive SQL]	258
RELEASE SAVEPOINT 语句	259
REMOVE 语句	260
RESIGNAL 语句	261
RESTORE 语句	262
RESUME 语句	267
RETURN 语句	269
REVOKE 语句	270

ROLLBACK 语句	272
ROLLBACK TO SAVEPOINT 语句	273
ROLLBACK TRANSACTION 语句 [T-SQL]	274
SAVEPOINT 语句	275
SAVE TRANSACTION 语句 [T-SQL]	276
SELECT 语句	277
SET 语句 [ESQL]	285
SET 语句 [T-SQL]	287
SET CONNECTION 语句 [ESQL] [Interactive SQL] ...	289
SET DESCRIPTOR 语句 [ESQL]	290
SET OPTION 语句	291
SET OPTION 语句 [Interactive SQL]	293
SET SQLCA 语句 [ESQL]	294
SIGNAL 语句	295
START DATABASE 语句 [Interactive SQL]	295
START ENGINE 语句 [Interactive SQL]	297
START JAVA 语句	298
STOP DATABASE 语句 [Interactive SQL]	298
STOP ENGINE 语句 [Interactive SQL]	299
STOP JAVA 语句	300
SYNCHRONIZE JOIN INDEX 语句	300
TRIGGER EVENT 语句	301
TRUNCATE TABLE 语句	302
UNION 操作	303
UPDATE 语句	304
UPDATE (定位) 语句 [ESQL] [SP]	308
WAITFOR 语句	309
WHENEVER 语句 [ESQL]	311
WHILE 语句 [T-SQL]	312
数据库选项	315
数据库选项简介	315
当前选项设置	316
数据库选项的作用域和持续时间	317
临时选项	317

公共选项	318
删除选项设置	318
初始选项设置	319
不建议使用的数据库选项	320
常规数据库选项	320
数据提取选项	327
Transact-SQL 兼容性选项	327
Transact-SQL 为实现 Adaptive Server	
Enterprise 兼容性的选项设置	329
Interactive SQL 选项	330
按字母顺序排列的选项列表	332
AGGREGATION_PREFERENCE 选项	332
ALLOW_NULLS_BY_DEFAULT 选项 [TSQL] ...	333
ANSI_CLOSE_CURSORS_ON_ROLLBACK 选	
项 [TSQL]	333
ANSI_PERMISSIONS 选项 [TSQL]	334
ANSINULL 选项 [TSQL]	334
ANSI_SUBSTRING 选项 [TSQL]	335
ANSI_UPDATE_CONSTRAINTS 选项	335
ALLOW_READ_CLIENT_FILE 选项	337
APPEND_LOAD 选项	337
ASE_BINARY_DISPLAY 选项	338
ASE_FUNCTION_BEHAVIOR 选项	338
AUDITING 选项 [database]	339
BIT_VECTOR_PINNABLE_CACHE_PERCEN	
T 选项	340
BLOCKING 选项	340
BT_PREFETCH_MAX_MISS 选项	341
BT_PREFETCH_SIZE 选项	341
BTREE_PAGE_SPLIT_PAD_PERCENT 选项 ...	342
CACHE_PARTITIONS 选项	343
CHAINED 选项 [TSQL]	344
CHECKPOINT_TIME 选项	344
CIS_ROWSET_SIZE 选项	345

CLOSE_ON_ENDTRANS 选项 [TSQL]	345
CONTINUE_AFTER_RAISERROR 选项 [TSQL]	345
CONVERSION_ERROR 选项 [TSQL]	346
CONVERSION_MODE 选项	346
CONVERT_VARCHAR_TO_1242 选项	352
COOPERATIVE_COMMIT_TIMEOUT 选项	353
COOPERATIVE_COMMITS 选项	353
CURSOR_WINDOW_ROWS 选项	354
DATE_FIRST_DAY_OF_WEEK 选项	354
DATE_FORMAT 选项	355
DATE_ORDER 选项	357
DBCC_LOG_PROGRESS 选项	357
DBCC_PINNABLE_CACHE_PERCENT 选项 ..	358
DEBUG_MESSAGES 选项	358
DEDICATED_TASK 选项	359
DEFAULT_DBSPACE 选项	359
DEFAULT_DISK_STRIPING 选项	361
DEFAULT_HAVING_SELECTIVITY_PPM 选项	361
DEFAULT_ISQL_ENCODING 选项 [Interactive SQL]	362
DEFAULT_KB_PER_STRIPE 选项	363
DEFAULT_LIKE_MATCH_SELECTIVITY_PPM 选项	363
DEFAULT_LIKE_RANGE_SELECTIVITY_PPM 选项	364
DEFAULT_PROXY_TABLE_ROW_COUNT 选 项	365
DEFAULT_TABLE_UDF_ROW_COUNT Option	365
DELAYED_COMMIT_TIMEOUT 选项	365
DELAYED_COMMITS 选项	366
DISABLE_RI_CHECK 选项	366
DIVIDE_BY_ZERO_ERROR 选项 [TSQL]	366
DQP_ENABLED 选项	367

EARLY_PREDICATE_EXECUTION 选项	367
ENABLE_LOB_VARIABLES 选项	368
EXTENDED_JOIN_SYNTAX 选项	368
FORCE_DROP 选项	369
FORCE_NO_SCROLL_CURSORS 选项	369
FORCE_UPDATABLE_CURSORS 选项	370
FP_LOOKUP_SIZE 选项	370
FP_LOOKUP_SIZE_PPM 选项	371
FP_PREDICATE_WORKUNIT_PAGES 选项	372
FPL_EXPRESSION_MEMORY_KB 选项	372
GARRAY_FILL_FACTOR_PERCENT 选项	373
GARRAY_INSERT_PREFETCH_SIZE 选项	373
GARRAY_PAGE_SPLIT_PAD_PERCENT 选项	374
GARRAY_RO_PREFETCH_SIZE 选项	374
HASH_PINNABLE_CACHE_PERCENT 选项	375
HASH_THRASHING_PERCENT 选项	375
HG_DELETE_METHOD 选项	376
HG_SEARCH_RANGE 选项	377
HTTP_SESSION_TIMEOUT 选项	377
IDENTITY_ENFORCE_UNIQUENESS 选项	378
IDENTITY_INSERT 选项	378
INDEX_ADVISOR 选项	379
INDEX_ADVISOR_MAX_ROWS 选项	381
INDEX_PREFERENCE 选项	382
INFER_SUBQUERY_PREDICATES 选项	383
IN_SUBQUERY_PREFERENCE 选项	383
IQGOVERN_MAX_PRIORITY 选项	384
IQGOVERN_PRIORITY 选项	385
IQGOVERN_PRIORITY_TIME 选项	385
ISOLATION_LEVEL 选项	386
JAVA_LOCATION 选项	386
JAVA_VM_OPTIONS 选项	387
JOIN_EXPANSION_FACTOR 选项	387
JOIN_OPTIMIZATION 选项	388

JOIN_PREFERENCE 选项	389
JOIN_SIMPLIFICATION_THRESHOLD 选项 ...	391
LARGE_DOUBLES_ACCUMULATOR 选项	391
LF_BITMAP_CACHE_KB 选项	392
LOAD_ZEROLENGTH_ASNULL 选项	392
LOCKED 选项	393
LOG_CONNECT 选项	393
LOG_CURSOR_OPERATIONS 选项	394
LOGIN_MODE 选项	394
LOGIN_PROCEDURE 选项	395
MAIN_RESERVED_DBSPACE_MB 选项	395
MAX_CARTESIAN_RESULT 选项	396
MAX_CLIENT_NUMERIC_PRECISION 选项 ..	397
MAX_CLIENT_NUMERIC_SCALE 选项	397
MAX_CONNECTIONS 选项	398
MAX_CUBE_RESULT 选项	398
MAX_CURSOR_COUNT 选项	399
MAX_DAYS_SINCE_LOGIN 选项	399
MAX_FAILED_LOGIN_ATTEMPTS 选项	399
MAX_HASH_ROWS 选项	400
MAX_IQ_THREADS_PER_CONNECTION 选 项	400
MAX_IQ_THREADS_PER_TEAM 选项	401
MAX_JOIN_ENUMERATION 选项	401
MAX_PREFIX_PER_CONTAINS_PHRASE 选 项	402
MAX_QUERY_PARALLELISM 选项	402
MAX_QUERY_TIME 选项	403
MAX_STATEMENT_COUNT 选项	403
MAX_TEMP_SPACE_PER_CONNECTION 选 项	404
MAX_WARNINGS 选项	405
MINIMIZE_STORAGE 选项	405
MIN_PASSWORD_LENGTH 选项	406

MONITOR_OUTPUT_DIRECTORY 选项	407
MPX_AUTOEXCLUDE_TIMEOUT 选项	408
MPX_HEARTBEAT_FREQUENCY 选项	408
MPX_IDLE_CONNECTION_TIMEOUT 选项	408
MPX_MAX_CONNECTION_POOL_SIZE 选项	408
MPX_MAX_UNUSED_POOL_SIZE 选项	408
MPX_WORK_UNIT_TIMEOUT 选项	408
NEAREST_CENTURY 选项 [TSQL]	409
NOEXEC 选项	409
NON_ANSI_NULL_VARCHAR 选项	410
NON_KEYWORDS 选项 [TSQL]	410
NOTIFY_MODULUS 选项	411
ODBC_DISTINGUISH_CHAR_AND_VARCHA R 选项	411
ON_CHARSET_CONVERSION_FAILURE 选项	411
ON_ERROR 选项 [Interactive SQL]	412
ON_TSQL_ERROR 选项 [TSQL]	412
OS_FILE_CACHE_BUFFERING 选项	413
OS_FILE_CACHE_BUFFERING_TEMPDB 选 项	414
PASSWORD_EXPIRY_ON_NEXT_LOGIN 选项	415
PASSWORD_GRACE_TIME 选项	415
PASSWORD_LIFE_TIME 选项	416
POST_LOGIN_PROCEDURE 选项	416
PRECISION 选项	416
PREFETCH 选项	417
PREFETCH_BUFFER_LIMIT 选项	417
PREFETCH_BUFFER_PERCENT 选项	418
PREFETCH_GARRAY_PERCENT 选项	418
PREFETCH_SORT_PERCENT 选项	419

PRESERVE_SOURCE_FORMAT 选项	
[database]	419
QUERY_DETAIL 选项	420
QUERY_NAME 选项	421
QUERY_PLAN 选项	421
QUERY_PLAN_AFTER_RUN 选项	422
QUERY_PLAN_AS_HTML 选项	422
QUERY_PLAN_AS_HTML_DIRECTORY 选项	
.....	423
QUERY_PLAN_TEXT_ACCESS 选项	424
QUERY_PLAN_TEXT_CACHING 选项	425
QUERY_ROWS_RETURNED_LIMIT 选项	426
QUERY_TEMP_SPACE_LIMIT 选项	426
QUERY_TIMING 选项	427
QUOTED_IDENTIFIER 选项 [TSQL]	427
RECOVERY_TIME 选项	428
RESERVED_KEYWORDS 选项	428
RETURN_DATE_TIME_AS_STRING 选项	429
ROW_COUNT 选项	430
SCALE 选项	430
SIGNIFICANTDIGITSFORDOUBLEEQUALITY	
选项	431
SORT_COLLATION 选项	431
SORT_PINNABLE_CACHE_PERCENT 选项 ..	432
SQL_FLAGGER_ERROR_LEVEL 选项 [TSQL]	
.....	433
SQL_FLAGGER_WARNING_LEVEL 选项	
[TSQL]	434
STRING_RTRUNCATION 选项 [TSQL]	434
SUBQUERY_CACHING_PREFERENCE 选项	
.....	435
SUBQUERY_FLATTENING_PERCENT 选项 ..	436
SUBQUERY_FLATTENING_PREFERENCE 选	
项	437

SUBQUERY_PLACEMENT_PREFERENCE 选项	438
SUPPRESS_TDS_DEBUGGING 选项	438
SWEEPER_THREADS_PERCENT 选项	439
TDS_EMPTY_STRING_IS_NULL 选项 [database]	439
TEMP_EXTRACT_APPEND 选项	440
TEMP_EXTRACT_BINARY 选项	440
TEMP_EXTRACT_COLUMN_DELIMITER 选项	441
TEMP_EXTRACT_DIRECTORY 选项	442
TEMP_EXTRACT_ESCAPE_QUOTES 选项	443
TEMP_EXTRACT_NAME 选项	443
TEMP_EXTRACT_NULL_AS_EMPTY 选项	445
TEMP_EXTRACT_NULL_AS_ZERO 选项	446
TEMP_EXTRACT_QUOTE 选项	447
TEMP_EXTRACT_QUOTES 选项	447
TEMP_EXTRACT_QUOTES_ALL 选项	448
TEMP_EXTRACT_ROW_DELIMITER 选项	449
TEMP_EXTRACT_SIZE 选项	449
TEMP_EXTRACT_SWAP 选项	450
TEMP_RESERVED_DBSPACE_MB 选项	451
TEMP_SPACE_LIMIT_CHECK 选项	452
TEXT_DELETE_METHOD 选项	453
TIME_FORMAT 选项	453
TIMESTAMP_FORMAT 选项	454
TOP_NSORT_CUTOFF_PAGES 选项	455
TRIM_PARTIAL_MBC 选项	455
TSQL_VARIABLES 选项 [TSQL]	456
USER_RESOURCE_RESERVATION 选项	456
VERIFY_PASSWORD_FUNCTION 选项	457
WASH_AREA_BUFFERS_PERCENT 选项	458
WAIT_FOR_COMMIT 选项	458
WD_DELETE_METHOD 选项	459

索引	461
----------	-----

读者

本手册的目标读者是需要有关 Sybase IQ SQL 语句和数据库选项的参考材料的 Sybase® IQ 用户。

有关 Sybase IQ 其它方面的参考资料（包括语言元素、数据类型、函数、系统过程和系统表）可在《参考：构件块、表和过程》中找到。其它手册提供有关如何执行特定任务的详细信息。在本参考手册中可以查找可用的 SQL 语法、参数和选项等信息。有关命令行实用程序启动参数，请参见《实用程序指南》。

SQL 语句

说明 Sybase IQ 中可用的 SQL 语句，包括只能在嵌入式 SQL 或 Interactive SQL 中使用的某些语句。

SQL 语法中的常见元素

许多 SQL 语句的语法中出现的语言元素。

有关此处描述的元素的信息，请参见《参考：构件块、表和过程》>“SQL 语言元素”中的“标识符”、“搜索条件”、“表达式”和“字符串”。

- **column-name** - 表示列名的标识符。
- **condition** - 取值为 **TRUE**、**FALSE** 或 **UNKNOWN** 的表达式。
- **connection-name** - 表示活动连接名称的字符串。
- **data-type** - 存储数据类型。
- **expression** - 表达式。
- **filename** - 包含文件名的字符串。
- **host-variable** - C 语言变量，声明为前面有一个冒号的宿主变量。
- **indicator-variable** - 类型为 **short int** 的另一个宿主变量，紧跟在常规宿主变量之后。指示符变量的前面也必须有一个冒号。指示符变量用于向数据库传入和从数据库传出空值。
- **number** - 任何数字序列，后跟一个可选的小数部分，前面有一个可选的负号。数字后面还可以有一个“e”和一个指数。例如，

```
42
-4.038
.001
3.4e10
1e-10
```

- **owner** - 表示拥有数据库对象的用户 **ID** 的标识符。
- **role-name** - 表示外键角色名称的标识符。
- **savepoint-name** - 表示保存点名称的标识符。
- **search-condition** - 取值为 **TRUE**、**FALSE** 或 **UNKNOWN** 的条件。
- **special-value** - 《参考：构件块、表和过程》>“SQL 语言元素”>“特殊值”中描述的特殊值之一。
- **statement-label** - 表示循环或复合语句标签的标识符。
- **table-list** - 表名列表，可能包含相关名。有关详细信息，请参见《FROM 子句》。
- **table-name** - 表示表名的标识符。
- **userid** - 表示用户名的标识符。用户 **ID** 不区分大小写，并且不会受到数据库的 **CASE RESPECT** 属性设置的影响。

- `variable-name` - 表示变量名的标识符。

另请参见

- FROM 子句 (第 194 页)

语法规约定

SQL 语法说明中使用的约定。

- 关键字 - 所有 SQL 关键字都以大写字母显示；但 SQL 关键字是不区分大小写的，因此您可以键入大写或小写格式的关键字。例如，**SELECT** 与 **Select** 相同，而 **Select** 又与 **select** 相同。
- 占位符 - 必须替换为相应标识符或表达式的项会显示为斜体。
- 续行符 - 以省略号 (...) 开头的行是上一行的继续。

- 可选部分 - 语句的可选部分以中括号括起。例如：

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

此示例表明 *savepoint-name* 是可选的。不要键入中括号。

- 重复项 - 对于重复项列表，只显示列表中的一个元素，后跟省略号。允许指定一个或多个列表元素。如果指定了多个元素且指明须用逗号进行分隔，则必须用逗号将它们隔开。例如：

```
UNIQUE ( column-name [ , ... ] )
```

该示例表明您可以多次指定 *column-name*，中间用逗号隔开。不要键入中括号。

- 备选选项 - 如果必须选择一个选项，则将备选选项放在大括号内。例如：

```
[ QUOTES { ON | OFF } ]
```

该示例表明，如果选择 **QUOTES** 选项，则必须提供 **ON** 或 **OFF** 之一。不要输入大括号。

- 一个或多个选项 - 如果选择多个选项，请用逗号分隔所选选项。例如：

```
{ CONNECT, DBA, RESOURCE }
```

语句适用性指示符

某些语句标题的后面有一个指示符（括在中括号内），用于指示语句的使用环境。

这些指示符如下：

- [ESQL] - 表示语句用于嵌入式 SQL 中。
- [Interactive SQL] - 表示语句仅用于 Interactive SQL (**dbisql**) 中。
- [SP] - 表示语句用于存储过程或批处理中。

- [T-SQL] - 表示实现该语句的目的是为了与 Adaptive Server® Enterprise 兼容。在有些情况下，该语句不能用于非 Transact-SQL 格式的存储过程中。在其它情况下，如果不存在 Transact-SQL 兼容性问题，则可以使用一个与建议采用的 ISO/ANSI SQL 标准更接近的替代语句。

如果使用两组括号，表示语句可用于两种环境。例如，[ESQL][SP] 表示语句既可用于嵌入式 SQL，也可以用于存储过程。

ALLOCATE DESCRIPTOR 语句 [ESQL]

为 SQL 描述符区 (SQLDA) 分配空间。

语法

```
ALLOCATE DESCRIPTOR descriptor-name
... [ WITH MAX { integer | host-variable } ]
```

参数

- **descriptor-name:** - *string*
有关详细信息，请参见《参考：构件块、表和过程》>“SQL 语言元素”。

示例

- **示例 1** - 下面的示例程序包括 **ALLOCATE DESCRIPTOR** 语句用法的一个示例。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

EXEC SQL INCLUDE SQLCA;

#include <sqldef.h>

EXEC SQL BEGIN DECLARE SECTION;
int      x;
short    type;
int      numcols;
char      string[100];
a_sql_statement_number stmt = 0;
EXEC SQL END DECLARE SECTION;

int main(int argc, char * argv[])
{
    struct sqllda *      sqllda1;

    if( !db_init( &sqlca ) ) {
```

```

        return 1;
    }
    db_string_connect(&sqlca, "UID=dba;PWD=sql;DBF=d:\\IQ-15_3\\
\\sample.db");

    EXEC SQL ALLOCATE DESCRIPTOR sqllda1 WITH MAX 25;

    EXEC SQL PREPARE :stmt FROM
        'select * from Employees';
    EXEC SQL DECLARE curs CURSOR FOR :stmt;
    EXEC SQL OPEN curs;

    EXEC SQL DESCRIBE :stmt into sqllda1;
    EXEC SQL GET DESCRIPTOR sqllda1 :numcols=COUNT;
        // how many columns?
    if( numcols > 25 ) {
        // reallocate if necessary
        EXEC SQL DEALLOCATE DESCRIPTOR sqllda1;
        EXEC SQL ALLOCATE DESCRIPTOR sqllda1
            WITH MAX :numcols;
    }
    type = DT_STRING;    // change the type to string
    EXEC SQL SET DESCRIPTOR sqllda1 VALUE 2 TYPE = :type;
    fill_sqllda( sqllda1 ); // allocate space for the variables

    EXEC SQL FETCH ABSOLUTE 1 curs USING DESCRIPTOR sqllda1;
    EXEC SQL GET DESCRIPTOR sqllda1 VALUE 2 :string = DATA;

    printf("name = %s", string );

    EXEC SQL DEALLOCATE DESCRIPTOR sqllda1;
    EXEC SQL CLOSE curs;
    EXEC SQL DROP STATEMENT :stmt;

    db_string_disconnect( &sqlca, " " );
    db_fini( &sqlca );

    return 0;
}

```

用法

使用此语句前，必须在 C 代码中声明以下内容：

```
struct sqllda * descriptor_name
```

您可以通过 **WITH MAX** 子句来指定描述符区内的变量数。缺省大小为 1。

在执行提取操作或任何访问描述符区内数据的语句之前，仍必须调用 **fill_sqllda** 为实际的数据项分配空间。

请参见“SQL Anywhere Server - 编程” > “SQL Anywhere 嵌入式 SQL” > “SQL 描述符区域 (SQLDA)”。

注意： 此参考指向 SQL Anywhere 文档。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 受 Open Client/Open Server 支持。

另请参见

- DEALLOCATE DESCRIPTOR 语句 [ESQL]（第 152 页）

ALTER DATABASE 语句

升级使用该软件以前版本创建的数据库，或者添加/删除 jConnect™ for JDBC™ 支持。请使用 Interactive SQL 运行该语句。

语法

```
ALTER DATABASE UPGRADE
    [ PROCEDURE ON ]
    [ JCONNECT { ON | OFF } ]
```

示例

- **示例 1** - 以下示例禁用 jConnect 支持：

```
ALTER DATABASE UPGRADE JCONNECT OFF
```

用法

ALTER DATABASE 语句升级使用该软件早期版本创建的数据库。它适用于维护版本和主版本。

升级数据库时，Sybase IQ 会做出以下更改：

- 将系统表升级到当前版本。
- 添加任何新的数据库选项。
- 启用当前版本中的新功能。

如果数据库是用该软件的当前版本创建的，也可以使用 **ALTER DATABASE UPGRADE** 只添加 jConnect 功能。

注意：

- 在升级之前，请参见《安装和配置指南》以了解备份建议。
- 在您运行 **ALTER DATABASE UPGRADE** 之前，请务必以限制用户连接的方式来启动服务器。有关说明及其它升级问题，请参见针对所用平台的《安装和配置指南》>“数据库升级”。
- 使用 **iqunload** 实用程序升级在 15.0 之前的版本中创建的数据库。请参见针对所用平台的《安装和配置指南》>“数据库升级”。

使用 **ALTER DATABASE UPGRADE** 之后，请关闭数据库。

- **PROCEDURE** 子句 - 在数据库中删除并重新创建所有 **dbo** 和 **sys** 拥有的过程。
- **JCONNECT** 子句 - 要使 Sybase jConnect JDBC 驱动程序能够访问系统目录信息，必须指定 **JCONNECT ON**。这会安装 jConnect 系统表和过程。若要排除 jConnect 系统对象，请指定 **JCONNECT OFF**。只要不访问系统目录信息，仍可以使用 JDBC。缺省设置为包括 jConnect 支持 (**JCONNECT ON**)。

副作用：

- 自动提交

标准

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

权限

必须具有 **DBA** 权限。

另请参见

- **CREATE DATABASE** 语句（第 68 页）

ALTER DBSPACE 语句

更改读/写模式、更改大小或扩展现有数据库空间。

语法

```
ALTER DBSPACE dbspace-name
{ ADD new-file-spec [, new-file-spec ... ]
| DROP FILE logical-file-name [, FILE logical-file-name ... ]
| RENAME TO newname | RENAME 'new-file-pathname'
| READONLY | READWRITE
| ONLINE | OFFLINE
| STRIPING{ ON | OFF }
| STRIPESIZEKB size-in-KB
ALTER FILE file-name
{ READONLY | [ FORCE ] READWRITE }
| SIZE file-size [ KB | MB | GB | TB ]
| ADD file-size [ KB | MB | GB | TB | PAGES ] }
RENAME PATH 'new-file-pathname'
RENAME TO newname
```


参数

- **new-file-spec:** -

```
FILE logical-file-name
'file-path' iq-file-opts
```

- **iq-file-opts:** -

```
[ [ SIZE ] file-size ]
...[ KB | MB | GB | TB ] ]
[ RESERVE reserve-size [ KB | MB | GB | TB ] ]
```

示例

- **示例 1** - 将名为 DspHist 的数据库空间的模式更改为 **READONLY**。

```
ALTER DBSPACE DspHist READONLY
```

- **示例 2** - 通过添加大小为 500MB 的文件 FileHist3, 向数据库空间 DspHist 添加 500MB。

```
ALTER DBSPACE DspHist
ALTER FILE FileHist3 ADD 500MB
```

- **示例 3** - 在 UNIX 系统上, 向数据库空间 DspHist 添加两个 500MB 文件。

```
ALTER DBSPACE DspHist ADD
FILE FileHist3 '/History1/data/file3' SIZE 500MB,
FILE FileHist4 '/History1/data/file4' SIZE 500
```

- **示例 4** - 将数据库空间 IQ_SYSTEM_TEMP 的大小增大 2GB。

```
ALTER DBSPACE IQ_SYSTEM_TEMP ADD 2 GB
```

- **示例 5** - 从数据库空间 DspHist 中删除两个文件。两个文件必须为空。

```
ALTER DBSPACE DspHist
DROP FILE FileHist2, FILE FileHist4
```

- **示例 6** - 将数据库空间 IQ_SYSTEM_MAIN 的大小增大 1000 页。(ADD 的缺省单位为页。)

```
ALTER DBSPACE IQ_SYSTEM_MAIN ADD 1000
```

用法

ALTER DBSPACE 更改读写模式, 更改联机/脱机状态, 更改文件大小, 重命名数据库空间名称、文件逻辑名或文件路径, 或设置数据库空间条带化参数。有关现有数据库空间的详细信息, 请运行 **sp_iqdbspace** 过程、**sp_iqdbspaceinfo** 过程、**sp_iqfile** 过程、**sp_iqdbspaceobjectinfo** 和 **sp_iqobjectinfo**。数据库空间和 dbfile 名称始终不区分大小写。如果数据库为 **CASE RESPECT**, 且操作系统支持区分大小写的文件, 则物理文件路径区分大小写。否则, 文件路径不区分大小写。

对于数据库空间名称和 dbfile 名称, 要么不用引号引起来, 要么用双引号引起来。请将 dbfile 的物理文件路径用单引号引起来。

在 Windows 中，如果您指定一个路径，则任何后跟 **n** 或 **x** 的反斜杠字符 (\) 都必须双写。这样可防止根据 SQL 中的字符串规则将这些反斜杠解释为换行符 (\n) 或十六进制数 (\x)。更安全的做法是始终使用两个反斜杠。

请参见《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqdbspace 过程”和《系统管理指南第一卷》>“数据库对象管理”>“数据存储”。

ADD FILE 子句 - 向指定 **dbspace** 中添加一个或多个文件。必须为每个文件指定 **dbfile** 名称和物理文件路径，且该名称和路径必须唯一。您可以向 **IQ** 主数据库空间、**IQ** 共享临时数据库空间或 **IQ** 临时数据库空间添加文件。您可以向只读数据库空间中添加一个文件，但数据库空间仍为只读。您只能以只读模式 (**ADD FILE** 的缺省设置) 向 **Multiplex** 共享临时 **dbspace** 添加文件。请参见《使用 Sybase IQ Multiplex》。

一个目录数据库空间只能包含一个文件，因此 **ADD FILE** 不可用于目录数据库空间。

DROP FILE 子句 - 从 **IQ** 数据库空间删除指定文件。该文件必须为空。您不能删除指定数据库空间中的最后一个文件。如果数据库空间中仅包含一个文件，则应该使用 **DROP DBSPACE**。**RENAME TO** 子句 - 将 *dbspace-name* 重命名为新名称。新名称必须在数据库中唯一。不能重命名 **IQ_SYSTEM_MAIN**、**IQ_SYSTEM_MSG**、**IQ_SYSTEM_TEMP**、**IQ_SHARED_TEMP** 或 **SYSTEM**。

RENAME 子句 - 重命名包含单个文件的数据库空间的路径名。从语义上看，它等同于 **ALTER FILE RENAME PATH** 子句。如果数据库空间包含多个文件，则系统会返回错误。

READONLY 子句 - 将 **IQ_SYSTEM_MAIN**、**IQ_SYSTEM_TEMP**、**IQ_SYSTEM_MSG**、**IQ_SHARED_TEMP** 和 **SYSTEM** 之外的任何数据库空间更改为只读。禁止对当前已分配至数据库空间的任何对象执行 **DML** 修改。只能用于 **IQ** 主存储区中的数据库空间。

READWRITE 子句 - 将数据库空间更改为读写。数据库空间必须联机。只能用于 **IQ main** 存储中的数据库空间。

ONLINE 子句 - 将脱机数据库空间和所有关联文件设置为联机状态。只能用于 **IQ main** 存储中的数据库空间。

OFFLINE 子句 - 将联机只读数据库空间和所有关联文件设置为脱机状态。（如果数据库空间为读写、已脱机或非 **IQ** 主存储区，则系统会返回错误。）只能用于 **IQ main** 存储中的数据库空间。

STRIPING 子句 - 按照指定条件更改数据库空间的磁盘条带化。当磁盘条带化被设置为 **ON** 时，会以循环分区方式从数据库空间内的每个文件分配数据。例如，所写的第一个数据库页面将进入第一个文件，所写的第二个页面将进入给定数据库空间内的下一个文件，以此类推。将跳过只读数据库空间。

STRIPESIZEKB 子句 - 指定磁盘条带化算法移至指定数据库空间的下一条带之前要写入每个文件的千字节数 (**KB**)。

ALTER FILE READONLY - 将指定文件更改为只读。文件必须与 **IQ** 主数据库空间关联。不能将 **IQ_SHARED_TEMP** 中的文件更改为 **READONLY** 状态。

ALTER FILE READWRITE - 将指定的 IQ 主存储区 **dbfile** 或临时存储 **dbfile** 更改为读写。文件必须与 IQ main 数据库空间或临时数据库空间关联。

ALTER FILE FORCE READWRITE - 将指定的共享临时存储 **dbfile** 的状态更改为读写，尽管辅助节点可能会出现已知文件状态问题。该文件可能与 IQ 主数据库空间、共享临时数据库空间或临时数据库空间相关联，但因为 **IQ_SYSTEM_MAIN** 中的新 **dbfile** 和用户主数据库空间创建为读写，所以此子句只会影响共享临时数据库空间。

ALTER FILE SIZE 子句 - 以千字节 (KB)、兆字节 (MB)、千兆字节 (GB) 或千吉字节 (TB) 为单位指定文件的新大小。缺省单位为兆字节。您只能在空闲列表（一种分配映射）有足够空间且数据库空间有足够保留的空间时增大数据库空间。减小数据库空间大小也只能在将要截断的部分未被使用的情况下才能进行。

ALTER FILE ADD 子句 - 以页、千字节 (KB)、兆字节 (MB)、千兆字节 (GB) 或千吉字节 (TB) 为单位扩展文件的大小。缺省值为 MB。您只能在空闲列表（一种分配映射）有足够空间且数据库空间有足够保留的空间时执行 **ADD** 子句。

您还可以通过 Sybase Central Dbspaces 窗口来查看和更改数据库空间的模式和大小。

ALTER FILE RENAME PATH 子句 - 重命名与指定文件关联的文件路径名。此子句只是将文件与新文件路径关联起来，而不再关联旧路径。子句并不实际更改操作系统文件名。必须通过操作系统才能更改文件名。数据库空间必须脱机才能重命名文件路径。当数据库空间更改为联机状态或当数据库重新启动时，将使用新路径。

您不能重命名 **IQ_SYSTEM_MAIN** 中的文件的路径，因为如果新路径不可访问，则数据库将无法启动。如果您需要重命名 **IQ_SYSTEM_MAIN** 中的文件的路径，请将文件设置为只读，清空并删除该文件，然后以新文件路径名再次添加该文件。

ALTER FILE RENAME TO 子句 - 将指定文件的逻辑名重命名为新名称。新名称必须在数据库中唯一。

副作用：

- 自动提交
- 自动检查点
- 如果将模式更改为 **READONLY**，则数据库空间中的内部数据库结构会立即重新分配到某个读写数据库空间。

标准

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

权限

必须具有 **SPACE ADMIN** 或 **DBA** 权限。

另请参见

- **CREATE DATABASE** 语句（第 68 页）

- CREATE DBSPACE 语句（第 77 页）
- DROP 语句（第 170 页）

ALTER DOMAIN 语句

重命名用户定义的域或数据类型。不重命名 Java 类型。

语法

```
ALTER { DOMAIN | DATATYPE } user-type  
RENAME new-name
```

参数

- **new-name:** - 表示新域名的标识符。
- **user-type:** - 要重命名的域的用户定义数据类型。

示例

- **示例 1** - 将 Address 域重命名为 MailingAddress:

```
ALTER DOMAIN Address RENAME MailingAddress
```

用法

ALTER DOMAIN 语句更新 SYSUSERTYPE 系统表中用户定义的域或数据类型的名称。请参见《参考：构件块、表和过程》>“系统表和视图”>“系统视图”>“SYSUSERTYPE 系统视图”。

您必须重新创建所有引用该用户定义的域或数据类型的过程、视图或事件，否则它们将继续引用原来的名称。

另请参见《参考：构件块、表和过程》>“SQL 数据类型”。

副作用：

- 自动提交

权限

必须具有 DBA 权限或者是创建域的数据库用户。

另请参见

- CREATE DOMAIN 语句（第 80 页）

ALTER EVENT 语句

更改事件的定义或者与之关联的、自动执行预定义操作的处理程序。还可以变更调度操作的定义。

语法

```
ALTER EVENT event-name
[ DELETE TYPE | TYPE event-type ]
{
    WHERE { trigger-condition | NULL }
    | { ADD | [ MODIFY ] | DELETE } SCHEDULE schedule-spec
}
[ ENABLE | DISABLE ]
[ [ MODIFY ] HANDLER compound-statement | DELETE HANDLER ]
```

参数

- event-type:** - BackupEnd | “Connect” | ConnectFailed | DatabaseStart | DBDiskSpace | “Disconnect” | GlobalAutoincrement | GrowDB | GrowLog | GrowTemp | LogDiskSpace | “RAISERROR” | ServerIdle | TempDiskSpace
- trigger-condition:** - [event_condition(*condition-name*) { = | < | > | != | <= | >= } *value*]
- schedule-spec:** - [*schedule-name*] { STARTTIME *start-time* | BETWEEN *start-time* AND *end-time* } [EVERY *period* { HOURS | MINUTES | SECONDS }] [ON { (*day-of-week*, ...) | (*day-of-month*, ...) }] [STARTDATE *start-date*]
- event-name | schedule-name:** - *identifier*
- day-of-week:** - *string*
- value | period | day-of-month:** - *integer*
- start-time | end-time:** - *time*
- start-date:** - *date*

用法

通过 **ALTER EVENT**，您可以更改使用 **CREATE EVENT** 创建的事件定义。可能的用法包括：

- 在开发期间更改事件处理程序。
- 在开发阶段，定义和测试没有触发器状态或日程表的事件处理程序，然后在事件处理程序完成后，使用 **ALTER EVENT** 添加执行条件。
- 通过禁用事件临时禁用事件处理程序。

使用 **ALTER EVENT** 更改事件时，请指定事件名称和（可选）日程表名称。

通过查询系统表 SYSEVENT 列出事件名称。例如：

```
SELECT event_id, event_name FROM SYS.SYSEVENT
```

SQL 语句

通过查询系统表 SYSSCHEDULE 列出日程表名称。例如：

```
SELECT event_id, sched_name FROM SYS.SYSSCHEDULE
```

每个事件都具有唯一的事件 ID。可使用 SYSEVENT 和 SYSSCHEDULE 的 event_id 列将事件与关联的日程表匹配。

DELETE TYPE 子句 - 删除事件与事件类型的关联。

ADD|MODIFY|DELETE SCHEDULE 子句 - 更改日程表的定义。在任一 ALTER EVENT 语句中，只能更改一个日程表。

WHERE 子句 - WHERE NULL 选项删除条件。

有关大多数参数的说明，请参见“CREATE EVENT 语句”。

另请参见《系统管理指南第二卷》>“使用日程表和事件自动完成任务”。

副作用：

- 自动提交

权限

必须具有 DBA 权限。

另请参见

- BEGIN……END 语句（第 49 页）
- CREATE EVENT 语句（第 82 页）

ALTER FUNCTION 语句

修改现有函数。在 ALTER FUNCTION 语句中包括修改后的整个函数。

语法

语法 1

```
ALTER FUNCTION [ owner.]function-name function-definition
```

```
function-definition : CREATE FUNCTION syntax
```

语法 2

```
ALTER FUNCTION [ owner.]function-name
```

```
SET HIDDEN
```

语法 3

```
ALTER FUNCTION [ owner.]function-name
```

```
RECOMPILE
```

用法

语法 1 - 在语法上与 **CREATE FUNCTION** 语句相同，只是第一个字有所不同。任意版本的 **CREATE FUNCTION** 语句均可更改。

系统将保留函数的现有权限，因此不必重新分配。如果执行了 **DROP FUNCTION** 和 **CREATE FUNCTION**，则必须重新分配执行权限。

语法 2 - 使用 **SET HIDDEN** 搅乱关联函数的定义，并使其不可读取。函数必须卸载并重新装载到其它数据库中。

警告！ **SET HIDDEN** 设置是不可逆的。如果您再次需要原始源，您必须在数据库外加以维护。

如果您使用 **SET HIDDEN**，则使用存储过程调试工具进行调试时，系统不会显示函数定义，在过程分析中也看不到函数定义。

另请参见《系统管理指南第二卷》>“使用过程和批处理”>“隐藏过程、函数和视图的内容”。

语法 3 - 使用 **RECOMPILE** 重新编译用户定义的函数。当您重新编译函数时，将会重新分析目录中存储的定义并验证语法。重新编译不会更改函数的保留源。当您重新编译函数时，**SET HIDDEN** 子句搅乱的定义仍保持混乱状态且不可读取。

副作用：

- 自动提交

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。

权限

必须是函数的所有者或者具有 DBA 权限。

另请参见

- ALTER PROCEDURE 语句（第 25 页）
- CREATE FUNCTION 语句（第 90 页）
- DROP 语句（第 170 页）

ALTER INDEX 语句

重命名基表或全局临时表中的索引以及用户显式创建的索引和外键的外键角色名。

语法

```
ALTER { INDEX index-name
| [INDEX] FOREIGN KEY role-name
| [INDEX] PRIMARY KEY
| ON [owner.]table-name { rename-clause | move-clause }
```

参数

- **rename-clause** - : **RENAME TO** | **AS** *new-name*
- **move-clause:** - **MOVE TO** *dbspace-name*

示例

- **示例 1** - 将主键 (c5 的 HG) 从数据库空间 Dsp4 移动到 Dsp8:

```
CREATE TABLE foo (
    c1 INT IN Dsp1,
    c2 VARCHAR(20),
    c3 CLOB IN Dsp2,
    c4 DATE,
    c5 BIGINT,
    PRIMARY KEY (c5) IN Dsp4) IN Dsp3);

CREATE DATE INDEX c4_date ON foo(c4) IN Dsp5;

ALTER INDEX PRIMARY KEY ON foo MOVE TO Dsp8;
```

- **示例 2** - 将 **DATE** 索引从 Dsp5 移动到 Dsp9:

```
ALTER INDEX c4_date ON foo MOVE TO Dsp9
```

- **示例 3** - 将表 `jal.mytable` 中的索引 `COL1_HG_OLD` 重命名为 `COL1_HG_NEW`:

```
ALTER INDEX COL1_HG_OLD ON jal.mytable
RENAME AS COL1_HG_NEW
```

- **示例 4** - 将表 `dba.Employees` 中的外键角色名称 `ky_dept_id` 重命名为 `emp_dept_id`:

```
ALTER INDEX FOREIGN KEY ky_dept_id
ON dba.Employees
RENAME TO emp_dept_id
```

用法

ALTER INDEX 语句重命名索引以及用户显式创建的索引和外键的外键角色名称。只能重命名基表或全局临时表中的索引。不能重命名用于强制执行键约束的索引。

ON 子句 - **ON** 子句指定包含要重命名的索引或外键的表的名称。

RENAME [**AS** | **TO**] 子句 - **RENAME** 子句指定索引或外键角色的新名称。

MOVE 子句 - **MOVE** 子句将指定的索引、唯一约束、外键或主键移动到指定的数据库空间。对于唯一约束或外键，您必须指定其唯一索引名称。

您必须对新数据库空间具有 **CREATE** 特权并且是表所有者，或者具有 **DBA** 或 **SPACE ADMIN** 权限。

注意： 如果尝试更改本地临时表中的索引，则系统会返回错误“未找到索引”。如果尝试更改非用户创建的索引，例如缺省索引 (**FP**)，则系统会返回错误“无法更改索引。只能更改所有者类型为 **USER** 的基表或全局临时表中的索引。”

副作用：

- 自动提交。在 **Interactive SQL** 的“结果”窗格中清除“结果”选项卡。关闭当前连接的所有游标。

标准

- **SQL** - 符合 **ISO/ANSI SQL** 标准。
- **Sybase** - 不受 **Adaptive Server Enterprise** 支持。

权限

必须是表所有者、对表具有 **REFERENCES** 权限，或者具有 **DBA** 或 **RESOURCE** 权限。对于 **ALTER INDEX MOVE TO** 语句，您必须对新数据库空间具有 **CREATE** 特权并且是表所有者，或者具有 **DBA** 或 **SPACE ADMIN** 权限。

另请参见

- **ALTER TABLE** 语句 (第 31 页)
- **CREATE INDEX** 语句 (第 97 页)
- **CREATE TABLE** 语句 (第 136 页)

ALTER LOGICAL SERVER 语句

修改数据库中现有的用户定义的逻辑服务器配置。

语法

```
ALTER LOGICAL SERVER logical-server-name
{ alter_clause }
```

参数

- **alter_clause:** -

```
{ ADD MEMBERSHIP '(' { ls-member, ... } ')'
  DROP MEMBERSHIP '(' { ls-member, ... } ')'
}
```

- **ls-member:** -

```
FOR LOGICAL COORDINATOR
| mpx-server-name
```

示例

- **示例 1** - 以下示例通过将 Multiplex 节点 n1 和 n2 添加至逻辑服务器 ls1，更改用户定义的逻辑服务器：

```
ALTER LOGICAL SERVER ls1 ADD MEMBERSHIP (n1, n2)
```

- **示例 2** - 以下示例添加 COORDINATOR 成员资格并将当前协调节点 n1 的成员资格从逻辑服务器 ls1 删除：

```
ALTER LOGICAL SERVER ls1 ADD MEMBERSHIP (FOR LOGICAL COORDINATOR)
ALTER LOGICAL SERVER ls1 DROP MEMBERSHIP (n1)
```

用法

仅适用于 Multiplex。

logical-server-name 指现有用户定义的逻辑服务器名，换言之，它不能是内置或保留逻辑服务器名。**SYS.ISYSIQLSMEMBER** 系统表存储逻辑服务器成员资格的定义。

成员节点添加至某一逻辑服务器或从中移除后，仅当该节点运行与 **ALTER LOGICAL SERVER** 关联的 TLV 日志后，才在启动或关闭时接受逻辑服务器连接。从逻辑服务器删除某一节点后，该节点现有逻辑服务器连接仍在运行，不过这些连接的分布式处理已停止。

发生如下情况时系统返回错误：

- 任意 *ls-member*（通过 **ADD MEMBERSHIP** 子句指定）已是逻辑服务器的成员。
- 任意 *ls-member*（通过 **DROP MEMBERSHIP** 子句指定）不是逻辑服务器的现有成员。
- 逻辑服务器成员资格更改导致成员资格重叠检查失败。

权限

必须具有 DBA 或 MPX ADMIN 权限。

ALTER LOGIN POLICY 语句

此语句执行两个功能。可修改数据库中现有登录策略的选项值，也可配置逻辑服务器访问。

语法

```
ALTER LOGIN POLICY policy-name {alter-clause}
```

参数

- **alter-clause:** -


```
{ { ADD | DROP | SET } LOGICAL SERVER
    ls-assignment-list
    | policy-option-name = policy-option-value [ LOGICAL SERVER
    ls-override-list ]
  }
```
- **ls-assignment-list:** -


```
{ { ls-name, ... } | SERVER | NONE | DEFAULT }
```
- **ls-override-list:** -


```
{ ls-name, ... }
```
- **ls-name:** -


```
{ OPEN | user-defined-ls-name }
```
- **policy-option-value:** -


```
{ UNLIMITED | DEFAULT | value }
```

示例

- **示例 1** – 请参见“逻辑服务器访问配置”和“登录策略选项配置”。

用法

仅适用于 Multiplex。

请参见“逻辑服务器访问配置”和“登录策略选项配置”。

权限

必须具有 DBA 或 USER ADMIN 权限。

逻辑服务器访问配置

可使用 `ALTER LOGIN POLICY` 配置逻辑服务器访问。

示例 1

假定 `root` 登录策略允许访问逻辑服务器 `ls4`、`ls5` 且登录策略 `lp1` 不包含逻辑服务器分配。如下语句可将登录策略 `lp1` 有效分配至逻辑服务器 `ls4`、`ls5`。

将逻辑服务器 `ls1` 分配给逻辑策略 `lp1`。

```
ALTER LOGIN POLICY lp1 ADD LOGICAL SERVER ls1
```

示例 2

本语句允许逻辑服务器 `ls2` 和 `ls3` 接受来自登录策略 `lp1` 的访问。

```
ALTER LOGIN POLICY lp1 ADD LOGICAL SERVER ls2, ls3
```

示例 3

修改登录策略 `lp1`，仅允许访问 `ls3` 和 `ls4`。

```
ALTER LOGIN POLICY lp1 ADD LOGICAL SERVER ls4
```

```
ALTER LOGIN POLICY lp1 DROP LOGICAL SERVER ls1, ls2
```

或者

```
ALTER LOGIN POLICY lp1 SET LOGICAL SERVER ls3, ls4
```

示例 4

修改登录策略 `lp1`，拒绝访问任意逻辑服务器。

```
ALTER LOGIN POLICY lp1 SET LOGICAL SERVER NONE
```

示例 5

删除登录策略 `lp1` 的当前逻辑服务器分配，允许其继承 `root` 登录策略的逻辑服务器分配。

```
ALTER LOGIN POLICY lp1 SET LOGICAL SERVER DEFAULT
```

用法

仅适用于 `Multiplex`。

ADD、**DROP** 或 **SET** 子句可用来配置某一登录策略的逻辑服务器分配：

- **ADD** - 向某一登录策略添加新逻辑服务器分配
- **DROP** - 从某一登录策略删除现有逻辑服务器分配
- **SET** - 使用一组新的逻辑服务器为某一登录策略替代所有逻辑服务器分配

仅使用一个 **ADD**、**DROP** 或 **SET** 子句。**SERVER**、**NONE** 和 **DEFAULT** 仅与 **SET** 子句搭配使用。对于每个 **ls-assignment** 列表或 **ls-override** 列表，仅指定特定逻辑服务器名一次。

发生下列情况则会返回错误：

- 已有指定 **ADD** 子句的逻辑服务器分配给了登录策略。
- 当前未有指定 **DROP** 子句的逻辑服务器分配给了登录策略。
- 逻辑服务器分配更改可能造成已分配逻辑服务器间成员资格重叠。

`SYS.ISYSIQLOGINPOLICYLSINFO` 存储逻辑服务器分配信息。对登录策略中登录策略选项的每次逻辑服务器覆盖，`ISYSIQLOGINPOLICYLSINFO` 中均有对应行。

登录策略选项配置

可使用 `ALTER LOGIN POLICY` 配置登录策略访问。

示例

以下示例覆盖两处逻辑服务器的登录策略设置，在逻辑服务器 `ls1` 启用分布式查询处理并增加逻辑服务器 `ls2` 的最大连接数：

```
ALTER LOGIN POLICY lp1 dqp_enabled=ON LOGICAL SERVER ls1;  
ALTER LOGIN POLICY lp2 max_connections=20 LOGICAL SERVER ls2;
```

用法

仅适用于 **Multiplex**。

逻辑服务器级覆盖意味着，某一特定登录策略选项对于不同逻辑服务器有着不同设置。`SYS.ISYSIQLSLOGINPOLICYOPTION` 为逻辑服务器覆盖春村登录策略选项值。对登录策略中登录策略选项的每次逻辑服务器覆盖，`ISYSIQLSLOGINPOLICYOPTION` 中均有对应行。

注意： 仅可以为登录策略选项 `max_connections`、`dqp_enabled` 指定逻辑服务器覆盖设置。（`dqp_enabled` 选项仅影响 **Multiplex** 服务器。请参见“**Multiplex 引用**”>“数据库选项”>“`dqp_enabled` 选项”。）无法为 `root` 登录策略指定逻辑服务器覆盖。

表 1. 登录策略选项

选项	描述	值	ROOT 策略的初始值	适用于
<code>dqp_enabled</code>	如为 ON，可为分配了该登录策略的用户启用分布式查询处理。	ON、OFF	ON	仅用于 Multiplex 服务器。所有用户（包括那些具有 DBA 权限的用户）...
<code>locked</code>	如果此选项的值为 ON，则会禁止用户创建新连接	ON、OFF	OFF	仅限没有 DBA 权限的用户

选项	描述	值	ROOT 策略的初始值	适用于
max_connections	用户可进行的最大并发连接数。	0 – 2147483647	无限制	仅限没有 DBA 权限的用户
max_days_since_login	同一用户进行的两次连续登录之间可经过的最大天数。	0 – 2147483647	无限制	仅限没有 DBA 权限的用户
max_failed_login_attempts	自上一次成功登录尝试后，帐户锁定前可登录到用户帐户的最大失败尝试次数。	0 – 2147483647	无限制	仅限没有 DBA 权限的用户
max_non_dba_connections	没有 DBA 权限的用户可以进行的最大并发连接数。只有 root 登录策略支持此选项。	0 – 2147483647	无限制	仅限没有 DBA 权限的用户。仅适用于 root 登录策略。
password_expiry_on_next_login	如果此选项的值为 ON，则用户的口令将会在下次登录时过期。	ON、OFF	OFF	所有用户（包括那些具有 DBA 权限的用户）
password_grace_time	在口令到期前剩余的天数，在此期间允许登录，但缺省 post_login 过程将发出警告。	0 – 2147483647	0	所有用户（包括那些具有 DBA 权限的用户）
password_life_time	必须更改口令前的最大天数。	0 – 2147483647	无限制	所有用户（包括那些具有 DBA 权限的用户）

注意： Sybase IQ 不再支持 **MULTIPLEX SERVER** 覆盖子句。为登录策略选项使用 **ALTER LOGIN POLICY** 语句搭配 **MULTIPLEX SERVER** 覆盖子句，系统将返回错误。

ALTER LS POLICY 语句

修改数据库中现有 root 逻辑服务器策略的某些或所有选项值。

语法

```
ALTER LS POLICY
policy-name
```

```
option-value-list
```

参数

- **option-value-list:** -

```
{option-name=value}
```

示例

- **示例 1** - 以下示例更改逻辑服务器策略:

```
ALTER LS POLICY root
ALLOW_COORDINATOR_AS_MEMBER=ON;
```

注意: ALLOW_COORDINATOR_AS_MEMBER 是逻辑服务器策略的唯一选项, root 是唯一的逻辑服务器策略。无法创建逻辑服务器策略。

用法

仅适用于 Multiplex。

权限

必须具有 DBA 或 MPX ADMIN 权限。

ALTER MULTIPLEX RENAME 语句

重命名 Multiplex 并将 Multiplex 名称存储在 SYS.ISYSIQINFO 系统表中。

语法

```
ALTER MULTIPLEX RENAME multiplex-name
```

用法

仅适用于 Multiplex。

创建 Multiplex 后将以协调器命名。Multiplex 名称只能用在 Sybase Central 中标识 Multiplexes 文件夹中的 Multiplex。此语句将自动提交。

权限

必须具有 DBA 或 MULTIPLEX ADMIN 权限。

ALTER MULTIPLEX SERVER 语句

更改给定服务器的名称、Catalog 文件路径、角色或状态。

语法

语法 1:

```
ALTER MULTIPLEX SERVER server-name server-option
```

语法 2:

```
ALTER MULTIPLEX SERVER PRIVATE NULL
```

参数

- **server-option:** -

```
{ RENAME new-server-name
  | DATABASE 'dbfile'
  | ROLE { WRITER | READER | COORDINATOR }
  | STATUS { INCLUDED | EXCLUDED }
  | ASSIGN AS FAILOVER SERVER
  | host-port-list }
```

- **host-port-list:** - { HOST '*hostname*' PORT *port number* ... } { PRIVATE HOST '*hostname*' PORT *port number* ... }

注意: Sybase 建议您先关闭某一目标服务器，然后再将其排除。如果不这样操作，被排除的服务器将自动关闭并需要 **ALTER MULTIPLEX SERVER *server-name* STATUS INCLUDED** 和同步操作以重新加入 Multiplex。

示例

- **示例** - 以下示例排除辅助服务器 `mpx_writer1`:

```
ALTER MULTIPLEX SERVER mpx_writer1 STATUS EXCLUDED
```

用法

仅适用于 Multiplex。

按如下方式更改 Multiplex 服务器:

RENAME - 更改给定服务器的名称。服务器将自动关闭。下次重新启动需要新名称。

DATABASE – 更改给定服务器的 **Catalog** 文件路径。服务器将自动关闭，下次应使用新 **Catalog** 路径启动。用户应负责（**Sybase Central** 可能会隐藏）重新分配 **Catalog** 文件本身。

ROLE – 更改给定服务器的角色。不允许用户更改协调器的角色或将角色更改为协调器。如果写入程序节点的角色更改为读取程序，服务器将关闭。

STATUS – 更改给定服务器的状态。不能排除故障切换节点，除非它是要排除的最后一个节点。排除后服务器将自动关闭。包括一个节点后，必须进行同步，然后重新启动服务器。

ASSIGN – 将给定服务器指定为新的故障切换服务器。此节点不应处于被排除的状态。**ASSIGN AS FAILOVER** 子句是一个独立的子句，不能与任何其它 **ALTER MULTIPLEX SERVER** 子句一起使用。

协调器必须正在运行，但您可以从 **Multiplex** 中的任何服务器运行 **ALTER MULTIPLEX SERVER** 命令。（**Sybase** 建议所有 DDL 语句均在协调器上运行。）除非将角色从读取程序更改为写入程序，否则在任何情况下，命名服务器都将自动关闭。

注意： **Sybase** 建议您先关闭某一目标服务器，然后再将其排除。如果不这样操作，被排除的服务器将自动关闭并需要 **ALTER MULTIPLEX SERVER server-name STATUS INCLUDED** 和同步操作以重新加入 **Multiplex**。

权限

必须具有 **DBA** 或 **MULTIPLEX ADMIN** 权限。

ALTER PROCEDURE 语句

用修改后的版本替换现有过程。在 **ALTER PROCEDURE** 语句中包括修改后的整个过程，并对该过程重新分配用户权限。

语法

语法 1

```
ALTER PROCEDURE [ owner.]procedure-name procedure-definition
```

语法 2

```
ALTER PROCEDURE [ owner.]procedure-name  
REPLICATE { ON | OFF }
```

语法 3

```
ALTER PROCEDURE [ owner.]procedure-name  
SET HIDDEN
```

语法 4

```
ALTER PROCEDURE [ owner.]procedure-name  
RECOMPILE
```

语法 5

```

ALTER PROCEDURE
[ owner.]procedure-name ( [ parameter, ...] )
[ RESULT (result-column, ...)]
EXTERNAL NAME 'external-call' [ LANGUAGE environment-name ] }

```

参数

- **procedure-definition** - **CREATE PROCEDURE** 语法 (名称之后)
- **environment-name** - **JAVA [DISALLOW | ALLOW SERVER SIDE REQUESTS]**
- **external-call** - [column-name:]function-name@library; ...

用法

ALTER PROCEDURE 语句必须包括 整个新过程。可以将 **PROC** 用作 **PROCEDURE** 的同义词。

ALTER PROCEDURE 语句在语法上与 **CREATE PROCEDURE** 语句相同。

- **语法 1** - 除第一个单词不同以外, **ALTER PROCEDURE** 语句 的语法与 **CREATE PROCEDURE** 语句的语法相同。Watcom 和 Transact-SQL 术语 过程都能通过使用 **ALTER PROCEDURE** 更改。

对于 **ALTER PROCEDURE**, 有关过程的现有权限 不会受到更改。如果先后执行 **DROP PROCEDURE** 和 **CREATE PROCEDURE**, 则会重新指定执行 权限。

- **语法 2** - 如果需要用 Sybase 复制服务器将过程复制到其他站点, 请对过程设置 **REPLICATE ON**。
- **语法 3** - 使用 **SET HIDDEN** 对关联 过程的定义进行模糊处理, 使之不可读。可将过程卸载并重新装载到其他数据库中。

注意: 这种设置不可逆。建议在数据库之外保留 原来的过程定义。

如果使用 **SET HIDDEN**, 则使用 调试程序进行调试将不会显示过程定义, 也无法通过过程概况分析获得过程定义。

不能并用语法 2 和语法 1。

- **语法 4** - 使用 **RECOMPILE** 语法 重新编译存储的过程。当重新编译一个过程时, 会重新分析存储在目录中的定义, 也会验证其语法。对于可生成结果集但不包含 **RESULT** 子句的过程, 数据库服务器会尝试确定 过程结果集的特点, 并将信息存储在目录中。如果自过程创建以来, 过程所引用的表发生变更, 从而添加、删除或重命名了列, 这些信息会很有用。

重新编译不会更改过程的定义。可以 重新编译使用 **SET HIDDEN** 子句隐藏其定义的过程, 但其定义仍是隐藏的。

- **语法 5** - 可将这种语法用于 Java UDF。

iq-environment-name: **JAVA [DISALLOW | ALLOW SERVER SIDE REQUESTS]:**

DISALLOW 为缺省值。

ALLOW 表示允许建立服务器端连接。

注意：除非有必要，否则不要指定 **ALLOW**。设置 **ALLOW** 会减慢某些类型的 Sybase IQ 表连接。

不要在同一查询中一同使用 UDF 与 **ALLOW SERVER SIDE REQUESTS** 和 **DISALLOW SERVER SIDE REQUESTS**。

如果要将 **ALTER PROCEDURE** 语句 用于表 UDF，则所要受到的限制与使用 **CREATE PROCEDURE** 语句（外部过程）时所要受到的限制相同。

标准

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

权限

必须是过程的所有者或者是 DBA。自动提交。

另请参见

- CREATE PROCEDURE 语句（第 112 页）

ALTER SERVER 语句

修改远程服务器的属性。

语法

```
ALTER SERVER server-name
[ CLASS 'server-class' ]
[ USING 'connection-info' ]
[ CAPABILITY 'cap-name' { ON | OFF } ]
[ CONNECTION CLOSE [ CURRENT | ALL | connection-id ] ]
```

参数

- **server-class:** - { *ASAJDBC* / *ASEJDBC* / *ASAODBC* / *ASEODBC* | *DB2ODBC* | *MSSODBC* | *ORAODBC* | *ODBC* }
- **connection-info:** - { *machine-name:port-number* [/ *dbname*] | *data-source-name* }
- **cap-name:** - 服务器功能的名称

示例

- **示例 1** - 更改名为 `ase_prod` 的 Adaptive Server Enterprise 服务器的服务器类，使它与 Sybase IQ 的连接基于 ODBC。数据源名称为 `ase_prod`。

```
ALTER SERVER ase_prod
CLASS 'ASEODBC'
USING 'ase_prod'
```

- **示例 2** - 更改服务器 `infodc` 的一项功能：

```
ALTER SERVER infodc
CAPABILITY 'insert select' OFF
```

- **示例 3** - 关闭与名为 `rem_test` 的远程服务器的所有连接：

```
ALTER SERVER rem_test
CONNECTION CLOSE ALL
```

- **示例 4** - 关闭与名为 `rem_test` 的远程服务器的连接（连接 ID 为 142536）：

```
ALTER SERVER rem_test
CONNECTION CLOSE 142536
```

用法

ALTER SERVER 所做的更改将直到下一次与远程服务器连接时才生效。

CLASS 子句 - 使用 **CLASS** 子句更改服务器类。有关服务器类的详细信息，请参见《系统管理指南第二卷》>“访问远程数据”和《系统管理指南第二卷》>“用于远程数据访问的服务器类”。

USING 子句 - **USING** 子句更改服务器的连接信息。有关连接信息的详细信息，请参见“CREATE SERVER 语句”。

CAPABILITY 子句 - **CAPABILITY** 子句可将服务器功能设置为 ON 或 OFF。服务器功能存储在系统表 `SYSCAPABILITY` 中。这些功能的名称存储在系统表 `SYSCAPABILITYNAME` 中。对于远程服务器，直到与该服务器建立第一个连接后，`SYSCAPABILITY` 表才会包含条目。在第一个连接中，Sybase IQ 向服务器询问其功能，然后填充 `SYSCAPABILITY`。对于后面的连接，它从该表中获取服务器的功能。

通常情况下，不需要变更服务器的功能。可能需要变更属于 ODBC 类的通用服务器的功能。

CONNECTION CLOSE 子句 - 当用户创建与远程服务器的连接时，在用户从本地数据库断开连接之前，远程连接不会关闭。您可以使用 **CONNECTION CLOSE** 子句显式关闭与远程服务器的连接。当远程连接处于非活动状态或不再需要时，此操作非常有用。

下列 SQL 语句等同，并且关闭与远程服务器的当前连接。

```
ALTER SERVER server-name CONNECTION CLOSE
```

```
ALTER SERVER server-name CONNECTION CLOSE CURRENT
```

使用此语法可关闭与远程服务器的 ODBC 和 JDBC 连接。无需 DBA 权限即可执行上述任一语句。

还可以通过指定连接 ID 来断开特定远程 ODBC 连接，或通过指定 ALL 关键字来断开所有远程 ODBC 连接。如果尝试通过指定连接 ID 或 ALL 关键字来关闭 JDBC 连接，则会发生错误。当 *connection-id* 标识的连接不是当前本地连接时，用户必须拥有 DBA 权限才能关闭连接。

副作用：

- 自动提交

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 受 Open Client/Open Server 支持。

权限

必须具有 DBA 权限才能执行此命令。

另请参见

- CREATE SERVER 语句（第 132 页）
- DROP SERVER 语句（第 177 页）

ALTER SERVICE 语句

变更 Web 服务。

语法

```
ALTER SERVICE service-name
[ TYPE 'service-type-string' ]
[ attributes ]
[ AS statement' ]
```

参数

- 属性：[AUTHORIZATION { ON | OFF }] [SECURE { ON | OFF }] [USER *user-name* | NULL] [URL [PATH] { PATH } { ON | OFF | ELEMENTS }] [USING (SOAP-prefix | NULL)] - *service-type-string*: { 'RAW' | 'HTML' | 'XML' | 'SOAP' | 'DISH' }

示例

- 示例 1 - 若要快速设置 Web 服务器，请使用 **-xs** 开关启动数据库服务器，然后执行以下语句：

```
CREATE SERVICE tables TYPE 'HTML'

ALTER SERVICE tables
AUTHORIZATION OFF
USER DBA
AS SELECT * FROM SYS.ISYSTAB
```

在执行这些语句之后，使用任意 Web 浏览器打开 URL <http://localhost/tables>。

用法

ALTER SERVICE 语句会将数据库服务器作为 Web 服务器。

service-name - 不能重命名 Web 服务。

service-type-string - 标识服务类型。该类型必须是列出的服务类型之一。无缺省值。

AUTHORIZATION 子句 - 确定当连接到该服务时用户是否必须指定用户名和口令。如果 **AUTHORIZATION** 为 **OFF**，则需要使用 **AS** 子句，并且必须用 **USER** 子句标识单个用户。使用该用户的帐户和权限执行所有请求。

如果 **AUTHORIZATION** 为 **ON**，则所有用户必须提供用户名和口令。或者，通过使用 **USER** 子句提供用户名或组名，您可以限制允许使用该服务的用户。如果用户名为 **NULL**，则所有已知用户都可以访问该服务。

缺省值为 **ON**。建议在启用了授权的情况下运行生产系统，并且通过将用户添加到组中来授予使用该服务的权限。

SECURE 子句 - 表示是否接受不安全的连接。**ON** 表示只接受 **HTTPS** 连接。在 **HTTP** 端口上接收的服务请求将自动重定向到 **HTTPS** 端口。如果设置为 **OFF**，则 **HTTP** 和 **HTTPS** 连接都接受。缺省值为 **OFF**。

USER 子句 - 如果禁用授权，则此参数成为必需的参数，并指定用于执行所有服务请求的用户 **ID**。如果启用授权（缺省值），则此可选子句标识允许访问该服务的用户或组。缺省值为 **NULL**，即向所有用户授予访问权限。

URL 子句 - 确定是否接受 **URI** 路径；如果接受，确定如何处理 **URI** 路径。**OFF** 指示 **URI** 请求中服务名后面不必有任何内容。**ON** 指示将 **URI** 中的其余部分解释为变量 **url** 的值。**ELEMENTS** 表示将 **URI** 路径的其余部分在斜线字符处分隔开，并分隔为每组最多包含 10 个元素的一个列表。将这些值指派给名为 **url** 的变量外加介于 1 和 10 之间的数字后缀；例如，前三个变量名称分别为 **url1**、**url2** 和 **url3**。如果提供的值不到 10 个，则将其余变量设置为 **NULL**。如果服务名以字符 / 结尾，则必须将 **URL** 设置为 **OFF**。缺省值为 **OFF**。

USING 子句 - 此子句仅适用于 **DISH** 服务。该参数指定名称前缀。只处理其名称以此前缀开头的 **SOAP** 服务。

语句 - 如果该语句为空值，则 **URI** 必须指定要执行的语句。否则，指定的 **SQL** 语句是唯一可通过该服务执行的语句。**SOAP** 服务必须含有语句；而 **DISH** 服务必须不含有语句。缺省值为 **NULL**。

强烈建议在生产系统中运行的所有服务都定义一个语句。只有在启用了授权的情况下，该语句才能为 NULL。

RAW - 将结果集发送到客户端，而不进行任何进一步的格式设置。可以通过在过程中显式生成所需的标记来产生格式化文档。

HTML - 自动将语句或过程的结果集转换为包含表的 HTML 文档格式。

XML - 结果集采用 XML 格式。如果还不是 XML，则自动转换为 XML RAW 格式。

SOAP - 请求必须为有效的简单对象访问协议 (SOAP) 请求。结果集将自动设置为 SOAP 响应的格式。有关 SOAP 标准的详细信息，请参见 www.w3.org/TR/SOAP。

DISH - Determine SOAP Handler (DISH) 服务充当一个或多个 SOAP 服务的代理。在使用时，该服务充当容纳多个 SOAP 服务并为这些服务提供访问权限的容器。Web 服务描述语言 (WSDL) 文件是针对每个所包含的 SOAP 服务自动生成的。所包含的 SOAP 服务由公用前缀标识，该前缀必须在 **USING** 子句中指定。

另请参见“SQL Anywhere Server - 编程” > “SQL Anywhere Web 服务” > “使用 SQL Anywhere 作为 HTTP Web 服务器”。

注意： 此参考指向 SQL Anywhere 文档。

标准

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

权限

必须具有 DBA 权限。

另请参见

- CREATE SERVICE 语句（第 133 页）
- DROP SERVICE 语句（第 178 页）

ALTER TABLE 语句

修改表的定义。

语法

```
ALTER TABLE [ owner. ] table-name
{ alter-clause, ... }
```

参数

- **alter-clause:** - **ADD** *create-clause* | **ALTER** *column-name* *column-alteration* | **ALTER** [**CONSTRAINT** *constraint-name*] **CHECK** (*condition*) | **DROP** *drop-object* | **RENAME** *rename-object* | *move-clause* | **SPLIT PARTITION** *partition-name* **INTO** (*partition-decl-1*, *partition-decl-2*) | **MERGE PARTITION** *partition-name-1* **INTO** *partition-name-2* | **UNPARTITION** | **PARTITION BY RANGE** (*partition-key*) *range-partition-decl*
- **create-clause:** - *column-name* *column-definition* [*column-constraint*] | *table-constraint* | **PARTITION BY** *partitioning-schema*
- **column-alteration:** - { *alterable-column-attribute* } [*alterable-column-attribute*...] | **ADD** [*constraint-name*] **CHECK** (*condition*) | **DROP** { **DEFAULT** | **CHECK** | **CONSTRAINT** *constraint-name* }
- **alterable-column-attribute:** - [**NOT**] **NULL** | **DEFAULT** *default-value* | [**CONSTRAINT** *constraint-name*] **CHECK** { **NULL** | (*condition*) }
- **column-constraint:** - [**CONSTRAINT** *constraint-name*] { **UNIQUE** | **PRIMARYKEY** | **REFERENCES** *table-name* [(*column-name*)] [*actions*] | **CHECK** (*condition*) | **IQ** **UNIQUE** (*integer*) }
- **drop-object:** - { *column-name* | **CHECK** | **CONSTRAINT** *constraint-name* | **UNIQUE** (*index-columns-list*) | **PRIMARY KEY** | **FOREIGN KEY** *fkey-name* | **PARTITION** *range-partition-name* }
- **move-clause:** - { **ALTER** *column-name* **MOVE** { **PARTITION** (*partition-name* **TO** *new-dbspace-name*) | **TO** *new-dbspace-name* } } | **MOVE PARTITION** *partition-name* **TO** *new-dbspace-name* | **MOVE TO** *new-dbspace-name* | **MOVE METADATA TO** *new-dbspace-name*
- **rename-object:** - *new-table-name* | *column-name* **TO** *new-column-name* | **CONSTRAINT** *constraint-name* **TO** *new-constraint-name* | **PARTITION** *partition-name* **TO** *new-partition-name*
- **column-definition:** - *column-namedata-type* [**NOTNULL** | **NULL**] [**IN** *dbspace-name*] [**DEFAULT** *default-value* | **IDENTITY**]
- **default-value:** - *special-value* | *string* | *global variable* | [-] *number* | (*constant-expression*) | *built-in-function* (*constant-expression*) | **AUTOINCREMENT** | **NULL** | **TIMESTAMP** | **LAST USER** | **USER**
- **special-value:** - **CURRENT** { **DATABASE** | **DATE** | **REMOTE USER** | **TIME** | **TIMESTAMP** | **USER** | **PUBLISHER** }
- **table-constraint:** - [**CONSTRAINT** *constraint-name*] { **UNIQUE** (*column-name* [, ...]) | **PRIMARYKEY** (*column-name* [, ...]) | *foreign-key-constraint* | **CHECK** (*condition*) }
- **foreign-key-constraint:** - **FOREIGNKEY** [*role-name*] [(*column-name* [, ...])] ... **REFERENCES** *table-name* [(*column-name* [, ...])] ... [*actions*] [

- **rename-object:** - *new-table-name* | *column-name* **TO** *new-column-name*
| **CONSTRAINT** *constraint-name* **TO** *new-constraint-name* | **PARTITION** *partition-name* **TO** *new-partition-name*
- **range-partitioning-scheme:** - **RANGE**(*partition-key*)(*range-partition-decl* [, *range-partition-decl* ...])
- **partition-key:** - *column-name*
- **range-partition-decl:** - *partition-name* **VALUES** <= ({ *constant* | **MAX** }) [*in dbspace-name*]
- **actions:** - [**ON** { **UPDATE** | **DELETE** } *action*]
- **action:** - { **RESTRICT** }

示例

- **示例 1** - 向 Employees 表添加一个新列，说明他们在哪个办公室工作：

```
ALTER TABLE Employees
ADD office CHAR(20)
```

- **示例 2** - 从 office 表中删除 Employees 列：

```
ALTER TABLE Employees
DROP office
```

- **示例 3** - 向 Customers 表添加一列，为每个客户指定一个销售联系人：

```
ALTER TABLE Customers
ADD SalesContact INTEGER
REFERENCES Employees (EmployeeID)
```

- **示例 4** - 向 CustomerNum 表添加一个新列 Customers，并指定缺省值 88：

```
ALTER TABLE Customers
ADD CustomerNum INTEGER DEFAULT 88
```

- **示例 5** - 只有 c2、c4 和 c5 的 **FP** 索引会从数据库空间 Dsp3 移动到 Dsp6。c1 的 **FP** 索引保留在 Dsp1 中。c3 的 **FP** 索引保留在 Dsp2 中。c5 的主键保留在 Dsp4 中。**DATE** 索引 c4_date 保留在 Dsp5 中。

```
CREATE TABLE foo (
    c1 INT IN Dsp1,
    c2 VARCHAR(20),
    c3 CLOB IN Dsp2,
    c4 DATE,
    c5 BIGINT,
    PRIMARY KEY (c5) IN Dsp4) IN Dsp3);

CREATE DATE INDEX c4_date ON foo(c4) IN Dsp5;
ALTER TABLE foo
MOVE TO Dsp6;
```

- **示例 6** - 只将 **FP** 索引 c1 从数据库空间 Dsp1 移动到 Dsp7：

```
ALTER TABLE foo ALTER c1 MOVE TO Dsp7
```

- **示例 7** - 此示例说明了许多 **ALTER TABLE** 子句移动、拆分、重命名和合并分区的使用法。

创建分区表:

```
CREATE TABLE bar (
    c1 INT,
    c2 DATE,
    c3 VARCHAR(10))
PARTITION BY RANGE(c2)
(p1 VALUES <= ('2005-12-31') IN dbsp1,
 p2 VALUES <= ('2006-12-31') IN dbsp2,
 p3 VALUES <= ('2007-12-31') IN dbsp3,
 p4 VALUES <= ('2008-12-31') IN dbsp4);

INSERT INTO bar VALUES(3, '2007-01-01', 'banana nut');
INSERT INTO bar VALUES(4, '2007-09-09', 'grape jam');
INSERT INTO bar VALUES(5, '2008-05-05', 'apple cake');
```

将分区 p2 移动到 dbsp5:

```
ALTER TABLE bar MOVE PARTITION p2 TO DBSP5;
```

将分区 p4 拆分为 2 个分区:

```
ALTER TABLE bar SPLIT PARTITION p4 INTO
(P41 VALUES <= ('2008-06-30') IN dbsp4,
 P42 VALUES <= ('2008-12-31') IN dbsp4);
```

下面的 **SPLIT PARTITION** 报告一个错误, 因为它需要移动数据。拆分后, 并非所有现有行都位于同一分区内。

```
ALTER TABLE bar SPLIT PARTITION p3 INTO
(P31 VALUES <= ('2007-06-30') IN dbsp3,
 P32 VALUES <= ('2007-12-31') IN dbsp3);
```

会报告以下错误:

“不允许移动数据, 无法拆分分区 p3。”

下面的 **SPLIT PARTITION** 报告一个错误, 因为它更改了分区的边界值:

```
ALTER TABLE bar SPLIT PARTITION p2 INTO
(p21 VALUES <= ('2006-06-30') IN dbsp2,
 P22 VALUES <= ('2006-12-01') IN dbsp2);
```

会报告以下错误:

“无法更改分区 p2 的边界值。”

将分区 p3 合并到 p2 中。由于不允许从高边界值分区合并到低边界值分区, 因此系统报告一个错误。

```
ALTER TABLE bar MERGE PARTITION p3 into p2;
```

会报告以下错误:

“分区 ‘p2’ 与分区 ‘p3’ 不相邻或者不在分区 ‘p3’ 之前。”

将分区 p2 合并到 p3 中:

```
ALTER TABLE bar MERGE PARTITION p2 INTO p3;
```

将分区 p1 重命名为 p1_new:

```
ALTER TABLE bar RENAME PARTITION p1 TO p1_new;
```

对表 bar 取消分区:

```
ALTER TABLE bar UNPARTITION;
```

对表 bar 进行分区。此命令报告一个错误，因为所有行必须位于第一个分区中。

```
ALTER TABLE bar PARTITION BY RANGE(c2)
  (p1 VALUES <= ('2005-12-31') IN dbbsp1,
   p2 VALUES <= ('2006-12-31') IN dbbsp2,
   p3 VALUES <= ('2007-12-31') IN dbbsp3,
   p4 VALUES <= ('2008-12-31') IN dbbsp4);
```

会报告以下错误:

“所有行必须位于第一个分区内。”

对表 bar 进行分区:

```
ALTER TABLE bar PARTITION BY RANGE(c2)
  (p1 VALUES <= ('2008-12-31') IN dbbsp1,
   p2 VALUES <= ('2009-12-31') IN dbbsp2,
   p3 VALUES <= ('2010-12-31') IN dbbsp3,
   p4 VALUES <= ('2011-12-31') IN dbbsp4);
```

用法

ALTER TABLE 语句更改以前创建的表中的表属性（列定义和约束）。其语法允许使用更改子句列表；但是，在每个 **ALTER TABLE** 语句中，只能添加、修改或删除一个表约束或列约束。

注意： 您不能更改本地临时表，但可以在仅有一个连接使用全局临时表时，更改全局临时表。

Sybase IQ 会强制执行 **REFERENCES** 和 **CHECK** 约束。在 **ALTER TABLE** 语句中添加的表和/或列检查约束不会被当作该更改表操作的一部分。有关 **CHECK** 约束的详细信息，请参见“**CREATE TABLE** 语句”。

如果视图定义中使用了 **SELECT ***，并且更改了 **SELECT *** 引用的某个表，则您必须运行 **ALTER VIEW <viewname> RECOMPILE** 以确保视图定义正确，并避免在查询该视图时出现意外结果。

ADD column-definition [column-constraint] - 向表添加一个新列。表必须为空才可指定 **NOT NULL**。添加 **IDENTITY** 或 **DEFAULT AUTOINCREMENT** 列时，表可能包含数据。如果列有缺省 **IDENTITY** 值，新列的所有行将用顺序值来填充。也可为单列键添加外键约束作为其列约束。**IDENTITY/DEFAULT AUTOINCREMENT** 列的值唯一地标识表中的每一行。**IDENTITY/DEFAULT AUTOINCREMENT** 列存储在插入和更新期间自动生成的序列号。**DEFAULT AUTOINCREMENT** 列也称为 **IDENTITY** 列。使用 **IDENTITY/DEFAULT AUTOINCREMENT** 时，列必须是整数数据类型之一，或精确数值类型，且

标度为 0。有关列约束和 **IDENTITY/DEFAULT AUTOINCREMENT** 列的详细信息，请参见“**CREATE TABLE 语句**”。

注意： 无法将外键约束添加到用 Sybase IQ 版本 12.4.3 或更早版本创建的 未受强制的主键上。

ALTER column-name column-alteration - 更改列的定义。允许进行以下修改：

- **SET DEFAULT default-value** - 更改表中现有列的缺省值。也可以将 **MODIFY** 子句用于此任务，但是 **ALTER** 符合 ISO/ANSI SQL 标准，而 **MODIFY** 则不符合该标准。修改缺省值不会更改表中的任何现有值。
- **DROP DEFAULT** - 删除表中现有列的缺省值。也可以将 **MODIFY** 子句用于此任务，但是 **ALTER** 符合 ISO/ANSI SQL 标准，而 **MODIFY** 则不符合该标准。删除缺省值不会更改表中的任何现有值。
- **ADD** - 在列中添加一个命名约束或一个 **CHECK** 条件。新的约束或条件仅在定义后应用于对表的操作。但不会对表中现有的值进行校验来确认其是否满足新增的约束或条件。
- **CONSTRAINT column-constraint-name** - 通过可选列约束名，您可以在以后修改或删除个别约束，而不必修改整个列约束。
- **[CONSTRAINT constraint-name] CHECK (condition)** - 使用此子句在列上添加 **CHECK** 约束。
- **SET COMPUTE (expression)** - 更改与计算列关联的表达式。当执行语句时，重新计算列中的值，如果新表达式无效，则语句执行失败。
- **DROP COMPUTE** - 将列从计算列更改为非计算列。此语句不更改表中的任何现有值。

DROP 分区子句 - 删除指定的分区。行被删除，分区定义被删除。您不能删除最后一个分区，因为删除最后一个分区会将分区表转变为未分区的表。（要合并分区表，请改用 **UNPARTITION** 子句。）例如：

```
CREATE TABLE foo (c1 INT, c2 INT)
  PARTITION BY RANGE (c1)
    (P1 VALUES <= (100) IN dbsp1,
     P2 VALUES <= (200) IN dbsp2,
     P3 VALUES <= (MAX) IN dbsp3
    ) IN dbsp4);
LOAD TABLE ...
ALTER TABLE DROP PARTITION P1;
```

ADD table-constraint - 为表添加约束。也可单列键或多列键添加外键约束作为其表约束。有关表约束的完整说明，请参见“**CREATE TABLE 语句**”。

如果指定了 **PRIMARY KEY**，则表一定不能已具有通过 **CREATE TABLE** 语句或另一 **ALTER TABLE** 语句创建的主键。

注意： 不能通过 **MODIFY** 修改表约束或列约束。要更改约束，请通过 **DELETE** 删除旧约束并通过 **ADD** 添加新约束。

DROP column-name - 从表中删除列。如果列包含在任何多列索引、唯一性约束、外键或主键中，则必须删除索引、约束或键后才能删除列。这不会删除引用该列的 **CHECK**

约束。只有在 **IDENTITY_INSERT** 处于关闭状态并且表不是本地临时表的情况下，才能删除 **IDENTITY/DEFAULT AUTOINCREMENT** 列。

DROP CHECK - 删除表的所有检查约束。这包括表检查约束和列检查约束。

DROP CONSTRAINT constraint-name - 删除表或指定列的命名的约束。

DROP UNIQUE (column-name,...) - 删除指定列上的唯一约束。还删除引用该唯一约束的任何外键（而不是主键）。如果有相关联的外键约束，系统会报告一个错误。在删除主键约束之前，请使用 **ALTER TABLE** 删除所有引用该主键的外键。

DROP PRIMARY KEY - 删除主键。还删除引用该表的主键的所有外键。如果有相关联的外键约束，系统会报告一个错误。如果主键是未受强制的主键，并存在相关联的未受强制的外键约束，**DELETE** 会返回一个错误。

DROP FOREIGN KEY role-name - 删除此表的具有给定角色名称的外键约束。系统将保留为外键约束隐式创建的非唯一 **HG** 索引。用户可以使用 **DROP INDEX** 语句显式删除 **HG** 索引。

DROP PARTITION - 删除分区 **p1** 中的行和 **p1** 的分区定义。如果插入的新行的 **c1** 列的值为 99，则它将被置于数据库空间 **dbbsp2** 中的分区 **p2** 下。

RENAME new-table-name - 将表名更改为 *new-table-name*。必须修改所有使用旧表名的应用程序。但是，被自动分配与旧表名相同名称的任何外键不会更改名称。

RENAME column-name TO new-column-name - 将列名更改为 *new-column-name*。必须修改任何使用旧列名的应用程序。

RENAME constraint-name TO new-constraint-name - 将约束名更改为 *new-constraint-name*。必须修改任何使用旧约束名的应用程序。

只要语句影响了当前正由其它连接使用的表，系统就会禁止 **ALTER TABLE**。**ALTER TABLE** 可能很耗时，当正在处理语句时，服务器不会处理引用该表的请求。

ALTER Column MOVE TO - 对于未分区的表，将指定列移动到新的数据库空间。而对于分区表，则不能请求 **ALTER Column MOVE TO** 子句。**ALTER Column MOVE PARTITION** 子句将指定分区的列移动到指定的数据库空间。

MOVE PARTITION - 将指定分区移动到新的数据库空间。

MOVE TO - 当表被映射至新的数据库空间时，移动同一数据库空间内驻留的所有表对象，包括列、索引、唯一约束、主键、外键和元数据。

每个表对象只能驻留于一个数据库空间内。任何类型的 **ALTER MOVE** 都会在整个移动期间阻止对表执行任何修改。

MOVE TABLE METADATA - 将表的元数据移动到新的数据库空间。对于分区表，**MOVE TABLE METADATA** 子句还会移动各分区之间共享的元数据。

您必须具有 **DBA** 或 **SPACE ADMIN** 权限，或者对新数据库空间具有 **CREATE** 特权，并且是表所有者或者对表具有更改权限。

SPLIT PARTITION - 将指定分区拆分为两个分区。仅当不必移动任何数据时才能拆分分区。待拆分分区的全部现有行在拆分后必须保留于一个分区内。*partition-decl-1* 的边界值必须小于 *partition-name* 的边界值，并且 *partition-decl-2* 的边界值必须等于 *partition-name* 的边界值。您可以为两个新分区指定不同名称。如果未指定新名称，旧的 *partition-name* 只能用于第二个分区。

MERGE PARTITION - 将 *partition-name-1* 合并到 *partition-name-2* 中。如果两个分区相邻则可以合并，数据将驻留于同一个数据库空间内。只能将低分区值的分区合并到具有高分区值的相邻分区内。请注意，服务器不会检查分区所合并到的数据库空间的 **CREATE** 权限。有关如何创建相邻分区的示例，请参见“**CREATE TABLE** 语句”中的“示例 3”。

UNPARTITION - 从分区表中删除分区。每个列都被置于一个数据库空间内。请注意，服务器不会检查所有分区数据所移动到的数据库空间的 **CREATE** 权限。**ALTER TABLE UNPARTITION** 会阻止所有数据库活动。

PARTITION BY - 对未分区的表进行分区。如果所有现有行都属于第一个分区，则可以对未分区的表进行分区。您可以为第一个分区指定与列或表所在数据库空间不同的另外一个数据库空间。但并不会移动现有行。而是将列/分区的正常数据库空间保留于现有列的 `SYS.ISSYSIQPARTITIONCOLUMN` 中。只有后来添加到第一个分区内的缺省或最大标识列将存储于第一个分区的指定数据库空间内。

RENAME PARTITION - 将现有分区名称重命名为新的分区名称。

副作用：

- 自动提交。**ALTER** 和 **DROP** 选项关闭当前连接的所有游标。系统还会清除 **dbisql** 数据窗口。
- **ALTER TABLE** 操作开始时，系统会执行检查点操作。
- 一旦更改了列或表，引用更改列的任何存储过程、视图或其它项将不再有效。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - Adaptive Server Enterprise 支持某些子句。

权限

对于 **MOVE** 语法，必须满足以下条件之一：

- 具有 **DBA** 或 **SPACE ADMIN** 权限
- 对新数据库空间具有 **CREATE** 权限并且是表所有者
- 对表具有 **ALTER** 权限

对于 **MOVE** 之外的语法，必须满足以下条件之一：

- 具有 **DBA** 权限
- 对新数据库空间具有 **CREATE** 权限并且是表所有者

- 对表具有 ALTER 权限

要求对表具有独占访问权限。

另请参见

- CREATE TABLE 语句（第 136 页）
- DROP 语句（第 170 页）
- IDENTITY_INSERT 选项（第 378 页）

ALTER TEXT CONFIGURATION 语句

改变文本配置对象。

语法

See below.

用法

有关语法和完整说明，请参见《Sybase IQ 中的非结构化数据分析》。

ALTER TEXT INDEX 语句

更改 TEXT 索引的定义。

语法

请参见下文。

用法

有关语法和完整说明，请参见《Sybase IQ 中的非结构化数据分析》。

ALTER USER 语句

更改用户设置。

语法

语法 1

```
ALTER USER user-name [ IDENTIFIED BY password ] [ LOGIN POLICY policy-name ]  
[ FORCE PASSWORD CHANGE { ON | OFF } ]
```

语法 2

ALTER USER *user-name* [**RESET LOGIN POLICY**]

示例

- **示例 1** - 更改名为 SQLTester 的用户。口令被设置为 “welcome”。SQLTester 用户被分配到 Test1 登录策略，下次登录时口令不会过期。

```
ALTER USER SQLTester
IDENTIFIED BY welcome
LOGIN POLICY Test1
FORCE PASSWORD CHANGE OFF
```

执行此命令的用户必须具有 **USER ADMIN** 和 **PERMS ADMIN** 或 **DBA** 权限。必须具有 **PERMS ADMIN** 权限才能更改口令，必须具有 **USER ADMIN** 权限才能更改登录策略。

用法

user--name - 用户的名称。

IDENTIFIED BY 子句 - 为用户提供口令的子句。

policy-name - 要将用户分配到的登录策略的名称。如果未指定 **LOGIN POLICY** 子句，则不会进行任何更改。

FORCE PASSWORD CHANGE 子句 - 控制用户登录时是否必须指定新口令。此设置覆盖其策略中的 **PASSWORD_EXPIRY_ON_NEXT_LOGIN** 选项设置。

RESET LOGIN POLICY 子句 - 将用户登录设置恢复为登录策略中的原始值。这通常会清除由于用户超出失败登录次数或超出上次登录后的最大天数而被隐式设置的所有锁。当您重置登录策略时，用户可以访问因超过 **MAX_FAILED_LOGIN_ATTEMPTS** 或 **MAX_DAYS_SINCE_LOGIN** 等登录策略选项限制而被锁定的帐户。

有关管理登录策略的详细信息，请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “管理用户 ID 和权限” > “管理登录策略”。

注意： 此参考指向 SQL Anywhere 文档。

《使用 Sybase IQ Multiplex》中介绍了适用于 Multiplex 的增强型 **ALTER LOGIN POLICY** 语法。

用户 ID 和口令不能：

- 以空格、单引号或双引号开头
- 以空格结尾
- 包含分号

如果您将 **PASSWORD_EXPIRY_ON_NEXT_LOGIN** 值设置为 **ON**，则分配到此登录策略的所有用户的口令都将在用户下次登录时立即过期。您可以使用 **ALTER USER** 和 **LOGIN POLICY** 子句强制用户在下次登录时更改口令。

标准

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

权限

用户可以更改自己的口令。要更改其他用户的口令，必须具有 DBA 或 PERMS ADMIN 权限。用户需要具有 DBA 或 USER ADMIN 权限，才能执行 **ALTER USER LOGIN POLICY**、**FORCE PASSWORD CHANGE** 或 **RESET LOGIN POLICY**。

另请参见

- COMMENT 语句（第 61 页）
- CREATE LOGIN POLICY 语句（第 108 页）
- CREATE USER 语句（第 147 页）
- DROP LOGIN POLICY 语句（第 175 页）
- DROP USER 语句（第 180 页）
- GRANT 语句（第 202 页）
- REVOKE 语句（第 270 页）
- ALTER LOGIN POLICY 语句（第 19 页）

ALTER VIEW 语句

用修改后的版本替换视图定义。

语法

语法 1 - 更改视图的结构

```
ALTER VIEW
... [ owner.]view-name [ ( column-name [ , ... ] ) ]
... AS select-statement
... [ WITH CHECK OPTION ]
```

语法 2 - 更改视图的属性

```
ALTER VIEW
... [ owner.]view-name
... { SET HIDDEN | RECOMPILE | DISABLE | ENABLE }
```

用法

AS - 用途和语法与 **CREATE VIEW** 语句相同。请参见“CREATE VIEW 语句”。

WITH CHECK OPTION - 用途和语法与 **CREATE VIEW** 语句相同。请参见“CREATE VIEW 语句”。

SET HIDDEN - 将视图的定义进行模糊处理，并隐藏该视图（例如，在 Sybase Central 中）。视图的显式引用仍有效。

警告！ SET HIDDEN 操作是不可逆的。

RECOMPILE - 为视图重新创建列定义。在功能上与 **ENABLE** 子句相同，不同的是，此子句还可以用于未被禁用的视图上。

DISABLE - 禁止数据库服务器使用视图。

ENABLE - 启用已被禁用的视图，该操作会使数据库服务器为该视图重新创建列定义。在启用视图之前，您必须启用该视图所依赖的所有视图。

当您变更视图时，视图的现有权限将保留，不需要重新指派。此外，您还可以不使用 **ALTER VIEW** 语句，而分别使用 **DROP VIEW** 和 **CREATE VIEW** 来删除视图和重新创建视图。否则，必须重新指派视图权限。

使用语法 1 完成视图变更之后，数据库服务器将重新编译该视图。根据您的执行更改的类型，如果存在相关视图，数据库服务器将尝试重新编译它们。如果您执行的更改影响相关视图，则该视图可能会变为无效视图，从而需要更改相关视图的定义。

警告！ 如果定义视图的 **SELECT** 语句中包含星号 (*)，则向基础表中添加列或从基础表中删除列时，视图中列的数目可能会变化。视图列的名称和数据类型也会变化。

语法 1 - 更改视图的结构。与变更表不同，当您变更表时，您的变更可能限于各个列，而变更视图结构则要求您用新定义更换整个视图定义，因此与创建视图更相似。有关用于定义视图结构的参数的说明，请参见“**CREATE VIEW** 语句”。

语法 2 - 更改视图属性，例如是否隐藏视图定义。

当您使用 **SET HIDDEN** 时，您可以卸载视图并将其重新装载到其它数据库中。使用调试程序进行调试并不显示视图定义，过程分析中也不显示视图定义。如果需要更改隐藏视图的定义，则必须删除该视图并使用 **CREATE VIEW** 语句重新创建。

当您使用 **DISABLE** 子句时，数据库服务器无法再使用该视图来回答查询。禁用视图类似于删除视图，不同的是，禁用视图时，视图定义仍保留于数据库中。禁用视图还会禁用所有相关视图。因此，**DISABLE** 子句不仅要求对被禁用视图具有独占访问权限，而且对同时将被禁用的所有相关视图也必须具有独占访问权限。

另请参见《系统管理指南第二卷》>“使用过程和批处理”>“隐藏过程、函数和视图的内容”以及《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sa_dependent_views 过程”。

有关数据库服务器如何处理视图依赖性的详细信息，请参见“SQL Anywhere Server - SQL 用法”>“使用数据库对象”>“使用视图”>“视图依赖性”。

注意： 此参考指向 SQL Anywhere 文档。

副作用：

- 自动提交

- 系统将从内存中卸载所有过程和触发器，因此引用该视图的所有过程或触发器都会反映新视图定义。如果您经常更改视图，过程和触发器的卸载与装载会影响性能。

标准

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

权限

必须是视图的所有者或者具有 DBA 权限。

另请参见

- CREATE VIEW 语句（第 151 页）
- DROP 语句（第 170 页）
- 确定和修复无效的相关视图（第 43 页）

确定和修复无效的相关视图

检查并更正由于基础表发生变化而变为无效视图的任何相关视图。

在大多数情况下，如果基础表发生变化，数据库服务器会自动重新编译视图，以使视图保持有效。但是，如果更改表的操作删除了视图定义引用的内容或对其进行了实质性更改，则相关视图将变为无效视图。例如，如果您删除了视图定义中引用的一列，则相关视图将不再有效。请更正视图定义并手动重新编译视图。

1. 运行 **sa_dependent_views** 以获取相关视图的列表。
2. 执行更改表的 DDL 操作。服务器将自动禁用相关视图，并在 DDL 完成后尝试重新编译这些视图。
3. 检查 **sa_dependent_views** 列出的所有视图是否有效。例如，执行 **SELECT * FROM myview** 之类的简单测试。
4. 如果某个视图无效，您可能需要更改视图定义来解决该问题。针对您所做的 DDL 更改检查视图定义，并进行必要的更改。运行 **ALTER VIEW RECOMPILE** 以更正视图定义。
5. 测试更正的视图以确保其能够正常工作。例如，执行 **SELECT * FROM myview** 之类的简单测试。

sa_dependent_views 会返回某个给定表或视图的所有相关视图的列表。请参见《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sa_dependent_views 过程”和“SQL Anywhere Server - SQL 参考”>“系统过程”>“按字母顺序排列的系统过程列表”>“sa_dependent_views 系统过程”。

有关数据库服务器如何处理视图依赖性的详细信息，请参见“SQL Anywhere Server - SQL 用法” > “使用数据库对象” > “使用视图” > “视图依赖性”。

注意： 这些参考指向 SQL Anywhere 文档。

另请参见

- ALTER VIEW 语句 (第 41 页)

BACKUP 语句

在一个或多个存档设备上备份 Sybase IQ 数据库。

语法

```
BACKUP DATABASE  
[ backup-option... ]  
TO archive_device [ archive-option... ]  
... [ WITH COMMENT string ]
```

参数

- **backup-option:** -

```
{ READWRITE FILES ONLY |  
READONLY } dbspace-or-file [, ... ] }  
CRC { ON | OFF }  
ATTENDED { ON | OFF }  
BLOCK FACTOR integer  
{ FULL | INCREMENTAL | INCREMENTAL SINCE FULL }  
VIRTUAL { DECOUPLED | ENCAPSULATED 'shell_command' }  
WITH COMMENT comment
```

- **dbspace-or-file:** -

```
{ DBSPACES identifier-list | FILES identifier-list }
```

- **identifier-list:** - *identifier* [, ...]
- **archive-option:** - **SIZE***integer***STACKER***integer*

示例

- **示例 1** - 此 UNIX 示例将 iqdemo 数据库备份到 Sun Solaris 平台上的磁带设备 /dev/rmt/0 和 /dev/rmt/2 中。在 Solaris 上，设备名后面的字母 *n* 指定“无关闭回绕”功能。请始终用 **BACKUP** 来指定此功能，并使用适用于您的 UNIX 平台

的命名约定（Windows 不支持此功能）。此示例备份了自上次完全备份以来服务器上的所有更改：

```
BACKUP DATABASE
INCREMENTAL SINCE FULL
TO '/dev/rmt/0n' SIZE 10000000
TO '/dev/rmt/2n' SIZE 15000000
```

注意：大小单位为千字节 (KB)，但在大多数情况下，大小小于 1GB 是不合适的。在此示例中，指定大小为 10GB 和 15GB。

- **示例 2** – 下面的 **BACKUP** 命令指定只读文件和数据库空间：

```
BACKUP DATABASE READONLY DBSPACES dsp1
TO '/dev/rmt/0'
```

```
BACKUP DATABASE READONLY FILES dsp1_f1, dsp1_f2
TO 'bkp.f1f2'
```

```
BACKUP DATABASE READONLY DBSPACES dsp2, dsp3
READONLY FILES dsp4_f1, dsp5_f2
TO 'bkp.RO'
```

用法

当您执行 **BACKUP** 命令时，IQ 数据库可能会打开，以供多个读取器和写入器使用。它充当只读用户，并依赖 Sybase IQ 的表级版本控制功能 获得一致的数据集。

BACKUP 将在开始之前隐式发出 **CHECKPOINT**，然后备份描述数据库的 **catalog** 表（以及已添加到 **Catalog** 存储中的任何其它表）。在第一阶段，**Sybase IQ** 不允许对数据库的元数据进行任何更改（例如，添加或删除列和表）。相应地，接下来对备份执行 **RESTORE** 最多只能恢复到初始 **CHECKPOINT**。

BACKUP 命令用于指定完全备份或增量备份。您可以选择两种增量备份。

INCREMENTAL 只备份自上次通过 **BACKUP** 备份任何类型（增量或完全）后发生更改和已提交的那些块。**INCREMENTAL SINCE FULL** 备份自上次完全备份后已发生更改的所有块。对于 **BACKUP** 命令而言，第一种类型的增量备份较小且速度较快，但对于 **RESTORE** 命令而言，第一种类型的增量备份的速度却较慢并且更加复杂。而对另一种类型的增量备份来说，情况则相反。这是因为第一种类型的增量备份通常会导致对于每个完全备份存档需要执行 N 组增量备份存档。如果需要恢复，DBA 必须首先 **RESTORE** 完全备份存档，然后按照适当的顺序恢复每个增量存档。（Sybase IQ 会跟踪所需的存档。）第二种类型的增量备份要求 DBA 仅恢复完全备份存档和最后一个增量存档。

使用 **BACKUP** 语句的 **VIRTUAL DECOUPLED** 和 **VIRTUAL ENCAPSULATED** 参数可支持增量虚拟备份。

您可以对表空间执行 OS 级复制，从而为一个或多个只读数据库空间制作虚拟备份，但 Sybase 建议您使用虚拟备份语句，因为它会将备份记录在 IQ 系统表中。请参见《参考：构件块、表和过程》>“系统表和视图”>“系统视图”>“SYSIQBACKUPHISTORY 系统视图”以及《参考：构件块、表和过程》>“系统表和视图”>“系统视图”>“SYSIQBACKUPHISTORYDETAIL 系统视图”。

READWRITE FILES ONLY 可以与 **FULL**、**INCREMENTAL** 和 **INCREMENTAL SINCE FULL** 一起使用，限制仅针对数据库中的读写文件集执行备份。读写数据库空间/文件必须是 IQ 数据库空间。

如果 **READWRITE FILES ONLY** 与 **INCREMENTAL** 或 **INCREMENTAL SINCE FULL** 备份一起使用，则备份操作不会备份自先决备份后已发生更改的只读数据库空间或 dbfile 上的数据。如果 **INCREMENTAL** 或 **INCREMENTAL SINCE FULL** 备份中不指定 **READWRITE FILES ONLY**，则备份操作会备份自先决备份后已发生更改的读写数据库空间与只读数据库空间上的所有数据库页。

CRC 子句 - 除了硬件中提供的错误检测之外，还可对每个块激活 32 位循环冗余检查。如果您指定此子句，则系统将在任何后续 **RESTORE** 操作过程中对针对备份计算出的数字进行检验，从而影响上述两个命令的性能。缺省值为 **ON**。

ATTENDED 子句 - 仅在备份到磁带设备时适用。如果使用 **ATTENDED ON**（缺省值），当磁带驱动器需要干预时，系统将向发出 **BACKUP** 语句的应用程序发送一条消息。这在某些情况下会发生，比如在需要新磁带时。如果您指定了 **OFF**，则 **BACKUP** 不会提示需提供新磁带。如果需要其它磁带并且已指定了 **OFF**，Sybase IQ 会出现错误并中止 **BACKUP** 命令。但是，由于自动栈式存储器驱动器需要时间来切换磁带，因此存在一个短暂的延迟。

BLOCK FACTOR 子句 - 指定一次写入的块数。其值必须大于 0，否则 Sybase IQ 会生成错误消息。对于 **UNIX** 系统，其缺省值为 25，而对于 **Windows** 系统为 15（以容纳更小的固定磁带块大小）。此子句可有效地控制用于缓冲区的内存量。实际的内存量为此值乘以块大小再乘以用于从数据库中提取数据的线程数。Sybase 建议至少将 **BLOCK FACTOR** 设置为 25。

FULL 子句 - 指定完全备份；数据库中正在使用的所有块都将保存到存档设备。这是缺省操作。

INCREMENTAL 子句 - 指定增量备份；自上次执行任意类型的备份后发生更改的所有块都将保存到存档设备。

READONLY FILES 不允许使用关键字 **INCREMENTAL**。

INCREMENTAL SINCE FULL 子句 - 指定增量备份；自上次完全备份后发生更改的所有块都将保存到存档设备。

VIRTUAL DECOUPLED 子句 - 指定分离式虚拟备份。要完成备份，您必须在分离式虚拟备份完成之后复制 IQ 数据库空间，然后执行非虚拟增量备份。

VIRTUAL ENCAPSULATED 子句 - 指定封装式虚拟备份。“**shell-command**”参数可以是字符串，也可以是包含作为封装式虚拟备份一部分执行的字符串的变量。作为备份操作的一部分，**shell** 命令对 IQ 存储执行系统级备份。

TO 子句 - 指定要用于备份的 **archive_device** 的名称，并用单引号分隔。**archive_device** 是存档文件的文件名或磁带驱动器设备名。如果使用多个存档设备，请使用单独的 **TO** 子句指定它们。（不允许使用逗号分隔的列表。）存档设备必须是不同的。**TO** 子句的数量决定了与输出设备有关的并行 Sybase IQ 尝试的数量。

BACKUP 和 **RESTORE** 以并行方式在您指定的所有存档设备中写入或写出 IQ 数据。Catalog 存储以串行方式写入第一个设备。并行度越高，备份和恢复越快。

Sybase IQ 最多支持 36 台硬件设备用于备份。为提高备份速度，每核心指定一或两台设备将有助于避免硬件和 IO 争用。在 **BACKUP** 命令中设置 **SIZE** 参数可避免每台备份设备创建多个文件并考虑在 **BACKUP** 命令的 **BLOCK FACTOR** 子句中使用的值。

BACKUP 将覆盖现有存档文件，除非您移动了旧文件或使用其它的 *archive_device* 名称或路径。

备份 API DLL 实现可用于指定在打开存档设备时要传递给 DLL 的参数。对于第三方实现，*archive_device* 字符串具有以下格式：

```
'DLLIdentifier::vendor_specific_information'
```

特定示例：

```
'spsc::workorder=12;volname=ASD002'
```

archive_device 字符串的长度最长为 1023 字节。*DLLIdentifier* 部分的长度必须为 1 到 30 字节且只能包含字母数字和下划线字符。字符串的 *vendor_specific_information* 部分将传递给第三方实现，而不检查其内容。在使用第三方实现时不要指定 **BACKUP** 命令的 **SIZE** 或 **STACKER** 子句，因为在字符串的 *vendor_specific_information* 部分中应该已对此信息进行编码。

注意： 只有某些第三方产品才可用 Sybase IQ 通过使用此语法进行认证。有关其它用法说明或限制，请参见《发行公告》。在使用任意第三方产品以这种方式备份 Sybase IQ 数据库之前，请确保该产品已获得认证。请参见《发行公告》，或参见位于 <http://www.sybase.com/support/techdocs/> 的“技术文档”中 Sybase IQ 产品的 Sybase 认证报告。

对于备份 API 的 Sybase 实现，您只需要指定磁带设备名或文件名。对于磁盘设备，您还应指定 **SIZE** 值，否则 Sybase IQ 将假定所创建的每个磁盘文件在 UNIX 中不超过 2GB，在 Windows 中不超过 1.5GB。下例是为某些 UNIX 系统指定了磁带设备的 Sybase API DLL 的存档设备：

```
'/dev/rmt/0'
```

SIZE 子句 - 指定每个输出设备的磁带或文件的最大容量（某些平台不能可靠地检测磁带结束标记）。在相应设备上使用的卷的长度不应小于此值。此值适用于磁带和磁盘文件，但不适用于第三方设备。

大小单位为千字节 (KB)，但一般来说，小于 1GB 是不合适的。例如，对于 3.5GB 的磁带，指定 3500000。缺省值取决于平台和介质。备份文件的最终大小是不确定的，因为备份是以大数据块为单位写入的。

SIZE 参数是针对各输出设备而言。**SIZE** 不限制每个设备的字节数，**SIZE** 限制文件大小。每个输出设备可以具有不同的 **SIZE** 参数。

在备份过程中，当写入给定设备的信息量达到 **SIZE** 参数指定的值时，**BACKUP** 将执行下列操作之一：

- 如果设备属于文件系统设备，则 **BACKUP** 将关闭当前文件并创建另一个同名文件，并将按升序排列的下一个数附加到文件名，例如，bkup1.dat1.1、bkup1.dat1.2、bkup1.dat1.3。
- 如果设备属于磁带单元，则 **BACKUP** 将关闭当前磁带，您需要装入另一个磁带。

您有责任在需要时装入其它磁带，或者确保磁盘具有足够的空间来容纳备份。

指定多个设备时，**BACKUP** 会将信息分配到所有设备。

表 2. BACKUP 缺省大小

平台	磁带的缺省大小	磁盘的缺省大小
UNIX	无	2GB
Windows	1.5GB SIZE 必须是 64 的倍数。其它值则向下舍入为 64 的倍数。	1.5GB

STACKER 子句 - 指定自动装载设备，并指定随设备一同装载的磁带数。此值在栈式存储器中不表示磁带位置，可以为零。当 **ATTENDED** 为 OFF 而 **STACKER** 为 ON 时，Sybase IQ 将等待预定的时间，然后允许自动装载下一个磁带。随 **SIZE** 子句提供的磁带数用于确定是否具有足够的空间来存储备份数据。请不要对第三方介质管理设备使用此子句。

WITH COMMENT 子句 - 指定存档文件和备份历史文件中记录的可选注释。最大长度为 32KB。如果您未指定值，将存储 NULL 字符串。

有关 **BACKUP** 的其它问题包括：

- **BACKUP** 不支持将原始设备用作存档设备。
- Windows 系统仅支持对磁带设备执行固定长度的 I/O 操作（有关此限制的详细信息，请参见《安装和配置指南》）。虽然 Windows 支持磁带分区，但是 Sybase IQ 不会使用磁带分区功能，因此，请不要为 **BACKUP** 使用其它应用程序设置磁带格式。Windows 的磁带设备具有较简单的命名策略，其中第一个磁带设备为 `\\.|tape0`，第二个磁带设备为 `\\.|tape1`，依此类推。

警告！ 对于备份（以及对于大多数其它情况），当反斜杠位于 `n`、`x` 或其它反斜杠之前时，Sybase IQ 会将字符串中的前导反斜杠视为转义字符。因此，当指定备份磁带设备时，必须根据 Windows 命名约定将每个反斜杠变为双反斜杠。例如，将您要备份到的第一个 Windows 磁带设备表示为“`\\\\.\\tape0`”，第二个表示为“`\\\\.\\tape1`”，依此类推。如果您忽略额外的反斜杠，或错误拼写了磁带设备的名称，并在您的系统中写入了一个无效磁带设备的名称，则 Sybase IQ 会将此名称解释为磁盘文件名称。

- Sybase IQ 在使用磁带之前不会回绕磁带。必须确保用于 **BACKUP** 或 **RESTORE** 的磁带位于正确的起点，然后才能将其放置在磁带设备中。Sybase IQ 在磁带回绕设备上使用磁带之后将回绕磁带。

- 在 **BACKUP** 和 **RESTORE** 操作过程中，如果 Sybase IQ 无法打开存档设备（例如，当它需要装载的介质时）并且 **ATTENDED** 参数为 ON，它会等待十秒钟并重试。它将无限期地继续这些尝试，直至成功或通过 Ctrl+C 终止操作。
- 如果您按 Ctrl+C，**BACKUP** 将失败并使数据库返回到开始备份之前所处的状态。
- 如果在诸如 RAID 等设备上使用磁盘条带化，则分条设备将被视为单个设备。
- 如果您要恢复 SQL Anywhere 数据库，请参见“SQL Anywhere Server - 数据库管理”>“维护数据库”>“备份和数据恢复”以了解其它选项。

另请参见《系统管理指南第一卷》>“数据备份、恢复和存档”。

副作用：

- 自动提交

标准

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

权限

必须是数据库的所有者或者具有 DBA 权限。没有 DBA 权限的用户必须具有 OPERATOR 权限。

另请参见

- RESTORE 语句（第 262 页）

BEGIN……END 语句

将 SQL 语句组合到一起。

语法

```
[ statement-label : ]
... BEGIN [ [ NOT ] ATOMIC ]
... [ local-declaration ; ... ]
... statement-list
... [ EXCEPTION [ exception-case ... ] ]
... END [ statement-label ]
```

参数

- **local-declaration:** - { *variable-declaration* | *cursor-declaration* | *exception-declaration* | *temporary-table-declaration* }

- **variable-declaration:** - **DECLARE** *variable-name* [, ...] *data-type* [{ = | **DEFAULT** } *initial-value*]
- **initial-value:** - *special-value* | *string* | [-] *number* | (*constant-expression*) | *built-in-function* (*constant-expression*) | **NULL**
- **special-value:** - **CURRENT** { **DATABASE** | **DATE** | **PUBLISHER** | **TIME** | **TIMESTAMP** | **USER** | **UTC TIMESTAMP** } | **USER**

示例

- 示例 1 - 过程的主体是复合语句:

```
CREATE PROCEDURE TopCustomer (OUT TopCompany CHAR(35), OUT
TopValue INT)
BEGIN
    DECLARE err_notfound EXCEPTION FOR
        SQLSTATE '02000' ;
    DECLARE curThisCust CURSOR FOR
        SELECT CompanyName, CAST(
            sum(SalesOrderItems.Quantity *
                Products.UnitPrice) AS INTEGER) VALUE
        FROM Customers
            LEFT OUTER JOIN Salesorders
            LEFT OUTER JOIN SalesOrderItems
            LEFT OUTER JOIN Products
        GROUP BY CompanyName ;
    DECLARE ThisValue INT ;
    DECLARE ThisCompany CHAR(35) ;
    SET TopValue = 0 ;
    OPEN curThisCust ;

    CustomerLoop:
    LOOP
        FETCH NEXT curThisCust
            INTO ThisCompany, ThisValue ;
        IF SQLSTATE = err_notfound THEN
            LEAVE CustomerLoop ;
        END IF ;
        IF ThisValue > TopValue THEN
            SET TopValue = ThisValue ;
            SET TopCompany = ThisCompany ;
        END IF ;
    END LOOP CustomerLoop ;

    CLOSE curThisCust ;
END
```

用法

过程或触发器的主体是复合语句。过程或触发器内的控制语句也可以使用复合语句。

复合语句允许一个或多个 **SQL** 语句组合在一起，按一个单元处理。复合语句以 **BEGIN** 开头，以 **END** 结尾。紧接在 **BEGIN** 后，复合语句可以有仅存在于复合语句中的本地

声明。复合语句可以有变量、游标、临时表或异常的本地声明。本地声明可由该复合语句中的任何语句引用，或者可由该复合语句内嵌套的任何复合语句中的任何语句引用。本地声明对于从复合语句中调用的其它过程不可见。

如果指定结尾 *statement-label*，它必须与开头 *statement-label* 匹配。可以使用 **LEAVE** 语句在复合语句之后的第一个语句处继续执行。作为过程主体的复合语句具有与过程或触发器同名的隐式标签。

原子语句是完全执行或根本不执行的语句。例如，更新数千行的 **UPDATE** 语句在更新许多行后可能会遇到错误。如果语句未完成，则所有更改可能会恢复回其原始状态。同样，如果指定 **BEGIN** 语句为原子语句，则此语句要么完全执行，要么根本不执行。

如果指定了 **initial-value**，则该变量会设置为该值。如果未指定 **initial-value**，则该变量在 **SET** 语句分配另一个值之前包含的是空值。

如果您指定了 **initial-value**，则该数据类型必须与 *data-type* 定义的类型相匹配。

有关复合语句和异常处理的完整说明，请参见《系统管理指南第二卷》>“使用过程和批处理”。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 受 Adaptive Server Enterprise 支持。这并不意味着支持复合语句中的所有语句。

在 Transact-SQL 中不要求使用 **BEGIN** 和 **END** 关键字。

BEGIN 和 **END** 可用在 Transact-SQL 中以将一组语句组成一个复合语句，因此，诸如 **IF ... ELSE** 之类的仅影响单个 SQL 语句的控制语句会影响整个组的性能。**ATOMIC** 关键字不受 Adaptive Server Enterprise 的支持。

在 Transact-SQL 中，**DECLARE** 语句不必紧跟在 **BEGIN** 之后，并且声明的游标或变量在复合语句的持续时间内存在。为了兼容，应该在复合语句的开头声明变量。

权限

无

另请参见

- **DECLARE LOCAL TEMPORARY TABLE** 语句（第 161 页）
- **DECLARE CURSOR** 语句 [ESQL] [SP]（第 155 页）
- **LEAVE** 语句（第 221 页）
- **RESIGNAL** 语句（第 261 页）
- **SIGNAL** 语句（第 295 页）

BEGIN PARALLEL IQ……END PARALLEL IQ 语句

将 **CREATE INDEX** 语句组合在一起，以便同时执行。

语法

```
... BEGIN PARALLEL IQ
    statement-list
... END PARALLEL IQ
```

参数

- **statement-list** – **CREATE INDEX** 语句的列表

示例

- **示例 1** – 下面的语句将自动执行。如果其中有一个命令失败，则整个语句将回退：

```
BEGIN PARALLEL IQ
    CREATE HG INDEX c1_HG on table1 (col1);
    CREATE HNG INDEX c12_HNG on table1 (col12);
    CREATE LF INDEX c1_LF on table1 (col1);
    CREATE HNG INDEX c2_HNG on table1 (col2);
END PARALLEL IQ
```

用法

BEGIN PARALLEL IQ ... END PARALLEL IQ 语句允许您执行一组 **CREATE INDEX** 语句，就像它们是一个 DDL 语句似的，因而可同时对多个 IQ 表创建索引。执行此语句时，您和其他用户将无法发出其它 DDL 语句。

可以在语句列表中指定多个表。粒度处于列级。换句话说，将依次执行同一列的多个索引。

注意： 此语句不支持 **TEXT** 索引。

副作用：

- 自动提交

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。有关对该语句中所包含语句的支持，请参见“**CREATE INDEX 语句**”。

权限

无

另请参见

- CREATE INDEX 语句 (第 97 页)

BEGIN TRANSACTION 语句 [T-SQL]

使用此语句可开始用户定义的事务。

注意： **BEGIN TRANSACTION** 是 T-SQL 构造，必须仅包含有效的 T-SQL 命令。不能将 T-SQL 和非 T-SQL 命令混合在一起。

语法

```
BEGIN TRAN[ SACTION ] [ transaction-name ]
```

示例

- **示例 1** – 下面的批处理报告 @@trancount 的连续值：0、1、2、1、0。这些值显示在服务器窗口上：

```
PRINT @@trancount
BEGIN TRANSACTION
PRINT @@trancount
BEGIN TRANSACTION
PRINT @@trancount
COMMIT TRANSACTION
PRINT @@trancount
COMMIT TRANSACTION
PRINT @@trancount
```

@@trancount 的值只可用于跟踪已发出的显式 **BEGIN TRANSACTION** 语句的数量。

当 Adaptive Server Enterprise 隐式启动事务时，@@trancount 变量设置为 1。当隐式启动事务时，Sybase IQ 不会将 @@trancount 值设置为 1。因此，在任何 **BEGIN TRANSACTION** 语句前（即使存在当前事务），Sybase IQ @@trancount 变量的值均为 0，而在 Adaptive Server Enterprise（在链接模式下）中，该变量的值为 1。

对于由 **BEGIN TRANSACTION** 语句启动的事务，执行第一个 **BEGIN TRANSACTION** 语句后，在 Sybase IQ 和 Adaptive Server Enterprise 中，@@trancount 的值均为 1。如果事务由其它语句隐式启动，然后执行了 **BEGIN TRANSACTION** 语句，则执行 **BEGIN TRANSACTION** 语句后，在 Sybase IQ 和 Adaptive Server Enterprise 中，@@trancount 的值均为 2。

用法

可选参数 *transaction-name* 是分配给该事务的名称。它必须是一个有效的标识符。仅在最外面的一对嵌套的 **BEGIN/COMMIT** 或 **BEGIN/ROLLBACK** 语句中使用事务名称。

在事务内部执行 **BEGIN TRANSACTION** 语句时，该语句会将事务嵌套级别增加一级。通过 **COMMIT** 语句可减少嵌套级别。当事务被嵌套时，只有最外层的 **COMMIT** 可使数据库更改永久生效。

Adaptive Server Enterprise 和 Sybase IQ 均有两种事务 模式。

缺省的 Adaptive Server Enterprise 事务模式称为非链接模式，它单独提交每个语句，除非执行显式 **BEGIN TRANSACTION** 语句 启动事务。相反，与 ISO SQL/2003 兼容的链式模式仅当执行显式 **COMMIT** 时或执行能完成自动提交的语句（如数据定义语句）时才提交事务。

有关 **BEGIN TRANSACTION** 语句 [T-SQL] 的详细信息，请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “**BEGIN TRANSACTION** 语句 [T-SQL]”。

注意： 此参考指向 SQL Anywhere 文档。

通过设置链式数据库选项可以控制模式。在 Sybase IQ 中，ODBC 和嵌入式 SQL 连接的缺省设置为 On，在这种情况下，Sybase IQ 将在链式模式下运行。（ODBC 用户还应该检查 AutoCommit ODBC 设置。）TDS 连接 的缺省值是 Off。

在非链式模式中，事务是在任何数据检索或修改语句之前隐式启动的。这些语句包括：**DELETE**、**INSERT**、**OPEN**、**FETCH**、**SELECT** 和 **UPDATE**。但仍必须用 **COMMIT** 或 **ROLLBACK** 语句显式结束事务。

不能在事务中变更链式选项。

注意： 当调用存储过程时，应该确保它在要求的事务模式下正确运行。

当前嵌套级别保存在全局变量 @@trancount 中。在执行第一个 **BEGIN TRANSACTION** 语句前，@@trancount 变量的值为 0，而仅当 @@trancount 等于 1 时执行 **COMMIT** 才会对数据库进行永久更改。

没有事务名称或保存点名称的 **ROLLBACK** 语句总是将语句回退到最外层的 **BEGIN TRANSACTION**（显式或隐式）语句，并且取消整个事务。

标准

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

权限

无

另请参见

- COMMIT 语句 (第 63 页)
- ROLLBACK TRANSACTION 语句 [T-SQL] (第 274 页)
- SAVE TRANSACTION 语句 [T-SQL] (第 276 页)
- ISOLATION_LEVEL 选项 (第 386 页)

CALL 语句

调用过程。

语法

语法 1

```
[ variable = ] CALL procedure-name ( [ expression ] [ , ... ] )
```

语法 2

```
[ variable = ] CALL procedure-name ( [ parameter-name = expression ] [ , ... ] )
```

示例

- **示例 1** - 调用 sp_customer_list 过程。此过程没有参数，并返回结果集：

```
CALL sp_customer_list()
```

- **示例 2** - 此 dbisql 示例创建一个返回客户（提供了该客户的 ID）所下的订单数的过程，创建一个保存结果的变量，调用此过程并显示结果：

```
CREATE PROCEDURE OrderCount (IN CustomerID INT, OUT Orders INT)
BEGIN
  SELECT COUNT("DBA".SalesOrders.ID)
  INTO Orders
  FROM "DBA".Customers
  KEY LEFT OUTER JOIN "DBA".SalesOrders
  WHERE "DBA".Customers.ID = CustomerID ;
END
go
-- Create a variable to hold the result
CREATE VARIABLE Orders INT
go

-- Call the procedure, FOR customer 101
-- -----
CALL OrderCount ( 101, Orders)
go
-----
-- Display the result
SELECT Orders FROM DUMMY
go
```

用法

CALL 调用之前使用 **CREATE PROCEDURE** 语句创建的过程。过程完成后，将复制回任何 **INOUT** 或 **OUT** 参数值。

您可以通过位置或使用关键字格式指定参数列表。通过位置，参数与过程的参数列表中相对应的参数匹配。通过关键字，参数与命名参数匹配。

在 **CREATE PROCEDURE** 语句中，可以为过程参数分配缺省值，缺少的参数被分配缺省值。或者，如果未设置缺省值，则分配空值。

在过程内部，当过程返回结果集时，**CALL** 可以用在 **DECLARE** 语句中。

注意： 不能引用 **CALL SQL** 语句中的表 UDF。

使用 **RETURN** 语句，过程可以返回整数值（比方说作为状态指示符）。使用等号作为赋值运算符，可以将该返回值保存在变量中：

```
CREATE VARIABLE returnval INT ;
returnval = CALL proc_integer ( arg1 = val1, ... )
```

注意： 不建议使用此语句来调用函数。要调用函数，请使用赋值语句来调用函数，然后将其结果赋值给变量。例如：

```
DECLARE varname INT;
SET varname=test( );
```

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不受 Adaptive Server Enterprise 支持。有关支持的一种替代方法，请参见“EXECUTE 语句 [ESQL]”。

权限

必须是过程的所有者、具有过程的 **EXECUTE** 权限或具有 **DBA** 权限。

另请参见

- **CREATE PROCEDURE** 语句 （第 112 页）
- **EXECUTE** 语句 [ESQL] （第 182 页）
- **GRANT** 语句 （第 202 页）

CASE 语句

根据多种情况选择执行路径。

语法

```
CASE value-expression
...WHEN [ constant | NULL ] THEN statement-list ...
... [ WHEN [ constant | NULL ] THEN statement-list ] ...
...ELSE statement-list
... END
```

示例

- **示例 1** - 此过程使用 **CASE** 语句，将演示数据库的 **Products** 表中列出的产品分为衬衣、帽子、短裤、未知几大类：

```
CREATE PROCEDURE ProductType (IN product_id INT, OUT type
CHAR(10))
BEGIN
  DECLARE prod_name CHAR(20) ;
  SELECT name INTO prod_name FROM "GROUPO"."Products"
  WHERE ID = product_id;
  CASE prod_name
  WHEN 'Tee Shirt' THEN
    SET type = 'Shirt'
  WHEN 'Sweatshirt' THEN
    SET type = 'Shirt'
  WHEN 'Baseball Cap' THEN
    SET type = 'Hat'
  WHEN 'Visor' THEN
    SET type = 'Hat'
  WHEN 'Shorts' THEN
    SET type = 'Shorts'
  ELSE
    SET type = 'UNKNOWN'
  END CASE ;
END
```

用法

CASE 语句是控制语句，它允许您根据表达式的值选择要执行的 SQL 语句列表。

如果 *value-expression* 的值存在匹配的 **WHEN** 子句，则执行 **WHEN** 子句中的 *statement-list*。如果不存在合适的 **WHEN** 子句，但存在 **ELSE** 子句，则执行 **ELSE** 子句中的 *statement-list*。执行过程从 **END** 之后的第一个语句开始继续。

注意： ANSI 标准允许有两种形式的 **CASE** 语句。虽然 Sybase IQ 允许有两种形式，但是当 **CASE** 位于谓词中时，为获得最佳性能，您必须使用此处显示的形式。

如果需要其它形式（也称为 ANSI 语法）以便与 SQL Anywhere 兼容，请参见“SQL Anywhere Server - SQL 参考”>“SQL 语句”>“CASE 语句”中的 **CASE** 语句 语法 2。

注意： 不要混淆 **CASE** 语句与 CASE 表达式的语法。

有关 CASE 表达式的信息，请参见《参考：构件块、表和过程》>“SQL 语言元素”>“表达式”。

标准

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

权限

无

另请参见

- BEGIN……END 语句（第 49 页）

CHECKPOINT 语句

对数据库执行检查点操作。

语法

CHECKPOINT

用法

CHECKPOINT 强制数据库服务器执行检查点操作。数据库服务器也会根据内部算法自动执行检查点操作。通常，应用程序无需发出 **CHECKPOINT**。有关检查点操作的完整说明，请参见《系统管理指南第一卷》>“数据备份、恢复和存档”。

标准

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

权限

必须具有 DBA 或 OPERATOR 权限才能对数据库执行检查点操作。

CLEAR 语句 [Interactive SQL]

关闭 Interactive SQL (**dbisql**) 中任何打开的结果集。

语法

CLEAR

用法

关闭任何打开的结果集并保留 “SQL Statements” 窗格的内容不变。

副作用：

CLEAR 语句会关闭与清除的数据关联的游标。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不适用。

权限

无

另请参见

- EXIT 语句 [Interactive SQL] (第 187 页)

CLOSE 语句 [ESQL] [SP]

关闭游标。

语法

CLOSE *cursor-name*

参数

- **cursor-name:** - { *identifier* | *host-variable* }

示例

- **示例 1** - 关闭嵌入式 SQL 中的游标：

```
EXEC SQL CLOSE employee_cursor;  
EXEC SQL CLOSE :cursor_var;
```

• 示例 2 - 使用游标:

```
CREATE PROCEDURE TopCustomer (OUT TopCompany CHAR(35), OUT  
TopValue INT)  
BEGIN  
    DECLARE err_notfound EXCEPTION  
    FOR SQLSTATE '02000' ;  
    DECLARE curThisCust CURSOR FOR  
        SELECT CompanyName,  
            CAST(          sum(SalesOrderItems.Quantity *  
                Products.UnitPrice) AS INTEGER) VALUE  
        FROM Customers  
        LEFT OUTER JOIN SalesOrders  
        LEFT OUTER JOIN SalesOrderItems  
        LEFT OUTER JOIN Products  
    GROUP BY CompanyName ;  
    DECLARE ThisValue INT ;  
    DECLARE ThisCompany CHAR(35) ;  
    SET TopValue = 0 ;  
    OPEN curThisCust ;  
    CustomerLoop:  
    LOOP  
        FETCH NEXT curThisCust  
        INTO ThisCompany, ThisValue ;  
        IF SQLSTATE = err_notfound THEN  
            LEAVE CustomerLoop ;  
        END IF ;  
        IF ThisValue > TopValue THEN  
            SET TopValue = ThisValue ;  
            SET TopCompany = ThisCompany ;  
        END IF ;  
    END LOOP CustomerLoop ;  
    CLOSE curThisCust ;  
END
```

用法

此语句关闭指定的游标。

标准

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

权限

游标在此前必须已打开。

另请参见

- DECLARE CURSOR 语句 [ESQL] [SP] (第 155 页)

- OPEN 语句 [ESQL] [SP] (第 245 页)
- PREPARE 语句 [ESQL] (第 251 页)

COMMENT 语句

在系统表中存储关于数据库对象的注释。

语法

```
COMMENT ON
{
  COLUMN [ owner.]table-name.column-name
  DBSPACE dbspace-name
  EVENT event-name
  EXTERNAL ENVIRONMENT environment-name
  EXTERNAL OBJECT object-name
  FOREIGN KEY [owner.]table-name.role-name
  INDEX [ [owner.]table.]index-name
  INTEGRATED LOGIN integrated-login-id
  JAVA CLASS java-class-name
  JAVA JAR java-jar-name
  KERBEROS LOGIN "client-Kerberos-principal"
  LOGIN POLICY policy-name
  MATERIALIZED VIEW [owner.]materialized-view-name
  PROCEDURE [owner.]table-name
  SERVICE web-service-name
  TABLE [ owner.]table-name
  TRIGGER [[ owner.]table-name.]trigger-name
  USER userid
  VIEW [ owner.]view-name }
IS comment
```

参数

- **comment:** - { *string* | NULL }
- **environment-name:** - JAVA | PERL | PHP | CLR | C_ESQL32 | C_ESQL64 | C_ODBC32 | C_ODBC64

示例

- **示例 1** - 向 Employees 表添加注释:

```
COMMENT
ON TABLE Employees
IS "Employee information"
```

- **示例 2** - 从 Employees 表中删除注释:

```
COMMENT
ON TABLE Employees
IS NULL
```

用法

COMMENT 语句可用于为数据库中的对象设置注释。**COMMENT** 语句更新 **ISYSREMARK** 系统表中的注释。通过将注释设置为 **NULL** 可以删除注释。对于某个索引或触发器上的注释，注释所有者即是为其定义该索引或触发器的表的所有者。

COMMENT ON DBSPACE、**COMMENT ON JAVA JAR** 和 **COMMENT ON JAVA CLASS** 语句可用于在 **SYS.ISYSREMARK** 系统表中设置 **Remarks** 列。通过将注释设置为 **NULL** 可删除注释。

无法为本地临时表添加注释。

注意： 仅 IQ Catalog 存储中的 SQL Anywhere 表支持物化视图。

标准

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

权限

必须是要注释的数据库对象的所有者，或者必须有 **DBA** 权限。（必须具有 **DBA** 或 **SPACE ADMIN** 权限才能发出带有 **DBSPACE** 子句的语句。）

COMMENT ON LOGICAL SERVER 语句

用户定义的逻辑服务器上的注释。

语法

```
COMMENT ON LOGICAL SERVER logical-server-name IS 'comment'
```

示例

- **示例** – 以下示例创建有关用户定义的逻辑服务器 *ls1* 的注释。

```
COMMENT ON LOGICAL SERVER ls1 IS 'ls1: Primary  
Logical Server' ;
```

用法

仅适用于 **Multiplex**。

权限

必须具有 **DBA** 或 **MPX ADMIN** 权限。

COMMIT 语句

使数据库更改永久有效，或终止用户定义的事务。

语法

语法 1

```
COMMIT [ WORK ]
```

语法 2

```
COMMIT TRAN[SACTION ] [ transaction-name ]
```

示例

- 示例 1 - 提交当前事务：

```
COMMIT
```

- 示例 2 - 下面的 Transact-SQL 批处理报告 @@trancount 的连续值：0、1、2、1、0：

```
PRINT @@trancount
BEGIN TRANSACTION
PRINT @@trancount
BEGIN TRANSACTION
PRINT @@trancount
COMMIT TRANSACTION
PRINT @@trancount
COMMIT TRANSACTION
PRINT @@trancount
go
```

用法

语法 1 - **COMMIT** 语句将结束事务，并使在该事务进行期间所做的所有更改在数据库中均永久有效。

数据定义语句自动执行提交。有关信息，请参见每个 SQL 语句的“副作用”列表。

如果数据库服务器检测到任何无效的外键，**COMMIT** 将失败。这产生的结果是：不可能通过任何无效外键结束事务。通常，在每项数据处理操作中会检查外键的完整性。但是，如果数据库选项 `WAIT_FOR_COMMIT` 设置为 `ON`，或者使用 **CHECK ON COMMIT** 子句定义了具体的外键，则数据库服务器会将完整性检查推迟到执行 **COMMIT** 语句之后。

语法 2 - 可以成对地使用 **BEGIN TRANSACTION** 和 **COMMIT TRANSACTION** 语句来构造嵌套事务。嵌套事务类似于保存点。当在一组嵌套事务的最外层执行时，此语句使数据库更改永久有效。在事务内部执行时，**COMMIT TRANSACTION** 会将事务的嵌套级别降低一级。当事务被嵌套时，只有最外层的 **COMMIT** 可使数据库更改永久生效。

SQL 语句

可选参数 *transaction-name* 是分配给该事务的名称。它必须是一个有效的标识符。仅在最外面的一对嵌套的 **BEGIN/COMMIT** 或 **BEGIN/ROLLBACK** 语句中使用事务名称。

可以使用一组选项来控制 **COMMIT** 语句的行为细节。请参见“**COOPERATIVE_COMMIT_TIMEOUT** 选项”、“**COOPERATIVE_COMMITS** 选项”、“**DELAYED_COMMITS** 选项”和“**DELAYED_COMMIT_TIMEOUT** 选项”。可以使用 **Commit** 连接属性返回当前连接的提交数。

副作用：

- 关闭打开为 **WITH HOLD** 的游标之外的所有游标。
- 删除此连接上声明的临时表的所有行，除非这些表是用 **ON COMMIT PRESERVE ROWS** 声明的。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Adaptive Server Enterprise 支持。语法 2 是 ISO/ANSI SQL 语法的 Transact-SQL 扩展。

权限

必须连接到数据库。

另请参见

- **BEGIN TRANSACTION** 语句 [T-SQL] (第 53 页)
- **CONNECT** 语句 [ESQL] [Interactive SQL] (第 65 页)
- **DISCONNECT** 语句 [Interactive SQL] (第 169 页)
- **ROLLBACK** 语句 (第 272 页)
- **SAVEPOINT** 语句 (第 275 页)
- **SET CONNECTION** 语句 [ESQL] [Interactive SQL] (第 289 页)
- **COOPERATIVE_COMMIT_TIMEOUT** 选项 (第 353 页)
- **COOPERATIVE_COMMITS** 选项 (第 353 页)
- **DELAYED_COMMITS** 选项 (第 366 页)
- **DELAYED_COMMIT_TIMEOUT** 选项 (第 365 页)

CONFIGURE 语句 [Interactive SQL]

激活 Interactive SQL (**dbisql**) 配置窗口。

语法

CONFIGURE

用法

dbisql 配置窗口显示所有 **dbisql** 选项的当前设置。它不显示数据库选项，也不允许修改数据库选项。

如果选择“永久”，这些选项将被写入数据库的 **SYSOPTION** 表中，并且数据库服务器将自动执行 **COMMIT**。如果您不选择“永久”，而单击“确定”，则这些选项只是暂时设置的，它们仅在当前数据库连接时保持有效。

标准

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

权限

无

另请参见

- SET OPTION 语句（第 291 页）

CONNECT 语句 [ESQL] [Interactive SQL]

建立与数据库的连接。

语法

语法 1

```
CONNECT
... [ TO engine-name ]
...[ DATABASE database-name ]
...[ AS connection-name ]
...[ USER ] userid [ IDENTIFIED BY ]
```

语法 2

```
CONNECT USING connect-string
```

参数

- **engine-name:** - 标识符、字符串或宿主变量
- **database-name:** - 标识符、字符串或宿主变量
- **connection-name:** - 标识符、字符串或宿主变量
- **userid:** - 标识符、字符串或宿主变量
- **password:** - 标识符、字符串或宿主变量

- **connect-string**: - 一个有效的连接字符串或宿主变量

示例

- 示例 1 - 嵌入式 SQL 中的 **CONNECT** 用法:

```
EXEC SQL CONNECT AS :conn_name  
USER :userid IDENTIFIED BY :password;  
EXEC SQL CONNECT USER "dba" IDENTIFIED BY "sql";
```

- 示例 2 - **dbisql** 中的 **CONNECT** 用法:

- 通过 **dbisql** 连接到数据库。显示的提示要求输入用户 ID 和口令:

```
CONNECT
```

- 以 DBA 身份通过 **dbisql** 连接到缺省数据库。显示口令提示:

```
CONNECT USER "DBA"
```

- 以 DBA 身份通过 **dbisql** 连接到演示数据库:

```
CONNECT  
TO <machine>_iqdemo  
USER "DBA"  
IDENTIFIED BY sql
```

其中 *<machine>_iqdemo* 是引擎名称。

- 使用连接字符串通过 **dbisql** 连接到演示数据库:

```
CONNECT  
USING 'UID=DBA;PWD=sql;DBN=iqdemo'
```

用法

CONNECT 语句建立与数据库的连接，该数据库通过在 *engine-name* 所标识的服务器上运行的 *database-name* 来标识。

嵌入式 SQL 行为 - 在嵌入式 SQL 中，如果未指定 *engine-name*，则系统将采用缺省的本地数据库服务器（启动的第一个数据库服务器）。如果未指定 *database-name*，则系统将采用给定服务器上的第一个数据库。

WHENEVER 语句、**SET SQLCA** 和一些 **DECLARE** 语句不会生成代码，因此它们在源文件中可以出现在 **CONNECT** 语句之前。否则，在成功执行 **CONNECT** 语句之前，不允许执行任何语句。

用户 ID 和口令用于对所有动态 SQL 语句执行权限检查。缺省情况下，口令区分大小写；用户 ID 不区分大小写。

有关连接算法的详细说明，请参见《系统管理指南：第一卷>“Sybase IQ 连接”>“Sybase IQ 建立连接的方式”。

DBISQL 行为 - 如果在 **CONNECT** 语句中未指定任何数据库或服务器，**dbisql** 将保持与当前数据库（而不是与缺省的服务器和数据库）的连接。如果指定了数据库名称但

未指定服务器名，**dbisql** 会尝试连接到当前服务器上的指定数据库。必须指定 **-n** 数据库开关中定义的数据库名称，而不是数据库文件名。如果指定了服务器名但未指定数据库名称，**dbisql** 会连接到指定服务器上的缺省数据库。例如，如果与数据库连接时执行以下批处理语句，则将在同一个数据库中创建两个表。

```
CREATE TABLE t1( c1 int );
CONNECT DBA IDENTIFIED BY sql;
CREATE TABLE t2 (c1 int );
```

在成功执行 **CONNECT** 语句之前，不允许执行其它数据库语句。

用户 ID 和口令用于对 SQL 语句执行权限检查。如果未指定口令或用户 ID 和口令，系统将提示用户键入缺少的信息。缺省情况下，口令区分大小写；用户 ID 不区分大小写。

多个连接通过当前连接的概念进行管理。在成功执行连接语句后，新连接将变成当前连接。若要切换到其它连接，请使用 **SET CONNECTION**。执行 **CONNECT** 语句时，系统不会关闭现有连接（如果有）。使用 **DISCONNECT** 可以删除连接。

静态 SQL 语句使用通过 **SQLPP** 语句行上的 **-I** 选项指定的用户 ID 和口令。如果没有给出 **-I** 选项，则 **CONNECT** 语句的用户 ID 和口令也将用于静态 SQL 语句。

不使用口令进行连接 - 如果使用 **DBA** 权限连接到用户 ID，则无需指定口令即可以连接到其它用户 ID。（**dbtran** 的输出要求使用此功能。）例如，如果您将以 **DBA** 身份通过 **Interactive SQL** 连接到数据库，则可以使用以下语句在不提供口令的情况下连接到数据库：

```
CONNECT other_user_id
```

在嵌入式 SQL 中，可以在不输入口令的情况下进行连接，方法是对口令使用宿主变量，并将该宿主变量的值设置成空值指针。

AS 子句 - 通过指定 **AS** 子句可对连接进行命名（可选）。这允许建立到同一个数据库的多个连接，或者建立到同一个或不同的多个数据库服务器的多个连接，所有连接都是同时发生的。每个连接均有自己的关联事务。事务之间有时甚至会发生锁定冲突。例如，当试图从两个不同的连接修改同一数据库中的同一记录时，就会出现这种情况。

语法 2 - *connect-string* 是格式为 **keyword=value** 的参数设置的列表，必须用单引号引起来。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - Open Client 嵌入式 SQL 支持 **CONNECT** 语句的另一种语法。

权限

无

另请参见

- DISCONNECT 语句 [Interactive SQL] (第 169 页)
- GRANT 语句 (第 202 页)
- SET CONNECTION 语句 [ESQL] [Interactive SQL] (第 289 页)

CREATE DATABASE 语句

创建由多个操作系统文件组成的数据库。

语法

```
CREATE DATABASE db-name
... [ [ TRANSACTION ] { LOG ON [ log-file-name ]
    [ MIRROR mirror-file-name ] } ]
... [ CASE { RESPECT | IGNORE } ]
... [ PAGE SIZE page-size ]
... [ COLLATION collation-label[( collation-tailoring-string ) ] ]
... [ ENCRYPTED [ TABLE ] {algorithm-key-spec | OFF } ]
... { ... [ BLANK PADDING ON ]
... [ JCONNECT { ON | OFF } ]
... [ IQ PATH iq-file-name ]
... [ IQ SIZE iq-file-size ]
... [ IQ PAGE SIZE iq-page-size ]
... [ BLOCK SIZE block-size ]
... [ IQ RESERVE sizeMB ]
... [ TEMPORARY RESERVE sizeMB ]
... [ MESSAGE PATH message-file-name ]
... [ TEMPORARY PATH temp-file-name ]
... [ TEMPORARY SIZE temp-db-size ]
... [ DBA USER userid ]
... [ DBA PASSWORD password ]
```

参数

- db-name | log-file-name | mirror-file-name | iq-file-name | message-file-name | temp-file-name: - *'file-name'*
- page-size: - { 4096 | 8192 | 16384 | 32768 }
- iq-page-size: - { 65536 | 131072 | 262144 | 524288 }
- block-size: - { 4096 | 8192 | 16384 | 32768 }
- collation-label: - *string*
- collation-tailoring-string: - *keyword=value*
- algorithm-key-spec: - ON | [ON] KEY *key* [ALGORITHM *AES-algorithm*]
| [ON] ALGORITHM *AES-algorithm* KEY *key* | [ON] ALGORITHM 'SIMPLE'
- AES-algorithm: - 'AES' | 'AES256' | 'AES_FIPS' | 'AES256_FIPS'
- key: - *quoted string*

示例

- **示例 1** – 以下 Windows 示例创建了一个名为 mydb 的 Sybase IQ 数据库，其对应的 mydb.db、mydb.iq、mydb.iqtmp 和 mydb.iqmsg 文件位于 C:\sql\data 目录中：

```
CREATE DATABASE 'C:\\sql\\data\\mydb'
BLANK PADDING ON
IQ PATH 'C:\\sql\\data'
IQ SIZE 2000
IQ PAGE SIZE 65536
```

- **示例 2** – 以下 UNIX 命令将创建一个带有用于 **IQ PATH** 和 **TEMPORARY PATH** 的原始设备的 Sybase IQ 数据库。将应用大小为 128KB 的缺省 IQ 页大小。

```
CREATE DATABASE '/sql/data/bigdb'
IQ PATH '/dev/md/rdisk/bigdb'
MESSAGE PATH '/sql/data/bigdb.iqmsg'
TEMPORARY PATH '/dev/md/rdisk/bigtmp'
```

- **示例 3** – 以下 Windows 命令将创建一个带有用于 **IQ PATH** 的原始设备的 Sybase IQ 数据库。请注意，原始设备名中使用的两个反斜杠（Windows 要求）：

```
CREATE DATABASE 'company'
IQ PATH '\\\\.\\E:'
JCONNECT OFF
IQ SIZE 40
```

- **示例 4** – 以下 UNIX 示例将使用 AES 加密算法和密钥 “is!seCret” 创建强加密 Sybase IQ 数据库。

```
CREATE DATABASE 'marvin.db'
BLANK PADDING ON
CASE RESPECT
COLLATION 'ISO_BINENG'
IQ PATH '/filesystem/marvin.main1'
IQ SIZE 6400
IQ PAGE SIZE 262144
TEMPORARY PATH '/filesystem/marvin.temp1'
TEMPORARY SIZE 3200
ENCRYPTED ON KEY 'is!seCret' ALGORITHM 'AES'
```

用法

用提供的名称和属性创建 IQ 数据库。必须使用 **IQ PATH** 子句才能创建 Sybase IQ 数据库；否则，可创建标准的 SQL Anywhere 数据库。

如果省略 **IQ PATH** 选项，则指定以下任一选项将产生错误：**IQ SIZE**、**IQ PAGE SIZE**、**BLOCK SIZE**、**MESSAGE PATH**、**TEMPORARY PATH** 和 **TEMPORARY SIZE**。

当 Sybase IQ 创建 IQ 数据库时，将自动生成四个数据库文件以存储构成 IQ 数据库的不同类型的数据。每个文件都对应一个 dbspace，即 Sybase IQ 用来标识数据库文件的逻辑名。这些文件如下：

- **db-name.db** 为保存目录 dbspace **SYSTEM** 的文件。该文件包含用来描述数据库及任何添加的标准 SQL Anywhere 数据库对象的系统表和存储过程。如果您没有添

加 .db 扩展名, Sybase IQ 将添加它。此初始 dbspace 包含 Catalog 存储, 并且您可以稍后添加 dbspace 以增加其大小。无法在原始分区上创建该文件。

- *db-name.iq* 为用来保存主要数据数据库空间 IQ_SYSTEM_MAIN 的文件的缺省名, 其中包含 IQ 表和索引。可使用 **IQ PATH** 子句指定不同的文件名。此初始数据库空间包含 IQ 存储。

警告! IQ_SYSTEM_MAIN 是一个特殊数据库空间, 它包含打开数据库所必需的结构: IQ db_identity 块、IQ 检查点日志、每个已提交事务和每个活动检查点事务的 IQ 前滚/回退位图、增量备份位图和空闲列表根页。当打开数据库时, IQ_SYSTEM_MAIN 始终处于联机状态。

管理员可以允许在 IQ_SYSTEM_MAIN 中创建用户表, 特别是当这些表比较小且包含重要内容时。但是, 更常见的情况是在创建该数据库后, 管理员会立即创建另一个主数据库空间, 撤消所有用户在数据库空间 IQ_SYSTEM_MAIN 中的 create 特权, 向选定用户授予对新的主数据库空间的 create 特权, 并将 PPUBLIC.default_dbspace 设置为新的主数据库空间。

- *db-name.iqtmp* 为用来保存初始临时数据库空间 IQ_SYSTEM_TEMP 的文件的缺省名。该文件包含特定查询生成的临时表。此文件所需的大小可根据查询类型和数据量的不同而有所不同。您可以使用 **TEMPORARY PATH** 子句指定不同的名称。此初始数据库空间包含临时存储。
- *db-name.iqmsg* 为包含消息跟踪数据库空间 IQ_SYSTEM_MSG 的文件的缺省名。您可以使用 **MESSAGE PATH** 子句指定不同的文件名。

除这些文件之外, IQ 数据库还拥有事务日志文件 (db-name.log), 并且还可能拥有一个事务日志镜像文件。

文件名和 **CREATE DATABASE** 语句:

文件名 (*db-name*、*log-file-name*、*mirror-file-name*、*iq-file-name*、*message-file-name*、*temp-file-name*) 是包含操作系统文件名的字符串。作为文字字符串, 它们必须用单引号引起来。

- 在 Windows 中, 如果您指定一个路径, 则任何后跟 n 或 x 的反斜杠字符 (\) 都必须双写。这样可防止根据 SQL 中的字符串规则将这些反斜杠解释为换行符 (\n) 或十六进制数 (\x)。更安全的做法是始终使用两个反斜杠。例如:

```
CREATE DATABASE 'c:\\sybase\\mydb.db' LOG ON 'e:\\logdrive\\mydb.log' JCONNECT OFF IQ PATH 'c:\\sybase\\mydb' IQ SIZE 40
```

- 如果未指定路径或指定了相对路径:
 - Catalog 存储文件 (*db-name.db*) 创建在服务器工作目录的相对目录下。
 - IQ 存储、临时存储和消息日志文件都创建在 Catalog 存储所在的目录下或其相对目录下。

建议使用相对路径名。

警告! 数据库文件、临时数据库空间和事务日志文件都必须位于与数据库服务器相同的物理计算机中。不要将数据库文件和事务日志文件放在网络驱动器上。然而, 事务日志应位于与其镜像不同的设备上。

在 UNIX 系统中，您可以创建符号链接，此类链接为间接指针，其中包含所指文件的路径名。可以使用符号链接作为相对路径名。为数据库文件名创建符号链接具有多个优点：

- 指向原始设备的符号链接可以具有有意义的名称，而实际设备名语法可能模糊不清。
- 使用符号名称可能会消除在恢复自备份后已移至新目录的数据库文件时存在的问题。

要创建符号链接，请使用 **ln -s** 命令。例如：

```
ln -s /disk1/company/iqdata/company.iq company_iq_store
```

创建该链接后，您可以在诸如 **CREATE DATABASE** 或 **RESTORE** 之类的命令中指定符号链接而不是完全限定路径名。

创建数据库或数据库空间时，每个数据库空间文件的路径必须是唯一的。如果您的 **CREATE DATABASE** 命令为这两种存储指定了相同的路径名和文件名，您将收到一个错误。

注意： 要创建 **Multiplex** 数据库，请参见《使用 Sybase IQ Multiplex》。

可以用下列任何一种方式创建一个唯一的路径：

- 为每个文件指定一个不同的扩展名（如 **mydb.iq** 和 **mydb.iqtmp**）
- 指定一个不同的文件名（如 **mydb.iq** 和 **mytmp.iq**）
- 指定一个不同的路径名（如 **/iqfiles/main/iq** 和 **/iqfiles/temp/iq**）或不同的原始分区
- 创建数据库时省略 **TEMPORARY PATH**。在此情况下，系统将在与 **Catalog** 存储相同的路径下创建临时存储，缺省名和扩展名为 **dbname.iqtmp**，其中 **dbname** 为数据库名称。

警告！ 在 UNIX 平台上，要保持数据库的一致性，则必须指定链接到不同文件的文件名。Sybase IQ 无法检测到链接文件指向的目标。即使命令中的文件名不同，也应确保它们不会指向同一个操作系统文件。

CREATE DATABASE 的子句和选项：

TRANSACTION LOG - 事务日志是数据库服务器用来记录所有数据库更改的文件。事务日志在系统恢复中起着重要的作用。如果不指定任何 **TRANSACTION LOG** 子句，或者如果您省略文件名的路径，则事务日志将位于与 **.db** 文件相同的目录中。但是，应该将它放在不同于 **.db** 和 **.iq** 的物理设备上。无法在原始分区上创建该文件。

MIRROR - 事务日志镜像是与事务日志完全相同的一个副本，通常在单独的设备上进行维护，以便更好地保护您的数据。缺省情况下，Sybase IQ 不使用镜像事务日志。如果您不想使用事务日志镜像，则必须提供文件名。如果使用相对路径，则系统将创建相对于 **Catalog** 存储 (**db-name.db**) 所在目录的事务日志镜像。Sybase 建议您始终创建事务日志的镜像副本。

CASE - 对使用 **CASE RESPECT** 创建的数据库而言，在比较和字符串操作中，所有受影响的值都区分大小写。数据库对象名称（如列、过程或用户 ID）则不受影响。数据库空间名称始终不区分大小写，无论 **CASE** 规范如何。

缺省值(**RESPECT**)为所有比较均区分大小写。**CASE RESPECT**所提供的性能要比**CASE IGNORE**好一些。

插入到表中的字符串始终按输入时的大小写存储，而不管数据库是否区分大小写。如果将字符串 **Value** 插入到字符数据类型的列中，该字符串在数据库中始终以大写 **V** 和其余字母小写的形式存储。**SELECT** 语句返回字符串作为 **Value**。但是，如果数据库不区分大小写，所有比较都会使 **Value** 与 **value**、**VALUE** 等相同。**IQ** 服务器可以大小写字母的任意组合返回结果，因此，不区分大小写的数据库中的结果不能区分大小写 (**CASE IGNORE**)。

例如，假定有以下表和数据：

```
CREATE TABLE tb (id int NOT NULL,
                  string VARCHAR(30) NOT NULL);
INSERT INTO tb VALUES (1, 'ONE' );
SELECT * FROM tb WHERE string = 'oNe' ;
```

SELECT 的结果可为 “oNe”（如在 **WHERE** 子句中所指定的那样），而不一定为 “ONE”（如在数据库中所存储的那样）。

类似地，

```
SELECT * FROM tb WHERE string = 'One' ;
```

的结果可以为 “One” ，

```
SELECT * FROM tb WHERE string = 'ONE' ;
```

的结果可以为 “ONE” 。

在创建所有数据库时至少使用一个用户 ID：

```
DBA
```

以及口令：

```
sql
```

在新数据库中，所有口令都区分大小写，无论数据库是否区分大小写均是如此。用户 ID 不受 **CASE RESPECT** 设置的影响。

PAGE SIZE - 数据库（包含 catalog 表）的 SQL Anywhere 段的 页大小可以为 4096、8192、16384 或 32768 字节。通常，使用缺省值 4096 (4KB)。大型数据库可能需要比缺省大小更大的页大小 以便提高性能。较小的值可能会限制 数据库能够支持的列数。如果指定一个小于 4096 的页大小，则 Sybase IQ 将使用 4096 的页大小。

启动数据库时，其页大小不能大于当前服务器的页大小。服务器页大小可从启动的第一组数据库中获取，或者在服务器命令行上使用 **-gp** 命令行选项来设置。

任何语句的命令行长度都限制为目录页大小。在大多数情况下，4KB 缺省值已足够用；但是，在某些情况下，为了容纳非常长的命令（如引用多个数据库空间的 **PAGE SIZE** 命令），需要更大的 **RESTORE** 值。此外，执行涉及大量表或视图的查询时，也可能需要较大的页大小。

由于缺省目录页大小为 4KB，因此仅当连接指向 `utility_db` 等数据库（页大小为 1024）时才会出现问题。此限制可能会导致引用大量数据库空间的 **RESTORE** 命令失败。为避免此问题，请确保 **SQL** 命令行长度小于目录页大小。

或者，使用 `-gp 32768` 启动引擎以增大目录页大小。

COLLATION - 用于在数据库中对字符数据类型进行排序和比较的归类序列。归类为正在使用的编码（字符集）提供字符比较和排序信息。如果未指定 **COLLATION** 子句，**Sybase IQ** 便会根据操作系统语言和编码选择一个归类。

对大多数操作系统来说，缺省归类序列为 **ISO_BINENG**，该序列可提供最佳性能。在 **ISO_BINENG** 中，归类顺序与 **ASCII** 字符集中的字符顺序相同。所有大写字母均位于所有小写字母之前（例如，“A”和“B”均位于“a”的前面）。

可以从支持归类的列表中选择归类。对于在 **Sybase IQ** 服务器上创建的 **SQL Anywhere** 数据库，归类也可以是 **Unicode** 归类算法 (**UCA**)。如果已指定 **UCA**，则还应指定 **ENCODING** 子句。有关 **ENCODING** 子句的详细信息，请参见“**SQL Anywhere Server - SQL 参考**”>“**SQL 语句**”>“**CREATE DATABASE 语句**”。

Sybase IQ 不支持 **IQ** 数据库的任何基于 **UCA** 的归类。如果在 **IQ** 数据库的 **CREATE DATABASE** 语句中指定基于 **UCA** 的归类，则服务器将返回错误“不支持 **UCA** 归类”且数据库创建将会失败。

请务必小心选择您的归类，创建数据库后不能更改归类。有关选择归类的信息，请参见《系统管理指南第一卷》>“国际语言和字符集”。

或者，您还可以指定归类定制选项 (*collation-tailoring-string*)，以便能够对字符的排序和比较进行更多的控制。这些选项采取“关键字=值对”的形式，在归类名称后以括号括起。

注意： 当为在 **Sybase IQ** 服务器中创建的 **SQL Anywhere** 数据库指定 **UCA** 归类时，支持若干归类定制选项。对于所有其它归类和 **Sybase IQ** 来说，只支持区分大小写的定制。同样，使用归类定制选项创建的数据库无法使用 15.0 版之前的数据库服务器启动。

Sybase IQ 的归类定制选项包含 **Sybase IQ** 数据库支持的 关键字、允许的替代表单 以及归类定制选项 (*collation-tailoring-string*) 允许 的值。

表 3. Sybase IQ 的归类 定制选项

关键字	归类	替代表单	允许值
CaseSensitivity	所有支持的归类	CaseSensitive, Case	<ul style="list-style-type: none">• respect 区分字母间的大小写差异。对于 UCA 归类，这相当于 UpperFirst。对于其它归类，respect 的值取决于归类自身。• ignore 忽略字母间的大小写差异。• UpperFirst 始终将大写排在前面 (Aa)。• LowerFirst 始终将小写排在前面 (aA)。

有关为 SQL Anywhere 数据库指定 UCA 归类时的语法及所支持的归类定制 选项的完整列表，请参见“SQL Anywhere Server - SQL 参考”>“SQL 语句”>“CREATE DATABASE 语句”。

注意： 此参考指向 SQL Anywhere 文档。

ENCRYPTED - 加密可使存储在物理 数据库文件中的数据变得无法阅读。 当对整个数据库进行加密时， 请使用 **CREATE DATABASE ENCRYPTED** 关键字且 不包含 **TABLE** 关键字。当只想对 SQL Anywhere 表启用表加密时， 请使用 **ENCRYPTED TABLE** 子句。 Sybase IQ 表不支持表级 加密。 启用表加密意味着 随后使用 **ENCRYPTED** 子句创建或更改的表将 使用创建数据库时指定的设置进行加密。

有两个级别的数据库和表加密：简单加密和高度加密。

- 简单加密等效于模糊处理。虽然这些数据无法阅读，但了解加密技术的人可能会破解数据。若要进行简单加密，请指定 **CREATE DATABASE** 子句 **ENCRYPTED ON ALGORITHM 'SIMPLE'**、**ENCRYPTED ALGORITHM 'SIMPLE'** 或指定 **ENCRYPTED ON** 子句而不指定算法或密钥。
- 高度加密可通过使用 128 位算法和安全密钥实现。如果没有密钥，数据将不可读取，而且实际上不可破译。对于高度加密，请指定使用 128 位或 256 位 AES 算法的 **CREATE DATABASE** 子句 **ENCRYPTED ON ALGORITHM**，并指定 **KEY** 子句来指定加密密钥。应为密钥选择的值长度至少为 16 个字符，同时包含大写和小写字符，且包括数字、字母和特殊字符。
每次启动数据库时，都需要此加密密钥。

警告！ 请保护好您的加密密钥！将密钥副本存储在安全位置。丢失了密钥将导致数据库完全无法访问，而无法访问的数据库是无法进行恢复的。

只能在数据库创建期间指定加密。对现有数据库进行加密需要完全卸载、重新创建数据库以及重新装载所有数据。

如果使用 **ENCRYPTED** 子句但未指定算法，则缺省算法为 AES。缺省情况下，加密处于关闭状态。

BLANK PADDING—缺省情况下，系统将忽略用于比较用途的尾随空白 (BLANK PADDING ON)，并且嵌入式 SQL 程序将填充读入字符数组中的字符串。提供此选项是为了与 ISO/ANSI SQL 标准兼容。

例如，以下两个字符串在使用 `BLANK PADDING ON` 创建的数据库中视为相等：

```
'Smith'
'Smith  '
```

注意： `CREATE DATABASE` 不再支持 `BLANK PADDING OFF`。

`JCONNECT`—要使用 Sybase jConnect for JDBC 驱动程序来访问系统目录信息，请安装 jConnect 支持。将 `JCONNECT` 设置为 `OFF` 可排除 jConnect 系统对象（缺省设置为 `ON`）。只要不访问系统信息，您就仍然可以使用 JDBC。

`IQ PATH`—包含 Sybase IQ 数据的主要段文件的路径名。可以指定操作系统文件或 I/O 设备的原始分区。（针对所用平台的《安装和配置指南》说明了如何指定原始分区的格式。）Sybase IQ 会自动检测基于指定路径名的类型。如果使用相对路径，则系统将创建相对于 Catalog 存储（.db 文件）所在目录的文件。

`IQ SIZE`—通过 `IQ PATH` 子句指定的原始分区或操作系统文件的大小（以 MB 为单位）。对于原始分区，应不指定 `IQ SIZE` 而始终采用缺省值，使用缺省值可允许 Sybase IQ 使用整个原始分区；如果为 `IQ SIZE` 指定一个值，则该值必须匹配 I/O 设备的大小，否则 Sybase IQ 将返回一个错误。对于操作系统文件，您可以指定以下表中从最小值到最大 4TB 之间的某个值。

操作系统文件的缺省大小取决于 `IQ PAGE SIZE`：

表 4. IQ 存储文件和临时存储文件的缺省大小和最小大小

IQ PAGE SIZE	IQ SIZE 缺省值	TEMPORARY SIZE 缺省值	最小显式 IQ SIZE	最小显式 TEMPORARY SIZE
65536	4096000	2048000	4MB	2MB
131072	8192000	4096000	8MB	4MB
262144	16384000	8192000	16MB	8MB
524288	32768000	16384000	32MB	16MB

`IQ PAGE SIZE` —数据库的 Sybase IQ 段（包含 IQ 表和索引）的页大小（字节）。该值必须为 2 的幂，从 65536 到 524288 字节。缺省值为 131072 (128KB)。页大小的其它值都将被更改为下一个更大的大小。`IQ` 页大小确定了缺省的 I/O 传输块大小和数据库的最大数据压缩。

为获得最佳性能，Sybase 推荐采用以下最小 `IQ` 页大小：

- 64KB (**IQ PAGE SIZE 65536**)，针对其最大表不超过 10 亿行的数据库，或总大小小于 8TB 的数据库。这是新数据库的绝对最小值。在 32 位平台上，`IQ` 页大小设置为 64KB 可实现最佳的性能。
- 128KB (**IQ PAGE SIZE 131072**)，针对位于 64 位平台上其最大表所含行数超过 10 亿且少于 40 亿的数据库，或总大小可能增至 8TB 或更大的数据库。128KB 为缺省 `IQ` 页大小。

- **256KB (IQ PAGE SIZE 262144)**，针对位于 64 位平台上其最大表所含行数超过 40 亿 的数据库，或总大小可能增加至 8TB 或更大的数据库。

非常宽的表（如带有多列宽 VARCHAR 数据的表 [列大小从 255 到 32,767 字节]）可能需要下一个更大的 **IQ PAGE SIZE**。

BLOCK SIZE—数据库的 Sybase IQ 段的 I/O 传输块大小（以字节为单位）。该值必须小于 **IQ PAGE SIZE**，并且必须为介于 4096 到 32768 之间的 2 的幂。该大小的其它值将更改为 下一个更大的大小。缺省值取决于 **IQ PAGE SIZE** 子句 的值。对于大多数应用程序来说，此缺省值为最佳设置。指定不同值之前，请参见《性能和调优指南》>“管理系统资源”。

IQ RESERVE—指定要为主 IQ 存储（**IQ_SYSTEM_MAIN** 数据库空间）保留的空间大小（以 MB 为单位），以便将来可以增加该 dbfile 的大小。*sizeMB* 参数可以为大于 0 的任意数字。数据库空间创建后，保留大小将无法更改。

指定 **IQ RESERVE** 后，数据库将更多空间用于内部（空闲列表）结构。如果保留大小过大，则内部结构所需的空间可能会大于指定大小，这会导致错误。

IQ RESERVE—指定要为临时 IQ 存储（**IQ_SYSTEM_TEMP** 数据库空间）保留的空间大小（以 MB 为单位），以便将来可以增加该 dbfile 的大小。*sizeMB* 参数可以为大于 0 的任意数字。数据库空间创建后，保留大小将无法更改。

指定 **TEMPORARY RESERVE** 后，数据库将更多空间用于内部（空闲列表）结构。如果保留大小过大，则内部结构所需的空间可能会大于指定大小，这会导致错误。

注意： 如果从备份恢复数据库，则临时数据库空间的保留大小和模式将丢失。

MESSAGE PATH—包含 Sybase IQ 消息跟踪文件的段 的路径名。您必须指定操作系统文件；消息文件不能位于原始分区上。如果使用相对路径或省略该路径，则系统将创建 相对于 .db 文件所在目录的消息文件。

TEMPORARY PATH—临时段文件的路径名，该文件包含特定查询生成的临时表。可以指定操作系统文件 或 I/O 设备的原始分区。（针对所用平台的《安装和配置指南》说明了 如何指定原始分区的格式。）Sybase IQ 会自动检测基于指定 路径名的类型。如果使用相对路径或省略该路径，则系统将创建 相对于 .db 文件所在目录的消息文件。

TEMPORARY SIZE—通过 **TEMPORARY PATH** 子句 指定的原始分区或操作系统文件的大小（以 MB 为单位）。对于原始分区，应不指定 **TEMPORARY SIZE** 而始终采用缺省值，使用缺省值可允许 Sybase IQ 使用整个原始分区。操作系统 文件的缺省值始终为 **IQ SIZE** 的值的一半。如果 IQ 存储位于原始分区上且 临时存储为操作系统文件，则缺省的 **TEMPORARY SIZE** 为 IQ 存储原始 分区大小的一半。

DBA USER—具有 DBA 权限的缺省用户帐户的用户名。如果您不指定此子句，Sybase IQ 会创建 缺省的 dba 用户 ID。

DBA PASSWORD—具有 DBA 权限的缺省用户帐户的口令。

副作用：

- 自动提交

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase — Adaptive Server Enterprise 提供了一个 **CREATE DATABASE** 语句，但具有不同选项。

权限

执行此语句所需的权限是在服务器命令行上使用 **-gu** 选项设置的。缺省设置为要求具有 DBA 权限。

运行服务器时所使用的帐户必须对创建文件的目录具有写权限。

另请参见

- CREATE DBSPACE 语句 (第 77 页)
- DROP DATABASE 语句 (第 173 页)

CREATE DBSPACE 语句

为 IQ main 存储或目录存储创建一个新的数据库空间及关联 dbfile。

语法

语法 1

仅用于 Catalog 存储 dbspace (SQL Anywhere (SA) dbspace)。

```
CREATE DBSPACE dbspace-name AS file-path CATALOG STORE
```

语法 2

用于 IQ 数据库空间。

```
CREATE DBSPACE dbspace-name USING file-specification  
[ IQ STORE ] iq-dbspace-opts
```

参数

- **file-specification:** - { *single-path-spec* | *new-file-spec* [, ...] }
- **single-path-spec:** - '*file-path*' | *iq-file-opts*
- **new-file-spec:** - **FILE** *logical-file-name* | '*file-path*' *iq-file-opts*
- **iq-file-opts:** - [[**SIZE**] *file-size*] ... [**KB** | **MB** | **GB** | **TB**]] [**RESERVE** *size* ... [**KB** | **MB** | **GB** | **TB**]]
- **iq-dbspace-opts:** - [**STRIPING**] { **ON** | **OFF** }] ... [**STRIPESIZE** *KBsizeKB*]

示例

- **示例 1** - 为 UNIX 系统上包含两个文件的 IQ 主存储区创建一个称为 DspHist 的数据库空间。每个文件大小为 1GB 并可扩大 500MB:

```
CREATE DBSPACE DspHist USING FILE
FileHist1 '/History1/data/file1'
SIZE 1000 RESERVE 500,
FILE FileHist2 '/History1/data/file2'
SIZE 1000 RESERVE 500;
```

- **示例 2** - 创建名为 DspCat2 的另一个目录数据库空间:

```
CREATE DBSPACE DspCat2 AS
'catalog_file2'
CATALOG STORE;
```

- **示例 3** - 为 IQ 存储创建名为 EmpStore1 的 IQ 主数据库空间（三种替代语法示例）:

```
CREATE DBSPACE EmpStore1
USING FILE EmpStore1
'EmpStore1.IQ' SIZE 8 MB IQ STORE;
```

```
CREATE DBSPACE EmpStore1
USING FILE EmpStore1
'EmpStore1.IQ' 8 IQ STORE;
```

```
CREATE DBSPACE EmpStore1
USING FILE EmpStore1
'EmpStore1.IQ' 8;
```

用法

CREATE DBSPACE 为 IQ 主存储区或 Catalog 存储创建一个新的数据库空间。您添加的数据库空间可以位于与初始数据库空间不同的磁盘设备上，允许创建大于一个物理设备的存储。

Syntax 1 为目录存储创建一个数据库空间，在该存储中数据库空间和 dbfile 具有相同的逻辑名称。该目录存储中的每个数据库空间都具有一个文件。

new-file-spec 为 IQ main 存储创建一个数据库空间。可以为 IQ main 存储指定一个或多个 dbfile。每个文件都需要 dbfile 名称和物理文件路径且必须是唯一的。

Dbspace 名称和 dbfile 名称始终都不区分大小写。如果数据库为 **CASE RESPECT**，则物理文件路径具有操作系统的区分大小写特性；如果数据库为 **CASE IGNORE**，则物理文件路径不区分大小写。

无法为 IQ 临时存储创建 dbspace。当创建一个新数据库或升级一个 Sybase IQ 15.3 之前版本的数据库时，将创建一个临时 dbspace IQ_SYSTEM_TEMP。可以使用 **ALTER DBSPACE ADD FILE** 语法将其它文件添加到 IQ_SYSTEM_TEMP dbspace 中。

RESERVE 子句—以千字节 (KB)、兆字节 (MB)、千兆字节 (GB) 或千吉字节 (TB) 为单位指定要保留的空间大小，从而可在将来增加数据库空间的大小。*size* 参数可为大于 0 的任何数字；缺省值为兆字节。创建数据库空间 dbfile 后不能更改保留大小。

指定 **RESERVE** 后，数据库将更多空间用于内部（空闲列表）结构。如果保留大小过大，则内部结构所需的空間可能会大于指定大小，这会导致错误。

有关缺省情况下创建的文件的名称和类型，请参见“**CREATE DATABASE 语句**”。

注意： 有关为 **Multiplex** 数据库创建 **dbspace** 的信息，请参见《使用 Sybase IQ **Multiplex**》。

可以用下列任何一种方式创建一个唯一的路径：

- 为每个文件指定一个不同的扩展名（如 **mydb.iq**）
- 指定一个不同的文件名（如 **mydb2.iq**）
- 指定一个不同的路径名（如 **/iqfiles/main/iq**）或不同的原始分区

警告！ 在 **UNIX** 平台上，要保持数据库的一致性，请指定链接到不同文件的文件名。**Sybase IQ** 无法检测到链接文件指向的目标。即使命令中的文件名不同，也应确保它们不会指向同一个操作系统文件。

dbspace-name 和 **dbfile-name** 是数据库空间和 **dbfile** 的内部名称。**filepath** 是 **dbfile** 的实际操作系统文件名，有时需要在前面添加路径。在数据库的目录存储所在的目录中创建不包含显式目录的 **filepath**。任何相对目录都是相对于该目录存储而言。

SIZE 子句—指定您在 **filepath** 中指定的操作系统文件的大小（从 0 至 4 千吉字节）。缺省值取决于存储类型和块大小。对于 **IQ main** 存储，缺省字节数等于块大小乘以 1000。不能为目录存储指定 **SIZE** 子句。

值为 0 的 **SIZE** 将创建具有最小大小的数据库空间，对 **IQ main** 存储来说，该大小为 8MB。

对于原始分区来说，不要显式指定 **SIZE**。**Sybase IQ** 将此参数自动设置为最大原始分区大小，如果尝试指定另一大小，将返回一个错误。

STRIPESIZEKB 子句 - 指定磁盘条带化算法移至指定数据库空间的下一条带之前要写入每个文件的千字节数 (**KB**)。

如果不指定条带化大小或带区大小，则应用选项 **DEFAULT_DISK_STRIPING** 和 **DEFAULT_KB_PER_STRIPE** 的缺省值。

一个数据库最多可具有 (32KB - 1) 个数据库空间，其中包括在创建数据库时创建的初始数据库空间。但是，您的操作系统可能会限制每个数据库的文件数。

另请参见《系统管理指南第一卷》>“数据库对象管理”。

副作用：

- 自动提交。
- 自动检查点。

标准

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

权限

必须具有 DBA 或 SPACE ADMIN 权限。

另请参见

- CREATE DATABASE 语句（第 68 页）
- DROP 语句（第 170 页）

CREATE DOMAIN 语句

在数据库中创建用户定义数据类型。

语法

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

参数

- **domain-name:** - 标识符
- **data-type:** - 具有精度和小数位数的内置数据类型
- **default-value:** - *special-value* | *string* | *global variable* | [-] *number* | (*constant-expression*) | *built-in-function*(*constant-expression*) | **AUTOINCREMENT** | **CURRENT DATABASE** | **CURRENT REMOTE USER** | **NULL** | **TIMESTAMP** | **LAST USER**
- **special-value:** - **CURRENT** { **DATE** | **TIME** | **TIMESTAMP** | **USER** | **PUBLISHER** } | **USER**

示例

- **示例 1** - 创建名为 **address** 的数据类型，它包含 35 个字符长的字符串并且可以为空值：

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

用法

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

Sybase 建议您使用 **CREATE DOMAIN** 而不是 **CREATE DATATYPE**，因为 **CREATE DOMAIN** 是 ANSI/ISO SQL3 术语。

创建数据类型的用户自动成为该数据类型的所有者。在 **CREATE DATATYPE** 语句中不能指定所有者。用户定义数据类型的名称必须是唯一的，所有用户都可以访问这个数据类型且不必将所有者作为前缀。

用户定义数据类型是数据库内的对象。其名称必须遵守标识符规则。与内置数据类型的名称一样，用户定义数据类型的名称始终不区分大小写。

缺省情况下，用户定义数据类型允许使用空值，除非 **allow_nulls_by_default** 选项设置为 **OFF**。此时，缺省情况下新的用户定义数据类型不允许空值。在用户定义数据类型上创建的列可否为空值取决于该用户定义数据类型的定义设置，而不是引用该列时的 **allow_nulls_by_default** 选项设置。任何在列定义中显式设置的空值或非空值都将覆盖用户定义数据类型设置。

使用 **CREATE DOMAIN** 语句可以对用户定义数据类型指定 **DEFAULT** 值。在数据类型上定义的任何列都将继承 **DEFAULT** 值规范。在列上显式指定的任何 **DEFAULT** 值都将覆盖为该数据类型指定的值。有关使用列 **DEFAULT** 值的详细信息，请参见《系统管理指南第一卷》>“数据完整性”>“列缺省值有助于保持数据完整性”。

使用 **CREATE DOMAIN** 语句可以将称为 **CHECK** 条件的规则添加至用户定义数据类型的定义中。

对于基表、全局临时表、本地临时表 and 用户定义数据类型，Sybase IQ 将强制执行 **CHECK** 约束。

要从数据库中删除数据类型，请使用 **DROP** 语句。必须是数据类型的所有者或者具有 **DBA** 权限才能删除用户定义数据类型。

另请参见《参考：构件块、表和过程》>“SQL 数据类型”。

副作用：

- 自动提交

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase — Adaptive Server Enterprise 不支持。Transact-SQL 使用 **sp_addtype** 系统过程以及 **CREATE DEFAULT** 和 **CREATE RULE** 语句提供类似功能。

权限

必须具有 **RESOURCE** 权限。

另请参见

- **DROP** 语句（第 170 页）

CREATE EVENT 语句

定义事件及其关联处理程序以自动执行预定操作。另外还定义了预定操作。

语法

```
CREATE EVENT event-name
[ TYPE event-type
  [ WHERE trigger-condition [ AND trigger-condition ], ... ]
  | SCHEDULE schedule-spec, ... ]
...[ ENABLE | DISABLE ]
...[ AT { CONSOLIDATED | REMOTE | ALL } ]
...[ HANDLER
      BEGIN
...
      END ]
```

参数

- **event-type:** - BackupEnd | “Connect” | ConnectFailed | DatabaseStart | DBDiskSpace | “Disconnect” | GlobalAutoincrement | GrowDB | GrowLog | GrowTemp | IQMainDBSpaceFree | IQTempDBSpaceFree | LogDiskSpace | “RAISERROR” | ServerIdle | TempDiskSpace
- **trigger-condition:** - event_condition(*condition-name*) { = | < | > | != | <= | >= } *value*
- **schedule-spec:** - [*schedule-name*] { STARTTIME *start-time* | BETWEEN *start-time* AND *end-time* } [EVERY *period* { HOURS | MINUTES | SECONDS }] [ON { (*day-of-week*, ...) | (*day-of-month*, ...) }] [STARTDATE *start-date*]
- **event-name | schedule-name:** - *identifier*
- **day-of-week:** - *string*
- **day-of-month | value | period:** - *integer*
- **start-time | end-time:** - *time*
- **start-date:** - *date*

示例

- **示例 1** - 指示数据库服务器每天凌晨 1 点执行自动增量备份:

```
CREATE EVENT IncrementalBackup
SCHEDULE
START TIME '1:00AM' EVERY 24 HOURS
HANDLER
  BEGIN
    BACKUP DATABASE INCREMENTAL
    TO 'backups/daily.incr'
  END
```

- **示例 2** – 指示数据库服务器每 10 分钟调用一次系统存储过程 **sp_iqspaceused**，然后将返回的当前日期和时间、当前的数据库连接数以及与使用 main 存储和临时 IQ 存储有关的当前信息存入表中：

```
CREATE TABLE mysummary(dt DATETIME,
    users INT, mainKB UNSIGNED BIGINT,
    mainPC UNSIGNED INT,
    tempKB UNSIGNED BIGINT,
    tempPC UNSIGNED INT) ;

CREATE EVENT mysummary
    SCHEDULE sched_mysummary
    START TIME '00:01 AM' EVERY 10 MINUTES
    HANDLER
    BEGIN
        DECLARE mt UNSIGNED BIGINT;
        DECLARE mu UNSIGNED BIGINT;
        DECLARE tt UNSIGNED BIGINT;
        DECLARE tu UNSIGNED BIGINT;
        DECLARE conncount UNSIGNED INT;

        SET conncount = DB_PROPERTY('ConnCount');
        CALL SP_IQSPACEUSED(mt,mu,tt,tu);

        INSERT INTO mysummary VALUES( NOW(),
            conncount, mu, (mu*100)/mt, tu,
            (tu*100)/tt );
    END;
```

- **示例 3** – 在含有事务日志文件的设备上的可用磁盘空间小于 30% 时，在服务器日志中张贴一条消息，但执行处理程序的频率不超过每 300 秒一次。

```
CREATE EVENT LowTxnLogDiskSpace
    TYPE DBDiskSpace
    WHERE event_condition( 'DBFreePercent' ) < 30
    AND event_condition( 'Interval' ) >= 300
    HANDLER
    BEGIN
        message 'Disk space for Transaction Log is low.';
    END;
```

有关更多的示例，请参见《系统管理指南第二卷》>“使用日程表和事件自动完成任务”>“事件的触发器状态”。

用法

主要有两种使用事件的方式：

- 调度操作 – 数据库服务器按时间调度执行一组操作。您可以使用此功能调度备份、有效性检查、报告表填充查询等。
- 数据库服务器在预定义的事件发生时执行一组操作。可以处理的事件包括磁盘空间限制（当磁盘使用超过了指定的百分比后）、服务器何时空闲等。

事件定义包括两个不同方面。触发器条件可以是一个事件，如磁盘使用超出了定义的阈值。调度指一组时间，其中每个时间充当一个触发器状态。当满足触发器状态时，

执行事件处理程序。事件处理程序包含一个或多个在复合语句 (**BEGIN... END**) 内指定的操作。

如果未提供触发条件或调度说明，则只有显式 **TRIGGER EVENT** 语句才能触发事件。在开发过程中，可能需要使用 **TRIGGER EVENT** 来开发和测试事件处理程序，并在测试完成后添加调度或 **WHERE** 子句。

事件错误记录到数据库服务器主控台。

触发事件处理程序后，服务器通过使用 **EVENT_PARAMETER** 函数向事件处理程序提供上下文信息（比如导致事件被触发的连接 ID）。

注意： 虽然事件中不允许包含返回结果集的语句，但您可以允许事件调用存储过程并将过程结果插入临时表中。请参见《系统管理指南第一卷》>“数据导入和导出”>“从数据库导出数据的方法”>“数据抽取工具”>“启用数据提取选项”>“提取限制”。

CREATE EVENT - *event-name* 是一个标识符。事件都有创建者，即创建事件的用户。事件处理程序只能在创建者的许可下执行。这与存储过程的执行是一样的。您不能创建由其他用户所拥有的事件。

通过查询系统表 **SYSEVENT**，可以列出事件名称。例如：

```
SELECT event_id, event_name FROM SYS.SYSEVENT
```

TYPE - *event-type* 是所列出的一组系统定义事件类型中的一种。事件类型不区分大小写。要指定此 *event-type* 触发事件的条件，请使用 **WHERE** 子句。

- **DiskSpace** 事件类型—如果数据库包含 **DiskSpace** 类型之一的事件处理程序，则数据库服务器每 30 秒对每一个与相关文件关联的设备上的可用空间进行检查。如果数据库有多个数据库空间位于不同的驱动器，则 **DBDiskSpace** 会检查每个驱动器并根据最低的可用空间执行操作。
LogDiskSpace 事件类型检查事务日志和任何事务日志镜像的位置，并根据最小的可用空间进行报告。
- **Globalautoincrement** 事件类型—当表的 **GLOBAL AUTOINCREMENT** 缺省值低于其范围的 1% 时，触发此事件。该处理程序的典型操作可能是为 **GLOBAL_DATABASE_ID** 选项请求新值。
对这种事件类型，可以使用以 **RemainingValues** 为参数的 **EVENT_CONDITION** 函数。
- **ServerIdle** 事件类型—如果数据库包含 **ServerIdle** 类型的事件处理程序，则服务器每 30 秒便会检查一次服务器的活动。

WHERE 子句 - 此触发器状态定义事件在什么情况下触发。例如，要在包含事务日志的磁盘 80% 以上的空间已满时执行操作，请使用以下触发条件：

```
... WHERE event_condition( 'LogDiskSpacePercentFree' ) < 20 ...
```

EVENT_CONDITION 函数的参数必须对该事件类型有效。

可以使用多个 **AND** 条件来构成 **WHERE** 子句，但不能使用 **OR** 条件或其它条件。

有关有效参数的信息，请参见《参考：构件块、表和过程》>“SQL 函数”>“按字母顺序排列的函数列表”>“EVENT_CONDITION 函数 [系统]”。

SCHEDULE – 指定预定操作何时发生。时间序列充当事件处理程序中定义的关联操作的一组触发条件。

对给定事件及其关联的处理程序，可以创建多个调度。这样可以实现复杂的调度。有多个调度时必须提供调度名称，而仅提供一个调度时，则是可选的。

通过查询系统表 SYSSCHEDULE，可以列出日程表名称。例如：

```
SELECT event_id, sched_name FROM SYS.SYSSCHEDULE
```

每个事件都具有唯一的事件 ID。可使用 SYSEVENT 和 SYSSCHEDULE 的 event_id 列将事件与关联的日程表匹配。

如果非反复出现的预定事件过去后，其调度将被删除，但不会删除事件处理程序。

预定事件的时间从调度创建时开始计算，并在事件处理程序完成执行时重新计算。计算下一次事件时间的方法是，检查事件的调度并找到即将到来的下一次预定时间。如果指示事件处理程序在 9:00 和 5:00 之间每隔一小时运行一次，并且它需要 65 分钟来执行，则它将分别在 9:00、11:00、1:00、3:00 和 5:00 运行。如果要重叠执行，则必须创建多个事件。

以下是调度定义的子句：

- **START TIME** – 在调度了事件的每一天中的最初预定时间。如果指定了 **START DATE**，则 **START TIME** 引用该日期。如果没有指定 **START DATE**，则 **START TIME** 是针对当前日期（除非该时间已过）和之后的每一天来说的。
- **BETWEEN...AND** – 一天中的一段时间，在该时间段之外，没有预定时间。如果指定了 **START DATE**，则在该日期之前不会有预定时间。
- **EVERY** – 连续预定事件之间的间隔。预定事件仅在当天的 **START TIME** 之后或 **BETWEEN...AND** 指定的范围内才会发生。
- **ON** – 预定事件发生日的列表。缺省值为每天。事件发生日可指定为周内某日或月内某日。
周内某日是星期一、星期二等等。也可以使用周内某日的简写形式，如 Mon、Tue 等等。数据库服务器可以识别 Sybase IQ 支持的任何语言的完整日期名和简写日期名。
月内某日是从 0 到 31 的整数。0 值表示任何月的最后一天。
- **START DATE** – 预定事件开始发生的日期。缺省为当前日期。

每当预定事件的处理程序完成后，即开始计算下一个预定时间和日期。

1. 如果使用了 **EVERY** 子句，则查看下次预定时间是否在当天，以及是否在 **BETWEEN...AND** 范围结束日期之前。如果是，即为下次调度时间。
2. 如果下一个预定时间不在当天，请查找下一个执行事件的日期。
3. 查找该日期的 **START TIME** 或 **BETWEEN...AND** 范围的开始日期。

ENABLE|DISABLE – 缺省情况下，事件处理程序处于启用状态。指定 **DISABLE** 后，即使是在预定时间内或满足触发条件时，事件处理程序也不会执行。**TRIGGER EVENT** 语句不会导致禁用的事件处理程序执行。

AT – 要在 **SQL Remote** 设置中的远程或统一数据库中执行事件，请使用此子句限制在哪个数据库中处理事件。缺省情况下，所有数据库都可以执行事件。

HANDLER – 每个事件都有一个处理程序。与存储过程的主体一样，处理程序是一个复合语句。然而，两者之间存在一些差异：您可以在复合语句内使用 **EXCEPTION** 子句处理错误，但不能使用存储过程中提供的 **ON EXCEPTION RESUME** 子句。

另请参见《系统管理指南第二卷》>“使用日程表和事件自动完成任务”。

副作用：

- 自动提交。
- 如果在执行期间未检测到任何错误，则系统将提交事件处理程序的操作。如果检测到错误，则系统将回退。

标准

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

权限

必须具有 **DBA** 权限。

在事件所有者的许可下，事件处理程序在一个单独的连接上执行。要使用 **DBA** 以外的权限执行，可以从事件处理程序内部调用过程：使用过程所有者的权限执行该过程。这个单独的连接不计入个人数据库服务器只能有 10 个连接的限制中。

另请参见

- **ALTER EVENT** 语句（第 13 页）
- **BEGIN……END** 语句（第 49 页）
- **COMMENT** 语句（第 61 页）
- **DROP** 语句（第 170 页）
- **TRIGGER EVENT** 语句（第 301 页）

CREATE EXISTING TABLE 语句

创建表示远程服务器上现有表的新代理表。

语法

```
CREATE EXISTING TABLE [owner.]table_name
[ ( column-definition, ... ) ]
AT 'location-string'
```

参数

- **column-definition:** - *column-namedata-type* [**NOT NULL**]
- **location-string:** - *remote-server-name*.[*db-name*].[*owner*].*object-name* | *remote-server-name*;*[db-name]*;*[owner]*;*object-name*

示例

- **示例 1** - 在远程服务器 `server_a` 上, 为 `nation` 表创建名为 `nation` 的代理表:

```
CREATE EXISTING TABLE nation
( n_nationkey int,
  n_name char(25),
  n_regionkey int,
  n_comment char(152))
AT 'server_a.dbl.joe.nation'
```

- **示例 2** - 在远程服务器 `server_a` 中为 `blurbs` 表创建了一个名为 `blurbs` 的代理表。Sybase IQ 将根据它从远程表中获得的元数据 派生出列列表:

```
CREATE EXISTING TABLE blurbs
AT 'server_a.dbl.joe.blurbs'
```

- **示例 3** - 在 Sybase IQ 远程 服务器 `remote_iqdemo_srv` 上, 为 `Employees` 表创建 名为 `rda_employee` 的代理表:

```
CREATE EXISTING TABLE rda_employee
AT 'remote_iqdemo_srv..dba.Employees'
```

用法

CREATE EXISTING TABLE 是 **CREATE TABLE** 的变体。**EXISTING** 关键字与 **CREATE TABLE** 一起使用时, 指定已存在于远程位置的表, 并指定 其元数据将导入到 Sybase IQ 中。以这种方式可使远程表成为用户 可以看见的实体。Sybase IQ 在创建该表前将确认它是否 存在于外部位置。

作为代理表使用的表, 其名称长度不能超过 30 个字符。

如果对象不存在 (作为主机数据文件或远程服务器对象), 此语句将被拒绝并出现一条错误消息。

主机数据文件或远程服务器表的索引信息被抽取出来用于创建系统表 `sysindexes` 的行。这定义了服务器方面的索引和关键字，使查询优化程序可以考虑该表上可能存在的任何索引。

参照约束在适当的时候传递到远程位置。

如果未指定列定义，**Sybase IQ** 会根据 它从远程表中获得的元数据来派生列列表。如果指定列定义，**Sybase IQ** 会对它们进行检验。当 **Sybase IQ** 检查列名、数据类型、长度和空值属性时：

- 列名称必须完全匹配（但忽略大小写）。
- **CREATE EXISTING TABLE** 中的数据类型必须匹配或者转换成远程位置的列的数据类型。例如，本地列的数据类型定义为 **NUMERIC**，而远程列的数据类型则为 **MONEY**。如果您从数据类型不匹配或存在其它不一致的表中进行选择，就可能会遇到错误。
- 系统将检查每个列的空值属性。如果本地列的空值属性与远程列的空值属性不相同，则系统将发出一条警告消息，但不会中止语句。
- 系统将检查每个列的长度。如果 **CHAR**、**VARCHAR**、**BINARY**、**DECIMAL** 和 **NUMERIC** 列的长度不匹配，则系统将发出一条警告消息，但不会中止命令。可以选择在 **CREATE EXISTING** 语句中仅包含实际远程列列表的子集。
- **AT** 指定 远程对象的位置。**AT** 子句支持将分号 (;) 用作分隔符。如果分号出现在位置字符串中，则它用作字段分隔符。如果没有分号，则使用句点作为字段分隔符。这样即可在数据库和所有者字段中 使用文件名和扩展名。分号字段分隔符主要用于当前不支持的服务器类；但是，您也可以在句点用作 字段分隔符的情况下使用它们。例如，以下语句将表 `proxy_al` 映射到 远程服务器 `myasa` 上的 **SQL Anywhere** 数据 库 `mydb`：

```
CREATE EXISTING TABLE
proxy_al
AT 'myasa;mydb;;al'
```

在 **Simplex** 环境中，不能在同一节点上创建引用远程表的代理表。在 **Multiplex** 环境中，不能创建引用在 **Multiplex** 中定义的远程表的代理表。

例如，在 **Simplex** 环境中，如果您尝试创建引用在同一节点上定义的基表 `Employees` 的代理表 `proxy_e`，则系统会拒绝 **CREATE EXISTING TABLE** 语句，并返回错误消息。在 **Multiplex** 环境中，如果从引用 **Multiplex** 中定义的远程表 `Employees` 的任何节点（协调节点或辅助节点）中创建代理表 `proxy_e`，则系统会拒绝 **CREATE EXISTING TABLE** 语句。

另请参见《系统管理指南第二卷》>“访问远程数据”和《系统管理指南第二卷》>“用于远程数据访问的服务器类”。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Open Client/Open Server 支持。

权限

必须具有 RESOURCE 权限。若要为其它用户创建表，您必须具有 DBA 权限。

另请参见

- CREATE TABLE 语句（第 136 页）

CREATE EXTERNLOGIN 语句

指派与远程服务器通信时使用的替代登录名和口令。

语法

```
CREATE EXTERNLOGIN login-name
                TO remote-server REMOTE LOGIN remote-user
[ IDENTIFIED BY remote-password ]
```

示例

- **示例 1** - 连接到服务器 **sybase1** 时，使用口令 **4TKNOX** 将名为 **DBA** 的本地用户映射到用户 **sa**：

```
CREATE EXTERNLOGIN dba
TO sybase1
REMOTE LOGIN sa
IDENTIFIED BY 4TKNOX
```

用法

CREATE EXTERNLOGIN 所做的更改将直到下一次与远程服务器连接时才生效。

缺省情况下，Sybase IQ 每次代表其客户端 连接到远程服务器时都会使用这些客户端的名称和口令。**CREATE EXTERNLOGIN** 指派在与远程服务器通信时 要使用的替代登录名和 口令。它以加密的形式将口令存储在内部。*remote_server* 必须 通过 ISYSSERVER 系统表中的条目 为本地服务器所知。有关详细信息，请参见“CREATE SERVER 语句”。

使用 **CREATE EXTERNLOGIN** 语句创建 远程登录并使用 **CREATE SERVER** 语句定义远程服务器可以针对 **INSERT...LOCATION** 设置外部登录名和口令，从而任何用户均可在任何上下文中使用登录名和口令。这可 避免可能难以获得登录名或口令而产生错误，并且是连接到远程服务器的推荐方式。

注意： 如果您使用的是当前连接的用户 ID 和口令，并且用户更改了口令，则您必须停止并重新启动服务器，新的口令才能在远程服务器上生效。更改缺省用户 ID 的口令不会影响使用 **CREATE EXTERNLOGIN** 创建的远程登录。

具有自动口令失效功能的站点应该就定期更新外部登录口令作出计划。

不能从事务内部使用 **CREATE EXTERNLOGIN**。

login-name—指定本地用户登录名。如果使用集成登录，*login-name* 是 Windows 用户 ID 所映射到的数据库用户。

TO—**TO** 子句指定远程服务器的名称。

REMOTE LOGIN—**REMOTE LOGIN** 子句为本地用户 *login-name* 指定用于 *remote-server* 的用户帐户。

IDENTIFIED BY—**IDENTIFIED BY** 子句指定 *remote-password* 是 *remote-user* 的口令。如果省略 **IDENTIFIED BY** 子句，则系统会将空值作为口令发送给远程服务器。如果指定 **IDENTIFIED BY " "**（空字符串），则发送的口令为空字符串。

remote-user 和 *remote-password* 组合对于 *remote-server* 必须有效。

副作用

- 自动提交

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Open Client/Open Server 支持。

权限

只有 DBA 或 USER ADMIN 帐户才能添加或修改外部登录名。

另请参见

- DROP EXTERNLOGIN 语句（第 174 页）
- INSERT 语句（第 211 页）
- CREATE SERVER 语句（第 132 页）

CREATE FUNCTION 语句

在数据库中创建一个新函数。

语法

语法 1

```
CREATE [ OR REPLACE ] [ TEMPORARY ] FUNCTION [ owner. ] function-name
( [ parameter, ... ] )
RETURNS data-type routine-characteristics
[ SQL SECURITY { INVOKER | DEFINER } ]
{ compound-statement
| AS tsql-compound-statement
| external-name }
```

语法 2

```
CREATE FUNCTION [ owner.]function-name ( [ parameter, ... ] )
RETURNS data-type
URL url-string
[ HEADER header-string ]
[ SOAPHEADER soap-header-string ]
[ TYPE { 'HTTP[:{ GET | POST } ] ' | 'SOAP[:{ RPC | DOC } ]' } ]
[ NAMESPACE namespace-string ]
[ CERTIFICATE certificate-string ]
[ CLIENTPORT clientport-string ]
[ PROXY proxy-string ]
```

参数

- url-string:** - '{ HTTP | HTTPS | HTTPS_FIPS } ://[user:password@]hostname[:port][path]'
- parameter:** - *IN*parameter-namedata-type [DEFAULT expression]
- routine-characteristics:** - ONEXCEPTIONRESUME | [NOT] DETERMINISTIC
- tsql-compound-statement:** - *sql-statement sql-statement ...*
- external-name:** - EXTERNALNAME *library-call* | EXTERNALNAME *java-call* LANGUAGE JAVA
- library-call:** - '[operating-system:]function-name@library, ...'
- operating-system:** - UNIX
- java-call:** - '[package-name.]class-name.method-namemethod-signature'
- method-signature:** - ([field-descriptor, ...]) return-descriptor
- field-descriptor 和 return-descriptor:** - Z | B | S | I | J | F | D | C | V | [descriptor | L class-name;

示例

- 示例 1** - 将 `firstname` 字符串与 `lastname` 字符串并置起来:

```
CREATE FUNCTION fullname (
    firstname CHAR(30),
    lastname CHAR(30) )
RETURNS CHAR(61)
BEGIN
    DECLARE name CHAR(61);
    SET name = firstname || ' ' || lastname;
    RETURN (name);
END
```

下面的示例说明了 **fullname** 函数的使用。

- 从两个提供的字符串中返回完整的名称:

```
SELECT fullname ('joe','smith')
```

fullname('joe', 'smith')

joe smith

- 列出所有雇员的姓名:

```
SELECT fullname (givenname, surname)
FROM Employees
```

fullname (givenname, surname)
Fran Whitney
Matthew Cobb
Philip Chin
Julie Jordan
Robert Breault
...

- **示例 2** – 使用 Transact-SQL 语法:

```
CREATE FUNCTION DoubleIt ( @Input INT )
RETURNS INT
AS
DECLARE @Result INT
SELECT @Result = @Input * 2
RETURN @Result
```

语句 `SELECT DoubleIt(5)` 返回值 10。

- **示例 3** – 创建一个用 Java 编写的外部函数:

```
CREATE FUNCTION dba.encrypt( IN name char(254) )
RETURNS VARCHAR
EXTERNAL NAME
'Scramble.encrypt (Ljava/lang/String;)Ljava/lang/String;'
LANGUAGE JAVA
```

用法

CREATE FUNCTION 语句将在数据库中创建用户定义的函数。通过指定所有者名称，可以为其他用户创建函数。根据权限，可以用与其它非集合函数的使用方法完全相同的方法使用用户定义的函数。

CREATE FUNCTION — 参数名称必须遵守 数据库标识符规则。它们必须是有效的 **SQL** 数据类型，而且必须以关键字 **IN** 作为前缀，以表明参数是为函数 提供值的表达式。

指定 **CREATE OR REPLACE FUNCTION** 将创建一个新函数或替换同名的现有函数。替换某个函数后，该函数的定义将更改，但现有的权限将保留。

不能将 **OR REPLACE** 子句与临时函数一起使用。

执行函数时，不必指定所有参数。如果在 **CREATE FUNCTION** 语句中提供了缺省值，则系统会为缺少的参数指派缺省值。如果调用者未提供参数且未设置缺省值，则发生错误。

如果指定 **TEMPORARY (CREATETEMPORARYFUNCTION)**，则将意味着该函数仅对创建它的连接可见，并在删除该连接时随之自动删除。也可以显式删除临时函数。无法

在临时函数上执行 **ALTER**、**GRANT** 或 **REVOKE** 操作，而且与其它函数不同，临时函数不会在目录或事务日志中予以记录。

具有临时函数创建者（当前用户）权限才能执行临时函数，并且临时函数只能由其创建者所有。因此，创建临时函数时不要指定所有者。

如果已连接到只读数据库，则可以创建和删除临时函数。

SQL SECURITY 一 定义是作为 **INVOKER**（调用函数的用户）还是作为 **DEFINER**（拥有函数的用户）执行函数。缺省为 **DEFINER**。

如果指定了 **SQL SECURITY INVOKER**，系统将使用更多内存，因为必须为调用过程的每个用户进行注释。此外，如果指定了 **SQL SECURITY INVOKER**，则系统也将以调用者身份执行名称解析。因此，请注意利用所有对象的相应所有者限定这些对象的名称（表、过程等）。

compound-statement——组用 **BEGIN** 和 **END** 括起来的 SQL 语句，中间用分号分隔。请参见“**BEGIN**……**END** 语句”。

tsql-compound-statement——一批 Transact-SQL 语句。请参见《参考：构件块、表和过程》>“与其它 Sybase 数据库的兼容性”>“Transact-SQL 过程语言概述”>“Transact-SQL 批处理概述”和“**CREATE PROCEDURE** 语句 [T-SQL]”。

EXTERNAL NAME——使用 **EXTERNAL NAME** 子句的函数是包含对外部库函数调用的包装。使用 **EXTERNAL NAME** 的函数在 **RETURNS** 子句之后可不含任何其它子句。库名可包含文件扩展名，在 Windows 中通常为 .dll，在 UNIX 中通常为 .so。在没有扩展名的情况下，该软件附加平台特定的缺省库文件扩展名。

临时函数不支持 **EXTERNAL NAME** 子句。请参见“SQL Anywhere Server - 编程”>“SQL Anywhere 外部调用接口”。

注意： 此参考指向 SQL Anywhere 文档。

EXTERNAL NAME LANGUAGE JAVA——将 **EXTERNAL NAME** 与 **LANGUAGE JAVA** 子句一起使用的函数是包含 Java 方法的包装。有关调用 Java 过程的信息，请参见“**CREATE PROCEDURE** 语句”。

ON EXCEPTION RESUME——使用 Transact-SQL-like 错误处理。请参见“**CREATE PROCEDURE** 语句”。

NOT DETERMINISTIC——指定为 **NOT DETERMINISTIC** 的函数每次在查询中调用时都将重新求值。不是以这种方式指定的函数的结果可以存入高速缓存以便提高性能，并且每次在查询求值过程中使用相同的参数调用函数时，都会重用高速缓存的结果。

对于具有副作用（如修改基础数据）的函数，应将其声明为 **NOT DETERMINISTIC**。例如，一个生成主键值并用在 **INSERT** ... **SELECT** 语句中的函数应声明为 **NOT**

DETERMINISTIC：

```
CREATE FUNCTION keygen( increment INTEGER )
RETURNS INTEGER
NOT DETERMINISTIC
```

```

BEGIN
    DECLARE keyval INTEGER;
    UPDATE counter SET x = x + increment;
    SELECT counter.x INTO keyval FROM counter;
    RETURN keyval
END
INSERT INTO new_table
SELECT keygen(1), ...
FROM old_table

```

如果函数对给定的输入参数总是返回相同的值，则可以将其声明为 **DETERMINISTIC**。

除非所有用户定义的函数都声明为 **NOT DETERMINISTIC**，否则它们将被视为确定型函数。确定型函数为相同参数返回一致的结果并且没有副作用。也就是说，数据库服务器假设对具有相同参数的同一函数的两次连续调用将返回相同的结果，同时不会对查询语义产生有害副作用。

注意：CLR、ESQL 和 ODBC、Perl 和 PHP 外部环境中的用户定义的函数由 Catalog 存储 (SQL Anywhere) 处理。它们不能利用 Sybase IQ 的性能特性。Java 中的用户定义的函数由 Sybase IQ 来处理，并可提高性能。

在某些情况下，如果在用户定义的函数中发出查询，SQL Anywhere 和 Sybase IQ 之间的语义差异可导致不同的查询结果。例如，Sybase IQ 将 CHAR 和 VARCHAR 数据类型视为各不相同的数据类型，而 SQL Anywhere 按 VARCHAR 的处理方式来处理 CHAR 数据。

要修改用户定义的函数或通过加密函数定义来隐藏函数的内容，请使用 **ALTER FUNCTION** 语句。

URL—仅用于定义 **HTTP** 或 **SOAP** Web 服务客户端函数。指定 Web 服务的 URL。其中的用户名和口令参数是可选的，它们提供了一种用于提供 **HTTP** 基本验证所需的证书的方法。**HTTP** 基本鉴定以 64 为基数对用户和口令信息进行编码并在 **HTTP** 请求的“Authentication”标头中传递编码后的信息。

对于 Web 服务客户端函数来说，返回类型为 **SOAP** 和 **HTTP** 的函数必须为字符数据类型中的一种，如 VARCHAR。返回的值为 **HTTP** 响应的正文。不含任何 **HTTP** 标头信息。如果需要详细信息（如状态信息），请使用过程而不是函数。

参数值作为请求的一部分进行传递。所用的语法取决于请求的类型。对于 **HTTP:GET**，参数将作为 URL 的一部分进行传递；对于 **HTTP:POST** 请求，值将位于请求的主体中。**SOAP** 请求的参数总是被绑定在请求主体中。

HEADER—创建 **HTTP** Web 服务客户端函数时，此子句用于添加或修改 **HTTP** 请求标头条目。仅可为 **HTTP** 标头指定可打印 ASCII 字符，且这些字符不区分大小写。有关如何使用此子句的详细信息，请参见“CREATE PROCEDURE 语句”的 **HEADER** 子句。

有关使用 **HTTP** 标头的详细信息，请参见“SQL Anywhere Server - 编程”>“**HTTP** Web 服务”»“使用 Web 客户端访问 Web 服务”»“开发 Web 客户端应用程序”»“Web 客户端函数和过程要求与推荐”>“**HTTP** 请求标头管理”。

注意： 此参考指向 SQL Anywhere 文档。

SOAPDHEADER—当将 **SOAP** Web 服务声明为函数时，此子句用于指定一个或多个 **SOAP** 请求 标头条目。**SOAP** 标头可声明为 静态常量，也可以使用参数 替代机制动态设置（为 hd1、hd2 等声明 **IN**、**OUT** 或 **INOUT** 参数）。Web 服务函数可以定义一个或多个 **IN** 模式 替代参数，但无法定义 **INOUT** 或 **OUT** 替代参数。有关如何使用该子句的详细信息，请参见“SQL Anywhere Server – SQL 参考”>“SQL 语句”>“CREATE PROCEDURE 语句（Web 客户端）”中的 **SOAPHEADER** 子句。

注意： 此参考指向 SQL Anywhere 文档。

TYPE—指定提出 Web 服务请求时使用的格式。如果指定 **SOAP** 或未包括类型子句，则使用缺省类型 **SOAP:RPC**。**HTTP** 意味着 **HTTP:POST**。**SOAP** 请求始终作为 XML 文档发送。**HTTP:POST** 始终用于发送 **SOAP** 请求。

NAMESPACE—仅适用于 **SOAP** 客户端函数并且标识出 **SOAP:RPC** 和 **SOAP:DOC** 请求通常都需要的方法命名空间。处理请求的 **SOAP** 服务器使用此命名空间来解释 **SOAP** 请求消息主体中的实体名称。可以从 Web 服务服务器中提供的 **SOAP** 服务的 WSDL 说明中获取该命名空间。缺省值为过程的 URL，最多为可选路径组件（但不包括）。

CERTIFICATE—要发出安全 (HTTPS) 请求，客户端必须具有对 HTTPS 服务器所用证书的访问权限。在分号分隔的关键字/值对的字符串中指定所需信息。证书可置于文件中且假设该文件的名称使用文件关键字，或者整个证书可置于字符串中，但不能同时出现这两种情况。它提供以下键：

键	缩写	描述
文件		证书的文件名
certificate	cert	证书
company	co	证书中指定的公司
unit		证书中指定的公司单元
名称		证书中指定的常见名称

只有定向到 **HTTPS** 服务器或可以从非安全服务器重定向至安全服务器的请求才需要证书。

CLIENTPORT—标识 **HTTP** 客户端过程使用 **TCP/IP** 在其上通信的端口号。假定该端口用于并且建议只用于跨防火墙的连接，因为防火墙根据 **TCP/UDP** 端口进行过滤。可以指定一个端口号、端口号范围或两者的组合，例如 **CLIENTPORT '85,90-97'**。

请参见《系统管理指南第一卷》>“连接和通信参数”>“网络通信参数”>“ClientPort 通信参数 [CPort]”。

PROXY—指定代理服务器的 **URI**。在客户端必须通过代理服务器访问网络时使用。表示该过程将连接到代理服务器并通过它向 Web 服务发送请求。

副作用

SQL 语句

- 自动提交

标准

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

权限

必须具有 RESOURCE 权限。

外部函数（包括 Java 函数）必须有 DBA 权限。

另请参见

- ALTER FUNCTION 语句（第 14 页）
- BEGIN……END 语句（第 49 页）
- CREATE PROCEDURE 语句（第 112 页）
- DROP 语句（第 170 页）
- RETURN 语句（第 269 页）

CREATE FUNCTION 语句 (Java UDF)

用于在数据库中新建外部 Java 表 UDF 函数。

语法

```
CREATE [ OR REPLACE | TEMPORARY ] FUNCTION [ owner.]function-name
( [ parameter, ... ] )
  RETURNS data-type routine-characteristics
  [ SQL SECURITY { INVOKER | DEFINER } ]
  { compound-statement
    | AS tsq1-compound-statement
    | EXTERNAL NAME 'java-call' LANGUAGE JAVA }
```

参数

- **参数** - INparameter-namedata-type [DEFAULTexpression]
- **routine-characteristics** - ONEXCEPTIONRESUME | [NOT] DETERMINISTIC
- **tsq1-compound-statement** - sql-statement sql-statement ...
- **environment-name** - JAVA [ALLOW | DISALLOW SERVER SIDE REQUESTS]
- **java-call** - '[package-name.]class-name.method-namemethod-signature'
- **method-signature** - ([field-descriptor, ...]) return-descriptor
- **field-descriptor** 和 **return-descriptor** - Z | B | S | I | J | F | D | C | V | [descriptor | L class-name;

示例

- **示例 1** – 创建一个用 Java 编写的外部函数：

```
CREATE FUNCTION dba.encrypt( IN name char(254) )
RETURNS VARCHAR
EXTERNAL NAME
'Scramble.encrypt (Ljava/lang/String;)Ljava/lang/String;'
LANGUAGE JAVA
```

用法

不许将 LONG BINARY 和 LONG VARCHAR 用作 return-value 数据类型。

EXTERNAL NAME LANGUAGE JAVA — 是将 **EXTERNAL NAME** 与 **LANGUAGE JAVA** 子句一起使用的函数，也是包含 Java 方法的包装。

iq-environment-name: JAVA [ALLOW/ DISALLOW SERVER SIDE REQUESTS]:

DISALLOW 为缺省值。

ALLOW 表示允许建立服务器端连接。

注意：除非有必要，否则不要指定 **ALLOW**。设置 **ALLOW** 会减慢某些类型的 Sybase IQ 表的连接。

不要在同一查询中同使用 UDF 与 **ALLOW SERVER SIDE REQUESTS** 和 **DISALLOW SERVER SIDE REQUESTS**。

标准

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

权限

必须具有 **RESOURCE** 权限。

外部 Java 函数必须有 **DBA** 权限。

CREATE INDEX 语句

创建指定表或表的索引。

语法

```
CREATE [ UNIQUE ] [ index-type ] INDEX [ IF NOT EXISTS ] index-name
...ON [ owner. ] table-name
... ( column-name [ , column-name ] ... )
...| [ WITH NULLS NOT DISTINCT ]
...[ { IN | ON } dbspace-name ]
```

```
...[ NOTIFY integer ]
...[ DELIMITED BY 'separators-string' ]
...[ LIMIT maxwordsize-integer ]
```

参数

- **index-type:** - { **CMP** | **HG** | **HNG** | **LF** | **WD** | **DATE** | **TIME** | **DTTM** }

示例

- **示例 1** - 创建 `projected_earnings` 和 `current_earnings` 列的比较索引。这些列是具有相同精度和小数位数的 `decimal` 列。

```
CREATE
```

```
CMP INDEX proj_curr_cmp
ON sales_data
( projected_earnings, current_earnings )
```

- **示例 2** - 在 `SalesOrderItems` 表的 `ID` 列上创建一个 `High_Group` 索引。该索引的数据页是从数据库空间 `Dsp5` 中分配的。

```
CREATE
```

```
HG INDEX id_hg
ON SalesOrderItems
( ID ) IN Dsp5
```

- **示例 3** - 在 `SalesOrderItems` 表的 `ProductID` 列上创建一个 `High_Group` 索引：

```
CREATE HG INDEX item_prod_hg
ON Sales_OrderItems
( ProductID)
```

- **示例 4** - 在 `SalesOrderItems` 表上为同一 `ProductID` 列创建一个 `Low_Fast` 索引，而不会出现任何通知消息：

```
CREATE LF INDEX item_prod
ON SalesOrderItems
( ProductID)
NOTIFY 0
```

- **示例 5** - 创建 `earnings_report` 表上的 `WD` 索引。将字符串的分隔符指定为空格、冒号、分号和句点。将字符串的长度限制为 25。

```
CREATE WD INDEX earnings_wd
ON earnings_report_table(varchar)
DELIMITED BY ' : ; . '
LIMIT 25
```

- **示例 6** - 在 `SalesOrders` 表的 `OrderDate` 列上创建一个 `DTTM` 索引：

```
CREATE DTTM INDEX order_dttm
ON SalesOrders
( OrderDate )
```

用法

CREATE INDEX 语句可创建指定表中指定列的索引。索引一旦创建后就再也不能在 SQL 语句中再次引用，除非使用 **DROP INDEX** 语句将它删除。

对于 Sybase IQ 表中的列，*index-type* 可指定为 **HG** (High_Group)、**HNG** (High_Non_Group)、**LF** (Low_Fast)、**WD** (Word)、**DATE**、**TIME** 或 **DTTM** (Datetime)。如果未指定 *index-type*，则缺省情况下创建 **HG** 索引。

要在 IQ 表中两个列之间的关系上创建索引，*index-type* 可指定为 **CMP** (Compare)。两个列必须具有相同的数据类型、精度和小数位数。对于 **CHAR**、**VARCHAR**、**BINARY** 或 **VARBINARY** 列，精度表示两列具有相同的宽度。

对于最快查询速度，列的索引的正确类型取决于：

- 列中的唯一值数
- 列将在查询中的使用方式
- 可用的磁盘空间大小

《系统管理指南第一卷》详细说明了索引类型并说明了如何为数据确定适当的索引类型。

可以在 IQ 表中的列上指定多个索引，但这些索引的索引类型必须不同。使用 **CREATE INDEX** 不能添加重复的索引类型。Sybase IQ 会选择适用于当前查询或查询的部分的最快索引。但各其它索引类型可能会显著增加该表的空间要求。

column-name—指定要进行索引的列的名称。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“FROM 子句”。）如果列名含有字母、数字和下划线以外的其它字符，则用引号 (") 将它引起来。

如果省略 **UNIQUE**，则可以仅指定 **HG** 索引。外键要求非唯一 **HG** 索引并且组合外键要求非唯一组合 **HG** 索引。唯一和不唯一的 **HG** 索引的多列组合键都具有最大宽度 5300 字节。如果 **CHAR** 或 **VARCHAR** 数据是组合键或单列 **HG**、**LF**、**HNG**、**DATE**、**TIME** 或 **DTTM** 索引，则这两种数据都不能超过 255 字节。

UNIQUE—**UNIQUE** 可确保在索引中的所有列中，表中的任意两行都不具有相同的值。每个索引键必须是唯一的，或者至少在一列中包含 **NULL**。可以对多个列创建唯一的 **HG** 索引，但不能使用其它索引类型创建多列索引。不能使用 **UNIQUE**、**CMP**、**HNG**、**WD**、**DATE** 或 **TIME** 索引类型指定 **DTTM**。

如果列定义允许使用空值并且不强制执行约束（主键或唯一），则 Sybase IQ 允许在用户创建的唯一多列 **HG** 索引的数据值中使用空值。有关详细信息，请参见“注释”中的“多列索引”。

如果指定了 **UNIQUE**，但未指定 **WITH NULLS NOT DISTINCT**，则每个索引键必须是唯一的，或者至少在一列中包含空值。例如，('a', NULL) 和 ('a', NULL) 两个条目中的每个条目都视为唯一的。

如果指定了 **UNIQUE...WITH NULLS NOT DISTINCT**，则无论是否包含空值，索引键必须是唯一的。例如，('a', NULL) 和 ('a', NULL) 两个条目被视为相同的，而不是唯一的。

IF NOT EXISTS — 如果指定的对象已存在，则不会进行任何更改，也不会返回错误。

IN—指定索引位置。如果省略 **IN** 子句，将在创建表的 **dbspace** 中创建索引。索引始终放在与其表类型相同的 **dbspace** (**IQ** 存储或临时存储) 中。装载索引时，数据将跨所有具备可用空间的该类型数据库文件分布。**Sybase IQ** 确保您指定的任何 *dbspace-name* 都适用于该索引。如果尝试为临时表上的索引指定 **IQ_SYSTEM_MAIN** 或其它主 **dbspace** (反之亦然)，则会收到一个错误。**dbspace** 名称始终不区分大小写，无论 **CREATE DATABASE**……**CASE IGNORE** 或 **CASE RESPECT** 规范如何。

DELIMITED BY — 指定用于 将列字符串分析为要存储到 该列的 **WD** 索引中的单词的分隔符。如果忽略该子句或指定该值为空字符串，则 **Sybase IQ** 会使用缺省分隔符集。缺省分隔符集是为缺省归类顺序 (**ISO-BINENG**) 设计的。它包括非 7 位 **ASCII** 字母数字字符的所有 7 位 **ASCII** 字符，但连字符和单引号除外。在缺省情况下，连字符和单引号属于字词的一部分。缺省分隔符集中有 64 个分隔符。例如，如果列值为以下字符串：

```
The cat is on the mat
```

并且使用了使用缺省分隔符的 **CASE IGNORE** 设置来创建数据库，下列单词会从该字符串存储到 **WD** 索引中：

```
cat is mat on the
```

如果指定多个 **DELIMITED BY** 和 **LIMIT** 子句，则不会返回任何错误，但仅使用每种类型的最后一个子句。

separators-string—分隔符字符串必须是数据库创建时使用的归类顺序中的 0 个或多个字符的序列。分隔符字符串中的每个字符都被当作一个分隔符。如果分隔符字符串中没有字符，则会使用缺省分隔符集。（每个分隔符必须是正在使用的归类序列中的单个字符。）分隔符字符串中的字符（分隔符）不能多于 256 个。

要将制表符指定为分隔符，可以在分隔符字符串内键入 <TAB> 字符，或者使用制表符的十六进制 **ASCII** 代码 \x09。“\t”指定两个分隔符，\ 和字母 t。要将换行符指定为分隔符，可以键入 <RETURN> 字符或十六进制 **ASCII** 代码 \x0a。

例如，子句 **DELIMITED BY ' : ; . \ / t '** 可指定以下七个分隔符：
space : ; . \ / t

表 5. 制表符和换行符作为分隔符

对于以下分隔符	在 DELIMITED BY 子句中使用此分隔符字符串
tab	' ' (键入 <TAB>) 或 '\x09'
newline	' ' (键入 <RETURN>) 或 '\x0a'

LIMIT—仅可用于创建 **WD** 索引。指定 **WD** 索引中允许的最大单词长度。分析过程中单词长度超过该值会导致错误。缺省值为 255 个字节。最小允许值为 1 而最大允许值为 255。如果 **CREATE INDEX** 语句中指定的或缺省确定的最大单词长度超过列宽，则所用的最大单词长度会自动减小至列宽。使用的最大允许单词长度越小，允许插入、删

除和更新使用的空间会更小、时间会更短。空词（两个相邻的分隔符）将被自动忽略。在创建 **WD** 索引后，到其列的任何插入都将使用创建时确定的分隔符和最大单词大小进行分析。创建索引之后，这些分隔符和最大单词大小将无法更改。

NOTIFY—在为索引成功添加 n 个记录后，会给出通知消息。这些消息会发送到标准输出设备。消息包含与内存使用情况、数据库空间以及当前使用的缓冲区大小有关的信息。缺省值为 100,000 条记录。要关闭 **NOTIFY**，请将其设置为 0。

注释

- 索引所有权—在 **CREATE INDEX** 语句中无法指定索引所有者。索引会自动由在其上定义这些索引的表的所有者所拥有。索引名称对每个所有者必须是唯一的。
- 不存在视图索引—无法为视图创建索引。
- 索引名称—每个索引的名称对给定的表必须是唯一的。
- 排它表使用—只要 **CREATE INDEX** 影响了当前正由其它连接修改的表，就会禁止该语句。但允许对正在同时添加索引的表进行查询。
- CHAR** 列—在创建 **WD** 索引后，到其列的任何插入都将使用分隔符进行分析，并且在创建索引后，最大单词大小将无法更改。

对于 **CHAR** 列，Sybase 建议您至少指定一个空格作为其中的一个分隔符或使用缺省分隔符集。Sybase IQ 会自动将 **CHAR** 列填充至最大列宽。除字符数据外，如果列包含还包含空白，则对 **WD** 索引数据的查询可能会返回令人误解的结果。例如，列 **CompanyName** 包含由分隔符分隔的两个单词，但第二个单词以空白填充：

```
'Concord' 'Farms'
```

假设用户输入以下查询：

```
SELECT COUNT(*)FROM Customers WHERE CompanyName contains
( 'Farms' )
```

分析程序确定字符串包含：

```
'Farms'
```

而不是：

```
'Farms'
```

并返回 0 而不是 1。您可以通过使用 **VARCHAR** 列而不是 **CHAR** 列避免此问题。

- 数据类型
 - 不能使用 **CREATE INDEX** 针对包含 **BIT** 数据的列创建索引。
 - 对于 255 字节以上的 **CHAR** 和 **VARCHAR** 数据，只能创建缺省索引、**CMP** 索引或 **WD** 索引。
 - 对于 **LONG VARCHAR** 数据，只能创建缺省索引和 **WD** 索引类型。
 - 对于 255 字节以上的 **BINARY** 和 **VARBINARY** 数据，只能创建缺省索引、**CMP** 索引和 **TEXT** 索引类型。
 - 不能对包含 **FLOAT**、**REAL** 或 **DOUBLE** 数据的列创建 **HNG** 索引或 **CMP** 索引。
 - TIME** 索引只能针对数据类型为 **TIME** 的列创建。

- **DATE** 索引只能针对数据类型为 **DATE** 的列创建。
- **DTTM** 索引只能针对数据类型为 **DATETIME** 或 **TIMESTAMP** 的列创建。
- 多列索引 — 可以对多个列创建唯一的或非唯一的 **HG** 索引。Sybase IQ 可对组成外键的一组列隐式创建一个非唯一的 **HG** 索引。

仅 **HG** 和 **CMP** 是可以包含多列的索引类型。不能对多个列创建唯一的 **HNG** 或 **LF** 索引，也不能对多个列创建 **DATE**、**TIME** 或 **DTTM** 索引。

多列串联键的最大宽度为 **5KB**（5300 字节）。允许的列数取决于 **5KB** 包含的列数。超过 255 字节的 **CHAR** 或 **VARCHAR** 数据不允许作为单列 **HG**、**LF**、**HNG**、**DATE**、**TIME** 或 **DTTM** 索引中的组合键的一部分。

基表上的多列索引在使用那些基表创建的连接索引中不会进行复制。

用于多列索引的 **INSERT** 必须包含索引的所有列。

在 **ORDER BY** 子句中对一个列进行查询的速度快于使用多列 **HG** 索引进行查询的速度。例如：

```
SELECT abs (x) from t1
ORDER BY x
```

在以上示例中，**HG** 索引在排序顺序中垂直投射 **x**。

若要增强查询性能，请在带有以下条件的 **SELECT** 或 **ORDER BY** 子句中使用多列 **HG** 索引对多个列执行 **ORDER BY** 操作（也可以包括 **ROWID**）：

- 索引中存在所有投射的列以及所有排序列（**ROWID** 除外）
- 排序关键字的顺序与前导 **HG** 列的顺序相符

如果多个多列 **HG** 索引满足这些条件，则使用具有最低离散值个数的索引。

如果查询具有一个 **ORDER BY** 子句，并且 **ORDER BY** 列列表为多列索引的前缀（**SELECT** 列表中引用的所有列都存在于一个多列索引中），则该多列索引执行垂直投射。例如：

```
SELECT x,z,y FROM T
ORDER BY x,y
```

如果 **SELECT** 列表中的基列中存在表达式，并且所有表达式中引用的所有列都存在于该多列索引中，则查询将使用多列索引。例如：

```
SELECT power(x,2), x+y, sin(z) FROM T
ORDER BY x,y
```

除以上两个示例之外，如果 **ROWID()** 函数位于 **SELECT** 列表表达式中，则将使用多列索引。例如：

```
SELECT rowid()+x, z FROM T
ORDER BY x,y,z
```

除以上三个示例之外，如果 **ROWID()** 存在于 **ORDER BY** 列表结尾处，并且该列表的列（不包括 **ROWID()**）以相同顺序使用多列索引，则多列索引将用于该查询。例如：

```
SELECT z,y FROM T
ORDER BY x,y,z,ROWID()
```

如果列定义允许使用空值并且不强制执行约束（主键或唯一），则 Sybase IQ 允许在用户创建的唯一多列 **HG** 索引的数据值中使用空值。此功能的规则如下：

- 空值会被视为未定义的值。
- 唯一索引列中允许具有空值的多个行。
 1. 在单列索引中，一个索引列中允许多个具有空值的行。
 2. 在多列索引中，只要其余列中的非空值可保证该索引中的唯一性，索引列允许多个具有空值的行。
 3. 在多列索引中，索引涉及的所有列中允许多个具有空值的行。

以下两个示例说明了这些规则。给定表 `table1`：

```
CREATE TABLE table1
(c1 INT NULL, c2 INT NULL, c3 INT NOT NULL);
```

对允许空值的列创建唯一的单列 **HG** 索引：

```
CREATE UNIQUE HG INDEX c1_hg1 ON table1 (c1);
```

根据上述规则 1，可以将空值插入到索引列的多个行：

```
INSERT INTO table1(c1,c2,c3) VALUES (NULL,1,1);
INSERT INTO table1(c1,c2,c3) VALUES (NULL,2,2);
```

对允许空值的列创建唯一多列 **HG** 索引：

```
CREATE UNIQUE HG INDEX c1c2_hg2 ON table1(c1,c2);
```

根据上述规则 2，必须确保索引的唯一性。以下 **INSERT** 不会成功，因为对行 1 和行 3 的多列索引 `c1c2_hg2` 具有相同的值：

```
INSERT INTO table1(c1,c2,c3) VALUES (NULL,1,3);
```

但根据规则 1 和 3，以下 **INSERT** 操作会成功：

```
INSERT INTO table1(c1,c2,c3) VALUES (NULL,NULL,3);
INSERT INTO table1(c1,c2,c3) VALUES (NULL,NULL,4);
```

唯一性会在多列索引中得到保留。

以下 **UPDATE** 操作会成功，因为规则 3 允许在多列索引的所有列中包含多个具有空值的行：

```
UPDATE table1 SET c2=NULL WHERE c3=1
```

如果多列 **HG** 索引受唯一约束制约，则索引中涉及的任何列都不允许空值。

- 并行索引创建—可以使用 **BEGIN PARALLEL IQ ... END PARALLEL IQ** 语句对多个 IQ 表中的 **CREATE INDEX** 语句进行分组，这样便可如同单个 DDL 语句一样执行这些语句了。有关详细信息，请参见“**BEGIN PARALLEL IQ ... END PARALLEL IQ** 语句”。

警告！ 对包含未提交数据的本地临时表使用 **CREATE INDEX** 命令会失败并产生错误消息“要创建索引，必须提交本地临时表 <tablename>。”请先提交本地临时表中的数据，然后再创建索引。

另请参见《系统管理指南 第一卷》>“Sybase IQ 索引”。

副作用

- 自动提交

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase — Adaptive Server Enterprise 拥有的 **CREATE INDEX** 语句比 Sybase IQ 更复杂。虽然在 Sybase IQ 中允许使用 Adaptive Server Enterprise 语法，但一些子句和关键字会被忽略。有关 Adaptive Server Enterprise **CREATE INDEX** 语句的完整语法，请参见《Adaptive Server Enterprise 参考手册 第二卷：命令》。

Adaptive Server Enterprise 索引可以是聚簇的或非聚簇的。聚簇索引检索数据几乎总是比非聚簇索引快。每个表只允许有一个聚簇索引。

Sybase IQ 不支持聚簇索引。SQL Anywhere 允许使用 **CLUSTERED** 和 **NONCLUSTERED** 关键字，但 Sybase IQ 会忽略它们。如果未指定 *index-type*，则 Sybase IQ 会对指定的列创建 **HG** 索引。

Sybase IQ 不允许使用 **DESC** 关键字。

对于 Sybase IQ 和 Adaptive Server Enterprise，给定表的索引名称必须都是唯一的。

权限

必须具有 **DBA** 权限或 **RESOURCE** 权限以及指定数据库空间中的 **CREATE** 权限才能创建索引。

另请参见

- **BEGIN PARALLEL IQ**……**END PARALLEL IQ** 语句（第 52 页）
- **CREATE JOIN INDEX** 语句（第 104 页）
- **DROP** 语句（第 170 页）
- **INDEX_PREFERENCE** 选项（第 382 页）
- **FROM** 子句（第 194 页）

CREATE JOIN INDEX 语句

创建连接索引（这可定义通过特定列预连接的一组表）以通过将表用在连接操作中来提高查询性能。

语法

```
CREATE JOIN INDEX join-index-name
    FOR join-clause
    IN dbspace-name
```


参数

- **join-clause:** - [() *join-expression* *join-type* *join-expression* [**ON** *search-condition*] [)]
- **join-expression:** - { *table-name* | *join-clause* }
- **join-type:** - [**NATURAL**] **FULL** [**OUTER**] **JOIN**
- **search-condition:** - [() *search-expression* [**AND** *search-expression*] [)]
- **search-expression:** - [() [*table-name.*] *column-name* = [*table-name.*] *column-name* [)]

示例

- **示例 1** - 使用 DepartmentID 列在 Departments 和 Employees 表之间创建连接索引，该列将作为 Departments 的主键和 Employees 的外键。

```
CREATE JOIN INDEX emp_dept_join
FOR Departments FULL OUTER JOIN Employees
ON Departments.DepartmentID = Employees.DepartmentID
```

- **示例 2** - 创建表 t1 和 t2，以后的数据分配将从缺省数据库空间进行，创建连接索引 t1t2，以后的数据分配将从数据库空间 Dsp6 进行。

```
CREATE TABLE t1(c1 int, c2 char(5));
CREATE TABLE t2(c1 int, c3 char(5));
CREATE JOIN INDEX t1t2 FOR t1
FULL OUTER JOIN t2 ON t2.c1=t1.c1 IN Dsp6;
```

用法

CREATE JOIN INDEX 在指定表的指定列上创建连接索引。一旦连接索引创建后，除非使用 **DROP JOIN INDEX** 将该索引删除或使用 **SYNCHRONIZE JOIN INDEX** 同步该索引，否则将不会再次引用该索引。

此语句仅支持连接类型 **FULL OUTER**；**OUTER** 关键字是可选的。

IN—指定连接索引的存放位置。如果省略 **IN** 子句，则 Sybase IQ 会在缺省 dbspace 中创建连接索引（由选项 **DEFAULT_DBSpace** 指定）。

ON—仅从两个表引用列。一个列集必须来自左侧子树中的单个表而另一个列集必须来自右侧子树中的某个表。支持的谓词仅为等值连接谓词。Sybase IQ 不允许单变量谓词、列内比较或不等值连接。

连接索引列必须具有相同的数据类型、精度和标度。

要指定由多部分组成的键，请包括链接由逻辑运算符 **AND** 连接的两个表的多个谓词。分离的 **ON** 子句不受支持；即 Sybase IQ 不允许连接谓词的逻辑运算符 **OR**。而且 **ON** 子句不接受标准的 **WHERE** 子句，因此无法指定别名。

可以使用 **NATURAL** 关键字，而非 **ON** 子句。**NATURAL** 连接是一种将列按名称进行组对并表示等值连接的连接。如果 **NATURAL** 连接生成涉及多个表对的谓词，则

CREATE JOIN INDEX 将返回错误。您可以指定 **NATURAL** 或 **ON**，但不要同时指定这两者。

CREATE JOIN INDEX 查找表中主外键关系 来确定一对多关系 的方向。（一对一关系的方向并不重要。）主键始终是“一”，而外键始终是“多”。如果未定义此信息，Sybase IQ 假定 左侧子树为“一” 而右侧子树为“多”。如果反过来，**CREATE JOIN INDEX** 将返回错误。

注意： 针对所有连接的查询优化 主要依赖基础主键。它们不需要外键。然而，您可以从使用外键中获益。如果您对负载进行设置以检查主键-外键关系，Sybase IQ 将会强制使用外键。

连接索引表必须为 Sybase IQ 基表。这些表不能为临时表、远程表或代理表。

基表上的多列索引在使用那些基表创建的连接索引中 不会进行复制。

星形连接索引是指其中位于星中心的单个表 以一对多关系与多个表相连接。要定义星形连接索引，必须先定义单列键 和主键，然后在 **CREATE JOIN INDEX** 语句中使用键连接语法。Sybase IQ 不支持将多个连接键列 用于任一连接的星形连接索引。

注意： 您必须先显式授予您的组中其他用户对基础“连接 虚拟表”的权限，然后这些用户 才能操作连接中的表。有关授予对连接虚拟表 的特权的信息，请参见《系统管理指南 第一卷》>“Sybase IQ 索引”>“使用连接索引”>“从连接索引表中进行插入或删除”。

另请参见《系统管理指南 第一卷》>“Sybase IQ 索引”。

副作用

- 自动提交

标准

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

权限

必须具有 DBA 权限或 RESOURCE 权限，是要进行连接的所有表的所有者，对数据库空间具有 CREATE 权限。

另请参见

- CREATE INDEX 语句（第 97 页）
- CREATE TABLE 语句（第 136 页）

CREATE LOGICAL SERVER 语句

创建用户定义的逻辑服务器。

语法

```
CREATE LOGICAL SERVER logical-server-name [ MEMBERSHIP
    '(' { ls-member, ... } ')' ]
```

参数

- **ls-member:** - **FOR LOGICAL COORDINATOR** | *mpx-server-name*

示例

- **示例 1** – 以下示例创建用户定义的逻辑服务器 *ls1*，其中包含三个 Multiplex 节点作为其成员。

```
CREATE LOGICAL SERVER ls1 MEMBERSHIP ( n1, n2, n3 )
```

用法

仅适用于 Multiplex。

Catalog 存储逻辑服务器及其成员资格定义。要向协调器定义逻辑成员资格，请在 MEMBERSHIP 子句中指定 FOR LOGICAL COORDINATOR。

如果在逻辑服务器创建过程中未指定成员时，则创建的逻辑服务器为空。

注意： 系统根本不会存储隐式逻辑服务器成员资格定义（如用于 OPEN、SERVER 逻辑服务器的定义）。

SYS.ISYSLOGICALSERVER 系统表存储有关逻辑服务器的信息。

SYS.ISYSLOGICALMEMBER 系统表存储有关逻辑服务器成员资格定义的信息。

logical-server-name 可以是除下列项目外的任意用户指定标识符：

- OPEN
- SERVER
- NONE
- 缺省值
- COORDINATOR
- ALL

将 root 逻辑服务器策略的 ALLOW_COORDINATOR_AS_MEMBER 选项从 ON 改为 OFF 并不会影响存储在 Catalog 中的成员资格信息。它仅影响逻辑服务器的有效配置。

即便在 `ALLOW_COORDINATOR_AS_MEMBER` 设置为 `OFF` 时，通过指定 `Multiplex` 服务器名或通过使用 `FOR LOGICAL COORDINATOR` 子句，都可以向当前协调器定义逻辑服务器成员资格。成员资格定义存储在 `Catalog`，但当 `Multiplex` 服务器作为协调器时，成员资格定义处于非活动状态。

权限

必须具有 `DBA` 或 `MULTIPLEX ADMIN` 权限。

CREATE LOGIN POLICY 语句

在数据库中创建登录策略。

语法

```
CREATE LOGIN POLICY policy-name policy-options
```

参数

- **policy-options:** - *policy-option* [*policy-option...*]
- **policy_option:** - *policy-option-name* = *policy-option-value* *policy-option-value* = { **UNLIMITED** | *legal-option-value* }

示例

- **示例 1** - 创建 `Test1` 登录策略。登录策略的口令有效期没有时间限制，允许用户最多尝试五次输入正确的口令，之后便会锁定帐户。

```
CREATE LOGIN POLICY Test1  
password_life_time=UNLIMITED  
max_failed_login_attempts=5;
```

用法

policy-name—登录策略的名称。

policy-option-name—登录策略选项的名称。如果未指定选项，则会应用 `root` 登录策略中的值。

policy-option-value—分配给登录策略选项的值。如果您指定 `UNLIMITED`，则不会强制执行任何限制。

如果不指定任何策略选项，则采用从根登录策略获得的登录策略值。

表 6. 登录策略选项

选项	描述	值	ROOT 策略的初始值	适用于
locked	如果此选项的值为 ON，则会禁止用户创建新连接	ON、OFF	OFF	仅限没有 DBA 权限的用户
max_connections	用户可进行的最大并发连接数。	0 - 2147483647	无限制	仅限没有 DBA 权限的用户
max_days_since_login	同一用户进行的两次连续登录之间可经过的最大天数。	0 - 2147483647	无限制	仅限没有 DBA 权限的用户
max_failed_login_attempts	自上一次成功登录尝试后，帐户锁定前可登录到用户帐户的最大失败尝试次数。	0 - 2147483647	无限制	仅限没有 DBA 权限的用户
max_non_dba_connections	没有 DBA 权限的用户可以进行的最大并发连接数。只有 root 登录策略支持此选项。	0 - 2147483647	无限制	仅限没有 DBA 权限的用户。仅适用于 root 登录策略。
password_expiry_on_next_login	如果此选项的值为 ON，则用户的口令将会在下次登录时过期。	ON、OFF	OFF	所有用户（包括那些具有 DBA 权限的用户）
password_grace_time	在口令到期前剩余的天数，在此期间允许登录，但缺省 post_login 过程将发出警告。	0 - 2147483647	0	所有用户（包括那些具有 DBA 权限的用户）
password_life_time	必须更改口令前的最大天数。	0 - 2147483647	无限制	所有用户（包括那些具有 DBA 权限的用户）

权限

必须具有 DBA 或 USER ADMIN 权限。

CREATE MESSAGE 语句 [T-SQL]

将用户定义消息添加到 SYSUSERMESSAGES 系统表，以供 **PRINT** 和 **RAISERROR** 语句使用。

语法

```
CREATE MESSAGE message-number  
... AS 'message-text'
```

用法

CREATE MESSAGE 将消息号与消息字符串关联。在 **PRINT** 和 **RAISERROR** 语句中可以使用消息号。

- **message_number**—要添加的消息的消息号。用户定义消息的消息号必须为 20000 或更大。
- **message_text**—要添加的消息的文本。最大长度是 255 字节。**PRINT** 和 **RAISERROR** 识别要输出的消息文本中的占位符。一条消息最多可包含 20 个任意顺序的、各不相同的占位符。消息的文本发送到客户端时，这些占位符将被替换为消息后面任意参数的格式化内容。

将消息翻译成一种具有不同语法结构的语言时，占位符进行编号以允许参数重排序。参数占位符的形式为 “%nn!”，即百分号 (%) 后跟一个 1 到 20 之间的整数，然后是一个感叹号 (!)，其中整数代表参数在参数列表中的位置。“%1!” 是第一个参数，“%2!” 是第二个参数，依此类推。

没有与 **sp_addmessage** 的 *language* 参数相对应的参数。

副作用

- 自动提交

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase — 在 Adaptive Server Enterprise 中，由 **sp_addmessage** 过程提供 **CREATE MESSAGE** 功能。

权限

必须具有 RESOURCE 权限。

另请参见

- **PRINT** 语句 [T-SQL] (第 253 页)

- RAISERROR 语句 [T-SQL] (第 256 页)

CREATE MULTIPLEX SERVER 语句

创建 Multiplex 服务器。

语法

变量声明:

```
CREATE MULTIPLEX SERVER server-name DATABASE 'dbfile
'host-port list [ ROLE { READER | WRITER } ] [ STATUS | {
INCLUDED | EXCLUDED } ]
```

参数

- **host-port-list:** - {[PRIVATE] HOST '*hostname*' PORT *port number* }

示例

- 示例 1 - 请参见“通过 Interactive SQL 添加 Multiplex 服务器”。

用法

仅适用于 Multiplex。

如果计划将 UNIX 软（符号）链接用于服务器路径，请在运行 **CREATE MULTIPLEX SERVER** 之前创建软链接。启动新服务器时，数据库文件路径必须匹配创建该服务器时指定的数据库文件路径。

根据服务器启动选项 **-n** 的规则选择 Multiplex 服务器的名称 (*server-name*)。请参见《实用程序指南》第 1 章“运行数据库服务器”中的“数据库服务器启动”。

创建初始 Multiplex 服务器时，协调节点行和辅助节点行将添加到 SYS.ISYSIQMPXSERVER。事务日志以两条单独 **CREATE MULTIPLEX SERVER** 命令的形式记录本操作，一条记录协调节点，一条记录辅助节点。

SYS.ISYSIQMPXSERVER 系统表将 HOST *hostname* PORT *portname* 对在其 connection_info 字符串中存储为 *host:port[:host:port...]*。

注意： 如果运行 Multiplex 服务器的计算机有多块冗余网卡映射到其它网络地址，则使用 Multiplex *host:port* 对。

可以按照任意顺序指定子句 DATABASE、*host-port list*、ROLE 和 STATUS。缺省 ROLE 是 READER。缺省 STATUS 是 INCLUDED。

host-port-list 关键字 PRIVATE 指定用于专用互连的特定 HOST PORT 对。一条单独专用互连，可提供 Multiplex 进程间通信 (MIPC) 所需的高可用性和高性能网络。Sybase

IQSybase IQ 自动打开专用端口；无需在启动服务器时 使用 `host-port-list` 将其列出。所有公共端口和专用端口都需要使用唯一端口号，以避免冲突。

添加服务器时，协调器必须正在运行，但您可以从 **Multiplex** 中的任何服务器运行 **CREATE MULTIPLEX SERVER** 命令。

此语句将自动提交。

权限

必须具有 DBA 或 MULTIPLEX ADMIN 权限。

CREATE PROCEDURE 语句

在数据库中创建新的用户定义 SQL 过程。

要创建外部过程接口，请参见“CREATE PROCEDURE 语句（外部过程）”。

语法

```
CREATE [ OR REPLACE | TEMPORARY ] PROCEDURE [ owner.]procedure-name
( [ parameter, ... ] ) {
[ RESULT ( result-column, ... ) | NO RESULT SET ]
[ SQL SECURITY { INVOKER | DEFINER } ]
[ ON EXCEPTION RESUME ] compound statement | AT location-string
```

参数

- **parameter:** - *parameter_mode**parameter-namedata-type* [**DEFAULT***expression*] | **SQLCODE** | **SQLSTATE**
- **parameter_mode:** - **IN** | **OUT** | **INOUT**
- **result-column:** - *column-namedata-type*

示例

- **示例 1** - 使用 case 语句对查询结果归类：

```
CREATE PROCEDURE ProductType (IN product_id INT, OUT type
CHAR(10))
BEGIN
    DECLARE prod_name CHAR(20) ;
    SELECT name INTO prod_name FROM "GROUP0"."Products"
    WHERE ID = product_id;
    CASE prod_name
    WHEN 'Tee Shirt' THEN
        SET type = 'Shirt'
    WHEN 'Sweatshirt' THEN
        SET type = 'Shirt'
    WHEN 'Baseball Cap' THEN
        SET type = 'Hat'
```



```

WHEN 'Visor' THEN
    SET type = 'Hat'
WHEN 'Shorts' THEN
    SET type = 'Shorts'
ELSE
    SET type = 'UNKNOWN'
END CASE ;
END

```

- **示例 2** – 使用游标并循环通过游标的行以返回单值：

```

CREATE PROCEDURE TopCustomer (OUT TopCompany CHAR(35), OUT
TopValue INT)
BEGIN
    DECLARE err_notfound EXCEPTION
        FOR SQLSTATE '02000' ;
    DECLARE curThisCust CURSOR FOR
    SELECT CompanyName, CAST(
        sum(SalesOrderItems.Quantity *
        Products.UnitPrice) AS INTEGER) VALUE
    FROM Customers
    LEFT OUTER JOIN SalesOrders
    LEFT OUTER JOIN SalesorderItems
    LEFT OUTER JOIN Products
    GROUP BY CompanyName ;

    DECLARE ThisValue INT ;
    DECLARE ThisCompany CHAR(35) ;
    SET TopValue = 0 ;
    OPEN curThisCust ;
    CustomerLoop:
    LOOP
        FETCH NEXT curThisCust
        INTO ThisCompany, ThisValue ;
        IF SQLSTATE = err_notfound THEN
            LEAVE CustomerLoop ;
        END IF ;
        IF ThisValue > TopValue THEN
            SET TopValue = ThisValue ;
            SET TopCompany = ThisCompany ;
        END IF ;
    END LOOP CustomerLoop ;
    CLOSE curThisCust ;
END

```

用法

CREATE PROCEDURE 在数据库中创建过程。具有 DBA 权限的用户可以通过指定 **owner** 为其他用户创建过程。过程由 **CALL** 语句激活。

注意： 创建存储过程有两种方式：ISO/ANSI SQL 和 T-SQL。例如，当使用 **CREATE PROCEDURE** 语法时，**BEGIN TRANSACTION** 特定于 T-SQL。请不要在创建存储过程时混合语法。请参见“**CREATE PROCEDURE 语句 [T-SQL]**”。

CREATE PROCEDURE—可以创建永久的或临时的 (TEMPORARY) 存储过程。可以将 **PROC** 用作 **PROCEDURE** 的同义词。

参数名必须符合其它数据库标识符（如列名）的规则，而且必须是有效的 SQL 数据类型。请参见《参考：构件块、表和过程》>“SQL 数据类型”。这些关键字具有以下含义：

参数可以使用以下关键字之一作为前缀：IN、OUT 和 INOUT。如果不指定关键字，参数在缺省情况下为 INOUT。这些关键字具有以下含义：

- IN—此参数是一个为过程提供值的表达式。
- OUT—此参数是一个可由过程赋值的变量。
- INOUT—此参数是一个为过程提供值的变量，并且可由过程赋值。

使用 **CALL** 执行过程时，不需要指定所有的参数。如果在 **CREATE PROCEDURE** 语句中提供了缺省值，则系统会为缺少的参数指派缺省值。如果 **CALL** 语句中未提供参数，且缺省值未设置，则会给出错误。

SQLSTATE 和 **SQLCODE** 是特殊参数，它们在过程结束时输出 **SQLSTATE** 或 **SQLCODE** 值（它们是 **OUT** 参数）。无论是否指定 **SQLSTATE** 和 **SQLCODE** 参数，总是可以在过程调用后立即检查 **SQLSTATE** 和 **SQLCODE** 特殊值以测试过程的返回状态。

SQLSTATE 和 **SQLCODE** 特殊值会由下一个 SQL 语句修改。如果将 **SQLSTATE** 或 **SQLCODE** 作为过程参数提供，则允许返回代码存储在变量中。

指定 **CREATE OR REPLACE PROCEDURE** 将创建一个新过程或替换同名的现有过程。此子句更改过程的定义，但保留现有权限。不能将 **OR REPLACE** 子句与临时过程一起使用。另外，如果被替换的过程已在使用中，则会返回错误。

如果指定 **CREATE TEMPORARY PROCEDURE**，则将意味着该存储过程仅对创建它的连接可见，并在删除该连接时随之自动删除。您也可以显式删除临时存储过程。您无法在临时存储过程上执行 **ALTER**、**GRANT** 或 **REVOKE**，而且与其它存储过程不同，临时存储过程不会在目录或事务日志中予以记录。

临时过程使用其创建者（当前用户）或指定所有者的权限执行。可以在以下情况下为临时过程指定所有者：

- 临时过程是在永久存储过程中创建的
- 临时和永久过程有同样的所有者

要删除临时过程的所有者，请先删除临时过程。

当您与只读数据库连接时，可以创建和删除临时存储过程；它们不能是外部过程。

例如，以下临时存储过程会删除名为 **CustRank** 的表（如果此表存在）。对于此示例，过程假定表名是唯一的，可以由过程创建者在不指定表所有者的情况下引用：

```
CREATE TEMPORARY PROCEDURE drop_table( IN @TableName char(128) )
BEGIN
    IF EXISTS ( SELECT * FROM SYS.SYSTAB WHERE
        table_name = @TableName )
    THEN EXECUTE IMMEDIATE
        'DROP TABLE "' || @TableName || '"';
    MESSAGE 'Table "' || @TableName ||
        '" dropped' to client;
```

```

END IF;
END;
CALL drop_table( 'CustRank' )

```

RESULT 声明结果集中的列的数量和类型。**RESULT** 关键字后面括在括号内的列表定义结果的列名和类型。描述 **CALL** 语句时，嵌入式 **SQL DESCRIBE** 或 **ODBC SQLDescribeCol** 会返回此信息。《参考：构件块、表和过程》>“SQL 数据类型”中列出了允许使用的数据类型。

有关从过程返回的结果集的详细信息，请参见《系统管理指南第二卷》>“使用过程和批处理”。

因执行的方式不同，有些过程可生成多个结果集。例如，下面的过程在有些情况下返回两列，在有些情况下返回一列。

```

CREATE PROCEDURE names( IN formal char(1))
BEGIN
    IF formal = 'n' THEN
        SELECT GivenName
        FROM Employees
    ELSE
        SELECT Surname, GivenName
        FROM Employees
    END IF
END

```

包含可变结果集的过程必须用不带 **RESULT** 子句的语句编写，或者用 Transact-SQL 编写。它们的使用受以下限制：

- 嵌入式 SQL—您必须在打开用于结果集的游标之后、但在返回任何行之前 **DESCRIBE** (描述) 过程调用，才能获得正确形式的结果集。需要 **DESCRIBE** 语句的 **CURSOR cursor-name** 子句。
- ODBC、OLE DB、ADO.NET—可变结果集过程可由 ODBC 应用程序使用。结果集的正确描述由驱动程序或提供程序完成。
- Open Client 应用程序—Open Client 应用程序可以使用可变结果集过程。

如果过程仅返回一个结果集，则应使用 **RESULT** 子句。该子句可防止 ODBC 和 Open Client 应用程序在游标打开后再次描述结果集。

要处理多个结果集，ODBC 必须描述当前正在执行的游标，而不是描述由过程定义的结果集。因此，ODBC 并不总是按照过程定义的 **RESULT** 子句中的定义来描述列名称。为避免这种问题，请在生成结果集的 **SELECT** 语句中使用列的别名。

NO RESULT SET 一声明此过程不返回结果集。当外部环境需要知道某个过程不返回结果集时，这是非常有用的。

SQL SECURITY—定义是作为 **INVOKER** (调用过程的用户) 还是作为 **DEFINER** (拥有过程的用户) 执行过程。缺省为 **DEFINER**。

如果指定了 **SQL SECURITY INVOKER**，将使用更多内存，因为必须为调用过程的每个用户进行注释。此外，还将以调用者身份执行名称解析。因此，请使用所有对象名(表、过程等)的相应所有者限定这些名称。例如，假定 user1 创建以下过程：

```
CREATE PROCEDURE user1.myProcedure()
  RESULT( columnA INT )
  SQL SECURITY INVOKER
BEGIN
  SELECT columnA FROM table1;
END;
```

如果 user2 尝试运行此过程，而且表 user2.table1 不存在，则系统会产生表查找错误。此外，如果 user2.table1 存在，则系统会使用该表，而不使用预期的 user1.table1。为防止出现这种情况，请在语句中限定表引用 (user1.table1，而不是仅 table1)。

如果使用 **ON EXCEPTION RESUME**，则过程会根据 **ON_TSQL_ERROR** 选项的设置来执行操作。如果 **ON_TSQL_ERROR** 设置为 **CONDITIONAL** (缺省值)，则会在下一条语句能处理错误时继续执行该语句；否则，便会退出。

错误处理语句包括：

- **IF**
- **SELECT @variable =**
- **CASE**
- **LOOP**
- **LEAVE**
- **CONTINUE**
- **CALL**
- **EXECUTE**
- **SIGNAL**
- **RESIGNAL**
- **DECLARE**
- **SET VARIABLE**

请勿将显式错误处理代码与 **ON EXCEPTION RESUME** 子句一起使用。

请参见“**ON_TSQL_ERROR** 选项 [TSQL]”。

AT location-string—在当前数据库中为 *location-string* 指定的远程过程创建代理存储过程。在 *location-string* 中，**AT** 子句支持将分号 (;) 用作分隔符。如果没有分号，则使用句点作为字段分隔符。这样即可在数据库和所有者字段中使用文件名和扩展名。

远程过程输出变量中最多只能返回 254 个字符。

如果远程过程可以返回结果集，但在所有情况下都能返回，则本地过程定义必须包含 **RESULT** 子句。

有关远程服务器的信息，请参见“**CREATE SERVER** 语句”。有关使用远程过程的信息，请参见《系统管理指南 第二卷》>“访问远程数据”>“Sybase IQ 和远程数据”>“远程过程调用 (RPC)”。

注意：当删除和创建过程时，在 Sybase IQ 之前创建的数据库可能最终达到 **proc_id** 的最大限制 32767，并导致 **CREATE PROCEDURE** 在 Sybase IQ 12.6 中返回“项已存

在”错误。有关解决方法，请参见《系统管理指南第一卷》>“故障排除提示”>“特定情况的解决方案”>“资源问题”>“过程标识符不足”。

副作用

- 自动提交

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase—Transact-SQL **CREATE PROCEDURE** 语句与此不同。
- SQLJ—提议的 SQLJ1 标准中指定了 Java 结果集的语法扩展。

权限

必须具有 **RESOURCE** 权限，否则只能创建临时过程。对于外部过程，或要为另一用户创建过程，则必须具有 **DBA** 权限。

另请参见

- 在过程中引用临时表（第 117 页）
- **BEGIN**……**END** 语句（第 49 页）
- **CALL** 语句（第 55 页）
- **CREATE PROCEDURE** 语句 [T-SQL]（第 118 页）
- **CREATE PROCEDURE** 语句（外部过程）（第 120 页）
- **CREATE SERVER** 语句（第 132 页）
- **DROP** 语句（第 170 页）
- **EXECUTE IMMEDIATE** 语句 [ESQL] [SP]（第 185 页）
- **GRANT** 语句（第 202 页）
- **RAISERROR** 语句 [T-SQL]（第 256 页）
- **ON_TSQL_ERROR** 选项 [TSQL]（第 412 页）

在过程中引用临时表

如果临时表定义不一致，则在过程之间共享该表会导致出现问题。

例如，假设有两个过程 `procA` 和 `procB`，这两个过程均定义临时表 `temp_table`，并调用名为 `sharedProc` 的另一过程。即未调用 `procA`，也未调用 `procB`，所以临时表还不存在。

现在假设 `temp_table` 在 `procA` 中的定义与在 `procB` 中的定义略有不同，而两个过程均使用相同的列名称和类型，但列顺序不同。

调用 `procA` 时，它返回预期结果。但调用 `procB` 时，它返回不同的结果。

这是因为，调用 `procA` 时，它创建 `temp_table`，然后调用 `sharedProc`。调用 `sharedProc` 时，会解析并验证其中的 **SELECT** 语句，然后缓存该语句解析后的表

示形式，这样执行另一 **SELECT** 语句时就可以再次使用该语句。缓存的形式可反映 `procA` 中表定义的列顺序。

调用 `procB` 时将重新创建 `temp_table`，但列顺序不同。`procB` 调用 `sharedProc` 时，数据库服务器使用 **SELECT** 语句的缓存表示形式。因此，结果不同。

可通过执行以下某项操作避免出现这种问题：

- 确保以此方式使用的临时表定义一致
- 考虑改用全局临时表

CREATE PROCEDURE 语句 [T-SQL]

创建与 Adaptive Server Enterprise 兼容 的新过程。

语法

Sybase IQ 支持 Transact-SQL **CREATE PROCEDURE** 语句的 下列子集：

```
CREATE [ OR REPLACE ] PROCEDURE [ owner.]procedure_name
... [ [ ( ] @parameter_name data-type [ = default ] [ OUTPUT ] [ , ... ]
[ ) ] ]
...[ WITH RECOMPILE ]
... AS
... statement-list
```

用法

Transact-SQL 和 Sybase IQ SQL 语句之间的差异：

- 指定 **CREATE OR REPLACE PROCEDURE** 将创建 一个新过程或替换同名的 现有过程。此子句更改过程的定义， 但保留现有权限。另外，如果被替换的过程已在使用中， 则会返回错误。
- 带 @ 前缀的变量名 — “@” 符表示 Transact-SQL 变量名，而 Sybase IQ 变量可以是任何有效的标识符，也可以带 @ 前缀。
- 输入和输出参数 — Sybase IQ 过程参数 指定为 **IN**、**OUT** 或 **INOUT**；而 Transact-SQL 过程参数在缺省情况下为 **INPUT** 参数，也可以指定为 **OUTPUT**。在 Sybase IQ 中声明为 **INOUT** 或 **OUT** 的参数在 Transact-SQL 中应使用 **OUTPUT** 进行声明。
- 参数缺省值—Sybase IQ 使用关键字 **DEFAULT** 为过程参数提供缺省值，而 Transact-SQL 使用等号 (=) 提供 缺省值。
- 返回结果集—Sybase IQ 使用 **RESULT** 子句 指定返回的结果集。在 Transact-SQL 过程中， 第一个查询的列名或别名将返回到 调用环境中：

```
CREATE PROCEDURE showdept @deptname varchar(30)
AS
  SELECT Employees.Surname, Employees.givenName
  FROM Departments, Employees
  WHERE Departments.DepartmentName = @deptname
```

```
AND Departments.DepartmentID =
    Employees.DepartmentID
```

对应的 Sybase IQ 过程：

```
CREATE PROCEDURE showdept(in deptname
    varchar(30) )
RESULT ( lastname char(20), firstname char(20))
ON EXCEPTION RESUME
BEGIN
    SELECT Employees.SurName, Employees.GivenName
    FROM Departments, Employees
    WHERE Departments.DepartmentName = deptname
    AND Departments.DepartmentID =
        Employees.DepartmentID
END
```

- 过程主体—Transact-SQL 过程的主体 是一系列以关键字 **AS** 开始的 Transact-SQL 语句。 Sybase IQ 过程的主体是用关键字 **BEGIN** 和 **END** 括起来的 复合语句。

注意： 创建存储过程有两种方式：T-SQL 和 SQL/92。例如，当使用 **CREATE PROCEDURE** 语法时，**BEGIN TRANSACTION** 特定于 T-SQL。请不要在创建存储过程时混合语法。

副作用

- 自动提交

标准

- SQL - ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase—Sybase IQ 支持 Adaptive Server Enterprise **CREATE PROCEDURE** 语句语法的子集。
如果提供了 Transact-SQL **WITH RECOMPILE** 可选子句，它将被忽略。数据库启动后，当过程第一次执行时，SQL Anywhere 总是重新编译它们，并存储编译的过程直到数据库停止。
不支持过程组。

权限

必须具有 **RESOURCE** 权限。

另请参见

- CREATE PROCEDURE** 语句 （第 112 页）

CREATE PROCEDURE 语句 (外部过程)

创建本机或外部过程的接口。

要创建 SQL 过程，请参见“CREATE PROCEDURE 语句”。要创建表 UDF 或 TPF，请参见“CREATE PROCEDURE 语句 (表 UDF)”。

语法

```
CREATE[ OR REPLACE ] PROCEDURE [ owner.]procedure-name
( [ parameter, ...] )
[ RESULT ( result-column, ...) | NO RESULT SET ]
[ DYNAMIC RESULT SETS integer-expression ]
[ SQL SECURITY { INVOKER | DEFINER } ]
[ EXTERNAL NAME 'external-call' [ LANGUAGE environment-name ]
```

参数

- **parameter:** - *parameter_mode**parameter-namedata-type* [**DEFAULT***expression*] | **SQLCODE** | **SQLSTATE**
- **parameter_mode:** - **IN** | **OUT** | **INOUT**
- **result-column:** - *column-namedata-type*
- **environment-name:** - **C_ESQL32** | **C_ESQL64** | **C_ODBC32** | **C_ODBC64** | **CLR** | **JAVA** | **PERL** | **PHP**

用法

过程的主体由复合语句组成。有关复合语句的信息，请参见“BEGIN……END 语句”。

注意：创建存储过程有两种方式：ISO/ANSI SQL 和 T-SQL。例如，当使用 **CREATE PROCEDURE** 语法时，**BEGIN TRANSACTION** 特定于 T-SQL。请不要在创建存储过程时混合语法。请参见“CREATE PROCEDURE 语句 [T-SQL]”。

CREATE PROCEDURE 在数据库中创建过程。具有 DBA 权限的用户可以通过指定 **owner** 为其他用户创建过程。过程由 **CALL** 语句激活。

如果存储过程返回一个结果集，则该函数也无法设置输出参数或返回返回值。

当从多个过程中引用临时表时，如果临时表定义不一致并且引用该表的语句被高速缓存，则可能会发生潜在问题。

可以创建这样的永久存储过程：它们调用用多种编程语言编写的外部或本机过程。可以将 **PROC** 用作 **PROCEDURE** 的同义词。

- **参数名称** - 参数名称必须符合其它数据库标识符（如列名）的规则。它们必须是有效的 SQL 数据类型。请参见《参考：构件块、表和过程》>“SQL 数据类型”。

参数可以使用以下关键字之一作为前缀：**IN**、**OUT** 或 **INOUT**。如果不指定这些值中的任何一个，参数在缺省情况下为 **INOUT**。关键字表示：

- **IN**—此参数是一个为过程提供值的表达式。
- **OUT**—此参数是一个可由过程赋值的变量。
- **INOUT**—此参数是一个为过程提供值的变量，并且可由过程赋值。

注意： **TABLE** 参数不能声明为 **INOUT** 或 **OUT**。请参见“**CREATE PROCEDURE** 语句（表 UDF）”。

使用 **CALL** 执行过程时，不需要指定所有的参数。如果在 **CREATE PROCEDURE** 语句中提供了缺省值，则系统会为缺少的参数指派缺省值。如果 **CALL** 语句中没有提供参数并且没有设置缺省值，则发生错误。

注意： 无法调用表 UDF。请参见“**CREATE PROCEDURE** 语句（表 UDF）”。

SQLSTATE 和 **SQLCODE** 是特殊 **OUT** 参数，它们在过程结束时输出 **SQLSTATE** 或 **SQLCODE** 值。无论是否指定 **SQLSTATE** 和 **SQLCODE** 参数，总是可以在过程调用后立即检查 **SQLSTATE** 和 **SQLCODE** 特殊值以测试过程的返回状态。

SQLSTATE 和 **SQLCODE** 特殊值会由下一个 SQL 语句修改。如果将 **SQLSTATE** 或 **SQLCODE** 作为过程参数提供，则允许返回代码存储在变量中。

- **OR REPLACE** – 指定 **OR REPLACE (CREATE OR REPLACE PROCEDURE)** 将创建一个新过程或替换同名的现有过程。此子句更改过程的定义，但保留现有权限。如果您尝试替换已在使用中的过程，则会返回错误。

您不能创建 **TEMPORARY** 外部调用过程。

有关从过程返回的结果集的详细信息，请参见《系统管理指南 第二卷》>“使用过程和批处理”。

- **RESULT** – 声明结果集中的列的数量和类型。**RESULT** 关键字后面括在括号内的列表定义结果的列名和类型。描述 **CALL** 语句时，嵌入式 **SQL DESCRIBE** 或 **ODBC SQLDescribeCol** 会返回此信息。《参考：构件块、表和过程》>“SQL 数据类型”中列出了允许使用的数据类型。

调用嵌入式 SQL (**LANGUAGE C_ESQL32**、**LANGUAGE C_ESQL64**) 或 **ODBC** (**LANGUAGE C_ODBC32**、**LANGUAGE C_ODBC64**) 外部函数的过程可以返回 0 个或 1 个结果集。

调用 **Perl** 或 **PHP** (**LANGUAGE PERL**、**LANGUAGE PHP**) 外部函数的过程无法返回结果集。调用由数据库服务器装载的本地函数的过程无法返回结果集。

调用 **CLR** 或 **Java** (**LANGUAGE CLR**、**LANGUAGE JAVA**) 外部函数的过程可以返回 0 个、1 个或更多结果集。

- **NO RESULT SET** – 声明此过程不返回结果集。当外部环境需要知道某个过程不返回结果集时，这是非常有用的。
- **DYNAMIC RESULT SETS** – 将此子句与 **LANGUAGE CLR** 和 **LANGUAGE JAVA** 调用合用。此子句仅在指定 **LANGUAGE** 的情况下有意义。如果指定了 **RESULT** 子句，则 **DYNAMIC RESULT SETS** 缺省值为 1。如果未指定 **RESULT** 子句，则

DYNAMIC RESULT SETS 缺省值为 0。请注意，调用 Perl 或 PHP (LANGUAGE PERL、LANGUAGE PHP) 外部函数的过程无法返回结果集。调用由数据库服务器装载的本地函数的过程 无法返回结果集。

- **SQL SECURITY** – 定义是作为 INVOKER (调用过程的用户) 还是作为 DEFINER (拥有过程的用户) 执行过程。缺省值为 DEFINER。对于外部调用，此子句在外部环境 中 为非限定对象引用建立所有权 上下文。

如果指定了 SQL SECURITY INVOKER，将使用更多内存，因为必须为调用过程的每个用户进行注释。此外，如果指定了 SQL SECURITY INVOKER，也将以调用者身份执行名称解析。因此，应注意 利用所有对象的相应所有者限定这些对象的名称 (表、过程等)。例如，假定 user1 创建以下过程：

```
CREATE PROCEDURE user1.myProcedure()
  RESULT( columnA INT )
  SQL SECURITY INVOKER
  BEGIN
    SELECT columnA FROM table1;
  END;
```

如果 user2 尝试运行此过程，而且表 user2.table1 不存在，则系统会产生表查找错误。此外，如果 user2.table1 存在，则系统会使用该表，而不使用预期的 user1.table1。为防止出现这种情况，请在语句中限定表引用 (user1.table1，而不是仅 table1)。

- **EXTERNAL NAME LANGUAGE** – EXTERNAL NAME LANGUAGE ‘native-call’ *native-call:[operating-system:]function-name@library, ...*

将 **EXTERNAL NAME** 与 **LANGUAGE JAVA** 子句一起使用的过程是包含 Java 方法的包装。

operating-system：UNIX—使用不带 LANGUAGE 属性的 **EXTERNAL NAME** 子句的过程可定义用某种编程语言 (如 C) 编写的本地函数的接口。本地函数由数据库服务器装载到其地址 空间中。

库名可包含文件扩展名，在 Windows 中通常为 .dll，在 UNIX 中通常为 .so。在没有扩展名 的情况下，该软件附加平台特定的 缺省库文件扩展名。以下是规范示例：

```
CREATE PROCEDURE mystring( IN instr LONG VARCHAR )
  EXTERNAL NAME
  'mystring@mylib.dll;Unix:mystring@mylib.so';
```

更简单的编写前一个 **EXTERNAL NAME** 子句的方法 (使用特定于平台的缺省设置)：

```
CREATE PROCEDURE mystring( IN instr LONG VARCHAR )
  EXTERNAL NAME 'mystring@mylib';
```

在被调用时，函数所在的库 会装载到数据库服务器的地址空间中。本地函数将作为服务器的一部分执行。这种情况下，如果函数引发错误，数据库服务器就会终止。

因此，建议使用 **LANGUAGE** 属性在外部环境中装载和执行函数。如果函数在外部环境中引发错误，数据库服务器将继续运行。

有关本地库调用的信息，请参见“SQL Anywhere Server - 编程” > “SQL Anywhere 外部调用接口”。

注意： 此参考指向 SQL Anywhere 文档。

EXTERNAL NAME LANGUAGE ‘c-call’ **LANGUAGE** { C_ESQL32 | C_ESQL64 | C_ODBC32 | C_ODBC64 } *c-call*:

[*operating-system*:]*function-name*@*library*, ...

operating-system: **UNIX**

要在外部环境中（而不是数据库服务器中）调用已编译的本机 C 函数，就要使用 **EXTERNAL NAME** 子句后跟 **LANGUAGE** 属性（指定 C_ESQL32、C_ESQL64、C_ODBC32 和 C_ODBC64 之一）来定义存储过程或函数。

如果指定 **LANGUAGE** 属性，函数所在的库就会由外部进程装载，而且外部函数将作为该外部进程的一部分执行。这种情况下，如果函数引发错误，数据库服务器将继续运行。

示例过程定义：

```
CREATE PROCEDURE ODBCinsert(
  IN ProductName CHAR(30),
  IN ProductDescription CHAR(50)
)
NO RESULT SET
EXTERNAL NAME 'ODBCexternalInsert@extodbc.dll'
LANGUAGE C_ODBC32;
```

请参见“SQL Anywhere Server - 编程” > “SQL Anywhere 外部环境支持” > “ESQL 和 ODBC 外部环境”。

注意： 此参考指向 SQL Anywhere 文档。

EXTERNAL NAME ‘clr-call’ **LANGUAGE CLR** *clr-call*: *dll-name*::*function-name* (*param-type-1*, ...)

operating-system: **UNIX**

要在外部环境中调用 .NET 函数，过程接口应使用后跟 **LANGUAGE CLR** 属性的 **EXTERNAL NAME** 子句来定义。

CLR 存储过程或函数在行为上与 SQL 存储过程或函数相同，不同之处是该过程或函数的代码是用 .NET 语言（如 C# 或 Visual Basic）编写的，而且该过程或函数的执行发生在数据库服务器之外（即，在单独的 .NET 可执行文件内）。

示例过程定义：

```
CREATE PROCEDURE clr_interface(
  IN p1 INT,
```

```

IN p2 UNSIGNED SMALLINT,
OUT p3 LONG VARCHAR)
NO RESULT SET
EXTERNAL NAME 'CLRlib.dll::CLRproc.Run( int, ushort, out
string )'
LANGUAGE CLR;

```

请参见“SQL Anywhere Server – 编程” > “SQL Anywhere 外部环境支持” > “CRL 外部环境”。

注意： 此参考指向 SQL Anywhere 文档。

EXTERNAL NAME ‘perl-call’ LANGUAGE CLR *perl-call*:

< file=*perl-call* > **\$sa_perl_return=perl-sub** (**\$sa_perl_arg0**, ...)

要在外部环境中调用 Perl 函数，就要使用 **EXTERNAL NAME** 子句后跟 LANGUAGE PERL 属性来定义过程 接口。

Perl 存储过程或函数在行为上与 SQL 存储过程或函数相同，不同之处是该过程或函数的代码是用 Perl 编写的，而且该过程或函数的执行发生在数据库服务器之外（即，在 Perl 可执行实例内）。

示例过程定义：

```

CREATE PROCEDURE PerlWriteToConsole( IN str LONG VARCHAR)
NO RESULT SET
EXTERNAL NAME '<file=PerlConsoleExample>
WriteToServerConsole( $sa_perl_arg0 )'
LANGUAGE PERL;

```

请参见“SQL Anywhere Server – 编程” > “SQL Anywhere 外部环境支持” > “PERL 外部环境”。

注意： 此参考指向 SQL Anywhere 文档。

EXTERNAL NAME ‘perl-call’ LANGUAGE PHP <file=*php-file*> **print php-func** (**\$argv[1]**, ...)

要在外部环境中调用 PHP 函数，就要使用 **EXTERNAL NAME** 子句后跟 LANGUAGE PHP 属性来定义过程接口。

PHP 存储过程或函数在行为上与 SQL 存储过程或函数相同，不同之处是该过程或函数的代码是用 PHP 编写的，而且该过程或函数的执行发生在数据库服务器之外（即，在 PHP 可执行实例内）。

示例过程定义：

```

CREATE PROCEDURE PHPPopulateTable()
NO RESULT SET
EXTERNAL NAME '<file=ServerSidePHPExample>
ServerSidePHPSub()'
LANGUAGE PHP;

```

请参见“SQL Anywhere Server – 编程” > “SQL Anywhere 外部环境支持” > “PHP 外部环境”。

注意： 此参考指向 SQL Anywhere 文档。

```
EXTERNAL NAME java-call LANGUAGE JAVA ‘java-call ‘ [ package-name. ]
class-name.method-name ( method-signature
method-signature: ([field-descriptor, ... ] ) return-descriptor
```

Java 方法签名是参数类型和 返回值类型的压缩字符 表示形式。

要在外部环境中调用 Java 方法，就要使用 **EXTERNAL NAME** 子句后跟 LANGUAGE JAVA 属性来定义 过程接口。

Java 接口存储过程或函数在行为上与 SQL 存储过程或函数相同，不同之处是该过程或函数的代码是用 Java 编写的，而且该过程或函数的执行发生在 数据库服务器之外（即在 Java 虚拟机内）。

示例过程定义：

```
CREATE PROCEDURE HelloDemo( IN
name LONG VARCHAR )
NO RESULT SET
EXTERNAL NAME 'Hello.main([Ljava/lang/String;)V'
LANGUAGE JAVA;
```

请参见“SQL Anywhere Server – 编程” > “SQL Anywhere 外部环境支持” > “Java 外部环境”。

注意： 此参考指向 SQL Anywhere 文档。

表 7. Java Field-descriptor 和 Return-descriptor

字段类型	Java 数据类型
B	byte
C	char
D	double
F	float
I	int
J	long
Lclass-name;	class-name 类的实例。类名必须是完全限定的，而且名称中的任何点 都必须由反斜杠代替。例如，java/lang/String
S	short
V	无类型

字段类型	Java 数据类型
Z	boolean
I	数组的每个维度都使用一个

例如：

```
double some_method(  
    boolean a,  
    int b,  
    java.math.BigDecimal c,  
    byte [][] d,  
    java.sql.ResultSet[] d ) {  
}
```

具有以下签名：

```
'(ZILjava/math/BigDecimal;[[B[Ljava/sql/ResultSet;)D'
```

副作用

- 自动提交

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase—Transact-SQL **CREATE PROCEDURE** 语句与此不同。
- SQLJ—提议的 SQLJ1 标准中指定了 Java 结果集的语法扩展。

权限

必须具有 **RESOURCE** 权限，否则只能创建临时过程。对于外部过程，或要为另一用户创建过程，则必须具有 **DBA** 权限。

另请参见

- ALTER PROCEDURE 语句 （第 25 页）
- BEGIN……END 语句 （第 49 页）
- CALL 语句 （第 55 页）
- CREATE PROCEDURE 语句 （第 112 页）
- CREATE PROCEDURE 语句 [T-SQL] （第 118 页）
- DROP 语句 （第 170 页）
- EXECUTE IMMEDIATE 语句 [ESQL] [SP] （第 185 页）
- GRANT 语句 （第 202 页）

CREATE PROCEDURE 语句 (Java UDF)

创建外部 Java 表 UDF 的接口。

外部过程的 **CREATE PROCEDURE** 语句参考信息位于单独的主题中。表 UDF 的 **CREATE PROCEDURE** 语句参考信息位于单独的主题中。

如果您的查询引用 Sybase IQ 表，请注意相对于仅引用 Catalog 存储表的查询，该查询要应用不同的语法和参数。

仅在 **FROM** 子句中支持 Java 表 UDF。

语法

对于至少引用一个 Sybase IQ 表的查询：

```
CREATE[ OR REPLACE ] PROCEDURE
[ owner.]procedure-name ( [ parameter, ... ] )
[ RESULT (result-column, ...)]
[ SQL SECURITY { INVOKER | DEFINER } ]
EXTERNAL NAME 'java-call' [ LANGUAGE Java ] }
```

对于仅引用 Catalog 存储表的查询：

```
CREATE[ OR REPLACE ] PROCEDURE
[ owner.]procedure-name ( [ parameter, ... ] )
[ RESULT (result-column, ...)]
| NO RESULT SET
[ DYNAMIC RESULT SETS integer-expression ]
[ SQL SECURITY { INVOKER | DEFINER } ]
EXTERNAL NAME 'java-call' [ LANGUAGE Java ] }
```

参数

- **参数** - 对于至少引用一个 Sybase IQ 表的查询：

[**IN***parameter_modeparameter-namedata-type* [**DEFAULT***expression*]

对于仅引用 Catalog 存储表的查询：

[**IN** | **OUT** | **INOUT**] *parameter_modeparameter-namedata-type* [**DEFAULT***expression*]

- **result-column** - *column-name data-type*
- **JAVA** - [**ALLOW** | **DISALLOW SERVER SIDE REQUESTS**]
- **java-call** - '[*package-name.class-name.method-name method-signature*]'

用法

对于 Java 表函数，只允许一个结果集。如果 Java 表函数使用 Sybase IQ 表连接或 Sybase IQ 表中某个列是 Java 表函数的参数，则仅支持一个结果集。

如果 Java 表函数是 **FROM** 子句中唯一的项目，则允许 *N* 个结果集。

JAVA: [**ALLOW** | **DISALLOW SERVER SIDE REQUESTS**]:

DISALLOW 为缺省值。

ALLOW 指示允许服务器端连接。

注意：除非有必要，否则不要指定 **ALLOW**。**ALLOW** 的设置会减慢特定类型的 Sybase IQ 表连接。如果将过程定义从 **ALLOW** 改为 **DISALLOW**（反之亦然），则直到建立新的连接后，才会确认更改。

不要在同一查询中将 UDF 与 **ALLOW SERVER SIDE REQUESTS** 和 **DISALLOW SERVER SIDE REQUESTS** 同时使用。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase—Transact-SQL **CREATE PROCEDURE** 语句与此不同。
- SQLJ—提议的 SQLJ1 标准中指定了 Java 结果集的语法扩展。

权限

必须具有 **RESOURCE** 权限，否则只能创建临时过程。具有 **DBA** 权限的用户可以通过指定 **owner** 为其他用户创建 UDF。用户必须具有 **DBA** 权限才能创建 外部 UDF，或者为另一用户创建外部 UDF。

CREATE PROCEDURE 语句（表 UDF）

创建外部表用户定义函数的接口（表 UDF）。用户必须经过专门授权才能使用表 UDF。

使用 **a_v4_extfn** API 定义表 UDF。针对不使用 **a_v3_extfn** 或者 **a_v4_extfn** API 的外部过程，请于相关专题中获取 **CREATE PROCEDURE** 语句的参考信息。针对 Java UDF，请于相关专题中获取 **CREATE PROCEDURE** 语句的参考信息。

语法

```
CREATE[ OR REPLACE ] PROCEDURE
[ owner.]procedure-name ( [ parameter[, ...]] )
| RESULT result-column [, ...] )
[ SQL SECURITY { INVOKER | DEFINER } ]
EXTERNAL NAME 'external-call'
```

参数

- 参数： - [**IN**] *parameter-namedata-type* [**DEFAULT** *expression*]
 | [**IN**] *parameter-nametable-type*
- 表类型： - **TABLE**(*column-namedata-type* [, ...])

- 外部调用： - `[column-name:]function-name@library, ...`

用法

语句 **CREATE PROCEDURE** 语句在数据库中 创建过程。具有 **DBA** 权限的用户可以通过指定所有者 为其他用户创建过程。

如果存储过程返回一个结果集，则该函数也无法 设置输出参数或返回返回值。

当从多个过程中引用临时表时， 如果临时表定义不一致并且引用该表的语句被高速缓存， 则可能会发生潜在问题。在过程中引用临时表时应小心谨慎。

您可以使用 **CREATE PROCEDURE** 语句，以创建使用不同于 **SQL** 的编程语言 实现的外部表 **UDF**。然而，在创建外部 **UDF** 之前 请注意表 **UDF** 的限制。

适用于标量参数、结果列和 **TABLE** 参数列 的数据类型必须为有效的 **SQL** 数据类型。

参数名必须符合其它数据库 标识符（如列名）的规则。它们必须是有效的 **SQL** 数据类型。

TPF 支持混合标量参数以及单个表参数。 **TABLE** 参数必须为输入行组定义模式， 输入行组则由 **UDF** 进行处理。 **TABLE** 参数的定义 包括列名和列数据类型。

```
TABLE(c1 INT, c2 CHAR(20))
```

上述示例定义了一个包含两列的模式， 分别是 **INT** 类型的 **c1** 列 和 **CHAR(20)** 类型的 **c2** 列。 **UDF** 所处理的每一行 都必须为包含两个值的元组。与标量参数不同， 无法为表参数分配默认值。

- **IN 关键字** - 参数可冠以关键字 **IN** 作为前缀：
 - **IN**—该参数 对象为标量参数提供值， 或者为 **UDF** 的 **TABLE** 参数提供一组值。

注意： 不能将 **TABLE** 参数声明为 **INOUT** 或者 **OUT**。只能有一个表参数（其位置并不重要）。

- **OR REPLACE** - 指定 **OR REPLACE (CREATE OR REPLACE PROCEDURE)** 将创建一个新过程或替换 同名的现有过程。该子句更改 过程的定义，但保留现有权限。如果您尝试替换已在使用中的过程， 则会返回错误。
- **RESULT** - 为外部 **UDF** 声明 其结果集中的列的名称和数据类型。 各列的数据类型必须为有效的 **SQL** 数据类型（例如，结果集中的列不能为 **TABLE** 数据类型）。结果中的数据隐含着 **TABLE** 数据类型。外部 **UDF** 只能有一个 **TABLE** 类型的结果集。

注意： **TABLE** 不是输出值。表 **UDF** 的结果集中不能包含有 **LONG VARBINARY** 或 **LONG VARCHAR** 数据类型，但是参数化表函数 (**TPF**) 的结果集中可以包含有大对象 (**LOB**) 数据类型。

TPF 不能生成 **LOB** 数据，但是其结果集中的列可以为 **LOB** 数据类型。然而，在输出中得到 **LOB** 数据的唯一方法是将列从输入表传递至输出表。如示例文件 `tpf_blob.cxx` 所示，描述属性

`EXTFNAPIV4_DESCRIBE_COL_VALUES_SUBSET_OF_INPUT` 允许执行该操作。

关于过程返回的结果集的详细信息，请参见《系统管理指南》：第二卷>使用过程和批处理。

- **SQL SECURITY** – 对于是以 **INVOKER**（调用过程的用户）身份，还是以 **DEFINER**（拥有过程的用户）身份执行过程进行定义。缺省时以 **DEFINER** 身份。

如果指定了 **SQL SECURITY INVOKER**，则将会占用更多的内存，因为必须为调用过程的每名用户进行标注。此外，如果指定了 **SQL SECURITY INVOKER**，也将以调用者身份执行名称解析。因此，应注意利用所有对象的相应所有者限定这些对象的名称（表、过程等）。例如，假设 **user1** 创建以下过程：

```
CREATE PROCEDURE user1.myProcedure()
  RESULT( columnA INT )
  SQL SECURITY INVOKER
BEGIN
  SELECT columnA FROM table1;
END;
```

如果 **user2** 试图运行此过程，而表 **user2.table1** 不存在，则会产生表查寻错误。此外，如果 **user2.table1** 存在，就会使用该表，而不使用预期的 **user1.table1**。为防止出现这种情况，请在语句中限定表引用（**user1.table1**，而不是仅 **table1**）。

- **EXTERNAL NAME** – **EXTERNAL NAME** *'external-call' : external-call; [operating-systemn.]function-name@library,*

外部 UDF 必须含有 **EXTERNAL NAME** 子句，该子句使用编程语言（例如 C 语言）定义了函数接口。数据库服务器则将函数装载入其地址空间。

库名可包含文件扩展名，在 Windows 中通常为 .dll，在 UNIX 中通常为 .so。在没有扩展名的情况下，该软件附加平台特定的缺省库文件扩展名。以下是规范示例。

```
CREATE PROCEDURE mystring( IN instr CHAR(255),
  IN input_table TABLE(A INT) )
  RESULT (CHAR(255))
EXTERNAL NAME
'mystring@mylib.dll;Unix:mystring@mylib.so'
```

更简单的编写前一个 **EXTERNAL NAME** 子句的方法（使用特定于平台的缺省设置）如下：

```
CREATE PROCEDURE mystring( IN instr CHAR(255),
  IN input_table TABLE(A INT) )
  RESULT (CHAR(255))
EXTERNAL NAME 'mystring@mylib'
```

标准

- SQL — 符合 ISO/ANSI SQL 标准。
- Sybase—Transact-SQL **CREATE PROCEDURE** 语句与此不同。
- SQLJ — Java 结果集与建议采用的标准的语法扩展相同。

权限

必须具有 **RESOURCE** 权限，否则只能创建临时 过程。具有 **DBA** 权限的用户可以通过指定所有者 为其他用户创建 **UDF**。用户必须具有 **DBA** 权限，才能创建 外部 **UDF**，或者为其他用户创建外部 **UDF**。

CREATE SCHEMA 语句

为数据库用户创建模式，模式是表、视图以及权限和其关联权限的集合。

语法

```
CREATE SCHEMA AUTHORIZATION userid
... [ { create-table-statement
| create-view-statement
| grant-statement } ] ...
```

用法

userid 必须是当前连接的用户 ID。无法为其他用户创建模式。用户 ID 不区分大小写。

如果 **CREATE SCHEMA** 语句中的任何语句失败，整个 **CREATE SCHEMA** 语句将回退。

CREATE SCHEMA 语句仅是一种将各 **CREATE** 和 **GRANT** 语句收集到一个操作中的方式。数据库中没有创建任何 **SCHEMA** 数据库对象，要删除对象，必须使用各 **DROP TABLE** 或 **DROP VIEW** 语句。要撤销权限，请对每个授予的权限使用 **REVOKE** 语句。

注意： **CREATE SCHEMA** 语句在活动 **Multiplex** 上是无效的。

各 **CREATE** 或 **GRANT** 语句不用语句分隔符分隔。语句分隔符用于标记 **CREATE SCHEMA** 语句其本身的结尾。

必须排序各 **CREATE** 或 **GRANT** 语句，以便在授予对象权限之前创建它们。

不建议为用户创建多个模式，将来的版本可能也不支持这样做。

副作用

- 自动提交

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase—Sybase IQ 不支持在 **CREATE SCHEMA** 语句中使用 **REVOKE** 语句，也不允许在 Transact-SQL 批处理或过程中使用它。

权限

必须具有 **RESOURCE** 权限。

另请参见

- CREATE TABLE 语句 (第 136 页)
- CREATE VIEW 语句 (第 151 页)
- GRANT 语句 (第 202 页)

CREATE SERVER 语句

向 ISYSSERVER 表中添加服务器。

语法

```
CREATE SERVER server-name
              CLASS 'server-class'
USING 'connection-info'
[ READ ONLY ]
```

参数

- **server-class:** - { **ASAJDBC** | **ASEJDBC** | **ASAODBC** | **ASEODBC** | **DB2ODBC** | **MSSODBC** | **ORAODBC** | **ODBC** }
- **connection-info:** - { *machine-name:port-number* [*/dbname*] | *data-source-name* }

示例

- **示例 1** - 为基于 JDBC 的 Adaptive Server Enterprise 服务器创建一个名为 `ase_prod` 的远程服务器。它的计算机名称是 “banana”，端口号是 3025。

```
CREATE SERVER ase_prod
CLASS 'asejdbc'
USING 'banana:3025'
```

- **示例 2** - 创建一个名为 `testasa` 的 SQL Anywhere 远程服务器，它位于 “apple” 计算机上，并监听端口号 2638。

```
CREATE SERVER testasa
CLASS 'asajdbc'
USING 'apple:2638'
```

- **示例 3** - 为 Oracle 服务器创建一个名为 `oracle723` 的远程服务器。它的 ODBC 数据源名称是 “oracle723”。

```
CREATE SERVER oracle723
CLASS 'oraodbc'
USING 'oracle723'
```

用法

CREATE SERVER 从 Sybase IQ 目录中定义远程 服务器。

有关服务器类和如何配置服务器的详细信息，请参见《系统管理指南第二卷》>“用于远程数据访问的服务器类”。

USING 子句—如果使用基于 JDBC 的服务器类，则 **USING** 子句为 *hostname:port-number* [/dbname]，其中：

- 主机名—是运行远程服务器的计算机。
- 端口号—是远程服务器监听的 TCP/IP 端口号。Sybase IQ 和 SQL Anywhere 缺省端口号为 2638。
- dbname—对于 SQL Anywhere 远程服务器，如果不指定 *dbname*，则使用缺省数据库。对于 Adaptive Server Enterprise，缺省值为 **master** 数据库，除使用 *dbname* 之外，另一个选择是通过其它某些方法（例如，在 **FORWARD TO** 语句中），使用另一个数据库。

有关详细信息，请参见《系统管理指南第二卷》>“用于远程数据访问的服务器类”>“基于 JDBC 的服务器类”。

如果使用基于 ODBC 的服务器类，则 **USING** 子句为 *data-source-name*。data-source-name 是 ODBC 数据源名称。

READ ONLY—**READ ONLY** 子句指定远程服务器是只读数据源。Sybase IQ 拒绝任何更新请求。

副作用

- 自动提交

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Open Client/Open Server 支持。

权限

必须具有 DBA 权限才能执行此命令。

另请参见

- ALTER SERVER 语句（第 27 页）
- DROP SERVER 语句（第 177 页）

CREATE SERVICE 语句

允许数据库服务器用作 Web 服务器。

语法

```
CREATE SERVICE service-name
            TYPE service-type-string
[ attributes ] [
AS statement ]
```

参数

- **attributes:** - [**AUTHORIZATION** { **ON** | **OFF** }] [**SECURE** { **ON** | **OFF** }] [**USER** { *user-name* | **NULL** }] [**URL** [**PATH/**] { **ON** | **OFF** | **ELEMENTS** }] [**USING** { *SOAP-prefix* | **NULL** }]
- **service-type-string:** - { 'RAW' | 'HTML' | 'XML' | 'SOAP' | 'DISH' }

示例

- **示例 1** - 若要快速设置 Web 服务器，请使用 **-xs** 开关启动数据库服务器，然后执行以下语句：

```
CREATE SERVICE tables TYPE 'HTML'
AUTHORIZATION OFF      USER DBA
```

```
AS SELECT * FROM SYS.ISYSTAB
```

执行完此语句之后，使用任意 Web 浏览器打开 URL <http://localhost/tables>。

用法

CREATE SERVICE 语句将使数据库服务器充当 Web 服务器。在 **SYSWEBSERVICE** 系统表中创建一个新条目。

service-name—Web 服务名可以是字母数字字符或以下字符组成的任何序列：“/”、“-”、“_”、“.”、“!”、“~”、“*”、“,”、“(”或“)”，但第一个字符不能以斜杠 (/) 开始，且名称中不能包含两个或两个以上连续的斜杠字符。

service-type-string - 标识服务类型。该类型必须是列出的服务类型之一。无缺省值。

AUTHORIZATION 子句 - 确定当连接到该服务时用户是否必须指定用户名和口令。如果 **AUTHORIZATION** 为 **OFF**，则需要使用 **AS** 子句，并且必须用 **USER** 子句标识单个用户。使用该用户的帐户和权限执行所有请求。

如果 **AUTHORIZATION** 为 **ON**，则所有用户必须提供用户名和口令。也可通过使用 **USER** 子句提供用户名或用户组名来限定允许使用服务的用户。如果用户名为 **NULL**，则所有已知用户都可以访问该服务。

缺省值为 **ON**。**Sybase** 建议在启用了授权的情况下运行生产系统，并且通过将用户添加到组中来授予使用该服务的权限。

SECURE 子句 - 表示是否接受不安全的连接。**ON** 表示只接受 **HTTPS** 连接。在 **HTTP** 端口上接收的服务请求将自动重定向到 **HTTPS** 端口。如果设置为 **OFF**，则 **HTTP** 和 **HTTPS** 连接都接受。缺省值为 **OFF**。

USER 子句 - 如果禁用授权，则此参数成为必需的参数，并指定用于执行所有服务请求的用户 **ID**。如果启用授权（缺省值），则此可选子句标识允许访问该服务的用户或组。缺省值为 **NULL**，即向所有用户授予访问权限。

URL 子句 - 确定是否接受 **URI** 路径；如果接受，确定如何处理 **URI** 路径。**OFF** 指示 **URI** 请求中服务名后面不必有任何内容。**ON** 指示将其余的 **URI** 解释为变量 *url* 的值。

ELEMENTS 表示按照斜线字符将 URL 路径的余下部分拆分为数量最多为 10 个的一组元素。将这些值指派给名为 **url** 的变量外加介于 1 和 10 之间的数字后缀，例如，前三个变量名称分别为 **url1**、**url2** 和 **url3**。如果提供的值不到 10 个，则将其余变量设置为 **NULL**。如果服务名以字符 / 结尾，则必须将 **URL** 设置为 **OFF**。缺省值为 **OFF**。

USING 子句一仅适用于 **DISH** 服务。该参数指定名称前缀。只处理其名称以此前缀开头的 **SOAP** 服务。

语句 - 如果该语句为空值，则 **URI** 必须指定要执行的语句。否则，指定的 **SQL** 语句是唯一可通过该服务执行的语句。该语句是 **SOAP** 服务所必需的，而 **DISH** 服务将忽略该语句。缺省值为 **NULL**。

Sybase 强烈建议在生产系统中运行的所有服务都定义一个语句。只有在启用了授权的情况下，该语句才能为 **NULL**。

RAW - 将结果集发送到客户端，而不进行任何进一步的格式设置。可以通过在过程中显式生成所需的标记来产生格式化文档，如以下示例所示。

HTML - 自动将语句或过程的结果集转换为包含表的 **HTML** 文档格式。

XML - 结果集采用 **XML** 格式。如果还不是 **XML**，则自动转换为 **XML RAW** 格式。

SOAP - 请求必须为有效的简单对象访问协议 (**SOAP**) 请求。结果集将自动设置为 **SOAP** 响应的格式。有关 **SOAP** 标准的详细信息，请参见 www.w3.org/TR/SOAP。

DISH - Determine SOAP Handler (DISH) 服务充当一个或多个 **SOAP** 服务的代理。在使用时，该服务充当容纳多个 **SOAP** 服务并为这些服务提供访问权限的容器。**Web** 服务描述语言 (**WSDL**) 文件是针对每个所包含的 **SOAP** 服务自动生成的。所包含的 **SOAP** 服务由公用前缀标识，该前缀必须在 **USING** 子句中指定。

有关使用 **Web** 服务的详细信息，请参见“**SQL Anywhere Server - 编程**”>“**HTTP Web 服务**”>“使用 **SQL Anywhere** 作为 **HTTP Web 服务器**”。

注意： 此参考指向 **SQL Anywhere** 文档。

标准

- **SQL** - 符合 **ISO/ANSI SQL** 标准。
- **Sybase** - 不受 **Adaptive Server Enterprise** 支持。

权限

必须具有 **DBA** 权限。

另请参见

- **ALTER SERVICE** 语句 (第 29 页)
- **DROP SERVICE** 语句 (第 178 页)

CREATE TABLE 语句

在数据库中或远程服务器上创建一个新表。

语法

```
CREATE [ { GLOBAL | LOCAL } TEMPORARY ] TABLE
[ IF NOT EXISTS ] [ owner. ] table-name
... ( column-definition [ column-constraint ] ...
[ , column-definition [ column-constraint ] ... ]
[ , table-constraint ] ... )
|[WITH NULLS NOT DISTINCT]
...[ IN dbspace-name ]
...[ ON COMMIT { DELETE | PRESERVE } ROWS
| NOT TRANSACTIONAL ]
[ AT location-string ]
[ PARTITION BY range-partitioning-scheme ]
```

参数

- **column-definition:** - *column-namedata-type* [[NOT] NULL] [IN *dbspace-name*] [DEFAULT *default-value* | IDENTITY] [PARTITION (*partition-name* IN *dbspace-name* [, ...])]
- **default-value:** - *special-value* | *string* | *global variable* [[-] *number* | (*constant-expression*) | *built-in-function* (*constant-expression*) | AUTOINCREMENT | CURRENT DATABASE | CURRENT REMOTE USER | NULL | TIMESTAMP | LAST USER
- **special-value:** - CURRENT { DATE | TIME | TIMESTAMP | USER | PUBLISHER } | USER
- **column-constraint:** - [CONSTRAINT *constraint-name*] { { UNIQUE | PRIMARYKEY | REFERENCES *table-name* [(*column-name*)] [action] } [IN *dbspace-name*] | CHECK (*condition*) | IQ UNIQUE (*integer*) }
- **table-constraint:** - [CONSTRAINT *constraint-name*] { { UNIQUE (*column-name* [, *column-name*] ...) | PRIMARYKEY (*column-name* [, *column-name*] ...) } [IN *dbspace-name*] | *foreign-key-constraint* | CHECK (*condition*) | IQ UNIQUE (*integer*) }
- **foreign-key-constraint:** - FOREIGNKEY [*role-name*] [(*column-name* [, *column-name*] ...)] ... REFERENCES *table-name* [(*column-name* [, *column-name*] ...)] ... [action] [IN *dbspace-name*]
- **action:** - ON { UPDATE | DELETE { RESTRICT } }
- **location-string:** - { *remote-server-name*.[*db-name*].[*owner*].*object-name* | *remote-server-name*;*db-name*;*owner*;*object-name* }
- **range-partitioning-scheme:** - RANGE(*partition-key*) (*range-partition-decl* [, *range-partition-decl* ...])
- **partition-key:** - *column-name*

- **range-partition-decl:** - *partition-name* **VALUES** <= ({ *constant-expr* | **MAX** })
[**IN** *dbspace-name*]

示例

- **示例 1** - 创建一个名为 SalesOrders2 的表，其中包含 5 列。
FinancialCode、OrderDate 和 ID 列的数据页位于数据库空间 Dsp3 中。整数列 CustomerID 的数据页位于数据库空间 Dsp1 中。CLOB 列 History 的数据页位于数据库空间 Dsp2 中。主键的数据页、ID 的 HG 位于数据库空间 Dsp4 中。

```
CREATE TABLE SalesOrders2 (
  FinancialCode CHAR(2),
  CustomerID int IN Dsp1,
  History CLOB IN Dsp2,
  OrderDate TIMESTAMP,
  ID BIGINT,
  PRIMARY KEY(ID) IN Dsp4
) IN Dsp3
```

- **示例 2** - 创建一个名为 fin_code2 的表，其中包含 4 列。code、type 和 id 列的数据页位于缺省数据库空间（它由数据库选项 DEFAULT_DBSPACE 的值决定）中。CLOB 列 description 的数据页位于数据库空间 Dsp2 中。外键 fk1 中的数据页、c1 的 HG 位于数据库空间 Dsp4 中：

```
CREATE TABLE fin_code2 (
  code INT,
  type CHAR(10),
  description CLOB IN Dsp2,
  id BIGINT,
  FOREIGN KEY fk1(id) REFERENCES SalesOrders(ID) IN Dsp4
)
```

- **示例 3** - 创建表 t1，其中，分区 p1 与 p2 相邻，分区 p2 与 p3 相邻：

```
CREATE TABLE t1 (c1 INT, c1 INT)
PARTITION BY RANGE(c1),
(p1 VALUES <= (0), p2 VALUES <= (10), p3 VALUES <= (100))
```

- **示例 4** - 创建包含 6 列和 3 个分区的分区表 bar，根据日期将数据映射到分区：

```
CREATE TABLE bar (
  c1 INT IQ UNIQUE(65500),
  c2 VARCHAR(20),
  c3 CLOB PARTITION (P1 IN Dsp11, P2 IN Dsp12,
    P3 IN Dsp13),
  c4 DATE,
  c5 BIGINT,
  c6 VARCHAR(500) PARTITION (P1 IN Dsp21,
    P2 IN Dsp22),
  PRIMARY KEY (c5) IN Dsp2) IN Dsp1
PARTITION BY RANGE (c4)
(P1 VALUES <= ('2006/03/31') IN Dsp31,
 P2 VALUES <= ('2006/06/30') IN Dsp32,
 P3 VALUES <= ('2006/09/30') IN Dsp33
) ;
```

针对每个分区的数据页分配:

分区	数据库空间	列
P1	Dsp31	c1、 c2、 c4、 c5
P1	Dsp11	c3
P1	Dsp21	c6
P2	Dsp32	c1、 c2、 c4、 c5
P2	Dsp12	c3
P2	Dsp22	c6
P3	Dsp33	c1、 c2、 c4、 c5、 c6
P3	Dsp13	c3
P1、 P2、 P3	Dsp1	c1 的查找存储和其它共享数据
P1、 P2、 P3	Dsp2	主键 (c5 的 HG)

- **示例 5** – 为图书馆数据库创建用于保存图书信息的表:

```
CREATE TABLE library_books (  
  isbn CHAR(20)          PRIMARY KEY IQ UNIQUE (150000),  
  copyright_date         DATE,  
  title                  CHAR(100),  
  author                 CHAR(50)  
)
```

- **示例 6** – 为图书馆数据库创建用于保存借出图书信息的表:

```
CREATE TABLE borrowed_book (  
  date_borrowed DATE NOT NULL,  
  date_returned DATE,  
  book          CHAR(20)  
                REFERENCES library_books (isbn),  
  CHECK( date_returned >= date_borrowed )  
)
```

- **示例 7** – 在远程服务器 SERVER_A 上创建表 t1, 并创建映射到该远程表的名为 t1 的代理表:

```
CREATE TABLE t1  
( a  INT,  
  b  CHAR(10))  
AT 'SERVER_A.db1.joe.t1'
```

- **示例 8** – 创建表 tab1, 其中包含以特殊常量 LAST USER 为缺省值的列 c1:

```
CREATE TABLE tab1(c1 CHAR(20) DEFAULT LAST USER)
```

用法

通过指定所有者名称，可为其他用户创建表。如果未指定 **GLOBAL TEMPORARY** 或 **LOCAL TEMPORARY**，表将作为基表被引用。否则，表为临时表。

与基表一样，创建的全局临时表也存在于数据库中并一直保留在数据库中，直到被 **DROP TABLE** 语句显式删除。临时表中的行仅对插入行的连接是可见的。来自相同或不同应用程序的多个连接可同时使用同一临时表，而每个连接只能看到自己的行。给定的连接会继承全局临时表在该连接首次引用它时存在的模式。连接结束时会删除临时表中的行。

创建本地临时表时，请忽略所有者说明。如果在创建临时表时指定了所有者（例如，**CREATE TABLE dbo.#temp(coll int)**），就会将其错误地创建成基表。

如果该连接中存在某一本地临时表，尝试创建同名的基表或全局临时表将失败，这是因为 *owner.table* 无法唯一地标识新表。

但是，可创建同名的本地临时表，作为现有基表或全局临时表。由于将首先解析本地临时表，因此对表名的引用将访问本地临时表。

例如，请考虑以下序列：

```
CREATE TABLE t1 (c1 int);
INSERT t1 VALUES (9);

CREATE LOCAL TEMPORARY TABLE t1 (c1 int);
INSERT t1 VALUES (8);

SELECT * FROM t1;
```

返回的结果为 8。对 t1 的任何引用都将引用本地临时表 t1，直到连接已删除本地临时表为止。

在过程中，如果要创建一个在过程完成后仍然保留的表，可使用 **CREATE LOCAL TEMPORARY TABLE** 语句，而不是 **DECLARE LOCAL TEMPORARY TABLE** 语句。使用 **CREATE LOCAL TEMPORARY TABLE** 语句创建的局部临时表会一直保留到它被显式删除或连接终止。

使用 **CREATE LOCAL TEMPORARY TABLE** 在 **IF** 语句中创建的本地临时表也在 **IF** 语句完成后继续存在。

不能用临时表创建连接索引。

请不要更新已成为任何连接索引的一部分的基表。不允许执行此操作，并会返回以下错误：

“无法更新表 %2，因为它在一个或多个连接索引中进行了定义”

Sybase IQ 不支持将 **CREATE TABLE ENCRYPTED** 子句用于对 Sybase IQ 表进行表级加密。不过，Sybase IQ 数据库中的 SQL Anywhere 表支持 **CREATE TABLE ENCRYPTED** 子句。

IF NOT EXISTS — 如果指定的对象已存在，则不会进行任何更改，也不会返回错误。

只有在您将索引声明为 **UNIQUE** 时，才能指定 **WITH NULLS NOT DISTINCT** 子句，该子句允许您指定索引键中的空值不是唯一的。有关详细信息，请参见 **UNIQUE** 子句。

IN—指定要在哪个数据库文件（数据库空间）中创建表。可使用该子句指定 **SYSTEM**，将永久表或临时表放在目录存储中。**IN** 子句的其它用法皆省略。**不能**使用该子句将 **IQ** 表放置到特定的数据库空间中。缺省情况下，所有永久表放置在主 **IQ** 存储中，所有临时表放置在临时 **IQ** 存储中。全局临时表和本地临时表永远不能出现在 **IQ** 存储中。

column-definition、*column-constraint*、*table-constraint* 和 *foreign-key* 子句中的 **IN** 子句用于指定将在其中创建 相应对象的 **dbspace**。如果省略 **IN** 子句，Sybase IQ 将会在已为其分配表的 **dbspace** 中 创建相应对象。

有关数据库空间的详细信息，请参见“**CREATE DBSPACE** 语句”。

ON COMMIT—仅允许临时表。缺省情况下，临时表的行将在 **COMMIT** 时被删除。

有关 **Multiplex** 全局临时表中的子句行为，请参见《使用 Sybase IQ **Multiplex**》>“**Multiplex** 事务”>“**DDL** 命令”>“角色限制”>“保留行”。

NOT TRANSACTIONAL—仅用于临时表。使用 **NOT TRANSACTIONAL** 创建的表不受 **COMMIT** 或 **ROLLBACK** 影响。

在某些情况下，**NOT TRANSACTIONAL** 子句可以提高性能，因为对非事务性临时表执行的操作不会导致在回退日志中生成条目。例如，在反复调用使用临时表的过程而不对 **COMMIT** 或 **ROLLBACK** 语句进行干预时，**NOT TRANSACTIONAL** 可能会非常有用。

CREATE TABLE 语句后面的带括号的列表中可以包含任意顺序的以下子句：

AT—用于在 *location-string* 指定的远程位置创建表。创建的本地表是映射至远程位置的代理表。用作代理表的表名不得超过 30 个字符。**AT** 子句支持将分号 (;) 用作分隔符。如果分号出现在 *location-string* 中，则它用作字段分隔符。如果没有分号，则使用句点作为字段分隔符。这样即可在数据库和所有者字段中使用文件名和扩展名。

分号字段分隔符主要用于当前不支持的 服务器类；但是，您也可以在 句点用作字段分隔符的情况下使用它们。例如，以下语句将表 **proxy_a** 映射到 远程服务器 **myasa** 上的 **SQL Anywhere** 数据 库 **mydb**：

```
CREATE TABLE proxy_al
AT 'myasa;mydb;al'
```

远程表的外键定义将被忽略。引用远程表的本地表的外键定义也会被忽略。如果远程服务器支持主键，则主键定义发送到此服务器。

在 **Simplex** 环境中，不能在同一节点上创建引用远程表的代理表。在 **Multiplex** 环境中，不能创建引用在 **Multiplex** 中定义的远程表的代理表。

例如，在 **Simplex** 环境中，如果您尝试创建引用在同一节点上定义的基表 **Employees** 的代理表 **proxy_e**，则系统会禁止使用 **CREATE TABLE ... AT** 语句，并返回错误消息。在 **Multiplex** 环境中，如果从引用 **Multiplex** 中定义的远程表 **Employees** 的任何节点（协调节点或辅助节点）中创建代理表 **proxy_e**，则系统会拒绝 **CREATE TABLE AT** 语句。

column-definition 一定义表中的列。《参考：构件块、表和过程》> “SQL 数据类型” 中说明了允许使用的数据类型。同一个表中的列不能同名。如果指定了 **NOT NULL**，或者列具有 **UNIQUE** 或 **PRIMARY KEY** 约束，则列不能含有任何空值。您最多可以创建 45,000 列；但如果一个表中的列多于 10,000，则可能会出现性能下降问题。每个表中允许 **NULL** 的列数限制大约为 $8 * (\text{数据库页大小} - 30)$ 。

- **DEFAULT default-value** 一定义表的列时，可使用 **CREATE TABLE**（以及 **ALTER TABLE**）语句中的 **DEFAULT** 关键字为列指定一个缺省值。如果为列指定了 **DEFAULT** 值，则此 **DEFAULT** 值将用作任一未指定列值的 **INSERT**（或 **LOAD**）语句中的列值。

有关使用列 **DEFAULT** 值的详细信息，请参见《系统管理指南第一卷》> “数据完整性”> “列缺省值有助于保持数据完整性”。

- **DEFAULT AUTOINCREMENT—DEFAULT AUTOINCREMENT** 列的值唯一地标识表中的每一行。这种类型的列还称为 **IDENTITY** 列，用于与 **Adaptive Server Enterprise** 相兼容。**IDENTITY/DEFAULT AUTOINCREMENT** 列存储在插入和更新期间自动生成的序列号。使用 **IDENTITY** 或 **DEFAULT AUTOINCREMENT** 时，列的标度必须为 0，且必须是整数数据类型之一，或精确数值类型。列的值也可以是 **NULL**。必须使用所有者的名称来限定指定的表名。

ON 可向表中插入行。如果没有指定 **IDENTITY/DEFAULT AUTOINCREMENT** 列的值，则生成一个比列中的任何其它值都大的唯一值。如果 **INSERT** 指定了列值，则使用该值；如果指定的值不大于当前列中的最大值，该值将用作后续插入的起点。

删除行不会减少 **IDENTITY/AUTOINCREMENT** 计数器的值。由于删除行而产生的间隙只能由使用插入时的显式赋值填充。数据库选项 **IDENTITY_INSERT** 必须设置为表名，以向 **IDENTITY/AUTOINCREMENT** 列执行插入操作。

例如，以下示例创建了一个带有 **IDENTITY** 列的表，并向它显示添加了一些数据：

```
CREATE TABLE mytable(c1 INT IDENTITY);
SET TEMPORARY OPTION IDENTITY_INSERT = "DBA".mytable;
INSERT INTO mytable VALUES(5);
```

显式插入小于该列最大值的行号后，后面没有显式赋值的行仍自动递增为比上一个最大值大 1 的值。

通过检查 @@identity 全局变量，可以找到最近一次插入的列值。

- **IDENTITY—使用 AUTOINCREMENT 缺省值的替代方法**，并与 **Transact-SQL** 兼容。在 **Sybase IQ** 中，可以使用 **IDENTITY** 或 **DEFAULT AUTOINCREMENT** 子句创建标识列。

table-constraint—帮助确保数据库中数据的完整性。共有四种类型的完整性约束：

- **UNIQUE** 约束—标识唯一—标识表中各行的一列或多列。表中任意两行的值在所有命名列中均不能相同。表可以有多个唯一约束。
- **PRIMARY KEY** 约束—与 **UNIQUE** 约束相同，但表只能有一个主键约束。不能为同一列同时指定主键约束和唯一约束。主键通常标识行的最佳标识符。例如，客户号可能是 **customer** 表的主键。

- **FOREIGN KEY** 约束—限制一组列的值，以与主键或另一表的唯一约束中的值相匹配。例如，外键约束可用于确保 **invoice** 表中的客户号与 **customer** 表中的客户号相对应。

注意： 不能在本地临时表上创建外键约束。全局临时表必须用 **ON COMMIT PRESERVE ROWS** 创建。

- **CHECK** 约束—允许对任意条件进行检验。例如，检查约束可用于确保名为 **Gender** 的列只包含值 **male** 或 **female**。表中的任何行都不能违反约束。如果 **INSERT** 或 **UPDATE** 语句会导致行违反约束，则不允许执行相应操作并且撤消语句的作用。列检查约束中以符号 “@” 开头的列标识符是实际列名的占位符。以下格式的语句：

```
CREATE TABLE t1(c1 INTEGER CHECK (@foo < 5))
```

与下面的语句完全相同：

```
CREATE TABLE t1(c1 INTEGER CHECK (c1 < 5))
```

表检查约束中出现的以符号 “@” 开头的列标识符不是占位符。

如果一条语句会引起数据库的更改并且这个更改会使数据库违反完整性约束，那么这条语句就不会得到有效执行，并且系统会报告一条错误。（有效是指该语句在系统检测到错误之前所做的任何更改都将被撤消。）

Sybase IQ 通过为该列创建 **HG** 索引来强制执行单列 **UNIQUE** 约束。

注意： 不能将使用 **BIT** 数据类型的列定义为 **UNIQUE** 或 **PRIMARY KEY** 约束。而且，缺省情况下，**BIT** 数据类型的列不允许 **NULL** 值；您可以通过显式定义该列允许 **NULL** 值来更改这种情况。

column-constraint—限制列可以容纳的值。列约束和表约束有助于确保数据库的数据完整性。如果语句会导致违反约束，则该语句的执行不会完成。该语句在检测到错误前所做的任何更改都被撤消并报告错误。列约束是相应表约束的简化。例如，下列语句是等效的：

```
CREATE TABLE Products (
    product_num integer UNIQUE
)
CREATE TABLE Products (
    product_num integer,
    UNIQUE ( product_num )
)
```

正常情况下使用列约束，除非约束引用表中的多列。这些情况下，必须使用表约束。

IQ UNIQUE 约束 — 该约束只能指定 给列。**IQ UNIQUE** 定义列的基数，用于从内部优化索引。其缺省值为 0，没有为 **IQ** 提供优化缺省 索引的信息。如果列需要的非重复计数（唯一值的个数）小于或等于 65536，就应该应用 **IQ UNIQUE** 约束。这样会使 Sybase IQ 优化该列的数据 的存储。

当 **MINIMIZE_STORAGE** 选项为 **ON** 时（新数据库缺省为 **OFF**），则等效于为每个新创建的列指定 **IQ UNIQUE 255**，而且除了唯一值多于 65536 的列皆不需要指定 **IQ**

UNIQUE。有关相关信息，请参见《系统管理指南第一卷》>“数据库对象管理”>“表管理”>“表创建准则”>“优化存储和查询性能”。

完整性约束

UNIQUE 或 UNIQUE (column-name, ...)—表中任意两行的值在所有命名列中均不能相同。表可以有多个唯一约束。

唯一约束与唯一索引是有区别的。唯一索引的列可以为 NULL，而唯一约束的列不能为 NULL。外键可以引用主键或者具有唯一约束的列，但不能引用具有唯一索引的列，因为唯一索引可以包括多个 NULL 实例。

PRIMARY KEY 或 PRIMARY KEY (column-name, ...)—表的主键由列出的列组成，而且指定的列都不可含有任何空值。Sybase IQ 可以确保表中的每一行都有唯一的主键值。表只能有一个 PRIMARY KEY。

采用第二种格式 (PRIMARY KEY 后面跟有列的列表) 时，创建的主键所包括的列是按其定义的顺序排序的，而不是按列出的顺序排序。

将一列指定为 PRIMARY KEY、FOREIGN KEY 或 UNIQUE 时，Sybase IQ 会自动为其创建一个 High_Group 索引。对于多列主键，该索引在主键上，不在各个列上。为了获得最佳性能，应单独用 HG 或 LF 索引对各列创建索引。

REFERENCES primary-table-name [(primary-column-name)]—该子句 将列定义为主键的外键，或主表的唯一约束。通常，外键是针对主键而言的，而非针对唯一约束。如果指定了主列名，则它必须与主表中的某一列相匹配，并且该主表受仅由此列组成的唯一约束或主键约束的制约。否则外键就会引用第二个表的主键。主键与外键必须具有相同的数据类型以及精度、标度和符号。仅会为单列外键创建非唯一单列 HG 索引。对于多列外键，Sybase IQ 会创建非唯一组合 HG 索引。唯一或非唯一 HG 索引的多列组合键的最大宽度为 1KB。

临时表不能有引用基表的外键，而基表不能有引用临时表的外键。本地临时表不能有外键，也不能被其引用。

FOREIGN KEY [role-name] [(...)] REFERENCES primary-table-name [(...)]—定义引用另一表的主键或唯一约束的外键。通常，外键是针对主键而言的，而非针对唯一约束。(按照这种描述，该另一表称为主表。)

如果未指定主表列名，主表列即是指表的主键中的列。如果未指定外键列名，则外键列的列名与主表的列名相同。如果指定外键列名，则还必须指定主键列名，列名按照列表中的位置成对出现。

如果主表与外键表不一样，则可能是由于唯一约束或主键约束是在被引用的键上定义的。被引用的键和外键必须具有同样多的列数，并拥有具有同样符号、精度和标度的同一数据类型。

行的外键值必须作为主表的某一行的候选键值出现，除非外键的允许空值的列中有一列或多列包含空值。

所有非显式定义的外键列在创建时会被自动赋予与主表中的相应列相同的数据类型。这些自动创建的列不能属于外表的主键。因此，同时用于主键和外键的列必须显式创建。

role-name 是外键的名称。*role-name* 的主要作用是区分同一表的两个外键。如果不指定 *role-name*，则按如下方式分配角色名：

1. 如果没有与表名同名的 *role-name* 外键，则将表名指派为 *role-name*。
2. 如果表名已被使用，则 *role-name* 为表名加上表的唯一 3 位零填充数字。

参照完整性动作定义为维护数据库中的外键关系而采取的操作。每当更改主键值或从数据库表中删除主键值时，其它表中的相应外键值也应该以某种方式修改。您可以指定 **ON DELETE** 子句，后跟 **RESTRICT** 子句。

RESTRICT—如果在数据库的其它地方有相应的外键时试图更新或删除主键值，将发生错误。如果试图更新外键因此通过候选键创建了不匹配的新值，将发生错误。这是缺省操作，除非您指定该 **LOAD** 拒绝违反参照完整性的行（可选）。这样会在语句级强制实施参照完整性。

如果使用 **CHECK ON COMMIT** 时未指定任何操作，则 **RESTRICT** 将被认作是 **DELETE** 的一项操作。Sybase IQ 不支持 **CHECK ON COMMIT**。

全局临时表不能有引用基表的外键，而基表不能有引用全局临时表的外键。本地临时表不能有外键，也不能被其引用。

CHECK（条件）—不允许行违反条件。如果 **INSERT** 语句会导致行违反条件，则不允许该操作并且撤消语句的执行结果。

只有条件为 **FALSE** 时才会拒绝更改；特别是，如果条件为 **UNKNOWN** 也会允许更改。Sybase IQ 不会强制实行 **CHECK** 条件。有关 **TRUE**、**FALSE** 和 **UNKNOWN** 条件的详细信息，请参见《参考：构件块、表和过程》>“SQL 语言元素”>“空值”和《参考：构件块、表和过程》>“SQL 语言元素”>“搜索条件”。

注意： Sybase 建议，除非您确认没有孤立外键，否则就不要在 Sybase IQ 中定义参照完整性 外键-主键关系。

远程表

远程表的外键定义将被忽略。引用远程表的本地表的外键定义也会被忽略。如果远程服务器支持主键，则主键定义发送到此服务器。

PARTITION BY RANGE—根据分区列中指定范围内的值指定要对其进行分区的行。

partition-key clause 中的 *column-name* 用于指定分区键列。Sybase IQ 支持一个分区键列。

range-partition-decl clause 中的 *partition-name* 指定要在其上存储表行的新分区的名称。在某个表的分区集中分区名称不能雷同。**partition_name clause** 是必需项。

VALUE 子句—根据范围分区条件，为每个分区指定上限（包含上限）。用户必须为每个范围分区指定分区条件，才能保证每一行只分配给一个分区。分区列允许 **NULL**，包含作为分区键值的 **NULL** 的行属于第一个表分区。然而，界限值不能是 **NULL**。

第一个分区没有下限（最小值）。分区键第一列中 **NULL** 单元格的行将进入第一个分区。对于最后一个分区，您可以指定上限或最大值（包含该值）。如果最后一个分区上限值不是最大值，则在加载或插入其中的分区键值大于最后一个分区上限值的行时将会生成错误。

MAX—表示无限大的上限，只能为最后一个分区指定。

IN—在 *partition-decl* 中，指定分区中的行所在的数据库空间。

以下限制影响范围分区表的分区键和界限值：

- 分区界限必须是常量，而不是常量表达式。
- 分区边界必须依照分区创建顺序按升序排列。也就是说，第二个分区上限必须大于第一个分区上限，依此类推。
此外，分区界限值必须与相应的分区键列数据类型兼容。例如，**VARCHAR** 与 **CHAR** 兼容。
- 如果界限值的数据类型与其相应的分区键列不同，**Sybase IQ** 会将界限值的数据类型转换为分区键列的数据类型，但存在以下例外情况：
- 不允许进行显式转换。此示例尝试将 **INT** 显式转换为 **VARCHAR**，并产生错误：

```
CREATE TABLE Employees(emp_name VARCHAR(20)) PARTITION BY
RANGE(emp_name) (p1 VALUES <=(CAST (1 AS VARCHAR(20))), p2 VALUES
<= (CAST (10 AS VARCHAR(20))))
```
- 不允许进行导致数据丢失的隐式转换。在此示例中，分区界限与分区键类型不兼容。舍入假定可导致数据丢失，并会产生错误：

```
CREATE TABLE emp_id (id INT) PARTITION BY RANGE(id) (p1 VALUES <=
(10.5), p2 VALUES <= (100.5))
```
- 在此示例中，分区界限与分区键数据类型兼容。界限值会直接转换为浮点值。不需要舍入，因而支持转换：

```
CREATE TABLE id_emp (id FLOAT) PARTITION BY RANGE(id) (p1 VALUES
<= (10), p2 VALUES <= (100))
```
- 不允许从非二进制数据类型转换为二进制数据类型。例如，不允许进行以下转换，并会返回错误：

```
CREATE TABLE newemp (name BINARY) PARTITION BY RANGE(name) (p1
VALUES <= ("Maarten"), p2 VALUES <= ("Zymmerman"))
```
- 在范围分区表中不能用空值作为界限。
- 如果分区键第一列中的单元格值的计算结果为空值，该行会在第一个分区。**Sybase IQ** 仅支持单个列分区键，因此分区键中的任何 **NULL** 会将该行分配给第一个分区。

另请参见《系统管理指南第一卷》>“数据库对象管理”>“表管理”>“表创建准则”。

副作用

- 自动提交

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
以下是供应商扩展：
 - { **IN** | **ON** } *dbspace-name* 子句
 - **ON COMMIT** 子句
 - 一些缺省值
- Sybase - 受 Adaptive Server Enterprise 支持，有一些差异。
 - 临时表 — 通过在 **CREATE TABLE** 语句中的表名前加上 # 符号，可以创建临时表。这些临时表是 Sybase IQ 声明的临时表，只能在当前连接中才可用。有关声明的临时表的信息，请参见“**DECLARE LOCAL TEMPORARY TABLE** 语句”。
 - 物理放置 — 在 Sybase IQ 和 Adaptive Server Enterprise 中，表的物理放置方式不同。Adaptive Server Enterprise 支持 **ON segment-name** 子句，Sybase IQ 中也支持它，但 *segment-name* 引用 IQ dbspace。
 - 约束 — Sybase IQ 不支持命名的约束或命名的缺省值，但确实支持允许在数据类型定义中封装约束和缺省值定义的用户定义数据类型。它还支持在 **CREATE TABLE** 语句中使用显式缺省值和 **CHECK** 条件。
 - 空值缺省值 — 缺省情况下，Adaptive Server Enterprise 中的列缺省为非空值，而 Sybase IQ 中的缺省设置为空值，以允许空值。可使用 **ALLOW_NULLS_BY_DEFAULT** 选项控制此设置。请参见“**ALLOW_NULLS_BY_DEFAULT** 选项 [T-SQL]”。要使您的数据定义语句成为可移植的语句，应显式指定 **NULL** 或 **NOT NULL**。

权限

必须具有 **RESOURCE** 权限。若要为其它用户创建表，您必须具有 **DBA** 权限。若要在 IQ main 存储数据库空间中创建基表，必须对指定的数据库空间具有 **DBA** 权限/**RESOURCE** 权限和 **CREATE** 权限。

另请参见

- **ALLOW_NULLS_BY_DEFAULT** 选项 [TSQL]（第 333 页）
- **ALTER TABLE** 语句（第 31 页）
- **CREATE DBSPACE** 语句（第 77 页）
- **CREATE INDEX** 语句（第 97 页）
- **DECLARE LOCAL TEMPORARY TABLE** 语句（第 161 页）
- **DROP** 语句（第 170 页）
- **MINIMIZE_STORAGE** 选项（第 405 页）

CREATE TEXT CONFIGURATION 语句

创建文本配置对象。

语法

请参见下文。

用法

有关语法和完整说明，请参见《Sybase IQ 中的非结构化数据分析》。

CREATE TEXT INDEX 语句

创建 **TEXT** 索引。

语法

请参见下文。

用法

有关语法和完整说明，请参见《Sybase IQ 中的非结构化数据分析》。

CREATE USER 语句

创建用户。

语法

```
CREATE USER user-name [ IDENTIFIED BY password ]  
[ LOGIN POLICY policy-name ]  
[ FORCE PASSWORD CHANGE { ON | OFF } ]
```

示例

- **示例 1** – 创建一个名为 SQLTester 的用户，口令为 welcome。将 SQLTester 用户分配到 Test1 登录策略，而且下次登录时口令到期：

```
CREATE USER SQLTester IDENTIFIED BY welcome  
LOGIN POLICY Test1  
FORCE PASSWORD CHANGE ON;
```

- **示例 2** – 创建一个名为 MyGroup 的组：

```
CREATE USER MyGroup;  
GRANT GROUP TO MyGroup;
```

用法

user-name—用户的名称。

IDENTIFIED BY 子句—为用户提供口令。

policy-name - 要将用户分配到的登录策略的名称。如果未指定 **LOGIN POLICY**，则不会进行任何更改。

FORCE PASSWORD CHANGE 子句 - 控制用户登录时是否必须指定新口令。此设置覆盖其策略中的 **PASSWORD_EXPIRY_ON_NEXT_LOGIN** 选项设置。

您不必为用户指定口令。没有口令的用户无法连接到相应数据库。如果要创建组，但不希望任何人使用组用户 **ID** 连接到数据库，则这很有用。用户 **ID** 必须是有效的标识符。

用户 **ID** 和口令不能：

- 以空格、单引号或双引号开头
- 以空格结尾
- 包含分号

口令可以是有效的标识符，也可以是以单引号括起的字符串（最多 255 个字节）。口令区分大小写。**Sybase** 建议口令由 7 位 **ASCII** 字符组成，因为如果数据库服务器不能将其从客户端的字符集转换为 **UTF-8**，则其它字符也许无法正常工作。

VERIFY_PASSWORD_FUNCTION 选项可用于指定函数，以实现口令规则（例如，口令至少必须包含一位）。如果使用口令检验函数，则不能在 **GRANT CONNECT** 语句中指定多个用户 **ID** 和口令。有关详细信息，请参见“**VERIFY_PASSWORD_FUNCTION** 选项”和“**GRANT** 语句”。

另请参见“**SQL Anywhere Server - 数据库管理**”>“**配置数据库**”>“**管理用户 ID 和权限**”>“**管理登录策略**”。

注意： 此参考指向 **SQL Anywhere** 文档。

标准

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

权限

需要具有 **DBA** 或 **USER ADMIN** 权限。

另请参见

- **COMMENT** 语句（第 61 页）

- CREATE LOGIN POLICY 语句 (第 108 页)
- DROP LOGIN POLICY 语句 (第 175 页)
- DROP USER 语句 (第 180 页)
- GRANT 语句 (第 202 页)
- PASSWORD_EXPIRY_ON_NEXT_LOGIN 选项 (第 415 页)
- VERIFY_PASSWORD_FUNCTION 选项 (第 457 页)
- ALTER LOGIN POLICY 语句 (第 19 页)

CREATE VARIABLE 语句

创建 SQL 变量。

语法

```
CREATE
[OR REPLACE]
VARIABLE
identifier data-type
[ { = | DEFAULT }
initial-value ]
```

参数

- **initial-value:** - *special-value* | *string* | [-] *number* | (*constant-expression*) | *built-in-function* (*constant-expression*) | **NULL**
- **special-value:** - **CURRENT** { **DATABASE** | **DATE** | **PUBLISHER** | **TIME** | **TIMESTAMP** | **USER** | **UTC TIMESTAMP** } | **USER**

示例

- **示例 1** - 下面这段代码可向数据库中插入一个大文本值:

```
EXEC SQL BEGIN DECLARE SECTION;
char buffer[5000];
EXEC SQL END DECLARE SECTION;
EXEC SQL CREATE VARIABLE hold_blob VARCHAR;
EXEC SQL SET hold_blob = '';
for (;;) {
    /* read some data into buffer ... */
    size = fread( buffer, 1, 5000, fp );
    if( size <= 0 ) break;
    /* add data to blob using concatenation
    Note that concatenation works for binary
    data too! */
    EXEC SQL SET hold_blob = hold_blob || :buffer;
}
EXEC SQL INSERT INTO some_table VALUES ( 1, hold_blob );
EXEC SQL DROP VARIABLE hold_blob;
```

用法

CREATE VARIABLE 语句用于创建指定数据类型的新变量。如果指定了 *initial-value*, 则该变量会设置为该值。如果未指定 *initial-value*, 则该变量在 **SET** 语句分配另一个值之前包含的是空值。

指定 **OR REPLACE** 子句将删除给定的变量（如果该子句已存在并替换其定义）。可使用 **OR REPLACE** 子句来替代 SQL 脚本中的 **VAREXISTS** 函数。

变量可以在 SQL 表达式中允许使用列名的任何地方使用。如果列名和变量名相同, 则使用变量值。

变量可以在 SQL 表达式中允许使用列名的任何地方使用。名称解析按如下方式执行:

- 与查询的 **SELECT** 列表中指定的任何别名进行匹配。
- 与任何被引用表的列名进行匹配。
- 假定名称为变量。

变量属于当前连接, 当从数据库断开连接或使用 **DROP VARIABLE** 语句时, 它就会消失。变量对其它连接不可见。变量不受 **COMMIT** 或 **ROLLBACK** 语句的影响。

使用 **CREATE VARIABLE** 语句创建的变量会在一个连接期间内存在, 即使该语句是在 **(BEGIN...END)** 语句中发出的。您必须使用 **DECLARE** 来创建 仅在 **(BEGIN...END)** 语句中 (例如, 在存储过程中) 存在的变量。

在从嵌入式 SQL 程序为 **INSERT** 或 **UPDATE** 语句创建较大文本或二进制对象时, 变量十分有用。

过程和触发器中的局部变量在复合语句中声明。请参见《系统管理指南第二卷》>“使用过程和批处理”>“控制语句”>“使用复合语句”。

如果您指定了 *initial-value*, 则该数据类型必须与 *data-type* 定义的类型相匹配。

另请参见《参考: 构件块、表和过程》>“SQL 数据类型”。

标准

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

权限

无

另请参见

- **BEGIN...END** 语句 (第 49 页)
- **DECLARE** 语句 (第 154 页)
- **DROP VARIABLE** 语句 (第 181 页)

- SET 语句 [ESQL] (第 285 页)

CREATE VIEW 语句

在数据库上创建视图。视图用于以一种不同的角度提供数据，即使它不是以那种方式存储的。

语法

```
CREATE [ OR REPLACE ] VIEW
... [ owner. ]view-name [ ( column-name [ , ... ] ) ]
... AS select-without-order-by
... [ WITH CHECK OPTION ]
```

示例

- **示例 1** – 创建一个仅显示男性雇员的全部信息的视图。此视图和基表具有同样的列名：

```
CREATE VIEW male_employee
AS SELECT *
FROM Employees
WHERE Sex = 'M'
```

- **示例 2** – 创建显示雇员及其所属部门的视图：

```
CREATE VIEW emp_dept
AS SELECT Surname, GivenName, DepartmentName
FROM Employees JOIN Departments
ON Employees.DepartmentID = Departments.DepartmentID
```

用法

通过指定 **owner** 名称，可以为其他用户创建视图。必须具有 **DBA** 权限才能为其他用户创建视图。

在 **SELECT**、**DELETE**、**UPDATE** 和 **INSERT** 语句中，可以使用视图名取代表名。但在数据库中，视图实际上不是以表的形式存在。每次使用时它们就会派生。视图作为 **CREATE VIEW** 语句中指定的 **SELECT** 语句的结果派生。视图中使用的表名应该由表所有者的用户 ID 限定。否则，一个不同的用户 ID 可能无法找到表或者可能得到错误的表。

指定 **OR REPLACE** 子句 (**CREATE OR REPLACE VIEW**) 将创建一个新视图或替换同名的现有视图。如果使用 **OR REPLACE** 子句，将保留现有权限，但将删除视图上的 **INSTEAD OF** 触发器。

视图中的列名由列名列表指定。如果没有指定列名列表，则视图列名来自选择列表项。要使用选择列表项中的名称，项必须是简单列名或者指定了别名（请参见“**SELECT** 语句”）。不能向视图添加或删除 **IDENTITY/AUTOINCREMENT** 列。

只要定义视图的 **SELECT** 语句不包含 **GROUP BY** 子句、集合函数或不涉及 **UNION** 操作，就可以更新视图。视图的更新会导致基表更新。

view-name—一个标识符。缺省所有者是当前用户 ID。

column-name—视图中的列名由 *column-name* 列表指定。如果没有指定列名列表，则视图列名来自选择列表项。要使用选择列表项中的名称，每项必须是简单列名或者指定了别名（请参见“**SELECT** 语句”）。

AS—视图所依据的 **SELECT** 语句不得包含 **ORDER BY** 子句、**SELECT** 列表中的子查询或一个 **TOP** 或 **FIRST** 限定。它可以包含一个 **GROUP BY** 子句或者是一个 **UNION**。

WITH CHECK OPTION — 拒绝 不满足 **SELECT** 语句定义的视图条件的任何视图更新和插入。但是，Sybase IQ 当前会忽略该选项（它支持出于兼容性考虑的语法）。

副作用

- 自动提交

标准

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

权限

必须有 **RESOURCE** 权限和视图定义中的表的 **SELECT** 权限。

另请参见

- **CREATE TABLE** 语句（第 136 页）
- **DROP** 语句（第 170 页）
- **SELECT** 语句（第 277 页）

DEALLOCATE DESCRIPTOR 语句 [ESQL]

释放与 SQL 描述符区相关联的内存。

语法

```
DEALLOCATE DESCRIPTOR descriptor-name :  
string
```

示例

- 示例 1 – 请参见“**ALLOCATE DESCRIPTOR 语句 [ESQL]**”。

用法

释放与描述符区关联的所有内存，包括数据项、指示符变量和结构本身。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 受 Open Client/Open Server 支持。

权限

无

另请参见

- ALLOCATE DESCRIPTOR 语句 [ESQL]（第 5 页）
- SET DESCRIPTOR 语句 [ESQL]（第 290 页）

声明部分 [ESQL]

声明嵌入式 SQL 程序中的宿主变量。宿主变量用于同数据库交换数据。

语法

```
EXEC SQL BEGIN DECLARE SECTION;
... C declarations
EXEC SQL END DECLARE SECTION;
```

示例

- 示例 1 -

```
EXEC SQL BEGIN DECLARE SECTION;
char *emp_lname, initials[5];
int dept;
EXEC SQL END DECLARE SECTION;
```

用法

声明部分只不过是**由 BEGIN DECLARE SECTION 和 END DECLARE SECTION 语句括起来的 C 变量声明部分**。声明部分使 SQL 预处理器可以识别用作宿主变量的 C 变量。并非所有的 C 声明在声明部分内都有效。请参见“SQL Anywhere Server - 编程”>“嵌入式 SQL”>“嵌入式 SQL 编程技巧”。

注意： 此参考指向 SQL Anywhere 文档。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。

权限

无

另请参见

- BEGIN……END 语句 (第 49 页)

DECLARE 语句

声明复合语句 (**BEGIN... END**) 中的 SQL 变量。

语法

```
DECLARE  
variable_name [ , ... ]  
data-type [{  
=  
| DEFAULT}  
initial-value]
```

参数

- **initial-value:** - *special-value* | *string* | [-] *number* | (*constant-expression*) | *built-in-function* (*constant-expression*) | **NULL**
- **special-value:** - **CURRENT** { **DATABASE** | **DATE** | **PUBLISHER** | **TIME** | **TIMESTAMP** | **USER** | **UTC TIMESTAMP** } | **USER**

示例

- **示例 1** - 下面批处理语句阐释了 **DECLARE** 语句的使用并在服务器窗口上输出消息:

```
BEGIN  
    DECLARE varname CHAR(61);  
    SET varname = 'Test name';  
    MESSAGE varname;  
END
```

用法

在过程主体中使用的变量可以用 **DECLARE** 语句声明。变量在声明它的复合语句的持续时间内存在，而且必须在复合语句内是唯一的。

过程的主体是复合语句，变量必须紧跟在 **BEGIN** 后声明。在 Transact-SQL 过程或触发器中没有此限制。

如果指定了 **initial-value**，则该变量会设置为该值。如果未指定 **initial-value**，则该变量在 **SET** 语句分配另一个值之前包含的是空值。

如果您指定了 **initial-value**，则该数据类型必须与 *data-type* 定义的类型相匹配。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Adaptive Server Enterprise 支持。
 - 为与 Adaptive Server Enterprise 兼容，变量名的前面必须有一个 @。
 - 在 Adaptive Server Enterprise 中，过程或触发器中声明的变量在过程或触发器的持续时间内存在。在 Sybase IQ 中，如果在复合语句内声明某变量，该变量仅在此复合语句的持续时间内存在（不论它是在 Sybase IQ SQL 还是在 Transact-SQL 复合语句内声明的）。

权限

无

另请参见

- BEGIN……END 语句（第 49 页）

DECLARE CURSOR 语句 [ESQL] [SP]

声明游标。游标是处理查询结果的主要方法。

语法

```
DECLARE cursor-name
[ SCROLL
  | NO SCROLL
  | DYNAMIC SCROLL
]
CURSOR FOR
{ select-statement
| statement-name
| USING variable-name }
```

参数

- **cursor-name**: - 标识符
- **statement-name**: - 标识符 | 宿主变量

- **column-name-list**: - 标识符
- **variable-name**: - 标识符

示例

- **示例 1** - 在嵌入式 SQL 中声明滚动游标:

```
EXEC SQL DECLARE cur_employee SCROLL CURSOR  
FOR SELECT * FROM Employees;
```

- **示例 2** - 在嵌入式 SQL 中为预准备语句声明游标:

```
EXEC SQL PREPARE employee_statement  
FROM 'SELECT emp_lname FROM Employees';  
EXEC SQL DECLARE cur_employee CURSOR  
FOR employee_statement ;
```

- **示例 3** - 在存储过程中使用游标:

```
BEGIN  
    DECLARE cur_employee CURSOR FOR  
        SELECT emp_lname  
        FROM Employees;  
    DECLARE name CHAR(40);  
    OPEN cur_employee;  
    LOOP  
        FETCH NEXT cur_employee INTO name;  
        ...  
    END LOOP;  
    CLOSE cur_employee;  
END
```

用法

DECLARE CURSOR 语句用 **SELECT** 语句或 **CALL** 语句的指定名称声明游标。

SCROLL—声明为 **SCROLL** 的游标支持 **FETCH** 语句的 **NEXT**、**PRIOR**、**FIRST**、**LAST**、**ABSOLUTE** 和 **RELATIVE** 选项。当 **SCROLL** 游标处于打开状态时，您可以通过它提取结果集中的任意行。

NO SCROLL—声明为 **NO SCROLL** 的游标仅限于使用 **FETCH NEXT** 和 **FETCH ABSOLUTE (0)** 查找操作在结果集中前进。

DYNAMIC SCROLL—声明为 **DYNAMIC SCROLL** 的游标支持 **FETCH** 语句的 **NEXT**、**PRIOR**、**FIRST**、**LAST**、**ABSOLUTE** 和 **RELATIVE** 选项。当 **DYNAMIC SCROLL** 游标处于打开状态时，您可以通过它提取结果集中的任意行。

因为一旦游标离开行，行就不能返回，所以游标没有敏感性限制。因此，当请求 **NO SCROLL** 游标时，Sybase IQ 提供效率最高的游标，即敏感性游标。

FOR statement-name—使用 **PREPARE** 语句为各语句命名。只能为预准备 **SELECT** 或 **CALL** 声明游标。

FOR READ ONLY—声明为 **FOR READ ONLY** 的游标不能在定位的 **UPDATE** 或定位的 **DELETE** 操作中使用。**READ ONLY** 是 **FOR** 子句的缺省值。

当打开游标时，由游标声明的 **FOR READ ONLY** 查看声明该游标的表的版本，而不是在首次执行 **FETCH** 时查看表的版本。

例如，提取游标时，只能从表中提取一行：

```
CREATE TABLE t1 ( c1 INT );
INSERT t1 VALUES ( 1 );

BEGIN
DECLARE t1_cursor CURSOR FOR SELECT * FROM t1
FOR READ ONLY;
OPEN t1_cursor;
INSERT t1 VALUES ( 2 );
FETCH T1_CURSOR;
END
```

FOR UPDATE—可以更新声明为 **FOR UPDATE** 的游标的游标结果集。可更新游标仅支持敏感性未定型行为，其它敏感性都会被忽略。

游标打开后，会 from 所有打开以进行更新的表上提取排它表锁。在同一事务中，打开以进行更新的表上不允许独立的 **LOAD TABLE**、**UPDATE**、**INSERT**、**DELETE** 和 **TRUNCATE** 语句，因为 Sybase IQ 一次只允许一条语句对表进行修改。您一次只能打开特定表上的一个可更新游标。

除了不能通过 Open Client 滚动外，可更新游标允许进行滚动。

OF column-name-list—来自定义为可更新的游标结果集（用 *select-statement* 指定）的列的列表。

USING variable-name—可以在存储过程和用户定义的函数中的变量上声明游标。此变量是包含游标 **SELECT** 语句的字符串。当处理 **DECLARE** 时，此变量必须可用，因此必须是以下项之一：

- 过程的参数。例如：

```
create function get_row_count(in qry varchar) returns int
begin
    declare crsr cursor using qry;    declare rowcnt
int;    set rowcnt = 0;    open crsr;    lp: loop        fetch
crsr;        if SQLCODE <> 0 then leave lp end if;        set
rowcnt = rowcnt + 1;        end loop;    return rowcnt; end
```

- 此变量被赋值后，嵌套在另一个 **BEGIN...END** 内。例如：

```
create procedure get_table_name(    in id_value int, out tabname
char(128)) begin    declare qry varchar;    set qry = 'select
table_name from SYS.ISYSTAB ' ||        'where table_id=' ||
string(id_value);    begin        declare crsr cursor using qry;
open crsr;        fetch crsr into tabname;        close crsr;    end end
```

嵌入式 SQL

使用 **PREPARE** 语句为各语句命名。只能为预准备 **SELECT** 或 **CALL** 声明游标。

可更新游标支持

可更新游标的 Sybase IQ 支持与可更新游标的 SQL Anywhere 支持相似。有关游标类型和使用游标的完整论述，请参见“SQL Anywhere Server - 编程”>“在应用程序中使用 SQL”>“游标简介”。

注意： 此参考指向 SQL Anywhere 文档。

Sybase IQ 支持一种类型的游标敏感性，该敏感性按照对基础数据的哪些更改是可见的进行定义。所有 Sybase IQ 游标都是敏感性游标，这意味着更改可能会反映在通过游标看到的结果集的成员资格、顺序或值中，或者也可能根本没有反映。

使用敏感性游标时，受定位的 **UPDATE** 和定位的 **DELETE** 影响的语句都会在游标结果集中可见，除非客户端高速缓存阻止看到这些更改。插入的行不可见。

已进行更新以使其不再满足打开的游标的 **WHERE** 子句的行仍然可见。

当使用游标时，在效率和一致性之间总存在平衡问题。敏感性游标以一致性为代价提供高效性能。

Sybase IQ 支持单个表上的可更新游标。

LONG VARCHAR 和 LONG BINARY 数据类型在可更新游标中不受支持。有关 Sybase IQ 中的 LONG VARCHAR 和 LONG BINARY 数据类型的信息，请参见《Sybase IQ 中的非结构化数据分析》。

可更新游标中不支持标量用户定义的函数和用户定义的集合函数。

Sybase IQ 中支持的用于可更新游标的查询说明如下：

- 选择列表中的表达式，该选择列表针对功能上不取决于要更新的列
- 敏感性未定型行为进行的任意子查询，也就是说对子查询所引用的数据的更改在游标结果集中不可见
- **ORDER BY** 子句；可能会更新 **ORDER BY** 列，但结果集不会重新排序
- 符合以下要求的列：
 - 列上没有 **CAST**
 - **SELECT** 子句中基表的基列
 - **SELECT** 子句中没有该列的表达式或函数，并且它在选择列表（例如，**SELECT c1, c1**）中不重复。
 - 如果指定了 **FOR UPDATE OF column-name-list** 子句，则仅限该子句中列出的基表的基列。

Sybase IQ 不允许使用在含有任何阻止一对一地将结果集行映射至基表中的行的运算符的查询上的可更新游标；特别是：

- **SELECT DISTINCT**
- 有 **UNION** 的运算符
- 有 **GROUP BY** 的运算符
- 有 **SET** 函数的运算符
- 有 **OLAP** 函数的运算符，**RANK()** 除外

请参见“UPDATE (定位) 语句 [ESQL] [SP]”的说明，了解用于更新游标结果集中的行的 **SET** 子句所允许的列和表达式的相关信息。

Sybase IQ 仅支持可更新游标上的插入，并且这些游标上的可非空列、非标识列都已选定且可更新。

在 Sybase IQ 中，打开的可更新游标内不允许有 **COMMIT** 和 **ROLLBACK**，即使该游标是作为持有游标而打开的。Sybase IQ 不支持可更新游标内有 **ROLLBACK TO SAVEPOINT**。

在游标打开后发生的任何故障都会导致已通过此打开的游标执行的所有操作回退。

可更新游标限制

声明的游标在以下情况下是只读游标，并且不可更新：

- 数据抽取工具在 **TEMP_EXTRACT_NAME1** 选项设置为路径名的情况下启用
- 作为连接索引或在连接索引内
- **ANSI_CLOSE_CURSORS_ON_ROLLBACK** 设为 **OFF**
- **CHAINED** 设为 **OFF**
- 语句为 **INSERT SELECT** 或 **SELECT INTO**
- 包括了一个以上的表
- 不存在可更新列

如果 Sybase IQ 未能设置请求的可更新游标，请参见 **.iqmsg** 文件以了解相关信息。

对可更新游标和 ODBC 有一个限制。使用以下 ODBC 函数一次最多只能更新、删除或插入 65535 行或 65535 个记录：

- **SQLSetPosSQL_UPDATE**、**SQL_DELETE** 和 **SQL_ADD**
- **SQLBulkOperationsSQL_ADD**、**SQL_UPDATE_BY_BOOKMARK** 和 **SQL_DELETE_BY_BOOKMARK**

语句属性中的最大值有一个视具体实现而定的限制，可控制受影响的行数最大值为 **UNSIGNED SMALL INT**，即 65535。

```
SQLSetStmtAttr(HANDLE, SQL_ATTR_ROW_ARRAY_SIZE, VALUE, 0)
```

可更新游标的不同之处

Sybase IQ 可更新游标与 ANSI SQL3 标准行为的区别如下：

- 持有游标更新在提交后关闭。
- Sybase IQ 在游标打开时会锁定表。
- 游标关闭时，系统按以下顺序执行所有更新、删除和插入操作：首先执行删除操作，接着执行更新操作，然后执行插入操作。

另请参见《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqcursorinfo 过程”。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase—受 Open Client/Open Server 支持。

权限

无

另请参见

- CALL 语句 (第 55 页)
- DELETE (定位) 语句 [ESQL] [SP] (第 165 页)
- FETCH 语句 [ESQL] [SP] (第 188 页)
- OPEN 语句 [ESQL] [SP] (第 245 页)
- PREPARE 语句 [ESQL] (第 251 页)
- SELECT 语句 (第 277 页)
- UPDATE (定位) 语句 [ESQL] [SP] (第 308 页)

DECLARE CURSOR 语句 [T-SQL]

声明一个与 Adaptive Server Enterprise 兼容的游标。

语法

```
DECLARE cursor-name  
... CURSOR FOR select-statement  
...[ FOR { READ ONLY | UPDATE } ]
```

用法

Sybase IQ 支持 Adaptive Server Enterprise 中不支持的 **DECLARE CURSOR** 语法。有关 **DECLARE CURSOR** 的完整语法的信息，请参见“DECLARE CURSOR 语句 [ESQL] [SP]”。

另请参见《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqcursorinfo 过程”。

标准

- SQL—FOR UPDATE 和 FOR READ ONLY 选项是 ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase—Adaptive Server Enterprise**DECLARE CURSOR** 语句的一些特性在 Sybase IQ 中不受支持。

- 在 Sybase IQ 术语中，过程或批处理中的 **DECLARE CURSOR** 必须紧跟在 **BEGIN** 关键字后。Transact-SQL 术语 没有此限制。
- 在 Adaptive Server Enterprise 中，当在过程或批处理中 声明游标时，它在过程或批处理的持续时间内 存在。在 Sybase IQ 中，如果在复合语句内声明某游标，该游标仅在此复合语句的持续时间内存在（不论它是在 Sybase IQ SQL 还是在 Transact-SQL 复合语句内声明的）。

权限

无

另请参见

- **DECLARE CURSOR** 语句 [ESQL] [SP]（第 155 页）

DECLARE LOCAL TEMPORARY TABLE 语句

声明本地临时表。

语法

```
DECLARE LOCAL TEMPORARY TABLE table-name
... ( column-definition [ column-constraint ] ...
[ , column-definition [ column-constraint ] ... ]
[ , table-constraint ] ... )
...[ ON COMMIT { DELETE | PRESERVE } ROWS
NOT TRANSACTIONAL ]
```

示例

- **示例 1** – 在嵌入式 SQL 中声明本地临时表：

```
EXEC SQL DECLARE LOCAL TEMPORARY TABLE MyTable (
    number INT
);
```

- **示例 2** – 在存储过程中声明本地临时表：

```
BEGIN
    DECLARE LOCAL TEMPORARY TABLE TempTab (
        number INT
    );
    ...
END
```

用法

DECLARE LOCAL TEMPORARY TABLE 声明本地临时表。

本地临时表以及其中的行仅对创建该表并插入这些行的连接可见。缺省情况下，临时表的行将在 **COMMIT** 时被删除。

复合语句中声明的局部临时表存在于复合语句中。否则，声明的局部临时表在连接断开前一直存在。

有关 *column-definition*、*column-constraint* 和 *table-constraint*，以及 **NOT TRANSACTIONAL** 子句的定义，请参见“**CREATE TABLE** 语句”。有关如何将数据选入临时表中的示例，请参见“**SELECT** 语句”。

隐式或显式创建本地临时表之后，在该临时表存在期间，不能创建同一名称的另一个临时表。例如，您可以隐式创建一个本地临时表：

```
select * into #tmp from table1
```

或者，您可以通过声明以下内容显式创建一个本地临时表：

```
declare local temporary table foo
```

然后，如果您再次尝试选择 `#tmp` 或 `foo`，或者声明 `#tmp` 或 `foo`，将会收到一个错误，指示 `#tmp` 或 `foo` 已存在。

声明本地临时表时，请忽略所有者说明。如果在同一会话的一个以上的 **DECLARE LOCAL TEMPORARY TABLE** 语句中指定相同的 `owner.table`，系统会报告一个语法错误。例如，在同一会话中执行下列语句时系统会报告错误：

```
DECLARE LOCAL TEMPORARY TABLE user1.temp(coll int);
DECLARE LOCAL TEMPORARY TABLE user1.temp(coll int);
```

如果忽略所有者名，则报告错误“项 `temp` 已存在”：

```
DECLARE LOCAL TEMPORARY TABLE temp(coll int);
DECLARE LOCAL TEMPORARY TABLE temp(coll int);
```

如果该连接中存在某一本地临时表，尝试创建同名的基表或全局临时表将失败，这是因为 *owner.table* 无法唯一地标识新表。

但是，可创建同名的本地临时表，作为现有基表或全局临时表。由于将首先解析本地临时表，因此对表名的引用将访问本地临时表。

例如，请考虑以下序列：

```
CREATE TABLE t1 (c1 int);
INSERT t1 VALUES (9);

DECLARE LOCAL TEMPORARY TABLE t1 (c1 int);
INSERT t1 VALUES (8);

SELECT * FROM t1;
```

返回的结果为 8。对 `t1` 的任何引用都将引用本地临时表 `t1`，直到连接已删除本地临时表为止。

不能在本地临时表上使用 **ALTER TABLE** 和 **DROP INDEX** 语句。

不能在本地临时表上使用 `sp_iqindex`、`sp_iqtablesize` 和 `sp_iqindexsize` 存储过程。

标准

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

权限

无

另请参见

- CREATE TABLE 语句 (第 136 页)
- SELECT 语句 (第 277 页)

DELETE 语句

从数据库中删除行。

语法

```
DELETE [ FROM ] [ owner. ]table-name  
...[ FROM table-list ]  
...[ WHERE search-condition ]
```

示例

- **示例 1** - 将 ID 为 105 的雇员从数据库中删除：

```
DELETE  
FROM Employees  
WHERE EmployeeID = 105
```

- **示例 2** - 从 FinancialData 表中删除 1993 年之前的所有数据：

```
DELETE  
FROM FinancialData  
WHERE Year < 1993
```

- **示例 3** - 从 Contacts 表中删除所有名称（如果它们在 Customers 表中已存在）：

```
DELETE  
FROM Contacts  
FROM Contacts, Customers  
WHERE Contacts.Surname = Customers.Surname  
AND Contacts.GivenName = Customers.GivenName
```

用法

DELETE 删除指定表中所有满足搜索条件的行。如果没有指定 **WHERE** 子句，则删除指定表中的所有行。

如果定义视图的 **SELECT** 语句的 **FROM** 子句中只有一个表，并且该语句不包含 **GROUP BY** 子句、集合函数或不涉及 **UNION** 运算，则可以对该视图使用 **DELETE** 语句。

DELETE 语句中可选的第二个 **FROM** 子句允许基于连接删除行。如果存在第二个 **FROM** 子句，则 **WHERE** 子句限定这第二个 **FROM** 子句的行。系统将根据第一个 **FROM** 子句中给定的表名从对应表中删除行。

可通过 **SYNCHRONIZE JOIN INDEX** 命令将 **DELETE** 对一个表的影响传递到引用该表的任何连接索引。处于性能方面的考虑，应在同步连接索引前执行尽可能多的删除操作。

注意： 不能对连接虚拟表使用 **DELETE** 语句。如果尝试删除连接虚拟表中的数据，会报告出错。

相关名解析

下面的语句说明了 **DELETE** 语句（带两个使用相关名的 **FROM** 子句）中表名存在的潜在多义性：

```
DELETE
FROM table_1
FROM table_1 AS alias_1, table_2 AS alias_2
WHERE ...
```

表 `table_1` 在第一个 **FROM** 子句中未用相关名进行标识，而在第二个 **FROM** 子句中则使用了相关名进行标识。在本例中，第一个子句中的 `table_1` 在第二个子句中用 `alias_1` 标识；此语句中只有一个 **table_1** 实例。

这是个特例，一般的规则是，如果一个表在同一语句中被标识两次，一次使用了相关名，另一次未用相关名，则认为这两次标识的是该表的两个实例。

请考虑以下示例：

```
DELETE
FROM table_1
FROM table_1 AS alias_1, table_1 AS alias_2
WHERE ...
```

在本例中，第二个 **FROM** 子句中有两个 `table_1` 实例。无法确定应使用哪个实例来判断第一个 **FROM** 子句。此时遵循一般的相关名规则，即不使用第二个子句中的任何一个实例来判断第一个 **FROM** 子句中的 `table_1`：语句中有三个 `table_1` 实例。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 受 Adaptive Server Enterprise（包括供应商扩展）支持。

权限

必须有表的 **DELETE** 权限。

另请参见

- FROM 子句 (第 194 页)
- INSERT 语句 (第 211 页)
- SYNCHRONIZE JOIN INDEX 语句 (第 300 页)
- TRUNCATE TABLE 语句 (第 302 页)

DELETE (定位) 语句 [ESQL] [SP]

删除位于游标当前位置的数据。

语法

```

WHERE      DELETE [ FROM table-spec ]
           CURRENT
           OF
           cursor-name

```

参数

- **cursor-name:** - *identifier* | *hostvar*
- **table-spec:** - [*owner.*] *correlation-name*
- **owner:** - *identifier*

示例

- **示例 1** - 从数据库中删除当前行:

```
DELETE WHERE CURRENT OF cur_employee
```

用法

这种形式的 **DELETE** 语句用于删除指定游标的当前行。当前行被定义为从游标读取的最后一行。

从中删除行的表由以下确定:

- 如果不包含 **FROM** 子句, 则游标只能位于单个表中。
- 如果游标用于连接查询 (包括使用包含连接的视图), 则必须使用 **FROM** 子句。仅删除指定表中的当前行。连接中涉及的其它表不会受到影响。
- 如果包含 **FROM** 子句, 但未指定表所有者, 则 *table-spec* 为第一个与任何相关名匹配的表。
 - 如果存在相关名, 则通过相关名来标识 *table-spec*。

- 如果相关名不存在，则 *table-spec* 必须可明确标识为游标中的表名。
- 如果包含 **FROM** 子句并且指定了表所有者，则 *table-spec* 必须可明确标识为游标中的表名。

定位 **DELETE** 语句可以使用于在视图上打开的游标，只要视图是可更新的即可。

定位 **DELETE** 语句影响的更改在游标结果集中是可见的，除非客户端缓存阻止看到这些更改。

另请参见《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqcursorinfo 过程”。

标准

- SQL—如果将 ANSI_UPDATE_CONSTRAINTS 选项设置为 OFF，则可以更新的游标范围可能包含 ISO/ANSI SQL 语法的供应商扩展。
- Sybase—Open Client/Open Server 支持使用嵌入式 SQL。在 SQL Anywhere 中支持使用过程和触发器。

权限

必须对游标中使用的表具有 DELETE 权限。

另请参见

- DECLARE CURSOR 语句 [ESQL] [SP]（第 155 页）
- INSERT 语句（第 211 页）
- UPDATE 语句（第 304 页）
- UPDATE（定位）语句 [ESQL] [SP]（第 308 页）

DESCRIBE 语句 [ESQL]

获取特定宿主变量的有关信息，这些宿主变量是存储从数据库检索的数据所必需的，或用于将数据传递到数据库。

语法

```
DESCRIBE
...[ USER TYPES ]
...[ { ALL | BIND VARIABLES FOR | INPUT
| OUTPUT | SELECT LIST FOR } ]
...[ { LONG NAMES [ long-name-spec ] | WITH VARIABLE RESULT } ]
...[ FOR ] { statement-name | CURSOR cursor-name }
...INTO sqlda-name
```

参数

- **long-name-spec:** - { **OWNER.TABLE.COLUMN** | **TABLE.COLUMN** | **COLUMN** }
- **statement-name:** - 标识符 | 宿主变量
- **cursor-name:** - 声明的游标
- **sqlda-name:** - 标识符

示例

- **示例 1** - 如何使用 **DESCRIBE** 语句:

```
sqllda = alloc_sqllda( 3 );
EXEC SQL DESCRIBE OUTPUT
  FOR employee_statement
  INTO sqllda;
if( sqllda->sqlld > sqllda->sqln ) {
  actual_size = sqllda->sqlld;
  free_sqllda( sqllda );
  sqllda = alloc_sqllda( actual_size );
  EXEC SQL DESCRIBE OUTPUT
    FOR employee_statement
    INTO sqllda;
}
```

用法

DESCRIBE 设置指定的 **SQLDA** 以描述指定语句的 **OUTPUT**（等效于 **SELECT LIST**）或 **INPUT (BIND VARIABLES)**。

在使用 **INPUT** 的情况下，**DESCRIBE BIND VARIABLES** 不会在 **SQLDA** 中设置数据类型：这需要由应用程序完成。使用 **ALL** 关键字可以在一个 **SQLDA** 中描述 **INPUT** 和 **OUTPUT**。

如果指定了语句名，则以前一定已经使用 **PREPARE** 语句用同样的语句名准备了此语句，并且以前一定已经分配了 **SQLDA**（请参见“**ALLOCATE DESCRIPTOR** 语句 [ESQL]”）。

如果指定了游标名，则以前必须已经声明和打开过此游标。缺省操作是描述 **OUTPUT**。只有 **SELECT** 语句和 **CALL** 语句具有 **OUTPUT**。将 **SQLDA** 的 **sqlld** 字段设置为零后，对其它任何语句或非动态游标执行 **DESCRIBE OUTPUT** 时都不会显示输出。

USER TYPES 一帶有 **USER TYPES** 子句的 **DESCRIBE** 语句返回有关列的用户定义数据类型的信息。通常，在前一个 **DESCRIBE** 返回 **DT_HAS_USERTYPE_INFO** 指示符时会执行这样的 **DESCRIBE**。

返回的信息和不带 **USER TYPES** 关键字的 **DESCRIBE** 返回的信息是相同的，但 **sqlname** 字段保存的是用户定义数据类型的名称，而不是列名。

如果 **DESCRIBE** 使用 **LONG NAMES** 子句，则 **sqldata** 字段会保存此信息。

SELECT—DESCRIBE OUTPUT 在 **SQLDA** 中填写每个选择列表项的数据类型和长度。选择列表项的名称字段也会填入一个名称。如果为选择列表项指定了别名，则名称就

是该别名。否则，名称从选择列表项派生：如果该项是简单列名，则使用该项；否则使用表达式的子字符串。**DESCRIBE** 还会将选择列表项的数目放在 **SQLDA** 的 **sqld** 字段中。

如果所描述的语句是两个或更多 **SELECT** 语句的 **UNION**，则为 **DESCRIBE OUTPUT** 返回的列名就是为第一个 **SELECT** 语句返回的列名。

CALL—DESCRIBE OUTPUT 语句将为过程中的每个 **INOUT** 或 **OUT** 参数填充 **SQLDA** 中的数据类型、长度和名称。**DESCRIBE OUTPUT** 还将 **INOUT** 或 **OUT** 参数的数目放在 **SQLDA** 的 **sqld** 字段中。

CALL (结果集)—DESCRIBE OUTPUT 将为过程定义中的每个 **RESULT** 列填充 **SQLDA** 中的数据类型、长度和名称。**DESCRIBE OUTPUT** 还将结果列的数目放在 **SQLDA** 的 **sqld** 字段中。

INPUT—绑定变量是数据库执行语句时由应用程序提供的值。绑定变量可以看作是语句的参数。**DESCRIBE INPUT** 用绑定变量名填充 **SQLDA** 中的名称字段。**DESCRIBE INPUT** 还会将绑定变量的数目放在 **SQLDA** 的 **sqld** 字段中。

DESCRIBE 使用 **SQLDA** 中的指示符变量提供其它信息。**DT_PROCEDURE_IN** 和 **DT_PROCEDURE_OUT** 位是在描述 **CALL** 语句时在指示符变量中设置的位。**DT_PROCEDURE_IN** 指示 **IN** 或 **INOUT** 参数，而 **DT_PROCEDURE_OUT** 则指示 **INOUT** 或 **OUT** 参数。过程的 **RESULT** 列将使这两个位清零。在 **DESCRIBE OUTPUT** 之后，这些位可用于区分有结果集（需要使用 **OPEN**、**FETCH**、**RESUME**、**CLOSE**）和没有结果集（需要使用 **EXECUTE**）的语句。仅当绑定变量是 **CALL** 语句的参数时，**DESCRIBE INPUT** 才会相应地设置 **DT_PROCEDURE_IN** 和 **DT_PROCEDURE_OUT**，作为 **CALL** 语句参数的表达式中的绑定变量会设置这些位。

DESCRIBE ALL 允许您通过向数据库服务器发送一条请求来描述 **INPUT** 和 **OUTPUT**。在多用户环境中，这样做具有性能优势。首先由 **INPUT** 信息填充 **SQLDA**，然后是 **OUTPUT** 信息。**sqld** 字段包含 **INPUT** 和 **OUTPUT** 变量的总数。指示符变量中的 **DT_DESCRIBE_INPUT** 位是为 **INPUT** 变量而设，对于 **OUTPUT** 变量则清零。

- **检索长列名 - LONG NAMES** 子句用于为语句或游标 检索列名。如不使用此子句，则列名长度存在 29 个字符的限制：如果使用该子句，任意长度的名称均受支持。

如果使用了 **LONG NAMES**，长名称就会放到 **SQLDA** 的 **SQLDATA** 字段中，就像从游标中读取一样。不填写任何其它字段（**SQLLEN**、**SQLTYPE** 等）。必须像设置 **FETCH SQLDA** 一样设置 **SQLDA**：对于每列，它必须包含一个对应条目，并且该条目必须属于字符串类型。

长名称的缺省规范为 **TABLE.COLUMN**。

- **描述可变结果集 - WITH VARIABLE RESULT** 语句用于描述可以有一个以上结果集（具有不同列数或列类型）的过程。

如果使用 **WITH VARIABLE RESULT**，则数据库 服务器将 **DESCRIBE** 子句之后的 **SQLCOUNT** 值设置为下列值之一：

- 0—结果集可能更改： 应该在每个 **OPEN** 语句后重新描述过程调用。
- 1—结果集是固定的。 不需要重新 描述。

有关使用 **SQLDA** 结构的详细信息，请参见“SQL Anywhere Server – 编程” > “嵌入式 SQL” > “SQL 描述符区 (SQLDA)”。

注意： 此参考指向 SQL Anywhere 文档。

标准

- SQL—一些子句是 ISO/ANSI SQL 语法的供应商扩展。
- Sybase—一些子句受 Open Client/Open Server 支持。

权限

无

另请参见

- **ALLOCATE DESCRIPTOR** 语句 [ESQL] (第 5 页)
- **DECLARE CURSOR** 语句 [ESQL] [SP] (第 155 页)
- **OPEN** 语句 [ESQL] [SP] (第 245 页)
- **PREPARE** 语句 [ESQL] (第 251 页)

DISCONNECT 语句 [Interactive SQL]

删除与数据库的连接。

语法

```
DISCONNECT [ { connection-name | CURRENT | ALL } ]
```

参数

- **connection-name**: - 标识符、字符串或宿主变量

示例

- **示例 1** - 如何使用嵌入式 SQL 中的 **DISCONNECT**:

```
EXEC SQL DISCONNECT :conn_name
```

- **示例 2** - 如何从 **dbisql** 中使用 **DISCONNECT** 来断开所有连接:

```
DISCONNECT ALL
```

用法

DISCONNECT 语句用于删除与数据库服务器的连接并释放该连接使用的所有资源。如果要删除的连接已在 **CONNECT** 语句中命名，则可以指定该名称。指定 **ALL** 会删除应用程序与所有数据库环境的所有连接。**CURRENT** 是缺省设置，用于删除当前连接。

系统会对已删除的连接执行隐式 **ROLLBACK**。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Open Client/Open Server 支持。

权限

无

另请参见

- **CONNECT** 语句 [ESQL] [Interactive SQL]（第 65 页）
- **SET CONNECTION** 语句 [ESQL] [Interactive SQL]（第 289 页）

DROP 语句

从数据库中删除对象。

语法

```
DROP
{
  DBSPACE dbspace-name
  {
    DATATYPE [ IF EXISTS ]
    DOMAIN [ IF EXISTS ] } datatype-name
    EVENT [ IF EXISTS ] event-name
    INDEX [ IF EXISTS ] [ [ owner ].table-name ].index-name
    JOIN INDEX [ owner ].join-index-name
    MESSAGE message-number
    TABLE [ IF EXISTS ] [ owner ].table-name
    VIEW [ IF EXISTS ] [ owner ].view-name
    PROCEDURE [ IF EXISTS ] [ owner ].procedure-name
    FUNCTION [ IF EXISTS ] [ owner ].function-name }
```

示例

- 示例 1 - 从数据库中删除 Departments 表：

```
DROP TABLE Departments
```

- 示例 2 - 从数据库中删除 emp_dept 视图：

```
DROP VIEW emp_dept
```

用法

DROP 语句删除指定数据库结构的定义。如果该结构是数据库空间，则必须先删除或重新分配该数据库空间中包含任何数据的所有表，再删除该数据库空间；将自动重新分配其它结构。如果该结构是表，则在删除过程中将自动删除该表中的所有数据。另外，**DROP TABLE** 语句还会删除该表的所有索引和键。但是，如果有任何连接索引使用该表，则无法删除该表。必须先使用 **DROP JOIN INDEX** 删除这些连接索引。

如果您不希望 **DROP** 语句试图删除不存在的数据库对象时返回错误，请使用 **IF EXISTS** 子句。

DROP INDEX 可删除任何显式创建的索引。它只在没有关联的主键、唯一或外键约束时，才会删除隐式创建的索引。

如果存在关联的未受强制外键，则对非唯一 **HG** 执行的 **DROP INDEX** 操作将失败。

警告！ 请勿删除由 DBO 用户所有的视图。删除此类视图或将它们转化为表可能导致出问题。

只要语句影响了当前正由其它连接使用的表，都会禁止 **DROP TABLE**、**DROP INDEX**、**DROP JOIN INDEX** 和 **DROP DBSPACE**。

DROP TABLE 如果主表具有与之关联的外键约束（包括未受强制的外键约束），则会禁止。

如果相应表包含 **IDENTITY** 列且 **IDENTITY_INSERT** 设置为该表，也会禁止 **DROP TABLE**。要删除该表，必须清除 **IDENTITY_INSERT**（即将 **IDENTITY_INSERT** 设置为空字符串 ''），或者将其设置为其它表名。

外键可以具有非唯一的单列或多列 **HG** 索引。主键可以具有唯一的单列或多列 **HG** 索引。无法删除为现有外键、主键和唯一约束隐式创建的 **HG** 索引。如果 DBA 要删除属于其他用户的连接索引，则此连接索引的名称必须用所有者名称加以限定。

四个初始数据库空间为 **SYSTEM**、**IQ_SYSTEM_MAIN**、**IQ_SYSTEM_TEMP** 和 **IQ_SYSTEM_MSG**。您不能删除这些初始数据库空间，但只要至少一个数据库空间仍处于读写模式，您就可以从 **IQ main** 存储或目录存储（可能包含多个数据库空间）中删除数据库空间。

必须先将数据库空间中的表删除，然后才能删除数据库空间。如果该数据库空间仍包含用户数据，则会返回错误；该数据库空间删除后，将自动重新分配其它结构。只有在使数据库空间处于只读模式后，才能删除它。

注意： 数据库空间被命令使用后，在任何时刻都可能包含数据，因而会禁止对它执行 **DROP DBSPACE**。

有关修改数据库空间的详细信息，请参见《系统管理指南第一卷》>“数据库对象管理”>“数据存储”。

其它连接正在使用相应过程时，禁止执行 **DROP PROCEDURE**。

如果表中用到相应数据类型，则会禁止执行 **DROP DATATYPE**。必须更改以某种用户定义数据类型定义的所有列的数据类型，才能删除这种数据类型。建议使用 **DROP DOMAIN** 而不是 **DROP DATATYPE**，因为 **DROP DOMAIN** 是 ANSI/ISO SQL3 草案中使用的语法。

副作用

- 自动提交。清空 **dbisql** 中的数据窗口。**DROP TABLE** 和 **DROP INDEX** 会关闭当前连接的所有游标。
- 局部临时表是个特例，删除它时不执行提交。

另请参见《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqdbspace 过程”和《系统管理指南第一卷》>“数据库对象管理”。

标准

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

权限

对于 **DROP DBSPACE**，必须具有 DBA 或 SPACE ADMIN 权限，并且必须是唯一的数据库连接。

要执行其它删除语句，用户必须是相应对象的所有者且获得 DBA 权限。

除非引用全局临时表的所有用户都已经断开连接，否则不能删除该全局临时表。

要执行 **DROP INDEX**，非 DBA 用户必须提供完全限定的索引名来删除 DBA 拥有的基表上的索引。DBA 或具有适当权限的用户可以在不使用完全限定名的情况下删除非 DBA 用户拥有的表上的索引。

DROP CONNECTION 语句

删除任意用户与数据库的连接。

语法

```
DROP CONNECTION connection-id
```

示例

- 示例 1 – 删除 ID 号为 4 的连接：

```
DROP CONNECTION 4
```

用法

DROP CONNECTION 通过删除与数据库的连接，将用户同数据库断开。不能删除当前连接，必须首先创建另一个连接，然后才能删除第一个连接。

连接的 *connection-id* 是通过使用 **connection_property** 函数索取连接号获得的。下面的语句将返回当前连接的连接 ID：

```
SELECT connection_property( 'number' )
```

标准

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

权限

需要 DBA 或 OPERATOR 权限。

另请参见

- CONNECT 语句 [ESQL] [Interactive SQL]（第 65 页）

DROP DATABASE 语句

删除数据库及其关联的数据库空间段文件。

语法

```
DROP DATABASE db-filename [ KEY key-spec ]
```

参数

- **key-spec**： - 一个字符串，包含混合大小写字符、数字、字母和特殊字符。防止该键受到命令解释器的解释或更改可能非常有必要。

示例

- **示例 1** - 删除数据库 mydb：

```
DROP DATABASE 'mydb.db'
```

- **示例 2** - 删除创建的具有密钥 is!seCret 的加密数据库 marvin.db：

```
DROP DATABASE 'marvin.db' KEY 'is!seCret'
```

- **示例 3** - 以下 UNIX 示例从 /s1/temp 目录中删除数据库 temp.db：

```
DROP DATABASE '/s1/temp/temp.db'
```

用法

DROP DATABASE 先删除所有与 IQ 存储和临时存储关联的数据库段文件，再删除目录存储文件。

必须先停止数据库，然后才能删除它。如果使用连接参数 **AUTOSTOP=no**，您可能需要发出 **STOP DATABASE** 语句。

您指定的 `db-filename` 对应于您使用 **CREATE DATABASE** 为数据库定义的 数据库文件名。如果在 **CREATE DATABASE** 命令中为此值 指定了目录路径，还必须为 **DROP DATABASE** 指定目录路径。否则，Sybase IQ 会在服务器文件所在的 缺省目录中查找数据库文件。

无法通过执行 **DROP DATABASE** 语句来删除定义了 DatabaseStart 事件的 IQ 数据库。

标准

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

权限

使用数据库服务器的 **-gu** 命令行选项设置所需的权限。缺省设置为要求具有 DBA 权限。

另请参见

- CREATE DATABASE 语句（第 68 页）
- STOP DATABASE 语句 [Interactive SQL]（第 298 页）

DROP EXTERNLOGIN 语句

从 Sybase IQ 系统表中删除外部登录名。

语法

```
DROP EXTERNLOGIN login-name  
TO remote-server
```

示例

- 示例 1 -

```
DROP EXTERNLOGIN dba TO sybase1
```

用法

DROP EXTERNLOGIN 所做的更改将直到下一次与远程服务器连接时才生效。

DROP EXTERNLOGIN 从 Sybase IQ 系统表中删除外部 登录名。

login-name—指定本地用户登录名。

TO—指定远程服务器的名称。本地用户用来登录该服务器的备用登录名和口令就是所删除的外部登录凭据。

副作用

- 自动提交

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Open Client/Open Server 支持。

权限

必须具有 DBA 或 USER ADMIN 权限。

另请参见

- CREATE EXTERNLOGIN 语句 (第 89 页)

DROP LOGIN POLICY 语句

从数据库中删除登录策略。

语法

```
DROP LOGIN POLICY policy-name
```

示例

- **示例 1** - 先创建 Test11 登录策略，然后将其删除：

```
CREATE LOGIN POLICY Test11;
DROP LOGIN POLICY Test11 ;
```

用法

如果您尝试删除已分配给用户的策略，则 **DROP LOGIN POLICY** 语句会失败。您可以使用 **ALTER USER** 语句更改用户的策略分配，或使用 **DROP USER** 语句删除用户。

权限

必须具有 DBA 或 USER ADMIN 权限。

另请参见

- ALTER USER 语句 (第 39 页)
- CREATE LOGIN POLICY 语句 (第 108 页)
- DROP USER 语句 (第 180 页)
- ALTER LOGIN POLICY 语句 (第 19 页)

DROP LOGICAL SERVER 语句

删除用户定义的逻辑服务器。

语法

```
DROP LOGICAL SERVER logical-server-name
```

示例

- **示例 1** – 以下示例删除用户定义的逻辑服务器 *ls1*。

```
DROP LOGICAL SERVER ls1
```

用法

仅适用于 Multiplex。

删除逻辑服务器时，Sybase IQ 内部执行如下 Catalog 更改操作：

- 删除逻辑服务器的所有成员资格定义。
- 从每个登录策略中删除显式分配给主逻辑服务器的 逻辑服务器分配。如果该服务器是分配到登录策略的唯一逻辑服务器，Sybase IQ 将该登录策略的逻辑服务器分配设置为 NONE。
- 从 ISYSIQ.LOGICALSERVER 删除逻辑服务器条目。

权限

必须具有 DBA 或 MULTIPLEX ADMIN 权限。

DROP MULTIPLEX SERVER 语句

从 Multiplex 删除服务器。

语法

```
DROP MULTIPLEX SERVER {server-name} [drop_mpx_server_clause]
```


参数

- **drop_mpx_server_clause:** - { **WITH DROP MEMBERSHIP** | **WITH DROP LOGICAL SERVER** }

示例

- 示例 1 -

```
DROP MULTIPLEX SERVER writer1
```

用法

仅适用于 Multiplex。

Sybase 建议您先关闭每台 Multiplex 服务器，然后再删除。此语句将自动提交。

如果尚未按建议停止，删除的服务器将在执行此语句之后自动关闭。

删除最后一台辅助服务器将使 Multiplex 转回 Simplex。删除 Multiplex 内的最后一台辅助服务器后，协调器将自动关闭。如果需要，它需要重新启动。请参见第 3.3.1 节，获取 Multiplex 转换为 Simplex 时所应用的规则集详细信息。

子句 **WITH DROP MEMBERSHIP** - 正在删除的 Multiplex 服务器拥有一项或多项逻辑服务器成员资格，**DROP MULTIPLEX SERVER** 失败并显示错误。使用 **WITH DROP MEMBERSHIP** 子句，删除逻辑服务器及其所有成员资格。

子句 **WITH DROP LOGICAL SERVER** - 删除前一辅助服务器时，由于存在一处或多处用户定义的逻辑服务器，**DROP MULTIPLEX SERVER** 命令失败。使用 **WITH DROP LOGICAL SERVER** 子句，删除前一辅助服务器及其所有用户定义的逻辑服务器。

注意： **WITH DROP LOGICAL SERVER** 子句仅在删除前一辅助服务器时有效。否则，系统将报告错误。

权限

必须具有 DBA 或 MULTIPLEX ADMIN 权限。

DROP SERVER 语句

从 Sybase IQ 系统表中删除远程服务器。

语法

```
DROP SERVER server-name
```

示例

- 示例 1 -

```
DROP SERVER ase_prod
```

用法

必须先删除已经为远程服务器定义的所有代理表，**DROP SERVER** 才会成功。

副作用

- 自动提交

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Open Client/Open Server 支持。

权限

只有 DBA 帐户可以删除远程服务器。

另请参见

- CREATE SERVER 语句（第 132 页）

DROP SERVICE 语句

删除 Web 服务。

语法

```
DROP SERVICE service-name
```

示例

- 示例 1 - 删除名为 tables 的 Web 服务：

```
DROP SERVICE tables
```

用法

DROP SERVICE 删除 Web 服务。

另请参见“SQL Anywhere Server - 编程” > “HTTP Web 服务” > “使用 SQL Anywhere 作为 HTTP Web 服务器”。

注意： 此参考指向 SQL Anywhere 文档。

标准

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

权限

必须具有 DBA 权限。

另请参见

- ALTER SERVICE 语句（第 29 页）
- CREATE SERVICE 语句（第 133 页）

DROP STATEMENT 语句 [ESQL]

释放语句占用的资源。

语法

```
DROP STATEMENT [ owner. ] statement-name
```

参数

- **statement-name:** - 标识符或宿主变量

示例

- 示例 1 -

```
EXEC SQL DROP STATEMENT S1;  
EXEC SQL DROP STATEMENT :stmt;
```

用法

DROP STATEMENT 释放由指定的准备好的语句占用的资源。这些资源由成功执行的 **PREPARE** 语句分配，通常直到释放数据库连接时才释放。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase—在 Open Client/Open Server 中不受支持。

权限

必须已经准备了此语句。

SQL 语句

另请参见

- PREPARE 语句 [ESQL] (第 251 页)

DROP TEXT CONFIGURATION 语句

删除文本配置对象。

语法

请参见下文。

用法

有关语法和完整说明，请参见《Sybase IQ 中的非结构化数据分析》。

DROP TEXT INDEX 语句

从数据库中删除 **TEXT** 索引。

语法

请参见下文。

用法

有关语法和完整说明，请参见《Sybase IQ 中的非结构化数据分析》。

DROP USER 语句

删除用户。

语法

```
DROP USER user-name
```

示例

- 示例 1 - 从数据库中删除用户 SQLTester:

```
DROP USER SQLTester;
```

用法

user-name—要删除的用户的名称。

另请参见“SQL Anywhere Server – 数据库管理”>“配置数据库”>“管理用户 ID 和权限”>“管理登录策略”。

注意： 此参考指向 SQL Anywhere 文档。

标准

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

权限

必须具有 DBA 或 USER ADMIN 权限。

另请参见

- CREATE LOGIN POLICY 语句（第 108 页）
- CREATE USER 语句（第 147 页）
- DROP LOGIN POLICY 语句（第 175 页）
- GRANT 语句（第 202 页）
- ALTER LOGIN POLICY 语句（第 19 页）

DROP VARIABLE 语句

删除 SQL 变量。

语法

```
DROP VARIABLE identifier
```

用法

DROP VARIABLE 消除以前用 **CREATE VARIABLE** 语句创建的 SQL 变量。释放数据库连接时，会自动删除变量。由于变量通常用于大对象，因此在使用后删除它们或者将它们设置为 NULL，可以释放大量资源（主要是磁盘空间）。

如果您不希望在 **DROP** 语句试图删除不存在的数据库对象时返回错误，请使用 **IF EXISTS** 子句。

标准

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

权限

无

另请参见

- CREATE VARIABLE 语句 (第 149 页)
- SET 语句 [ESQL] (第 285 页)

EXECUTE 语句 [ESQL]

执行 SQL 语句。

语法

语法 1

```
EXECUTE statement-name  
... [ { USING DESCRIPTOR sqlda-name | USING host-variable-list } ]  
... [ { INTO DESCRIPTOR into-sqlda-name | INTO into-host-variable-  
list } ]  
... [ ARRAY :nnn } ]
```

语法 2

```
EXECUTE IMMEDIATE statement
```

参数

- **statement-name:** - 标识符或宿主变量
- **sqlda-name:** - 标识符
- **into-sqlda-name:** - 标识符
- **statement:** - 字符串或宿主变量

示例

- 示例 1 - 执行 DELETE:

```
EXEC SQL EXECUTE IMMEDIATE  
'DELETE FROM Employees WHERE EmployeeID = 105';
```

- 示例 2 - 执行预准备 DELETE 语句:

```
EXEC SQL PREPARE del_stmt FROM  
'DELETE FROM Employees WHERE EmployeeID = :a';  
EXEC SQL EXECUTE del_stmt USING :employee_number;
```

- 示例 3 - 执行预准备查询:

```
EXEC SQL PREPARE sell FROM  
'SELECT Surname FROM Employees WHERE EmployeeID = :a';  
EXEC SQL EXECUTE sell USING :employee_number INTO :emp_lname;
```

用法

语法 1 执行指定的动态语句，这是以前准备好的语句。如果动态语句包含为请求（绑定变量）提供信息的宿主变量占位符，则 *sqllda-name* 必须指定一个指向 **SQLDA**（它包含的描述符足够语句中出现的所有绑定变量使用）的 C 指针变量，或者必须在 *host-variable-list* 中提供绑定变量。

可选的 **ARRAY** 子句可与预准备 **INSERT** 语句一起使用，从而允许宽插入，即一次插入多行，这样可以提高性能。值 **nnn** 是要插入的行数。**SQLDA** 必须包含 **nnn** *（每行的列数）个变量。第一行放在 **SQLDA** 变量 0 和（每行的列数）-1 之间，依此类推。

SELECT 语句或 **CALL** 语句产生的 **OUTPUT** 将放入变量列表中的变量中，或放入指定 **SQLDA** 描述的 程序数据区中。**OUTPUT**（选择列表或参数）和宿主变量列表或 **SQLDA** 描述符数组之间是一一对应的对应关系。

如果 **EXECUTE** 与 **INSERT** 语句一起使用，则插入的行在第二个描述符中返回。例如，使用生成主键值的自动增量主键时，**EXECUTE** 将提供一种机制，用于立即重新读取该行和确定分配给该行的主键值。

语法 2 是一种 **PREPARE** 和 **EXECUTE** 语句的简写形式，适用于不含绑定变量或输出的语句。包含在相应字符串或宿主变量中的 **SQL** 语句将会立即得到执行，并且在完成后会被删除。

EXECUTE 可用于可以准备的任何 **SQL** 语句。游标用于返回数据库中的许多行的 **SELECT** 语句或 **CALL** 语句。

注意： 在 **EXECUTE** 语句中不能引用表 UDF。

成功执行 **INSERT**、**UPDATE** 或 **DELETE** 语句后，将用上述操作所影响的行数填写 **SQLCA** (**SQLCOUNT**) 中的 *sqlerrd[2]* 字段。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 在 Open Client/Open Server 中受支持。

权限

会检查用户对所执行的语句是否具备相应权限。

另请参见

- **DECLARE CURSOR** 语句 [ESQL] [SP]（第 155 页）
- **PREPARE** 语句 [ESQL]（第 251 页）

EXECUTE 语句 [T-SQL]

调用过程，作为 **CALL** 语句的与 Adaptive Server Enterprise 兼容的 备用方案。

语法

```
EXECUTE [ @return_status = ] [owner.]procedure_name
... { [ @parameter-name = ] expression
| [ @parameter-name = ] @variable [ output ] } ,...
```

示例

- 示例 1 - 创建过程 p1:

```
CREATE PROCEDURE p1( @var INTEGER = 54 )
AS
PRINT 'on input @var = %1! ', @var
DECLARE @intvar integer
SELECT @intvar=123
SELECT @var=@intvar
PRINT 'on exit @var = %1!', @var;
```

- 为参数提供输入值 23，执行此过程。如果是从 Open Client 应用程序连接的，则 **PRINT** 消息会显示 在客户端窗口中。如果是从 ODBC 或嵌入式 SQL 应用程序连接的，则消息会显示在数据库 服务器窗口中。

```
EXECUTE p1 23
```

- 下面是另一种执行该过程的方法，适用于存在几个参数时的情况：

```
EXECUTE p1 @var = 23
```

- 使用参数的缺省值 执行该过程：

```
EXECUTE p1
```

- 执行该过程，并将返回值存储在变量中，以便检查返回状态：

```
EXECUTE @status = p1 23
```

用法

EXECUTE 用于执行存储过程，可选择性地提供过程参数以及检索输出值 和返回状态信息。

实现 **EXECUTE** 是为了与 Transact-SQL 兼容，但此语句可以在 Transact-SQL 或 Sybase IQ 批处理 和过程中使用。

注意： 在 **EXECUTE** 语句中不能引用表 UDF。

权限

用户必须是相应过程的所有者，对该过程有 EXECUTE 权限 或获得 DBA 权限。

另请参见

- CALL 语句（第 55 页）

EXECUTE IMMEDIATE 语句 [ESQL] [SP]

实现从过程中执行动态构造的语句。

语法

语法 1

```
EXECUTE IMMEDIATE [ execute-option ] string-expression
```

```

                                execute-option:
WITH
                                QUOTES [ ON | OFF ]
| WITH
                                ESCAPES { ON | OFF }
| WITH RESULT SET { ON | OFF }
```

语法 2

```
EXECUTE ( string-expression )
```

示例

- **示例 1** - 此过程创建一个表，其中表名作为过程的参数提供。整个 **EXECUTE IMMEDIATE** 语句必须在一行上。

```

CREATE PROCEDURE CreateTableProc(
    IN tablename char(30)
)
BEGIN
    EXECUTE IMMEDIATE 'CREATE TABLE ' || tablename ||
    ' ( column1 INT PRIMARY KEY) '
END;
```

调用过程并创建表 mytable:

```
CALL CreateTableProc( 'mytable' )
```

用法

EXECUTE IMMEDIATE 扩展了可以从过程内执行的语句的范围。它使您可以执行动态准备的语句，比如用传递到过程中的参数构造的语句。

该语句中的文字字符串必须用单引号引起来，且必须不同于 **PREPARE** 或 **EXECUTE IMMEDIATE** 语句中的任何现有语句名称。整个语句必须在一行上。

EXECUTE IMMEDIATE 执行的语句只能引用全局变量。

在 Transact-SQL 存储过程内部只能使用语法 2。

WITH QUOTES—当指定 **WITH QUOTES** 或 **WITH QUOTES ON** 时，会认为字符串表达式中的所有双引号都用于界定标识符。当未指定 **WITH QUOTES** 或指定了 **WITH QUOTES OFF** 时，对字符串表达式中的双引号的处理方式取决于 **QUOTED_IDENTIFIER** 选项的当前设置。

如果使用传递到存储过程中的对象名构造要执行的语句，但该名称可能需要双引号并且当 **QUOTED_IDENTIFIER** 设置为 **OFF** 时可能调用该过程，这时 **WITH QUOTES** 很有用。

请参见“**QUOTED_IDENTIFIER** 选项 [TSQL]”。

WITH ESCAPES—**WITH ESCAPES OFF** 使得字符串表达式中的任何转义序列（如 **\n**、**\x** 或 **\w**）被忽略。例如，两个连续的反斜杠仍保留为两个反斜杠，而不会转换为一个反斜杠。缺省设置等同于 **WITH ESCAPES ON**。

可以使用 **WITH ESCAPES OFF** 来更简便地执行引用包含反斜杠的文件名的动态构造语句。

在某些上下文中，执行 **EXECUTE IMMEDIATE** 之前会转换 *string-expression* 中的转义序列。例如，会先分析复合语句，再执行它，且不论 **WITH ESCAPES** 如何设置，在此分析过程中都会转换转义序列。在此类环境中，使用 **WITH ESCAPES OFF** 可防止发生进一步的转换。例如：

```
BEGIN
DECLARE String1 LONG VARCHAR;
DECLARE String2 LONG VARCHAR;
EXECUTE IMMEDIATE
  'SET String1 = 'One backslash: \\ \ ' ';
EXECUTE IMMEDIATE WITH ESCAPES OFF
  'SET String2 = 'Two backslashes: \\ \ ' ';
SELECT String1, String2
END
```

WITH RESULT SET—可以通过指定 **WITH RESULT SET ON** 让 **EXECUTE IMMEDIATE** 语句返回结果集。使用该子句，包含它的过程被标记为返回结果集。如果不包含此子句，则如果在调用该过程时该语句不产生结果集，将报告一个错误。

注意： 缺省选项为 **WITH RESULT SET OFF**，表示语句执行时不产生结果集。

副作用

无。但是，如果此语句是存在自动提交副作用的数据定义语句，则确实会发生这种提交。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase—在 Open Client/Open Server 中受支持。

权限

无。此语句用过程所有者的权限执行，而不是用调用过程的用户的权限执行。

另请参见

- BEGIN……END 语句（第 49 页）
- CREATE PROCEDURE 语句（第 112 页）
- QUOTED_IDENTIFIER 选项 [TSQL]（第 427 页）

EXIT 语句 [Interactive SQL]

退出 Interactive SQL。

语法

```
{ EXIT | QUIT | BYE } [ return-code ]
```

return-code: number | connection-variable

示例

- - 如果表 T 有行，将把 Interactive SQL 返回值设置为 1，如果 T 中没有行，则设置为 0。

```
CREATE VARIABLE rowCount INT;
CREATE VARIABLE retcode INT;
SELECT COUNT(*) INTO rowCount FROM T;
IF( rowCount > 0 ) THEN
    SET retcode = 1;
ELSE
    SET retcode = 0;
END IF;
EXIT retcode;
```

注意：不能编写以下语句，因为 **EXIT** 是 Interactive SQL 语句（而不是 SQL 语句），而且不能在其它 SQL 块语句中包含 Interactive SQL 语句：

```
CREATE VARIABLE rowCount INT;
SELECT COUNT(*) INTO rowCount FROM T;
IF( rowCount > 0 ) THEN
    EXIT 1;    // <-- not allowed
ELSE
```

```
EXIT 0;      //  <-- not allowed
END IF;
```

用法

将 **Interactive SQL** 作为窗口式程序运行时，会关闭 **Interactive SQL** 窗口，而以命令提示符（批处理）模式运行时，会完全终止 **Interactive SQL**。在这两种情况下，数据库连接也被关闭。关闭数据库连接前，如果 **COMMIT_ON_EXIT** 选项设置为 **ON**，则 **Interactive SQL** 自动执行 **COMMIT** 语句。如果此选项设置为 **OFF**，则 **Interactive SQL** 执行隐式 **ROLLBACK**。缺省情况下，**COMMIT_ON_EXIT** 选项设置为 **ON**。

在批处理文件中可以使用可选的返回代码来指示 **Interactive SQL** 命令文件中的命令是成功还是失败。缺省返回码为 0。

副作用

- 如果选项 **COMMIT_ON_EXIT** 设置为 **ON**（缺省），则自动执行提交；否则此语句执行隐式回退。
- 在 **Windows** 操作系统中，提供的可选返回值是 **ERRORLEVEL**。

标准

- **SQL - ISO/ANSI SQL** 语法的供应商扩展。
- **Sybase**—不适用于 **Adaptive Server Enterprise**。

权限

无

另请参见

- **SET OPTION** 语句（第 291 页）

FETCH 语句 [ESQL] [SP]

重新定位游标并从中获取数据。

语法

```
FETCH
{ NEXT | PRIOR | FIRST | LAST
| ABSOLUTE row-count | RELATIVE row-count }
... cursor-name
... { [ INTO host-variable-list ]
| USING DESCRIPTOR sql-da-name
| INTO variable-list }
... [ PURGE ] [ BLOCK n ] [ ARRAY fetch-count ]
... INTO variable-list
```

```
... IQ CACHE row-count
```

参数

- **cursor-name:** - 标识符或宿主变量
- **sqlda-name:** - 标识符
- **host-variable-list:** - 可以包含指示符变量
- **row-count:** - 数字或宿主变量
- **fetch-count:** - 整数或宿主变量

示例

- **示例 1 - 嵌入式 SQL 示例:**

```
EXEC SQL DECLARE cur_employee CURSOR FOR
SELECT EmployeeID, Surname FROM Employees;
EXEC SQL OPEN cur_employee;
EXEC SQL FETCH cur_employee
INTO :emp_number, :emp_name:indicator;
```

- **示例 2 - 过程示例:**

```
BEGIN
    DECLARE cur_employee CURSOR FOR
        SELECT Surname
        FROM Employees;
    DECLARE name CHAR(40) ;
    OPEN cur_employee;
    LOOP
        FETCH NEXT cur_employee into name ;
        .
        .
        .
    END LOOP
    CLOSE cur_employee;
END
```

用法

FETCH 可从指定的游标中检索一行。

ARRAY 子句允许宽提取，即同时检索多行，这样可能会提高性能。

游标在此前必须已打开。

SELECT 语句结果中的一行被放入变量列表的变量中。选择列表和宿主变量列表之间是一对一的对应关系。

SELECT 语句返回的结果中的一行或多行放在变量列表的变量中或指定 **SQLDA** 描述的程序数据区中。在以上两种情况下，选择列表与宿主变量列表或 **SQLDA** 描述符数组之间都是一对一的对应关系。

INTO 子句是可选的。如果未指定该子句，则 **FETCH** 仅定位游标（参见以下段落）。

可指定一个可选定位参数用来允许在提取行之前移动游标。缺省值为 **NEXT**，它使游标在提取行之前向前移动一行。**PRIOR** 使游标在提取前向后移动一行。

RELATIVE 定位用于在读取前将游标沿任一方向移动指定的行数。正数表示向前移动，负数表示向后移动。因此，**NEXT** 等同于 **RELATIVE 1**，**PRIOR** 等同于 **RELATIVE -1**。**RELATIVE 0** 检索上一提取语句在此游标上所检索的行。

ABSOLUTE 定位参数用于转到具体行。0 表示第一行前面的位置。请参见《系统管理指南第二卷》>“使用过程和批处理”。

1 表示第一行，依此类推。负数用于指定从游标末尾开始计数的绝对位置。**-1** 表示游标的最后一行。**FIRST** 是 **ABSOLUTE 1** 的简写形式。**LAST** 是 **ABSOLUTE -1** 的简写形式。

注意： Sybase IQ 处理 **FIRST**、**LAST**、**ABSOLUTE** 和为负的 **RELATIVE** 选项时的效率低于处理某些其它 DBMS 产品时的效率，因此在使用它们时会影响性能。

OPEN 最初将游标定位在第一行之前。

当打开游标时，由游标声明的 **FOR READ ONLY** 查看声明该游标的表的版本，而不是在首次执行 **FETCH** 时查看表的版本。

如果提取操作中包括定位参数，而位置在允许的游标位置外，则会发出 **SQL_E_NOTFOUND** 警告。

IQ CACHE 子句指定 FIFO 队列中可以缓存的最大行数。如果不为 **IQ CACHE** 子句指定值，则会使用 **CURSOR_WINDOW_ROWS** 数据库选项的值。**CURSOR_WINDOW_ROWS** 的缺省设置为 200 行。

在嵌入式 SQL 中使用 **FETCH** 和 **OPEN** 语句

以下子句仅供嵌入式 SQL 使用：

- **USING DESCRIPTOR***sqllda-name*
- **INTO***host-variable-list*
- **PURGE**
- **BLOCK***n*
- **ARRAY***fetch-count*
- 在 *cursor-name* 和 *row-count* 中使用 *host-variable*

DECLARE CURSOR 在 C 源代码中必须出现在 **FETCH** 的前面，而 **OPEN** 语句必须在 **FETCH** 之前执行。如果宿主变量将用于游标名，则实际上是 **DECLARE** 语句生成代码，因此该语句必须在 **FETCH** 之前执行。

在多用户环境下，客户端一次可提取多行。这类提取称为块提取或多行提取。第一个提取导致从服务器发送回若干行。客户端将这些行放入缓冲区，后面的提取从这些缓冲区中检索，而不对服务器发出新的请求。

BLOCK 子句就应用程序可提取多少行向客户端和服务器给出提示。特殊值 0 表示请求将发送到服务器，并且只返回一行（没有行块）。

PURGE 子句使客户端刷新其所有行的缓冲区，然后向服务器发送提取请求。此提取请求可能返回行块。

如果在提取时返回 **SQLSTATE_NOTFOUND** 警告，则 **SQLCA (SQLCOUNT)** 的 *sqlerrd[2]* 字段中包含尝试的提取超出允许游标位置的行数。（游标可在某行上、第一行之前或最后一行之后。）如果未找到行但位置有效，则值为 0，例如，当定位到游标的最后一行上时执行 **FETCH RELATIVE 1**。如果所尝试的提取超出了游标的末尾，则为正值；如果所尝试的提取位于游标开头的前面，则为负值。

成功执行了 **FETCH** 语句后，**SQLCA (SQLIOCOUNT)** 的 *sqlerrd[1]* 字段递增，增量为执行提取所需的输入/输出操作数。该字段实际上在每个数据库语句执行时都递增。

要在嵌入式 SQL 中使用宽提取，请将 **FETCH** 语句包括在代码中：

```
EXEC SQL FETCH . . . ARRAY nnn
```

其中 *ARRAY nnn* 是 **FETCH** 语句的最后一项。提取计数 *nnn* 可以是一个宿主变量。**SQLDA** 必须包含 *nnn* * (每行的列数) 个变量。第一行放在 **SQLDA** 变量 0 和 (每行的列数) -1 之间，依此类推。

服务器在 **SQLCOUNT** 中返回提取的记录数，并始终返回大于零的 **SQLCOUNT**（除非出现错误）。旧版本的服务器仅返回一行，而 **SQLCOUNT** 设置为零。因此如果在 **SQLCOUNT** 为零时没有出现错误情况，则表示提取了一个有效行。

标准

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

权限

游标必须是打开的，并且用户必须对游标声明中所引用的表具有 **SELECT** 权限。

另请参见

- **DECLARE CURSOR** 语句 [ESQL] [SP]（第 155 页）
- **OPEN** 语句 [ESQL] [SP]（第 245 页）
- **PREPARE** 语句 [ESQL]（第 251 页）
- **CURSOR_WINDOW_ROWS** 选项（第 354 页）

FOR 语句

为游标中的每行重复执行一次语句列表。

语法

```
[ statement-label: ]  
FOR for-loop-name AS cursor-name [ cursor-type ] CURSOR
```

```

{ FOR statement
... [ { FOR { UPDATE cursor-concurrency | FOR READ ONLY } ]
    | USING variable-name }
DO statement-list
    END FOR [ statement-label ]

```

参数

- **cursor-type**: - NO SCROLL | DYNAMIC SCROLL | SCROLL | INSENSITIVE
| SENSITIVE
- **cursor-concurrency**: - BY { VALUES | TIMESTAMP | LOCK }
- **variable-name**: - *identifier*

示例

- **示例 1** - 下面的代码段阐释了 **FOR** 循环的用法:

```

FOR names AS curs CURSOR FOR
SELECT Surname
FROM Employees
DO
    CALL search_for_name( Surname );
END FOR;

```

用法

FOR 是控制语句，它允许对游标中的每一行执行一次 SQL 语句列表。

FOR 语句等效于这样一个复合语句：游标有一个 **DECLARE**，游标结果集中的每一列有一个变量 **DECLARE**，后面跟一个循环，该循环将游标中的一行读入局部变量，并对游标中的每一行执行一次 *statement-list*。

有关 *cursor-type* 参数的说明 以及更多示例，请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “FOR 语句”。

注意： 此参考指向 SQL Anywhere 文档。

所声明的局部变量的名称和数据类型都是从游标中使用的 *statement* 派生的。在 **SELECT** 语句中，数据类型是选择列表中表达式的数据类型。名称是选择列表项的别名（如果存在这样的别名），否则名称是列的名称。任何不是简单列引用的选择列表项都必须有别名。在 **CALL** 语句中，名称和数据类型取自过程定义中的 **RESULT** 子句。

LEAVE 语句可在 **END FOR** 语句后的第一个语句处用于重新开始执行。如果指定结尾 *statement-label*，它必须与开头 *statement-label* 匹配。

标准

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

权限

无

另请参见

- DECLARE CURSOR 语句 [ESQL] [SP] (第 155 页)
- FETCH 语句 [ESQL] [SP] (第 188 页)
- LEAVE 语句 (第 221 页)
- LOOP 语句 (第 241 页)

FORWARD TO 语句

向远程服务器发送本地语法。

语法

语法 1

```
FORWARD TO server-name { sql-statement }
```

语法 2

```
FORWARD TO [ server-name ]
```

示例

- **示例 1** - 与远程服务器 `ase_prod` 的直通会话:

```
FORWARD TO aseprod
SELECT * from titles
SELECT * from authors
FORWARD TO
```

用法

FORWARD TO 使用户能够 指定需要直通连接的服务器。此语句的用法如下:

- 将语句发送到远程服务器 (语法 1)
- 将 Sybase IQ 置于直通模式, 以便向远程服务器发送一系列的语句 (语法 2)

代表用户与 *server-name* 建立连接时, 服务器使用:

- 用 **CREATE EXTERNLOGIN** 设置的远程登录别名
- 如果未设置远程登录别名, 则使用用于与 Sybase IQ 通信的名称和口令

如果无法与指定的服务器建立连接, 在返回给用户的消息中会指出原因。

在将语句传递给请求的服务器后, 所有结果都会转换成客户端程序可以识别的形式。

server-name 是远程服务器的名称。

sql-statement 是远程服务器的本地语法中的命令。该命令或命令组使用大括号 ({}) 或单引号括起来。

如果指定了 *server_name*，但未指定 **FORWARD TO** 查询中的语句，则会话将进入直通模式，且所有后续查询都直接传递到远程服务器。若要关闭直通模式，请发出不指定 *server_name* 的 **FORWARD TO**。

注意： **FORWARD TO** 语句是服务器指令，不能用在存储过程、触发器、事件或批处理中。

副作用

- 在 **FORWARD TO** 会话的持续时间内，远程连接设置为 AUTOCOMMIT（非链接）模式。在 **FORWARD TO** 语句之前未完成的任何工作都自动提交。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Open Client/Open Server 支持。

权限

无

另请参见

- CREATE EXTERNLOGIN 语句（第 89 页）
- CREATE SERVER 语句（第 132 页）

FROM 子句

指定 **SELECT** 语句中涉及的数据库表或视图。

语法

```
... FROM table-expression [, ...]
```

```
table-expression :  
    table-name  
    | view-name  
    | procedure-name  
    | common-table-expression  
    | (subquery) [[ AS ] derived-table-name [column_name, ...]]  
    | derived-table  
    | join-expression  
    | ( table-expression, ... )  
    | openstring-expression  
    | apply-expression  
    | contains-expression  
    | dml-derived-table
```

```
table-name :
    [ userid.]table-name ]
    [ [ AS ] correlation-name ]
    [ FORCE INDEX ( index-name ) ]
```

```
view-name :
    [ userid.]view-name [ [ AS ] correlation-name ]
```

```
procedure-name :
    [ owner, ] procedure-name ([ parameter, ...])
    [ WITH( column-name datatype,)]
    [ [ AS] correlation-name ]
```

```
parameter :
    scalar-expression | table-parameter
```

```
table-parameter :
TABLE(select-statement) [ OVER (table-parameter-over)]
```

```
table-parameter-over :
[ PARTITION BY {ANY| NONE| table-expression } ]
[ ORDER BY { expression | integer } [ ASC | DESC ] [, ...] ]
```

```
derived-table :
    ( select-statement )
    [ AS ] correlation-name [ ( column-name, ... ) ]
```

```
join-expression :
    table-expression join-operator table-expression
    [ ON join-condition ]
```

```
join-operator :
    [ KEY | NATURAL ] [ join-type ] JOIN
    | CROSS JOIN
```

```
join-type :
    INNER
    | LEFT [ OUTER ]
    | RIGHT [ OUTER ]
    | FULL [ OUTER ]
```

```
openstring-expression :
    OPENSTRING ( { FILE | VALUE } string-expression )
    WITH ( rowset-schema )
    [ OPTION ( scan-option ... ) ]
    [ AS ] correlation-name
```

```
apply-expression :
    table-expression { CROSS | OUTER } APPLY table-expression
```

```
contains-expression :
    { table-name | view-name } CONTAINS ( column-name [...], contains-
    query ) [ [ AS ] score-correlation-name ]
```

```
rowset-schema :
    column-schema-list
    | TABLE [owner.]table-name [ ( column-list ) ]
```

SQL 语句

```
column-schema-list :  
    { column-name user-or-base-type | filler( ) } [ , ... ]
```

```
column-list :  
    { column-name | filler( ) } [ , ... ]
```

```
scan-option :  
    BYTE ORDER MARK { ON | OFF }  
    | COMMENTS INTRODUCED BY comment-prefix  
    | DELIMITED BY string  
    | ENCODING encoding  
    | ESCAPE CHARACTER character  
    | ESCAPES { ON | OFF }  
    | FORMAT { TEXT | BCP }  
    | HEXADECIMAL { ON | OFF }  
    | QUOTE string  
    | QUOTES { ON | OFF }  
    | ROW DELIMITED BY string  
    | SKIP integer  
    | STRIP { ON | OFF | LTRIM | RTRIM | BOTH }
```

```
contains-query :  
    string
```

```
dml-derived-table :  
    ( dml-statement ) REFERENCING ( [ table-version-names | NONE ] )
```

```
dml-statement :  
    insert-statement  
    delete-statement  
    update-statement  
    merge-statement
```

```
table-version-names :  
    OLD [ AS ] correlation-name [ FINAL [ AS ] correlation-name ]  
    | FINAL [ AS ] correlation-name
```

示例

- **示例 1** - 以下是有效的 **FROM** 子句:

```
...  
FROM Employees  
...  
...  
FROM Employees NATURAL JOIN Departments  
...  
...  
FROM Customers  
KEY JOIN SalesOrders  
KEY JOIN SalesOrderItems  
KEY JOIN Products  
...
```

- **示例 2** - 下面的查询阐释如何在查询中使用 派生表:

```
SELECT Surname, GivenName, number_of_orders  
FROM Customers JOIN
```

```
( SELECT CustomerID, count(*)
  FROM SalesOrders
  GROUP BY CustomerID )
AS sales_order_counts ( CustomerID,
                        number_of_orders )
ON ( Customers.ID = sales_order_counts.cust_id )
WHERE number_of_orders > 3
```

用法

SELECT 语句需要用一个表列表来指定 该语句要使用的表。

注意： 虽然这里的说明针对的是表， 但除非另有说明， 同样适用于视图。

FROM 表列表创建由所有指定表中的所有列 组成的结果集。 构成的表中行的 所有组合最初都在结果集中， 但使用连接条件和/或 **WHERE** 条件 通常会减少组合数。

- **SCALAR** - *scalar-parameter* 是有效的 SQL 数据类型的任何对象。
- **TABLE** - 可以使用 表、视图或常见的 *table-expression* 名称 指定 **TABLE** 参数，如果在 **TABLE** 参数外也使用该对象， 则将它们视为该对象的新实例。

此查询阐释了有效的 **FROM** 子句， 其中对同一表 T 的两个引用将视为 同一表 T 的两个不同的实例。

```
SELECT * FROM T, my_proc(TABLE(SELECT T.Z, T.X FROM T)
OVER(PARTITION BY T.Z));
```

表参数化函数 (TPF) 示例 - 此查询阐释有效的 **FROM** 子句。

```
SELECT * FROM R, SELECT * FROM my_udf(1);
SELECT * FROM my_tpf(1, TABLE(SELECT c1, c2 FROM t))
(my_proc(R.X, TABLE T OVER PARTITION BY T.X)) AS XX;
```

如果使用子查询来定义 **TABLE** 参数， 则必须具有以下限制：

- **TABLE** 参数 必须是 **IN** 类型。
- **PARTITION BY** 或 **ORDER BY** 子句必须引用派生表 和外部引用的列。 *expression-list* 中的表达式可以是 整数 *K*，它引用 **TABLE** 输入参数的第 *K* 列。

注意： 表 UDF 只能在 SQL 语句的 **FROM** 子句 中引用。

- **PARTITION BY** - **PARTITION BY** 子句从逻辑上指定执行引擎 如何执行函数调用。执行引擎 必须为每个分区调用函数， 并且该函数 必须在每次调用时对整个分区进行处理。

PARTITION BY 还指定必须如何对输入 数据进行分区， 以便每次调用函数时 只处理一个分区的数据。 函数调用 的次数必须等于分区数。 对于 TPF， 通过运行时 服务器和 UDF 之间的 动态协商来建立并行特性。 如果可针对 *N* 输入分区并行执行 TPF， 则可以使用 $M \leq N$ 对函数进行 *M* 次实例化。 每次函数 实例化可以多次调用， 每次调用只占用 一个分区。

注意： 执行引擎可以按任意分区顺序调用 函数， 并且无论分区顺序如何， 函数将假定返回 相同的结果集。 分区 不能拆分成两次函数调用。

只能为 **PARTITION BY** *expression-list* 或 **PARTITION BY ANY** 子句指定一个 **TABLE** 输入参数。对于必须指定的所有其它 **TABLE** 输入参数，请显式或隐式指定 **PARTITION BY NONE** 子句。

- **ORDER BY** – **ORDER BY** 子句指定执行引擎 将通过 *expression-list* 对每个分区中的输入数据进行排序。UDF 需要每个分区都具有此 物理属性。如果只有一个分区，则所有 输入数据将按 **ORDER BY** 指定进行排序。可以使用 **PARTITION BY NONE** 子句（或不使用 **PARTITION BY**）子句为 **ORDER BY** 子句指定任何 **TABLE** 输入 参数。

有关 **FROM** 子句中用于全文本 搜索的 *contains-expression* 的信息，请参见《Sybase IQ 中的非结构化数据分析》。

- **连接** – *join-type* 关键字包括：

表 8. FROM 子句 join-type 关键字

join-type 关键字	描述
CROSS JOIN	返回两个源表的笛卡儿乘积（矢量积）
NATURAL JOIN	比较两个表中所有对应同名列是否等同（特殊等值连接；列的长度和数据类型都相同）
KEY JOIN	将第一个表的外键值 限制为与第二个表的主键值相等
INNER JOIN	丢弃结果表中所有未在两个表中 都具备对应行的行
LEFT OUTER JOIN	保留左表的不匹配行， 但丢弃右表中的不匹配行
RIGHT OUTER JOIN	保留右表的不匹配行， 但丢弃左表中的不匹配行
FULL OUTER JOIN	保留左表和右表中的 所有不匹配行

不要混淆 **FROM** 子句中的逗号方式连接和关键字方式连接。同一查询 可以使用两种方式编写，即分别使用这些连接方式中的一种。请优先使用 ANSI 语法关键字方式连接。

下面的查询使用逗号方式连接：

```
SELECT *
  FROM Products pr, SalesOrders so, SalesOrderItems si
 WHERE pr.ProductID = so.ProductID
       AND pr.ProductID = si.ProductID;
```

同一查询使用关键字方式连接（更为可取）：

```
SELECT *
  FROM Products pr INNER JOIN SalesOrders so
       ON (pr.ProductID = so.ProductID)
     INNER JOIN SalesOrderItems si
       ON (pr.ProductID = si.ProductID);
```

ON 子句用于过滤内部连接、左连接、右连接和完全连接的数据。交叉连接没有 **ON** 子句。在内部连接中，**ON** 子句 等同于 **WHERE** 子句。不过，在外部连接中，**ON** 和 **WHERE** 子句 还是有所差别。在外部连接中，**ON** 子句用于过滤矢量积的行， 然后将通过空值扩展的不匹配行 纳入结果中。**WHERE** 子句则可 对外部连接生成的

匹配行和不匹配行中的行 都予以消除。务必确保所需的不匹配行 不会被 **WHERE** 子句中的谓词消除。

不能在外部连接 **ON** 子句内使用子查询。

有关如何编写 Transact-SQL 兼容连接的信息，请参见《参考：构件块、表和过程》>“与其它 Sybase 数据库的兼容性”。

可通过指定 *userid* 对不同用户 拥有的表进行限定。当前用户所属组拥有的表 在缺省情况下无需指定用户 **ID** 便可找到。

相关名用于为表赋予一个仅供 **SQL** 语句 使用的临时名称。在引用必须由表名限定的列，但表名很长不方便键入时，相关名颇为有用。在同一查询中多次引用同一表时，也有必要使用相关名来 区别各个表的实例。如果没有指定 相关名，则表名将在当前语句中 用作相关名。

如果表表达式中的同一个表 使用了两次相同的相关名，该表按仅列出 一次处理。例如，在下面的语句中：

```
SELECT *
FROM SalesOrders
KEY JOIN SalesOrderItems,
SalesOrders
KEY JOIN Employees
```

SalesOrders 表的两个实例按一个实例处理，因此等效于：

```
SELECT *
FROM SalesOrderItems
KEY JOIN SalesOrders
KEY JOIN Employees
```

与之相反的是，在下面语句中 **Person** 表因具有两个不同的相关名 **HUSBAND** 和 **WIFE**，被视为两个实例处理：

```
SELECT *
FROM Person HUSBAND, Person WIFE
```

无需在 **FROM** 子句中使用一个或多个表或视图，您便可通过 **SELECT** 语句在不创建视图的情况下在组中使用组，或者与组连接。**SELECT** 语句的这种用法 称为派生表。

连接列 需要相似的数据类型来获得最佳性能。

- **性能注意事项** – 在优化程序启用的情况下，Sybase IQ 允许在 **FROM** 子句中使用 16 到 64 个表，具体随查询而定；不过，如果在非常复杂的查询的 **FROM** 子句中使用 16 到 18 个以上的表，则性能可能会受到影响。

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

如果您的查询 不需要 **FROM** 子句，则可以通过添加 **FROM iq_dummy** 子句强制 Sybase IQ 处理查询，在这种情况下，`iq_dummy` 表示在数据库中创建的 包含一行和一列的表。

标准

- SQL — 符合 ISO/ANSI SQL 标准。
- Sybase — 某些版本的 Adaptive Server Enterprise 中不 支持 **JOIN** 子句。这样就必须改用 **WHERE** 子句构建 连接。

权限

必须连接到数据库。

另请参见

- DELETE 语句 （第 163 页）
- SELECT 语句 （第 277 页）

GET DESCRIPTOR 语句 [ESQL]

检索描述符区内变量的信息，或从描述符区的某个变量中检索实际数据。

语法

```
GET DESCRIPTOR descriptor-name  
... { hostvar = COUNT } | VALUE n assignment [, ...] }
```

参数

- **assignment:** - *hostvar*= { TYPE | LENGTH | PRECISION | SCALE | DATA | INDICATOR | NAME | NULLABLE | RETURNED_LENGTH }

示例

- **示例 1** - 有关示例，请参见 “ALLOCATE DESCRIPTOR 语句 [ESQL]”。

用法

值 *n* 指定要在描述符区内检索其信息的变量。

在执行 **GET DESCRIPTOR ... DATA** 时会进行类型检查，以确保宿主变量和描述符变量的数据类型相同。**GET DESCRIPTOR ... DATA** 不支持 LONG VARCHAR 和 LONG BINARY。

如果出现错误，则在 SQLCA 中返回该错误。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Open Client/Open Server 支持。

权限

无

另请参见

- ALLOCATE DESCRIPTOR 语句 [ESQL]（第 5 页）
- DEALLOCATE DESCRIPTOR 语句 [ESQL]（第 152 页）
- SET DESCRIPTOR 语句 [ESQL]（第 290 页）

GOTO 语句 [T-SQL]

分支到带标签的语句。

语法

```
label:  
GOTO label
```

示例

- **示例 1** - 下面的 Transact-SQL 批处理在服务器窗口上四次输出消息 “yes”：

```
declare @count smallint  
select @count = 1  
restart:  
    print 'yes'  
    select @count = @count + 1  
    while @count <=4  
        goto restart
```

用法

Transact-SQL 过程或批处理中的任何语句都可以带标签。标签名是一个有效标识符后跟一个冒号。在 **GOTO** 语句中不使用冒号。

标准

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

权限

无

GRANT 语句

对特定用户授予权限以及创建新用户 ID。

语法

语法 1 - 授予权限

```
GRANT authority, ...  
TO userid, ...
```

```
authority:  
BACKUP  
| DBA  
| GROUP  
| MEMBERSHIP IN GROUP userid [, ...]  
| MULTIPLEX ADMIN  
| OPERATOR  
| PERMS ADMIN  
| PROFILE  
| READCLIENTFILE  
| READFILE  
| [ RESOURCE | ALL ]  
| SPACE ADMIN  
| USER ADMIN  
| VALIDATE  
| WRITECLIENTFILE
```

语法 2 - 在组中授予组状态或成员资格

```
GRANT { GROUP | MEMBERSHIP IN GROUP userid, ... }  
TO userid, ...
```

语法 3 - 授予数据库对象权限

```
GRANT permission, ...  
ON [ owner.]table-name  
TO userid [, ...]  
[ WITH GRANT OPTION ]  
[ FROM userid ]
```

```
permission:  
| ALL [ PRIVILEGES ]  
| ALTER  
| DELETE  
| INSERT  
| REFERENCES [ ( column-name [, ...] ) ]  
| SELECT [ ( column-name [, ...] ) ]  
| UPDATE [ ( column-name, ... ) ]
```

语法 4 - 授予执行权限

```
GRANT EXECUTE ON [ owner.]procedure-name
TO userid [, ...]
```

语法 5 - 授予集成登录权限

```
GRANT INTEGRATED LOGIN TO user_profile_name [, ...]
AS USER userid
```

语法 6 - 授予 Kerberos 登录权限

```
GRANT KERBEROS LOGIN TO client-Kerberos-principal, ...
AS USER userid
```

语法 7 - 授予连接权限

```
GRANT CONNECT TO userid [, ...]
IDENTIFIED BY password [, ...]
```

语法 8 - 授予在 dbspace 中执行创建操作的权限

```
GRANT CREATE ON dbspace_name
TO userid [, ...]
```

示例

- **示例 1** - 为数据库创建两个新用户：

```
GRANT
CONNECT TO Laurel, Hardy
IDENTIFIED BY Stan, Ollie
```

- **示例 2** - 向用户 Laurel 授予对 Employees 表的权限。

```
GRANT
SELECT, INSERT, DELETE
ON Employees
TO Laurel
```

- **示例 3** - 允许用户 Hardy 执行 Calculate_Report 过程。

```
GRANT
EXECUTE ON Calculate_Report
TO Hardy
```

- **示例 4** - 为用户 Lawrence 和 Swift 授予对数据库空间 DspHist 的 CREATE 权限：

```
GRANT
CREATE ON DspHist
TO LAWRENCE, SWIFT
```

- **示例 5** - 为用户 Fiona 和 Ciaran 授予对数据库空间 DspHist 的 CREATE 权限：

```
GRANT CREATE ON DspHist TO Fiona, Ciaran
```

用法

GRANT 语句用于为个别用户 ID 和组授予数据库权限。它也可用于创建和删除用户和组。

GRANT authority 子句 - 授予用户以下权限之一：

- **BACKUP** 权限 – 授予备份数据库的权限。请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “GRANT 语句”。
- **DBA** 权限 – “数据库管理员”授权可赋予用户执行任何操作的权限。此权限通常是组织中负责管理数据库的人员保留的。
- **MULTIPLEX ADMIN** 权限 – 允许用户执行 Multiplex 管理任务，如创建和删除 Multiplex 服务器。有关完整说明，请参见《使用 Sybase IQ Multiplex》> “Multiplex 服务器管理” > “管理权限” > “MULTIPLEX ADMIN 权限”。
- **OPERATOR** 权限 – 允许用户对数据库进行检查点和备份操作，删除连接，以及监控系统。有关完整说明，请参见《系统管理指南第一卷》> “管理用户 ID 和权限”。
- **PERMS ADMIN** 权限 – 允许用户管理数据权限、组、授权和口令。有关完整说明，请参见《系统管理指南第一卷》> “管理用户 ID 和权限”。
- **PROFILE** 权限 – 授予用户执行分析和诊断操作的权限。请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “GRANT 语句”。
- **READCLIENTFILE** 权限 – 允许用户从客户端计算机上的文件中进行读取（例如在装载数据时）。请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “GRANT 语句”。
- **READFILE** 权限 – 允许用户使用 OPENSTRING 子句对文件执行 **SELECT** 语句。请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “GRANT 语句”。
- **RESOURCE** 权限 – 允许用户创建数据库对象，如表、视图和存储过程。在语法 1 中，**ALL** 是 **RESOURCE** 的同义词，它与 Adaptive Server Enterprise 兼容。
- **SPACE ADMIN** 权限 – 允许用户管理数据库空间。有关完整说明，请参见《系统管理指南第一卷》> “管理用户 ID 和权限”。
- **USER ADMIN** 权限 – 允许用户管理用户、外部登录和登录策略。有关完整说明，请参见《系统管理指南第一卷》> “管理用户 ID 和权限”。
- **VALIDATE** 权限 – 允许用户执行多个 **VALIDATE** 语句支持的验证操作，如验证数据库、验证表和索引，以及验证校验和。此权限还允许用户使用 Sybase Central 中的校验实用程序 (**dbvalid**) 和校验数据库向导。请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “GRANT 语句”。
- **WRITECLIENTFILE** 权限 – 允许用户对客户端计算机上的文件进行写入（例如在卸载数据时）。请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “GRANT 语句”。

注意： 这些参考指向 SQL Anywhere 文档。

GROUP 子句 – 允许用户拥有成员。有关完整说明，请参见《系统管理指南第一卷》> “管理用户 ID 和权限”。

MEMBERSHIP IN GROUP 子句 – 允许用户从组中继承表权限和引用组创建的表，而不必限定表名。

如果不希望某个特定用户访问特定的表、视图或过程，则不要使用该用户成为对该对象拥有权限的组的成员。

GRANT permission 子句 - 授予对单个表或视图的权限。可以将表权限列在一起，或者通过指定 **ALL** 一次性授予所有六种权限。如果指定了 **WITH GRANT OPTION**，则指定的用户 ID 还会得到向其他用户 ID **GRANT**（授权）同样权限的授权。

- **ALL** 权限 - 在语法 3 中，授予所有权限。
- **ALTER** 权限 - 用户可以使用 **ALTER TABLE** 语句更改相应表。对视图不允许此权限。
- **DELETE** 权限 - 用户可从相应表或视图删除行。
- **INSERT** 权限 - 用户可向指定的表或视图插入行。
- **REFERENCES** 权限 - 用户可在指定的表上创建索引，并创建引用指定表的外键。如果指定了列名，则用户仅可引用这些指定的列。列的 **REFERENCES** 权限不能授予视图，只能授予表。
- **SELECT** 权限 - 用户可查看相应视图或表中的信息。如果指定了列名，则用户仅可查看这些指定的列。列的 **SELECT** 权限不能授予视图，只能授予表。
- **UPDATE** 权限 - 用户可更新相应视图或表中的行。如果指定了列名，则用户仅可更新这些指定的列。列的 **UPDATE** 权限不能授予视图，只能授予表。若要更新表，用户必须对表拥有 **SELECT** 和 **UPDATE** 权限。

例如，若要向用户 Laurel 授予对 Employees 表的 **SELECT** 和 **UPDATE** 权限：

```
GRANT
SELECT, UPDATE ( street )
ON Employees
TO Laurel
```

EXECUTE ON 子句 - 授予执行过程的权限。

INTEGRATED LOGIN TO 子句 - 在一个或多个 Windows 用户配置文件和一个现有数据库用户 ID 之间创建显式集成登录映射，使成功登录到本地计算机的用户不必提供用户 ID 或口令，就可以连接到数据库。

KERBEROS LOGIN TO 子句 - 创建从一个或多个 Kerberos 主体到现有数据库用户 ID 的 Kerberos 验证登录映射。这使成功登录到 Kerberos 的用户（具有一个有效 Kerberos 票据授予票据的用户）不必提供用户 ID 或口令，就可以连接到数据库。请参见“SQL Anywhere Server - SQL 参考”>“SQL 语句”>“GRANT 语句”。

CONNECT TO 子句 - 创建新用户。**GRANT CONNECT** 也可由任何用户用来更改他们自己的口令。

注意： Sybase 建议使用 **CREATE USER** 语句创建用户。请参见“CREATE USER 语句”。

若要创建用空字符串作为口令的用户：

```
GRANT CONNECT TO userid IDENTIFIED BY ""
```

如果您具有 DBA 或 PERMS ADMIN 权限，可以更改任何现有用户的口令：

```
GRANT CONNECT TO userid IDENTIFIED BY password
```

您还可使用该命令添加新用户。因此，如果您在本来要添加新用户时无意中输入了现有用户的用户 ID，则实际上是在更改现有用户的口令。您不会收到警告，因为这被视为正常操作。此行为与 Sybase IQ 12 版以前的版本有所不同。

为避免出现这种情况，请使用系统过程 **sp_addlogin** 和 **sp_adduser** 添加用户。如果您尝试添加现有用户 ID，与 Adaptive Server Enterprise 和 Sybase IQ 12 版之前的版本一样，这些过程会提示您出错。

注意： 使用系统过程（而不是 **GRANT** 和 **REVOKE**）来添加和删除用户 ID。

若要创建没有口令的用户：

```
GRANT CONNECT TO userid
```

用户 ID 不区分大小写。

没有口令的用户不能连接到数据库。当您创建组且不希望任何人连接到相应的组用户 ID 时，这一限制颇为有用。

如《参考：构件块、表和过程》>“SQL 语言元素”>“标识符”中所述，口令必须为有效的标识符。口令的最大长度为 255 个字节。如果数据库选项

VERIFY_PASSWORD_FUNCTION 已设置为一个值而非空字符串，则 **GRANT CONNECT TO *userid* IDENTIFIED BY *password*** 语句将调用由此选项值标识的函数。如函数返回 NULL，则表明口令符合规则。如果设置了 **VERIFY_PASSWORD_FUNCTION** 选项，则只能使用 **GRANT CONNECT** 语句指定一个 *userid* 和 *password*。请参见“**VERIFY_PASSWORD_FUNCTION** 选项”。

以下名称不能用作有效的数据库用户 ID 和口令：

- 以空格、单引号或双引号开头的名称
- 以空格结尾的名称
- 包含分号的名称

CREATE ON 子句 - 向指定的用户和/或组授予对指定的数据库空间的 **CREATE** 权限。

另请参见“**REVOKE** 语句”。

副作用：

- 自动提交。

注意： 这些参考指向 SQL Anywhere 文档。

标准

- SQL—语法 3 是初级功能。语法 4 是持久存储模块功能。其它语法是 ISO/ANSI SQL 语法的供应商扩展。

- Sybase—Adaptive Server Enterprise 中支持语法 1 和 3。Adaptive Server Enterprise 和 Sybase IQ 中的安全模型不同，因此其它语法也不同。

权限

- 对于语法 1 和 2，必须满足以下条件之一：
 - 要向任何用户授予 **DBA** 权限，您必须具有 **DBA** 权限。
 - 要执行 **GRANT GROUP**、**GRANT MEMBERSHIP IN GROUP** 或向任何用户授予任何其它权限，您必须具有 **DBA** 或 **PERMS ADMIN** 权限。
- 对于语法 3，必须满足以下条件之一：
 - 相应表是您创建的。
 - 已经用 **GRANT OPTION** 授予您对该表的权限。
 - 您具有 **DBA** 或 **PERMS ADMIN** 权限。
- 对于语法 4，必须满足以下条件之一：
 - 相应过程是您创建的。
 - 您具有 **DBA** 或 **PERMS ADMIN** 权限。
- 对于语法 5，您必须具有 **DBA** 或 **USER ADMIN** 权限。
- 对于语法 6，您必须具有 **DBA** 或 **USER ADMIN** 权限。
- 对于语法 7，必须满足以下条件之一：
 - 如果要创建新用户，您必须具有 **DBA** 或 **USER ADMIN** 权限。
 - 您要更改的是您自己的口令。
 - 如果要更改其他用户的口令，您必须具有 **DBA** 或 **PERMS ADMIN** 权限。
 如果要更改其他用户的口令，则相应用户不可连接到数据库。
- 对于语法 8，您必须具有 **DBA** 或 **SPACE ADMIN** 权限。

另请参见

- **CREATE USER** 语句（第 147 页）
- **REVOKE** 语句（第 270 页）
- **VERIFY_PASSWORD_FUNCTION** 选项（第 457 页）

IF 语句

用于按条件执行 SQL 语句。

语法

```
IF search-condition THEN statement-list
... [ ELSEIF search-condition THEN statement-list ]...
... [ ELSE statement-list ]
... END IF
```

示例

- 示例 1 - 下面的过程阐释了 **IF** 语句的用法:

```
CREATE PROCEDURE TopCustomer (OUT TopCompany CHAR(35), OUT
TopValue INT)
BEGIN
    DECLARE err_notfound EXCEPTION
    FOR SQLSTATE '02000' ;
    DECLARE curThisCust CURSOR FOR
    SELECT CompanyName, CAST(      sum(SalesOrderItems.Quantity *
Products.UnitPrice) AS INTEGER) VALUE
    FROM Customers
    LEFT OUTER JOIN SalesOrders
    LEFT OUTER JOIN SalesOrsderItems
    LEFT OUTER JOIN Product
    GROUP BY CompanyName ;

    DECLARE ThisValue INT ;
    DECLARE ThisCompany CHAR(35) ;
    SET TopValue = 0 ;
    OPEN curThisCust ;
    CustomerLoop:
    LOOP
        FETCH NEXT curThisCust
        INTO ThisCompany, ThisValue ;
        IF SQLSTATE = err_notfound THEN
            LEAVE CustomerLoop ;
        END IF ;
        IF ThisValue > TopValue THEN
            SET TopValue = ThisValue ;
            SET TopCompany = ThisCompany ;
        END IF ;
    END LOOP CustomerLoop ;
    CLOSE curThisCust ;
END
```

- 示例 2 - 下面的过程阐释了 **ELSEIF** 语句的用法:

```
BEGIN
    DECLARE X INT;
    SET X = 1;
    IF X = 1 THEN
        PRINT '1';
    ELSEIF X = 2 THEN
        PRINT '2';
    ELSE
        PRINT 'something else';
    ENDIF
END
```

用法

IF 语句可用于按条件执行第一个 *search-condition* 计算结果为 TRUE 的 SQL 语句列表。

如果没有 *search-condition* 计算为 **TRUE**，并且存在 **ELSE** 子句，则执行 **ELSE** 子句中的 *statement-list*。如果没有 *search-condition* 计算为 **TRUE**，并且不存在 **ELSE** 子句，则表达式将返回空值。

执行过程从 **END IF** 之后的第一个语句开始继续。

将变量与 **IF** 语句内的 **SELECT** 语句返回的单个值进行比较时，必须首先将 **SELECT** 的结果赋给另一变量。

注意： 不要混淆 **IF** 语句和 **IF** 表达式的语法。

不能嵌套 **IF** 语句。

有关 **IF** 表达式的信息，请参见《参考：构件块、表和过程》>“SQL 语言元素”>“表达式”。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase—Transact-SQL **IF** 语句的语法略有不同。

权限

无

另请参见

- **BEGIN**……**END** 语句（第 49 页）

IF 语句 [T-SQL]

用于按条件执行 Transact-SQL 语句，可用作 Sybase IQ**IF** 语句的替代项。

语法

```

        IF expression
... statement
... [ ELSE [ IF expression ] statement ]...
```

示例

- **示例 1** - Transact-SQL **IF** 语句的用法：

```

IF (SELECT max(id) FROM sysobjects) < 100
    RETURN
ELSE
    BEGIN
        PRINT 'These are the user-created objects'
        SELECT name, type, id
```

```
FROM sysobjects
WHERE id < 100
END
```

- **示例 2** – Transact-SQL **ELSEIF** 语句的用法：

```
BEGIN
DECLARE @X INT
SET @X = 1
IF @X = 1
PRINT '1'
ELSEIF @X = 2
PRINT '2'
ELSE
PRINT 'something else'
END
```

用法

Transact-SQL **IF** 条件和 **ELSE** 条件各控制一个 SQL 语句或复合语句（在关键字 **BEGIN** 和 **END** 之间）的执行。

与 Sybase IQ**IF** 语句相反，Transact-SQL **IF** 语句中 没有 **THEN**。Transact-SQL 版本 也没有 **ELSEIF** 或 **END IF** 关键字。

将变量与 **IF** 语句内的 **SELECT** 语句返回的单个值进行比较时，必须首先将 **SELECT** 的结果赋给另一变量。

注意： 不能嵌套 **IF** 语句。

标准

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

权限

无

INCLUDE 语句 [ESQL]

将一个文件纳入将由 SQL 源语言预处理器扫描的源程序中。

语法

```
INCLUDE filename
```

参数

- **filename:** - 标识符

用法

INCLUDE 语句与 C 预处理器 **#include** 指令非常相似。

不过 SQL 预处理器读取给定的文件，并将其内容插入输出 C 文件中。因此，如果一个包含文件含有 SQL 预处理器所需的信息，则应使用嵌入式 **SQL INCLUDE** 语句将其纳入。

特别标识了两个文件名：**SQLCA** 和 **SQLDA**。使用嵌入式 SQL 的每个 C 程序都必须要在任何其它嵌入式 SQL 语句之前包含下面的语句：

```
EXEC SQL INCLUDE SQLCA;
```

此语句必须出现在 C 程序中允许使用静态变量声明的位置。许多嵌入式 SQL 语句需要由 SQL 预处理器在 **SQLCA** 包含语句的位置声明的变量（对程序员不可见）。如果使用了任何 **SQLDA**，必须包含 **SQLDA** 文件。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Open Client/Open Server 支持。

权限

无

INSERT 语句

将来自当前数据库其它位置的单行（语法 1）或所选多行（语法 2）插入表中。插入来自其它数据库的所选多行（语法 3）。

语法

语法 1

```
INSERT [ INTO ] [ owner. ] table-name [ ( column-name [, ...] ) ]
... VALUES ( [ expression | DEFAULT, ... ] )
```

or

```
INSERT [ INTO ] [ owner. ] table-name DEFAULT VALUES
```

语法 2

```
INSERT [ INTO ] [ owner. ] table-name [ ( column-name [, ...] ) ]
... insert-load-options insert-select-load-options
... select-statement
```

语法 3

```

INSERT [ INTO ] [ owner.]table-name[ ( column-name [, ...] ) ]
... insert-load-options insert-select-load-options
LOCATION 'servername.dbname'
[ location-options ]
... { { select-statement } | 'select statement' }

```

参数

- **insert-load-options:** - [**LIMIT** *number-of-rows*] [**NOTIFY** *number-of-rows*]
[**SKIP** *number-of-rows*] [**START ROW ID** *number*]
- **insert-select-load-options:** - [**WORD SKIP** *number*] [**IGNORE**
CONSTRAINT *constrainttype* [, ...]] [**MESSAGE LOG** 'string' **ROW LOG**
'string' [**ONLY LOG** *logwhat* [, ...]]] [**LOG DELIMITED BY** 'string']
- **constrainttype:** - { **CHECK** *integer* | **UNIQUE** *integer* | **NULL** *integer* | **FOREIGN**
KEY *integer* | **DATA VALUE** *integer* | **ALL** *integer* }
- **logwhat:** - { **CHECK** | **ALL** | **NULL** | **UNIQUE** | **DATA VALUE** | **FOREIGN KEY** |
WORD }
- **location-options:** - [**ENCRYPTED PASSWORD**] [**PACKETSIZE** *packet-size*]
[**QUOTED_IDENTIFIER** { **ON** | **OFF** }] [**ISOLATION LEVEL** { **READ**
UNCOMMITTED | **READ COMMITTED** | **SERIALIZABLE** }]

示例

- **示例 1** - 将 Eastern Sales 部门添加到数据库。

```

INSERT INTO Departments
(DepartmentID, DepartmentName, DepartmentHeadID)
VALUES (600, 'Eastern Sales', 501)

```

- **示例 2** - 使用部门主管的名字及其部门名称填充表 dept_head:

```

INSERT INTO dept_head (name, dept)
NOTIFY 20
SELECT Surname || ' ' || GivenName
AS name,
dept_name
FROM Employees JOIN Departments
ON EmployeeID= DepartmentHeadID

```

- **示例 3** - 将远程服务器 detroit 上 Sybase IQ 数据库 iqdet 中的 lineitem 表的 l_shipdate 和 l_orderkey 列数据 插入当前数据库中 lineitem 表的 对应列:

```

INSERT INTO lineitem
(l_shipdate, l_orderkey)
LOCATION 'detroit.iqdet'
PACKETSIZE 512
' SELECT l_shipdate, l_orderkey
FROM lineitem '

```

- **示例 4 (仅在 Catalog 存储中)** - 该示例将向表中插入三行。 仅在 Catalog 存储 (SQL Anywhere) 表中支持包含多个值列表的插入。 如果对 IQ 表使用, 则此语法会产生错误。

```
INSERT INTO T (c1,c2,c3)
VALUES (1,10,100), (2,20,200), (3,30,300);
```

- **示例 5 (仅在 Catalog 存储中)** – 在此示例中，INSERT 语句会向包含四列（每列都具有缺省名称）的表中插入三行。仅在 Catalog 存储 (SQL Anywhere) 表中支持包含多个值列表的插入。如果对 IQ 表使用，则此语法会产生错误。

```
INSERT INTO T ()
VALUES (), (), ();
```

用法

语法 1 允许插入包含指定的表达式值的单行。如果未指定列名的列表，则值将以创建时所使用的顺序（与用 **SELECT *** 检索的顺序相同）插入到表列中。行插入到表中的任意位置。（在关系数据库中，表是不排序的）。

语法 2 允许用户使用完全通用的 **SELECT** 语句的结果对表执行大量插入。除非 **SELECT** 语句包含 **ORDER BY** 子句，否则插入将以任意顺序进行。选择列表中的列将按顺序与列列表中指定的列匹配，或者按这些列的创建顺序匹配。

注意： **NUMBER(*)** 函数可在通过 **INSERT** 语句的语法 2 生成主键时使用。请参见《参考：构件块、表和过程》>“SQL 函数”。

语法 3 **INSERT...LOCATION** 是允许您从 Adaptive Server Enterprise 或 Sybase IQ 数据库插入数据的语法 2 的另一种表现方式。在 **LOCATION** 子句中指定的 *servername.dbname* 用于标识 **FROM** 子句的表中的远程服务器和数据库。要使用语法 3，您要连接到的 Adaptive Server Enterprise 或 Sybase IQ 远程服务器必须存在于本地计算机的 Sybase Open Client interfaces 或 *sql.ini* 文件中。

在使用语法 3 的查询中，最多可以插入 2147483647 行。

SELECT 可以使用大括号或多个直单引号分隔。

注意： 在 ODBC 标准中，大括号表示转义序列的开头和结尾，在 ODBC 或 Sybase Central 上下文中可能会导致错误。解决方法是使用单引号来转义 **SELECT** 语句。

本地 Sybase IQ 服务器连接至您在 **LOCATION** 子句中指定的服务器和数据库。将返回远程表的查询结果，本地服务器会将结果插入当前数据库中。如果您没有在 **LOCATION** 子句中指定服务器名称，Sybase IQ 将忽略您指定的任意数据库名称，因为本地服务器中的当前数据库是唯一的选择。

如果 Sybase IQ 连接到了远程服务器，且已使用 **CREATE EXTERNLOGIN** 创建远程登录，使用 **CREATE SERVER** 语句定义远程服务器，**INSERT...LOCATION** 则将使用当前连接的用户 ID 的远程登录。如果未定义远程服务器，或未对当前连接的用户 ID 创建远程登录，Sybase IQ 则使用当前连接的用户 ID 和口令进行连接。

注意： 如果您使用的是当前连接的用户 ID 和口令，并且用户更改了口令，则您必须停止并重新启动服务器，新的口令才能在远程服务器上生效。更改缺省用户 ID 的口令不会影响使用 **CREATE EXTERNLOGIN** 创建的远程登录。

使用 **CREATE EXTERNLOGIN** 语句创建远程登录并使用 **CREATE SERVER** 语句定义远程服务器可以针对 **INSERT...LOCATION** 设置外部登录名和口令，从而任何用户均可

在任何上下文中使用登录名和口令。这可避免可能因难以获得登录名或口令而产生错误，并且是连接到远程服务器的推荐方式。

例如，用户 russid 连接到 Sybase IQ 数据库并执行下面的语句：

```
INSERT local_SQL_Types LOCATION 'asel.aselddb'
{SELECT int_col FROM SQL_Types};
```

在 asel 服务器上，存在口令为 sybase 的用户 ID aseluser。aseluser 是表 SQL_Types 的所有者。在 IQ 服务器上，远程服务器的定义如下：

```
CREATE SERVER asel CLASS 'ASEJDBC'
USING 'system1:4100' ;
```

在 IQ 服务器上，外部登录的定义如下：

```
CREATE EXTERNLOGIN russid TO asel REMOTE LOGIN aseluser IDENTIFIED BY
sybase;
```

INSERT...LOCATION 对于用户 russid，使用用户 ID aseluser 和口令 sybase 连接到远程服务器 asel。

使用 **ENCRYPTED PASSWORD** 参数指定连接到远程服务器时 Open Client Library 缺省口令加密的使用。如果指定了 **ENCRYPTED PASSWORD**，但是远程服务器不支持 Open Client Library 缺省口令加密，则会报告错误，指示使用了无效的用户 ID 或口令。

在用作远程服务器时，Sybase IQ 支持 TDS 口令加密。Sybase IQ 服务器接受由客户端发送的使用加密口令的连接。有关针对口令加密设置的连接属性的信息，请参见针对 Open Server 15.5 的《软件开发工具包 15.5》>“Open Client Library/C 参考手册”>“Client-Library 主题”>“安全性功能”>“Adaptive Server Enterprise 安全性功能”>“安全握手：加密口令”。

注意： 口令加密需要 Open Client 15.0。TDS 口令加密需要 Open Client 15.0 ESD #7 或更高版本。

要使 Sybase IQ 服务器接受具有加密口令的 jConnect 连接，请将 jConnect **ENCRYPT_PASSWORD** 连接属性设置为 true。

PACKETSIZE 参数指定 TDS 包的大小，以字节为单位。在大多数平台上，TDS 包大小缺省为 512 字节。如果应用程序通过网络接收大量的 text 数据或批量数据时，则使用较大的包也许可以显著提高性能。

packet-size 的值必须是 512 的倍数，等于缺省的网络包大小，或者介于缺省网络包大小和最大网络包大小之间。最大网络包大小和缺省网络包大小是 512 的倍数，介于 512 - 524288 字节之间。最大网络包大小始终大于或等于缺省网络包大小。有关网络包大小的详细信息，请参见 Adaptive Server Enterprise 《系统管理指南第一卷》。

如果未指定 **INSERT...LOCATIONPACKETSIZE***packet-size* 或将其指定为零，则会使用相应平台的缺省包大小值。

如果 **INSERT...LOCATION** 在 Sybase IQ 服务器和远程 Sybase IQ 或 Adaptive Server Enterprise 服务器之间传输数据，则 **INSERT...LOCATION TDS PACKETSIZE** 参数的值始终为 512 字节，即使您为 **PACKETSIZE** 指定了不同值也不例外。

注意： 如果指定了错误的包大小（例如 933，不是 512 的倍数），则连接尝试将失败，引发 Open Client **ct_connect** “连接 失败” 错误。任何不成功的连接尝试都会返回 通用的 “连接失败” 消息。 Adaptive Server Enterprise 错误日志可能包含有关连接失败原因的 更多具体信息。

使用 **QUOTED_IDENTIFIER** 指定远程服务器上的 **QUOTED_IDENTIFIER** 选项的设置。缺省设置为 “OFF”。仅在 **SELECT** 语句中的任一标识符用双引号引起的情况下（如下例中使用 “c1” ），才可 将 **QUOTED_IDENTIFIER** 设置为 “ON”：

```
INSERT INTO foo
LOCATION 'ase.database'
QUOTED_IDENTIFIER ON {select "c1" from xxx};
```

使用 **ISOLATION LEVEL** 指定远程服务器连接的隔离级别。

隔离级别	特性
READ UNCOMMITTED	<ul style="list-style-type: none">• 隔离级别 0• 允许读取有写锁定或无写锁定的行• 未应用读取锁定• 无法确保并发事务将不会修改行或回退对行所做的更改
READ COMMITTED	<ul style="list-style-type: none">• 隔离级别 1• 只允许读取没有写锁定的行• 仅为读取当前行获取并保持读取锁定，但当游标离开该行时释放读取锁定• 无法确保数据在事务执行过程中不发生更改
SERIALIZABLE	<ul style="list-style-type: none">• 隔离级别 3• 只允许读取结果中没有写锁定的行• 打开游标时获取读锁定，并一直保持到事务结束为止

请参见 “SQL Anywhere Server - SQL 用法” > “使用事务和隔离级别” > “隔离级别和一致性”。

注意： 此参考指向 SQL Anywhere 文档。

Sybase IQ 不支持 Adaptive Server Enterprise 数据类型 **TEXT**，但是，可从长度大于 255 个字节的 **IQ CHAR** 或 **VARCHAR** 列以及 数据类型为 **TEXT** 的 **ASE** 数据库列执行 **INSERT...LOCATION**（语法 3）。如果 Sybase IQ 支持内部转换，可将 **ASE TEXT** 和 **IMAGE** 列插入到 其它 Sybase IQ 数据类型的列中。缺省情况下，如果远程数据列包含 2GB 以上的数据，Sybase IQ 则会以无提示方式将列值截断为 2GB。

警告! Sybase IQ 不支持 Adaptive Server Enterprise 数据类型 UNICHAR、UNIVARCHAR 或 UNITEXT。在 ISO_BINENG 归类中，从 UNICHAR 或 UNITEXT 到 CHAR 或 CLOB 列中的 **INSERT...LOCATION** 命令可正确执行；如果发生这种情况，则这些列中的数据可能不一致。只有转换失败时，这种情况下才会报告错误。

用户必须专门获得使用许可才能使用 非结构化数据分析选件的大对象变量功能。请参见《Sybase IQ 中的非结构化数据分析》。

注意: 如果使用 **INSERT...LOCATION** 来插入从 **VARBINARY** 列中所选的数据，请将远程数据库的 **ASE_BINARY_DISPLAY** 设置为 OFF。

INSERT...LOCATION (语法 3) 不支持在 **SELECT** 语句中使用变量。

可以对视图执行插入，条件是定义相应视图的 **SELECT** 语句在 **FROM** 子句中只有一个表，且不含 **GROUP BY** 子句、集合函数或涉及 **UNION** 操作。

插入到表中的字符串始终按输入时的大小写存储，而不管数据库是否区分大小写。因此，插入到表中的字符串“Value”在数据库中保存时 V 始终为大写，其余的字母为小写。**SELECT** 语句以 *Value* 形式返回该字符串。但是，如果数据库不区分大小写，所有比较都会使 *Value* 与 *value*、**VALUE** 等相同。而且，如果单列主键已经包含 *Value* 条目，则会拒绝对 *value* 执行 **INSERT**，因为它会导致主键不唯一。

只要执行 **INSERT ... LOCATION** 语句，Sybase IQ 就会装载确定语言、归类序列、字符集和日期/时间格式所需的本地化信息。如果数据库使用平台的非缺省区域设置，则必须在本地客户端设置环境变量以确保 Sybase IQ 装载正确的信息。

如果设置了 LC_ALL 环境变量，则 Sybase IQ 将该变量的值用作区域设置名。如果未设置 LC_ALL，Sybase IQ 将使用 LANG 环境变量的值。如果这两个变量均未设置，Sybase IQ 将使用区域设置文件中的缺省条目。有关示例，请参见《系统管理指南第一卷》>“国际语言和字符集”>“为 INSERT...LOCATION 语句设置区域设置”。

使用 **DEFAULT VALUES** 和 **VALUES** 子句可指定要插入的值。要插入 **CREATE TABLE** 语句中指定的缺省列值，请指定 **DEFAULT VALUES**。指定 **DEFAULT VALUES** 在语义上等同于指定以下显式语法：

```
INSERT [INTO] <tablename>
VALUES(default, default, ..., default)
```

其中，缺省条目数等于表中列数。例如：

```
INSERT INTO table1 DEFAULT VALUES
```

也可以使用 **INSERT VALUES(DEFAULT ...)** 子句插入到 NULL 列。

LIMIT 选项用于指定从查询插入表中的行的最大数目。缺省值为 0，表示无限制。最大数量为 2GB -1。

NOTIFY 选项用于指定每次将多少行成功插入表中后，您将得到通知消息。缺省值为每 100,000 行。

SKIP 选项定义要在用于该插入操作的输入表开头处跳过的行数。缺省值为 0。

START ROW ID 选项指定 IQ 表中应开始插入的行的记录标识号。缺省情况下，在表中有空间的任何位置插入新行，并且每项插入操作都从新行开始执行。

不允许对分区表使用 **LOAD TABLE** 和 **INSERT** 命令的 **START ROW ID** 子句。

有关 *insert-select-load-options* **WORD SKIP**、**IGNORE CONSTRAINT**、**MESSAGE LOG**、**ROW LOG** 和 **LOG DELIMITED BY** 以及 *constrainttype* 和 *logwhat* 参数的信息，请参见“LOAD TABLE 语句”。

用于多列索引的 **INSERT** 必须包含索引的所有列。

对于 **INSERT...VALUES**、**INSERT...SELECT** 和 **INSERT...LOCATION**，Sybase IQ 支持列 **DEFAULT** 值。如果为列指定了 **DEFAULT** 值，则此 **DEFAULT** 值将用作任一未指定列值的 **INSERT**（或 **LOAD**）语句中的列值。

有关在执行插入操作时如何使用列 **DEFAULT** 值的详细信息，请参见《系统管理指南第一卷》>“数据完整性”>“列缺省值有助于保持数据完整性”。

如果存储过程或函数使用 **COMMIT**、**ROLLBACK** 或一些 **ROLLBACK TO SAVEPOINT** 语句，则不允许从该存储过程或函数执行 **INSERT**。有关详细信息，请参见《系统管理指南第二卷》>“使用过程和批处理”>“控制语句”>“原子复合语句”以及《系统管理指南第 2 卷》>“使用过程和批处理”>“过程中的事务和保存点”。

SELECT...FROM 的结果可能略微不同于 **INSERT...SELECT...FROM** 的结果，原因是针对插入期间的优化而进行的不精确数据类型（如 **DOUBLE** 或 **NUMERIC**）的内部数据转换。如果需要更精确的结果，可能的解决方法是将列声明为具有较高精度的 **DOUBLE** 或 **NUMERIC** 数据类型。

另请参见《系统管理指南第一卷》>“导入和导出数据”>“使用 **INSERT** 语句”。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 受 Adaptive Server Enterprise 支持（不含 *insert-load-options*）。

权限

必须有表的 **INSERT** 权限。

另请参见

- **CREATE EXTERNLOGIN** 语句（第 89 页）
- **DELETE** 语句（第 163 页）
- **LOAD TABLE** 语句（第 223 页）
- **SYNCHRONIZE JOIN INDEX** 语句（第 300 页）

INSTALL JAVA 语句

使 Java 类可供在数据库中使用。

语法

```
INSTALL JAVA [ install-mode ] [ JAR jar-name ] FROM source
```

参数

- **install-mode:** - { **NEW** | **UPDATE** }
- **source:** - { **FILE** *filename* | **URL** *url-value* }

示例

- **示例 1** - 通过提供文件名和类的位置，安装用户创建的名为“Demo”的 Java 类：

```
INSTALL JAVA NEW
FROM FILE 'D:\JavaClass\Demo.class'
```

安装后，使用类的名称引用它，不再使用它的原始文件路径位置。例如，下面的语句使用上一语句中安装的类：

```
CREATE VARIABLE d Demo
```

如果 Demo 类是软件包 sybase.work 的成员，必须使用该类的完全限定名：

```
CREATE VARIABLE d sybase.work.Demo
```

- **示例 2** - 安装 zip 文件中包含的所有类，并将数据库中的这些类与 JAR 文件名关联：

```
INSTALL JAVA
JAR 'Widgets'
FROM FILE 'C:\Jars\Widget.zip'
```

不保留 zip 文件的位置，并且必须使用完全限定的类名（包名和类名）引用类。

用法

- **安装模式** - 如果指定 **NEW** 安装模式，则引用的 Java 类必须是新类，而不是目前已安装好的类的更新。如果数据库中存在同名的类而且使用 **NEW** 安装模式，则会出现 错误。

UPDATE 安装模式用于指定引用的 Java 类可以包含 已安装在给定数据库中的 Java 类的 替代类。

要更新类或 JAR，您必须具有 DBA 权限，并且磁盘上的某个文件中要有较新版本的已编译的类文件或 JAR 文件。

只有在安装类之后建立的新连接或者在安装类之后首次使用类的新连接才使用新定义。一旦 Java VM 装载了某个类定义，就会将其一直保留在内存中，直到连接关闭为止。

如果已使用基于当前连接中的某个类的 Java 类或对象，则必须断开连接并重新连接，才能使用新的类定义。

如果省略安装模式，则缺省安装模式为 **NEW**。

- **JAR** - JAR — 指定 *file-name* 或 *text-pointer* 必须指定 JAR 文件或包含 JAR 的列。JAR 文件通常带扩展名 *.jar* 或 *.zip*。

可对已安装好的 JAR 和 zip 文件进行压缩或解压缩。但是，通过 Sun JDK **jar** 实用程序生成的 JAR 文件不受支持。但支持由其他 zip 实用程序生成的文件。

如果指定了 JAR 选项，则安装了 JAR 包含的类后，会以 JAR 的形式保留 JAR。该 JAR 是与这些类的每一个关联的 JAR。用 JAR 选项安装在数据库中的 JAR 称为数据库的保留 JAR。

保留下来的 JAR 会在 **INSTALL** 和 **REMOVE** 语句中引用。保留下来的 JAR 不影响其他 Java-SQL 类应用。SQL 系统用保留下来的 JAR 处理其他系统对与给定数据相关联的类的请求。如果所请求的类具有关联 JAR，则 SQL 系统可以直接提供该 JAR，而不是单个类。

jar-name 是一个字符串值，最长 255 个字节。*jar-name* 用于在后续 **INSTALL**、**UPDATE** 和 **REMOVE** 语句中标识保留下来的 JAR。

- **源代码** - 用于指定 Java 类的安装位置。

对于 *file-name*，受支持的格式包括完全限定文件名（如 “c:\libs\jarname.jar” 和 “/usr/u/libs/jarname.jar”）和相对文件名（相对于数据库服务器的当前工作目录）。

filename 必须标识类文件 或者 JAR 文件。

每个类的类定义都是在首次使用该类时由每个连接的 VM 载入的。当您 **INSTALL** 类时，将隐式重新启动连接上的 VM。因此，无论 **INSTALL** 的 *install-mode* 是 **NEW** 还是 **UPDATE**，都可以直接访问新类。

对于其他连接，VM 下次第一次访问新类时会装入新类。如果新类已由 VM 装入，则为该连接重新启动 VM（例如，用 **STOP JAVA** 和 **START JAVA** 重新启动）后，才能用该连接看到新类。

标准

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

权限

- 执行 **INSTALL** 语句需要 DBA 权限。

- 任何用户可以任何方式引用所有已安装的类。

另请参见

- REMOVE 语句（第 260 页）

IQ UTILITIES 语句

收集有关 Sybase IQ 数据库缓冲区高速缓存的统计信息。

语法

```
IQ UTILITIES { MAIN | PRIVATE }
[ INTO ] table-name
{ START MONITOR [ 'monitor-options' ]
| STOP MONITOR }
```

参数

- **monitor-options**: - { **-summary** | { **-append** | **-truncate** } **-bufalloc** | **-cache** | **-cache_by_type** | **-contention** | **-debug** | **-file_suffix.suffix** | **-io** | **-intervalseconds** | **-threads** }...

示例

- **示例 1** - 启动缓冲区高速缓存监视器并记录 IQ 临时缓冲区高速缓存的活动：

```
IQ UTILITIES PRIVATE INTO monitor START MONITOR '-cache -interval
20'
```

用法

START MONITOR 启动 IQ 缓冲区高速缓存监控器。对于 **START** 和 **STOP MONITOR**, *table_name* 为虚拟表。可以指定任何 IQ 基表或临时表，但是最好准备一个表专用于监控。结果将存储在一个文本文件中，对于 **MAIN** 缓冲区高速缓存结果，为 `dbname.connection#-main-iqmon`；对于 **PRIVATE**（临时）缓冲区高速缓存结果，则为 `dbname.connection#-temp-iqmon`。请从同一数据库再次运行监视器，连接数将覆盖以前的结果。要设置监视器输出文件的目录位置，请设置 **MONITOR_OUTPUT_DIRECTORY** 选项。

monitor-options 定义结果的内容和频率。可以指定多个选项，且必须用引号将其引起来。

- **-summary** 显示主缓冲区高速缓存和临时（专用）缓冲区高速缓存的摘要信息。此选项是缺省设置。
- **-append** | **-truncate** 分别附加到现有输出文件或截断现有输出文件。Truncate 是缺省值。

- **-bufalloc** 显示有关主缓冲区分配器或临时缓冲区分配器的信息，该分配器可在缓冲区高速缓存中保留空间以供排序、散列和位图等对象使用。
- **-cache** 详细显示主缓冲区高速缓存或临时缓冲区高速缓存的活动。
- **-cache_by_type** 按 IQ 页面类型详细显示主缓冲区高速缓存或临时缓冲区高速缓存的活动。该格式主要用于向 Sybase 技术支持部门提供信息。
- **-contention** 显示很多关键的缓冲区高速缓存和内存管理器锁。
- **-debug** 显示性能监视器可用的所有信息，而不考虑是否存在标准显示模式来显示同样的信息。该选项主要用于向 Sybase 技术支持部门提供信息。
- **-file_suffixsuffix** 可创建名为 <dbname>.<connid>-<main_or_temp>-<suffix> 的监控输出文件。缺省值为 iqmon。
- **-io** 显示主缓冲区高速缓存或临时缓冲区高速缓存 I/O 速率和数据压缩比率。
- **-interval** 指定报告间隔（以秒为单位）。缺省值为每 60 秒。最小值为每 2 秒。
- **-threads** 显示有关正在处理的线程的信息。

有关利用 **IQ UTILITIES** 创建可扩展 Sybase IQ 系统存储过程功能的过程的高级应用，请参见

- 《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqsysmon 过程”
- 《性能和调优指南》>“监控和调优性能”
- 《系统管理指南第二卷》>“使用过程和批处理”
-

标准

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

权限

无

另请参见

- **MONITOR_OUTPUT_DIRECTORY** 选项（第 407 页）

LEAVE 语句

退出复合语句或 **LOOP** 以继续执行。

语法

```
LEAVE statement-label
```

示例

- **示例 1** - 下面的代码段显示如何使用 **LEAVE** 语句退出循环:

```
SET i = 1;
lbl:
LOOP
    INSERT
    INTO Counters ( number )
    VALUES ( i ) ;
    IF i >= 10 THEN
        LEAVE lbl ;
    END IF ;
    SET i = i + 1
END LOOP lbl
```

- **示例 2** - 下面的代码段在嵌套循环中使用 **LEAVE**:

```
outer_loop:
LOOP
    SET i = 1;
    inner_loop:
    LOOP
        ...
        SET i = i + 1;
        IF i >= 10 THEN
            LEAVE outer_loop
        END IF
    END LOOP inner_loop
END LOOP outer_loop
```

用法

LEAVE 是控制语句，它允许您退出带标签的复合语句或带标签的循环。执行在复合语句或循环之后的第一个语句处重新开始。

复合语句是过程的主体，它具有与过程同名的隐式标签。

标准

- SQL - 符合 ISO/ANSI SQL 标准。
- Sybase - 不受 Adaptive Server Enterprise 支持。BREAK 语句为 Transact-SQL 兼容过程提供类似的功能。

权限

无

另请参见

- BEGIN……END 语句 (第 49 页)
- FOR 语句 (第 191 页)
- LOOP 语句 (第 241 页)

LOAD TABLE 语句

将数据从外部文件导入数据库表。

语法

```
LOAD [ INTO ] TABLE [ owner.]table-name
... ( load-specification [, ...] )
... { FROM | USING [ CLIENT ] FILE }
{ 'filename-string' | filename-variable } [, ...]
... [ CHECK CONSTRAINTS { ON | OFF } ]
... [ DEFAULTS { ON | OFF } ]
... [ QUOTES OFF ]
... ESCAPES OFF
... [ FORMAT { ascii | binary | bcp } ]
... [ DELIMITED BY 'string' ]
... [ STRIP { ON | OFF | RTRIM } ]
... [ WITH CHECKPOINT { ON | OFF } ]
... [ BYTE ORDER { NATIVE | HIGH | LOW } ]
... [ LIMIT number-of-rows ]
... [ NOTIFY number-of-rows ]
... [ ON FILE ERROR { ROLLBACK | FINISH | CONTINUE } ]
... [ PREVIEW { ON | OFF } ]
... [ ROW DELIMITED BY 'delimiter-string' ]
... [ SKIP number-of-rows ]
... [ HEADER SKIP number [ HEADER DELIMITED BY 'string' ] ]
... [ WORD SKIP number ]
... [ START ROW ID number ]
... [ ON PARTIAL INPUT ROW { ROLLBACK | CONTINUE } ]
... [ IGNORE CONSTRAINT constrainttype [, ...] ]
... [ MESSAGE LOG 'string' ROW LOG 'string' [ ONLY LOG logwhat
[, ...] ] ]
... [ LOG DELIMITED BY 'string' ]
```

参数

- **load-specification**: - { *column-name* [*column-spec*] | **FILLER** (*filler-type*) }
- **column-spec**: - { **ASCII** (*input-width*) | **BINARY** [**WITH NULL BYTE**] | **PREFIX** { **1** | **2** | **4** } | 'delimiter-string' | **DATE** (*input-date-format*) | **DATETIME** (*input-datetime-format*) | **ENCRYPTED** (*data-type* 'key-string' [, 'algorithm-string']) | **DEFAULT** *default-value* } [**NULL** ({ **BLANKS** | **ZEROS** | 'literal', ... })]
- **filler-type**: - { *input-width* | **PREFIX** { **1** | **2** | **4** } | 'delimiter-string' }
- **constrainttype**: - { **CHECK** *integer* | **UNIQUE** *integer* | **NULL** *integer* | **FOREIGN KEY** *integer* | **DATA VALUE** *integer* | **ALL** *integer* }
- **logwhat**: - { **CHECK** | **ALL** | **NULL** | **UNIQUE** | **DATA VALUE** | **FOREIGN KEY** | **WORD** }

示例

- **示例 1** – 将数据从一个文件装载到 Windows 系统上的 Products 表中。使用制表符作为列分隔符，后跟 Description 和 Color 列。

```
LOAD TABLE Products
( ID ASCII(6),
  FILLER(1),
  Name   ASCII(15),
  FILLER(1),
  Description  '\x09',
  Size   ASCII(2),
  FILLER(1),
  Color   '\x09',
  Quantity  PREFIX 2,
  UnitPrice  PREFIX 2,
  FILLER(2) )
FROM 'C:\\mydata\\source1.dmp'
QUOTES OFF
ESCAPES OFF
BYTE ORDER LOW
NOTIFY 1000
```

- **示例 2** – 从客户端计算机上的文件 a.inp 装载数据:

```
LOAD TABLE t1(c1,c2,filler(30))
USING CLIENT FILE 'c:\\client-data\\a.inp'
QUOTES OFF ESCAPES OFF
IGNORE CONSTRAINT UNIQUE 0, NULL 0
MESSAGE LOG 'c:\\client-data\\m.log'
ROW LOG 'c:\\client-data\\r.log' ONLY LOG UNIQUE
```

- **示例 3** – 将数据从两个文件装载到 UNIX 系统上的 product_new 表中（该表允许空值）。制表符为缺省的列分隔符，并且使用换行符作为行分隔符:

```
LOAD TABLE product_new
( id,
  name,
  description,
  size,
  color   '\x09'  NULL( 'null', 'none', 'na' ),
  quantity  PREFIX 2,
  unit_price  PREFIX 2 )
FROM '/s1/mydata/source2.dump',
     '/s1/mydata/source3.dump'
QUOTES OFF
ESCAPES OFF
FORMAT ascii
DELIMITED BY '\x09'
ON FILE ERROR CONTINUE
ROW DELIMITED BY '\n'
```

- **示例 4** – 忽略 10 字长度违规；出现第 11 字时，配置新错误并回退装载:

```
load table PTAB1(
  ck1      ',' null ('NULL') ,
  ck3fk2c2      ',' null ('NULL') ,
  ck4      ',' null ('NULL') ,
  ck5      ',' null ('NULL') ,
```



```

        ck6c1      ',' null ('NULL') ,
        ck6c2      ',' null ('NULL') ,
        rid        ',' null ('NULL') )
FROM 'ri_index_selfRI.inp'
row delimited by '\n'
LIMIT 14 SKIP 10
IGNORE CONSTRAINT UNIQUE 2, FOREIGN KEY 8
word skip 10 quotes off escapes off strip
off

```

- **示例 5** – 使用 **FORMAT BCP** 装载选项从 **BCP** 字符文件 `bcp_file.bcp` 将数据装载到表 `t1`:

```

LOAD TABLE t1 (c1, c2, c3)
FROM 'bcp_file.bcp'
FORMAT BCP
...

```

- **示例 6** – 使用 **DEFAULT** 装载选项，将缺省值 12345 装载到 `c1`，并使用 `LoadConst04.dat` 文件中的数据装载 `c2` 和 `c3`:

```

LOAD TABLE t1 (c1 DEFAULT '12345 ' , c2, c3, filler(1))
FROM 'LoadConst04.dat'
STRIP OFF
QUOTES OFF
ESCAPES OFF
DELIMITED BY ',' ;

```

- **示例 7** – 使用 **FORMAT BCP** 装载选项，利用文件 `bcp_file.bcp` 中的数据装载 `c1` 和 `c2`，并将 `c3` 设置为值 10:

```

LOAD TABLE t1 (c1, c2, c3 DEFAULT '10' )
FROM 'bcp_file.bcp'
FORMAT BCP
QUOTES OFF
ESCAPES OFF;

```

- **示例 8** – 以下代码段在标题行用 “&&” 分隔的数据文件开头处会忽略一个标题行:

```

LOAD TABLE
...HEADER SKIP 1 HEADER DELIMITED by '&&'

```

- **示例 9** – 以下代码段在每个标题行用 “\n” 分隔的数据文件开头处会忽略 2 个标题行:

```

LOAD TABLE
...HEADER SKIP 2

```

用法

使用 **LOAD TABLE** 语句允许从带有 ASCII 或二进制数据的文件到数据库表的高效批量插入。

使用 **LOAD TABLE** 选项也可以控制违反完整性约束时的装载行为并且记录有关违规的信息。

对临时表可以使用 **LOAD TABLE**，但是该临时表必须是用 **ON COMMIT PRESERVE ROWS** 声明的，否则下一条 **COMMIT** 将删除已装载的行。

您还可以指定多个文件来装载数据。在 **FROM** 子句中，指定每个 *filename-string*（各项之间以逗号分隔）。由于资源约束，Sybase IQ 无法保证能够装载所有数据。如果资源分配失败，整个装载事务将被回退。一次读取一个文件，并按 **FROM** 子句中指定的顺序处理这些文件。任何 **SKIP** 或 **LIMIT** 值都只在装载开始时应用，而不是针对每个文件应用。

注意： 当装载 Multiplex 数据库时，请在所有文件名中使用绝对（完全限定）路径。请不要使用相对路径名。

LOAD TABLE 支持装载大对象 (LOB) 数据。请参见《Sybase IQ 中的非结构化数据分析》。

Sybase IQ 支持从 ASCII 和二进制数据装载，并且支持固定长度格式和可变长度格式。要处理所有这些格式，您必须提供一个 *load-specification* 来通知 Sybase IQ 需要从源文件中的每个“列”或字段获得何种数据。使用 *column-spec* 可以定义下列格式：

- 固定字节长度的 ASCII。*input-width* 值为表示每条记录中输入字段的固定宽度的整数值（单位为字节）。
- 使用一定数量的 **PREFIX** 字节（1、2 或 4）指定二进制输入长度的二进制或非二进制字段。

PREFIX 子句有两个相关部分：

- 前缀值 - 始终为二进制值。
- 关联数据字节 - 始终为字符格式，永远不采用二进制格式。

如果使用提取功能卸载数据时将 **TEMP_EXTRACT_BINARY** 选项设置为 **ON**，则在装载二进制数据时必须针对每个列使用 **BINARY WITH NULL BYTE** 参数。

- 由分隔符分隔的可变长度字符。可以将终结符指定为十六进制 ASCII 字符。*delimiter-string* 可以是最多 4 个字符的任意字符串，包括可打印字符的任意组合以及表示不打印字符的任意 8 位十六进制 ASCII 代码。例如，指定：
 - '\x09'，以将制表符表示为终结符。
 - '\x00'，表示空终结符（如同“C”字符串中一样，没有可见终结符）。
 - '\x0a'，表示作为终结符的换行符。还可以使用特殊字符组合“\n”作为换行符。

注意： 分隔符字符串长度可为 1 到 4 个字符，不过，您在 **DELIMITED BY** 子句中只能指定单个字符。对于 **BCP**，分隔符最多可达 10 个字符。

- **DATE** 或 **DATETIME** 字符串（ASCII 字符形式）。您必须使用 Sybase IQ 支持 的日期数据类型和日期时间数据类型的对应格式之一 定义字符串的 *input-date-format* 或 *input-datetime-format*。使用 **DATE** 表示日期值，并使用 **DATETIME** 表示日期时间和时间值。

表 9. 日期和时间格式设置

选项	含义
yyyy 或 YYYY yy 或 YY	表示年份数字。缺省为当前年份。
mm 或 MM	表示 月份数字。适当情况下始终在月份数字中使用前导零或空白，例如，“05”表示 5 月。 DATE 值 必须包含月份。例如，如果您输入的 DATE 值为 1998，则会收到错误提示。如果您输入“03”，则 Sybase IQ 将应用缺省年份和日期，并将其转换为“1998-03-01”。
dd 或 DD jjj 或 JJJ	表示日期数字。缺省日期为 01。适当情况下始终在日期数字中使用前导零，例如，“01”表示 1 号。J 或 j 表示一年中的儒略日（1 至 366）。
hh HH	表示小时。小时基于 24 小时制。请始终对小时使用前导零或空（如有必要），例如，“01”表示凌晨 1 点。“00”也是有效值，表示中午 12 点。
nn	表示分钟。适当情况下始终在分钟中使用前导零，例如，“08”表示 8 分钟。
ss[.ssssss]	表示秒和秒的小数部分。
aa	表示 AM 或 PM 指示。
pp	仅当需要时 表示 p.m. 标记。（这与 12.0 之前的 Sybase IQ 版本 不兼容；先前“pp”与“aa”是 同义词。）
hh	Sybase IQ 假设分钟和秒钟 均为零。例如，如果您输入的 DATETIME 值为“03”，则 Sybase IQ 将该值转换为“03:00:00.0000”。
hh:nn 或 hh:mm	Sybase IQ 假设秒钟 为零。例如，如果您输入的时间值为“03:25”，则 Sybase IQ 将该值转换 为“03:25:00.0000”。

表 10. 示例 **DATE** 和 **DATETIME** 格式选项

输入数据	格式规范
12/31/98	DATE ('MM/DD/YY')
19981231	DATE ('YYYYMMDD')
123198140150	DATETIME ('MMDDYYhhnnss')
14:01:50 12-31-98	DATETIME ('hh:mm:ss MM-DD-YY')
18:27:53	DATETIME ('hh:mm:ss')
12/31/98 02:01:50AM	DATETIME ('MM/DD/YY hh:mm:ssaa')

Sybase IQ 具有针对常用日期、时间和日期时间格式的装载优化。如果您要装载的数据与其中一种格式匹配，则使用相应的格式可以显著缩短装载时间。有关这些格式的列表以及有关装载日期和日期时间数据时优化性能的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”。

还可以将日期/时间字段指定为 **ASCII** 固定宽度字段（如上所述），并使用 **FILLER(1)** 选项以跳过列分隔符。有关指定日期和时间数据的详细信息，请参见《参考：构件块、表和过程》>“**SQL 数据类型**”>“日期和时间数据类型”或《系统管理指南第一卷》>“导入和导出数据”。

装载到表列中时，*column-spec* 的 **NULL** 部分指示如何将某些输入值视为 **NULL** 值。这些字符可能包括 **BLANKS**、**ZEROS** 或您定义的任何其它文字列表。指定空值或从源文件读取空值时，目标列必须能够包含 **NULL**。

ZEROS 含义如下：如果（并且仅当）输入数据（转换前，如果为 **ASCII**）均为二进制零（而不为字符零），则此单元设置为 **NULL**。

- 如果输入数据为字符零，则：
 1. **NULL (ZEROS)** 永远不会导致单元变为 **NULL**。
 2. **NULL ('0')** 导致单元变为 **NULL**。
- 如果输入数据为二进制零（清除所有位），则：
 1. **NULL (ZEROS)** 导致单元变为 **NULL**。
 2. **NULL ('0')** 永远不会导致单元变为 **NULL**。

例如，如果 **LOAD** 语句包含 `col1 date('yyymmdd') null(zeros)` 并且日期为 000000，则会收到错误消息，指示 000000 无法转换为 **DATE(4)**。若要使 **LOAD TABLE** 在数据为 000000 时在 `col1` 中插入空值，请将 **NULL** 子句编写为 `null('000000')`，或将数据修改为等同二进制零并使用 **NULL(ZEROS)**。

如果 **VARCHAR** 单元的长度为零 并且该单元不为空值，则将获得一个零长度单元。对于所有其它数据类型来说，如果单元长度为零，则 **Sybase IQ** 将插入 **NULL**。这是 **ANSI** 行为。对于零长度字符数据的非 **ANSI** 处理，请设置 **NON_ANSI_NULL_VARCHAR** 数据库 选项。

使用 **DEFAULT** 选项可指定装载缺省列值。即使列未在表模式中定义缺省值，您仍可以向该列装载缺省值。在装载时，此功能可提供更大的灵活性。

- 要使用 **LOAD TABLE DEFAULTS** 语句中指定的缺省值，必须将 **LOAD TABLE** 选项设置为 **ON**。如果 **DEFAULTS** 选项为 **OFF**，则不会使用指定的装载缺省值，而会在列中插入空值。
- **LOAD TABLE** 命令必须至少包含一个需要从 **LOAD TABLE** 命令中指定的文件中进行装载的列。否则将报告错误，并且不会执行装载。
- 指定的装载缺省值必须符合支持的列缺省值和缺省值限制，如《系统管理指南第一卷》>“数据完整性”>“列缺省值有助于保持数据完整性”中描述的一样。**LOAD TABLE DEFAULT** 选项不支持 **AUTOINCREMENT**、**IDENTITY** 或 **GLOBAL AUTOINCREMENT** 作为装载缺省值。
- **LOAD TABLE DEFAULT default-value** 必须与数据库的字符集相同。
- 对于在 **LOAD TABLE DEFAULT** 子句中指定的装载缺省值，不支持对缺省值进行加密。
- 对于在表中插入的每个行，将计算因评估指定装载缺省值而引起的约束冲突的总数。

load-specification 的另一重要部分是 **FILLER** 选项。此选项指示您希望跳过源输入文件中的指定字段。例如，在输入文件中的行末尾或整个字段可能会存在不希望添加到表中的字符。与 *column-spec* 定义一样，使用 **FILLER** 可以指定 **ASCII** 固定字节长度、由分隔符分隔的可变长度字符，以及使用 **PREFIX** 字节的二进制字段。

filename-string 以字符串的形式传递到服务器。因此，该字符串遵循的格式要求与其它 SQL 字符串相同。需要特别指出的是：

- 要在 Windows 系统中指示目录路径，反斜杠字符 \ 必须用两个反斜杠来表示。因此，要将数据从文件 `c:\temp\input.dat` 装载到 **Employees** 表的语句是：

```
LOAD TABLE Employees
FROM 'c:\\temp\\input.dat' ...
```

- 路径名对应于数据库服务器，而不是客户端应用程序。如果在某一其它计算机的数据库服务器上运行此语句，则目录名是指服务器计算机上的目录，而不是指客户端计算机上的目录。

每个语句子句的说明跟在以下命令之后：

USING— **USING FILE** 从服务器装载一个或多个文件。此子句的作用等同于指定 **FROM filename** 子句。 **USING CLIENT FILE** 从客户端批量装载一个或多个文件。客户端上的文件的字符集必须与服务器归类的字符集相同。 **Sybase IQ** 以串行方式处理文件列表中的文件。每个文件在处理时将锁定为读取模式，然后解锁。客户端批量装载不需要管理开销（如额外的磁盘空间、内存或网络监控守护程序要求）。

批量装载大对象时， **USING CLIENT FILE** 子句同时适用于主文件和辅助文件。（如果您有非结构化数据分析选件，请参见《**Sybase IQ** 中的非结构化数据分析》了解详细信息。）

在客户端装载过程中，将在客户端主机上创建 **IGNORE CONSTRAINT** 日志文件，创建日志文件过程中出现的任何错误会导致操作回退。

客户端批量装载受使用命令序列协议的 **Interactive SQL** 和 **ODBC/JDBC** 客户端支持，而不受使用 **TDS** 协议的客户端支持。为了确保网络数据安全，请使用传送层安全性。要控制谁能够使用客户端批量装载，请使用安全功能 **(-sf)** 服务器启动开关、 **ALLOW_READ_CLIENT_FILE** 数据库选项和/或 **READCLIENTFILE** 访问控制。

请参见“**SQL Anywhere Server - SQL 用法**” > “远程数据和批量操作” > “导入和导出数据” > “访问客户端计算机上的数据” > “客户端数据安全性”和“**SQL Anywhere Server - SQL 用法**” > “远程数据和批量操作” > “导入和导出数据” > “访问客户端计算机上的数据” > “客户端数据安全性”。

注意： 这些参考指向 **SQL Anywhere** 文档。

LOAD TABLE FROM 子句已废弃，但可使用该子句指定服务器上存在的某个文件。

以下示例从客户端计算机上的文件 `a.inp` 装载数据。

```
LOAD TABLE t1(c1,c2,filler(30))
USING CLIENT FILE 'c:\\client-data\\a.inp'
QUOTES OFF ESCAPES OFF
```

```
IGNORE CONSTRAINT UNIQUE 0, NULL 0
MESSAGE LOG 'c:\\client-data\\m.log'
ROW LOG 'c:\\client-data\\r.log'
ONLY LOG UNIQUE
```

CHECK CONSTRAINTS—此选项缺省值为 ON。指定 **CHECK CONSTRAINTS ON** 时，将评估检查约束，您可以任意忽略或记录评估结果。

设置 **CHECK CONSTRAINTS OFF** 将导致 Sybase IQ 忽略所有检查约束违规。这很有用，例如在重建数据库时。如果表的检查约束调用尚未创建的用户定义的函数，则重建将失败，除非此选项设置为 **OFF**。

此选项与以下选项互斥。如果在同一装载中指定了以下任何选项，将导致错误：

- **IGNORE CONSTRAINT ALL**
- **IGNORE CONSTRAINT CHECK**
- **LOG ALL**
- **LOG CHECK**

DEFAULTS—如果 **DEFAULTS** 为 ON（缺省值）并且列具有缺省值，则使用该值。如果 **DEFAULTS** 选项为 OFF，则将任何未在列表中显示的值指定为 NULL。

DEFAULTS 选项的设置适用于所有列 **DEFAULT** 值，包括 **AUTOINCREMENT**。

有关列的 **DEFAULT** 值在装载和插入中的用法的详细信息，请参见《系统管理指南第一卷》>“数据完整性”>“列缺省值有助于保持数据完整性”。

QUOTES—此参数是可选的，缺省值为 ON。将 **QUOTES** 选项打开时，**LOAD TABLE** 要求将输入字符串引在引号字符内。引号字符是撇号（单引号）或引号（双引号）。字符串中出现的第一个这样的字符将被视为该字符串的引号字符。字符串数据必须以匹配的引号结束。

在 **QUOTES ON** 的情况下，可以在列值中包括列或行分隔符。假定前导引号字符和结尾引号字符不是值的一部分并从装载的数据值中排除。

要在使用 **QUOTES ON** 的值中包括引用符号，必须使用两个引号。例如，下面的行在第三列中包含一个单引号字符值：

```
'123 High Street, Anytown', '(715)398-2354', ' ' ' '
```

在打开 **STRIP** 时（缺省情况），首先将从值中去除尾随空白，然后再插入这些值。仅去除不带引号的字符串的尾随空白。带引号的字符串保留其尾随空白。仅当 **QUOTES** 设置为 时，才会剪裁前导空白或 **TAB** 字符。

数据提取工具提供多个用于处理引号的选项（**TEMP_EXTRACT_QUOTES**、**TEMP_EXTRACT_QUOTES_ALL** 和 **TEMP_EXTRACT_QUOTE**）。如果您计划提取数据以便将其以缺省 **ASCII** 提取方式装载到 **IQ** 表以及包含列或行分隔符的字符串字段中，请使用 **TEMP_EXTRACT_BINARY** 选项提取并对 **LOAD TABLE** 使用 **FORMAT binary** 和 **QUOTES OFF** 选项。

限制：

- **QUOTES ON** 仅适用于列分隔的 ASCII 字段。
- 在 **QUOTES ON** 的情况下，列分隔符或行终结符的第一个字符不能是单引号或双引号。
- **QUOTES** 选项不适用于从辅助文件装载二进制大对象 (BLOB) 或字符大对象 (CLOB) 数据，而无论其设置如何。前导或尾随引号将作为 CLOB 数据的一部分来装载。使用 **QUOTES ON** 选项，位于引号之间的两个连续引号将作为两个连续引号进行装载。
- Adaptive Server Enterprise BCP 不支持选项 **QUOTES**。拷入或拷出所有字段数据时等同于 **QUOTES OFF** 设置。因为 Sybase IQLOAD TABLE 语句的缺省设置是 **QUOTES ON**，所以在将 ASE 数据从 BCP 输出导入到 Sybase IQ 表时必须指定 **QUOTES OFF**。

例外：

- 如果 **LOAD TABLE** 在引起来的字段的结尾引号字符之后遇到非空白字符，将报告以下错误并回退装载操作：

在括起来的字段的结束引号字符后发现非空白文本。SQLSTATE: QTA14 SQLCODE: -1005014L

- 在 **QUOTES ON** 的情况下，如果将单引号或双引号指定为列分隔符的第一个字符，则将报告错误并且装载操作会失败：

在 **QUOTES** 选项设置为 **ON** 的情况下，列分隔符或行终结符的第一个字符不能是单引号或双引号。SQLSTATE: QCA90 SQLCODE: -1013090L

有关 **QUOTES** 选项的示例，请参见《系统管理指南第一卷》>“导入和导出数据”>“使用 **LOAD TABLE** 语句进行批量装载”。

ESCAPES—如果省略输入字段的 *column-spec* 定义，则在 **ESCAPES** 为 **ON** (缺省值) 的情况下，跟在反斜杠字符后的字符将由数据库服务器识别并解释为特殊字符。可以将换行符以组合形式 `\n` 包括在内，其它字符则可以作为十六进制 ASCII 代码包括在数据中，例如 `\x09` 表示制表符。两个连续的反斜线字符 (`\\`) 被解释为单个反斜线。对于 Sybase IQ，必须设置 **ESCAPES OFF**。

FORMAT—Sybase IQ 支持 ASCII 和二进制输入字段。此格式通常由上述 *column-spec* 定义。如果省略列的此定义，缺省情况下，Sybase IQ 使用此选项定义的格式。假定输入行具有 **ascii** (缺省值) 或 **binary** 字段，每行各一个输入行，并使用列分隔符来分隔各个值。

有关 Sybase IQ 使用 **FORMAT BINARY** 和 **BINARY** 列规范子句生成可供 **LOAD TABLE** 语句读取的数据文件所使用的二进制格式的详细说明，请参见《系统管理指南第一卷》>“导入和导出数据”>“二进制装载格式”。

Sybase IQ 还接受 BCP 字符文件中的数据作为 **LOAD TABLE** 命令的输入。

- 使用 **LOAD TABLE FORMAT BCP** 语句装载到 Sybase IQ 表的 BCP 数据文件必须使用 **-c** 选项以跨平台文件格式导出 (**BCP OUT**)。
- 对于 **FORMAT BCP**，**LOAD TABLE** 语句的缺省列分隔符为 <制表符>，缺省行终结符为 <换行符>。

- 对于 **FORMAT BCP**，行中的最后一列必须以行终结符终止，而不能由列分隔符终止。如果列分隔符位于行终结符之前，则将列分隔符视为数据的一部分。
- 装载规范中除最后一列以外的所有列中的数据只能以列分隔符分隔。对于除最后一列以外的所有列，如果在列分隔符之前先遇到行终结符，则将行终结符视为列数据的一部分。
- 可通过 **DELIMITED BY** 子句指定列分隔符。对于 **FORMAT BCP**，分隔符的长度必须小于或等于 10 个字符。如果分隔符长度大于 10，则返回错误。
- 对于 **FORMAT BCP**，装载规范只能包含列名、**NULL** 和 **ENCRYPTED**。如果在装载规范中指定了任何其它选项，则返回错误。

例如，以下 **LOAD TABLE** 装载规范为有效规范：

```
LOAD TABLE x( c1, c2 null(blanks), c3 ) FROM 'bcp_file.bcp' FORMAT
BCP ...
```

```
LOAD TABLE x( c1 encrypted(bigint,'KEY-ONE','aes'), c2, c3 ) FROM
'bcp_file.bcp' FORMAT BCP ...
```

有关 **LOAD TABLE ENCRYPTED** 子句的信息，请参见“Sybase IQ 中的高级安全性”。

DELIMITED BY—如果在 *column-spec* 定义中省略列分隔符，则缺省的列分隔符为逗号。通过提供单个 ASCII 字符或十六进制字符表示形式可指定其它列分隔符。

DELIMITED BY 子句是：

```
... DELIMITED BY '\x09' ...
```

若要将换行符作为分隔符使用，您可以指定特殊组合“\n”或其 ASCII 值“\x0a”。虽然可以在 *column-spec delimiter-string* 中指定最多 4 个字符，但在 **DELIMITED BY** 子句中只能指定单个字符。

STRIP—**STRIP** 子句指定在插入不带引号的值之前是否应去除其尾随空白。**LOAD TABLE** 命令接受以下 **STRIP** 关键字：

- **STRIP OFF**—不会去除尾随空白。
- **STRIP RTRIM**—去除尾随空白。
- **STRIP ON**—不推荐使用。等同于 **STRIP RTRIM**。

在打开 **STRIP** 时（缺省情况），Sybase IQ 首先将从值中去除尾随空白，然后再插入这些值。这仅适用于 **VARCHAR** 数据。**STRIP OFF** 会保留尾随空白。

仅去除不带引号的字符串的尾随空白。带引号的字符串保留其尾随空白。如果不需要空白敏感性，可以使用 **FILLER** 选项作为替代选项，以使要跳过的字节数更加具体化，而不是指出要跳过所有尾随空格。对于 Sybase IQ，**STRIP OFF** 会更有效，在处理尾随空白时，它将遵循 ANSI 标准。（CHAR 数据始终会填充补空，因此 **STRIP** 选项仅影响 **VARCHAR** 数据。）

STRIP 选项仅适用于长度可变的非二进制数据，而不适用于 ASCII 固定宽度插入。例如，假设以下模式：

```
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

```

始终剪裁二进制数据中的尾随空白。

WITH CHECKPOINT—此选项仅在在 Sybase IQ 数据库中装载 SQL Anywhere 表时才会有用。

此子句用于指定是否执行检查点。缺省设置为 **OFF**。如果将此子句设置为 **ON**，则会在成功完成并记录语句之后执行检查点操作。如果服务器在提交连接后及下次检查点之前出现故障，则必须具有用来装载表的数据文件才能成功完成恢复。不过，如果指定了 **WITH CHECKPOINT ON**，并且此后需要恢复，则在恢复时不必存在数据文件。

如果数据库损坏，需要使用备份并应用当前的日志文件，则无论此子句的设置情况如何，都需要数据文件。

警告！ 如果将数据库选项 **CONVERSION_ERROR** 设置为 **OFF**，则可能会将错误的数
据装载到表中而得不到任何错误报告。如果未指定 **WITH CHECKPOINT ON**，并且数据
库需要恢复，则当恢复过程中 **CONVERSION_ERROR** 为 **ON**（缺省值）时，恢复可能
失败。建议您不要在 **CONVERSION_ERROR** 设置为 **OFF** 且未指定 **WITH CHECKPOINT**
ON 的情况下装载表。

另请参见“**CONVERSION_ERROR** 选项 [TSQL]”。

有关自动恢复 Sybase IQ 数据的信息，请参见《系统管理指南第一卷》>“系统恢复
和数据库修复”。

BYTE ORDER — 指定读取时的字节顺序。此选项适用于所有二进制输入字段。如
果未定义，则此选项将会被忽略。Sybase IQ 始终以负责其运行的计算机的 本机格式
读取二进制数据（缺省值为 **NATIVE**）。

- **HIGH**（当多字节数量以高位字节优先时 [对于类似 Sun、IBM AIX 和 HP 之类的大型平台而言]）。
- **LOW**（当多字节数量以低位字节优先时 [对于类似 Windows 之类的小型平台而言]）。

LIMIT—指定要插入表中的最大行数。缺省值为 0，表示无限制。最大值为 $2^{31} - 1$
(2147483647) 行。

NOTIFY—指定每次在表中成功插入指定行数时将通过消息通知您。缺省值为每 100,000 行。此选项的值将覆盖 **NOTIFY_MODULUS** 数据库选项的值。

ON FILE ERROR—指定 Sybase IQ 在因为输入文件不存在或没有读取该文件的正确权限而不能打开此文件时应执行的操作。您可以指定以下操作中的一个：

- **ROLLBACK** 中止整个事务（缺省设置）。
- **FINISH** 结束已完成的插入操作并终止装载操作。
- **CONTINUE** 返回一个错误但仅跳过该文件以继续执行装载操作。

只允许一个 **ON FILE ERROR** 子句：

PREVIEW—显示目标表中输入的布局，包括每列的起始位置、名称和数据类型。Sybase IQ 将在装载过程开始时显示此信息。如果正在写入日志文件，此信息也包括在日志中。

ROW DELIMITED BY—指定最大长度 4 字节的字符串，以指示输入记录的结尾。仅当行中的所有字段为以下情形中任意一种情形时才能使用此选项：

- 由列终结符分隔
- 通过 **DATE** 或 **DATETIME** *column-spec* 选项定义数据
- **ASCII** 固定长度字段

如果任何输入字段包含二进制数据，则不能使用此选项。使用此选项，行终结符将使任何缺失字段设置为 **NULL**。所有行都必须具有相同的行分隔符，并且必须区别于所有列分隔符。行分隔符字符串和字段分隔符字符串彼此不能为初始子集。例如，不能指定 “*” 作为字段分隔符并指定 “*#” 作为行分隔符，但可以与该行分隔符一起指定 “#” 作为字段分隔符。

如果行缺少其分隔符，则 Sybase IQ 返回一个错误并回退整个装载事务。唯一的例外是回退该行并返回一条警告消息的文件的最后一条记录。在 Windows 上，通常由换行符后跟回车符表示行分隔符。对于此选项或 **FILLER**，您可能需要将此指定为 *delimiter-string*（请参见以上说明）。

SKIP—定义此装载要从输入表的开头跳过的行数。要跳过的最大行数为 $2^{31} - 1$ (2147483647)。缺省值为 0。

HEADER SKIP...HEADER DELIMITED BY—指定 **LOAD TABLE** 在数据文件开头要跳过的行数，包括标题行。在跳过指定的行数之前，将忽略所有的 **LOAD TABLE** 列规范及其它装载选项。

- 要跳过的行数大于或等于零。
- 行由 **HEADER DELIMITED BY** 子句中指定的 1 到 4 个字符分隔符字符串确定。缺省的 **HEADER DELIMITED BY** 字符串为 ‘\n’ 字符。
- **HEADER DELIMITED BY** 字符串的最大长度为四个字符。如果字符串长度大于 4 或小于 1，将返回一个错误。
- 如果指定了非零 **HEADER SKIP** 值，则包括 **HEADER DELIMITED BY** 分隔符的所有数据都将被忽略，直到分隔符的出现次数达到 **HEADER SKIP** 子句中指定的次数。

- 在跳过指定的行数之前，将忽略所有的 **LOAD TABLE** 列规范及其它装载选项。在跳过指定的行数后，**LOAD TABLE** 列规范及其它装载选项将应用于剩余的数据。
- 仅在数据开头处忽略“标题”字节。如果在 **USING** 子句中指定了多个文件，**HEADER SKIP** 将只忽略从第一个文件的第一行开始的数据，直到跳过指定的标题行，即使这些行存在于后面的文件中。**LOAD TABLE** 在开始解析实际数据后不会寻找标题。
- 如果 **LOAD TABLE** 在跳过 **HEADER SKIP** 指定的行数之前处理所有输入数据，则不会报告错误。

WORD SKIP—遇到数据长度超过创建单词索引时指定的限制时，允许装载继续进行。

如果因为字超过最大允许大小而未能装载某一行，则会将一条警告写入 `.iqmsg` 文件。还可以选择将 **WORD** 大小违规记录到 **MESSAGE LOG** 文件中，并将被拒绝的行记录到 **LOAD TABLE** 语句中指定的 **ROW LOG** 文件。

- 如果未指定此选项，则 **LOAD TABLE** 将在第一次遇到长度超过指定限制的字时报告错误并回退。
- *number* 指定了忽略“不支持超过最大允许字长的字度”错误的次数。
- 0（零）表示没有限制。

START ROW ID—指定 Sybase IQ 表中应开始插入的行的记录标识号。

不允许对分区表使用 **LOAD TABLE** 和 **INSERT** 命令的 **START ROW ID** 子句。

ON PARTIAL INPUT ROW—指定在装载期间遇到部分输入行时采取的操作。您可以指定以下各选项之一：

- **CONTINUE** 发出一条警告并继续装载操作。这是缺省设置。
- **ROLLBACK** 中止整个装载操作并报告错误。

```
Partial input record skipped at EOF.  SQLSTATE:
QDC32      SQLSTATE:  -1000232L
```

IGNORE CONSTRAINT—指定是否忽略装载期间发生的 **CHECK**、**UNIQUE**、**NULL**、**DATA VALUE** 和 **FOREIGN KEY** 完整性约束违约，以及在启动回退之前忽略的最大违约数。指定每个 *constrainttype* 都具有以下结果：

- **CHECK limit**—如果 *limit* 指定零值，将要忽略的 **CHECK** 约束违约次数为无限次。如果未指定 **CHECK**，则首次发生任何 **CHECK** 约束违约时将导致 **LOAD** 语句回退。如果 *limit* 为非零，则 *limit*+1 次 **CHECK** 约束违约将导致装载回退。
- **UNIQUE limit**—如果 *limit* 指定为零，则忽略的 **UNIQUE** 约束违约次数将为无限次。如果 *limit* 为非零，则 *limit*+1 次 **UNIQUE** 约束违约将导致装载回退。
- **NULL limit**—如果 *limit* 指定为零，则忽略的 **NULL** 约束违约次数将为无限次。如果 *limit* 为非零，则 *limit*+1 次 **NULL** 约束违约将导致装载回退。
- **FOREIGN KEY limit**—如果 *limit* 指定为零，则忽略的 **FOREIGN KEY** 约束违约次数将为无限次。如果 *limit* 为非零，则 *limit*+1 次 **FOREIGN KEY** 约束违约将导致装载回退。
- **DATA VALUE limit**—如果数据库选项 **CONVERSION_ERROR** 为 **ON**，则将报告错误并且该语句将回退。如果 *limit* 指定为零，则忽略的 **DATA VALUE** 约束违约（数

据类型转换错误) 次数将为无限次。如果 *limit* 为非零, 则 *limit*+1 次 DATA VALUE 约束违约将导致装载回退。

- **ALL *limit***—如果数据库选项 `CONVERSION_ERROR = ON`, 则将报告错误并且该语句将回退。如果 *limit* 指定为零, 则忽略的所有完整性约束违约的累积总数将为无限次。如果 *limit* 为非零, 则所有忽略的 **UNIQUE**、**NULL**、**DATA VALUE** 和 **FOREIGN KEY** 完整性约束违约的累计总数超过 *limit* 值时, 装载将回退。例如, 指定以下 **IGNORE CONSTRAINT** 选项:

```
IGNORE CONSTRAINT NULL 50, UNIQUE 100, ALL 200
```

完整性约束违约的总数不能超过 200, 而 **NULL** 和 **UNIQUE** 约束违约的总数分别不能超过 50 和 100。只要超出这些限制中的任何一个, **LOAD TABLE** 语句便将回退。

注意: 单个行可具有多个完整性约束违约。每出现一次完整性约束违约, 便计数一次, 直至该类型违约的限定值。

如果要记录忽略的完整性约束违约, Sybase 强烈建议将 **IGNORE CONSTRAINT** 选项限制设置为非零值。记录过多的违规数会影响装载性能。

如果在 **IGNORE CONSTRAINT** 子句中未指定 **CHECK**、**UNIQUE**、**NULL** 或 **FOREIGN KEY**, 则在第一次发生这些完整性约束违约类型中的一种时, 装载将回退。

如果未在 **IGNORE CONSTRAINT** 子句中指定 **DATA VALUE**, 则当这种类型的完整性约束违约首次发生时, 将会回退装载, 但如果数据库选项 `CONVERSION_ERROR = OFF` 则例外。如果 `CONVERSION_ERROR = OFF`, 则对于任何 **DATA VALUE** 约束违约, 将报告警告并继续装载。

装载完成时, 将在 `.iqmsg` 文件中记录一条与完整性约束违约有关的信息性消息。此消息包含装载时发生的完整性约束违约次数及跳过的行数。

MESSAGE LOG—指定在其中记录完整性约束违约相关信息的文件的名称及要记录的违约类型。时间戳表示记录在 **MESSAGE LOG** 和 **ROW LOG** 文件中的装载开始和完成时间。必须指定 **MESSAGE LOG** 和 **ROW LOG**, 否则将不记录与完整性违约有关的信息。

- 如果未指定 **ONLY LOG** 子句, 则不会记录有关完整性约束违约的任何信息。只记录表示装载开始和完成时间的时间戳。
- 对于 **ONLY LOG** 子句中指定的所有完整性约束类型违约, 或指定关键字 **WORD** 时的所有字索引长度违约, 将记录相关违约信息。
- 如果正在记录约束违约, 则每次发生完整性约束违约时, 将在 **MESSAGE LOG** 文件中正好生成一行信息。

MESSAGE LOG 文件中的行数 (已报告错误) 可能超出 **IGNORE CONSTRAINT** 选项限制, 因为装载是由并行运行的多个线程执行的。多个线程可以报告约束违约次数已超出指定限定值。

- 如果正在记录约束违约, 则对于某一给定行, 将在 **ROW LOG** 文件中正好记录一行信息, 而不考虑在该行发生的完整性约束违约的次数。

MESSAGE LOG 文件中的不同错误数可能与 **ROW LOG** 文件中的行数不完全匹配。行数差是由于 **MESSAGE LOG** 的上述并行装载处理造成的。

- **MESSAGE LOG** 和 **ROW LOG** 文件不能为原始分区或命名管道。
- 如果 **MESSAGE LOG** 或 **ROW LOG** 文件已经存在，则新信息会附加到文件。
- 如果为 **MESSAGE LOG** 或 **ROW LOG** 文件指定了无效文件名，将产生错误。
- 如果为 **MESSAGE LOG** 和 **ROW LOG** 文件指定了相同的文件名，将产生错误。

IGNORE CONSTRAINT 和 **MESSAGE LOG** 选项的各种组合将导致不同的记录操作。

表 11. **LOAD TABLE** 记录操作

已指定 IGNORE CONSTRAINT ?	已指定 MESSAGE LOG ?	操作
yes	yes	回退之前，将记录所有被忽略的完整性约束违约（包括用户指定的限制）。
no	yes	回退之前，将记录第一次完整性约束违约。
yes	no	不记录。
no	no	不记录。第一次完整性约束违约将导致回退。

注意： 如果要记录忽略的完整性约束违约，Sybase 强烈建议将 **IGNORE CONSTRAINT** 选项限制设置为非零值。如果单行出现多次完整性约束违约，则将针对每次违约在 **MESSAGE LOG** 文件中写入一行。记录过多的违规数会影响装载性能。

LOG DELIMITED BY—指定 **ROW LOG** 文件中各数据值之间的分隔符。缺省分隔符为逗号。

有关 **MESSAGE LOG** 和 **ROW LOG** 文件内容和格式的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“使用 **LOAD TABLE** 语句批量装载”。

将 **FORMAT BCP** 指定为 **LOAD TABLE** 子句后，Sybase IQ 不再返回错误消息。此外，会检验以下情况，并返回正确的错误消息：

- 如果指定装载格式不为 **ASCII**、**BINARY** 或 **BCP**，Sybase IQ 现在返回消息“只有 **ASCII**、**BCP** 和 **BINARY** 是受支持的 **LOAD** 格式。”
- 如果 **LOAD TABLE** 列规范 包含除列名、**NULL** 或 **ENCRYPTED** 以外的任何内容，Sybase IQ 返回错误 消息“对于 **LOAD ... FORMAT BCP** 无效的装载规范。”
- 如果 **FORMAT BCP** 装载的列分隔符或行终结符大小 大于 10 个字符，Sybase IQ 返回 消息“分隔符 ‘%2’ 的长度必须为 1 到 %3 个字符”（其中 %3 等于 10）。对于 **FORMAT BCP** 和 **FORMAT ASCII** 出现的错误或警告情况，与这些情况相对应的消息对于两种格式而言都是相同的。

- 如果指定的装载缺省值为 **AUTOINCREMENT**、**IDENTITY** 或 **GLOBAL AUTOINCREMENT**，Sybase IQ 返回错误 “缺省值 %2 不能用作 LOAD 缺省值。 %1”
- 如果 **LOAD TABLE** 规范不包含 需要从指定文件装载的 任何列，Sybase IQ 则报告错误 “LOAD 语句必须至少包含 要从输入文件中装载的 1 列。”，并回退 **LOAD TABLE** 语句。
- 如果装载超出了具有 **TEXT** 索引的文本文档的最大词语数限制，Sybase IQ 返回错误 “文本文档超出 最大词语数。 每个文档最多支持 4294967295 个词语。”

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase—不适用。

权限

执行 **LOAD TABLE** 语句所需的权限取决于数据库服务器的 **-gl** 命令行选项，如下所示：

- 如果 **-gl** 选项设置为 **ALL**，则您必须是表的所有者，拥有 **DBA** 权限或拥有 **ALTER** 权限。
- 如果 **-gl** 选项设置为 **DBA**，则您必须拥有 **DBA** 权限。
- 如果 **-gl** 选项设置为 **NONE**，则不允许执行 **LOAD TABLE**。

有关详细信息，请参见《实用程序指南》> “start_iq 数据库服务器启动实用程序”> “start_iq 服务器选项”中的 **-gl** 命令行选项。

LOAD TABLE 表还需要对表拥有 写入锁。

另请参见

- **INSERT** 语句 （第 211 页）
- **LOAD_ZEROLENGTH_ASNULL** 选项 （第 392 页）
- **NON_ANSI_NULL_VARCHAR** 选项 （第 410 页）

存储大小

字符数据的存储大小、给定的列定义大小和输入数据大小。

表 12. 字符数据的存储大小

数据类型	列定义	输入数据	存储
CHARACTER, CHAR	(32K - 1) 字节的宽度	(32K - 1) 字节	(32K - 1) 字节
VARCHAR, CHARACTER VARY- ING	(32K - 1) 字节的宽度	(32K - 1) 字节	(32K - 1) 字节

LOCK TABLE 语句

禁止其它并发事务在指定时间内访问或修改表。

语法

```
LOCK TABLE table-list [ WITH HOLD ] IN { SHARE | WRITE
      | EXCLUSIVE } MODE [ WAIT time ]
```

参数

- **table-list:** - [*owner.*] *table-name* [, [*owner.*] *table-name*, ...]
time:
string

示例

- **示例 1** - 对 Customers 和 Employees 表获取 WRITE 锁（如果在 5 分 3 秒内可用）：

```
LOCK TABLE Customers, Employees IN WRITE MODE WAIT
'00:05:03'
```

- **示例 2** - 无限期等待，直到对 Customers 和 Employees 表的 WRITE 锁可用或发生中断为止：

```
LOCK TABLE Customers, Employees IN WRITE MODE WAIT
```

用法

table-name—该表必须为基表，而不能为视图。WRITE 模式仅对 IQ 基表有效。LOCK TABLE 要么锁定表列表中的所有表，要么不锁定任何表。当针对 SQL Anywhere 表获取锁，或者当获取 SHARE 或 EXCLUSIVE 锁时，则只能指定一个表。使用标准 Sybase IQ 对象限定规则分析 *table-name*。有关相关信息，请参见《参考：构件块、表和过程》>“SQL 语言元素”>“标识符”和《系统管理指南第一卷》>“数据库对象管理”>“表管理”>“表创建准则”>“表类型”。

WITH HOLD—如果指定该子句，锁一直保持到连接结束时。如果没有指定该子句，则提交或回退当前事务时将释放锁。

SHARE—禁止其它事务修改表，但允许它们读取表。在该模式下，只要没有其它的事务锁定所修改的行，就可以间接地或显式地使用 LOCK TABLE 更改表中的数据。

WRITE—禁止其它事务修改表的列表。无条件提交连接最外面的事务。事务的快照版本不是由 LOCK TABLE IN WRITE MODE 语句建立的，而是通过执行 Sybase IQ 处理的下一命令建立的。

参与连接索引的 IQ 表上的 WRITE 模式锁还将锁定下列各项：

- 当 **X** 为非顶层表时，将连接索引层次的顶层表锁定在 **WRITE** 模式中
- 对应的连接虚拟表 (JVT)

当事务提交或回退时或者连接断开时，才能释放 **WRITE** 模式锁。

EXCLUSIVE—禁止其它事务访问表。在此模式下，任何其它事务都不能对表执行任何种类的查询、更新或任何其它操作。如果用 **LOCK TABLE t IN EXCLUSIVE MODE** 排它性地锁定表 **t**，则服务器的缺省行为是不获取 **t** 的行锁。将 **SUBSUME_ROW_LOCKS** 选项设置为 **OFF** 可以禁用此行为。请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “数据库选项” > “按字母顺序排列的选项列表” > “**subsume_row_locks** 选项”。

注意： 此参考指向 SQL Anywhere 文档。

对协调器上的 **IQ** 主存储区中的表所运行的 **LOCK TABLE** 语句，不会影响从辅助服务器上的连接访问这些表。例如：

在协调器连接上，执行以下命令：

```
LOCK TABLE coord1 WITH HOLD IN EXCLUSIVE MODE
```

协调器上的 **sp_iqlocks** 确认表 **coord1** 具有排它 (E) 锁。

在辅助服务器上的连接上运行的 **sp_iqlocks** 的结果不会显示表 **coord1** 上的排它锁。此连接上的用户可查看对协调器上的表 **coord1** 的更新。

协调器上的其它连接可查看 **coord1** 上的排它锁，并且尝试通过协调器上的其它连接从表 **coord1** 中进行选择将返回“用户 **DBA** 锁定了 **coord1** 中的行。”

WAIT time—**Wait** 选项指定所有锁类型的最大阻塞时间。当锁定模式为 **WRITE** 时，该选项是必需的。给定时间参数后，仅当锁在指定时间内可用时，服务器才会锁定指定表。可以采用 **hh:nn:ss:sss** 格式指定时间参数。如果指定了日期部分，服务器将忽略此部分，并将此参数转换为时间戳。如果未给定时间参数，服务器将无限期地等待，直到 **WRITE** 锁可用或发生中断为止。

LOCK TABLE 不支持在视图上执行。尝试锁定视图将获取共享模式锁，而不管命令中指定的模式如何。共享模式锁阻止其它事务修改表模式。

Transact-SQL (T-SQL) 存储过程语言不支持 **LOCK TABLE**。例如，以下语句返回 **Syntax error near LOCK**:

```
CREATE PROCEDURE tproc()
AS
BEGIN
COMMIT;
LOCK TABLE t1 IN SHARE MODE
INSERT INTO t1 VALUES(30)
END
```

Watcom-SQL 存储过程语言支持 **LOCK TABLE**。缺省命令分隔符为分号 (;)。例如：

```
CREATE PROCEDURE tproc()
AS
```



```
BEGIN
COMMIT;
LOCK TABLE t1 IN SHARE MODE
INSERT INTO t1 VALUES(30)
END
```

另请参见《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqlocks 过程”。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase - Adaptive Server Enterprise 中支持。Adaptive Server Enterprise 中不支持 **WITH HOLD** 子句。Adaptive Server Enterprise 提供在 SQL Anywhere 中不受支持的 **WAIT** 子句。

权限

若要以 **SHARE** 模式锁定表，需要 **SELECT** 权限。

若要以 **EXCLUSIVE** 模式锁定表，必须是表所有者或者有 **DBA** 权限。

另请参见

- **SELECT** 语句（第 277 页）

LOOP 语句

重复执行语句列表。

语法

```
[ statement-label: ]
... [ WHILE search-condition ] LOOP
... statement-list
... END LOOP [ statement-label ]
```

示例

- **示例 1** - 过程中的 **WHILE** 循环：

```
...
SET i = 1 ;
WHILE i <= 10 LOOP
    INSERT INTO Counters( number ) VALUES ( i ) ;
    SET i = i + 1 ;
END LOOP ;
...
```

- **示例 2** - 过程中带标签的循环：

SQL 语句

```
SET i = 1;
lbl:
LOOP
    INSERT
    INTO Counters( number )
    VALUES ( i ) ;
    IF i >= 10 THEN
        LEAVE lbl ;
    END IF ;
    SET i = i + 1 ;
END LOOP lbl
```

用法

WHILE 和 **LOOP** 语句都是控制语句，当 *search-condition* 计算结果为 **TRUE** 时，使用这两个语句可以重复执行 SQL 语句列表。**LEAVE** 语句可在 **END LOOP** 语句后的第一个语句处用于重新开始执行。

如果指定结尾 *statement-label*，它必须与开头 *statement-label* 匹配。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase—在 Adaptive Server Enterprise 中不受支持。**WHILE** 语句在 Transact-SQL 存储过程中提供了循环。

权限

无

另请参见

- FOR 语句（第 191 页）
- LEAVE 语句（第 221 页）
- WHILE 语句 [T-SQL]（第 312 页）

MESSAGE 语句

显示一条消息。

语法

```
MESSAGE expression, ...
[ TYPE { INFO | ACTION | WARNING | STATUS } ]
[ TO { CONSOLE
    | CLIENT [ FOR { CONNECTION conn_id [ IMMEDIATE ] | ALL } ]
    | [ EVENT | SYSTEM ] LOG }
[ DEBUG ONLY ] ]
```

参数

- **conn_id:** - *integer*

示例

- **示例 1** - 在数据库服务器消息窗口中显示字符串 The current date and time, 以及当前日期和时间:

```
CREATE PROCEDURE message_test ()
BEGIN
MESSAGE 'The current date and time: ', Now();
END;
CALL message_test();
```

- **示例 2** - 若要在 ODBC 中注册回调, 首先要声明消息处理程序:

```
void SQL_CALLBACK my_msgproc(
    void *      sqlca,
    unsigned char    msg_type,
    long          code,
    unsigned short   len,
    char*          msg )
{ ... }
```

通过调用 `SQLSetConnectAttr` 函数安装已声明的消息处理程序:

```
rc = SQLSetConnectAttr(
    dbc,
    ASA_REGISTER_MESSAGE_CALLBACK,
    (SQLPOINTER) &my_msgproc, SQL_IS_POINTER );
```

用法

MESSAGE 语句显示消息, 该消息可以是任何表达式。子句可指定显示消息的位置。

发布 **MESSAGE ... TO CLIENT** 语句的过程必须与连接相关联。

例如, 以下不显示消息框, 因为事件发生在连接之外:

```
CREATE EVENT CheckIdleTime TYPE ServerIdle
WHERE event_condition( 'IdleTime' ) > 100
HANDLER
BEGIN
MESSAGE 'Idle engine' type warning to client;
END;
```

但是, 在下面的示例中, 消息被写入到服务器主控台:

```
CREATE EVENT CheckIdleTime TYPE ServerIdle
WHERE event_condition( 'IdleTime' ) > 100
HANDLER
BEGIN
MESSAGE 'Idle engine' type warning to console;
END;
```

有效的表达式可以包含用引号引起来的字符串或其它常量、变量或函数。但是，在 **MESSAGE** 语句的输出中不允许有查询，即使表达式定义包括查询，也是如此。

FOR 子句可用于通知其它应用程序在服务器上检测到的事件，而不需要该应用程序显式地检查事件。使用 **FOR** 子句后，接收者在下次执行 **SQL** 语句时会接收到该消息。如果接收者当前正在执行 **SQL** 语句，则将在该语句完成后接收到消息。如果正在执行的语句是存储过程调用，则在调用完成前便会接收到消息。

如果应用程序需要在消息发出不久和连接并未执行 **SQL** 语句时得到通知，可以使用第二个连接。此连接可执行一条或多条 **WAITFOR DELAY** 语句。这些语句不消耗服务器或网络上的大量资源（而轮询方法则会消耗大量资源），但允许应用程序在消息发出后不久接收到通知。

ESQL 和 ODBC 客户端通过消息回调函数接收消息。在每种情况下，都必须注册这些函数。若要注册 ESQL 消息处理程序，请使用 **db_register_callback** 函数。

ODBC 客户端可以用 **SQLSetConnectAttr** 函数注册回调函数。

有关使用回调函数的详细信息，请参见“SQL Anywhere Server - 编程”>“数据库工具接口 (DBTools)”>“使用数据库工具接口”>“使用回调函数”。

注意： 此参考指向 SQL Anywhere 文档。

TYPE—只有将消息发送到客户端时 **TYPE** 子句才有效。客户端应用程序必须决定如何处理消息。**Interactive SQL** 在以下位置显示消息：

- **INFO** - “消息”窗口（缺省值）。
- **ACTION**- 含“确定”按钮的消息框。
- **WARNING** - 含“确定”按钮的消息框。
- **STATUS** - “消息”窗格。

TO—指定消息的目标：

- **CONSOLE** - 将消息发送到数据库服务器窗口。**CONSOLE** 是缺省设置。
- **CLIENT** - 将消息发送到客户端应用程序。您的应用程序必须决定如何处理消息，您可以根据 **TYPE** 信息作出决定。
- **LOG** - 将消息发送到 -o 选项指定的服务器日志文件中。

FOR—对于消息 **TO CLIENT**，此子句指定哪个连接接收关于消息的通知：

- **CONNECTION** *conn_id* - 指定消息接收者的连接 ID。
- **IMMEDIATE** - 请参见“SQL Anywhere Server - SQL 参考”>“SQL 语句”>“**MESSAGE** 语句”。
- **ALL** - 指定所有打开的连接接收消息。

DEBUG ONLY—用于控制是否通过更改 **DEBUG_MESSAGES** 选项的 设置启用或禁用添加到存储过程中的调试消息。如果指定了 **DEBUG ONLY**，则仅在 **DEBUG_MESSAGES** 选项设置为 **ON** 的情况下才执行 **MESSAGE** 语句。

注意： **DEBUG ONLY** 将 **DEBUG_MESSAGES** 选项设置为 **Off** 时，消息并不占用大量资源，因此在生产系统中，通常可以将这些语句保留在存储过程中。不过，在可能会频繁执行这些语句的地方应慎用它们；否则，它们可能会导致性能略微降低。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase — Adaptive Server Enterprise 中不支持。Transact-SQL **PRINT** 语句提供类似的功能，在 SQL Anywhere 中可用。

权限

必须连接到数据库。

执行包含 **FOR** 子句的 **MESSAGE** 语句需要拥有 **DBA** 权限。

另请参见

- **CREATE PROCEDURE** 语句（第 112 页）
- **WAITFOR** 语句（第 309 页）
- **DEBUG_MESSAGES** 选项（第 358 页）

OPEN 语句 [ESQL] [SP]

用于打开以前声明的游标以访问数据库中的信息。

语法

```
OPEN cursor-name
... [ USING [ DESCRIPTOR { sqlda-name | host-variable [, ...] } ] ]
... [ WITH HOLD ]
```

参数

- **cursor-name:** - 标识符或宿主变量
- **sqlda-name:** - identifier

示例

- **示例 1** - 在嵌入式 SQL 中使用 **OPEN**:

```
EXEC SQL OPEN employee_cursor;
```

和

```
EXEC SQL PREPARE emp_stat FROM 'SELECT EmployeeID, Surname FROM  
Employees WHERE name like ?'; EXEC SQL DECLARE employee_cursor  
CURSOR FOR emp_stat; EXEC SQL OPEN employee_cursor USING :pattern;
```

- **示例 2** – 来自过程的示例:

```
BEGIN  
DECLARE cur_employee CURSOR FOR  
    SELECT Surname  
    FROM Employees ;  
DECLARE name CHAR(40) ;  
OPEN cur_employee;  
LOOP  
    FETCH NEXT cur_employee into name ;  
    ...  
END LOOP  
CLOSE cur_employee;  
END
```

用法

缺省情况下，在当前事务（**COMMIT** 或 **ROLLBACK**）结束时会自动关闭所有游标。可选的 **WITH HOLD** 子句使游标对后面的事务保持打开。游标一直保持打开，直到当前连接结束或执行了显式 **CLOSE** 语句。终止连接时自动关闭游标。

游标位于第一行之前。请参见《系统管理指南第二卷》>“使用过程和批处理”。

当打开游标时，由游标声明的 **FOR READ ONLY** 查看声明该游标的表的版本，而不是在首次执行 **FETCH** 时查看表的版本。

USING DESCRIPTOR *sqllda-name*、**host-variable** 和 **BLOCK n** 格式仅适用于嵌入式 SQL。

如果游标名称是由标识符或字符串指定的，则相应的 **DECLARE CURSOR** 语句在 C 程序中必须出现在 **OPEN** 之前；如果游标名称是由宿主变量指定的，则 **DECLARE CURSOR** 语句必须在 **OPEN** 语句之前执行。

可选 **USING** 子句指定绑定到已为其声明游标的 **SELECT** 语句中占位符绑定变量的宿主变量。

成功执行 **OPEN** 语句之后，**SQLCA (SQLIOESTIMATE)** 的 *sqlerrd[3]* 字段中将填入读取查询的所有行所需的输入/输出操作次数的一个估计值。此外，**SQLCA (SQLCOUNT)** 的 *sqlerrd[2]* 字段将填入游标中的实际行数（大于或等于 0 的值）或其估计值（绝对值为估计值的负数）。如果数据库服务器不统计行数即可计算出该值，则 *sqlerrd[2]* 字段就是实际行数。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase - Adaptive Server Enterprise 支持简单的 **OPEN***cursor-name* 语法。Adaptive Server Enterprise 存储过程不支持任何其它子句。Open Client/Open Server 支持 **USING** 描述符 或主机名变量语法。

权限

- 必须有 **SELECT** 语句中所有表的 **SELECT** 权限，或者有 **CALL** 语句中的过程的 **EXECUTE** 权限。
- 当游标位于 **CALL** 语句上时，**OPEN** 导致过程一直执行，直到遇到第一个结果集（不含 **INTO** 子句的 **SELECT** 语句）为止。如果该过程结束时未找到任何结果集，则会设置 **SQLSTATE_PROCEDURE_COMPLETE** 警告。

另请参见

- **CLOSE** 语句 [ESQL] [SP]（第 59 页）
- **DECLARE CURSOR** 语句 [ESQL] [SP]（第 155 页）
- **FETCH** 语句 [ESQL] [SP]（第 188 页）
- **PREPARE** 语句 [ESQL]（第 251 页）
- **RESUME** 语句（第 267 页）

OUTPUT 语句 [Interactive SQL]

将当前查询结果写入文件。

语法

```
OUTPUT TO filename
[ APPEND ] [ VERBOSE ]
[ FORMAT output-format ]
[ ESCAPE CHARACTER character ]
[ DELIMITED BY string ]
[ QUOTE string [ ALL ] ]
[ COLUMN WIDTHS ( integer, ... ) ]
[ HEXADECIMAL { ON | OFF | ASIS } ]
[ ENCODING encoding ]
[ WITH COLUMN NAMES ]
```

参数

- **output-format**: - **TEXT** | **FIXED** | **HTML** | **SQL** | **XML**
- **encoding**: - *string* 或 *identifier*

示例

- **示例 1** - 将 **Employees** 表的内容 放在文本文件中：

```
SELECT * FROM Employees;
OUTPUT TO employees.txt FORMAT TEXT
```

- **示例 2** - 将 **Employees** 表的内容放置在现有文件的末尾，同时还在此文件中包含有关查询的所有消息：

```
SELECT * FROM Employees;
OUTPUT TO employees.txt APPEND VERBOSE
```

- **示例 3** – 导出包含嵌入式换行符的值。换行符的数值为 10，在 SQL 语句中可以用字符串 '\x0a' 代替该值。

执行具有 **HEXADECIMAL ON** 的语句：

```
SELECT 'line1\x0aline2'; OUTPUT TO file.txt HEXADECIMAL ON
```

将会得到一个文件，其中的一行包含下列文字：

```
line10x0aline2
```

执行具有 **HEXADECIMAL OFF** 的相同语句：

```
line1\x0aline2
```

如果将 **HEXADECIMAL** 设置为 **ASIS**，您会得到含有两行的文件：

```
'line1
line2'
```

使用 **ASIS** 时生成两行，这是因为已导出的嵌入式换行符没有转换成两位数的十六进制形式，也没有前缀。

用法

OUTPUT 语句将当前查询所检索的信息复制到文件中。

可以使用可选的 **FORMAT** 子句指定输出格式。如果未指定 **FORMAT** 子句，则使用 **Interactive SQL OUTPUT_FORMAT** 选项设置。

当前查询为 **SELECT** 或 **LOAD TABLE** 语句，生成了显示在“结果”窗格的“结果”选项卡上的信息。如果当前查询不存在，则 **OUTPUT** 语句将报告错误。

注意： **OUTPUT** 在向另一个应用程序提供查询的结果或报告时尤为有用，但对于批量操作不建议使用该语句。对于大容量数据移动，请使用 **SELECT** 语句的 **ASCII** 和 **BINARY** 数据提取功能。提取功能提供更好的大规模数据移动性能，并创建可用于负载的输出文件。

APPEND—此可选关键字用于将查询结果附加到现有输出文件的末尾，但不覆盖文件以前的内容。如果不使用 **APPEND** 子句，缺省情况下 **OUTPUT** 语句将覆盖输出文件的内容。当输出格式为 **TEXT**、**FIXED** 或 **SQL** 时，**APPEND** 关键字有效。

VERBOSE—当含有可选的 **VERBOSE** 关键字时，有关查询的错误消息、用于选择数据的 SQL 语句和数据本身写入输出文件。如果省略 **VERBOSE**（缺省设置），则只将数据写入文件。当输出格式为 **TEXT**、**FIXED** 或 **SQL** 时，**VERBOSE** 关键字有效。

FORMAT - 允许的输出格式包括：

- **TEXT**—输出 **TEXT** 格式的文件，且文件中每行代表一数据行。所有值都用逗号分隔，字符串用撇号（单引号）引起来。可以使用 **DELIMITED BY** 和 **QUOTE** 子句更改分隔符和引号字符串。如果在 **QUOTE** 子句中指定 **ALL**，则所有的值（不仅是字符串）都将用引号引起来。**TEXT** 是缺省的输出格式。

还可以使用其它三种特殊序列。两个字符 `\n` 代表 换行符；`\\` 代表单个 `\`；而序列 `\xDD` 代表具有十六进制代码 `DD` 的 字符。

如果要导出具有字符串返回值的 Java 方法，则必须使用 **HEXADECIMAL OFF** 子句。

- **FIXED** - 输出结果是每列都有固定宽度的固定格式。可以使用 **COLUMN WIDTHS** 子句为每一列指定宽度。此格式不输出任何列标题。

如果省略 **COLUMN WIDTHS**，则每列的宽度均由该列的数据类型计算得出，并且大小足以存放该数据类型的任意值。例外是 **LONG VARCHAR** 和 **LONG BINARY** 数据缺省为 32KB。

- **HTML** - 输出结果是超文本标记语言格式。
- **SQL** - 输出为重新创建表中信息所需的 **Interactive SQL INPUT** 语句。

注意： Sybase IQ 不支持 **INPUT** 语句。您需要将此语句编辑为有效的 **LOAD TABLE** (或 **INSERT**) 语句以使用它将数据装载回文件。

- **XML** - 输出结果是以 UTF-8 编码且包含嵌入式 **DTD** 的 XML 文件。二进制值以 **CDATA** 块进行编码，二进制数据显示为两位数十六进制字符串。**LOAD TABLE** 语句不接受将 XML 作为文件格式。

ESCAPE CHARACTER - 以十六进制代码和符号存储的字符的缺省转义字符是反斜杠 (`\`)，例如 `\x0A` 是换行符。

使用 **ESCAPE CHARACTER** 子句可以对此缺省值进行更改。例如，要将感叹号用作转义字符，请输入：

```
... ESCAPE CHARACTER '!'
```

DELIMITED BY—**DELIMITED BY** 子句仅用于 **TEXT** 输出格式。分隔符字符串放置在列之间（缺省为逗号）。

QUOTE—**QUOTE** 子句仅用于 **TEXT** 输出格式。引号字符串放置在字符串值的两边。缺省值为单引号字符。如果在 **QUOTE** 子句中指定 **ALL**，则引号字符串将被置于所有值的两边，而不仅仅是字符串的两边。

COLUMN WIDTHS - **COLUMN WIDTHS** 子句用于为 **FIXED** 格式的输出指定列宽。

HEXADECIMAL—**HEXADECIMAL** 子句 指定如何仅为 **TEXT** 格式卸载二进制 数据。设置为 **ON** 时，以 `0xabcd` 格式卸载二进制数据。设置为 **OFF** 时，二进制数据在卸载时转义 (`\xab\xcd`)。设置为 **ASIS**，值按原样写入，即不发生任何转义，即使值中包含控制字符也是如此。**ASIS** 对包含格式设置字符（如制表符或回车）的 文本非常有用。

ENCODING—指定用于写入文件的编码。**ENCODING** 子句 只能用于 **TEXT** 格式。

如果未指定 *encoding*，**Interactive SQL** 按如下方式确定用于写文件的代码页，列表中位置靠前的代码页值优先于位置靠后的值：

- 使用 **DEFAULT_ISQL_ENCODING** 选项指定的代码页（如果设置此选项）
- 运行 **Interactive SQL** 的计算机的缺省代码页

副作用

- 在 Interactive SQL 中，“结果”选项卡仅显示当前查询的结果。所有先前的查询结果都替换为当前查询结果。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不适用。

权限

无

另请参见

- SELECT 语句（第 277 页）
- DEFAULT_ISQL_ENCODING 选项 [Interactive SQL]（第 362 页）

PARAMETERS 语句 [Interactive SQL]

指定 Interactive SQL (**dbisql**) 命令文件的参数。

语法

```
PARAMETERS parameter1, parameter2, ...
```

示例

- 示例 1** - 以下 **dbisql** 命令文件具有两个参数：

```
PARAMETERS department_id, file ;  
SELECT Surname  
FROM Employees  
WHERE DepartmentID = {department_id}  
>#{file}.dat;
```

用法

PARAMETERS 指定命令文件中有多少个参数，并命名这些参数以便将来可在命令文件中引用它们。

通过将参数放到命令文件中您要替代命名参数的位置来引用参数：

```
{parameter1}
```

括号和参数名之间不能有空格。

如果用少于所需数目的参数调用命令文件，**dbisql** 将提示您提供缺少参数的值。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不适用。

权限

无

另请参见

- READ 语句 [Interactive SQL] (第 258 页)

PREPARE 语句 [ESQL]

准备要稍后执行的语句或用于游标的语句。

语法

```
PREPARE statement-name
FROM statement [ FOR { READ ONLY | UPDATE [ OF column-name-list ] } ]
... [ DESCRIBE describe-type INTO [ [ SQL ] DESCRIPTOR ] descriptor ]
... [ WITH EXECUTE ]
```

参数

- **statement-name:** - 标识符或宿主变量
- **statement:** - 字符串或宿主变量
- **describe-type:** - { ALL | BIND VARIABLES | INPUT | OUTPUT | SELECT LIST } ...
{ LONG NAMES [[OWNER.]TABLE.]COLUMN] | WITH VARIABLE RESULT }

示例

- 示例 1 - 准备简单查询:

```
EXEC SQL PREPARE employee_statement FROM
'SELECT Surname FROM Employees';
```

用法

PREPARE 语句准备 *statement* 中指定的 SQL 语句，并使准备好的语句与 *statement-name* 关联。

在执行语句或者打开游标（如果是 **SELECT** 语句的话）时，引用上述的语句名称。

statement-name 可以是自动包含的 `sqlca.h` 头文件中定义的

`a_sql_statement_number` 类型的宿主变量。如果用标识符替换 *statement-name*，则每个模块只有一个语句可以使用该 *statement-name* 进行。

如果用宿主变量替换 *statement-name*，它的类型必须是 `short int`。`sqlca.h` 中含有此类型的类型定义，名为 `a_sql_statement_number`。SQL 预处理器识别此类型，因此可在 **DECLARE** 节中使用。在执行 **PREPARE** 语句期间，宿主变量由数据库填充，不必由程序员进行初始化。

FOR UPDATE | FOR READ ONLY 定义游标可更新性，如果游标使用该语句。**FOR READ ONLY** 游标不能用于 **UPDATE**（定位）或 **DELETE**（定位）操作中。缺省值为 **FOR READ ONLY**。为响应指定 **FOR UPDATE** 的游标的任何请求，Sybase IQ 提供了对值敏感的游标或敏感性未定型的游标。不敏感的游标或者敏感性未定型游标不能更新。

如果使用了 **DESCRIBE INTO DESCRIPTOR**，则会在指定的描述符中对准备好的语句进行说明。说明类型可能是 **DESCRIBE** 语句所允许的任何说明类型。

如果使用了 **WITH EXECUTE** 子句，则当且仅当此语句不是 **CALL** 或 **SELECT** 语句且不含任何宿主变量时才会执行。成功执行此语句后，立即将其删除。如果 **PREPARE** 和 **DESCRIBE**（如果有）成功，但此语句无法执行，则会设置警告“SQLCODE 111, SQLSTATE 01W08”，但并不删除此语句。

DESCRIBE INTO DESCRIPTOR 和 **WITH EXECUTE** 子句可能会提高性能，因为它们减少所需的客户端/服务器通信。

WITH VARIABLE RESULT 子句用于描述可以有一个以上结果集（具有不同列数或列类型）的过程。

如果使用 **WITH VARIABLE RESULT**，则数据库服务器将 **DESCRIBE** 子句之后的 **SQLCOUNT** 值设置为下列值之一：

- 0—结果集可能更改：应该在每个 **OPEN** 语句后重新描述过程调用。
- 1—结果集是固定的。不需要重新描述。

可以准备以下语句：

- **ALTER**
- **CALL**
- **COMMENT ON**
- **CREATE**
- **DELETE**
- **DROP**
- **GRANT**
- **INSERT**
- **REVOKE**
- **SELECT**
- **SET OPTION**

支持准备 **COMMIT**、**PREPARE TO COMMIT** 和 **ROLLBACK** 语句是为了兼容性。但是，Sybase 建议用静态嵌入式 SQL 执行所有的事务管理操作，因为某些应用程序环境可能要求使用它。另外，其它嵌入式 SQL 系统不支持动态事务管理操作。

注意： 请确保在使用后用 **DROP** 删除相应语句。否则，不释放与此语句关联的内存。

副作用

- 此前使用同一名称准备的所有语句都会丢失。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase—受 Open Client/Open Server 支持。

权限

无

另请参见

- DECLARE CURSOR 语句 [ESQL] [SP] (第 155 页)
- DESCRIBE 语句 [ESQL] (第 166 页)
- DROP 语句 (第 170 页)
- EXECUTE 语句 [ESQL] (第 182 页)
- OPEN 语句 [ESQL] [SP] (第 245 页)

PRINT 语句 [T-SQL]

在数据库服务器的消息窗口中显示一则消息。

语法

```
PRINT format-string [, arg-list]
```

示例

- **示例 1** - 在服务器消息窗口中显示一则消息：

```
CREATE PROCEDURE print_test
AS
PRINT 'Procedure called successfully'
```

下面的语句向客户端返回字符串“过程已成功调用”：

```
EXECUTE print_test
```

- **示例 2** - 使用 **PRINT** 语句中的占位符；执行过程内的以下语句：

```
DECLARE @var1 INT, @var2 INT
SELECT @var1 = 3, @var2 = 5
PRINT 'Variable 1 = %1!, Variable 2 = %2!', @var1, @var2
```

- **示例 3** - 使用 **RAISERROR** 禁止连接：

```
CREATE procedure DBA.login_check()  
begin  
    // Allow a maximum of 3 concurrent connections  
    IF( db_property('ConnCount') > 3 ) then  
        raiserror 28000  
        'User %1! is not allowed to connect -- there are  
        already %2! users logged on',  
        current user,  
        cast(db_property('ConnCount') as int)-1;  
    ELSE  
        call sp_login_environment;  
    end if;  
end  
go  
grant execute on DBA.login_check to PUBLIC  
go  
set option PUBLIC.Login_procedure='DBA.login_check'  
go
```

有关禁止连接的其它方法，请参见“**LOGIN_PROCEDURE 选项**”或《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqmodifylogin 过程”。

用法

如果从 Open Client 应用程序或 JDBC 应用程序连接，则 **PRINT** 语句将消息返回到客户端窗口。如果从嵌入式 SQL 或 ODBC 应用程序连接，则消息显示在数据库服务器窗口上。

对于可选参数列表中的参数，格式字符串可以包含占位符。这些占位符的形式为 **%nn!**，其中 **nn** 是介于 1 和 20 之间的整数。

标准

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

权限

必须连接到数据库。

另请参见

- MESSAGE 语句（第 242 页）
- LOGIN_PROCEDURE 选项（第 395 页）

PUT 语句 [ESQL]

在指定的游标中插入行。

语法

```

        PUT
        cursor-name [ USING
        DESCRIPTOR
        sqlda-name
| FROM
        hostvar-list ] [ INTO { DESCRIPTOR
        into-sqlda-name
| into-hostvar-list } ] [ ARRAY :nnn ]

```

参数

- **cursor-name:** - *identifier* 或 *hostvar*
- **sqlda-name:** - *identifier*
- **hostvar-list:** - 可以包含指示符变量

示例

- **示例 1** - 在嵌入式 SQL 中使用 **PUT**:

```
EXEC SQL PUT cur_employee FROM :EmployeeID, :Surname;
```

用法

在指定的游标中插入行。列的值从第一个 **SQLDA** 或宿主变量列表中获取，与 **INSERT** 语句中的列（对于 **INSERT** 游标）或选择列表中的列（对于 **SELECT** 游标）是一一对应的关系。

PUT 语句只能在满足以下条件的 **INSERT** 或 **SELECT** 语句的游标上使用：此语句的 **FROM** 子句引用单个表，或引用由单个基表组成的可更新视图。

如果 **SQLDA** 中的 **sqldata** 指针为空指针，则表示没有为该列指定任何值。如果有 **DEFAULT VALUE** 与该列关联，则使用缺省值；否则使用空值。

第二个 **SQLDA** 或宿主变量列表包含 **PUT** 语句的结果。

可选的 **ARRAY** 子句可用于执行大范围放入，即一次插入多行，这样可以提高性能。值 *nnn* 是要插入的行数。**SQLDA** 必须包含 *nnn** (*columns per row*) 变量。第一行放在 **SQLDA** 变量 0 和 (*columns per row*) - 1 之间，依此类推。

注意：对于滚动（对值敏感）游标，如果新行与 **WHERE** 子句匹配并且键集游标未完成填充，则显示插入的行。对于动态游标，如果插入的行与 **WHERE** 子句匹配，则可能显示该行。不敏感游标不能更新。

有关将 LONG VARCHAR 或 LONG BINARY 值放进数据库的信息，请参见“SET 语句 [ESQL]”。

副作用

- 当用对值敏感的游标（即键设置驱动的游标）插入行时，插入的行出现在结果集的末尾处，即使它们和查询的 **WHERE** 子句不匹配，或者当 **ORDER BY** 子句将它们正常地放置在结果集的其它位置时也是如此。有关详细信息，请参见“SQL Anywhere Server - 编程” > “在应用程序中使用 SQL” > “SQL Anywhere 游标” > “对值敏感的游标”。

注意：此参考指向 SQL Anywhere 文档。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase—受 Open Client/Open Server 支持。

权限

必须有 INSERT 权限。

另请参见

- DELETE（定位）语句 [ESQL] [SP]（第 165 页）
- INSERT 语句（第 211 页）
- SET 语句 [ESQL]（第 285 页）
- UPDATE 语句（第 304 页）
- UPDATE（定位）语句 [ESQL] [SP]（第 308 页）

RAISERROR 语句 [T-SQL]

发出错误信号并向客户端发送消息。

语法

```
RAISERROR error-number [ format-string ] [, arg-list ]
```

示例

- **示例 1** – 引发错误 99999（该错误在用户定义的错误范围内）并向客户端发送消息：


```
RAISERROR 99999 'Invalid entry for this
column: %1!', @val
```

用法

RAISERROR 语句允许发出用户定义的错误并向客户端发送消息。

error-number 是大于 17000 的五位整数。错误号存储在全局变量 @@error 中。

error-number 和 *format-string* 参数之间没有逗号。逗号后的第一项解析为参数列表中的第一项。

如果未提供 *format-string* 或者其为空，则使用错误号在系统表中查找 错误消息。

Adaptive Server Enterprise 从 SYSMESSAGES 表获取消息 17000-19999。在 Sybase IQ 中，此表为空视图，因此这个范围内的错误消息应该提供 格式字符串。错误号大于或等于 20000 的消息从 SYS.SYSUSERMESSAGES 表中获取。

format-string 的长度最大为 255 字节。这与 Adaptive Server Enterprise 中相同。

SQL Server 或 Adaptive Server Enterprise**RAISERROR** 语句支持的扩展值在 Sybase IQ 中不受支持。

对于可选参数列表中的参数，格式字符串可以包含占位符。这些占位符的形式为 %nn!，其中 nn 是介于 1 和 20 之间的整数。

中间 **RAISERROR** 状态和代码信息在过程终止后会丢失。如果在返回时伴随 **RAISERROR** 发生了错误，则返回错误信息，而 **RAISERROR** 信息将丢失。应用程序可以通过在不同的执行点检查 @@error 全局变量来查询中间 **RAISERROR** 状态。

标准

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

权限

必须连接到数据库。

另请参见

- CONTINUE_AFTER_RAISERROR 选项 [TSQL] (第 345 页)
- ON_TSQL_ERROR 选项 [TSQL] (第 412 页)

READ 语句 [Interactive SQL]

从文件中读取 Interactive SQL (**dbisql**) 语句。

语法

```
READ filename [ parameter ] ...
```

示例

- 示例 1 -

```
READ status.rpt '160'
```

- 示例 2 -

```
READ birthday.sql [>= '1988-1-1'] [<= '1988-1-30']
```

- 示例 3 -

```
[test1.sql]
PARAMETERS par1, par2;

BEGIN
DECLARE v_par1 int;
DECLARE v_par2 varchar(200);

SET v_par1 = {par1};
SET v_par2 = {par2};

MESSAGE STRING('PAR1 Value: ', v_par1 ) TO CLIENT;
MESSAGE STRING('PAR2 Value: ', v_par2 ) TO CLIENT;

END;

(USR1)> READ test1.sql 123 '041028'
PAR1 Value: 123
PAR2 Value: 041028
```

注意： 第二个参数值 041028 必须用引号引起来，因为 *v_par2* 声明为字符数据类型。

用法

READ 语句从指定的文件中读取 **dbisql** 语句序列。此文件可以包含任何有效的 **dbisql** 语句，包括可以嵌套至任何深度的其它 **READ** 语句。

当查找命令文件时，**dbisql** 首先 搜索当前目录，再搜索环境变量 **SQLPATH** 中 指定的目录，然后搜索环境变量 **PATH** 中 指定的目录。如果命名文件没有文件扩展名，**dbisql** 还会搜索 每个目录以查找具有扩展名 **.sql** 的相同文件名。

可以在命令文件的名称后面列出参数。这些参数对应于语句文件开头处 **PARAMETERS** 语句上指定的参数（请参见“**PARAMETERS 语句**”）。**dbisql** 随后会在源文件中包含以下内容的位置替换相应参数：

```
{ parameter-name }
```

其中 *parameter-name* 是相应参数的名称。

传递给命令文件的参数可以是标识符、数字、带引号的标识符或字符串。如果用引号将参数括起来，则在替换时引号也放到文本中。不是标识符、数字或字符串（包含空格或制表符）的参数必须用中括号 (**[]**) 括起来。这允许在命令文件中执行任意文本替换。

作为参数传递到 **READ** 语句的 **SQL** 字符文字（包括字符数据）应用引号引起来。

如果没有足够的参数传递到命令文件，**dbisql** 会提示您提供缺少参数的值。

READ 语句也支持 **ENCODING** 子句，可让您指定用于读取文件的编码。请参见“**SQL Anywhere Server - SQL 参考**” > “**SQL 语句**” > “**SQL 语句**” > “**READ 语句 [Interactive SQL]**”。

注意： 此参考指向 **SQL Anywhere** 文档。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不适用。

权限

无

另请参见

- **DEFAULT_ISQL_ENCODING** 选项 [Interactive SQL]（第 362 页）
- **PARAMETERS 语句** [Interactive SQL]（第 250 页）

RELEASE SAVEPOINT 语句

释放当前事务内的保存点。

语法

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

用法

savepoint-name 是在当前事务内的 **SAVEPOINT** 语句上指定的标识符。如果省略 *savepoint-name*，则释放最近的保存点。

有关保存点的说明，请参见《系统管理指南第二卷》>“使用过程和批处理”。释放保存点不执行任何类型的 **COMMIT**，它只是从当前活动保存点的列表中删除相应保存点。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase—不受 Adaptive Server Enterprise 支持。通过使用嵌套事务，可以采用 Adaptive Server Enterprise 兼容的方式实现类似的功能。

权限

当前事务中必须有相应的 **SAVEPOINT**。

另请参见

- ROLLBACK TO SAVEPOINT 语句（第 273 页）
- SAVEPOINT 语句（第 275 页）

REMOVE 语句

从数据库删除类、软件包或 JAR 文件。删除的类不再可供用作变量类型。

语法

```
REMOVE JAVA classes_to_remove
```

参数

- **classes_to_remove:** - { **CLASS***java_class_name* [, *java_class_name*] ... | **PACKAGE***java_package_name* [, *java_package_name*] ... | **JAR***jar_name* [, *jar_name*] ... [**RETAIN CLASSES**] }
- **jar_name:** - *character_string_expression*

示例

- **示例 1** - 从当前数据库中删除名为“Demo”的 Java 类：

```
REMOVE JAVA CLASS Demo
```

用法

任何要删除的类、软件包或 JAR 必须已安装。

java_class_name—要删除的一个或多个 Java 类的名称。这些类必须是当前数据库中已安装的类。

java_package_name—要删除的一个或多个 **Java** 软件包的名称。这些软件包的名称必须为当前数据库中的软件包的名称。

jar_name—最大长度为 255 的字符串值。

每个 **jar_name** 必须与当前数据库中保留的 **JAR** 的 **jar_name** 相等。**jar_name** 的等同性由 **SQL** 系统的字符串比较规则确定。

如果指定了 **JAR...RETAIN CLASSES**，则指定的 **JAR** 将不再保留在数据库中，并且保留的类没有与之关联的 **JAR**。如果指定了 **RETAIN CLASSES**，则这是 **REMOVE** 语句的唯一操作。

标准

- **SQL**—ISO/ANSI **SQL** 语法的供应商扩展。
- **Sybase**—不受 **Adaptive Server Enterprise** 支持。通过使用嵌套事务，可以采用 **Adaptive Server Enterprise** 兼容的方式实现类似的功能。

权限

必须拥有 **DBA** 权限或必须拥有相应对象。

RESIGNAL 语句

重新发出异常情况的信号。

语法

```
RESIGNAL [ exception-name ]
```

示例

- **示例 1** – 下面这段代码向应用程序返回除“未找到列”外的所有异常：

```
...
DECLARE COLUMN_NOT_FOUND EXCEPTION
FOR SQLSTATE '52003';
...
EXCEPTION
WHEN COLUMN_NOT_FOUND THEN
SET message='Column not found' ;
WHEN OTHERS THEN
RESIGNAL ;
```

用法

在异常处理程序中，使用 **RESIGNAL** 可在异常仍处于活动状态时退出复合语句，或停止报告另一个指定的异常。异常将由其它异常处理程序处理或返回给应用程序。异常处理程序在 **RESIGNAL** 之前的任何操作都被撤消。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase—在 Adaptive Server Enterprise 中不受支持。Transact-SQL 过程中的错误处理是用 **RAISERROR** 语句完成的。

权限

无

另请参见

- BEGIN……END 语句（第 49 页）
- SIGNAL 语句（第 295 页）

RESTORE 语句

从一个或多个存档 设备恢复 Sybase IQ 数据库备份。

语法

语法 1

```
RESTORE DATABASE 'db_file'  
FROM 'archive_device' [ FROM 'archive_device' ]…  
… [ CATALOG ONLY ]  
… [ KEY key_spec ]  
… [ [ RENAME logical-dbfile-name TO 'new-dbspaced-path']…  
    | VERIFY [ COMPATIBLE ] ]
```

语法 2

```
                RESTORE DATABASE 'database-name'  
[ restore-option ... ]  
FROM 'archive_device' ...
```

参数

- **db_file:** – 要恢复的数据库的相对路径或绝对路径。可以是初始位置，也可以是目录存储文件的新位置。
- **key_spec:** – 用引号括起来的包含大小写字符、数字、字母和特殊字符的字符串。防止该键受到命令解释器的解释或更改可能非常有必要。
- **restore-option:** –

```
READONLY dbspace-or-file [, … ]  
KEY key_spec  
RENAME file-name TO new-file-path ...
```

示例

- **示例 1** – 此 UNIX 示例从 Sun Solaris 平台上的磁带设备 `/dev/rmt/0` 和 `/dev/rmt/2` 中恢复 `iqdemo` 数据库。在 Solaris 上，设备名后面的字母 *n* 指定“关闭时不回绕”。若要随 **RESTORE** 指定此功能，请使用适用于您的 UNIX 平台的命名约定。（Windows 不支持此功能。）

```
RESTORE DATABASE 'iqdemo'
FROM '/dev/rmt/0n'
FROM '/dev/rmt/2n'
```

- **示例 2** – 恢复名为 `marvin` 的加密数据库，该数据库用密钥 `is!seCret` 进行了加密：

```
RESTORE DATABASE 'marvin'
FROM 'marvin_bkup_file1'
FROM 'marvin_bkup_file2'
FROM 'marvin_bkup_file3'
KEY 'is!seCret'
```

- **示例 3** – 以下示例显示了 **BACKUP** 语句和两种可能的 **RESTORE** 语句的语法。（此示例使用 `iqdemo` 数据库中的对象仅是为了进行说明。请注意，`iqdemo` 包括一个名为 `iq_main` 的用户数据库空间示例，您的数据库中可能不包括该示例。）

给定以下 **BACKUP** 语句：

```
BACKUP DATABASE READONLY DBSPACES iq_main
TO '/system1/IQ15/demo/backup/iqmain'
```

可以使用下面任意一种 **RESTORE** 语句来恢复数据库空间 `iq_main`：

```
RESTORE DATABASE 'iqdemo' READONLY DBSPACES iq_main
FROM '/system1/IQ15/demo/backup/iqmain'
```

或

```
RESTORE DATABASE 'iqdemo'
FROM '/system1/IQ15/demo/backup/iqmain'
```

选择性备份会备份所有 **READWRITE** 数据库空间或特定只读数据库空间或 `dbfile`。选择性备份是完整备份或增量备份的子类型。

注意：

- 可以执行 **READONLY** 选择性备份，并从该备份中恢复所有对象（如以上第二个示例中所示）。
- 也可以执行包括全部内容的备份，并选择性地恢复只读文件和数据库空间。
- 可以执行多个只读文件和数据库空间的 **READONLY** 选择性备份，并选择性地恢复只读文件和数据库空间的子集。请参见“权限”。
- 只有只读文件自备份以来尚未发生更改的情况下，才可以恢复只读备份。一旦再次将数据库空间设置为读写模式，只读备份将无效，除非将数据库的整个读写部分恢复为只读数据库空间原为只读模式的点上。

- Sybase 建议您坚持选择性或非选择性备份子类型中的一种。如果必须要从非选择性备份转换为选择性备份（反之亦然），请始终在转换为新的子类型之前执行一次非选择性完整备份以确保涵盖了所有变更。
- **示例 4** – 在不执行任何写入操作的情况下使用 **VERIFY** 子句来验证数据库存档的语法：

```
RESTORE DATABASE <database_name.db>
FROM ' /sys1/dump/dmp1 '
FROM ' /sys1/dump/dmp2 '
VERIFY
```

在使用验证时，请指定不同的数据库名以避免出现“数据库名不唯一”错误。例如，如果原始数据库为 `iqdemo.db`，请改用 `iq_demo_new.db`：

```
RESTORE DATABASE iqdemo_new.db FROM iqdemo.bkp VERIFY
```

用法

RESTORE 命令要求 DBA 对数据库独占访问。通过对 DBA 设置 **-gd** 开关可以获得此独占访问权，这是启动服务器引擎时的缺省设置。

在启动数据库之前发出 **RESTORE** 命令（必须连接到 `utility_db` 数据库）。完成对相应类型的备份指定 **RESTORE** 命令之后，该数据库即可使用。数据库将处于上次恢复备份后第一个隐式 **CHECKPOINT** 结束时所存在的状态。现在，可以指定 **START DATABASE** 以允许其他用户访问恢复后的数据库。

包括所有子句的完整 **RESTORE** 命令的最大大小是 32 KB。

当恢复至原始设备时，请确保设备的容量足以容纳您要恢复的数据库空间。如果裸设备大小不足以恢复数据库空间，**IQ RESTORE** 会检查裸设备的大小并返回错误。请参见《系统管理指南第一卷》>“数据备份、恢复和存档”>“恢复数据库”>“**RESTORE** 语句”>“移动数据库文件”>“恢复至原始设备”。

BACKUP 允许指定完整备份或增量备份。有两种增量备份。**INCREMENTAL** 仅备份那些自上次执行任意类型（增量或完全）的备份之后已发生更改并已提交的块。

INCREMENTAL SINCE FULL 备份自上次完全备份后进行了更改的所有块。如果在用 **RESTORE** 执行完全备份恢复后执行一次或多次增量备份（任一类型），则在连续的 **RESTORE** 命令之间不允许对数据库进行任何更改。此规则防止从需要进行崩溃恢复的数据库或已进行更改的数据库的增量备份执行 **RESTORE**。您仍可以使用 **RESTORE** 从完全备份恢复以覆盖这样的数据库。

在开始完全恢复之前，您必须删除以下两个文件：目录存储文件（缺省名为 `dbname.db`）和事务日志文件（缺省名为 `dbname.log`）。

如果对增量备份进行恢复，则 **RESTORE** 可以确保以恰当的顺序访问备份介质集。此顺序为：首先恢复最后的完全备份磁带集，接着是第一个增量备份磁带集，然后是更近一些的磁带集，以此类推，最后是最近的增量备份磁带集。如果 DBA 生成了

INCREMENTAL SINCE FULL 备份，则只需要完全备份磁带集和最近的 **INCREMENTAL SINCE FULL** 备份磁带集；不过，如果自执行 **INCREMENTAL SINCE FULL** 备份后又执行了 **INCREMENTAL** 备份，则还必须应用此备份。

Sybase IQ 确保恢复顺序是正确的，否则它会显示错误。在恢复期间发生任何其它错误均会导致将数据库标记为损坏并将使数据库不可用。若要清除损坏的数据库，请从完全备份执行 **RESTORE**，然后再恢复任一附加增量备份。由于很可能是这些备份之一损坏，因此您可能需要忽略较晚的备份集而使用较早的备份集。

要从存档备份恢复只读文件或数据库空间，当发出 **RESTORE** 语句时，数据库可以正在运行，管理员可以连接到数据库。如果只读文件路径名与数据库系统表信息匹配，则无需与备份中的名称相匹配。

数据库必须处于未运行状态，才能对 **READWRITE FILES ONLY** 或所有文件备份执行 **FULL**、**INCREMENTAL SINCE FULL** 或 **INCREMENTAL** 恢复。要恢复只读文件的备份，数据库可以正在运行，也可以未在运行。当恢复只读数据库空间中的特定文件时，该数据库空间必须处于脱机状态。当恢复读写数据库空间中的只读文件时，该数据库空间可处于联机或脱机状态。恢复将关闭只读文件，恢复文件，并在恢复结束时重新打开这些文件。

可使用选择性恢复来恢复只读数据库空间，条件是该数据库空间仍处于相同的只读状态。

FROM - 指定要用于恢复的 *archive_device* 的名称，并用单引号分隔。如果要使用多个存档设备，请使用单独的 **FROM** 子句指定它们。不允许使用逗号分隔的列表。存档设备必须是不同的。**FROM** 子句的数量决定了与输出设备有关的并行 **Sybase IQ** 尝试的数量。

使用备份/还原 API DLL 实现可以指定要在打开存档设备时传递给 DLL 的参数。对于第三方实现，*archive_device* 字符串具有以下格式：

```
'DLLIdentifier::vendor_specific_information'
```

以下是一个特定示例：

```
'spsc::workorder=12;volname=ASD002'
```

archive_device 字符串的长度最长为 1023 个字节。*DLLIdentifier* 部分的长度必须为 1 到 30 个字节且只能包含字母数字和下划线字符。字符串的 *vendor_specific_information* 部分将传递给第三方实现，而不检查其内容。

注意： 只有某些第三方产品才可用 **Sybase IQ** 通过 使用此语法进行认证。有关其它用法说明或限制，请参见“发行公告”。在使用任意第三方产品 备份 **Sybase IQ** 数据库之前，请确保该产品已获得认证。请参见“发行公告”或“技术文档”中 **Sybase IQ** 产品的 **Sybase** 认证 报告。

对于备份/恢复 API 的 **Sybase** 实现，无需指定磁带设备名或文件名以外的信息。不过，如果您使用的是磁盘设备，则必须在 **RESTORE** 上指定与执行备份时给定的数量相同的存档设备；否则，您用于执行恢复的设备数量会与用于执行备份的设备数量不同。指定 UNIX 系统的非回绕磁带设备的 **Sybase API DLL** 存档设备的特定示例如下：

```
'/dev/rmt/0n'
```

CATALOG ONLY—仅从存档介质恢复备份标头记录。

RENAME—将一个或多个 Sybase IQ 数据库文件 恢复到一个新位置。指定每个您要移动的 *dbspace-name* (按其 在 **SYSDATABASE** 表中的显示方式)。将 *new-dbspace-path* 指定为该数据库空间的新原始分区，或者新的完整路径名或相对路径名。

如果创建数据库文件时使用的是相对路径，则缺省情况下这些文件将恢复至相对于目录存储文件 (**SYSTEM** 数据库空间) 的位置，并且不需要使用重命名子句。如果创建数据库文件时使用的是绝对路径且未为文件指定重命名子句，则将其恢复至其初始位置。

RENAME 子句中的相对路径名的工作方式与创建数据库或数据库空间时相同：主 IQ 存储数据库空间、临时存储数据库空间和消息日志将恢复至 *db_file* (目录存储) 位置的相对位置；用户创建的 IQ 存储数据库空间将恢复至包含主 IQ 数据库空间的目录的相对目录。

请勿使用 **RENAME** 子句移动保存目录存储的 **SYSTEM** 数据库空间。若要移动目录存储及任何相对于它创建且未在 **RENAME** 子句中指定的文件，请在 *db_file* 参数中指定新位置。

VERIFY [COMPATIBLE]— 指令 服务器检验指定的 Sybase IQ 数据库备份存档 以进行完整、增量、基于完整的增量或虚拟 备份。备份必须是 Sybase IQ 12.6 版或更高版本。检验过程 会检查指定的存档有无恢复过程检查的错误，但不执行写入操作。所有状态信息和检测到的 错误都会被写入服务器日志文件中。

不能将 **RENAME** 子句与 **VERIFY** 子句一起使用；将报告错误。

备份检验过程可以在数据库所在主机之外的主机上运行。您必须具有 DBA、BACKUP 或 OPERATOR 权限才可运行 **RESTORE VERIFY**。

如果在使用 **VERIFY** 时指定了 **COMPATIBLE** 子句，则会对现有的数据库文件进行增量存档的兼容性检查。如果数据库文件不存在于调用 **RESTORE...VERIFY COMPATIBLE** 的系统上，就会返回错误。如果在检验完整备份时指定了 **COMPATIBLE**，此关键字就会被忽略；在恢复完整备份时不需要进行兼容性检查。

必须具有数据库和日志文件 (.db 和 .log)，才能验证完整备份中只读数据库空间的备份。如果没有这些文件，就请在不使用 **READONLY dbspace** 子句的情况下运行 **RESTORE...VERIFY** 来验证整个备份。

请参见《系统管理指南第一卷》>“数据备份、恢复和存档”>“恢复数据库”>“RESTORE 语句”>“在恢复后验证数据库”。

注意： 备份存档的检验不同于数据库一致性检查程序 (DBCC) 验证模式 (**sp_iqcheckdb 'verify...'**)。 **RESTORE VERIFY** 验证备份存档的一致性，以确保它可以恢复，而 DBCC 验证数据库数据的一致性。

在开始备份之前先运行 **sp_iqcheckdb 'verify...'**。如果备份了不一致的数据库，然后从同一备份存档进行恢复，则数据会继续处于不一致状态，即使 **RESTORE VERIFY** 报告检验成功也是如此。

其它 **RESTORE** 问题:

- **RESTORE** 恢复至磁盘时，不支持将原始设备用作存档设备。
- Sybase IQ 在使用磁带之前不会回绕磁带；当回绕磁带设备时，它只是在使用磁带之后回绕磁带。必须在启动 **RESTORE** 之前将每个磁带放在 Sybase IQ 数据的开始处。
- 在执行 **BACKUP** 和 **RESTORE** 操作过程中，如果 Sybase IQ 无法打开存档设备（例如，当它需要装载介质时）并且 **ATTENDED** 选项为 **ON**，则 Sybase IQ 会等待十秒钟，以让您将下一磁带放入驱动器 然后重试。它将无限期地继续这些尝试，直至成功或使用 **Ctrl+C** 终止操作。
- 如果按 **Ctrl+C**，**RESTORE** 将失败并将数据库返回至其在恢复开始前的状态。
- 如果使用磁盘条带化，则经过分条的磁盘将被视为单个设备。
- 在恢复过程中，**SYSTEM** 数据库空间的 **SYSFILE** 系统表中的 **file_name** 列不会更新。对于 **SYSTEM** 数据库空间，创建数据库时，**file_name** 列始终显示其名称。**SYSTEM** 数据库空间的文件名是数据库文件的名称。

另请参见《系统管理指南第一卷》>“数据备份、恢复和存档”。

标准

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

权限

必须具有 **DBA** 权限。当 **-gu** 服务器启动选项设置为值 **DBA**（缺省值）时，具有 **SPACE ADMIN** 权限的用户可以执行只读选择性恢复。

另请参见

- **BACKUP** 语句（第 44 页）

RESUME 语句

在查询后重新开始过程。

语法

语法 1

```
RESUME cursor-name
```

语法 2

```
RESUME [ ALL ]
```

参数

- **cursor-name:** - identifier
- **cursor-name:** - 标识符或宿主变量

示例

- **示例 1 - 嵌入式 SQL 示例:**

```
EXEC SQL RESUME cur_employee;
```

和

```
EXEC SQL RESUME :cursor_var;
```

- **示例 2 - dbisql 例如:**

```
CALL sample_proc() ;  
RESUME ALL;
```

用法

RESUME 语句重新执行返回结果集的过程。

该过程一直执行，直到遇见下一个结果集（不带 **INTO** 子句的 **SELECT** 语句）为止。如果该过程结束时未找到任何结果集，则会设置 **SQLSTATE_PROCEDURE_COMPLETE** 警告。使用 **RESUME** 重新开始 **SELECT** 语句的游标时，也会设置此警告。

注意: **dbisqlc** 中支持语法 1 **RESUME** 语句，但此语句在 **dbisql** (Interactive SQL) 中无效或当使用 **SQL Anywhere JDBC** 驱动程序连接至数据库时无效。

dbisqlRESUME 语句（语法 2）重新开始当前过程。如果未指定 **ALL**，则执行 **RESUME** 将显示下一结果集，或者，如果不再返回任何结果集，则结束过程。

dbisqlRESUME ALL 语句将遍历过程中的所有结果集而不显示这些结果集，然后结束过程。这主要在测试过程时有用。

标准

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

权限

游标在此前必须已打开。

另请参见

- **DECLARE CURSOR** 语句 [ESQL] [SP]（第 155 页）

RETURN 语句

无条件地退出函数或过程，并根据需要提供返回值。**RETURN** 之后的语句不再执行。

语法

```
RETURN [ ( expression ) ]
```

示例

- **示例 1** - 返回三个数字的乘积：

```
CREATE FUNCTION product ( a numeric,
                           b numeric ,
                           c numeric)
RETURNS numeric
BEGIN
    RETURN ( a * b * c ) ;
END
```

- **示例 2** - 计算三个数字的乘积：

```
SELECT product (2, 3, 4)
```

```
product (2,3,4)
24
```

- **示例 3** - 避免执行复杂查询（如果这样做没有意义的话）：

```
CREATE PROCEDURE customer_products
( in customer_id integer DEFAULT NULL)
RESULT ( id integer, quantity_ordered integer )
BEGIN
    IF customer_id NOT IN (SELECT ID FROM Customers)
    OR customer_id IS NULL THEN
        RETURN
    ELSE
        SELECT ID,sum(
            SalesOrderItems.Quantity )
        FROM Products,
            SalesOrderItems,
            SalesOrders
        WHERE SalesOrders.CustomerID = customer_id
        AND SalesOrders.ID = SalesOrderItems.ID
        AND SalesOrderItems.ProductID = Products.D
        GROUP BY Products.ID
    END IF
END
```

用法

如果提供了 *expression*，则 *expression* 的值作为函数或过程的值返回。

在函数中，表达式的数据类型应当与函数的 **RETURN** 的数据类型相同。

RETURN 在过程中使用是为了与 Transact-SQL 兼容，并且它用于返回整数错误代码。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase—Transact-SQL 过程使用返回语句返回整数错误代码。

权限

无

另请参见

- BEGIN……END 语句（第 49 页）
- CREATE PROCEDURE 语句（第 112 页）

REVOKE 语句

删除指定用户的权限。

语法

语法 1

```
REVOKE
{
  BACKUP
  CONNECT
  DBA
  GROUP
  INTEGRATED LOGIN
  KERBEROS LOGIN
  MEMBERSHIP IN GROUP userid [, ...]
  MULTIPLEX ADMIN
  OPERATOR
  PERMS ADMIN
  PROFILE
  RESOURCE
  SPACE ADMIN
  USER ADMIN }
... FROM userid [, ...]
```

语法 2

```
REVOKE
{ ...ALL [ PRIVILEGES ] | ALTER | DELETE | INSERT
  | REFERENCE | SELECT [ ( column-name [, ...] ) ] | UPDATE [ ( column-
name, ...) ] }
... ON [ owner.]table-name FROM userid [, ...]
```

语法 3

```
REVOKE EXECUTE ON [ owner.]procedure-name FROM userid [, ...]
```

语法 4

```
REVOKE CREATE ON dbspace-name FROM userid [, ...]
```

示例

- 示例 1 - 阻止用户 dave 插入到 Employees 表中:

```
REVOKE INSERT ON Employees FROM dave
```

- 示例 2 - 撤消用户 Jim 的资源权限:

```
REVOKE RESOURCE FROM Jim
```

- 示例 3 - 阻止用户 dave 更新 Employees 表:

```
REVOKE UPDATE ON Employees FROM dave
```

- 示例 4 - 撤消用户配置文件名 Administrator 的集成登录映射:

```
REVOKE INTEGRATED LOGIN FROM Administrator
```

- 示例 5 - 禁止 Finance 组执行 **sp_customer_list** 过程:

```
REVOKE EXECUTE ON sp_customer_list  
FROM finance
```

- 示例 6 - 从数据库中删除用户 ID franw:

```
REVOKE CONNECT FROM franw
```

- 示例 7 - 撤消用户 Smith 对数据库空间 DspHist 的 CREATE 特权:

```
REVOKE CREATE ON DspHist FROM Smith
```

- 示例 8 - 从数据库中撤消用户 ID fionat 对数据库空间 *DspHist* 的 CREATE 权限:

```
REVOKE CREATE ON DspHist FROM fionat
```

用法

REVOKE 语句用于删除使用 **GRANT** 语句赋予的权限。

语法 1 用于撤消特殊的用户权限，语法 2 用于撤消表权限。语法 3 用于撤消执行过程的权限。**REVOKE CONNECT** 用于从数据库删除用户 ID。

注意： 使用系统过程（而不是 **GRANT** 和 **REVOKE**）来添加和删除用户 ID。

REVOKE GROUP 自动撤消组内所有成员的成员资格。

REVOKE CREATE 删除指定用户 ID 对特定数据库空间的 CREATE 权限。

不能针对组中的特定用户撤消权限。如果不希望某个特定用户访问特定的表、视图或过程，则不要使用该用户成为对该对象拥有权限的组的成员。

注意： 如果用户拥有数据库对象（例如表），则不能撤消该用户的连接权。使用 **REVOKE** 语句或 **sp_dropuser** 过程尝试执行此操作会返回错误，如“不能删除在运行时系统中拥有表的用户。”

SQL 语句

副作用

- 自动提交

标准

- SQL—语法 1 是 ISO/ANSI SQL 语法的供应商扩展。语法 2 是初级功能。语法 3 是持久存储模块功能。
- Sybase—Adaptive Server Enterprise 支持语法 2 和语法 3。Adaptive Server Enterprise 不支持语法 1。Sybase IQ 和 Adaptive Server Enterprise 的用户管理和安全模型不同。

权限

必须是要撤消的权限的授予者，或者是有 DBA 权限的用户。

对于语法 1，REVOKE CONNECT、REVOKE INTEGRATED LOGIN 和 REVOKE KERBEROS LOGIN 需要 DBA 或 USER ADMIN 权限。REVOKE GROUP、REVOKE (权限，DBA 除外) 和 REVOKE MEMBERSHIP IN GROUP 需要 DBA 或 PERMS ADMIN 权限。只有 DBA 可以撤消 DBA 权限。

如果撤消其他用户的 **CONNECT** 权限或表权限，则该用户不能连接到数据库。

对于语法 2，REVOKE、REVOKE ALTER、REVOKE DELETE、REVOKE INSERT、REVOKE REFERENCE、REVOKE SELECT 和 REVOKE UPDATE 需要 DBA 或 PERMS ADMIN 权限。

对于语法 3，您必须具有 DBA 或 PERMS ADMIN 权限。

对于语法 4，您必须具有 DBA 或 SPACE ADMIN 权限。

另请参见

- GRANT 语句 (第 202 页)

ROLLBACK 语句

撤消自上次 **COMMIT** 或 **ROLLBACK** 以来所做的任何更改。

语法

```
ROLLBACK [ WORK ]
```

用法

ROLLBACK 结束一个逻辑工作单元 (事务)，并撤消该事务期间内对数据库进行的所有更改。事务是指在一次数据库连接中 **COMMIT** 或 **ROLLBACK** 语句所做的数据库工作。

副作用

- 关闭所有不以 **WITH HOLD** 方式打开的游标。
- 释放由发出 **ROLLBACK** 的事务所持有的锁。

标准

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

权限

必须连接到数据库。

另请参见

- COMMIT 语句（第 63 页）
- ROLLBACK TO SAVEPOINT 语句（第 273 页）

ROLLBACK TO SAVEPOINT 语句

取消自 **SAVEPOINT** 以来进行的所有更改。

语法

```
ROLLBACK TO SAVEPOINT [ savepoint-name ]
```

用法

ROLLBACK TO SAVEPOINT 语句撤消自 **SAVEPOINT** 建立以来所做的所有更改。

不撤消在 **SAVEPOINT** 之前所做的更改；它们仍然处于待执行状态。有关保存点的说明，请参见《系统管理指南第二卷》>“使用过程和批处理”。

savepoint-name 是在当前事务内的 **SAVEPOINT** 语句上指定的标识符。如果省略 *savepoint-name*，则释放最近的保存点。指定保存点之后的所有保存点都自动释放。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase - Adaptive Server Enterprise 不支持保存点。若要用 Adaptive Server Enterprise 兼容的方式实现类似的功能，可使用嵌套事务。

权限

当前事务中必须有相应的 **SAVEPOINT**。

另请参见

- **RELEASE SAVEPOINT** 语句 (第 259 页)
- **ROLLBACK** 语句 (第 272 页)
- **SAVEPOINT** 语句 (第 275 页)

ROLLBACK TRANSACTION 语句 [T-SQL]

取消自 **SAVE TRANSACTION** 以来进行的所有更改。

语法

```
ROLLBACK TRANSACTION [ savepoint-name ]
```

示例

- **示例 1** - 返回五行，值分别为 10、20 等。**ROLLBACK TRANSACTION** 语句撤消 **DELETE**（而不是先前的 **INSERT** 或 **UPDATE**）的结果。

```
BEGIN

    SELECT row_num INTO #tmp
    FROM sa_rowgenerator( 1, 5 )
    UPDATE #tmp SET row_num=row_num*10
    SAVE TRANSACTION before_delete
    DELETE FROM #tmp WHERE row_num >= 3
    ROLLBACK TRANSACTION before_delete
    SELECT * FROM #tmp
END
```

用法

ROLLBACK TRANSACTION 撤消自使用 **SAVE TRANSACTION** 建立保存点以来所做的所有更改。不撤消在 **SAVE TRANSACTION** 之前所做的更改；它们仍然处于待执行状态。

savepoint-name 是在当前事务内的 **SAVE TRANSACTION** 语句上指定的标识符。如果省略 *savepoint-name*，所有未完成的更改都会被回退。指定保存点之后的所有保存点都自动释放。

请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “**ROLLBACK TRANSACTION** 语句 [T-SQL]”。

注意： 此参考指向 SQL Anywhere 文档。

标准

- ISO/ANSI SQL 语法的供应商扩展。

权限

当前事务中必须有相应的 **SAVE TRANSACTION**。

另请参见

- **BEGIN TRANSACTION** 语句 [T-SQL] (第 53 页)
- **SAVE TRANSACTION** 语句 [T-SQL] (第 276 页)

SAVEPOINT 语句

在当前事务中建立保存点。

语法

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

用法

savepoint-name 是可用于 **RELEASE SAVEPOINT** 或 **ROLLBACK TO SAVEPOINT** 语句的标识符。

事务结束后，所有保存点都自动释放。请参见《系统管理指南第二卷》>“使用过程和批处理”。

执行触发器或原子复合语句时建立的保存点在原子操作结束时自动释放。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase—在 Adaptive Server Enterprise 中不受支持。若要用 Adaptive Server Enterprise 兼容的方式实现类似的功能，可使用嵌套事务。

权限

无

另请参见

- **RELEASE SAVEPOINT** 语句 (第 259 页)
- **ROLLBACK TO SAVEPOINT** 语句 (第 273 页)

SAVE TRANSACTION 语句 [T-SQL]

在当前事务中建立保存点。

语法

```
SAVE TRANSACTION [ savepoint-name ]
```

示例

- **示例 1** – 返回五行，值分别为 10、20 等。**ROLLBACK TRANSACTION** 语句撤消 DELETE（而不是先前的 INSERT 或 UPDATE）的结果。

```
BEGIN
    SELECT row_num INTO #tmp
    FROM sa_rowgenerator( 1, 5 )
    UPDATE #tmp SET row_num=row_num*10
    SAVE TRANSACTION before_delete
    DELETE FROM #tmp WHERE row_num >= 3
    ROLLBACK TRANSACTION before_delete
    SELECT * FROM #tmp
END
```

请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “SAVE TRANSACTION 语句 [T-SQL]”。

注意： 此参考指向 SQL Anywhere 文档。

用法

在当前事务中建立保存点。*savepoint-name* 是可在 **ROLLBACK TRANSACTION** 语句中使用的标识符。事务结束后，所有保存点都自动释放。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。

权限

无

另请参见

- BEGIN TRANSACTION 语句 [T-SQL]（第 53 页）
- ROLLBACK TRANSACTION 语句 [T-SQL]（第 274 页）

SELECT 语句

从数据库检索信息。

语法

```
SELECT [ ALL | DISTINCT ] [ FIRST | TOP number-of-rows ] select-list
... [ INTO { host-variable-list | variable-list | table-name } ]
... [ INTO LOCAL TEMPORARY TABLE { table-name } ]
... [ FROM table-list ]
... [ WHERE search-condition ]
... [ GROUP BY [ expression [, ...]
                | ROLLUP ( expression [, ...] )
                | CUBE ( expression [, ...] ) ] ]
... [ HAVING search-condition ]
... [ ORDER BY { expression | integer } [ ASC | DESC ] [, ...] ]
... [ row-limitation-option ]
```

参数

- **select-list:** -

```
{ column-name
  | expression [ [ AS ] alias-name ]
  | * }
```

- **row-limitation-option:** -

```
LIMIT { [ offset-expression, ] limit-expression
        | limit-expression OFFSET offset-expression }
```

- **limit-expression:** - *simple-expression*
- **offset-expression:** - *simple-expression*
- **simple-expression:** -

```
integer
| variable
| ( simple-expression )
| ( simple-expression { + | - | * } simple-expression )
```

示例

- **示例 1** - 列出系统目录中的所有表和视图:

```
SELECT tname
FROM SYS.SYSCATALOG
WHERE tname LIKE 'SYS%';
```

- **示例 2** - 列出所有客户及其订单总值:

```
SELECT CompanyName,
       CAST( sum(SalesOrderItems.Quantity *
                 Products.UnitPrice) AS INTEGER) VALUE
FROM Customers
```

```

LEFT OUTER JOIN SalesOrders
LEFT OUTER JOIN SalesOrderItems
LEFT OUTER JOIN Products
GROUP BY CompanyName
ORDER BY VALUE DESC

```

- **示例 3** - 列出雇员人数:

```

SELECT count(*)
FROM Employees;

```

- **示例 4** - 嵌入式 SQL SELECT 语句:

```

SELECT count(*) INTO :size FROM Employees;

```

- **示例 5** - 按年份、模型和颜色列出总销售额:

```

SELECT year, model, color, sum(sales)
FROM sales_tab
GROUP BY ROLLUP (year, model, color);

```

- **示例 6** - 选择所有具有一定折扣的项目, 放入临时表:

```

SELECT * INTO #TableTemp FROM lineitem
WHERE l_discount < 0.5

```

- **示例 7** - 返回在按姓氏对雇员进行排序时首先出现的雇员的信息:

```

SELECT FIRST *
FROM Employees
ORDER BY Surname;

```

- **示例 8** - 返回在按姓氏对员工姓名排序时出现的前五名雇员:

```

SELECT TOP 5 *
FROM Employees
ORDER BY Surname;

```

```

SELECT *
FROM Employees
ORDER BY Surname
LIMIT 5;

```

- **示例 9** - 列出按姓氏以降序进行排序时出现的第五个和第六个雇员:

```

SELECT *
FROM Employees
ORDER BY Surname DESC
LIMIT 4,2;

```

- **示例 10** - 返回在按姓氏对员工姓名排序时出现的前五名雇员:

```

CREATE OR REPLACE VARIABLE atop INT = 10;

```

```

SELECT TOP (atop -5) *
FROM Employees
ORDER BY Surname;

```

```

SELECT *
FROM Employees
ORDER BY Surname
LIMIT (atop-5);

```

- **示例 11** - 列出按姓氏以降序进行排序时出现的第五个和第六个雇员:

```

CREATE OR REPLACE VARIABLE atop INT = 10;

```

```
SELECT *
FROM Employees
ORDER BY Surname DESC
LIMIT (atop - 8) OFFSET (atop -2 -3 -1);
```

用法

可在 Interactive SQ 中使用 **SELECT** 语句 浏览数据库中的数据，或者将数据从数据库导出到外部文件。

也可以在过程或嵌入式 SQL 中使用 **SELECT** 语句。带 **INTO** 子句的 **SELECT** 语句用于在 **SELECT** 语句只返回一行时从数据库中检索结果。（用 **SELECT INTO** 创建的表不继承 **IDENTITY/AUTOINCREMENT** 表。）对于多行查询，必须使用游标。如果选择多列且没有使用 *#table*，则 **SELECT INTO** 会创建一个永久基表。无论列数多少，**SELECT INTO #table** 始终会创建临时表。**SELECT INTO** 单列表会选择到宿主变量中。

注意： 在编写对 **SELECT INTO** 临时表执行的脚本和存储过程时，在 **CAST** 表达式中对不是基列的选择列表项进行封装。这样可保证临时表的列数据类型是所需的数据类型。

如果表的名称相同但所有者不同，则需要提供别名。没有别名的查询将返回错误结果：

```
SELECT * FROM user1.t1
WHERE NOT EXISTS
(SELECT *
FROM user2.t1
WHERE user2.t1.col1 = user1.t.col1);
```

要返回正确结果，请为每个表使用一个别名：

```
SELECT * FROM user1.t1 U1
WHERE NOT EXISTS
(SELECT *
FROM user2.t1 U2
WHERE U2.col1 = U1.col1);
```

具有 *variable-list* 的 **INTO** 子句仅用在过程中。

在 **SELECT** 语句中， 存储过程调用可出现在基表或视图 允许的任意位置。请注意，需要考虑 CIS 功能补偿性能注意事项。例如，**SELECT** 语句也可以 返回来自过程的结果集。有关语法和示例，请参见“SQL Anywhere Server – SQL 参考”>“SQL 语句”>“FROM 子句”。

注意： 此参考指向 SQL Anywhere 文档。

有关在存储过程内影响从临时表进行选择限制， 请参见《系统管理指南第二卷》>“使用过程和批处理”>“过程简介”>“在结果集内返回过程结果”。

ALL 或 **DISTINCT**—如果二者都未指定，则将检索满足 **SELECT** 语句的子句的所有行。如果指定 **DISTINCT**，则会消除重复的输出行。这叫做语句结果的投影。在许多

情况下，当指定 **DISTINCT** 时，很多语句的执行时间会显著延长，因此请仅在必要时才使用 **DISTINCT**。

如果使用 **DISTINCT**，则该语句不能包含使用 **DISTINCT** 参数的集合函数。

FIRST 或 **TOP number-of-rows**—指定从查询返回的行数。**FIRST** 返回从查询选择的第一行。**TOP** 从查询返回指定的行数，其中 *number-of-rows* 的范围为 1 - 2147483647，可以是整数常量或整数变量。

FIRST 和 **TOP** 主要与 **ORDER BY** 子句一起使用。如果没有与 **ORDER BY** 子句一起使用这些关键字，则同一查询每次运行的结果可能会不同，因为优化程序可能会选择不同的查询计划。

FIRST 和 **TOP** 只允许在查询的顶级 **SELECT** 中使用，因此它们不能用于派生表或视图定义。在视图定义中使用 **FIRST** 或 **TOP** 可能会导致在视图上运行查询时忽略该关键字。

使用 **FIRST** 等同于将 **ROW_COUNT** 数据库选项设置为 1。使用 **TOP** 等同于将 **ROW_COUNT** 选项设置为相同的行数。如果 **TOP** 和 **ROW_COUNT** 都进行了设置，则 **TOP** 的值优先。

在涉及全局变量、系统函数或代理表的查询中使用 **TOP** 时，**ROW_COUNT** 选项可能会产生不一致的结果。有关详细信息，请参见“**ROW_COUNT** 选项”。

select-list—*select-list* 是一个由逗号分隔的表达式列表，用于指定从数据库检索的内容。如果指定一个星号 (*)，将选中 **FROM** 子句 (*table-name* 指定表的所有列) 中的所有表的所有列。*select-list* 中允许使用集合函数和解析函数。请参见《参考：构件块、表和过程》>“SQL 函数”。

注意：在 Sybase IQ 中，顶级 **SELECT** 的选择列表中允许标量子查询（嵌套的 **select**），这与在 SQL Anywhere 和 Adaptive Server Enterprise 中一样。子查询不能用在条件值表达式中（例如 **CASE** 语句中）。

子查询还可以在 **WHERE** 或 **HAVING** 子句谓词（支持的谓词类型之一）中使用。但在 **WHERE** 或 **HAVING** 子句中，子查询不能用在值表达式中，也不能用在 **CONTAINS** 或 **LIKE** 谓词中。外部链接的 **ON** 子句或 **GROUP BY** 子句中不允许子查询。

有关子查询的用法的详细信息，请参见《参考：构件块、表和过程》>“SQL 语言元素”>“表达式”>“表达式中的子查询”以及《参考：构件块、表和过程》>“SQL 语言元素”>“搜索条件”>“搜索条件中的子查询”。

alias-names 在整个查询中 都可以用来表示带别名的表达式。Interactive SQL 还在 **SELECT** 语句的每个输出列的顶部显示 别名。如果表达式后面 没有指定可选的 *alias-name*，Interactive SQL 将显示该表达式。如果为列别名使用与列名一样的名称或表达式，该名称将处理为带别名的列，而不是表列名。

INTO host-variable-list—仅用于嵌入式 SQL。它指定 **SELECT** 语句结果的位置。*select-list* 中的每一项都必须有一个 *host-variable*。选择列表中的项依次放入宿主变量中。每个

host-variable 还可以有一个指示符宿主变量，以便程序可以判定选择列表项是否为 NULL。

INTO variable-list—仅用在过程中。它指定 **SELECT** 语句结果的位置。选择列表中的每一项都必须有一个变量。选择列表中的项依次放入变量中。

INTO table-name—用于创建表并用数据填充表。

如果表名以 # 开头，则该表创建为临时表。否则，该表创建为永久基表。对于所要创建的永久表，查询必须满足以下条件：

- *select-list* 中包含多个项目，**INTO** 目标是单个 *table-name* 标示符，或
- *select-list* 包含一个 *，**INTO** 目标指定为 *owner.table*。

若要创建有一列的永久表，表名必须指定为 *owner.table*。可忽略临时表的所有者说明。

作为创建表的副作用，此语句会在执行前导致 **COMMIT**。执行此语句需要具有 **RESOURCE** 权限。新表未被授予任何权限：该语句是后接 **INSERT... SELECT** 的 **CREATE TABLE** 简写形式。

不允许在存储过程或函数中执行 **SELECT INTO**，因为 **SELECT INTO** 为原子语句，无法在原子语句中执行 **COMMIT**、**ROLLBACK** 或某些 **ROLLBACK TO SAVEPOINT** 语句。有关详细信息，请参见《系统管理指南第二卷》>“使用过程和批处理”>“控制语句”>“原子复合语句”和《系统管理指南第二卷》>“使用过程和批处理”>“过程中的事务和保存点”。

用此语句创建的表没有定义主键。可以使用 **ALTER TABLE** 添加主键。在对表应用任何 **UPDATE** 或 **DELETE** 之前应添加主键；否则，这些操作会使受影响的行的所有列值记录在事务日志中。

该子句仅限于在有效的 SQL Anywhere 查询中使用。不支持 Sybase IQ 扩展。

INTO LOCAL TEMPORARY TABLE—创建本地临时表并用查询结果对其进行填充。使用此子句时，临时表名不必以 # 开头。

FROM table-list—一行是从 *table-list* 指定的表和视图中检索的。连接可使用连接运算符进行指定。有关详细信息，请参见“FROM 子句”。不带 **FROM** 子句的 **SELECT** 语句可用于显示不是从表中派生的表达式的值。例如：

```
SELECT @@version
```

显示全局变量 @@version 的值。这等效于：

```
SELECT @@version
FROM DUMMY
```

注意： 如果省略 **FROM** 子句，或者查询中的所有表都在 **SYSTEM** 数据库空间中，则查询将由 SQL Anywhere 而非 Sybase IQ 处理且行为可能不同，特别是关于语法和语义限制和选项设置的影响方面。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，则可以通过添加 “**FROM iq_dummy**” 子句强制 Sybase IQ 处理查询，在这种情况下，**iq_dummy** 表示在数据库中创建的包含一行和一列的表。

WHERE search-condition—指定从 **FROM** 子句命名的表选择哪些行。还用于在多个表之间进行连接。这是通过在 **WHERE** 子句中放置一个条件来完成的，该条件将一个表中的一列或一组列与另一个表中的一列或一组列相关。两个表都必须在 **FROM** 子句中列出。

一组列的 **SELECT** 和 **WHERE** 子句中不允许使用相同的 **CASE** 语句。请参见《参考：构件块、表和过程》> “SQL 语言元素”> “搜索条件”。

Sybase IQ 还支持对子查询谓词执行析取。每个子查询可以与其它谓词一起显示在 **WHERE** 或 **HAVING** 子句内，并可使用 **AND** 或 **OR** 运算符进行组合。请参见《参考：构件块、表和过程》> “SQL 语言元素”> “搜索条件”> “搜索条件中的子查询”> “分离子查询谓词”。

GROUP BY—可以按列、别名或函数进行分组。**GROUP BY** 表达式也必须出现在选择列表中。指定列、别名或函数的每个不同的值集在查询结果均可以找到与之相对应的一行。结果行通常称为组，因为对于表列表中的每组行，结果中均有一行。如果是 **GROUP BY**，所有空值都视为完全相同。然后可将集合函数应用于这些组以获得有意义的结果。

GROUP BY 必须含有一个以上的常量。不需向 **GROUP BY** 子句添加常量即可选择分组查询中的常量。如果 **GROUP BY** 表达式只含有一个常量，则会返回错误，查询也会被拒绝。

如果使用 **GROUP BY**，选择列表、**HAVING** 子句和 **ORDER BY** 子句则无法引用除 **GROUP BY** 子句中之外的任何标示符。但有以下例外：*select-list* 和 **HAVING** 子句可以包含集合函数。

ROLLUP 运算符—**ROLLUP** 子句中的 **GROUP BY** 运算符可用不同的详尽程度来分析小计。它可以创建从详细级别一直累计到总计的小计。

ROLLUP 运算符要求以参数的方式提供分组表达式的有序列表。**ROLLUP** 首先计算 **GROUP BY** 中指定的标准集合值。然后，**ROLLUP** 在整个分组列的列表中从右侧移到左侧，并以累积方式创建更高级别的小计。在结尾处创建总计。如果 n 代表分组列数，则 **ROLLUP** 会创建 $n+1$ 个级别的小计。

对 **ROLLUP** 运算符的限制如下：

- **ROLLUP** 支持可用于 **GROUP BY** 子句的所有集合函数，但 **ROLLUP** 目前支持 **COUNT DISTINCT** 和 **SUM DISTINCT**。
- **ROLLUP** 仅可用于 **SELECT** 语句；在 **SELECT** 子查询中不可使用 **ROLLUP**。
- 当前不支持将多个 **ROLLUP**、**CUBE** 和 **GROUP BY** 列组合在同一个 **GROUP BY** 子句中的分组规范。
- 不支持以常量表达式作为 **GROUP BY** 键。

GROUPING 可与 **ROLLUP** 运算符配合使用来区分存储空值和 **ROLLUP** 创建的查询结果中的空值。

ROLLUP 语法:

```
SELECT ... [ GROUPING ( column-name ) ...] ... GROUP BY
[ expression [, ...] | ROLLUP ( expression [, ...] ) ]
```

有关运算符表达式的格式, 请参见《参考: 构件块、表和过程》> “SQL 语言元素”> “表达式”。

GROUPING 采用列名作为参数并返回布尔值:

表 13. 使用 **ROLLUP** 运算符时 **GROUPING** 返回的值

如果结果值是	GROUPING 返回
由 ROLLUP 运算创建的空值	1 (TRUE)
指示该行是小计所在行的空值	1 (TRUE)
并非由 ROLLUP 运算创建	0 (FALSE)
存储的 NULL	0 (FALSE)

有关 **ROLLUP** 示例, 请参见《系统管理指南第二卷》> “使用 OLAP”。

CUBE 运算符—Theoperator in the **CUBE** 子句中的 **GROUP BY** 运算符可将数据分布到具有多个维度的分组中, 并以此来分析数据。**CUBE** 需要分组表达式 (维度) 的有序列表作为参数, 并让 **SELECT** 语句计算所有可能维度组的组合的小计。

对 **CUBE** 运算符的限制如下:

- **CUBE** 支持可用于 **GROUP BY** 子句的所有集合函数, 但 **CUBE** 目前不支持 **COUNT DISTINCT** 或 **SUM DISTINCT**。
- **CUBE** 目前不支持逆分布解析函数 **PERCENTILE_CONT** 和 **PERCENTILE_DISC**。
- **CUBE** 仅可用于 **SELECT** 语句; 在 **SELECT** 子查询中不可使用 **CUBE**。
- 当前不支持将多个 **ROLLUP**、**CUBE** 和 **GROUP BY** 列组合在同一个 **GROUP BY** 子句中的 **GROUPING** 规范。
- 不支持以常量表达式作为 **GROUP BY** 键。

GROUPING 可与 **CUBE** 运算符配合使用来区分存储空值和 **CUBE** 创建的查询结果中的空值。

CUBE 语法:

```
SELECT ... [ GROUPING ( column-name ) ...] ... GROUP BY
[ expression [, ...] | CUBE ( expression [, ...] ) ]
```

GROUPING 采用列名作为参数并返回布尔值:

表 14. 使用 CUBE 运算符时 GROUPING 返回的值

如果结果值是	GROUPING 返回
由 CUBE 运算创建的空值	1 (TRUE)
指示该行是小计所在行的空值	1 (TRUE)
并非由 CUBE 运算创建	0 (FALSE)
存储的 NULL	0 (FALSE)

生成查询计划时，Sybase IQ 优化程序 会估计通过 **GROUP BY CUBE** 散列运算 生成的组的总数。MAX_CUBE_RESULTS 数据库选项 对优化程序视为可以运行的散列算法的估计行数 设置一个上限。如果实际行数 超过 MAX_CUBE_RESULT 选项值，优化程序将停止处理查询，并返回错误消息“估计数目：nnn 超过 GROUP BY CUBE 或 ROLLUP 的 DEFAULT_MAX_CUBE_RESULT”，其中 nnn 是 IQ 优化程序估计的数值。有关设置 MAX_CUBE_RESULT 选项的信息，请参见“MAX_CUBE_RESULT 选项”。

有关 **CUBE** 示例，请参见《系统管理指南第二卷》>“使用 OLAP”。

HAVING search-condition—根据组值而不是各行值。只有当此语句有 **GROUP BY** 子句或选择列表只包括集合函数时，才能使用 **HAVING** 子句。**HAVING** 子句中引用的任何列名都必须位于 **GROUP BY** 子句中或用作 **HAVING** 子句中的集合函数的参数。

ORDER BY—排序查询结果。**ORDER BY** 列表中的每一项均可标记为 **ASC** 以按升序排序，或者标记为 **DESC** 以按降序排序。如果两者都未指定，则假定为升序。如果表达式是整数 **n**，则查询结果按选择列表中的第 **n** 项排序。

在嵌入式 SQL 中，**SELECT** 语句用于从数据库中检索结果，并通过 **INTO** 子句将值放入宿主变量。**SELECT** 语句必须只返回一行。对于多行查询，必须使用游标。

不能在 **SELECT** 列表中包含 Java 类，但可以创建一个充当 Java 类的包装的函数或变量，然后选择它。

row-limitation 子句 - 行限制子句允许您只返回满足 **WHERE** 子句的行的子集。一次只能指定一个 **row-limitation** 子句。如果指定该子句，以一种有意义的方式对这些行进行排序需要使用 **ORDER BY** 子句。

LIMIT 和 **OFFSET** 参数可以是针对宿主变量、整数常量或整数变量的简单算术表达式。**LIMIT** 参数的结果值必须是大于或等于 0 的值。**OFFSET** 参数的结果值必须是大于或等于 0 的值。如果未指定 *offset-expression*，缺省值是 0。

行限制子句 **LIMIT***offset-expression, limit-expression* 等同于 **LIMIT***limit-expression***OFFSET***offset-expression*。

缺省情况下禁用 **LIMIT** 选项。使用 **RESERVED_KEYWORDS** 选项以启用 **LIMIT** 关键字。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase - 受 Adaptive Server Enterprise 支持，有一些语法差异。

权限

必须有指定表和视图的 SELECT 权限。

另请参见

- CREATE VIEW 语句 (第 151 页)
- DECLARE CURSOR 语句 [ESQL] [SP] (第 155 页)
- FETCH 语句 [ESQL] [SP] (第 188 页)
- FROM 子句 (第 194 页)
- MAX_CUBE_RESULT 选项 (第 398 页)
- OPEN 语句 [ESQL] [SP] (第 245 页)
- UNION 操作 (第 303 页)
- RESERVED_KEYWORDS 选项 (第 428 页)
- ROW_COUNT 选项 (第 430 页)
- SUBQUERY_CACHING_PREFERENCE 选项 (第 435 页)

SET 语句 [ESQL]

为 SQL 变量赋值。

语法

```
SET identifier = expression
```

示例

- **示例 1** – 下面这段代码可向数据库中插入一个大文本值：

```
EXEC SQL BEGIN DECLARE SECTION;
char buffer[5001];
EXEC SQL END DECLARE SECTION;

EXEC SQL CREATE VARIABLE hold_text VARCHAR;
EXEC SQL SET hold_text = '';
for(;;) {
    /* read some data into buffer ... */
    size = fread( buffer, 1, 5000, fp );
    if( size <= 0 ) break;

    /* buffer must be null-terminated */
```

```
buffer[size] = '\0';
/* add data to blob using concatenation */
EXEC SQL SET hold_text = hold_text || :buffer;
}
EXEC SQL INSERT INTO some_table VALUES ( 1, hold_text );
EXEC SQL DROP VARIABLE hold_text;
```

- **示例 2** - 下面这段代码可向数据库中插入一个大文本值:

```
EXEC SQL BEGIN DECLARE SECTION;
DECL_BINARY( 5000 ) buffer;
EXEC SQL END DECLARE SECTION;
EXEC SQL CREATE VARIABLE hold_blob LONG BINARY;
EXEC SQL SET hold_blob = '';
for(;;) {
    /* read some data into buffer ... */
    size = fread( &(amp;buffer.array), 1, 5000, fp );
    if( size <= 0 ) break;
    buffer.len = size;

    /* add data to blob using concatenation
       Note that concatenation works for
       binary data too! */
    EXEC SQL SET hold_blob = hold_blob || :buffer;
}
EXEC SQL INSERT INTO some_table VALUES ( 1, hold_blob );
EXEC SQL DROP VARIABLE hold_blob;
```

用法

SET 语句向以前使用 **CREATE VARIABLE** 语句创建的变量赋予一个新值。

SQL 语句中允许使用列名的任何位置都可以使用变量。如果没有与标识符相匹配的列名，数据库服务器会检查是否存在这样的匹配变量，然后使用其值。

变量是当前连接的本地对象，当您与数据库断开连接或使用 **DROP VARIABLE** 时消失。它们不受 **COMMIT** 或 **ROLLBACK** 语句的影响。

从嵌入式 SQL 程序中为 **INSERT** 或 **UPDATE** 语句创建大文本或二进制对象时需要使用变量，因为嵌入式 SQL 宿主变量仅限于 32,767 个字节。

请参见《参考：构件块、表和过程》>“SQL 语言元素”>“搜索条件”。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase—不支持。在 Adaptive Server Enterprise 中，用不带表的 **SELECT** 语句给变量赋值，这是一种 Transact-SQL 语法，Sybase IQ 也支持该语法。在 Adaptive Server Enterprise 中，**SET** 语句用于设置数据库选项。

权限

无

另请参见

- CREATE VARIABLE 语句（第 149 页）
- DROP VARIABLE 语句（第 181 页）

SET 语句 [T-SQL]

以与 Adaptive Server Enterprise 兼容的方式 设置数据库选项。

语法

```
SET option-name option-value
```

用法

在 Sybase IQ 中，用 **SET OPTION** 语句设置数据库选项。不过，Sybase IQ 还支持 用 Adaptive Server Enterprise**SET** 语句设置 对兼容性特别有用的选项集。

表 15. Transact-SQL SET 选项

选项名	选项值
ANSINULL	ON OFF
ANSI_PERMISSIONS	ON OFF
CLOSE_ON_ENDTRANS	ON
QUOTED_IDENTIFIER	ON OFF
ROWCOUNT	<i>integer</i>
STRING_RTRUNCATION	ON OFF
TRANSACTION ISOLATION LEVEL	0 1 2 3

您可以使用 Sybase IQ 和 Adaptive Server Enterprise 中的 Transact-SQL **SET** 语句 来设置以下选项：

- SET ANSINULL { ON | OFF }—在 Sybase IQ 和 Adaptive Server Enterprise 中，比较值与 NULL 的缺省行为是不同的。将 **ANSINULL** 设置为 OFF 可提供与 Transact-SQL 兼容的 NULL 比较。
- SET ANSI_PERMISSIONS { ON | OFF }—在 Sybase IQ 和 Adaptive Server Enterprise 中，关于执行含有列引用的 **DELETE** 所需 权限的缺省行为是 不同的。将 **ANSI_PERMISSIONS** 设置为 OFF 可提供与 Transact-SQL 兼容的 **DELETE** 权限。
- SET CLOSE_ON_ENDTRANS { ON }—当 **CLOSE_ON_ENDTRANS** 设置为 ON（缺省及唯一允许的值）时，游标在事务结束时关闭。当选项设置为 ON 时，**CLOSE_ON_ENDTRANS** 提供 Transact-SQL 兼容的行为。

- **SET QUOTED_IDENTIFIER { ON | OFF }**—控制用双引号括起来的字符串是被解释为标识符 (ON) 还是普通字符串 (OFF)。
- **SET ROWCOUNT *integer***—Transact-SQL **ROWCOUNT** 选项将为所有游标读取的行数限制为指定的整数。这包括通过重新定位游标读取的行。超出此最大值的所有读取操作都返回警告。当应 **OPEN** 请求返回游标的行数估计值时应考虑该选项设置。

注意： Sybase IQ 支持 `@@rowcount` 全局变量。**SELECT**、**INSERT**、**DELETE** 和 **UPDATE** 语句影响 **ROWCOUNT** 选项 的值。**ROWCOUNT** 选项不影响 游标操作、**IF** 语句或创建/删除 表或过程。

在 Sybase IQ 中，如果 **ROWCOUNT** 设置比 **dbisql** 可显示 的行数大，**dbisql** 可能会执行额外的读取 以重新定位游标。实际显示 的行数可能比请求的数目少。另外，如果由于截断警告而重新读取任何行， 计数可能不准确。

如果值为零，则重置该选项以获取所有行。

- **SET STRING_RTRUNCATION { ON | OFF }**—在 Sybase IQ 和 Adaptive Server Enterprise 中， 指定 SQL 字符串数据时 截断非空格字符的缺省行为 是不同的。将 **STRING_RTRUNCATION** 设置为 ON 可提供与 Transact-SQL 兼容的字符串比较， 其中包括十六进制字符串（二进制数据类型）比较。
- **SET TRANSACTION ISOLATION LEVEL { 0 | 1 | 2 | 3 }**—设置当前连接的 锁定隔离级别， 如《系统管理指南第一卷》>“事务和版本控制”中所述。对于 Adaptive Server Enterprise，只有 1 和 3 是有效的选项。对于 Sybase IQ， 只有 3 是有效选项。

此外，出于兼容性考虑，Sybase IQ 也允许以下 **SET** 语句， 但它不起作用：

- **SET PREFETCH { ON | OFF }**

标准

- SQL—ISO/ANSI SQL 语法的 Transact-SQL 扩展。
- Sybase—Sybase IQ 支持 Adaptive Server Enterprise 数据库 选项的子集。

权限

无

另请参见

- **SET OPTION** 语句（第 291 页）

SET CONNECTION 语句 [ESQL] [Interactive SQL]

更改活动的数据库连接。

语法

```
SET CONNECTION [connection-name]
```

参数

- **connection-name**: - 标识符、字符串或宿主变量

示例

- **示例 1** - 在嵌入式 SQL 中:

```
EXEC SQL SET CONNECTION :conn_name
```

- **示例 2** - 在 **dbisql** 中，将当前连接设置为名为“conn1”的连接:

```
SET CONNECTION conn1
```

用法

当前连接状态被保存起来，并在重新成为活动连接时恢复。如果省略 *connection-name*，并且存在未命名的连接，则该连接将成为活动连接。

注意： 当在嵌入式 SQL 中打开游标时，它们与当前连接关联。连接更改后，游标名不再能访问。这些游标在原地保持活动，并在关联的连接再次成为活动连接时恢复可访问性。

标准

- SQL - **dbisql** 用法是 ISO/ANSI SQL 语法的供应商扩展。嵌入式 SQL 是完整级别特性。
- Sybase - 受 Open Client/Open Server 支持。

权限

无

另请参见

- CONNECT 语句 [ESQL] [Interactive SQL]（第 65 页）
- DISCONNECT 语句 [Interactive SQL]（第 169 页）

SET DESCRIPTOR 语句 [ESQL]

描述 SQL 描述符区中的变量，将数据放入描述符区。

语法

```
SET DESCRIPTOR descriptor-name
... { COUNT = { integer | hostvar }
    | VALUE n assignment [, ...] }
```

参数

- **assignment:** - { { **TYPE** | **SCALE** | **PRECISION** | **LENGTH** | **INDICATOR** } = { *integer* | *hostvar* } | **DATA** = *hostvar* }

示例

- 示例 1 - 请参见 ALLOCATE DESCRIPTOR 语句 [ESQL]。

用法

SET...COUNT 设置描述符区内所描述变量的数目。此数值不能超过分配描述符区时指定的变量数。

值 *n* 指定对其上执行赋值操作的描述符区中的变量。

在执行 **SET...DATA** 时会进行类型检查，以确保描述符区中的变量类型与宿主变量相同。

如果发生错误，SQLCA 中会返回代码。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase—受 Open Client/Open Server 支持。

权限

无

另请参见

- ALLOCATE DESCRIPTOR 语句 [ESQL]（第 5 页）
- DEALLOCATE DESCRIPTOR 语句 [ESQL]（第 152 页）

SET OPTION 语句

更改数据库选项。

语法

```
SET [ EXISTING ] [ TEMPORARY ] OPTION
... [ userid. | PUBLIC.]option-name = [ option-value ]
```

参数

- **userid:** - 标识符、字符串或宿主变量
- **option-name:** - 标识符、字符串或宿主变量
- **option-value:** - 宿主变量（允许使用指示符）、字符串、标识符或数字

示例

- **示例 1** - 设置 **DATE_FORMAT** 选项:

```
SET OPTION public.date_format = 'Mmm dd yyyy'
```

- **示例 2** - 将 **WAIT_FOR_COMMIT** 选项设置为 on:

```
SET OPTION wait_for_commit = 'on'
```

- **示例 3** - 嵌入式 SQL 示例:

```
EXEC SQL SET OPTION :user.:option_name = :value;
EXEC SQL SET TEMPORARY OPTION Date_format = 'mm/dd/yyyy';
```

用法

SET OPTION 语句用于更改影响数据库行为以及数据库与 Transact-SQL 兼容性的选项。设置选项的值可更改所有用户或单个用户的行为，作用域可以是临时的，也可以是永久的。

选项的分类如下:

- 常规数据库选项
- Transact-SQL 兼容性数据库选项

指定某个用户 ID 或 PUBLIC 用户 ID 可确定该选项是为单个用户、为由 *userid* 表示的用户组，还是为 PUBLIC 用户 ID（所有用户都是其成员的用户组）设置的。如果选项适用于组用户 ID，选项设置就不由组的成员继承 — 更改仅应用到组用户 ID。如果未指定用户组，则所做选项更改应用于发出 **SET OPTION** 语句的当前登录用户 ID。

例如，以下语句对 PUBLIC 用户 ID 应用选项更改:

```
SET OPTION Public.login_mode = standard
```

只有拥有 DBA 特权的用户才有权设置 PUBLIC 用户 ID 的选项。

在嵌入式 SQL 中，只有数据库选项可以临时设置。

为 PUBLIC 用户 ID 更改选项的值，相当于为没有设置该值的所有用户设置此选项的值。如果某选项没有 PUBLIC 用户 ID 设置，则无法为单个用户 ID 设置此选项的值。

用户不能设置其他用户的选项，除非拥有 DBA 权限。

用户可使用 **SET OPTION** 语句来更改其自己用户 ID 的值。只有在具有 DBA 权限的情况下才允许为其他用户 ID 设置选项的值。

如果使用 **EXISTING** 关键字，则无法为个别用户 ID 设置选项值，除非已经有该选项的 PUBLIC 用户 ID 设置。

将 **TEMPORARY** 关键字添加到 **SET OPTION** 语句中会改变更改生效的持续时间。如果没有 **TEMPORARY** 关键字，则对选项的更改将是永久性更改：在使用 **SET OPTION** 显式地更改之前，它不会变化。

当使用单个用户 ID 应用 **SET TEMPORARY OPTION** 时，新选项值在该用户登录到数据库后即生效。

如果对 PUBLIC 用户 ID 使用 **SET TEMPORARY OPTION**，则更改在数据库运行时间内一直生效。当数据库关闭时，PUBLIC 用户 ID 的 **TEMPORARY** 选项恢复为其永久值。

相对于永久性地设置选项的值，临时设置 PUBLIC 用户 ID 的选项更具安全优势。例如，在启用 LOGIN_MODE 选项时，数据库依赖于其所在的系统的登录安全性。临时启用该选项意味着，对于依赖于 Windows 域的安全性的数据库，如果关闭该数据库并将它复制到本地计算机，它的安全不会受到威胁。在这种情况下，临时启用的 LOGIN_MODE 恢复为它的永久值，这可能是“标准”（一种不允许集成登录的模式）。

如果忽略 *option-value*，将从数据库中删除指定的选项设置。如果它是个人选项设置，则所用的值会恢复为 PUBLIC 设置。如果删除一个 **TEMPORARY** 选项，则选项设置会恢复为永久设置。

注意： 对于所有接受整数值的数据库选项，Sybase IQ 会截去 *option-value* 设置的任何小数部分，只保留整数值。例如，值 3.8 将被截断为 3。

当设置为字符串时，*option-value* 的最大长度为 127 个字节。

警告！ 不支持从游标中读取行时更改选项设置，因为这会导致意外的行为。例如，在从游标中读取时更改 DATE_FORMAT 设置会在结果集的行中返回不同的日期格式。不要在读取行时更改选项设置。

有关特定数据库选项的信息，请参见“数据库选项”。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。

- Sybase - Adaptive Server Enterprise 不支持。Sybase IQ 通过 **SET** 语句支持某些 Adaptive Server Enterprise 选项。

权限

设置自己的选项不需要任何权限。必须有 DBA 权限才能为其他用户或 PUBLIC 设置数据库选项。

另请参见

- 数据库选项（第 315 页）

SET OPTION 语句 [Interactive SQL]

更改 Interactive SQL (**dbisql**) 选项。

语法

语法 1

```
SET [ TEMPORARY ] OPTION
... [ userid. | PUBLIC. ] option-name = [ option-value ]
```

语法 2

```
SET PERMANENT
```

语法 3

```
SET
```

参数

- **userid**: - 标识符、字符串或宿主变量
- **option-name**: - 标识符、字符串或宿主变量
- **option-value**: - 宿主变量（允许使用指示符）、字符串、标识符或数字

用法

SET PERMANENT（语法 2）在 **SYSOPTION** 系统表中存储所有当前的 **dbisql** 选项。每次为当前用户 ID 启动 **dbisql** 时，都会自动建立这些设置。

语法 3 用于显示所有当前选项设置。如果为 **dbisql** 或数据库服务器设置了临时选项，将显示这些设置；否则，将显示永久选项设置。

如果在设置选项时错误地键入该选项的名称，该错误名称将保存在 **SYSOPTION** 表中。可以通过使用选项名称后跟等号但不带值的形式设置选项 **PUBLIC**，来从 **SYSOPTION** 表中删除错误键入的名称：

```
SET OPTION PUBLIC.a_mistyped_name=;
```

另请参见

- 数据库选项（第 315 页）

SET SQLCA 语句 [ESQL]

指示 SQL 预处理器使用缺省全局 *sqlca* 以外的 SQLCA。

语法

```
SET SQLCA sqlca
```

参数

- **Sqlca:** - 标识符或字符串

示例

- **示例 1** - 该函数位于 Windows DLL 中。每个使用 DLL 的应用程序都有自己的 SQLCA。

```
an_sql_code FAR PASCAL ExecutesSQL( an_application *app, char
*com )
{
    EXEC SQL BEGIN DECLARE SECTION;
    char *sqlcommand;
    EXEC SQL END DECLARE SECTION;
    EXEC SQL SET SQLCA "&app->.sqlca";
    sqlcommand = com;
    EXEC SQL WHENEVER SQLERROR CONTINUE;
    EXEC SQL EXECUTE IMMEDIATE :sqlcommand;
    return( SQLCODE );
}
```

用法

当前 SQLCA 指针隐式传递给各嵌入式 SQL 语句中的数据库接口库。C 源文件中位于此语句后的所有嵌入式 SQL 语句都使用新的 SQLCA。仅当编写重入代码时才需要使用此语句。*sqlca* 应该引用局部变量。任何全局变量或模块静态变量都可能会由另一线程修改。

请参见“SQL Anywhere Server - 编程” > “嵌入式 SQL” > “SQL 通信区域 (SQLDA)”。

注意： 此参考指向 SQL Anywhere 文档。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase—不受 Open Client/Open Server 支持。

权限

无

SIGNAL 语句

发出例外情况的信号。

语法

```
SIGNAL exception-name
```

用法

SIGNAL 可让您引发异常。有关如何处理异常的说明，请参见《系统管理指南第二卷》>“使用过程和批处理”。

标准

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

权限

无

另请参见

- BEGIN……END 语句（第 49 页）
- RESIGNAL 语句（第 261 页）

START DATABASE 语句 [Interactive SQL]

启动指定数据库服务器上的数据库。

语法

```
START DATABASE database-file  
... [ AS database-name ]  
... [ ON engine-name ]
```

```
... [ AUTOSTOP { YES | NO } ]  
... [ KEY key ]
```

示例

- **示例 1** - 在 UNIX 系统上，启动当前服务器上的数据库文件 `/s1/sybase/sample_2.db`:

```
START DATABASE '/s1/sybase/sample_2.db'
```

- **示例 2** - 在 Windows 系统上，作为 `sam2` 启动服务器 `eng1` 上的数据库文件 `c:\sybase\sample_2.db`:

```
START DATABASE 'c:\sybase\sample_2.db'  
AS sam2  
ON eng1
```

用法

数据库服务器必须正在运行。必须指定数据库文件的完整路径，除非此文件位于当前目录中。

START DATABASE 语句不会将 **dbisql** 连接到指定的数据库：必须发出 **CONNECT** 语句才能进行连接。

如果未指定 *database-name*，则为数据库分配缺省名。此缺省名是数据库文件的根。例如，为文件 `c:\sybase\IQ_15\demo\iqdemo.db` 中的数据库赋予缺省名 `iqdemo`。

如果未指定 *engine-name*，则假定采用缺省数据库服务器。缺省数据库服务器是当前运行的服务器之中首先启动的服务器。

AUTOSTOP 子句的缺省设置为 **YES**。如果将 **AUTOSTOP** 设置为 **YES**，则在删除连接数据库的最后一个连接时将卸载数据库。如果将 **AUTOSTOP** 设置为 **NO**，则不卸载数据库。

如果数据库是强加密的，请使用 **KEY** 子句输入 **KEY** 值（口令）。

Sybase 建议仅启动给定 Sybase IQ 数据库服务器 上的一个数据库。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不适用。

权限

必须具有 **DBA** 权限。

START ENGINE 语句 [Interactive SQL]

启动数据库服务器。

语法

```
START ENGINE AS engine-name [ STARTLINE command-string ]
```

示例

- 示例 1 - 启动名为 eng1 的数据库服务器，而不启动其上的任何数据库：

```
START ENGINE AS eng1
```

- 示例 2 - **STARTLINE** 子句的用法：

```
START ENGINE AS eng1 STARTLINE 'start_iq -c 8096'
```

用法

若要为服务器指定一组选项，请在命令字符串中使用 **STARTLINE** 关键字。

符合《实用程序指南》> “start_iq 数据库服务器启动实用程序”中的数据库服务器命令行说明的命令字符串即为有效。

注意： 需要使用几个服务器选项才能使 Sybase IQ 良好地工作。为了确保使用正确的选项集，Sybase 建议通过使用 Sybase Central 或具有 **start_iq** 命令的配置文件来启动服务器。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不适用。

权限

无

另请参见

- STOP ENGINE 语句 [Interactive SQL]（第 299 页）

START JAVA 语句

启动 Java VM。

语法

```
START EXTERNAL ENVIRONMENT JAVA
```

示例

- 示例 1 – 启动 Java VM。

```
START EXTERNAL ENVIRONMENT JAVA
```

用法

可在方便的时候使用 **START EXTERNAL ENVIRONMENT JAVA** 装载 Java VM，这样如果用户开始使用 Java 功能，则装载 Java VM 时一开始不会出现暂停。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase—不适用。

权限

必须具有 DBA 权限。

另请参见

- STOP JAVA 语句（第 300 页）

STOP DATABASE 语句 [Interactive SQL]

停止指定数据库服务器上的数据库。

语法

```
STOP DATABASE database-name  
... [ ON engine-name ]  
... [ UNCONDITIONALLY ]
```

示例

- 示例 1 – 停止缺省服务器上名为 `sample` 的数据库：

```
STOP DATABASE sample
```

用法

如果未指定 *engine-name*，则在所有运行的引擎中搜索具有指定名称的数据库。

启动数据库时 **-n** 参数或 **DBN (DatabaseName)** 连接参数中指定的名称是 *database-name*。此名称通常是保存 Catalog 存储的数据库文件的文件名，不带 *.db* 扩展名，但也可以是用户定义的任何名称。

如果提供了 **UNCONDITIONALLY**，则数据库即使存在连接也会停止。如果未指定 **UNCONDITIONALLY**，则当数据库存在连接时不停止数据库。

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不适用。

权限

必须具有 DBA 权限。

另请参见

- DISCONNECT 语句 [Interactive SQL]（第 169 页）
- START DATABASE 语句 [Interactive SQL]（第 295 页）

STOP ENGINE 语句 [Interactive SQL]

停止数据库服务器。

语法

```
STOP ENGINE engine-name [ UNCONDITIONALLY ]
```

示例

- **示例 1** - 停止名为 *sample* 的数据库服务器：

```
STOP ENGINE sample
```

用法

如果提供了 **UNCONDITIONALLY**，则数据库即使存在连接也会停止。如果未指定 **UNCONDITIONALLY**，则当数据库存在连接时不停止数据库。

SQL 语句

标准

- SQL - ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 不适用。

权限

无

另请参见

- START ENGINE 语句 [Interactive SQL] (第 297 页)

STOP JAVA 语句

用于释放与 Java VM 相关联的资源。

语法

STOP EXTERNAL ENVIRONMENT JAVA

用法

STOP EXTERNAL ENVIRONMENT JAVA 的主要用途是减少 系统资源用量。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase—不适用。

权限

DBA 授权

另请参见

- START JAVA 语句 (第 298 页)

SYNCHRONIZE JOIN INDEX 语句

在更新一个或多个连接索引的基表中的某个基表后同步相应索引。

语法

SYNCHRONIZE JOIN INDEX [*join-index-name* [, *join-index-name*]...]

示例

- **示例 1** - 同步连接索引 emp_dept_join1 和 emp_dept_join2:

```
SYNCHRONIZE JOIN INDEX emp_dept_join1, emp_dept_join2
```

用法

当更新某连接索引对应的基表时，**Sybase IQ** 会将此连接索引 标记为不可用。先前利用此连接索引的查询 将改为执行即席连接，这可能会影响 这些查询的性能。使用 **SYNCHRONIZE JOIN INDEX** 命令可以更新 此连接索引，以使其 可供查询使用。

注意： 连接索引定义两个表列之间的一对多关系（亦称作主键对外键）。如果在“一个”（或主键）列中插入内容导致出现一个或多个重复值，则连接索引将变为无效且无法同步。必须先删除包含重复值的行，然后 **SYNCHRONIZE JOIN INDEX** 才能使其重新有效。

同步连接索引可能会非常耗时，具体取决于组成连接的基表的大小。何时使用此命令的决定权 在您手中。您可以将其作为批处理作业 安排在您预期系统工作量较少的晚间或周末进行。您可以在 **Sybase IQ** 提交 一系列插入和删除之后立即执行此命令，以使连接索引尽早可供使用。但是，请勿每次 插入或删除之后都同步连接索引，这是因为更新连接索引的时间取决于针对表的更新的顺序。

SYNCHRONIZE JOIN INDEX 允许您指定 多个 *join-index-name*，以逗号分隔。您必须是每个连接索引的所有者或者是 **DBA**。如果不 指定 *join-index-name*，则 **Sybase IQ** 将同步您拥有的 所有连接索引（如果您是 **DBA**，则同步数据库中的所有连接索引），这可能会对系统性能造成不利影响。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase—不适用。

权限

必须是连接索引的所有者或者是 **DBA**。

另请参见

- **CREATE JOIN INDEX** 语句（第 104 页）

TRIGGER EVENT 语句

触发指定事件。事件可以是 为事件触发器定义的事件，或是调度事件。

语法

```
TRIGGER EVENT event-name [ ( parm = value, ... ) ]
```

用法

操作与特定的触发器条件或 **CREATE EVENT** 语句的调度有关。即使没有到达预定时间或出现触发器条件，您也可以使用 **TRIGGER EVENT** 强制执行事件处理程序。

TRIGGER EVENT 不执行已禁用的事件处理程序

触发条件导致执行事件处理程序时，数据库服务器可以使用 `event_parameter` 函数为事件处理程序提供上下文信息。使用 **TRIGGER EVENT** 可以显式提供这些参数，以模拟事件处理程序的上下文。

当触发事件时，指定事件名称。通过查询系统表 `SYSEVENT`，可以列出事件名称。例如：

```
SELECT event_id, event_name FROM SYS.SYSEVENT
```

请参见《系统管理指南第二卷》>“使用日程表和事件自动完成任务”。

权限

必须具有 DBA 权限。

另请参见

- ALTER EVENT 语句（第 13 页）
- CREATE EVENT 语句（第 82 页）

TRUNCATE TABLE 语句

从表中删除所有行，而不删除表定义。

语法

语法 1

```
TRUNCATE TABLE [ owner. ] table-name
```

语法 2

```
TRUNCATE TABLE [ owner . ] table [ PARTITION partition-name ]
```

示例

- 示例 1 – 删除 Sale 表的所有行：

```
TRUNCATE TABLE Sale
```

用法

TRUNCATE TABLE 等效于不带 **WHERE** 子句的 **DELETE** 语句，只不过不在事务日志中记入各个行的删除情况。执行 **TRUNCATE TABLE** 语句后，表结构及所有索引仍然存在，直至发出 **DROP TABLE** 语句。列定义和约束保持不变，且权限仍然有效。

与数据定义语句一样，**TRUNCATE TABLE** 语句作为单个语句记入事务日志。每个删除的行不记入事务日志。

分区子句指定要截断哪个分区。它不会影响其它分区中的数据。

另请参见《系统管理指南第一卷》>“事务和版本控制”。

标准

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

权限

- 必须是表的所有者或者有 **DBA** 权限。
- 对于临时表和基表，可以执行 **TRUNCATE TABLE**，而其他用户则对这样的表具有读取访问权限。此行为有别于 **SQL Anywhere**，在其中需要具有独占访问权限才能截断基表。**Sybase IQ** 表版本控制确保在进行 **TRUNCATE TABLE** 时其他用户具有读取访问权限；但是，这些用户看到的表的版本取决于读取和写入事务的提交时间。

另请参见

- **DELETE** 语句（第 163 页）

UNION 操作

合并两个或多个 **select** 语句的结果。

语法

```

      select-without-order-by
... UNION [ ALL ] select-without-order-by
... [ UNION [ ALL ] select-without-order-by ]...
... [ ORDER BY integer [ ASC | DESC ] [, ...] ]

```

示例

- 示例 1 – 列出雇员和客户的所有不重复的姓：

```

SELECT Surname
FROM Employees

```

```
UNION
SELECT Surname
FROM Customers
```

用法

使用 **UNION** 可将多个 **SELECT** 语句的结果合并成一个较大的结果。各构成的 **SELECT** 语句在选择列表中必须有相同的项目数，并且不能包含 **ORDER BY** 子句。请参见“FROM 子句”。

UNION ALL 的结果是合并各构成的 **SELECT** 语句的结果。**UNION** 的结果与 **UNION ALL** 的结果相同，但会删除重复的行。删除重复的行需要额外的处理，所以应尽量使用 **UNION ALL** 而不是 **UNION**。

如果两个选择列表中的相应项具有不同的数据类型，Sybase IQ 将为结果中的相应列选择数据类型，并自动相应地转换各构成的 **SELECT** 语句中的列。

如果使用 **ORDER BY**，则仅允许按照列表顺序使用整数。这些整数指定要排序的列的位置。

显示的列名称与第一个 **SELECT** 语句显示的列名称相同。

注意： 当 **SELECT** 语句包括常量值和 **UNION ALL** 视图，但省略了 **FROM** 子句时，使用 `iq_dummy` 可避免发生错误。有关详细信息，请参见“FROM 子句”。

标准

- SQL—符合 ISO/ANSI SQL 标准。
- Sybase - 受 Adaptive Server Enterprise 支持，还支持 **COMPUTE** 子句。

权限

必须具有各构成的 **SELECT** 语句的 **SELECT** 权限。

另请参见

- FROM 子句（第 194 页）
- SELECT 语句（第 277 页）

UPDATE 语句

修改单个表的现有行，或修改仅包含一个表的视图的现有行。

语法

```
UPDATE table
... SET [column-name = expression, ...
... [ FROM table-expression, ]
```



```
... [ WHERE search-condition ]
... [ ORDER BY expression [ ASC | DESC ] , ...]

FROM table-expression
```

参数

- **table-expression:** - *table-spec* | *table-expression join-type table-spec* [**ON** *condition*] | *table-expression*, ...

示例

- **示例 1** - 将雇员 Philip Chin (雇员 129) 从销售部转移到市场部:

```
UPDATE Employees
SET DepartmentID = 400
WHERE EmployeeID = 129;
```

- **示例 2** - 市场部 (400) 将奖金从每位雇员基本薪水的 4% 提高到 6%:

```
UPDATE Employees
SET bonus = base * 6/100
WHERE DepartmentID = 400;
```

- **示例 3** - 每位雇员由于获得部门奖金而增加了收入:

```
UPDATE Employees
SET emp.Salary = emp.Salary + dept.bonus
FROM Employees emp, Departments dept
WHERE emp.DepartmentID = dept.DepartmentID;
```

- **示例 4** - 另一种由于部门奖金而增加每位雇员收入的方法:

```
UPDATE Employees
SET emp.salary = emp.salary + dept.bonus
FROM Employees emp JOIN Departments dept
ON emp.DepartmentID = dept.DepartmentID;
```

用法

您对其使用 **UPDATE** 的表可能是基表或临时表。

注意: 基表不能构成任何连接索引。

每个指定列被设置为等号右边表达式的值。表达式中甚至可以使用 *column-name* - 将使用旧值。

FROM 子句可以包含多个表及连接条件，并返回由连接条件和/或 **WHERE** 条件指定和过滤的所有表的所有列。

在 **FROM** 子句中使用错误的连接条件会导致不可预料的结果。如果 **FROM** 子句指定一对多连接且 **SET** 子句引用连接“多”侧的某单元，则从选择的第一个值更新此单元。换句话说，如果连接条件导致按照行 ID 更新表的多个行，则返回的第一个行将成为更新结果。例如:

```
UPDATE T1
SET T1.c2 = T2.c2
FROM T1 JOIN TO T2
ON T1.c1 = T2.c1
```

如果表 T2 依照 T2.c1 具有多个行，则结果可能如下：

T2.c1	T2.c2	T2.c3
1	4	3
1	8	1
1	6	4
1	5	2

如果不带 **ORDER BY** 子句，则 T1.c2 可能为 4、6、8 或 9。

- 如果带有 **ORDER BY T2.c3**，T1.c2 则更新为 8。
- 如果带有 **ORDER BY T2.c3 DESC**，T1.c2 则更新为 6。

Sybase IQ 拒绝要更新的表 在外部连接的空值提供方 的任何 **UPDATE** 语句。换句话说：

- 在左外部连接中，连接左侧的表不能缺少有关连接列的任何行。
- 在右外部连接中，连接右侧的表不能缺少有关连接列的任何行。
- 在完全外部连接中，两侧的表均不能缺少有关连接列的任何行。

例如，在下面的语句中，表 T1 在左外部连接的左侧，因此不能缺少任何行：

```
UPDATE T1
SET T1.c2 = T2.c4
FROM T1 LEFT OUTER JOIN T2
ON T1.rowid = T2.rowid
```

通常情况下，行以什么顺序更新并不重要。但是，与 **NUMBER(*)** 函数一起使用时，排序可用于按指定的顺序在行中添加递增的编号。如果不使用 **NUMBER(*)** 函数，请避免使用 **ORDER BY** 子句，因为 **UPDATE** 语句在不使用此子句的情况下性能会更好。

在 **UPDATE** 语句中，如果在 **SET** 子句中使用 **NUMBER(*)** 函数且 **FROM** 子句指定一对多连接，则 **NUMBER(*)** 将生成增加但由于删除行而不按顺序递增的唯一编号。有关 **NUMBER(*)** 函数的详细信息，请参见《参考：构件块、表和过程》>“SQL 函数”>“按字母顺序排列的函数列表”>“NULLIF 函数 [杂项]”。

当 **FROM** 子句包含多个连接的表时，可以使用 **ORDER BY** 子句控制 **UPDATE** 的结果。

Sybase IQ 忽略搜索的 **UPDATE** 中的 **ORDER BY** 子句，并返回指示相应语法不是有效 ANSI 语法的信息。

如果未指定 **WHERE** 子句，则更新所有行。如果指定 **WHERE** 子句，则 Sybase IQ 仅更新满足搜索条件的行。

每个 **SET** 子句的左侧均必须是基表中的列。

如果定义视图的 **SELECT** 语句不包含 **GROUP BY** 子句、集合函数或不涉及 **UNION** 子句，则可以更新视图。视图应仅包含一个表。

插入到表中的字符串始终按输入时的大小写存储，而不管数据库是否区分大小写。因此，用字符串 **Value** 更新的字符数据类型列在数据库中保存的形式始终是 **V** 为大写，其它字母均为小写。**SELECT** 语句以 **Value** 形式返回该字符串。但是，如果数据库不区分大小写，所有比较都会使 **Value** 与 **value**、**VALUE** 等相同。IQ 服务器可以大小写字母的任意组合返回结果，因此，不区分大小写的数据库中的结果不能区分大小写 (**CASE IGNORE**)。而且，如果单列主键已经包含 **Value** 条目，则会拒绝 **value** 的 **INSERT**，因为它会导致主键不唯一。

如果更新违背任何检查约束，则将回退整个语句。

Sybase IQ 支持在 **SET** 子句中使用标量子查询，例如：

```
UPDATE r
SET r.o = (SELECT MAX(t.o)
FROM t ... WHERE t.y = r.y),
r.s = (SELECT SUM(x.s)
FROM x ...
WHERE x.x = r.x)
WHERE r.a = 10
```

Sybase IQ 支持在 **UPDATE** 语句中使用 **DEFAULT** 列值。如果某列具有 **DEFAULT** 值，则在未显式修改此列的值的任何 **UPDATE** 语句中，该 **DEFAULT** 值将用作此列的值。

有关列的 **DEFAULT** 值用法的详细信息，请参见《系统管理指南第一卷》>“数据完整性”>“列缺省值有助于数据完整性”。

有关更新作为其它类型 **DEFAULT** 列的 **IDENTITY/AUTOINCREMENT** 列的详细信息，请参见“**CREATE TABLE** 语句”。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase - 除下列例外情形之外，IQ **UPDATE** 语句的语法通常与 Adaptive Server Enterprise **UPDATE** 语句的语法 1 兼容：Sybase IQ 支持在 **FROM** 子句中使用多个表及连接条件。
远程表的更新受限于 CIS 支持的 Sybase IQ 语法，如《系统管理指南第二卷》>“访问远程数据”和《系统管理指南第二卷》>“用于远程数据访问的服务器类”所述。

权限

必须具有所修改列的 **UPDATE** 权限。

另请参见

- **CREATE TABLE** 语句 (第 136 页)

UPDATE (定位) 语句 [ESQL] [SP]

修改位于游标当前位置的数据。

语法

```

UPDATE
  table-list
SET
  set-item, ...
WHERE
  CURRENT
  OF
  cursor-name

```

参数

- **cursor-name:** - identifier | hostvar
- **set-item:** - *column-name* [*.field-name*··] = *scalar-value*

示例

- **示例 1** - 使用 **WHERE CURRENT OF** 游标的 **UPDATE** 语句:

```

UPDATE Employees SET surname = 'Jones'
WHERE CURRENT OF emp_cursor

```

用法

这种形式的 **UPDATE** 语句更新指定游标的当前行。当前行被定义为从游标中最近成功读取的一行，对游标的最后一项操作不能是定位 **DELETE** 语句。

SET—*set-item* 中引用的列必须位于更新的基表中。它们不能引用别名或其它表或视图中的列。如果要更新的表在游标说明中被赋予了一个相关名，那么在 **SET** 子句中必须使用此相关名。**SET** 子句右侧的表达式可能引用查询的 **SELECT** 子句中的列、常量、变量和表达式。*set-item* 表达式不能包含函数或表达式。

在指定查询的当前行中，请求的列被设置为该行的指定值。这些列必须位于指定的打开游标的选择列表中。

定位 **UPDATE** 语句影响的更改在游标结果集中是可见的，除非客户端缓存阻止看到这些更改。已进行更新以使其不再满足打开的游标的 **WHERE** 子句的行仍然可见。

Sybase 不建议在 **WHERE CURRENT OF** 子句中使用 **ORDER BY**。可能会更新 **ORDER BY** 列，但结果集不会重新排序。结果似乎未按顺序提取并且似乎是错误的。

由于 Sybase IQ 不支持 **CREATE VIEW... WITH CHECK OPTION**，所以定位 **UPDATE** 不支持此选项。**WITH CHECK OPTION** 不允许 将创建无法通过视图看到的行 的更新。

定位 **UPDATE** 无法更新 rowid 列。

Sybase IQ 支持重复更新结果集中的 同一行。

另请参见《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqcursorinfo 过程”。

标准

- The range of cursors that can be updated may contain vendor extensions to ISO/ANSI SQL grammar if the ANSI_UPDATE_CONSTRAINTS option is set to OFF.
- Open Client/Open Server 支持使用嵌入式 SQL，SQL Anywhere 支持使用过程和触发器。

权限

必须有所修改的列的 UPDATE 权限。

另请参见

- DECLARE CURSOR 语句 [ESQL] [SP]（第 155 页）
- DELETE 语句（第 163 页）
- DELETE（定位）语句 [ESQL] [SP]（第 165 页）
- UPDATE 语句（第 304 页）

WAITFOR 语句

将对当前连接的处理延迟指定的时间长度或延迟到一个给定时间。

语法

```

DELAY
    WAITFOR {
        time | TIME
        time }
    [ CHECK EVERY
        integer ]
    [ AFTER MESSAGE BREAK ]
  
```

参数

- **Time:** - string

示例

- 示例 1 - 等待 3 秒钟:

```
WAITFOR DELAY '00:00:03'
```

- 示例 2 - 等待 0.5 秒钟 (500 毫秒):

```
WAITFOR DELAY '00:00:00:500'
```

- 示例 3 - 一直等待直到晚上 8 点:

```
WAITFOR TIME '20:00'
```

用法

WAITFOR 语句定期唤醒 (缺省情况下每 5 秒钟一次), 以检查是否语句已取消或消息已接收。如果这两种情况均未发生, 该语句会继续等待。

如果使用 **DELAY**, 处理将暂停特定的时间长度。如果指定 **TIME**, 处理将一直暂停到服务器时间到达指定的时间。

如果当前服务器时间大于指定的时间, 则处理将一直暂停到第二天的这个时间。

WAITFOR 提供了替代以下语句的另一种方法, 对于选择不在数据库中启用 Java 的客户可能很有用:

```
call java.lang.Thread.sleep( <time_to_wait_in_millisecs> )
```

在许多情况下, 使用预定事件比使用 **WAITFOR TIME** 要好, 因为预定事件在它们自己的连接中执行。

CHECK EVERY 子句—此可选子句控制 **WAITFOR** 语句唤醒的频率。缺省情况下, **WAITFOR** 每 5 秒钟唤醒一次。该值以毫秒为单位, 最小值为 250 毫秒。

AFTER MESSAGE BREAK 子句—**WAITFOR** 语句可用于等待来自另一连接的消息。大多数情况下, 当接收到消息时, 将消息转发到执行 **WAITFOR** 语句的应用程序, 同时 **WAITFOR** 语句继续等待。如果指定了 **AFTER MESSAGE BREAK** 子句, 当接收到来自另一连接的消息时, **WAITFOR** 语句完成。未将消息文本转交给应用程序, 但是可通过获取 `MessageReceived` 连接属性的值来访问该消息。

副作用

- 此语句的实现在等待时使用工作线程。这最多使用由 **-gn** 服务器命令行选项指定的线程中的一个线程。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。
- Sybase - Adaptive Server Enterprise 也实现 此语句。

权限

无

另请参见

- CREATE EVENT 语句（第 82 页）

WHENEVER 语句 [ESQL]

在嵌入式 SQL 程序中指定错误处理方式。

语法

```
WHENEVER
{ SQLERROR | SQLWARNING | NOTFOUND }
... { GOTO label | STOP | CONTINUE | C code; }
```

参数

- **label:** - 标识符

示例

- 示例 1 -

```
EXEC SQL WHENEVER NOTFOUND GOTO done;
```

- 示例 2 -

```
EXEC SQL WHENEVER SQLERROR
{
    PrintError( &sqlca );
    return( FALSE );
};
```

用法

WHENEVER 可将此语句放置在嵌入式 SQL C 程序中的任何位置，它不生成任何代码。预处理器在每个连续的 SQL 语句后生成代码。错误操作对 **WHENEVER** 语句源行中的所有嵌入式 SQL 语句一直保持有效，直到发生下一个具有相同错误条件的 **WHENEVER** 语句或者直到源文件结束。

缺省操作为 **CONTINUE**。

WHENEVER 提供 语句是为了在简单程序中方便使用。多数情况下，检查错误的最简单方法是直接检查 SQLCA (SQLCODE) 的 `sqlcode` 字段。在这种情况下，不使用 **WHENEVER**。 **WHENEVER** 语句使预处理器在每个语句后生成 `if(SQLCODE)` 测试。

注意： 错误条件的生效依据是 C 语言源文件的位置，而不是语句的执行时间。

标准

- SQL—ISO/ANSI SQL 语法的供应商扩展。

SQL 语句

- Sybase—受 Open Client/Open Server 支持。

权限

无

WHILE 语句 [T-SQL]

重复执行某语句或复合语句。

语法

```
WHILE expression  
... statement
```

示例

- 示例 1 -

```
WHILE (SELECT AVG(unit_price) FROM Products) < 30  
BEGIN  
    DELETE FROM Products  
    WHERE UnitPrice = MAX(UnitPrice)  
    IF ( SELECT MAX(UnitPrice) FROM Products ) < 50  
        BREAK  
END
```

如果最贵产品的价格低于 \$50，**BREAK** 语句将中断 **WHILE** 循环。否则继续循环，直到平均价格高于 \$30。

用法

除非各语句被组织成位于关键字 **WHILE** 和 **BEGIN** 之间的复合语句，否则 **END** 条件只影响单个 SQL 语句的执行。

BREAK 语句和 **CONTINUE** 语句可用于控制复合语句中的语句执行。**BREAK** 语句终止循环，执行在标记循环结束的 **END** 关键字后重新开始。**CONTINUE** 语句使 **WHILE** 循环重新开始，跳过 **CONTINUE** 后面的所有语句。

标准

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

权限

无

另请参见

- BEGIN……END 语句 (第 49 页)

数据库选项

数据库选项和 Interactive SQL 选项自定义和修改数据库行为。Sybase IQ 数据库选项分为三类：常规、Transact-SQL 兼容性和 Interactive SQL。

数据库选项简介

数据库选项控制数据库行为的诸多方面，包括兼容性、错误处理和并发。

例如，可以将数据库选项用于如下目的：

- 兼容性 – 使您可以控制 Sybase IQ 数据库操作与 Adaptive Server Enterprise 类似的程度，以及不符合 SQL92 的 SQL 是否生成错误。
- 错误处理 – 使您可以控制出现错误（如以零为除数的错误或溢出错误）时的操作。
- 并发和事务 – 使您可以使用选项控制并发程度以及 COMMIT 行为的细节。

您可以使用 **SET OPTION** 语句来设置选项，所用的常规语法如下：

```
SET [ EXISTING ] [ TEMPORARY ] OPTION  
... [ userid. | PUBLIC. ]option-name = [ option-value ]
```

指定一个用户 ID 或组名将仅为该用户或组设置选项。每个用户都属于 PUBLIC 组。如果未指定用户 ID 或组，则所做的选项更改将应用于发出 **SET OPTION** 语句的当前登录用户 ID。

例如，以下语句会将更改应用于 PUBLIC 用户 ID（所有用户都属于该 ID 代表的用户组）：

```
SET OPTION Public.login_mode = standard
```

注意： 当您为 TEMPORARY 设置选项而不指定用户或组时，新的选项值将仅对发出该语句的当前登录用户 ID 生效，且仅在连接时期内有效。当您为 PUBLIC 组的 TEMPORARY 设置选项时，更改会在数据库运行期间一直保持在原位 — 当数据库关闭时，PUBLIC 组的 TEMPORARY 选项将恢复为其永久值。

当您在不发 **TEMPORARY** 关键字的情况下设置选项时，新的选项值会成为发出该语句的用户或组的永久值。

有关临时和永久选项值的详细信息，请参见“数据库选项的作用域和持续时间”、“临时选项”以及“SET OPTION 语句”。

当设置为字符串时，*option-value* 的最大长度为 127 个字节。

注意： 对于所有接受整数值的数据选项，Sybase IQ 会截去 *option-value* 设置的任何小数部分，只保留整数值。例如，值 3.8 将被截断为 3。

警告！ 不要在读取行时更改选项设置。

另请参见

- 数据库选项的作用域和持续时间（第 317 页）
- 临时选项（第 317 页）
- SET OPTION 语句（第 291 页）

当前选项设置

可以使用 **sp_iqcheckoptions**、**sa_conn_properties**、**SET** 语句、Sybase Central 和 SYSOPTIONS 系统视图获得选项设置的列表或单个选项的值。

- 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。

sp_iqcheckoptions 还将列出已由缺省值更改为其它值的服务器启动选项。

当 DBA 运行 **sp_iqcheckoptions** 时，他或她会查看为所有组 and 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。

sp_iqcheckoptions 存储过程不需要任何参数。在 Interactive SQL 中，运行：

```
sp_iqcheckoptions
```

请参见《参考：构件块、表和过程》。

系统表 DBA.SYSOPTIONDEFAULTS 包含 Sybase IQ 和 SQL Anywhere 选项的所有名称和缺省值。可以查询此表以查看所有选项缺省值。

- 您的连接的当前选项设置作为连接属性的子集提供。您可以使用 **sa_conn_properties** 系统过程列出所有连接属性。

```
call sa_conn_properties
```

- 在 Interactive SQL 中，不带参数的 **SET** 语句可列出选项的当前设置：

```
SET
```

- 在 Sybase Central 中，可以右键单击数据库，然后从子菜单中选择“选项”。
- 查询 SYSOPTIONS 系统视图：

```
SELECT *  
FROM SYSOPTIONS
```

这会显示所有 PUBLIC 值，以及已显式设置的 USER 值。

- 使用 **connection_property** 系统函数可获取个别选项设置。例如，以下语句将返回 Ansinull 选项的值：

```
SELECT connection_property ('Ansinull')
```

数据库选项的作用域和持续时间

可以将选项设置为三种级别的作用域：公共、用户和临时。

临时选项优先于用户设置和公共设置。用户级选项优先于公共设置。如果为当前用户设置用户级选项，也会设置对应的临时选项。

某些选项（如 COMMIT 行为）的作用域是数据库范围。设置这些选项要求具有 DBA 权限。其它选项（如 ISOLATION_LEVEL）也可以只应用于当前连接，而且无需特殊权限。

根据选项的不同，对选项设置进行更改的时间也不同。对全局选项（如 RECOVERY_TIME）的更改在下次启动服务器时进行。一些选项在服务器重新启动后生效：

需要重新启动服务器的数据库选项
CACHE_PARTITIONS
CHECKPOINT_TIME
OS_FILE_CACHE_BUFFERING
OS_FILE_CACHE_BUFFERING_TEMPDB
PREFETCH_BUFFER_LIMIT
PREFETCH_BUFFER_PERCENT
RECOVERY_TIME
SWEEPER_THREADS_PERCENT
WASH_AREA_BUFFERS_PERCENT

只作用于当前连接的选项通常会立即生效。例如，可以在事务中间更改选项设置。

警告！ 在游标打开时更改选项会导致不稳定的结果。例如，在游标打开时更改 DATE_FORMAT 可能不会改变下一行的格式。根据游标的检索方式，可能要经过几行之后，更改才会对用户奏效。

临时选项

将 **TEMPORARY** 关键字添加到 SET OPTION 语句中会改变更改的持续时间。

通常选项更改是永久的：在使用 **SET OPTION** 显式地更改之前，它不会变化。

执行 **SET TEMPORARY OPTION** 语句时，新选项值只对当前连接生效，其有效时间也只是连接的持续时间。

在使用 **SET TEMPORARY OPTION** 设置 PUBLIC 选项时，更改在数据库运行的任何时间都有效。当数据库关闭时，PUBLIC 用户 ID 的 TEMPORARY 选项恢复为其永久值。

为 PUBLIC 用户 ID 临时设置选项可提供更高的安全性。例如，在启用 LOGIN_MODE 选项时，数据库依赖于其所在的系统的登录安全性。临时启用 LOGIN_MODE 选项意味着，对于依赖于 Windows 域的安全性的数据库，如果关闭该数据库并将它复制到本地计算机，它的安全不会受到威胁。在这种情况下，LOGIN_MODE 选项将恢复为其永久值，该值可能是 Standard，即不允许集成登录的模式。

公共选项

更改 PUBLIC 用户 ID 某个选项的值将为尚未设置自己的值的所有用户设置该选项的值。

只有拥有 DBA 权限的用户才有权设置 PUBLIC 用户 ID 的选项。

如果某选项没有 PUBLIC 用户 ID 设置，则无法为单个用户 ID 设置此选项的值。

删除选项设置

忽略 option-value 以从数据库中删除选项设置。

如果忽略 option-value，将从数据库中删除指定的选项设置。如果 option-value 是一个个人选项设置，则它的值会恢复为 PUBLIC 设置。如果删除一个 TEMPORARY 选项，则选项设置会恢复为永久设置。

例如，将 ANSINULL 选项重新设置为其缺省值：

```
SET OPTION ANSINULL =
```

如果在设置选项时错误地键入该选项的名称，该错误名称将保存在 SYSOPTION 表中。可以通过使用选项名称后跟等号但不带值的形式设置选项 PUBLIC，来从 SYSOPTION 表中删除错误键入的名称：

```
SET OPTION PUBLIC.a_mistyped_name=;
```

例如，如果设置选项时错误地键入名称，可以通过在 SYSOPTIONS 视图中进行选择来验证该选项是否已保存：

```
SET OPTION PUBLIC.a_mistyped_name='ON';
SELECT * FROM SYSOPTIONS ORDER BY 2;
```

user_name	选项	设置
PUBLIC	a_mistyped_name	ON
PUBLIC	Abort_On_Error_File	
PUBLIC	Abort_On_Error_Line	0
PUBLIC	Abort_On_Error_Number	0
...		

通过将错误键入的选项设置为无值以删除该选项，然后验证该选项是否已删除：

```
SET OPTION PUBLIC.a_mistyped_name=;  
SELECT * FROM SYSOPTIONS ORDER BY 2;
```

user_name	选项	设置
PUBLIC	Abort_On_Error_File	
PUBLIC	Abort_On_Error_Line	0
PUBLIC	Abort_On_Error_Number	0
...		

如果您删除 PUBLIC 选项，然后再尝试添加 USER 选项，将显示一条错误消息：

```
无法执行语句。  
无效的选项 'chained' -- 不存在 PUBLIC 设置  
SQLCODE=-200?ODBC 3 State="42000"  
Line 1,Column 29
```

为了将 PUBLIC 选项重新设置为缺省值，请显式设置缺省值：

```
SET OPTION PUBLIC.chained ='ON' ;
```

初始选项设置

可以利用存储过程来配置用户的初始选项设置。

与 Sybase IQ 的连接应通过 TDS（表式 数据流）协议（Open Client 与 jConnect™ for JDBC™ 连接）或通过 Sybase IQ 协议（ODBC、嵌入式 SQL）实现。

如果用户既使用 TDS 又使用 Sybase IQ 的特定协议，则可以使用存储过程配置其初始设置。在出厂配置中，Sybase IQ 就是使用这种方法设置 Open Client 连接和 jConnect 连接以反映 Adaptive Server Enterprise 的缺省行为。

初始设置可使用 LOGIN_PROCEDURE 选项进行控制，该选项在执行所有检查验证连接是否有效后调用。LOGIN_PROCEDURE 选项命名了一个存储过程以在用户连接时使用。缺省设置为使用 **sp_login_environment** 系统存储过程。您可以指定不同的存储过程。请参见《参考：构件块、表和过程》。

sp_login_environment 过程将检查是否正在通过 TDS 建立连接。如果是，则它调用 **sp_tsql_environment** 过程，该过程会将当前连接的几个选项设置为新的缺省值。

另请参见

- LOGIN_PROCEDURE 选项（第 395 页）

不建议使用的数据库选项

有关本版本不建议使用的数据库 选项的信息，请参见 《Sybase IQ15.4 新增功能摘要》。

常规数据库选项

常规数据库选项是指除 Transact-SQL 兼容性选项和 Interactive SQL 选项之外的所有选项分类。

注意： Sybase 技术支持部门可能会要求您使用此表未列出的另外一些内部选项。

表 16. 常规数据库特定选项

选项名	允许值	缺省设置
AGGREGATION_PREFERENCE	-3 到 3	0
ALLOW_READ_CLIENT_FILE	ON、OFF	OFF
APPEND_LOAD	ON、OFF	OFF
AUDITING	ON、OFF	OFF
BIT_VECTOR_PINNABLE_CACHE_PERCENT*	0 - 100	40
BLOCKING	OFF	OFF
BT_PREFETCH_MAX_MISS	0 - 1000	2
BT_PREFETCH_SIZE	0 - 100	10
BTREE_PAGE_SPLIT_PAD_PERCENT	0 - 90	50
CACHE_PARTITIONS	2 的乘方 (0 到 64 次)	0
CHECKPOINT_TIME	分钟数	60
CIS_ROWSET_SIZE	integer	50
CONVERSION_MODE	0, 1	0
CONVERT_VARCHAR_TO_1242	ON、OFF	OFF
COOPERATIVE_COMMIT_TIMEOUT	integer	250

选项名	允许值	缺省设置
COOPERATIVE_COMMITS	ON、OFF	ON
CURSOR_WINDOW_ROWS	20 - 100000	200
DATE_FIRST_DAY_OF_WEEK	0 - 6	0
DATE_FORMAT	string	'YYYY-MM-DD'
DATE_ORDER	'YMD'、'DMY'、'MDY'	'YMD'
DBCC_LOG_PROGRESS	ON、OFF	OFF
DBCC_PINNABLE_CACHE_PERCENT	0 - 100	50
DEBUG_MESSAGES	ON、OFF	OFF
DEFAULT_DBSPACE	string	" (空字符串)
DEFAULT_DISK_STRIPING	ON、OFF	ON
DEDICATED_TASK	ON、OFF	OFF
DEFAULT_HAVING_SELECTIVITY_PPM	0 - 1000000	0
DEFAULT_KB_PER_STRIPE	1 - 最大 unsigned bigint	1
DEFAULT_LIKE_MATCH_SELECTIVITY_PPM	0 - 1000000	150000
DEFAULT_LIKE_RANGE_SELECTIVITY_PPM	1 - 1000000	150000
DELAYED_COMMIT_TIMEOUT	integer	500
DELAYED_COMMITS	OFF	OFF
DISABLE_RI_CHECK	ON、OFF	OFF
EARLY_PREDICATE_EXECUTION	ON、OFF	ON
ENABLE_LOB_VARIABLES	ON、OFF	OFF
EXTENDED_JOIN_SYNTAX	ON、OFF	ON
FORCE_DROP	ON、OFF	OFF
FORCE_NO_SCROLL_CURSORS	ON、OFF	OFF
FORCE_UPDATABLE_CURSORS	ON、OFF	OFF
FP_LOOKUP_SIZE	1 MB - 4096 MB	16 MB

选项名	允许值	缺省设置
FP_LOOKUP_SIZE_PPM	1 - 1000000	2500
FP_PREDICATE_WORKUNIT_PAGES	integer	200
FP_PREFETCH_SIZE	0 - 100	10
FPL_EXPRESSION_MEMORY_KB	0 - 20000	1024
GARRAY_FILL_FACTOR_PERCENT	0 - 1000	25
GARRAY_INSERT_PREFETCH_SIZE	0 - 100	3
GARRAY_PAGE_SPLIT_PAD_PERCENT	0-100	25
GARRAY_RO_PREFETCH_SIZE	0 - 100	10
HASH_PINNABLE_CACHE_PERCENT*	0 - 100	20
HASH_THRASHING_PERCENT	0 - 100	10
HG_DELETE_METHOD	0 - 3	0
HG_SEARCH_RANGE	integer	10
HTTP_SESSION_TIMEOUT	整数 (1 - 525600)	30
IDENTITY_ENFORCE_UNIQUENESS	ON、OFF	OFF
IDENTITY_INSERT	string	" (空字符串)
INDEX_ADVISOR	ON、OFF	OFF
INDEX_PREFERENCE	-10 - 10	0
INFER_SUBQUERY_PREDICATES	ON、OFF	ON
IN_SUBQUERY_PREFERENCE	-3 - 3	0
IQGOVERN_MAX_PRIORITY	1 - 3	2
IQGOVERN_PRIORITY	1 - 3	2
IQGOVERN_PRIORITY_TIME	1 - 1000000 秒	0 (禁用)
ISOLATION_LEVEL	0, 1, 2, 3	0
JAVA_LOCATION	string	" (空字符串)
JAVA_VM_OPTIONS	string	" (空字符串)
JOIN_EXPANSION_FACTOR	0 - 100	30

选项名	允许值	缺省设置
JOIN_OPTIMIZATION	ON、OFF	ON
JOIN_PREFERENCE	-7 - 7	0
JOIN_SIMPLIFICATION_THRESHOLD	1 - 64	15
LARGE_DOUBLES_ACCUMULATOR	ON、OFF	OFF
LF_BITMAP_CACHE_KB	1 - 8	4
LOAD_ZEROLENGTH_ASNULL	ON、OFF	OFF
LOCKED	ON、OFF	OFF
LOG_CONNECT	ON、OFF	ON
LOG_CURSOR_OPERATIONS	ON、OFF	OFF
LOGIN_MODE	STANDARD、MIXED、INTEGRATED	STANDARD
LOGIN_PROCEDURE	string	sp_login_environment
MAIN_RESERVED_DBSPACE_MB	大于等于 200 的整数（以 MB 为单位）	200
MAX_CARTESIAN_RESULT	integer	100000000
MAX_CLIENT_NUMERIC_PRECISION	0 - 126	0
MAX_CLIENT_NUMERIC_SCALE	0 - 126	0
MAX_CONNECTIONS	0 - 2147483647	无限制
MAX_CUBE_RESULT	0 - 4294967295	10000000
MAX_CURSOR_COUNT	integer	50
MAX_DAYS_SINCE_LOGIN	0 - 2147483647	无限制
MAX_FAILED_LOGIN_ATTEMPTS	0 - 2147483647	无限制
MAX_HASH_ROWS	小于等于 4294967295 的整数	2500000
MAX_IQ_THREADS_PER_CONNECTION	3 - 10000	144
MAX_IQ_THREADS_PER_TEAM	1 - 10000	144

选项名	允许值	缺省设置
MAX_JOIN_ENUMERATION	1 - 64	15
MAX_NON_DBA_CONNECTIONS	0 - 2147483647	无限制
MAX_PREFIX_PER_CONTAINS_PHRASE	0 - 300	1
MAX_QUERY_PARALLELISM	integer	64
MAX_QUERY_TIME	0 - $2^{32} - 1$	0 (禁用)
MAX_STATEMENT_COUNT	integer	100
MAX_TEMP_SPACE_PER_CONNECTION	integer	0
MAX_WARNINGS	integer	$2^{48} - 1$
MINIMIZE_STORAGE	ON、OFF	OFF
MIN_PASSWORD_LENGTH	大于等于 0 的整数	0 个字符
MONITOR_OUTPUT_DIRECTORY	string	<i>database directory</i>
NOEXEC	ON、OFF	OFF
NON_ANSI_NULL_VARCHAR	ON、OFF	OFF
NOTIFY_MODULUS	integer	100000
ODBC_DISTINGUISH_CHAR_AND_VARCHAR	ON、OFF	OFF
ON_CHARSET_CONVERSION_FAILURE	string	IGNORE
OS_FILE_CACHE_BUFFERING	ON、OFF	OFF
PASSWORD_GRACE_TIME	0 - 2147483647	0
PASSWORD_EXPIRY_ON_NEXT_LOGIN	ON、OFF	OFF
PASSWORD_LIFE_TIME	0 - 2147483647	无限制
POST_LOGIN_PROCEDURE	string	dbo.sa_post_login_procedure
PRECISION	126	126

选项名	允许值	缺省设置
PREFETCH	ON、OFF	ON
PREFETCH_BUFFER_LIMIT	integer	0
PREFETCH_BUFFER_PERCENT	0 - 100	40
PREFETCH_GARRAY_PERCENT	0 - 100	60
PREFETCH_SORT_PERCENT	0 - 100	20
PRESERVE_SOURCE_FORMAT	ON、OFF	ON
QUERY_DETAIL	ON、OFF	OFF
QUERY_NAME	string	" (空字符串)
QUERY_PLAN	ON、OFF	ON
QUERY_PLAN_AFTER_RUN	ON、OFF	OFF
QUERY_PLAN_AS_HTML	ON、OFF	OFF
QUERY_PLAN_AS_HTML_DIRECTO- RY	string	" (空字符串)
QUERY_PLAN_TEXT_ACCESS	ON、OFF	OFF
QUERY_PLAN_TEXT_CACHING	ON、OFF	OFF
QUERY_ROWS_RETURNED_LIMIT	integer	0
QUERY_TEMP_SPACE_LIMIT	integer	0
QUERY_TIMING	ON、OFF	OFF
RECOVERY_TIME	分钟数	2
RETURN_DATE_TIME_AS_STRING	ON、OFF	OFF
ROW_COUNT	integer	0
SCALE	0 - 126	38
SIGNIFICANTDIGITSFORDOU- BLEEQUALITY	0 - 15	0
SORT_COLLATION	Internal、collation_name 或 collation_id	Internal
SORT_PINNABLE_CACHE_PERCENT*	0 - 100	20
SUBQUERY_CACHING_PREFERENCE	-3 - 3	0

选项名	允许值	缺省设置
SUBQUERY_FLATTENING_PERCENT	0, 1 - 2^{32} - 1	100
SUBQUERY_FLATTENING_PREFERENCE	-3 - 3	0
SUBQUERY_PLACEMENT_PREFERENCE	-1 - 1	0
SUPPRESS_TDS_DEBUGGING	ON、OFF	OFF
SWEEPER_THREADS_PERCENT	1 到 40	10
TDS_EMPTY_STRING_IS_NULL	ON、OFF	OFF
TEMP_DISK_PER_STRIPE	大于 0 的整数 (以 KB 为单位)	1
TEMP_EXTRACT_APPEND	ON、OFF	OFF
TEMP_EXTRACT_BINARY	ON、OFF	OFF
TEMP_EXTRACT_COLUMN_DELIMITER	string	' '
TEMP_EXTRACT_DIRECTORY	string	" (空字符串)
TEMP_EXTRACT_ESCAPE_QUOTES	ON、OFF	OFF
TEMP_EXTRACT_NAME1 - TEMP_EXTRACT_NAME8	string	" (空字符串)
TEMP_EXTRACT_NULL_AS_EMPTY	ON、OFF	OFF
TEMP_EXTRACT_NULL_AS_ZERO	ON、OFF	OFF
TEMP_EXTRACT_QUOTE	string	" (空字符串)
TEMP_EXTRACT_QUOTES	ON、OFF	OFF
TEMP_EXTRACT_QUOTES_ALL	ON、OFF	OFF
TEMP_EXTRACT_ROW_DELIMITER	string	" (空字符串)
TEMP_EXTRACT_SIZE1 - TEMP_EXTRACT_SIZE8	AIX & HP-UX: 0 - 64GB Solaris: & Linux 0 - 512GB Windows: 0 - 128GB	0
TEMP_EXTRACT_SWAP	ON、OFF	OFF
TEMP_RESERVED_DBSPACE_MB	大于等于 200 的整数 (以 MB 为单位)	200

选项名	允许值	缺省设置
TEMP_SPACE_LIMIT_CHECK	ON、OFF	ON
TEXT_DELETE_METHOD	0 - 2	0
TIME_FORMAT	string	'HH:NN:SS.SSS'
TIMESTAMP_FORMAT	string	'YYYY-MM-DD HH:NN:SS.SSS'
TOP_NSORT_CUTOFF_PAGES	1 - 1000	1
TRIM_PARTIAL_MBC	ON、OFF	OFF
USER_RESOURCE_RESERVATION	integer	1
VERIFY_PASSWORD_FUNCTION	string	" (空字符串)
WASH_AREA_BUFFERS_PERCENT	1 - 100	20
WAIT_FOR_COMMIT	ON、OFF	OFF
WD_DELETE_METHOD	0 - 3	0

另请参见

- Transact-SQL 兼容性选项 (第 327 页)
- Interactive SQL 选项 (第 330 页)
- 按字母顺序排列的选项列表 (第 332 页)

数据提取选项

使用数据提取功能，可以通过将 **SELECT** 语句的输出从标准接口重定向到一个或多个磁盘文件或命名管道，从数据库中提取数据。

TEMP_EXTRACT_... 数据库选项用于控制数据提取功能。

请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”>“提取选项”。

Transact-SQL 兼容性选项

Transact-SQL 兼容性选项使 Sybase IQ 行为能够与 Adaptive Server Enterprise 兼容，或既支持旧行为又允许使用 ISO SQL92 行为。

为实现与 Adaptive Server Enterprise 的进一步兼容，可使用 Transact-SQL **SET** 语句替代 Sybase IQ**SET OPTION** 语句在当前连接期间设置部分选项。

表 17. Transact-SQL 兼容性选项

选项	允许值	缺省设置
ALLOW_NULLS_BY_DEFAULT	ON、OFF	ON
ANSI_BLANKS*	ON、OFF	OFF
ANSI_CLOSE_CURSORS_ON_ROLLBACK	ON	ON
ANSI_INTEGER_OVERFLOW*		
ANSI_PERMISSIONS	ON、OFF	ON
ANSINULL	ON、OFF	ON
ANSI_SUBSTRING	ON、OFF	ON
ANSI_UPDATE_CONSTRAINTS	OFF、CURSORS、STRICT	CURSORS
ASE_BINARY_DISPLAY	ON、OFF	OFF
ASE_FUNCTION_BEHAVIOR	ON、OFF	OFF
CHAINED	ON、OFF	ON
CLOSE_ON_ENDTRANS	ON	ON
CONTINUE_AFTER_RAISERROR	ON、OFF	ON
CONVERSION_ERROR	ON、OFF	ON
DIVIDE_BY_ZERO_ERROR	ON、OFF	ON
ESCAPE_CHARACTER*	保留	保留
FIRE_TRIGGERS*	ON、OFF	ON
NEAREST_CENTURY	0 - 100	50
NON_KEYWORDS	以逗号分隔的关键字列表	不关闭任何关键字
ON_TSQL_ERROR	STOP、CONTINUE、CONDITIONAL	CONDITIONAL
QUERY_PLAN_ON_OPEN*		
QUOTED_IDENTIFIER	ON、OFF	ON
RI_TRIGGER_TIME*		

选项	允许值	缺省设置
SQL_FLAGGER_ERROR_LEVEL	E、I、F、W、OFF、SQL:1992/Entry、SQL:1992/Intermediate、SQL:1992/Full、SQL:1999/Core、SQL:1999/Package、SQL:2003/Core、SQL:2003/Package	OFF
SQL_FLAGGER_WARNING_LEVEL	E、I、F、W、OFF、SQL:1992/Entry、SQL:1992/Intermediate、SQL:1992/Full、SQL:1999/Core、SQL:1999/Package、SQL:2003/Core、SQL:2003/Package	OFF
STRING_RTRUNCATION	ON、OFF	ON
TEXTSIZE*		
TSQL_HEX_CONSTANT*		
TSQL_VARIABLES	ON、OFF	OFF

注意： 选项名称旁边的星号 (*) 表示 Sybase IQ 目前不支持该选项。

另请参见

- 常规数据库选项（第 320 页）
- Interactive SQL 选项（第 330 页）
- 按字母顺序排列的选项列表（第 332 页）
- SET 语句 [T-SQL]（第 287 页）

Transact-SQL 为实现 Adaptive Server Enterprise 兼容性的选项设置

有一些选项的缺省设置与 Adaptive Server Enterprise 的缺省设置不同。为确保兼容行为，应显式设置这些选项。

在使用 Open Client 或 JDBC 接口进行连接时，为使当前连接与 Adaptive Server Enterprise 兼容，会显式设置某些选项设置。

有关如何进行设置的信息，请参见《参考：构件块、表和过程》。

表 18. 为实现 ASE 兼容性需要显式设置的 Transact-SQL 选项

选项	ASE 兼容设置
ALLOW_NULLS_BY_DEFAULT	OFF
ANSINULL	OFF
CHAINED	OFF

选项	ASE 兼容设置
CONTINUE_AFTER_RAISERROR	ON
DATE_FORMAT	YYYY-MM-DD
DATE_ORDER	MDY
ESCAPE_CHARACTER	OFF
ISOLATION_LEVEL	1
ON_TSQL_ERROR	CONDITIONAL
QUOTED_IDENTIFIER	OFF
TIME_FORMAT	HH:NN:SS.SSS
TIMESTAMP_FORMAT	YYYY-MM-DD HH:NN:SS.SSS
TSQL_VARIABLES	OFF

Interactive SQL 选项

Interactive SQL 选项可用于更改 Interactive SQL 与数据库的交互方式。

语法 1

```
SET [ TEMPORARY ] OPTION
... [ userid. | PUBLIC. ]option-name = [ option-value ]
```

语法 2

```
SET
PERMANENT
```

语法 3

```
SET
```

参数

```
userid:
identifier, string or host-variable
```

```
option-name:
identifier, string or host-variable

option-value:
host-variable (indicator allowed), string, identifier,
or number
```

描述
包含 **TEMPORARY** 关键字的语法 1 不能在复合语句的 **BEGIN** 和 **END** 关键字之间使用。

语法 2 **SET PERMANENT** 在 SYSOPTIONS 系统表中存储所有当前的 Interactive SQL 选项。每次针对当前用户 ID 启动 Interactive SQL 时，都会自动确定这些设置。

语法 3 用于显示所有当前选项设置。如果存在为 Interactive SQL 或数据库服务器设置的临时选项，则会显示这些临时选项；否则显示永久选项设置。

表 19. Interactive SQL 选项

选项	允许值	缺省设置
DEFAULT_ISQL_ENCODING	标识符或字符串	空字符串（使用系统代码页）
NULLS*	字符串	NULL
ON_ERROR	STOP、CONTINUE、PROMPT、EXIT、NOTIFY_CONTINUE、NOTIFY_STOP、NOTIFY_EXIT	PROMPT
OUTPUT_FORMAT*	ASCII、DBASEII、DBASEIII、EXCEL、FIXED、FOXPRO、HTML、LOTUS、SQL、XML	ASCII
OUTPUT_LENGTH*	非负整数	0（没有截断）
OUTPUT_NULLS*	字符串	'NULL'
STATISTICS*	0, 3, 4, 5, 6	3
TRUNCATION_LENGTH*	整数	256

注意： 选项名称旁边的星号 (*) 表示 Sybase IQ 目前不支持该选项。

另请参见

- 常规数据库选项（第 320 页）

- Transact-SQL 兼容性选项 (第 327 页)
- 按字母顺序排列的选项列表 (第 332 页)

按字母顺序排列的选项列表

常规、Transact-SQL 兼容性和 Interactive SQL 数据库选项说明。某些选项名称的后面有一个类指示符 (括在中括号内)。

数据库选项类指示符是:

- - 该选项用于更改 Interactive SQL 与数据库交互的方式。[TSQL]
- [TSQL] - 该选项允许 Sybase IQ 行为变得与 Adaptive Server Enterprise 兼容, 或既支持旧行为又允许 ISO SQL92 行为。

另请参见

- 数据库选项简介 (第 315 页)
- 常规数据库选项 (第 320 页)
- Transact-SQL 兼容性选项 (第 327 页)
- Interactive SQL 选项 (第 330 页)

AGGREGATION_PREFERENCE 选项

控制用于处理集合的算法的选择。

允许值

-3 到 3

默认值

0

范围

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

描述

对于查询中的集合 (**GROUP BY**、**DISTINCT**、**SET** 函数), Sybase IQ 优化程序可以选择使用多种算法 对该集合进行相应处理。AGGREGATION_PREFERENCE 使您可以覆盖优化程序 在选择算法时的开销决策。该选项不会覆盖用于确定算法 在查询引擎内是否合法的内部规则。

此选项通常用于内部测试, 以及优化程序处理不好的手动调优查询。只有经验丰富的 DBA 才能使用。如果需要设置 AGGREGATION_PREFERENCE, 请通知 Sybase 技术支持部门, 因为设置此选项可能意味着需要更改优化程序。

表 20. AGGREGATION_PREFERENCE 的值

值	操作
0	让优化程序选择
1	选用已进行排序的集合
2	选用采用 IQ 索引的集合
3	选用具有散列的集合
-1	不选用已进行排序的集合
-2	不选用采用 IQ 索引的集合
-3	不选用具有散列的集合

ALLOW_NULLS_BY_DEFAULT 选项 [TSQL]

控制是否允许创建时未指定是 NULL 还是 NOT NULL 的新列包含空值。

允许值

ON、OFF

默认值

ON

OFF (对于 Open Client 和 JDBC 连接)

描述

包括 ALLOW_NULLS_BY_DEFAULT 选项是为了与 Transact-SQL 兼容。

ANSI_CLOSE_CURSORS_ON_ROLLBACK 选项 [TSQL]

控制执行 ROLLBACK 时是否关闭使用 WITH HOLD 打开的游标。

允许值

ON

默认值

ON

描述

ANSI SQL/3 标准要求事务回退时关闭所有游标。此选项强制执行该行为，而且不能更改。CLOSE_ON_ENDTRANS 选项将覆盖此选项。

ANSI_PERMISSIONS 选项 [TSQL]

为 DELETE 和 UPDATE 语句控制权限检查。

允许值

ON、OFF

默认值

ON

描述

在 ANSI_PERMISSIONS 为 ON 的情况下，将检查 **DELETE** 和 **UPDATE** 语句的 SQL92 权限要求。 Adaptive Server Enterprise 中的缺省值为 OFF。下表简要介绍了其中的差别：

表 21. ANSI_PERMISSIONS 选项的作用

SQL 语句	ANSI_PERMISSIONS 为 OFF 时所需的权限	ANSI_PERMISSIONS 为 ON 时所需的权限
UPDATE	正在对其设置值的列上的 UPDATE 权限	正在对其设置值的列上的 UPDATE 权限。 出现在 WHERE 子句中的所有列上的 SELECT 权限。 SET 子句右侧的所有列上的 SELECT 权限。
DELETE	对表的 DELETE 权限	对表的 DELETE 权限。 SELECT permission on all columns appearing in the WHERE clause.

只能为 PUBLIC 组设置 ANSI_PERMISSIONS 选项。不允许使用专用设置。

ANSINULL 选项 [TSQL]

控制对将 = 和 != 与 NULL 结合使用的解释。

允许值

ON、OFF

默认值

ON

描述

在 ANSINULL 为 ON 的情况下，使用 “=” 或 “!=” 与 NULL 进行比较的结果是未知结果。其中包括其它运算（如 CASE）隐含的比较的结果。

将 ANSINULL 设置为 OFF 可允许与 NULL 的比较产生已知结果，以实现与 Adaptive Server Enterprise 的兼容。

注意：与 SQL Anywhere 不同，对于包含空值的列中集合函数，Sybase IQ 不会生成警告“集合函数中的空值已消除” (SQLSTATE=01003)。

ANSI_SUBSTRING 选项 [TSQL]

控制在为 **start** 或 **length** 参数提供负值时 **SUBSTRING (SUBSTR)** 函数的行为。

允许值

ON、OFF

默认值

ON

描述

ANSI_SUBSTRING 选项设置为 On 时，**SUBSTRING** 函数的行为相当于 ANSI/ISO SQL/2003 行为。对零或负的 **start** 偏移量的处理方式就好像用“非字符”对字符串左侧进行了填充，而且在 **length** 为负值时会给出错误。

当此选项设置为 OFF 时，**SUBSTRING** 函数的行为与先前版本的 Sybase IQ 的行为相同：负的起始偏移表示从字符串末尾开始的偏移，而负的长度表示所需的子串在起始偏移左侧结束的字符长度。使用为 0 的起始偏移量等同于起始偏移量为 1。

避免在 **SUBSTRING** 函数中使用非正起始偏移量或负的 **length**。请尽可能改用 **LEFT** 或 **RIGHT** 函数。

示例

以下示例显示根据 **SUBSTRING** 选项的设置，**ANSI_SUBSTRING** 函数返回值的不同：

```
SUBSTRING( 'abcdefgh',-2,4 );
ansi_substring = Off ==> 'gh'
// substring starts at second-last character
ansi_substring = On  ==> 'a'
// takes the first 4 characters of
// ???abcdefgh and discards all ?
```

```
SUBSTRING( 'abcdefgh',4,-2 );
ansi_substring = Off ==> 'cd'
ansi_substring = On  ==> value -2 out of range
for destination
```

```
SUBSTRING( 'abcdefgh',0,4 );
ansi_substring = Off ==> 'abcd'
ansi_substring = On  ==> 'abc'
```

ANSI_UPDATE_CONSTRAINTS 选项

控制允许的更新的范围。

允许值

OFF、Cursors、STRICT

*默认值***CURSORS***描述*

Sybase IQ 提供了几种扩展，它们允许进行 ANSI SQL 标准所不允许的更新。这些扩展为执行更新提供了高效且功能强大的机制。然而，在某些情况下，它们会导致非直觉行为。如果在设计用户应用程序时未预计使用这些扩展行为，此行为可能会导致异常（如丢失更新）。

ANSI_UPDATE_CONSTRAINTS 控制是否将更新限制在 SQL92 标准允许的范围内。

如果将此选项设置为 STRICT，将无法进行以下更新：

- 对包含 JOINS 的游标的更新
- 对出现在 ORDER BY 子句中的列的更新
- 不允许在 UPDATE 语句中出现 FROM 子句。

如果将此选项设置为 CURSORS，则相同的限制会生效，但只用于游标。如果游标不是使用 FOR UPDATE 或 FOR READ ONLY 打开的，数据库服务器将根据 SQL92 标准确定是否允许更新。

如果 ANSI_UPDATE_CONSTRAINTS 设置为 CURSORS 或 STRICT，则包含 ORDER BY 子句的游标缺省为 FOR READ ONLY；否则，它们缺省为 FOR UPDATE。

示例

根据 ANSI_UPDATE_CONSTRAINTS 设置的不同，以下代码会产生不同的作用：

```
CREATE TABLE mmg (a CHAR(3));
CREATE TABLE mmg1 (b CHAR(3));

INSERT INTO mmg VALUES ('001');
INSERT INTO mmg VALUES ('002');
INSERT INTO mmg VALUES ('003');
INSERT INTO mmg1 VALUES ('003');
SELECT * FROM mmg;
SELECT * FROM mmg1;
```

选项 1：将 ANSI_UPDATE_CONSTRAINTS 设置为 STRICT：

```
SET OPTION public.Ansi_update_constraints = 'strict';
DELETE MMG FROM MMG1 WHERE A=B;
```

这样会产生错误，指示不允许进行所尝试的更新操作。

选项 2：将 ANSI_UPDATE_CONSTRAINTS 设置为 CURSORS 或 OFF：

```
SET OPTION public.Ansi_update_constraints = 'CURSORS'; // or 'OFF'
DELETE mmg FROM mmg1 WHERE A=B;
```

在这种情况下，删除操作可以完成而不会出现错误。

另请参见

- UPDATE 语句（第 304 页）

ALLOW_READ_CLIENT_FILE 选项

启用客户端数据传输。

请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “数据库选项” > “数据库选项简介” > “按字母顺序排列的选项列表” > “allow_read_client_file 选项 [数据库]”。

注意： 这些参考指向 SQL Anywhere 文档。

APPEND_LOAD 选项

帮助减少版本控制页的空间使用情况。

允许值

ON、OFF

默认值

OFF

范围

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

描述

APPEND_LOAD 选项应用于 **LOAD**、**INSERT...SELECT** 和 **INSERT...VALUES** 语句并将在下次执行该语句时生效。

当 APPEND_LOAD 选项为 OFF 时，Sybase IQ 将重新使用已删除行的行 ID。将此选项设置为 ON 可将新数据附加到表的末尾。

APPEND_LOAD 对于分区表和未分区的表，行为有所不同。分区表中的每个分区都指派有行 ID 范围。对于分区表，当 APPEND_LOAD 为 ON 时，新行将附加到相应分区的末尾。如果 APPEND_LOAD 为 OFF，装载则重新使用已删除的行中的第一个可用行 ID 和空格。

对于未分区的表，当 APPEND_LOAD 为 ON 时，新行将添加到位于表行末尾的最大行 ID 后面。当 APPEND_LOAD 为 OFF 时，装载操作将重新使用已删除行的 ID。对于非分区表，还可以使用 **LOAD** 或 **INSERTSTART ROW ID** 子句指定插入的开始行，以控制插入行的位置。

ASE_BINARY_DISPLAY 选项

指定 Sybase IQ 二进制列 的显示与 Adaptive Server Enterprise 二进制 列的显示一致。

允许值

ON、OFF

默认值

OFF

范围

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

描述

ASE_BINARY_DISPLAY 会使 **SELECT** 语句的输出受到影响。

此选项只影响 IQ 存储中的列。而 不会影响变量、目录存储列或 SQL Anywhere 列。当此选项为 ON 时，Sybase IQ 将以可读的 ASCII 格式显示列；例如 0x1234567890abcdef。当此选项为 OFF 时，Sybase IQ 会将列显示为二进制输出（而非 ASCII）。

将 **ASE_BINARY_DISPLAY** 设置为 OFF 可支持对二进制数据库类型执行批量复制操作。Sybase IQ 支持通过 **LOAD TABLE USING CLIENT FILE** 语句批量 装载远程数据。

另请参见

- **LOAD TABLE** 语句（第 223 页）

ASE_FUNCTION_BEHAVIOR 选项

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。

允许值

ON、OFF

默认值

OFF

范围

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

描述

在 **ASE_BEHAVIOR_FUNCTION** 处于 ON 的情况下，一些 Sybase IQ 数据类型转换函数（包括 **HEXTOINT** 和 **INTTOHEX**）返回的输出与 Adaptive Server Enterprise 函数的输

出一致。ASE 输出与 Sybase IQ 输出在格式和长度方面存在差异，ASE 主要使用有符号 32 位作为缺省设置，Sybase IQ 则主要使用无符号 64 位作为缺省设置。

Sybase IQ 不支持 64 位整数，因为 ASE 没有 64 位整数的数据类型。

有关 **INTTOHEX** 和 **HEXTOINT** 函数行为的详细信息，请参见《参考：构件块、表和过程》。

示例

在此示例中，根据 ASE_FUNCTION_BEHAVIOR 选项是 ON 还是 OFF，**HEXTOINT** 函数将返回不同的值。

HEXTOINT 函数在 ASE_FUNCTION_BEHAVIOR 设置为 OFF 的情况下返回 4294967287:

```
select hextoint( 'ffffffff7' ) from iq_dummy
```

HEXTOINT 函数在 ASE_FUNCTION_BEHAVIOR 设置为 ON 的情况下返回 -9:

```
select hextoint( 'ffffffff7' ) from iq_dummy
```

另请参见

- **CONVERSION_ERROR** 选项 [TSQL] (第 346 页)

AUDITING 选项 [database]

在数据库中启用和禁用审计。

允许值

ON、OFF

默认值

OFF

范围

只能为 PUBLIC 组设置。设置将立即生效。需要具有 DBA 权限。

描述

此选项用于打开和关闭审计功能。

审计是事务日志中有关数据库中许多事件的详细信息的记录。审计可提供一些安全功能，但会降低一些性能。打开数据库的审计功能时，无法停止使用事务日志。必须关闭审计功能才能关闭事务日志。无法以只读模式启动打开审计功能的数据库。

为使 **AUDITING** 选项起作用，必须将审计选项设置为 ON，并使用

sa_enable_auditing_type 系统过程指定要审计的信息的类型。如果满足下列任一条件，则不会进行审计：

- AUDITING 选项设置为 OFF
- 审计选项已禁用

如果将 AUDITING 选项设置为 ON 但不指定审计选项，则会记录所有类型的审计信息。或者，您可以使用 **sa_enable_auditing_type** 记录以下任意组合：权限检查、连接尝试、DDL 语句、公共选项和触发器。请参见《参考：构件块、表和过程》。

BIT_VECTOR_PINNABLE_CACHE_PERCENT 选项

持久位矢量对象在用户临时内存中可以固定的最大百分比。

允许值

0 - 100

默认值

40

范围

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

描述

BIT_VECTOR_PINNABLE_CACHE_PERCENT 用于控制用户临时内存分配中任一持久位矢量对象可以固定的内存百分比。其缺省值为 40%，通常用户不应更改该值。

此选项主要供 Sybase 技术支持部门使用。如果更改

BIT_VECTOR_PINNABLE_CACHE_PERCENT 的值，应当谨慎行事；首先要分析将会对多种查询产生的影响。

另请参见

- HASH_PINNABLE_CACHE_PERCENT 选项（第 375 页）
- SORT_PINNABLE_CACHE_PERCENT 选项（第 432 页）

BLOCKING 选项

控制响应锁定冲突的行为。

允许值

OFF

默认值

OFF

范围

可针对单个连接或 PUBLIC 组进行设置。设置将立即生效。

描述

当 **BLOCKING** 设置为 **OFF** 时，如果某个事务尝试执行写入操作但被另一事务的读取锁阻止，则会收到一条错误。

BT_PREFETCH_MAX_MISS 选项

控制 Sybase IQ 确定是否继续针对给定查询 预取 B-tree 页的方式。

允许值

0 - 1000

默认值

2

范围

可针对单个连接或 **PUBLIC** 组进行设置。设置将立即生效。

描述

只有在 Sybase 技术支持部门的指导下执行此操作时才使用。对于使用 **HG** (**High_Group**) 索引的查询，Sybase IQ 将按顺序预取 **B-tree** 页，直到确定不再需要预取为止。对于某些查询，此选项可能会提早关闭预取功能。增大 **BT_PREFETCH_MAX_MISS** 的值将使 Sybase IQ 更有可能继续进行预取，但与此同时，也可能会不必要地增加 I/O。

如果使用 **HG** 索引的查询的运行速度比预期慢很多，请尝试逐步增大 **BT_PREFETCH_MAX_MISS** 的值。

可尝试使用不同的设置以找出可实现最佳性能的设置。对于大多数查询，实用的设置范围为从 1 到 10。

另请参见

- **BT_PREFETCH_SIZE** 选项（第 341 页）
- **PREFETCH_BUFFER_LIMIT** 选项（第 417 页）

BT_PREFETCH_SIZE 选项

限制用于 **High_Group B-tree** 的预先读取缓冲区的大小。

允许值

0 - 100。设置为 0 将禁用 **B-tree** 预取。

默认值

10

范围

只能针对单个用户进行设置。设置将立即生效。

描述

对于任何对 **High_Group** 索引的顺序访问（如 **INSERT**、大型 **DELETE**、范围谓词和 **DBCC**（数据库一致性检查程序）），缺省情况下均已激活 **B-tree** 预取。

BT_PREFETCH_SIZE 可限制用于 **B-tree** 页的预先读取缓冲区的大小。减少预取大小可节省缓冲区空间，但有时也会降低性能。增加预取大小则可能带来边际收益。对于非唯一的 **High_Group** 索引，此选项应与选项 **PREFETCH_GARRAY_PERCENT**、**GARRAY_INSERT_PREFETCH_SIZE** 和 **GARRAY_RO_PREFETCH_SIZE** 结合使用。

另请参见

- **GARRAY_INSERT_PREFETCH_SIZE** 选项（第 373 页）
- **GARRAY_RO_PREFETCH_SIZE** 选项（第 374 页）
- **PREFETCH_GARRAY_PERCENT** 选项（第 418 页）

BTREE_PAGE_SPLIT_PAD_PERCENT 选项

确定在对 **B-Tree** 结构进行页面拆分的过程中每页的填充因子。

允许值

0 - 90

默认值

50

范围

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

描述

HG、**LF**、**DT**、**TIME** 和 **DTTM** 索引使用 **B-Tree** 结构。**B-Tree** 页的拆分会尝试将指定的百分比留空，以避免当新键插入索引时进行拆分。

索引在页级保留存储空间，插入附加数据时可将这些存储空间分配给新键。保留空间会消耗额外的磁盘空间，但对增量插入的性能有所帮助。如果计划进行增量插入，而且这些新行不包含索引中已经存在的值，那么为

GARRAY_PAGE_SPLIT_PAD_PERCENT 选项设置一个非零值可提高增量插入的性能。

如果未计划逐步更新索引，则可以减少此选项的值以节省磁盘空间。

另请参见

- **GARRAY_FILL_FACTOR_PERCENT** 选项（第 373 页）

- GARRAY_PAGE_SPLIT_PAD_PERCENT 选项（第 374 页）

CACHE_PARTITIONS 选项

设置要用于主缓冲区高速缓存和临时缓冲区高速缓存的分区数。

允许值

0, 1, 2, 4, 8, 16, 32, 64

0: Sybase IQ 自动将分区数 计算为 $\text{number_of_cpus} / 8$ ，并舍入 为最接近的 2 的乘方（最大为 64 次方）。

1: 只有 1 个分区；此值将禁用分区操作。

2 - 64: 分区数；必须是 2 的乘方。

默认值

0 (Sybase IQ 自动计算分区数)。

范围

只能为 PUBLIC 组设置。下次启动数据库服务器时将对当前数据库起作用。

描述

通过减少锁争用，对缓冲区高速缓存进行分区 有时可以提高具有多个 CPU 系统的性能。通常，应当信赖 Sybase IQ 自动计算的 值，该值是以系统中的 CPU 数为依据的。但是，如果您发现多 CPU 配置中的装载或查询性能比预期低， 则可以通过为 CACHE_PARTITIONS 设置其它值 来提高装载或查询性能。请参见《系统管理指南 第一卷》>“事务和版本控制”>“管理锁的工具”>“调查锁争用的工具”。

CPU 数和平台都可能对理想的分区数产生影响。可尝试使用不同的值以确定适用于您的配置的最佳设置。

您为 CACHE_PARTITIONS 设置的值同时适用于主缓冲区高速缓存和临时缓冲区高速缓存。对于每个缓冲区高速缓存，绝对最大分区数都是 64。

-iqpartition 服务器选项用于在服务器级设置分区限制。如果在服务器启动时指定了 **-iqpartition**，该选项将始终覆盖 CACHE_PARTITIONS 设置。请参见《实用程序指南》。

分区数不会影响其它缓冲区高速缓存设置。而且，也不会影响 IQ 监视器收集的统计信息；所有分区的统计信息均作为单个值进行累计和报告。

示例

在具有 100 个 CPU 的系统中，如果您不设置 CACHE_PARTITIONS，Sybase IQ 会自动将分区数设置为 16：

100 个 CPU/8 = 12，舍入为 16。

在这种设置中，共有 16 个分区用于 main 高速缓存，16 个分区用于临时高速缓存。

在具有 100 个 CPU 的相同系统中，要将分区数显式设置为 8，请指定：

```
SET OPTION "PUBLIC".CACHE_PARTITIONS=8
```

CHAINED 选项 [TSQL]

控制在缺少 **BEGIN TRANSACTION** 语句时的事务模式。

允许值

ON、OFF

OFF（对于 Open Client 和 JDBC 连接）

默认值

ON

描述

控制 Transact-SQL 事务模式。在非链式模式 (**CHAINED = OFF**) 中，除非执行显式 **BEGIN TRANSACTION** 语句来启动事务，否则每个语句都是单独提交的。在链式模式 (**CHAINED = ON**) 中，事务是在任何数据检索或修改语句之前隐式启动的。对于 Adaptive Server Enterprise，缺省设置为 **OFF**。

CHECKPOINT_TIME 选项

设置数据库服务器在不进行检查点操作的情况下运行的最大时长（以分钟为单位）。

允许值

整数

默认值

60

范围

只能为 PUBLIC 组设置。设置此选项需要具有 DBA 权限。必须关闭并重新启动数据库服务器，才能使更改生效。

描述

此选项与 **RECOVERY_TIME** 一起使用，以决定应在何时进行检查点操作。

另请参见

- **RECOVERY_TIME** 选项（第 428 页）

CIS_ROWSET_SIZE 选项

设置每次读取从远程服务器返回的行数。

允许值
整数

默认值
50

范围
可针对单个连接或 PUBLIC 组进行设置。在与远程服务器建立新连接时生效。

描述
此选项会在您使用 ODBC 连接到远程数据库服务器时设置 ODBC **FetchArraySize** 的值。有关远程数据访问的信息，请参见《系统管理指南第二卷》。

CLOSE_ON_ENDTRANS 选项 [TSQL]

控制在事务结束时关闭游标。

允许值
ON

默认值
ON

描述
当 CLOSE_ON_ENDTRANS 设置为 ON（缺省值及唯一允许值）时，游标将在事务结束时关闭，该行为与 Transact-SQL 兼容。

CONTINUE_AFTER_RAISERROR 选项 [TSQL]

控制 RAISERROR 语句后面跟随的行为。

允许值
ON、OFF

默认值
ON

描述
RAISERROR 语句用在过程中以产生错误。如果 CONTINUE_AFTER_RAISERROR 设置为 OFF，当遇到 **RAISERROR** 语句时将停止执行过程。

如果 **CONTINUE_AFTER_RAISERROR** 设置为 **ON**，**RAISERROR** 语句将不再发出有关执行结束错误的信号。而是在过程完成时存储 **RAISERROR** 状态码和消息并返回最新的 **RAISERROR**。如果导致 **RAISERROR** 的过程是从其它过程调用的，则在最外层的调用过程结束之前，不会返回 **RAISERROR**。

过程终止后，中间 **RAISERROR** 状态和代码将会丢失。如果在返回时伴随 **RAISERROR** 发生了错误，则返回错误信息，而 **RAISERROR** 信息将丢失。应用程序可以通过在不同的执行点检查 @@error 全局变量来查询中间 **RAISERROR** 状态。

只有在 **ON_TSQL_ERROR** 选项设置为 **CONDITIONAL**（缺省设置）时，才使用 **CONTINUE_AFTER_RAISERROR** 的设置来控制 **RAISERROR** 语句后面的行为。如果将 **ON_TSQL_ERROR** 选项设置为 **STOP** 或 **CONTINUE**，**ON_TSQL_ERROR** 设置会优先于 **CONTINUE_AFTER_RAISERROR** 设置。

另请参见

- **ON_TSQL_ERROR** 选项 [TSQL]（第 412 页）

CONVERSION_ERROR 选项 [TSQL]

控制从数据库读取信息时对数据类型转换错误的报告。

允许值

ON、OFF

默认值

ON

描述

当从数据库读取数据或向数据库中插入数据时，此选项用于控制数据库是将数据类型转换故障报告为错误（**CONVERSION_ERROR** 设置为 **ON**）还是报告为警告（**CONVERSION_ERROR** 设置为 **OFF**）。

当 **CONVERSION_ERROR** 设置为 **ON** 时，产生 **SQLE_CONVERSION_ERROR** 错误。

如果此选项设置为 **OFF**，则生成警告 **SQLE_CANNOT_CONVERT**。为 **LOAD** 语句执行数据转换的每个线程最多可以向 .iqmsg 文件写入一条警告消息。

如果只以警告的形式报告转换错误，则使用 **NULL** 值代替不能进行转换的值。在嵌入式 SQL 中，对于导致错误的列，指示符变量将被设置为 -2。

CONVERSION_MODE 选项

针对各种操作限制二进制数据类型（**BINARY**、**VARBINARY** 和 **LONG BINARY**）与其它非二进制数据类型（**BIT**、**TINYINT**、**SMALLINT**、**INT**、**UNSIGNED INT**、**BIGINT**、**UNSIGNED BIGINT**、**CHAR**、**VARCHAR** 和 **LONG VARCHAR**）之间的隐式转换。

允许值

0, 1

默认值

0

范围

可以进行公开设置，也可以进行临时设置。设置此选项无需 DBA 权限。

描述

缺省值 0 将保留 12.7 版之前的隐式转换行为。将 **CONVERSION_MODE** 设置为 1，可限制在执行 **INSERT**、**UPDATE** 操作以及查询时将二进制数据类型隐式转换为任何其他非二进制数据类型。这种限制二进制转换模式还适用于 **LOAD TABLE** 缺省值和 **CHECK** 约束。使用此选项可防止对加密数据进行隐式数据类型转换（这种转换会产生在语义上无意义的操作）。

有关数据类型转换的详细信息，请参见《系统管理指南第一卷》。

有关列加密的详细信息，请参见“Sybase IQ 中的高级安全性”。用户必须获得专门许可，才能使用 Sybase IQ 中的“高级安全性”选项的加密列功能。

隐式转换限制

CONVERSION_MODE 选项限制二进制模式值 1 (**CONVERSION_MODE** = 1) 将限制以下操作的隐式转换：

- **LOAD TABLE**（具有 **CHECK** 约束或缺省值）
- **INSERT...SELECT**、**INSERT...VALUE** 和 **INSERT...LOCATION**
- 特定类型的 **UPDATE**
- 通过可更新游标进行的特定类型的 **INSERT** 和 **UPDATE**
- 一般查询的各个方面

限制 **LOAD TABLE** 的隐式二进制转换模式

这种限制隐式二进制转换模式（**CONVERSION_MODE** 设置为 1）适用于具有 **CHECK** 约束或缺省值的 **LOAD TABLE**。

示例

```
CREATE TABLE t3 (c1 INT,
                  csi SMALLINT,
                  cvb VARBINARY(2),
                  CHECK (csi<cvb));
SET TEMPORARY OPTION CONVERSION_MODE = 1;
```

以下请求：

```
LOAD TABLE t3(c1 ' ', csi ' ', cvb ' ',)
FROM '/s1/mydata/t3.inp'
```

```
QUOTES OFF ESCAPES OFF
ROW DELIMITED BY '\n'
```

失败，显示以下消息：

```
“谓词中的数据类型比较无效 (t3.csi < t3.cvb), [-1001013] ['QFA13']”
```

限制 **INSERT** 的隐式二进制转换模式

这种限制隐式二进制转换模式（`CONVERSION_MODE` 设置为 1）适用于 **INSERT...SELECT**、**INSERT...VALUE** 和 **INSERT...LOCATION**。

示例

```
CREATE TABLE t1 (c1 INT PRIMARY KEY,
  cbt BIT NULL,
  cti TINYINT,
  csi SMALLINT,
  cin INTEGER,
  cui UNSIGNED INTEGER,
  cbi BIGINT,
  cub UNSIGNED BIGINT,
  cch CHAR(10),
  cvc VARCHAR(10),
  cbn BINARY(8),
  cvb VARBINARY(8),
  clb LONG BINARY,
  clc LONG VARCHAR);

CREATE TABLE t2 (c1 INT PRIMARY KEY,
  cbt BIT NULL,
  cti TINYINT,
  csi SMALLINT,
  cin INTEGER,
  cui UNSIGNED INTEGER,
  cbi BIGINT,
  cub UNSIGNED BIGINT,
  cch CHAR(10),
  cvc VARCHAR(10),
  cbn BINARY(8),
  cvb VARBINARY(8),
  clb LONG BINARY,
  clc LONG VARCHAR);

CREATE TABLE t4 (c1 INT, cin INT DEFAULT 0x31);

SET TEMPORARY OPTION CONVERSION_MODE = 1;
```

以下请求：

```
INSERT INTO t1(c1, cvb) SELECT 99, cin FROM T2
WHERE c1=1
```

失败，显示以下消息：

```
“无法将列 'cvb' 从数据类型 (integer) 转换为所请求的数据类型 (varbinary)。
[-1013043] ['QCA43']”
```

限制 UPDATE 的隐式二进制转换模式

这种限制隐式二进制转换模式（CONVERSION_MODE 设置为 1）适用于特定类型的 **UPDATE**。

限制隐式二进制转换模式适用于：

- **UPDATE SET VALUE FROM***expression*（包括常量）
- **UPDATE SET VALUE FROM***other column*
- **UPDATE SET VALUE FROM***host variable*
- **JOIN UPDATE SET VALUE FROM***column of other table*

示例

以下请求：

```
UPDATE t1 SET cbi=cbn WHERE c1=1
```

失败，显示以下消息：

```
“无法将列 'cbi' 从数据类型 (binary) 隐式转换为数据类型 (begin)。[-1000187]
['QCB87']”
```

限制通过可更新游标对定位 INSERT 和定位 UPDATE 进行的隐式二进制转换模式

这种限制隐式二进制转换模式（CONVERSION_MODE 设置为 1）通过可更新游标适用于特定类型的 **INSERT** 和 **UPDATE**。

限制隐式二进制转换模式适用于：

- **PUT***cursor-name***USING** ... *host-variable*
- 来自其它列的定位 **UPDATE**
- 来自常量的 **UPDATE**
- 来自宿主变量的定位 **UPDATE**

限制查询的隐式二进制转换模式

一般来讲，这种限制隐式二进制转换模式（CONVERSION_MODE 设置为 1）适用于查询的各个方面。

比较运算符

当 CONVERSION_MODE = 1 时，该限制适用于以下运算符：

- =、!=、<、<=、>=、<>、!>、!<
- BETWEEN ... AND
- IN

用于以下子句的搜索条件：

- **WHERE** 子句

- **HAVING** 子句
- **CHECK** 子句
- 连接中的 **ON** 短语
- **IF/CASE** 表达式

示例

以下查询：

```
SELECT COUNT(*) FROM T1  
WHERE cvb IN (SELECT csi FROM T2)
```

失败，显示以下消息：

```
“谓词中的数据类型比较无效 (t1.cvb IN (SELECT t1.csi ...)), [-1001013]  
['QFA13']”
```

字符串函数

当 **CONVERSION_MODE = 1** 时，该限制适用于以下字符串函数：

- **CHAR**
- **CHAR_LENGTH**
- **DIFFERENCE**
- **LCASE**
- **LEFT**
- **LOWER**
- **LTRIM**
- **PATINDEX**
- **RIGHT**
- **RTRIM**
- **SIMILAR**
- **SORTKEY**
- **SOUNDEX**
- **SPACE**
- **STR**
- **TRIM**
- **UCASE**
- **UPPER**

示例

以下查询：

```
SELECT ASCII(cvb) FROM t1 WHERE c1=1
```

失败，显示以下消息：

```
“数据例外 - 无法进行数据类型转换。ASCII 的参数必须是字符串, [-1009145]  
['QFA2E']”
```

以下函数允许使用字符串参数或二进制参数。当 `CONVERSION_MODE = 1` 时，该限制适用于混合类型参数，即一个是字符串参数，另一个是二进制参数。

- **INSERTSTR**
- **LOCATE**
- **REPLACE**
- **STRING**
- **STUFF**

示例

以下查询：

```
SELECT STRING(cvb, cvc) FROM t1 WHERE c1=1
```

如果列 `cvb` 定义为 `VARBINARY`，列 `cvc` 定义为 `VARCHAR`，则失败，

显示以下消息：

```
“数据例外 - 无法进行数据类型转换。STRING 的参数必须全部是二进制或字符串，
[-1009145] ['QFA2E']”
```

该限制不适用于以下字符串函数：

- **BIT_LENGTH**
- **BYTE_LENGTH**
- **CHARINDEX**
- **LENGTH**
- **OCTET_LENGTH**
- **REPEAT**
- **REPLICATE**
- **SUBSTRING**

算术运算和函数

当 `CONVERSION_MODE = 1` 时，该限制适用于算术运算中使用的以下运算符：

`+`, `-`, `*`, `/`

该限制适用于逐位表达式中使用的以下逐位运算符：

`&` (AND)、`|` (OR)、`^` (XOR)

该限制还适用于以下函数的整数参数：

- **ROUND**
- **“TRUNCATE”**
- **TRUNCNUM**

示例

以下查询：

```
SELECT ROUND(4.4, cvb) FROM t1 WHERE C1=1
```

失败，显示以下消息：

```
“数据例外 - 无法进行数据类型转换。ROUND 的第二个参数不能转换为整数, [-1009145]
['QFA2E']”
```

各种函数的整数参数

当 `CONVERSION_MODE = 1` 时，该限制适用于以下函数的整数参数：

- **ARGN**
- **SUBSTRING**
- **DATEADD**
- **YMD**

示例

以下查询：

```
SELECT ARGN(cvb, csi, cti) FROM t1 WHERE c1=1
```

失败，显示以下消息：

```
“数据例外 - 无法进行数据类型转换。ARGN 的第一个参数不能转换为整数, [-1009145]
['QFA2E']”
```

解析函数、集合函数和数值函数

当 `CONVERSION_MODE = 1` 时，对于需要使用数值表达式作为参数的解析函数、集合函数和数值函数，不再应用其它限制。

CONVERT_VARCHAR_TO_1242 选项

将 12.4.2 版之前的 VARCHAR 数据转换为压缩格式。

允许值

ON、OFF

默认值

OFF

范围

只能为 PUBLIC 组设置。在任何模式下运行 **sp_iqcheckdb** 时生效。

描述

帮助进一步压缩数据并提高性能，尤其适用于具有许多可变字符串的数据库。

设置此选项，然后只针对在 12.4.2 版之前创建的 VARCHAR 列运行 **sp_iqcheckdb**，且只运行一次。

COOPERATIVE_COMMIT_TIMEOUT 选项

控制事务日志中 COMMIT 条目写入磁盘的时间。

允许值

整数（以毫秒为单位）

默认值

250

范围

可针对单个连接或 PUBLIC 组进行设置。设置将立即生效。

描述

只有在 COOPERATIVE_COMMITS 设置为 ON 时，此选项才有意义。数据库服务器在写入磁盘之前会等待指定的毫秒数，以便其它连接可以填充日志页。缺省设置为 250 毫秒。

另请参见

- COOPERATIVE_COMMITS 选项（第 353 页）

COOPERATIVE_COMMITS 选项

控制提交写入磁盘的时间。

允许值

ON、OFF

默认值

ON

范围

可针对单个连接或 PUBLIC 组进行设置。设置将立即生效。

描述

如果 COOPERATIVE_COMMITS 设置为 OFF，则数据库服务器会在接收到 **COMMIT** 后立即将其写入磁盘，然后允许应用程序继续运行。

如果 COOPERATIVE_COMMITS 设置为 ON（缺省值），则数据库服务器不会立即将 **COMMIT** 写入磁盘，而是要求应用程序等待一段时间（即 COOPERATIVE_COMMIT_TIMEOUT 选项设置的最长时间），以便在提交写入磁盘之前将其它内容置于页面中。

将 COOPERATIVE_COMMITS 设置为 On，并且增大 COOPERATIVE_COMMIT_TIMEOUT 设置，可以削减磁盘 I/O 数，从而增加数据库服务器的总体吞吐量，但代价是各个连接的周转时间会更长。

另请参见

- COOPERATIVE_COMMIT_TIMEOUT 选项（第 353 页）

CURSOR_WINDOW_ROWS 选项

定义缓冲区中的游标行数。

允许值

20 - 100000

默认值

200

范围

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

描述

当应用程序打开游标时，Sybase IQ 会创建一个 FIFO（先进先出）缓冲区，用于保存查询生成的数据行。CURSOR_WINDOW_ROWS 用于定义缓冲区中可以容纳的行数。如果游标是在 NO SCROLL 之外的任意模式下打开的，Sybase IQ 允许最多向后滚动缓冲区中允许的总行数，超过这个总行数就要重新启动查询。对于 NO SCROLL 游标则不是这样，因为它们不允许向后滚动。

例如，当此选项设置为缺省值时，缓冲区最初包含查询结果集中的 1 至 200 行。如果读取前 300 行，缓冲区中将包含 101 至 300 行。您可以用极少的开销在该缓冲区中前后滚动。如果滚动至 101 行之前，Sybase IQ 将重新启动该查询，直至所需的行回到缓冲区中为止。执行此操作的开销可能很大，因此应用程序应当尽可能避免出现这种情况。一种方法是增加 CURSOR_WINDOW_ROWS 的值以容纳尽可能大的滚动区域；不过对于大多数应用程序，缺省设置 200 已足够。

DATE_FIRST_DAY_OF_WEEK 选项

确定一周的第一天。

允许值

0 - 6

默认值

0（星期日）

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

描述
此选项指定哪一天是一周的第一天。缺省情况下，星期日是一周的第 1 天，星期一是第 2 天，星期二是第 3 天，依此类推：

表 22. DATE_FIRST_DAY_OF_WEEK 有效值

值	第一天
0	星期日
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

例如，如果将 DATE_FIRST_DAY_OF_WEEK 选项的值更改为 3，星期三将成为一周的第 1 天，星期四成为第 2 天，依此类推。此选项只会影响 DOW 和 DATEPART 函数。

SQL Anywhere 选项 FIRST_DAY_OF_WEEK 可实现同样的功能，但赋予的值为 1 至 7 而不是 0 至 6。1 代表星期一，7 代表星期日（缺省值）。

DATE_FORMAT 选项

设置用于从数据库中检索的日期的格式。

允许值
字符串

默认值
'YYYY-MM-DD'。这种格式符合 ISO 日期格式说明。

范围
可针对单个连接或 PUBLIC 组进行设置。设置将立即生效。

描述
该格式是使用以下符号的字符串：

表 23. DATE_FORMAT 字符串中使用的符号

符号	描述
yy	表示年份的 2 位数
yyyy	表示年份的 4 位数
mm	两位数表示的月份，如果跟在冒号后面（例如“hh:mm”这样的格式），则为两位数表示的分钟数
mmm	3 个字符表示的月份名称
mmmm[m...]	月份的字符长格式 — 有多少个 m，就有多少个字符，直至指定的 m 个数超过相应月份名称中的字符数为止。
d	一位数表示的星期几（0 代表星期日，6 代表星期六）
dd	两位数表示的日期
ddd	3 个字符表示的星期几。
dddd[d...]	星期天数的字符长格式 — 有多少个 d，就有多少个字符，直至指定的 d 个数超过相应星期名称中的字符数为止。
jjj	一年中的某天，从 1 到 366

注意：在日期格式字符串中，不支持使用多字节字符。即使数据库的归类顺序是多字节归类顺序（如 932JPN），也只能使用单字节字符。使用连接运算符可以在日期格式字符串中引入多字节字符。例如，如果“?”表示一个多字节字符，则可以使用连接运算符将该多字节字符移到日期格式字符串的外面：

```
SELECT DATEFORMAT (StartDate, 'yy') + '?'
FROM Employees;
```

每个符号都会替换为与要设置格式的日期对应的数据。任何表示字符而不是数字输出的格式符号都可以采用大写形式，这样会使替换的字符也显示为大写。对于数字，在格式字符串中使用混合的大小写会取消前导零的显示。

可以通过更改符号的大小写控制数字的填充。大小写一致的值（MM、mm、DD 或 dd）都用零填充数字。大小写混合的值（Mm、mM、Dd 或 dD）使数字不会用零填充；值将获得所需的任意大小的空间。例如：

```
SELECT dateformat ( cast ( '2011/01/01' as date ), 'yyyy/Mm/Dd' )
```

返回值以下值：

```
2011/1/1
```

示例

此表介绍了 DATE_FORMAT 设置与来自下列语句（执行时间为 2011 年 5 月 21 日，星期六）的输出：

```
SELECT CURRENT DATE
```

表 24. DATE_FORMAT 设置

DATE_FORMAT	SELECT CURRENT DATE
yyyy/mm/dd/dd	2011/05/21/sat
jjj	141
mmm yyyy	may 2011
mm-yyyy	05-2011

另请参见

- RETURN_DATE_TIME_AS_STRING 选项（第 429 页）
- TIME_FORMAT 选项（第 453 页）

DATE_ORDER 选项

控制对日期格式的解释。

允许值

'MDY'、'YMD' 或 'DMY'

默认值

'YMD'。这种格式符合 ISO 日期格式说明。

描述

DATE_ORDER 用于确定 10/11/12 是表示 1912 年 10 月 11 日、1910 年 11 月 12 日还是 1912 年 11 月 10 日。该选项的值可以是 'MDY'、'YMD' 或 'DMY'。

DBCC_LOG_PROGRESS 选项

报告 sp_iqcheckdb 系统存储过程的进度。

允许值

ON、OFF

默认值

OFF

范围

可针对单个连接或 PUBLIC 组进行设置。在下次执行 sp_iqcheckdb 时生效。

描述

当 DBCC_LOG_PROGRESS 设置为 ON 时，**sp_iqcheckdb** 系统存储过程会将进度消息发送给 IQ 消息文件。通过这些消息，用户可以紧跟 **sp_iqcheckdb** 操作的进度。

有关存储过程，请参见《参考：构件块、表和过程》。

示例

命令 **sp_iqcheckdb** ‘check database’ 的进度日志输出示例：

```
IQ Utility Check Database Start CHECK STATISTICS table: tloansf Start  
CHECK STATISTICS for field: aqsn_dt Start CHECK STATISTICS processing  
index: IQ_IDX_T444_C1_FP Start CHECK STATISTICS processing index:  
tloansf_aqsn_dt_HNG Done CHECK STATISTICS field: aqsn_dt
```

命令 **sp_iqcheckdb** ‘allocation table nation’ 的进度日志输出示例：

```
Start ALLOCATION table: nation Start ALLOCATION processing index:  
nationhgl Done ALLOCATION table: nation Done ALLCOATION processing  
index: nationhgl
```

DBCC_PINNABLE_CACHE_PERCENT 选项

控制 **sp_iqcheckdb** 系统存储过程使用的高速缓存百分比。

允许值

0 - 100

默认值

50

范围

可针对单个连接或 PUBLIC 组进行设置。在下次执行 **sp_iqcheckdb** 时生效。

描述

sp_iqcheckdb 系统存储过程使用固定数量的缓冲区，该数量由此选项决定。缺省情况下，将保留较大比例的高速缓存以最大限度地提高 **sp_iqcheckdb** 性能。

有关存储过程，请参见《参考：构件块、表和过程》。

有关 **sp_iqcheckdb** 的故障排除信息，请参见《系统管理指南第二卷》>“系统恢复和数据库修复”>“数据库检验”>“运行 **sp_iqcheckdb** 的问题”。

DEBUG_MESSAGES 选项

控制是否执行包含 DEBUG ONLY 子句的 MESSAGE 语句。

允许值

ON、OFF

默认值

OFF

描述

此选项可用于控制在包含指定了 **DEBUG ONLY** 子句的 **MESSAGE** 语句的存储过程中调试消息的行为。缺省情况下，此选项设置为 **Off**，在执行 **MESSAGE** 语句时不显示调试消息。通过将 **DEBUG_MESSAGES** 设置为 **ON**，可以启用所有存储过程中的调试消息。

注意： **DEBUG ONLY** 将 **DEBUG_MESSAGES** 选项设置为 **OFF** 时，消息并不占用大量资源，因此在生产系统中，通常可以将这些语句保留在存储过程中。不过，在可能会频繁执行这些语句的地方应慎用它们；否则，它们可能会导致性能略微降低。

另请参见

- **MESSAGE** 语句（第 242 页）

DEDICATED_TASK 选项

将请求处理任务专用于处理来自单个连接请求。

允许值

ON、OFF

默认值

OFF

范围

在当前连接的整个期间只能设置为临时选项。设置此选项需要具有 **DBA** 权限。

描述

DEDICATED_TASK 连接选项设置为 **ON** 时，一个请求处理任务将专门处理一个连接的请求。通过在启用此选项的情况下预先建立连接，可以在数据库服务器意外地不响应时收集有关其状态的信息。

DEFAULT_DBSPACE 选项

更改在其中创建了表或连接索引的缺省数据库空间。

允许值

包含数据库空间名称的字符串

默认值

"（空字符串）

范围

可针对单个连接或 **PUBLIC** 组进行设置。设置会立即生效。需要具有 **DBA** 权限才能针对组或除了当前用户以外的用户设置此选项。设置将立即生效。

描述

DEFAULT_DBSPACE 允许管理员设置组或用户的缺省数据库空间，或允许用户设置自己的缺省数据库空间。

IQ_SYSTEM_TEMP 将始终用于全局临时表，除非使用了指定 **SYSTEM** 的表 **IN** 子句，在这种情况下将创建 **SA** 全局临时表。

创建数据库时，当 **PUBLIC.DEFAULT_DBSPACE** 选项设置为空或显式设置为 **IQ_SYSTEM_MAIN** 时，将创建和暗示系统数据库空间 **IQ_SYSTEM_MAIN**。Sybase 建议管理员在创建数据库后，立即创建第二个主数据库空间，从 **PUBLIC** 调用数据库空间 **IQ_SYSTEM_MAIN** 中的 **CREATE** 特权，将新主数据库空间的数据库空间中的 **CREATE** 授予选定用户或 **PUBLIC**，并将 **PUBLIC.DEFAULT_DBSPACE** 设置为新主数据库空间。例如：

```
CREATE DBSPACE user_main USING FILE user_main
'user_main1' SIZE 10000;
GRANT CREATE ON user_main TO PUBLIC;
REVOKE CREATE ON IQ_SYSTEM_MAIN FROM PUBLIC;
SET OPTION PUBLIC.DEFAULT_DBSPACE = 'user_main';
```

示例

在此示例中，将所有数据库空间中的 **CONNECT** 和 **RESOURCE** 特权授予用户 **usrA** 和 **usrB**，并对每一用户授予对特定数据库空间的 **CREATE** 特权：

```
GRANT CONNECT, RESOURCE TO usrA, usrB
IDENTIFIED BY pwdA, pwdB;
GRANT CREATE ON dbsp1 TO usrA;
GRANT CREATE ON dbsp3 TO usrB;
SET OPTION "usrA".default_dbpace = 'dbsp1' ;
SET OPTION "usrB".default_dbpace = 'dbsp3' ;
SET OPTION "PUBLIC".default_dbpace = dbsp2;

CREATE TABLE "DBA".t1(c1 int, c2 int);
INSERT INTO t1 VALUES (1, 1);
INSERT INTO t1 VALUES (2, 2);
COMMIT;
```

UsrA 连接:

```
CREATE TABLE "UsrA".t1(c1 int, c2 int);
INSERT INTO t1 VALUES (1, 1);
INSERT INTO t1 VALUES (2, 2);
COMMIT;
```

UsrB 连接:


```
CREATE TABLE "UsrB" .t1(c1 int, c2 int);
INSERT INTO t1 VALUES (1, 1);
INSERT INTO t1 VALUES (2, 2);
COMMIT;
```

DBA 连接:

```
SELECT Object, DbpaceName, ObjSize
FROM sp_iqindexinfo();
```

sp_iqindexinfo 结果:

DBA.t1	dbsp2	200k
DBA.t1.ASIQ_IDX_T730_C1_FP	dbsp2	288k
DBA.t1.ASIQ_IDX_T730_C2_FP	dbsp2	288k
usrA.t1	dbsp1	200k
usrA.t1.ASIQ_IDX_T731_C1_FP	dbsp1	288k
usrA.t1.ASIQ_IDX_T731_C2_FP	dbsp1	288k
usrB.t1	dbsp3	200k
usrB.t1.ASIQ_IDX_T732_C1_FP	dbsp3	288k
usrB.t1.ASIQ_IDX_T732_C2_FP	dbsp3	288k

DEFAULT_DISK_STRIPING 选项

设置所有数据库空间的缺省磁盘条带化值。

允许值

ON、OFF

默认值

ON

范围

只能为 PUBLIC 组设置。需要具有 DBA 权限。

描述

缺省情况下，IQ main 存储中所有数据库空间的磁盘条带化都为 ON。如果 **CREATE DBSPACE** 未定义条带化，此选项仅用于 **CREATE DBSPACE** 和定义缺省条带化值。

另请参见

- CREATE DBSPACE 语句（第 77 页）

DEFAULT_HAVING_SELECTIVITY_PPM 选项

为大部分 **HAVING** 子句的优化程序提供缺省选择性估计值（以百万分率表示）。

允许值

0 - 1000000

默认值

0

范围

可针对单个连接或 PUBLIC 组进行设置。设置将立即生效。

描述

DEFAULT_HAVING_SELECTIVITY_PPM 为 **HAVING** 子句设置选择性，覆盖优化程序估计值。**HAVING** 子句可过滤 **GROUP BY** 子句的结果或选择列表中只包含集合函数的查询的结果。当 DEFAULT_HAVING_SELECTIVITY_PPM 设置为缺省值 0 时，优化程序将估计 **HAVING** 子句过滤的行数。有时 IQ 优化程序获得的信息不全面，无法选择精确的选择性，在这种情况下会选择通用估计值 40%。

DEFAULT_HAVING_SELECTIVITY_PPM 使用户可以替换查询中所有 **HAVING** 谓词的优化程序估计值。

用户还可以指定查询中各个 **HAVING** 子句的选择性，如《参考：构件块、表和过程》中所述。

DEFAULT_ISQL_ENCODING 选项 [Interactive SQL]

指定 **READ** 和 **OUTPUT** 语句使用的代码页。

允许值

identifier 或 *string*

缺省值

使用系统代码页（空字符串）

范围

在当前连接的整个期间只能设置为临时选项。

描述

DEFAULT_ISQL_ENCODING 用于指定读取或写入文件时使用的代码页。不能永久设置此选项。缺省代码页是所运行平台的缺省代码页。在英语版 Windows 计算机上，缺省代码页为 1252。

Interactive SQL 按以下方法确定用于特定 **OUTPUT** 或 **READ** 语句的代码页，其中，列表中位置靠前的代码页值优先于位置靠后的代码页值：

- 在 **OUTPUT** 或 **READ** 语句的 **ENCODING** 子句中指定的代码页
- 使用 DEFAULT_ISQL_ENCODING 选项指定的代码页（如果设置此选项）
- 运行 Interactive SQL 的计算机的缺省代码页

有关支持的代码页的列表，请参见“SQL Anywhere Server - 数据库管理”>“配置数据库”>“国际语言和字符集”>“字符集和归类参考信息”>“支持的归类和替代归类”。

另请参见“SQL Anywhere Server - 数据库管理”>“配置数据库”>“国际语言和字符集”>“了解字符集”>“了解字符集”。

注意： 这些参考指向 SQL Anywhere 文档。

示例

将编码设置为 UTF-16（用于读取 Unicode 文件）：

```
SET TEMPORARY OPTION DEFAULT_ISQL_ENCODING = 'UTF-16'
```

另请参见

- OUTPUT 语句 [Interactive SQL]（第 247 页）
- READ 语句 [Interactive SQL]（第 258 页）

DEFAULT_KB_PER_STRIPE 选项

设置在写入操作移至下一个带区以前数据写入的带区量上限（以 KB 为单位）。

此设置是 IQ main 存储中所有数据库空间的缺省大小。

允许值

1 到最大整数

默认值

1

范围

只能为 PUBLIC 组设置。需要具有 DBA 权限。

描述

缺省值为 1KB 意味着压缩一页，且压缩的页作为单一操作写入磁盘。无论选择的页面大小如何，下一操作将写入该数据库空间的下一数据库文件。

要在移至下一带区前向同一带区写入多页，请更改 DEFAULT_KB_PER_STRIPE 设置。例如，如果页大小为 128KB，并且 DEFAULT_KB_PER_STRIPE 设置为 512KB，Sybase IQ 将对页写入排队并在压缩页达到 512KB 的最小值后写入磁盘。

如果 **CREATE DBSPACE** 不指定条带化大小，此选项仅用于 **CREATE DBSPACE** 和定义 IQ main 存储中数据库空间的缺省磁盘条带化大小。

另请参见

- CREATE DBSPACE 语句（第 77 页）

DEFAULT_LIKE_MATCH_SELECTIVITY_PPM 选项

为大部分 **LIKE** 谓词的优化程序提供缺省选择性估计值（以百万分率表示）。

允许值

0 到 1000000

默认值

150000

范围

可针对单个连接或 PUBLIC 组进行设置。设置将立即生效。

描述

DEFAULT_LIKE_MATCH_SELECTIVITY_PPM 可为一般 **LIKE** 为此设置缺省选择性，例如 **LIKE** '*string%string*'，其中 % 是通配符。

如果未提供其它选择性信息，并且匹配字符串不是以一组常量字符后跟单个通配符开头，则优化程序会使用此选项。

如果该列包含 **LF** 索引或单字节、双字节或三字节的 **FP** 索引，则优化程序可以获得准确的信息且不需要使用该值。

用户还可以指定查询中的选择性。有关用户提供的条件提示，请参见《参考：构件块、表和过程》。

另请参见

- DEFAULT_LIKE_RANGE_SELECTIVITY_PPM 选项（第 364 页）
- FP_LOOKUP_SIZE 选项（第 370 页）

DEFAULT_LIKE_RANGE_SELECTIVITY_PPM 选项

为前导常量 **LIKE** 谓词的优化程序提供缺省选择性估计值（以百万分率表示）。

允许值

1 到 1000000

默认值

150000

范围

可针对单个连接或 PUBLIC 组进行设置。设置将立即生效。

描述

DEFAULT_LIKE_RANGE_SELECTIVITY_PPM 可为 **LIKE** 谓词设置缺省选择性，格式为 **LIKE** '*string%*'，其中匹配字符串是一组常量字符后跟一个通配符 (%)。如果未提供其它选择性信息，优化程序会使用此选项。

如果该列包含 **LF** 索引或单字节、双字节或三字节的 **FP** 索引，则优化程序可以获得准确的信息且不需要使用该值。

用户还可以指定查询中的选择性。有关用户提供的条件提示，请参见《参考：构件块、表和过程》。

另请参见

- `DEFAULT_LIKE_MATCH_SELECTIVITY_PPM` 选项 (第 363 页)
- `FP_LOOKUP_SIZE` 选项 (第 370 页)

DEFAULT_PROXY_TABLE_ROW_COUNT 选项

让您能够覆盖从代理表返回的行数的缺省估计值。

允许值

0 到 4294967295

缺省值

200000

范围

设置此选项需要具有 DBA 权限。只能为 PUBLIC 组临时设置。

DEFAULT_TABLE_UDF_ROW_COUNT Option

使您得以替换表 UDF (C、C++ 或 Java 表 UDF) 所返回的默认行数估计。

允许值

0 至 4294967295

缺省值

200000

范围

设置此选项需要具有 DBA 权限。只能为 PUBLIC 组临时设置。

描述

表 UDF 可以使用 **DEFAULT_TABLE_UDF_ROW_COUNT** 选项为查询处理器提供估计行数 (表 UDF 将返回该值)。这是 Java 表 UDF 传递此信息的唯一方法。然而, 对于使用 C/C++ 编写的表 UDF, UDF 开发人员应该考虑在 `describe` 阶段发布此信息, 即使用 `EXTFNAPIV4_DESCRIBE_PARM_TABLE_NUM_ROWS` 描述参数发布它所期望返回的行数。 `EXTFNAPIV4_DESCRIBE_PARM_TABLE_NUM_ROWS` 的值始终会覆盖 **DEFAULT_PROXY_TABLE_UDF_ROW_COUNT** 选项的值。

DELAYED_COMMIT_TIMEOUT 选项

确定服务器在执行一条 **COMMIT** 后何时将控制返回应用程序。

允许值

整数 (以毫秒为单位) 。

默认值

500

描述

此选项将被 Sybase IQ 忽略，因为 DELAYED_COMMITS 只能 设置为 OFF。

DELAYED_COMMITS 选项

确定服务器在执行一条 **COMMIT** 后何时将控制返回应用程序。

允许值

OFF

默认值

OFF。此值对应于 ISO COMMIT 行为。

描述

如果设置为 OFF（Sybase IQ 允许的唯一值），应用程序必须等到 **COMMIT** 写入磁盘。对于 ANSI/ISO COMMIT 行为，必须将此选项设置为 OFF。

DISABLE_RI_CHECK 选项

允许装载、插入、更新或删除操作绕过参照完整性检查，以提高性能。

允许值

ON、OFF

默认值

OFF

范围

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

描述

当 DISABLE_RI_CHECK 设置为 ON 时，用户需要确保请求过程中不发生破坏参照完整性的情况。

DIVIDE_BY_ZERO_ERROR 选项 [TSQL]

控制对以零为除数这一情况的报告。

允许值

ON、OFF

默认值

ON

范围

此选项指示是否以错误形式报告以零为除数这一情况。如果此选项设置为 ON，以零为除数会产生一个 SQLSTATE 22012 错误。

如果此选项设置为 OFF，则以零为除数不是错误；会返回 NULL。

DQP_ENABLED 选项

使用临时数据库选项 **dqp_enabled**，您可以在连接级别启用或禁用 DQP。

您可以将临时数据库选项 **dqp_enabled** 设置为 OFF 以禁用当前连接的 DQP。您可以将该选项设置为 ON（缺省值）以启用当前连接的 DQP，但前提是当前连接的逻辑服务器的用户登录策略为该用户启用了 DQP。

用户必须获得 Multiplex Grid Option 使用许可才能运行辅助服务器。有关 **dqp_enabled** 语法和完整说明，请参见《使用 Sybase IQ Multiplex》。

EARLY_PREDICATE_EXECUTION 选项

控制是否在查询优化之前执行简单本地谓词。

允许值

ON、OFF

默认值

ON

范围

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

描述

如果此选项为 ON（缺省设置），优化程序会在查询优化（包括连接排序、连接算法选择和分组算法选择）之前查找、准备和执行只包含本地列和约束的谓词，以提高查询计划中“估计结果行数”值的精度。如果此选项为 OFF，优化程序将查找并准备简单谓词，但不会在查询优化之前执行这些简单谓词。如果不执行谓词，产生的“估计结果行数”值的精度将比较低。

一般来讲，EARLY_PREDICATE_EXECUTION 应始终保持为 ON，因为这样会改善很多查询的查询计划。

请注意，如果 `EARLY_PREDICATE_EXECUTION` 为 `ON`，Sybase IQ 会在生成查询计划之前 为所有查询执行本地谓词，即使 `NOEXEC` 选项为 `ON` 也是如此。生成的查询计划与运行时计划相同。

根节点的查询计划中包括以下信息：

- 用于执行本地不变谓词的线程数：如果大于 1，表示并行执行本地不变谓词
- `Early_Predicate_Execution`：指示该选项是否为 `OFF`
- 游标创建时间：游标创建的时间

简单谓词（其执行受此选项控制）在查询计划中称为不变谓词。如果叶节点有任何本地不变谓词，该节点的查询计划将包括以下信息：

- 执行不变谓词后生成的行数：执行本地不变谓词后产生的实际结果
- 执行不变谓词后的估计行数：使用估计的本地不变谓词选择性计算得出
- 条件开始时间：执行本地不变谓词的开始时间
- 条件完成时间：执行本地不变谓词的结束时间
- 条件经历时间：执行本地不变谓词经历的时间

ENABLE_LOB_VARIABLES 选项

控制大对象变量的数据类型转换。

用户必须获得非结构化数据分析选件的使用许可 才能使用大对象变量。有关 `ENABLE_LOB_VARIABLES` 语法和完整说明，请参见《Sybase IQ 中的非结构化数据分析》。

EXTENDED_JOIN_SYNTAX 选项

控制对于包含多表连接的不明确语法的查询，是允许使用还是报告为错误。

允许值

`ON`、`OFF`

默认值

`ON`

描述

此选项针对包含具有不明确语法（原因是具有空值的表存在重复相关名）的外部连接的查询报告语法错误。

以下连接子句显示了满足 `C1` 条件时报告的查询类型：

```
( R left outer join T , T join S on ( C1 ) )
```

如果 `EXTENDED_JOIN_SYNTAX` 选项设置为 `ON`，此查询的解释如下（其中满足 `C1` 和 `C2` 条件）：

```
( R left outer join T on ( C1 ) ) join S on ( C2 )
```


FORCE_DROP 选项

使 Sybase IQ 在执行 **DROP** 命令的过程中泄漏（而不是回收）数据库磁盘空间。

允许值

ON、OFF

默认值

OFF

范围

设置此选项需要具有 **DBA** 权限。可针对单个连接或 **PUBLIC** 组进行临时设置。设置将立即生效。

描述

必须删除已损坏的索引、连接索引、列或表，并将 **FORCE_DROP** 选项设置为 **ON**。这样可以避免使用所删除对象中的不正确或可疑的文件空间分配信息对空闲列表进行不正确的更新。删除损坏的对象后，可以使用 **-iqfreq** 和 **-iqdroplks** 服务器开关回收文件空间。

在强制删除对象时，必须确保只有 **DBA** 连接到数据库。强制删除后，必须立即重新启动服务器。

除非 Sybase 技术支持部门指导您强制删除对象，否则不要尝试该操作。

FORCE_DROP 有关系统恢复和数据库修复的过程，请参见《系统管理指南：第一卷》。

FORCE_NO_SCROLL_CURSORS 选项

强制所有游标成为非滚动游标。

允许值

ON、OFF

默认值

OFF

范围

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

描述

缺省情况下，所有游标均为滚动游标。如果滚动游标未声明任何宿主变量，将使 Sybase IQ 创建用于临时存储结果的缓冲区。结果集中的每一行都存储为允许向后滚动。

将 **FORCE_NO_SCROLL_CURSORS** 设置为 ON 会降低临时存储需求。如果正在检索的行数非常庞大（数百万），则选项可能会很有用。但如果前端应用程序经常使用向后滚动游标操作，将此选项设置为 OFF 会提高速度。

如果前端应用程序很少执行向后滚动，请将 **FORCE_NO_SCROLL_CURSORS** = 'ON' 作为永久 PUBLIC 选项以使用更少内存，并提高查询性能。

FORCE_UPDATABLE_CURSORS 选项

控制是否可对尚未声明为可更新游标的游标进行更新。

允许值

ON、OFF

默认值

OFF

范围

可针对单个连接或 PUBLIC 组进行临时设置。不需要 DBA 权限。设置将立即生效。

描述

当 **FORCE_UPDATABLE_CURSORS** 选项为 ON 时，可以对尚未声明为可更新游标的游标进行更新。此选项允许在前端应用程序中使用可更新游标，而无需指定 **FOR UPDATE** 语句的 **DECLARE CURSOR** 子句。

除非有绝对的必要，否则 Sybase 不建议您使用 **FORCE_UPDATABLE_CURSORS**。

FP_LOOKUP_SIZE 选项

指定 Sybase IQ 中使用的最大查找页数和控制分配给创建查找 FP 索引的高速缓存量，尤其是 FP(3) 索引。

允许值

1 MB - 4096 MB

默认值

16 MB

范围

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

描述

FP_LOOKUP_SIZE 控制最大查找页数。

FP_LOOKUP_SIZE 必须设置为 Public，因此允许的语法为：

```
SET OPTION public.FP_LOOKUP_SIZE = 1
```

以下数据库选项支持 3 字节索引：

- INDEX_ADVISOR
- MINIMIZE_STORAGE
- FP_LOOKUP_SIZE_PPM

以下存储过程支持 3 字节索引：

- **sp_iqcheckdb**
- **sp_iqcolumn**
- **sp_iqindexadvice**
- **sp_iqindexmetadata**
- **sp_iqindexsize**
- **sp_iqindex**
- **sp_iqindexfragmentation**
- **sp_iqrebuildindex**
- **sp_iqrowdensity**

另请参见

- FP_LOOKUP_SIZE_PPM 选项（第 371 页）
- MINIMIZE_STORAGE 选项（第 405 页）

FP_LOOKUP_SIZE_PPM 选项

将 Sybase IQ 中 FP 查找的 存储大小限制在此主内存的值（以百万分率表示） 之内。

允许值

1 到 1000000

默认值

2500

范围

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

描述

FP_LOOKUP_SIZE_PPM 控制为所有 FP 查找索引（尤其是为 FP(3) 索引）创建查找 FP 索引时所分配的 main 高速缓存量。

此选项控制最大查找页数，并将该数字限制在主内存的值（以百万分率表示）之内，即 $FP_LOOKUP_SIZE_PPM * \text{size of main memory} / 1,000,000$ 的值，其中主内存大小由 **-iqmc** 服务器启动参数确定。

以下选项支持 3 字节索引：

- `FP_LOOKUP_SIZE`
- `INDEX_ADVISOR`
- `MINIMIZE_STORAGE`

另请参见

- `FP_LOOKUP_SIZE` 选项（第 370 页）
- `MINIMIZE_STORAGE` 选项（第 405 页）

FP_PREDICATE_WORKUNIT_PAGES 选项

指定缺省索引中使用的并行度。

允许值

整数

默认值

200

范围

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

描述

缺省索引以并行方式计算某些谓词，如 `SUM`、`RANGE`、`MIN`、`MAX` 和 `COUNT DISTINCT`。`FP_PREDICATE_WORKUNIT_PAGES` 通过指定每个线程处理的页数对所使用的并行度产生影响。要提高并行度，应减少此选项的值。

FPL_EXPRESSION_MEMORY_KB 选项

控制对涉及函数表达式（针对包含枚举存储的列）的查询的优化的内存使用情况。

允许值

0 - 20000

默认值

1024 千字节

范围

可针对单个连接或 `PUBLIC` 组进行临时设置。设置将立即生效。

描述

`FPL_EXPRESSION_MEMORY_KB` 控制对涉及函数表达式（针对包含枚举存储的列）的查询的优化的内存使用情况。此选项使 `DBA` 可以约束该优化占用的内存，并与其它 Sybase IQ 内存需求（如高速缓存）进行平衡。将此选项设置为 0 将关闭优化功能。

GARRAY_FILL_FACTOR_PERCENT 选项

指定每个 **HG** GArray 页要保留以供将来向现有组进行增量插入的空间百分比。

允许值

0 - 1000

默认值

25

范围

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

描述

GArray 将尝试拉长每个组以容纳该值所设置的一块空闲空间。这部分空间用于添加到现有索引组的行。

HG 索引可针对每个组保留一些存储空间（将组定义为具有相等值的一组行）。保留空间会消耗额外的磁盘空间，但对向 **HG** 索引进行增量插入的性能有所帮助。

如果计划将来向 **HG** 索引进行增量插入，而且这些新行的值已存在于索引中，那么为此选项设置一个非零值可提高增量插入的性能。

如果未计划逐步更新索引，则可以减少此选项的值以节省磁盘空间。

另请参见

- **GARRAY_PAGE_SPLIT_PAD_PERCENT 选项**（第 374 页）

GARRAY_INSERT_PREFETCH_SIZE 选项

指定用于预取的页数。

允许值

0 - 100

默认值

3

范围

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

描述

此选项定义在对具有 **HG** 索引的列进行插入的过程中，预先读取的数据库页数。

除非有 Sybase 技术支持部门的指导，否则不要设置此选项。

另请参见

- GARRAY_FILL_FACTOR_PERCENT 选项 (第 373 页)

GARRAY_PAGE_SPLIT_PAD_PERCENT 选项

确定在对 Garray 进行页面拆分的过程中每页的填充因子，并指定每个 **HG** Garray 页要保留用于将来增量插入的空间百分比。

允许值

0 - 100

默认值

25

范围

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

描述

GArray 页的拆分会尝试将该百分比留空。这部分空间用于添加到新索引组的行。

HG 索引可以在页级保留存储空间，插入附加行时可将这些存储空间分配给新组。保留空间会消耗额外的磁盘空间，但对向 **HG** 索引进行增量插入的性能有所帮助。

如果计划将来向 **HG** 索引进行增量插入，而且这些新行不包含索引中已经存在的值，那么为 GARRAY_PAGE_SPLIT_PAD_PERCENT 选项设置一个非零值可提高增量插入的性能。

如果未计划逐步更新索引，则可以减少此选项的值以节省磁盘空间。

另请参见

- GARRAY_FILL_FACTOR_PERCENT 选项 (第 373 页)

GARRAY_RO_PREFETCH_SIZE 选项

指定用于预取的页数。

允许值

0 - 100

默认值

10

范围

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

描述

此选项定义在对具有 **HG** 索引的列进行查询的过程中，预先读取的数据库页数。

除非有 Sybase 技术支持部门的指导，否则不要设置此选项。

HASH_PINNABLE_CACHE_PERCENT 选项

控制散列对象在用户临时内存中可以固定的最大百分比。

允许值

0 - 100

默认值

20

范围

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

描述

HASH_PINNABLE_CACHE_PERCENT 用于控制用户临时内存分配中任何一个散列对象可以固定的内存百分比。缺省值为 20%，但在运行复杂查询时应将此数字减少为 10%，或在使用只需一个大型散列对象即可运行的简单查询（如大型 **IN** 子查询）时将其增加至 50%。

HASH_PINNABLE_CACHE_PERCENT 主要供 Sybase 技术支持部门使用。如果更改该选项的值，应当谨慎行事；首先要分析将会对多种查询产生的影响。

另请参见

- **BIT_VECTOR_PINNABLE_CACHE_PERCENT** 选项（第 340 页）
- **SORT_PINNABLE_CACHE_PERCENT** 选项（第 432 页）

HASH_THRASHING_PERCENT 选项

指定对于包括涉及散列算法的查询的语句，在其执行过程中允许的硬盘 **I/O** 数百分比，达到该百分比后将回退语句并报告错误消息。

允许值

0 - 100

默认值

10

范围

可针对单个连接或 **PUBLIC** 组进行设置。设置将立即生效。

描述

如果某个使用散列算法的查询产生过大的硬盘 I/O 数（从内存到磁盘的分页缓冲数），则会对查询性能产生负面影响，还可能使服务器的性能受到影响。

`HASH_THRASHING_PERCENT` 可以控制在语句回退并返回错误消息之前允许的硬盘 I/O 百分比。错误消息的文本为 `Hash insert thrashing detected`（检测到散列插入抖动）或 `Hash find thrashing detected`（检测到散列查找抖动）。

`HASH_THRASHING_PERCENT` 的缺省值为 10%。增大此值将允许在回退之前对磁盘进行更多分页，而减小此值将减少在回退之前允许的分页。

有关控制过多分页和使用 `HASH_THRASHING_PERCENT` 的详细信息，请参见《系统管理指南第一卷》>“故障排除提示”>“特定情况的解决方案”>“处理问题”>“意外的长期装载或查询”。

另请参见

- `HASH_PINNABLE_CACHE_PERCENT` 选项（第 375 页）

HG_DELETE_METHOD 选项

指定在 **HG** 索引中进行删除时所使用的算法。

允许值

0 - 3

默认值

0

范围

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

描述

此选项用于选择 **HG** 索引在删除操作过程中使用的算法。开销模型在选择相应的删除算法时会考虑 CPU 相关开销以及 I/O 相关开销。开销模型考虑下列各项：

- 删除行
- 索引大小
- 索引数据类型的宽度
- 索引数据的基数
- 可用临时高速缓存
- 与计算机相关的 I/O 和 CPU 特性
- 可用 CPU 和线程
- 参照完整性开销

要强制实施“小型”方法，则将此选项设置为 1。要强制实施“大型”方法，则将此选项设置为 2。要强制实施“中型”方法，则将此选项设置为 3。

HG_SEARCH_RANGE 选项

指定在对 **HG** 索引中的范围谓词进行评估时使用的 **BTree** 页数的上限。

允许值
整数

默认值
10

范围
可针对单个连接或 **PUBLIC** 组进行设置。设置将立即生效。

描述
此选项的缺省设置适用于大多数查询。

此选项可有效地控制优化程序在搜索要用于范围谓词的最佳索引所用的时间。为此选项设置更高的值可能会导致在优化程序中的查询时间更长，但这样可能会选择一个更好的索引来解决范围谓词问题。

HTTP_SESSION_TIMEOUT 选项

指定客户端放弃之前等待 **HTTP** 会话超时的时间（以分钟为单位）。

允许值
整数 (0 - 525600)

默认值
30

范围
需要具有 **DBA** 权限。只能为 **PUBLIC** 组设置。

描述
此选项为 **Web** 服务应用程序提供可变会话超时控制。**Web** 服务应用程序可以在任何拥有 **HTTP** 会话的请求内更改超时值，但对超时值的更改可能会在 **HTTP** 会话超时后影响队列中的后续请求。**Web** 应用程序必须包含逻辑以检测客户端是否尝试访问已不存在的 **HTTP** 会话。这可以通过检查 **SessionCreateTime** 连接属性值以确定时间戳是否有效来完成：如果 **HTTP** 请求与当前 **HTTP** 会话不相关联，则 **SessionCreateTime** 连接属性包含空字符串。

另请参见

另请参见“SQL Anywhere Server - 编程” > “HTTP Web 服务” > “使用 SQL Anywhere 作为 HTTP Web 服务器”。

请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “连接属性、数据库属性和数据库服务器属性” > “连接属性”中的 **SessionCreateTime** 和 **http_session_timeout** 属性。

注意： 这些参考指向 SQL Anywhere 文档。

IDENTITY_ENFORCE_UNIQUENESS 选项

针对每个 IDENTITY/AUTOINCREMENT 列创建唯一的 **HG** 索引（前提是该列尚未成为主键）。

允许值

ON、OFF

默认值

OFF

范围

只能针对用户或 PUBLIC 组进行临时设置（对于连接）。设置将立即生效。

描述

当此选项设置为 ON 时，会针对将来的标识列创建 **HG** 索引。只有满足以下条件才能删除索引：进行删除的用户是唯一使用该表的用户，且该表不是本地临时表。

另请参见

- QUERY_PLAN 选项（第 421 页）

IDENTITY_INSERT 选项

使用户可以在 IDENTITY 或 AUTOINCREMENT 列中插入值或对该列进行更新。

允许值

= 'tablename'

默认值

选项未设置。

范围

只能针对用户或 PUBLIC 组进行临时设置（对于连接）。设置将立即生效。

注意： 如果为当前选项设置用户级选项，则还会设置相应的临时选项。请参见“数据库选项的作用域和持续时间”。

描述

设置 `IDENTITY_INSERT` 后，可以进行插入/更新。必须指定表名才能标识要插入或更新的列。如果您不是表的所有者，请将表名限定为所有者名称。

要删除包含 `IDENTITY` 列的表，`IDENTITY_INSERT` 不能设置为指向该表。

示例

如果使用表 `Employees` 来运行显示插入：

```
SET TEMPORARY OPTION IDENTITY_INSERT = 'Employees'
```

要禁用此选项，请指定等号和空字符串：

```
SET TEMPORARY OPTION IDENTITY_INSERT = ''
```

在以 `DBA` 身份连接到数据库的情况下，演示用户级选项对临时选项的影响（请参见“注释”）：

```
SET OPTION IDENTITY_INSERT = 'Customers'
```

对于用户 `DBA`，该选项的值设置为 `Customers`，该值是当前连接的临时设置。对于随后以 `DBA` 身份连接到该数据库的其他用户，其 `IDENTITY_INSERT` 选项值也为 `Customers`。

另请参见

- 数据库选项的作用域和持续时间（第 317 页）
- `QUERY_PLAN` 选项（第 421 页）

INDEX_ADVISOR 选项

生成相应消息，这些消息建议增加列索引以提高一个或多个查询的性能。

允许值

ON、OFF

默认值

OFF

范围

可针对用户或 `PUBLIC` 组进行临时设置（对于连接）。设置将立即生效。

描述

当此选项设置为 `ON` 时，索引顾问会在 Sybase IQ 查询计划中打印索引建议，如果未启用查询计划，则将其作为 Sybase IQ 消息日志文件中单独的消息进行打印。这些消息以字符串“`Index Advisor:`”开头，并且您可以使用该字符串在 Sybase IQ 消息文件中对其进行搜索和过滤。输出格式为 `OWNER.TABLE.COLUMN`。

设置 `INDEX_ADVISOR` 和 `INDEX_ADVISOR_MAX_ROWS` 可累计索引建议。

注意： 当 **INDEX_ADVISOR_MAX_ROWS** 设置为 **ON** 时，索引建议不会作为 单独的消息写入 **Sybase IQ** 消息文件。但 **Sybase IQ** 消息文件中的查询计划中 仍会显示建议。

表 25. 索引顾问

情况	建议
单列中的本地谓词（根据情况需要 HG 、 LF 、 HNG 、 DATE 、 TIME 或 DATETIME 索引）。	建议将 <index-type> 索引添加到列 <col> 中
单列连接键（适用 LF 或 HG 索引）。	将 LF 或 HG 索引添加到连接键 <col> 中
单列候选键索引（存在一个 HG ，但可以更改为唯一的 HG 或） LF	将连接键 <col> 更改为唯一的 LF 或 HG 索引
连接键的数据类型不匹配，重新生成具有匹配数据类型的一列会有所帮助。	使连接键 <col1> 和 <col2> 的数据类型相同
子查询谓词列（适用 LF 或 HG 索引）。	将 LF 或 HG 索引添加到子查询列 <col> 中
分组列（适用 LF 或 HG 索引）。	为分组列 <col> 创建 LF 或 HG 索引
单表列间比较（两列的数据类型相同，建议使用 CMP 索引）。	为 <col1>、<col2> 创建 CMP 索引
若干列（存在一个 LF 或 HG 索引，不同值的数量符合要求，建议将 FP 转换为一个或两个字节的 FP 索引）。	使用 “optimize storage=on” 的设置重建 <col>
要支持查找宽度为 3 个字节的缺省索引	使用 IQ UNIQUE 约束值 65537 重建 FP 索引作为 3 字节的 FP

您可以自行决定从附加索引中获益的查询的数量，以及创建并维护索引的开销是否值得。在某些情况下，无法确定添加建议的索引会带来多少性能上的提高（如果有）。

例如，假定有若干用作连接键的列。**Sybase IQ** 大量使用 **HG** 或 **LF** 索引提供的 元数据来生成更好/更快 的查询计划以执行查询。在没有 **HG** 或 **LF** 索引的连接列中 放置这样一个索引会使 **IQ** 优化程序更有希望 选择更快的连接计划，但如果不添加索引 而再次运行查询，将很难确定查询性能是 保持不变还是因新索引而有所提高。

示例

查询计划设置为 **OFF** 时的索引顾问输出：

```
I. 03/30 14:18:45. 00000000002 Advice: Add HG or LF index
on DBA.ta.c1 Predicate: (ta2.c1 < BV(1))
```

查询计划设置为 **ON** 时的索引顾问输出：

注意： 此方法可累计多个查询的索引顾问信息，这样随着时间的推移，可以在中心位置跟踪若干查询的建议。

```
I. 03/30 14:53:24. 0000000008 [20535]: 6      ...#03: Leaf
I. 03/30 14:53:24. 0000000008 [20535]:      Table Name: tb
I. 03/30 14:53:24. 0000000008 [20535]:      Condition 1
(Invariant):
(tb.c3 =tb.c4)
I. 03/30 14:53:24. 0000000008 [20535]:      Condition 1 Index
Advisor:
Add a CMP index on DBA.tb (c3,c4)
```

另请参见

- FP_LOOKUP_SIZE 选项（第 370 页）
- INDEX_ADVISOR_MAX_ROWS 选项（第 381 页）
- MINIMIZE_STORAGE 选项（第 405 页）
- QUERY_PLAN 选项（第 421 页）

INDEX_ADVISOR_MAX_ROWS 选项

将索引顾问存储的最大唯一提示消息数设置为 `max_rows`。

允许值

值	描述
0	最小值将禁用对索引建议的收集
4294967295	允许的最大值

默认值

0

范围

可以临时设置（对当前连接），也可以为用户/组（例如 `PUBLIC` 或 `DBA`）持久存在。设置将立即生效。

描述

INDEX_ADVISOR_MAX_ROWS 限制由索引顾问存储的消息数。一旦达到指定限制，**INDEX_ADVISOR** 就不再存储新的建议。但是，它将继续更新现有建议消息的计数和时间戳。

```
SET OPTION public.Index_Advisor_Max_Rows = max_rows;
```

另请参见

- FP_LOOKUP_SIZE 选项（第 370 页）
- INDEX_ADVISOR 选项（第 379 页）

INDEX_PREFERENCE 选项

控制对用于查询的索引的选择。

允许值

-10 到 10

默认值

0

范围

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

描述

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

表 26. INDEX_PREFERENCE 有效值

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

值	操作
-4	避免 CMP 索引
-5	避免缺省索引
-6	避免 WD 索引
-8	避免 DATE 索引
-9	避免 TIME 索引
-10	避免 DTTM 索引

INFER_SUBQUERY_PREDICATES 选项

控制优化程序的其它子查询谓词的推断。

允许值

ON、OFF

默认值

ON

范围

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

描述

INFER_SUBQUERY_PREDICATES 控制是否允许优化程序跨越简单等同连接谓词，通过传递闭包，从现有子查询谓词推断其它子查询谓词。在优化程序选择进行此推断的大多数情况中，查询运行速度会更快。这种性能改进有某些例外，因此可能需要进行实验，以确保此选项适合具体环境。

IN_SUBQUERY_PREFERENCE 选项

控制用于处理 IN 子查询的算法的选择。

允许值

-3 到 3

默认值

0

范围

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

描述

IQ 优化程序有一个选项，可以选择处理 **IN** 子查询的几个算法。此选项允许您在选择要使用的算法时覆盖优化程序的开销决策。它不会覆盖用于确定算法在查询引擎内是否合法的内部规则。

IN_SUBQUERY_PREFERENCE 通常用于内部测试，以及用于手动调优优化程序处理不好的查询。只有经验丰富的 **DBA** 才能使用。使用此选项的唯一原因是如果优化程序严重低估了子查询所产生的行数，并且散列对象正在抖动。在设置此选项之前，请尝试通过查找丢失的索引和依赖性谓词来改进错误的估计。

如果需要设置 **IN_SUBQUERY_PREFERENCE**，请通知 Sybase 技术支持部门，因为设置此选项可能意味着相应地更改优化程序。

表 27. **IN_SUBQUERY_PREFERENCE** 有效值

值	操作
0	让优化程序选择
1	首选基于排序的 IN 子查询
2	首选纵向 IN 子查询（其中，子查询是查询计划中的叶节点的子节点）
3	首选基于散列的 IN 子查询
-1	避免基于排序的 IN 子查询
-2	避免纵向 IN 子查询
-3	避免基于散列的 IN 子查询

IQGOVERN_MAX_PRIORITY 选项

限制允许的 **IQGOVERN_PRIORITY** 设置。

允许值

1 - 3

默认值

2

范围

可针对单个连接或 **PUBLIC** 组进行临时设置。设置此选项需要具有 **DBA** 权限。设置将立即生效。

描述

限制允许的 **IQGOVERN_PRIORITY** 设置，这将影响到用户的查询排队等待执行的顺序。在允许值的范围中，1 表示高优先级、2（缺省值）表示中优先级，3 表示低

优先级。如果用户设置比 `IQGOVERN_MAX_PRIORITY` 更高的 `IQGOVERN_PRIORITY`，则 Sybase IQ 返回错误。

另请参见

- `IQGOVERN_PRIORITY` 选项（第 385 页）
- `IQGOVERN_PRIORITY_TIME` 选项（第 385 页）

IQGOVERN_PRIORITY 选项

为在 `-iqgovern` 队列中等待的每个查询分配优先级。

允许值

1 - 3

默认值

2

范围

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

描述

分配一个值，用于确定用户的查询排队等待执行的顺序。在允许值的范围内，1 表示高优先级，2（缺省值）表示中优先级，3 表示低优先级。可以由任何用户为每个用户或 `Public` 临时设置此开关。只有在已执行所有更高优先级的查询后，才会运行具有更低优先级的查询。

此选项受选项 `IQGOVERN_MAX_PRIORITY` 的每个用户或每个组的值限制。

另请参见

- `IQGOVERN_MAX_PRIORITY` 选项（第 384 页）
- `IQGOVERN_PRIORITY_TIME` 选项（第 385 页）

IQGOVERN_PRIORITY_TIME 选项

限制高优先级查询在启动之前在队列中的等待时间。

允许值

0 - 1,000,000 秒。必须小于 `IQGOVERN_MAX_PRIORITY`。

默认值

0（禁用）

范围

只能为 `PUBLIC` 组设置。需要具有 DBA 权限。设置将立即生效。

描述

限制高优先级（优先级 1）查询在启动之前在队列中的等待时间。当达到限制时，即使查询超过 **-iqgovern** 设置所允许的查询数，也会启动查询。您必须属于 DBA 组，才能更改此开关。范围是从 1 到 1,000,000 秒。缺省值 (0) 将禁用此功能。

IQGOVERN_PRIORITY_TIME 必须设置 PUBLIC。

另请参见

- IQGOVERN_MAX_PRIORITY 选项（第 384 页）
- IQGOVERN_PRIORITY 选项（第 385 页）

ISOLATION_LEVEL 选项

控制目录存储表的锁定隔离级别。

允许值

0、1、2 或 3

默认值

0

描述

每个锁定隔离级别定义如下：

- 0 – 允许脏读、非重复读取和幻像行。
- 1 – 阻止脏读。允许非重复读取和幻像行。
- 2 – 阻止脏读并保证可重复读取。允许幻像行。
- 3 – 可序列化。不允许脏读，保证可重复读取，并且不允许幻像行。

ISOLATION_LEVEL 确定目录存储中表的 隔离级别。Sybase IQ 始终对 IQ 存储中的表强制应用级别 3。级别 3 等同于 ANSI 级别 4。

JAVA_LOCATION 选项

为数据库指定 Java VM 的路径。

允许值

字符串

默认值

空字符串

范围

只能为 PUBLIC 组设置。需要具有 DBA 权限。

描述

缺省情况下，此选项包含空字符串。在此情况下，数据库服务器在 JAVA_HOME 环境变量、路径和其它位置搜索 Java VM。

另请参见

- JAVA_VM_OPTIONS 选项（第 387 页）

JAVA_VM_OPTIONS 选项

指定供数据库服务器在启动 Java VM 时使用的命令行选项。

允许值

字符串

默认值

空字符串

范围

只能为 PUBLIC 组设置。需要具有 DBA 权限。

描述

JAVA_VM_OPTIONS 指定数据库服务器在启动由 JAVA_LOCATION 选项指定的 Java VM 时使用的命令行选项。这些附加选项可以用于设置 Java VM 以实现调试目的或作为 UNIX 平台上的服务运行。在某些情况下，需要有附加选项才能在 64 位模式（而不是 32 位模式）中使用 Java VM。

请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “数据库选项” > “按字母顺序排列的选项列表” > “java vm 选项”。

注意： 此参考指向 SQL Anywhere 文档。

另请参见

- JAVA_LOCATION 选项（第 386 页）

JOIN_EXPANSION_FACTOR 选项

控制在少见的复杂情况中对优化程序的连接结果进行估计的保守程度。

允许值

1 - 100

默认值

30

范围

可针对单个连接或 `PUBLIC` 组进行临时设置。设置将立即生效。

描述

此选项控制在对特定连接的输入已经经过了至少一个可以导致行的多个副本从正在连接的表中凸显出来的中间连接的情况下，对连接优化程序的结果大小进行估计时的保守程度。

零级别指示优化程序应当使用与如果没有中间扩展连接则会使用的估计方法相同的、在中间扩展连接以上的估计方法。

这将导致最积极（小）的估计连接结果大小。

100 级别指示一旦有中间扩展连接则优化程序应当在估计时更保守，并且这将导致最保守（大）的估计连接结果大小。

通常，不需要更改该值。如果更改，则 Sybase 建议将 `JOIN_EXPANSION_FACTOR` 设置为临时或用户选项。

JOIN_OPTIMIZATION 选项

启用或禁用连接顺序的优化。

允许值

ON、OFF

默认值

ON

范围

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

描述

当 `JOIN_OPTIMIZATION` 选项是 `ON` 时，Sybase IQ 将优化连接顺序，以减少中间结果和排序的大小，并平衡系统负载。当选项是 `OFF` 时，连接顺序由 **SELECT** 语句的 **FROM** 中的包的顺序决定。

`JOIN_OPTIMIZATION` 应始终设置为 `ON`。

`JOIN_OPTIMIZATION` 控制连接的顺序，但不控制表的顺序。为了显示差别，下面有四个表的 **FROM** 子句为例：

```
FROM A, B, C, D
```

缺省情况下，此 **FROM** 子句将创建还可以显式表示为以下示例的连接的左深计划：

```
FROM ((A, B), C), D)
```

如果关闭 `JOIN_OPTIMIZATION`，则会按照 **FROM** 子句中指定的顺序准确保持这些连接在表集合上的顺序。因此 **A** 和 **B** 必须首先连接，然后该结果必须连接到表 **C**，最终连接到表 **D**。此选项不会控制每个连接的左/右方向。甚至在 `JOIN_OPTIMIZATION` 关闭时，在假定有上面的 **FROM** 子句时，优化程序也可以产生像下面这样的连接计划：

```
FROM ((C, (A, B)), D)
```

或

```
FROM (((B, A), C), D)
```

或

```
FROM (D, ((A, B), C))
```

在所有这些情况中，**A** 和 **B** 首先连接，然后该结果连接到 **C**，最终该结果连接到表 **D**。连接的顺序保持不变，但表的出现顺序不同。

通常，如果关闭 `JOIN_OPTIMIZATION`，则可能应当在 **FROM** 子句中使用圆括号（与上面的示例一样），以确保获取所需的连接顺序。如果要将 **A** 和 **B** 连接到 **C** 和 **D** 的连接结果，则可以通过使用圆括号指定此连接：

```
FROM ((A, B), (C, D))
```

注意，即使所有表以相同顺序出现，上面的 **FROM** 子句的连接顺序也不同于原始 **FROM** 子句示例。

`JOIN_OPTIMIZATION` 仅仅是为了诊断模糊连接性能问题，或为了手动优化少量预定义的查询，就应当将设置为 **OFF**。在 `JOIN_OPTIMIZATION` 关闭时，查询可以连接最多 128 个表，但也可能遭受严重的性能降级。

警告！ 如果关闭 `JOIN_OPTIMIZATION`，则 Sybase IQ 无法确保包含连接的查询有最佳性能。Y 您要对查询的性能方面完全负责。

JOIN_PREFERENCE 选项

控制处理连接时的算法选择。

允许值

-7 到 7

默认值

0

范围

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

描述

对于查询中的连接，IQ 优化程序有几个用于处理连接的算法可供选择。
JOIN_PREFERENCE 允许您在选择要使用的算法时覆盖优化程序的基于成本的决策。它不会覆盖用于确定算法在查询引擎内是否合法的内部规则。如果将它设置为任何非零值，则查询中的每个连接都会受影响；您不能在查询中使用它有选择地修改几个连接中的某一个。

此选项通常用于内部测试，只有经验丰富的 DBA 才能使用。

表 28. JOIN_PREFERENCE 有效值

值	操作
0	让优化程序选择
1	首选排序合并
2	首选嵌套循环
3	首选嵌套循环叠加
4	首选散列
5	首选散列叠加
6	首选预连接
7	首选排序合并叠加
-1	避免排序合并
-2	避免嵌套循环
-3	避免嵌套循环叠加
-4	避免散列
-5	避免散列叠加
-6	避免预连接
-7	避免排序合并叠加

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')
```

JOIN_SIMPLIFICATION_THRESHOLD 选项

控制在应用任何连接优化程序简化机制之前被连接在一起的最小表数。

允许值

1 - 64

默认值

15

范围

可针对单个连接或 PUBLIC 组进行临时设置。设置将立即生效。

描述

通过分别处理查找表（即非选择性维度表）和有效的笛卡儿乘积的表，查询优化程序可简化对连接顺序的优化。在简化之后，它将优化剩余表的连接顺序，直到达到由 MAX_JOIN_ENUMERATION 设置的限制。

如果将此选项设置为大于 MAX_JOIN_ENUMERATION 的当前值的其它某个值，则操作无效。

将此值设置为小于 MAX_JOIN_ENUMERATION 的值，可能改进在优化那些包含很多连接的查询时所需要的时间，但也可能阻止优化程序找到最佳的可能连接计划。

通常，不需要更改该值。如果更改，则 Sybase 建议将 JOIN_SIMPLIFICATION_THRESHOLD 设置为临时或用户选项，并设置为至少为 9 的某个值。

另请参见

- MAX_JOIN_ENUMERATION 选项（第 401 页）

LARGE_DOUBLES_ACCUMULATOR 选项

控制使用哪个累计器计算浮点数字的 SUM 或 AVG。

允许值

ON、OFF

默认值

OFF

范围

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

描述

浮点型和双精度型的小型累计器在处理 1e-20 到 1e20 的幅度范围内的加数时准确度很高。在此范围以外它会失去一些准确性，但对很多应用程序仍然足够好。小型累计器允许优化程序选择散列，以便更容易获得比大型累计器更快的性能。大型累计器对所有浮点型和双精度型的准确性都很高，但其大小通常会妨碍对散列优化的使用。缺省是小型累计器。

LF_BITMAP_CACHE_KB 选项

指定将用于装载到 LF 索引中的内存数量。

允许值

1 - 8

默认值

4

范围

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

描述

LF_BITMAP_CACHE_KB 定义在装载到 LF 索引中时每个非重复值所使用的堆内存数量 (KB)。缺省分配 4KB。如果特定表的所有 LF 索引的离散值个数之和相对较高（大于 10,000），那么堆内存使用量可能增加到由于系统页错误而会影响装载性能的程度。如果是这种情况，请减少 LF_BITMAP_CACHE_KB 的值。

以下公式显示如何计算在装载期间特定 LF 索引所使用的堆内存（字节）：

```
Heap-memory-used = (lf_bitmap_cache_kb * 1024)
* lf-distinct-count-for-column
```

如果使用缺省值 4KB，则具有 1000 条非重复值的 LF 索引可以在装载期间使用最多 4MB 堆内存。

LOAD_ZEROLENGTH_ASNULL 选项

指定 LOAD 语句在某些情况下的行为。

允许值

ON、OFF

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

默认值

OFF

描述

此选项指定 LOAD 语句在以下情况下的行为：

- 将零长度数据值插入数据类型为 CHAR、VARCHAR、LONG VARCHAR、BINARY、VARBINARY 或 LONG BINARY 的列中
- NULL column-spec；例如，还为该相同列提供 NULL (ZEROS) 或 NULL (BLANKS)

将 LOAD_ZEROLENGTH_ASNULL 设置为 ON 会在符合上面的条件时装载零长度值作为 NULL。

如果将 LOAD_ZEROLENGTH_ASNULL 设置为 OFF，则会装载零长度值作为零长度，并服从于选项 NON_ANSI_NULL_VARCHAR 的设置。

另请参见

- NON_ANSI_NULL_VARCHAR 选项（第 410 页）
- LOAD TABLE 语句（第 223 页）

LOCKED 选项

如果设置登录策略，则防止具有该策略的用户建立新连接。

有关详细信息，请参见《系统管理指南第一卷》>“管理用户 ID 和权限”。

另请参见

- CREATE LOGIN POLICY 语句（第 108 页）
- ALTER LOGIN POLICY 语句（第 19 页）

LOG_CONNECT 选项

控制用户连接的记录。

允许值

ON、OFF

默认值

ON

范围

只能为 PUBLIC 组设置。设置将立即生效。

描述

此选项是 ON 时，每次用户连接到 Sybase IQ 数据库或与其断开时，将在 IQ 消息日志（.iqmsg 文件）中记录一条消息。

注意： 如果用户连接时将此选项设置为 **OFF**（禁用连接记录），然后在用户断开连接之前将其打开，则消息日志将显示该用户断开但未连接。

LOG_CURSOR_OPERATIONS 选项

控制游标操作的记录。

允许值

ON、OFF

默认值

OFF

范围

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

描述

此选项是 **ON** 时，每次您打开或关闭游标时，将在 **IQ** 消息日志中记录一条消息。正常情况下，此选项应当是 **OFF**，这是缺省值。仅当遇到问题并且必须向 **Sybase** 技术支持部门提供调试数据时，才应打开它。

LOGIN_MODE 选项

控制对数据库的集成登录的使用。

允许值

Standard、Mixed 或 Integrated

默认值

标准

范围

只能为 **PUBLIC** 组设置。设置将立即生效。

描述

此选项指定是否允许集成登录。值不区分大小写：

- **Standard** – 缺省设置，它不允许集成登录。如果尝试集成登录连接，则会发生错误。
- **Mixed** – 允许集成登录和标准登录。
- **Integrated** – 如果使用此设置，则必须使用集成登录执行所有登录到数据库的操作。

警告！ 如果将 LOGIN_MODE 设置为 Integrated，则只有那些已被授予集成登录映射的用户才能进行连接。试图使用用户 ID 和口令进行连接时生成错误。这种情况的唯一例外是具有 DBA 授权（完全管理权限）的用户。

LOGIN_PROCEDURE 选项

指定在启动时设置连接兼容性选项的登录过程。

允许值
字符串

默认值

sp_login_environment 系统过程

范围

可针对单个连接或 PUBLIC 组进行设置。设置此选项需要具有 DBA 权限。设置将立即生效。

描述

初始连接兼容性选项设置可使用 LOGIN_PROCEDURE 选项进行控制，该选项在执行所有检查验证连接是否有效后调用。LOGIN_PROCEDURE 选项命名了一个存储过程以在用户连接时使用。缺省设置为使用 **sp_login_environment** 系统存储过程。您可以指定不同的存储过程。对于事件连接，不执行 LOGIN_PROCEDURE 选项指定的过程。

sp_login_environment 过程将检查是否正在通过 TDS 建立连接。如果通过 TDS 建立连接，则 **sp_login_environment** 调用 **sp_tsql_environment** 过程，该过程会针对当前连接将几个选项设置为新的缺省值。

有关 LOGIN_PROCEDURE 选项和示例的详细信息，请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “数据库选项” > “数据库选项 简介” > “按字母顺序排列的选项列表” > “按字母顺序排列的选项列表” > “login_procedure 选项”。

注意： 此参考指向 SQL Anywhere 文档。

另请参见

- 初始选项设置（第 319 页）

MAIN_RESERVED_DBSPACE_MB 选项

控制 Sybase IQ 在 IQ main 存储中保留的空间数量。

允许值

大于或等于 200 的整数（以兆字节为单位）

默认值

200; Sybase IQ 实际保留 IQ_SYSTEM_MAIN 中最后一个读写文件的最多 50% 和最少 1%

范围

只能为 PUBLIC 组设置。设置此选项需要具有 DBA 权限。设置将立即生效。不需要重新启动服务器即可更改保留空间大小。

描述

MAIN_RESERVED_DBSPACE_MB 控制 Sybase IQ 在 IQ main 存储中 为在发布保存点、提交和检查点操作期间所使用的 某些小型但关键的数据结构留出的空间数量。对于生产数据库， 请将该值设置为介于 200MB 和 1GB 之间的值。IQ 页大小 和并发连接数越大， 需要的保留空间就越多。

保留的空间大小是通过计算 IQ_SYSTEM_MAIN 中最后一个读写文件的最多 50% 和最少 1% 得出的。

MAX_CARTESIAN_RESULT 选项

限制从笛卡儿连接产生的行数。

允许值

任意整数

可以为用户或 PUBLIC 组进行临时设置（对于连接）。设置将立即生效。

默认值

100000000

范围

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

描述

MAX_CARTESIAN_RESULT 可以限制 从包含笛卡儿连接（通常是创建查询时 丢失一个或多个连接条件的结果） 的查询所得到的结果行数。 如果 Sybase IQ 找不到包含在此限制以下的 估计结果的笛卡儿连接的查询计划， 则它会拒绝查询并返回错误。 如果将 MAX_CARTESIAN_RESULT 设置为 0， 则会禁用对笛卡儿连接的结果行数的检查。

MAX_CLIENT_NUMERIC_PRECISION 选项

控制发送到客户端的数值数据的最大精度。

允许值

0 - 126

默认值

0

范围

可由任何用户在任何级别进行设置。此选项会立即生效。

描述

Sybase IQ 执行其计算时，会将数据类型提升到合适大小，以确保准确性。所提升的数据类型的大小可能超过 Open Client 和某些 ODBC 应用程序可以正确处理的范围。

当 MAX_CLIENT_NUMERIC_PRECISION 是非零值时，Sybase IQ 将检查数字结果列是否超过此值。如果结果列大于 MAX_CLIENT_NUMERIC_PRECISION 允许的值，并且 Sybase IQ 无法将它转换到指定的精度，则查询会返回以下错误：

```
"Data Exception - data type conversion is not possible %1 SQLCODE = -1001006" ( "数据例外 - 无法执行数据类型转换 %1 SQLCODE = -1001006" )
```

注意：在 SQL Anywhere 中，数值函数支持的最大值是 255。如果数值函数的精度超过支持的最大值，则会出现以下错误：“函数 ‘_funcname’ 的结果数据类型超过支持的 最大数值精度 255。请为数值函数 ‘location’ 的精度设置正确的值”

另请参见

- MAX_CLIENT_NUMERIC_SCALE 选项（第 397 页）
- PRECISION 选项（第 416 页）

MAX_CLIENT_NUMERIC_SCALE 选项

控制发送到客户端的数值数据的最大标度。

允许值

0 - 126

默认值

0

范围

可由任何用户在任何级别进行设置。此选项会立即生效。

描述

Sybase IQ 执行其计算时，会将数据类型提升到合适的标度和大小，以确保准确性。所提升的数据类型可能大于原始定义的数据大小。可以将此选项设置为要为数字结果指定的标度。

乘、除、加、减和集合函数都可以有超过最大精度和标度的结果。

例如，将 DECIMAL(59,2) 乘以 DECIMAL(88,2)，结果可能需要 DECIMAL(147,4)。如果 MAX_CLIENT_NUMERIC_PRECISION 是 126，则仅在结果中保留 126 位数。如果 MAX_CLIENT_NUMERIC_SCALE 是 4，则结果作为 DECIMAL(126,4) 返回。如果 MAX_CLIENT_NUMERIC_SCALE 是 2，则结果作为 DECIMAL(126,2) 返回。在这两种情况中，都有溢出的可能性。

另请参见

- MAX_CLIENT_NUMERIC_PRECISION 选项（第 397 页）
- SCALE 选项（第 430 页）

MAX_CONNECTIONS 选项

指定用户允许的最大并发连接数。

有关详细信息，请参见《SQL Anywhere Server - 数据库管理》。

MAX_CUBE_RESULT 选项

设置 IQ 优化程序为 **GROUP BY CUBE** 算法考虑的最大行数。

允许值

0 - 4294967295

默认值

10000000

范围

可由任何用户在任何级别进行设置。此选项会立即生效。

描述

生成查询计划时，IQ 优化程序会估计通过 **GROUP BY CUBE** 散列运算生成的组的总数。IQ 优化程序使用散列算法执行 **GROUP BY CUBE** 操作。此选项设置优化程序为可以运行的散列算法考虑的估计行数的上限。如果实际行数超过 MAX_CUBE_RESULT 值，优化程序将停止处理查询，并返回错误“Estimate number: *nnn* exceeds the default MAX_CUBE_RESULT of GROUP BY CUBE or ROLLUP”（估计数目：超过 GROUP BY CUBE 或 ROLLUP 的缺省 DEFAULT_MAX_CUBE_RESULT），其中 *nnn* 是 IQ 优化程序估计的数值。

将 `MAX_CUBE_RESULT` 设置为零将覆盖缺省值。将此选项设置为零时，IQ 优化程序将不检查行限制，并允许查询运行。建议不要将 `MAX_CUBE_RESULT` 设置为零，因为查询可能不成功。

MAX_CURSOR_COUNT 选项

指定资源调控器，以限制连接可以一次使用的最大游标数。

允许值
整数

默认值
50

范围
可针对单个连接或 `PUBLIC` 组进行设置。设置将立即生效。需要具有 `DBA` 权限才能针对所有连接设置此选项。

描述
所指定的资源调控器允许 `DBA` 限制用户可以拥有的每个连接上的游标数。如果某个操作超过对连接的限制，则会生成一个错误，指示已超过该限制。

如果连接执行一个存储过程，则该过程会在过程所有者的权限下执行。然而，由该过程所用的资源将分配给当前连接。

通过将 `MAX_CURSOR_COUNT` 设置为 0（零），可以删除资源限制。

MAX_DAYS_SINCE_LOGIN 选项

指定同一用户在两次连续登录之间经过的最大天数。

有关详细信息，请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “管理用户 ID 和权限” > “管理登录策略”。

注意： 此参考指向 SQL Anywhere 文档。

MAX_FAILED_LOGIN_ATTEMPTS 选项

指定自上一次成功尝试以来，要在锁定帐户前登录到用户帐户的最大失败尝试次数。

有关详细信息，请参见“SQL Anywhere Server - 数据库管理” > “配置数据库” > “管理用户 ID 和权限” > “管理登录策略”。

注意： 此参考指向 SQL Anywhere 文档。

对于指定用户，Multiplex 环境中的每个节点都会保留其自己的失败登录计数和最后一次失败登录时间。

在 Sybase IQ 中, `MAX_FAILED_LOGIN_ATTEMPTS` 适用于所有用户, 包括具有 DBA 权限的用户。Sybase IQ 服务器在最后一次失败登录尝试 15 分钟之后, 会自动解锁具有 DBA 权限的用户帐户。

在以下情况下, 具有 DBA 权限的用户无法登录:

- 登录次数超出用户登录策略允许的最多失败登录尝试次数;
以及
- 用户当前登录时间和最后一次失败登录时间差距少于 15 分钟。

重新启动数据库后, 允许具有 DBA 权限的用户再进行一次成功登录尝试, 即使失败登录尝试次数已超出登录策略允许的最大次数也是如此。

MAX_HASH_ROWS 选项

设置 IQ 优化程序为散列算法考虑的最大行数。

允许值

从 1 到 4294967295 之间的整数

默认值

2500000

范围

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

描述

生成查询计划时, IQ 优化程序可能在处理查询的特定部分时有几个可供选择的算法(散列、排序、索引)。这些选项通常取决于要由该部分查询处理或生成的估计行数。此选项设置有多少估计行考虑使用散列算法的上限。

例如, 如果在两个表之间有连接, 并且从两个表进入连接的估计行数超过 `MAX_HASH_ROWS` 的值, 则优化程序不会考虑散列连接。在每个用户有超过 50MB 临时缓冲区高速缓存空间的系统上, 可能需要考虑让此选项有更高的值。

MAX_IQ_THREADS_PER_CONNECTION 选项

控制每个连接的线程数。

允许值

3 - 10000

默认值

144

范围

可以是临时的，也可以是永久的。设置此选项不需要 DBA 权限。只能为 PUBLIC 组设置。设置将立即生效。

描述

允许您对在连接上执行的命令所使用的线程数（以及由此导致的系统资源数量）进行约束。对于大多数应用程序，请使用缺省值。

MAX_IQ_THREADS_PER_TEAM 选项

控制为执行在连接中执行的单个操作（例如针对列的 **LIKE** 谓词）而分配的线程数。

允许值

1 - 10000

默认值

144

范围

可以是临时的，也可以是永久的。设置此选项不需要 DBA 权限。只能为 PUBLIC 组设置。设置将立即生效。

描述

允许您约束分配到单个操作的线程数（以及由此导致的系统资源数量）。此连接的所有同时执行的团队总数受相关选项 **MAX_IQ_THREADS_PER_CONNECTION** 的限制。对于大多数应用程序，请使用缺省值。

另请参见

- **MAX_IQ_THREADS_PER_CONNECTION** 选项（第 400 页）

MAX_JOIN_ENUMERATION 选项

控制在已应用优化程序简化之后要为连接顺序优化的最大表数。

允许值

1 - 64

每个 **FROM** 子句限制为最多只能具有 64 个表。但是实际上，**FROM** 子句中有效的表数限制通常低于该值，而且部分取决于这些表之间连接关系的复杂性。有效限制受 **MAX_JOIN_ENUMERATION** 的设置约束。优化程序将尝试简化 **FROM** 子句内连接关系的设置。如果这些简化无法将必须同步考虑的连接设置降低到不高于 **MAX_JOIN_ENUMERATION** 的当前设置，则查询将返回错误。

警告！ 将 **MAX_JOIN_ENUMERATION** 设置为高于缺省值 16 应谨慎执行，特别是在具有深度连接关系的查询导致优化程序所需的时间量大幅增加的情况下。对于仅使用线

性链连接关系的查询，**MAX_JOIN_ENUMERATION** 设置为 64 仍能提供合理的优化时间。

默认值

15

范围

可针对单个连接或 **PUBLIC** 组进行临时设置。设置将立即生效。

描述

通过分别处理查找表（即非选择性维度表）和有效的笛卡儿乘积的表，查询优化程序可简化对连接顺序的优化。在简化之后，它将继续优化剩余表的连接顺序，直到达到由 **MAX_JOIN_ENUMERATION** 设置的限制。如果超过此限制，则拒绝查询，并返回错误。然后，用户可以简化查询，或尝试增加限制。

通常，不需要更改该值。如果更改，则 Sybase 建议将 **MAX_JOIN_ENUMERATION** 设置为临时或用户选项。

MAX_PREFIX_PER_CONTAINS_PHRASE 选项

指定在文本搜索表达式中允许的前缀词语数目。

用户必须获得非结构化数据分析选项的使用许可才能使用 **TEXT** 索引以及执行全文本搜索。

有关 **MAX_PREFIX_PER_CONTAINS_PHRASE** 语法 和完整说明，请参见《Sybase IQ 中的非结构化数据分析》。

MAX_QUERY_PARALLELISM 选项

设置 **GROUP BY** 操作的并行执行以及 **UNION** 的分支的上限。

允许值

小于、大于或等于 CPU 数的整数。

默认值

64

范围

可针对单个连接或 **PUBLIC** 组进行临时设置。设置将立即生效。

描述

此参数设置限制优化程序将允许查询运算符进行并行程度的上限。该选项可能影响许多查询连接、**GROUP BY**、**UNION**、**ORDER BY** 和其它查询运算符的 CPU 使用情况。

具有超过 64 个 CPU 核心的系统通常受益于较大值，最多可达到系统上 CPU 核心的总数（即最大值 512）；您可以通过不断尝试来找出此参数适用于您的系统和查询的最佳值。

具有 64 个或更少 CPU 核心的系统不需要减少此值，除非显示了过多的系统时间。在这种情况下，您可以尝试减少此值以确定这种调整是否能够降低 CPU 系统时间、缩短查询响应时间和提高总体系统吞吐量。

MAX_QUERY_TIME 选项

设置时间限制，以便优化程序可以禁止非常长的查询。

允许值

0 到 $2^{32} - 1$ 分钟

默认值

0（禁用）

范围

可以在会话（临时）、用户或 PUBLIC 级别进行设置。

描述

如果查询运行时间超过 MAX_QUERY_TIME 设置，则 Sybase IQ 会停止查询，并将消息发送到用户和 IQ 消息文件。例如：

```
The operation has been cancelled -- Max_Query_Time exceeded.
```

MAX_QUERY_TIME 仅应用于查询，不应用于正在修改数据库内容的任何 SQL 语句。

MAX_STATEMENT_COUNT 选项

指定资源调控器，以限制连接可以一次使用的最大准备语句数。

允许值

整数

默认值

100

范围

可针对单个连接或 PUBLIC 组进行设置。设置将立即生效。需要具有 DBA 权限才能针对所有连接设置此选项。

描述

所指定的资源调控器允许 DBA 限制用户可以拥有的每个连接的准备语句数。如果某个操作超过对连接的限制，则会生成一个错误，指示已超过该限制。

如果连接执行一个存储过程，则该过程会在过程所有者的权限下执行。然而，由该过程所用的资源将分配给当前连接。

通过将 `MAX_STATEMENT_COUNT` 设置为 0（零），可以删除资源限制。

MAX_TEMP_SPACE_PER_CONNECTION 选项

限制每个连接使用的临时存储空间。

允许值

整数（以 MB 为单位）

默认值

0（不限制临时存储使用情况）

范围

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

描述

通过控制每个连接的空间，DBA 利用该选项可管理对空间的装载和查询。如果连接超过 `MAX_TEMP_SPACE_PER_CONNECTION` 指定的运行时间限额，Sybase IQ 会回退当前语句并将该消息返回到 IQ 消息文件或客户端用户：

```
The current operation has been cancelled:  
Max_Temp_Space_Per_Connection exceeded
```

可能填充缓冲区高速缓存的情况包括：读取或写入错误、主或临时空间不足或内存不足。Sybase IQ 可能返回在这些情况中遇到的第一个错误，DBA 必须确定相应的解决方案。有关详细信息，请参见“错误消息”和《系统管理指南第一卷》>“故障排除提示”。

在分布式查询处理事务中，Sybase IQ 将通过限制参与分布式查询的所有节点使用的共享空间和本地临时空间总量对于共享临时存储使用为 `QUERY_TEMP_SPACE_LIMIT` 和 `MAX_TEMP_SPACE_PER_CONNECTION` 选项设置的值。这意味着无论有多少节点参与，任何单一查询都不能超过临时空间总量限制（来自 `IQ_SYSTEM_TEMP` 和 `IQ_SHARED_TEMP` 数据库空间）。

例如，如果限制为 100，四个节点中每个使用 25 个单位量的临时空间，则查询位于限制内。不过，如果任一节点使用的空间总量超过 100，则查询将回退。

示例

为所有连接设置限制 500GB：

```
SET OPTION  
PUBLIC.MAX_TEMP_SPACE_PER_CONNECTION = 512000
```

为所有连接设置限制 10TB：

```
SET OPTION
PUBLIC.MAX_TEMP_SPACE_PER_CONNECTION = 10485760
```

为用户 wilson 设置限制 5000MB:

```
SET OPTION
wilson.MAX_TEMP_SPACE_PER_CONNECTION = 5000
```

另请参见

- QUERY_TEMP_SPACE_LIMIT 选项（第 426 页）

MAX_WARNINGS 选项

控制允许的最大警告数。

允许值
任意整数

默认值
 $2^{48} - 1$

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

描述
此选项可以限制在 DDL 命令期间有关拒绝的值、行不匹配等的警告数。缺省值不限制可以接收的数字。

MINIMIZE_STORAGE 选项

最小化最近创建列的磁盘空间使用量。

允许值
ON、OFF

默认值
OFF

范围
可以为 PUBLIC 组或临时使用而设置该选项。设置此选项无需 DBA 授权。此选项会立即生效。

描述
当 MINIMIZE_STORAGE 是 ON 时，IQ 将通过在只要合适则每行可仅使用一字节的磁盘空间，以优化对新列的存储。缺省情况下，此选项对 PUBLIC 组是 OFF，对所

有最近创建的列不执行专用的存储优化；当 `MINIMIZE_STORAGE` 对 `PUBLIC` 组是 `OFF` 但作为临时用户选项是 `ON` 时，对该用户 `ID` 创建的新列使用一字节存储。

将 `MINIMIZE_STORAGE` 设置为 `ON` 等同于在每个新列上放 **`IQ UNIQUE 255`** 子句，但某些一字节存储天生太宽的数据类型除外。当 `MINIMIZE_STORAGE` 设置为 `ON` 时，不必指定 **`IQ UNIQUE`**，具有 65536 个以上唯一值的列除外。

注意： `IQ UNIQUE` 值大于 65536 可以允许创建 3 字节索引，而以前当 `MINIMIZE_STORAGE` 是 `ON` 时此类值用于防止这种情况。如果当 `MINIMIZE_STORAGE` 是 `ON` 时想阻止专用的存储优化，请赋予 `IQ UNIQUE` 一个大于 16777216 的约束值。

当主内存与列数量的比值较大时，将 `MINIMIZE_STORAGE` 设为 `ON` 会有所帮助。否则，新列的存储通常受益于将此选项设为 `OFF`。

在 **`IQ UNIQUE`** 或 **`CREATE TABLE`** 中显式指定 **`ALTER TABLE ADD COLUMN`** 将覆盖该列的 `MINIMIZE_STORAGE` 设置。

另请参见

- `FP_LOOKUP_SIZE` 选项（第 370 页）
- `INDEX_ADVISOR` 选项（第 379 页）

MIN_PASSWORD_LENGTH 选项

设置数据库中新口令的最小长度。

允许值

大于或等于零的整数

值以字节为单位。对于单字节字符集，它与字符数相等。

默认值

0 个字符

范围

可以为 `PUBLIC` 组设置。设置将立即生效。设置此选项需要具有 `DBA` 权限。

描述

此选项允许 `DBA` 限制所有新口令的最小长度，以获得更大安全性。现有口令不受影响。

示例

将新口令的最小长度设置为 6 个字节：

```
SET OPTION PUBLIC.MIN_PASSWORD_LENGTH = 6
```

MONITOR_OUTPUT_DIRECTORY 选项

控制 IQ 缓冲区高速缓存监视器的输出文件的位置。

允许值
字符串。

默认值
与数据库所在目录相同。

范围
可以为 PUBLIC 组设置。设置将立即生效。设置此选项需要具有 DBA 权限。

描述
MONITOR_OUTPUT_DIRECTORY 控制在其中创建 IQ 监视器输出文件的目录，不管正在监视什么或使用什么监视模式。用于启动监视器的虚表可以是临时表，也可以是永久表。目录可以在任何物理计算机上。

所有监视器输出文件都在监视器运行时段内使用，该时段不能超过连接的生命周期。在监视器运行停止之后，输出文件仍然存在。连接可以同时运行最多两个性能监视器，一个用于 main 高速缓存，另一个用于临时高速缓存。连接可以连续地运行监视器任意次数。

DBA 可以使用 PUBLIC 设置将所有监视器输出放在相同目录中，也可以为单个用户设置不同的目录。

示例
此示例显示如何声明监视器输出的临时表，并设置其位置，然后让监视器开始将主和临时缓冲区高速缓存的文件发送到该位置。

注意： 在此示例中，输出目录字符串被设置为 “/tmp” 和 “tmp/”。后面的斜杠（“/”）是正确的，并受接口支持。该示例演示缓冲区高速缓存监视器不需要永久表；可以使用临时表。

```
declare local temporary table dummy_monitor (dummy_column integer)

set option Monitor_Output_Directory = "/tmp"
iq utilities main into dummy_monitor start monitor '-debug -interval
2'

set option Monitor_Output_Directory = "tmp/"

iq utilities private into dummy_monitor start monitor '-debug -
interval 2'
```

MPX_AUTOEXCLUDE_TIMEOUT 选项

指定自动排除协调节点上辅助节点的超时时间。

0 表示将不自动排除节点。此选项 不适用于指定的故障切换节点。用户必须 获得 Multiplex Grid Option 使用许可才能运行辅助节点。有关 MPX_AUTOEXCLUDE_TIMEOUT 语法和 完整说明，请参见《使用 Sybase IQ Multiplex》。

MPX_HEARTBEAT_FREQUENCY 选项

指定 heartbeat 线程唤醒和清除辅助节点上连接归集之前的间隔。

用户必须获得 Multiplex Grid Option 使用许可才能运行 辅助节点。有关 MPX_HEARTBEAT_FREQUENCY 语法 和完整说明，请参见《使用 Sybase IQ Multiplex》。

MPX_IDLE_CONNECTION_TIMEOUT 选项

指定辅助节点上连接归集中未使用的连接将在多长时间之后关闭。

用户必须获得 Multiplex Grid Option 使用许可才能运行 辅助节点。有关 MPX_IDLE_CONNECTION_TIMEOUT 语法 和完整说明，请参见《使用 Sybase IQ Multiplex》。

MPX_MAX_CONNECTION_POOL_SIZE 选项

指定辅助节点上连接归集中允许的最大连接数。

用户必须获得 Multiplex Grid Option 使用许可才能运行 辅助节点。有关 MPX_MAX_CONNECTION_POOL_SIZE 语法 和完整说明，请参见《使用 Sybase IQ Multiplex》。

MPX_MAX_UNUSED_POOL_SIZE 选项

指定辅助节点上连接归集中未使用的最大连接数。

用户必须获得 Multiplex Grid Option 使用许可才能运行 辅助节点。有关 MPX_MAX_UNUSED_POOL_SIZE 语法 和完整说明，请参见《使用 Sybase IQ Multiplex》。

MPX_WORK_UNIT_TIMEOUT 选项

Multiplex DQP 领导节点将分布式工作单元重新分配到另一个 DQP 工作节点之前的时间，以秒为单位。

用户必须获得 Multiplex Grid Option 使用许可才能运行 辅助节点。

有关 MPX_WORK_UNIT_TIMEOUT 语法和完整说明，请参见《使用 Sybase IQ Multiplex》。

NEAREST_CENTURY 选项 [TSQL]

控制在字符串到日期转换中对两位数表示的年份的解释。

允许值

0 - 100

默认值

50

描述

NEAREST_CENTURY 控制在从字符串转换到日期或时间戳时对两位数表示的年份的处理。

NEAREST_CENTURY 设置为一个用作切换点的数字值。小于该值的两位数表示的年份将转换成 20yy，而大于或等于该值的年份则转换成 19yy。

Adaptive Server Enterprise 和 Sybase IQ 的行为是使用最近的世纪，以便如果年份值 yy 小于 50，则将年份设置为 20yy。

NOEXEC 选项

生成优化程序查询计划，而不是执行计划。

允许值

ON、OFF

默认值

OFF

范围

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

描述

在确定如何处理查询时，IQ 优化程序将生成查询计划，以映射它如何计划让查询引擎处理查询。如果将此选项设置为 ON，则优化程序会将查询的计划发送到 IQ 消息文件，而不是将它提交给查询引擎。NOEXEC 影响查询和包括查询的命令。

将 NOEXEC 设置为 ON 还会阻止执行 **INSERT...VALUES**、**INSERT...SELECT**、**INSERT...LOCATION**、**SELECT...INTO**、**LOAD TABLE**、**UPDATE**、**TRUNCATE TABLE**、**DELETE**、**SYNCHRONIZE JOIN INDEX**，以及可更新的游标操作。

当 **EARLY_PREDICATE_EXECUTION** 选项设置为 ON 时，Sybase IQ 会在生成查询计划之前 为所有查询执行本地谓词，即使 NOEXEC 选项设置为 ON 也是如此。生成的查询计划与运行时计划相同。

另请参见

- `EARLY_PREDICATE_EXECUTION` 选项 (第 367 页)

NON_ANSI_NULL_VARCHAR 选项

控制是否出于插入/装载/更新操作而将零长度的 `VARCHAR` 数据视为 `NULL`。

允许值

`ON`、`OFF`

默认值

`OFF`

范围

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

描述

`NON_ANSI_NULL_VARCHAR` 允许您在装载或更新操作期间在处理零长度 `VARCHAR` 数据时恢复到非 ANSI (版本 12.03.1) 行为。如果将此选项设置为 `OFF`，则在装载、插入或更新期间，零长度 `VARCHAR` 作为零长度进行存储。如果将此选项设置为 `ON`，则零长度 `VARCHAR` 数据在装载、插入或更新时作为 `NULL` 进行存储。

NON_KEYWORDS 选项 [TSQL]

关闭单个关键字，允许将其用作标识符。

允许值

字符串

默认值

" (空字符串)

描述

`NON_KEYWORDS` 关闭单个关键字。如果在数据库中有一个标识符，而它现在是关键字，则可以在所有应用程序或脚本中在该标识符两侧添加双引号，也可以使用 `NON_KEYWORDS` 选项关闭该关键字。

以下语句阻止将 **`TRUNCATE`** 和 **`SYNCHRONIZE`** 识别为关键字：

```
SET OPTION NON_KEYWORDS = 'TRUNCATE, SYNCHRONIZE'
```

此选项的每个新设置都会替换以前的设置。此语句清除所有以前设置：

```
SET OPTION NON_KEYWORDS =
```

该选项的副作用是不能使用那些使用关闭关键字的 `SQL` 语句；它们会产生语法错误。

NOTIFY_MODULUS 选项

控制由某些命令发出的通知消息的缺省频率。

允许值
任意整数

默认值
100000

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

描述
此选项设置 Sybase IQ 为某些产生通知消息的命令所发出的缺省通知消息数。用于某些命令（如 **CREATE INDEX**、**LOAD TABLE** 和 **DELETE**）的 **NOTIFY** 子句将覆盖此值。不支持 **NOTIFY** 子句（如 **SYNCHRONIZE JOIN INDEX**）的其它命令始终使用此值。缺省值不限制您可以接收的消息数。

ODBC_DISTINGUISH_CHAR_AND_VARCHAR 选项

控制 Sybase IQ 和 SQL Anywhere ODBC 驱动程序如何描述 CHAR 列。

允许值
ON、OFF

默认值
OFF

描述
当打开连接时，Sybase IQ 和 SQL Anywhere ODBC 驱动程序将使用此选项的设置来确定如何描述 CHAR 列。如果 ODBC_DISTINGUISH_CHAR_AND_VARCHAR 设置为 OFF（缺省值），则 CHAR 列描述为 SQL_VARCHAR。如果此选项设置为 ON，则 CHAR 列描述为 SQL_CHAR。VARCHAR 列始终描述为 SQL_VARCHAR。

ON_CHARSET_CONVERSION_FAILURE 选项

控制在字符转换期间遇到错误时要采取的操作。

允许值
String。有关允许值，请参见“说明”。

默认值
IGNORE

描述

ON_CHARSET_CONVERSION_FAILURE 控制在字符转换期间遇到错误时要采取的操作：

字符转换错误	操作
IGNORE	不会出现错误和警告。
WARNING	将替代和非法字符报告为警告。不转换非法字符。
ERROR	将替代和非法字符报告为错误。

单字节到单字节的转换器无法报告替代和非法字符，且必须设置为 IGNORE。

ON_ERROR 选项 [Interactive SQL]

控制在 Interactive SQL 中执行语句期间遇到错误时要采取的操作。

允许值

字符串。有关允许值，请参见“说明”。

缺省值

PROMPT

描述

控制在执行语句期间遇到错误时要采取的操作：

- STOP - Interactive SQL 停止从文件执行语句，并返回到语句窗口等待输入。
- PROMPT - Interactive SQL 提示用户查看是否要继续。
- CONTINUE - 错误显示在“Messages”窗格，并且 Interactive SQL 继续执行语句。
- EXIT - Interactive SQL 终止。
- NOTIFY_CONTINUE - 报告错误，并提示用户按 **Enter** 键或单击“确定”继续。
- NOTIFY_STOP - 报告错误，并提示用户按 **Enter** 键或单击“确定”停止执行语句。
- NOTIFY_EXIT - 报告错误，并提示用户按 **Enter** 键或单击“确定”终止 Interactive SQL。

在执行 .SQL 文件时，STOP 和 EXIT 这两个值是等效的。

ON_TSQL_ERROR 选项 [TSQL]

控制存储过程中的错误处理。

允许值

字符串。有关允许值，请参见“说明”。

默认值

CONDITIONAL

描述

ON_TSQL_ERROR 控制存储过程中的错误处理：

- **STOP** – 发现错误时立即停止执行。
- **CONDITIONAL** – 如果过程使用 **ON EXCEPTION RESUME**，则错误后面的语句将处理该错误，并继续执行，否则退出。
- **CONTINUE** – 继续执行，不管下面是什么语句。如果有多个错误，则返回存储过程中所遇到的第一个错误。此选项最近似地镜像了 Adaptive Server Enterprise 行为。

使用 ON_TSQL_ERROR 的 CONDITIONAL 和 CONTINUE 设置 的目的都是为了与 Adaptive Server Enterprise 兼容，而 CONTINUE 可最近似地 模拟 Adaptive Server Enterprise 的行为。建议使用 CONDITIONAL 设置（尤其是在开发新的 Transact-SQL 存储过程时），因为 CONDITIONAL 允许早一些报告错误。

《参考：构件块、表和过程》中对 Adaptive Server Enterprise 兼容性进行了说明。

当此选项设置为 STOP 或 CONTINUE 时，它会取代 CONTINUE_AFTER_RAISERROR 选项的设置。然而，当此选项设置为 CONDITIONAL（缺省值）时，**RAISERROR** 语句后的行为由 CONTINUE_AFTER_RAISERROR 选项的设置决定。

另请参见

- CREATE PROCEDURE 语句（第 112 页）
- CREATE PROCEDURE 语句 [T-SQL]（第 118 页）
- RAISERROR 语句 [T-SQL]（第 256 页）
- CONTINUE_AFTER_RAISERROR 选项 [TSQL]（第 345 页）

OS_FILE_CACHE_BUFFERING 选项

控制文件系统缓冲针对 IQ 主数据库空间的使用。

允许值

ON、OFF

默认值

OFF；缺省值仅影响最近创建的数据库。

范围

只能为 PUBLIC 组设置。数据库必须关闭并重新启动，才能使更改生效。设置此选项需要具有 DBA 权限。

描述

如果将 `OS_FILE_CACHE_BUFFERING` 设置为 `OFF`，将阻止文件系统对 `IQ` 主存储文件进行缓冲。如果关闭文件系统缓冲，则将数据副本从文件系统缓冲区高速缓存保存到 `IQ` 的主缓冲区高速缓存。通常，这会减少由 `IQ` 缓冲区管理器和操作系统的文件系统缓冲区竞争内存而导致的分页。当 `OS_FILE_CACHE_BUFFERING` 减少分页时，此选项会改进性能；但是，如果数据库的 `IQ` 页大小小于文件系统的块大小（通常仅在测试情况中），那么，尤其是在多用户操作期间，性能会降低。

尝试使用此选项以确定不同情况的最佳设置。必须重新启动数据库才能使新设置生效。

此直接 I/O 性能选项仅在 Sun Solaris UFS、Linux、Linux IBM、AIX 和 Windows 文件系统上可用。此选项对于 HP-UX 和 HP-UXi 不起作用，并且不影响原始磁盘上的数据库。在 Linux，直接 I/O 在内核版本 2.6.x 中受支持。

要在 Linux 内核版本 2.6 和 AIX 上启用直接 I/O，还要将环境变量 `IQ_USE_DIRECTIO` 设置为 1。缺省情况下，直接 I/O 在 Linux 内核版本 2.6 和 AIX 上处于禁用状态。`IQ_USE_DIRECTIO` 对 Sun Solaris 和 Windows 不起作用。

注意：在 Linux 内核版本 2.4 上，Sybase IQ 不支持直接 I/O。如果在 Linux 内核版本 2.4 上设置了 `IQ_USE_DIRECTIO` 环境变量，则 Sybase IQ 服务器不启动。报告错误“错误：块 I/O 参数无效，可能 <pathname> 是目录，或者它超过平台的最大文件大小限制，或者您正尝试在不被支持的 OS 上使用直接 IO”。

`OS_FILE_CACHE_BUFFERING_TEMPDB` 控制 `IQ` 临时存储文件的文件系统缓冲。

另请参见

- `OS_FILE_CACHE_BUFFERING_TEMPDB` 选项（第 414 页）

OS_FILE_CACHE_BUFFERING_TEMPDB 选项

控制文件系统缓冲针对 `IQ` 临时数据库空间的使用。

允许值

ON、OFF

默认值

OFF

范围

只能为 `PUBLIC` 组设置。数据库必须关闭并重新启动，才能使更改生效。设置此选项需要具有 `DBA` 权限。

描述

如果将 `OS_FILE_CACHE_BUFFERING_TEMPDB` 设置为 `OFF`，将阻止 `IQ` 临时存储文件的文件系统缓冲。如果关闭文件系统缓冲，则将数据副本从文件系统缓冲区高速

缓存保存到 IQ 的主缓冲区高速缓存。通常，这会减少由 IQ 缓冲区管理器和操作系统的文件系统缓冲区竞争内存而导致的分页。当

OS_FILE_CACHE_BUFFERING_TEMPDB 减少分页时，此选项会改进性能；但是，如果数据库的 IQ 页大小小于文件系统的块大小（通常仅在测试情况中），那么，尤其是在多用户操作期间，性能会降低。

尝试使用此选项以确定不同情况的最佳设置。必须重新启动数据库才能使新设置生效。

此直接 I/O 性能选项仅在 Sun Solaris UFS、Linux、Linux IBM、AIX 和 Windows 文件系统上可用。此选项对于 HP-UX 和 HP-UXi 不起作用，并且不影响原始磁盘上的数据库。在 Linux，直接 I/O 在内核版本 2.6.x 中受支持。

要在 Linux 内核版本 2.6 和 AIX 上启用直接 I/O，还要将环境变量 IQ_USE_DIRECTIO 设置为 1。缺省情况下，直接 I/O 在 Linux 内核版本 2.6 和 AIX 上处于禁用状态。IQ_USE_DIRECTIO 对 Sun Solaris 和 Windows 不起作用。

注意： 在 Linux 内核版本 2.4 上，Sybase IQ 不支持直接 I/O。如果在 Linux 内核版本 2.4 上设置了 IQ_USE_DIRECTIO 环境变量，则 Sybase IQ 服务器不启动。报告错误“错误：块 I/O 参数无效，可能 <pathname> 是目录，或者它超过平台的最大文件大小限制，或者您正尝试在不被支持的 OS 上使用直接 IO”。

OS_FILE_CACHE_BUFFERING 控制 IQ 主存储文件的文件系统缓冲。

另请参见

- OS_FILE_CACHE_BUFFERING 选项（第 413 页）

PASSWORD_EXPIRY_ON_NEXT_LOGIN 选项

如果为用户分配登录策略并将该策略的选项设置为 ON，则下次登录时立即将用户的口令标记为到期。

有关详细信息，请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “CREATE LOGIN POLICY 语句”。

注意： 此参考指向 SQL Anywhere 文档。

PASSWORD_GRACE_TIME 选项

指定在口令到期前剩余的天数，在此期间允许登录，但缺省登录后过程将发出警告。

有关详细信息，请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “CREATE LOGIN POLICY 语句”。

注意： 此参考指向 SQL Anywhere 文档。

PASSWORD_LIFE_TIME 选项

指定必须更改口令前的最大天数。

有关详细信息，请参见“SQL Anywhere Server - SQL 参考” > “SQL 语句” > “CREATE LOGIN POLICY 语句”。

注意： 此参考指向 SQL Anywhere 文档。

POST_LOGIN_PROCEDURE 选项

指定其结果集包含在用户成功登录后客户端应用程序立即显示的消息的登录过程。

允许值
字符串

默认值
dbo.sa_post_login_procedure

范围
可针对单个连接或 PUBLIC 组进行设置。设置此选项需要具有 DBA 权限。设置将立即生效。

描述
用户成功登录后，将立即执行缺省登录后过程 **dbo.sa_post_login_procedure**。

如果您拥有 DBA 授权，则可以通过新建过程并设置 POST_LOGIN_PROCEDURE 以调用新过程来自定义登录后操作。不要编辑 **dbo.sa_post_login_procedure**。必须在使用的每个数据库中创建自定义登录后过程。

登录后过程支持客户端应用程序 Interactive SQL、Interactive SQL Classic 和用于 Sybase Central 的 IQ 插件。

另请参见

- LOGIN_PROCEDURE 选项（第 395 页）

PRECISION 选项

指定在任何十进制算法的结果中的最大数字位数，仅用于在目录存储上的查询。

允许值
126

默认值
126

范围

仅允许 PUBLIC 设置。

描述

精度是小数点左右两侧的数字总数。缺省 PRECISION 值固定在 126。{SCALE 选项指定当算法结果被截断到 PRECISION 所指定的最大长度时在小数点之后的最小位数，仅用于在目录存储上的查询。

注意：在 SQL Anywhere 中，数值函数支持的最大值是 255。如果数值函数的精度超过支持的最大值，则会出现以下错误：“函数 '_funcname' 的结果数据类型超过支持的最大数值精度 255。 请为数值函数 'location' 的精度 设置正确的值”

另请参见

- SCALE 选项（第 430 页）
- MAX_CLIENT_NUMERIC_PRECISION 选项（第 397 页）

PREFETCH 选项

允许打开或关闭提取，或使用 ALWAYS 值预取甚至 SENSITIVE 游标类型的游标结果，以及涉及代理表的游标的游标结果。

允许值

ON、OFF、ALWAYS

默认值

ON

范围

可针对单个连接或 PUBLIC 组进行设置。设置将立即生效。

描述

仅对于目录存储，PREFETCH 控制是否在行对客户端应用程序可用之前将其提取到客户端。即使当客户端应用程序一次请求一行时（例如，当在游标的行上循环时），也会一次读取多行，这样做会通过限制对数据库的请求数量而最大程度减少响应时间，并提高总吞吐量。

Open Client 和 JDBC 连接以及对于 IQ 存储会忽略 PREFETCH 的设置。

PREFETCH_BUFFER_LIMIT 选项

指定用于预取的内存量。

允许值

整数

默认值

0

范围

只能为 PUBLIC 组设置。设置此选项需要具有 DBA 授权。关闭并重新启动数据库服务器，以使更改生效。

描述

PREFETCH_BUFFER_LIMIT 定义 Sybase IQ 可用于执行预取（对数据库页的提前读取）的高速缓存页数。

除非有 Sybase 技术支持部门的指导，否则不要设置此选项。

另请参见

- PREFETCH_BUFFER_PERCENT 选项（第 418 页）

PREFETCH_BUFFER_PERCENT 选项

指定用于预取的内存百分比。

允许值

0 - 100

默认值

40

范围

只能为 PUBLIC 组设置。设置此选项需要具有 DBA 授权。关闭并重新启动数据库服务器，以使更改生效。

描述

PREFETCH_BUFFER_PERCENT 是 PREFETCH_BUFFER_LIMIT 的替代选项，因为它指定预取可以使用的高速缓存的百分比。

除非有 Sybase 技术支持部门的指导，否则不要设置此选项。

另请参见

- PREFETCH_BUFFER_LIMIT 选项（第 417 页）

PREFETCH_GARRAY_PERCENT 选项

指定为插入到 HG 索引而分配的预取资源的百分比。

允许值

0 - 100

默认值

60

范围

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

描述

与 PREFETCH_SORT_PERCENT 相似，此选项指定在插入到 **HG** 索引中时要使用的预取资源的百分比。

除非有 Sybase 技术支持部门的指导，否则不要设置此选项。

PREFETCH_SORT_PERCENT 选项

指定为排序对象而分配的预取资源的百分比。

允许值

0 - 100

默认值

20

范围

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

描述

PREFETCH_SORT_PERCENT 指定单个排序对象所使用的预取资源的百分比。增加此值可以改进插入和删除的单用户性能，但在多用户操作时可能有不良效果。

除非有 Sybase 技术支持部门的指导，否则不要设置此选项。

PRESERVE_SOURCE_FORMAT 选项 [database]

控制是否在系统文件中保存过程、视图和事件处理程序的原始源定义。如果保存，则它将保存在 SYSTABLE、SYSPROCEDURE 和 SYSEVENT 中的源列中。

允许值

ON、OFF

默认值

ON

范围

仅允许 PUBLIC 设置。

描述

当 PRESERVE_SOURCE_FORMAT 为 ON 时，数据库服务器会保存来自过程、视图和事件上的 **CREATE** 和 **ALTER** 语句的格式化源代码，并将原源定义置于相应系统表的 source 列。

未格式化的源文本存储在同一系统表中的 proc_defn 和 view_defn 列内。然而，这些定义在 Sybase Central 中并不容易读取。格式化的源列允许您使用自选的空格、注释和大小写来查看这些定义。

可以将此选项关闭以减少用于保存数据库中对象定义的空间。只能为 PUBLIC 组设置此选项。

QUERY_DETAIL 选项

指定是否包括查询计划中“查询详细信息”部分的其它查询信息。

允许值

ON、OFF

默认值

OFF

范围

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

描述

当 QUERY_DETAIL 和 QUERY_PLAN（或 QUERY_PLAN_AS_HTML）都打开时，Sybase IQ 将在产生其查询计划时显示有关查询的其它信息。当 QUERY_PLAN 和 QUERY_PLAN_AS_HTML 为 OFF 时，则忽略此选项。

当 QUERY_PLAN 是 ON（缺省值）时，尤其是如果 QUERY_DETAIL 也是 ON 时，可能需要启用消息日志回卷或消息日志存档，以避免填满消息日志文件。有关消息日志回卷，请参见《系统管理指南第一卷》。

另请参见

- QUERY_PLAN 选项（第 421 页）
- QUERY_PLAN_AS_HTML 选项（第 422 页）

QUERY_NAME 选项

对所执行的查询在其查询计划中命名。

允许值

最多 80 个字符的以引号分隔的字符串。

默认值

" (空字符串)

范围

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

描述

可以为 QUERY_NAME 选项指定任何以引号分隔的字符串值，最多 80 个字符。例如：

```
set temporary option Query_Name = 'my third query'
```

设置此选项时，在发送到 .iqmsg 文件或 .html 文件的查询计划中，在靠近计划开头处将包括一行类似下面的句子：

```
Query_Name: 'my third query'
```

如果在脚本中在每个查询之前将该选项设置为其它值，将更容易标识特定查询的正确查询计划。查询名称也将添加到 HTML 查询计划的文件名中。此选项对查询没有其它影响。

QUERY_PLAN 选项

指定是否将其它查询计划 输出到 Sybase IQ 消息文件。

允许值

ON、OFF

默认值

ON

范围

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

描述

打开此选项时，Sybase IQ 将在 IQ 消息文件中产生文本 查询计划。这些查询计划显示查询树结构以及有关优化和执行 的详细信息。关闭此选项时，不显示这些消息。该信息将发送到 <dbname>.iqmsg 文件。

另请参见

- `QUERY_DETAIL` 选项 (第 420 页)
- `QUERY_PLAN_AFTER_RUN` 选项 (第 422 页)
- `QUERY_PLAN_AS_HTML` 选项 (第 422 页)

QUERY_PLAN_AFTER_RUN 选项

在执行完查询之后，打印整个查询计划。

允许值

ON、OFF

默认值

OFF

范围

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

描述

打开 `QUERY_PLAN_AFTER_RUN` 时，在已经运行完查询之后，将打印查询计划。这将允许查询计划包括其它信息，例如从查询的每个节点传递过来的实际行数。

要 let 此选项工作，必须将 `QUERY_PLAN` 选项设置为 ON (缺省值)。可以将此选项与 `QUERY_DETAIL` 一起使用，以便在查询计划报告中生成其它信息。

另请参见

- `QUERY_DETAIL` 选项 (第 420 页)
- `QUERY_PLAN` 选项 (第 421 页)
- `QUERY_PLAN_AS_HTML` 选项 (第 422 页)

QUERY_PLAN_AS_HTML 选项

以 HTML 格式生成图形化的查询计划，以便在 Web 浏览器中查看。

允许值

ON、OFF

默认值

OFF

范围

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

描述

QUERY_PLAN_AS_HTML 导致以 HTML 格式生成图形化查询计划。

设置此选项时，还应当为每个查询设置 QUERY_NAME 选项，以便知道哪个查询与该查询计划关联。

Sybase IQ 会将计划写入与 .iqmsg 文件相同的目录内的文件中，该文件名为：

`user-name_query-name_YYYYMMDD_HHMMSS_query-number.html`

例如，如果用户 DBA 将临时选项 QUERY_NAME 设置为 'Query_1123'，则在 2011 年 5 月 18 日早上 8:30 分创建的文件将命名为

DBA_Query_1123_20110518_083000_1.html。日期、时间和唯一编号将自动追加到文件名的末尾，以确保不会覆盖现有文件。

注意： 如果使用此功能，请监视磁盘空间使用情况，以便能留下足够空间供 .iqmsg 和日志文件增长。启用 IQ 消息日志回卷或消息日志存档可避免填满消息日志文件。

有关消息日志回卷，请参见《系统管理指南第一卷》。

QUERY_PLAN_AS_HTML 独立操作 QUERY_PLAN 选项的设置。换句话说，如果 QUERY_PLAN_AS_HTML 是 ON，则无论 QUERY_PLAN 是否是 ON，您都会获得 HTML 格式的查询计划。

很多常用浏览器的较新版本都支持此功能。对于非常复杂的查询所生成的计划，某些浏览器可能会遇到问题。

另请参见

- QUERY_NAME 选项（第 421 页）
- QUERY_PLAN 选项（第 421 页）
- QUERY_PLAN_AFTER_RUN 选项（第 422 页）

QUERY_PLAN_AS_HTML_DIRECTORY 选项

指定 Sybase IQ 将在其中写入 HTML 查询计划的目录。

允许值

包含目录路径名称的字符串

默认值

"（空字符串）

范围

可针对单个连接或 PUBLIC 组进行临时设置。设置此选项需要具有 DBA 授权。设置将立即生效。

描述

打开 `QUERY_PLAN_AS_HTML` 选项并用 `QUERY_PLAN_AS_HTML_DIRECTORY` 选项指定了一个目录时, Sybase IQ 会将 HTML 查询计划写入该指定目录中。此选项通过允许 HTML 查询计划在服务器目录以外生成提高了安全性。如果未使用 `QUERY_PLAN_AS_HTML_DIRECTORY` 选项, 则将查询计划发送到缺省目录 (`.iqmsg` 文件目录) 中。

如果 `QUERY_PLAN_AS_HTML` 选项为 ON 并将 `QUERY_PLAN_AS_HTML_DIRECTORY` 设置为不存在的目录, 则 Sybase IQ 不会保存 HTML 查询计划, 并且不会生成错误。在这种情况下, 将继续运行查询, 并向 IQ 消息文件中记录一条消息, 以便 DBA 知道 HTML 查询计划尚未写入。如果指定的目录路径 或对该目录的权限不正确, 则会将消息 “打开 HTML 查询计划时出错: `file-name` 被写入 `.iqmsg` 文件”。

示例

创建示例目录 `/system1/users/DBA/html_plans`, 并设置对该目录的正确权限。然后, 设置选项并运行查询:

```
SET TEMPORARY OPTION QUERY_PLAN_AS_HTML = 'ON' ;
SET TEMPORARY OPTION QUERY_PLAN_AS_HTML_DIRECTORY = '/system1/
users/DBA/html_plans' ;
SELECT coll FROM tabl;
```

HTML 查询计划将写入指定目录 `/system1/users/DBA/html_plans` 中的文件。

另请参见

- `QUERY_PLAN_AS_HTML` 选项 (第 422 页)

QUERY_PLAN_TEXT_ACCESS 选项

允许或阻止用户从 Interactive SQL 客户端访问查询计划或使用 SQL 函数获取计划。

允许值

ON、OFF

默认值

OFF

范围

修改此选项需要 DBA 权限。可针对单个连接或 PUBLIC 组进行临时设置。设置将立即生效。

描述

当 `QUERY_PLAN_TEXT_ACCESS` 选项为 ON 时, 用户可以从 Interactive SQL 客户端查看、保存和打印查询计划。当该选项为 OFF 时, 则不会对查询计划进行高速缓存, 并且与查询计划相关的其它数据库选项对 Interactive SQL 客户端的查询计划显示不起任何作用。将会显示以下错误消息:

“No plan available.The database option QUERY_PLAN_TEXT_ACCESS is OFF.” (计划不可用。数据库选项 QUERY_PLAN_TEXT_ACCESS 为 OFF。)

另请参见

- QUERY_DETAIL 选项 (第 420 页)
- QUERY_PLAN_AFTER_RUN 选项 (第 422 页)
- QUERY_PLAN_AS_HTML 选项 (第 422 页)
- QUERY_PLAN_TEXT_CACHING 选项 (第 425 页)
- OUTPUT 语句 [Interactive SQL] (第 247 页)

QUERY_PLAN_TEXT_CACHING 选项

允许用户指定 Sybase IQ 是否针对用户所执行的查询生成 IQ 计划 并对其进行高速缓存。

允许值

ON、OFF

默认值

OFF

范围

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

描述

IQ 查询计划的大小会发生变化，对于复杂查询可能会变得非常大。用于 Interactive SQL 客户端显示的高速缓存计划可能有较高的资源要求。QUERY_PLAN_TEXT_CACHING 选项为用户提供了控制对计划进行高速缓存的资源的机制。如果关闭该选项（缺省值），查询计划将不会对该用户连接进行高速缓存。

注意： 如果关闭 QUERY_PLAN_TEXT_ACCESS，则对于来自该用户的连接，无论 QUERY_PLAN_TEXT_CACHING 如何设置都不会对查询计划进行高速缓存。

另请参见

- QUERY_DETAIL 选项 (第 420 页)
- QUERY_PLAN_AFTER_RUN 选项 (第 422 页)
- QUERY_PLAN_AS_HTML 选项 (第 422 页)
- QUERY_PLAN_TEXT_ACCESS 选项 (第 424 页)
- OUTPUT 语句 [Interactive SQL] (第 247 页)

QUERY_ROWS_RETURNED_LIMIT 选项

根据结果集的估计大小设置拒绝查询的行阈值。

允许值
任意整数

默认值
0

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

描述
如果 Sybase IQ 所接收查询的结果行的 估计数目大于
QUERY_ROWS_RETURNED_LIMIT 的值， 会拒绝该查询并显示如下消息：

```
Query rejected because it exceeds resource:  
Query_Rows_Returned_Limit (由于超过资源， 查询被拒绝：  
Query_Rows_Returned_Limit)
```

如果将此选项设置为零（缺省值），则不存在限制，因而不会根据其输出中的行数拒绝任何查询。

QUERY_TEMP_SPACE_LIMIT 选项

指定拒绝查询前临时空间的最大估计量。

允许值
任意整数

默认值
0（无限制）

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

描述
如果 Sybase IQ 所接收的查询估计要求临时结果空间 大于此选项的值，会拒绝该查询并显示如下消息：

```
Query rejected because it exceeds total space resource limit (由于超过  
总空间资源限制， 查询被拒绝)
```

将此选项设置为零（缺省值）时，对查询使用的临时存储没有限制。

用户可能在自己的环境中覆盖此选项来运行可能填满整个临时存储的查询。要防止失控的查询填满临时存储，DBA 可以设置选项

MAX_TEMP_SPACE_PER_CONNECTION。MAX_TEMP_SPACE_PER_CONNECTION 选项将监控和限制所有 DML 语句（而不仅仅是查询）的实际临时存储使用情况。

在分布式查询处理事务中，Sybase IQ 将通过限制参与分布式查询的所有节点使用的共享空间和本地临时空间总量对于共享临时存储使用为 QUERY_TEMP_SPACE_LIMIT 和 MAX_TEMP_SPACE_PER_CONNECTION 选项设置的值。这意味着无论有多少节点参与，任何单一查询都不能超过临时空间总量限制（来自 IQ_SYSTEM_TEMP 和 IQ_SHARED_TEMP dbspace）。

例如，如果限制为 100，四个节点中每个使用 25 个单位量的临时空间，则查询位于限制内。不过，如果任一节点使用的空间总量超过 100，则查询将回退。

另请参见

- MAX_TEMP_SPACE_PER_CONNECTION 选项（第 404 页）

QUERY_TIMING 选项

确定是否收集特定的计时统计信息并在查询计划中显示出来。

允许值

ON、OFF

默认值

OFF

范围

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

描述

此选项控制对查询引擎中有关子查询及其它某些重复函数的计时统计信息的收集。此参数通常应为 OFF（缺省值），因为对于很短的相关子查询，对每个子查询执行进行计时可能会减慢查询速度。

查询计时在查询计划详细信息中表示为一系列时间戳。这些时间戳对应于查询运算符阶段（条件、准备、提取、完成）。HTML 和 Interactive SQL 查询计划以图形方式将查询计时显示为时间轴。

QUOTED_IDENTIFIER 选项 [TSQL]

控制对用双引号括起的字符串的解释。

允许值

ON、OFF

OFF（对于 Open Client 连接）。

默认值

ON

描述

QUOTED_IDENTIFIER 控制用双引号括起来的字符串是被解释为标识符 (ON) 还是普通字符串 (OFF)。包括此选项是为了与 Transact-SQL 兼容。

如果 QUOTED_IDENTIFIER 设置为 OFF，Sybase Central 和 Interactive SQL 会暂时将其设置为 ON。此时系统会显示一条消息，以通知您这一更改。此更改仅对 Sybase Central 或 Interactive SQL 连接有效。JDBC 驱动程序也会将 QUOTED_IDENTIFIER 临时设置为 ON。

RECOVERY_TIME 选项

设置数据库服务器从系统失败中恢复可花费的最长时间（以分钟为单位）。

允许值

整数（以分钟为单位）

默认值

2

范围

只能为 PUBLIC 组设置。设置将在重新启动服务器后生效。

描述

此选项与 CHECKPOINT_TIME 选项一起使用时，可决定应在何时执行检查点操作。

试探法根据自上次检查点操作之后执行的操作来计算恢复时间。因此，恢复时间并不精确。

另请参见

- CHECKPOINT_TIME 选项（第 344 页）

RESERVED_KEYWORDS 选项

启用缺省情况下禁用的各个关键字。

允许值

字符串

缺省值

空字符串

描述

该选项只能为 **PUBLIC** 组设置。不允许用户和临时设置。

该选项将启用缺省情况下禁用的各个关键字。只能启用 **LIMIT** 关键字。

示例

以下语句允许将 **LIMIT** 关键字识别为关键字：

```
SET OPTION RESERVED_KEYWORDS = 'LIMIT';
```

不能启用关键字 **SET**、**OPTION** 和 **OPTIONS**。通过以下方式可确定某个字是否会确认为关键字（按优先顺序排列）：

- 出现在 SQL Anywhere 保留字列表中
- 通过 **RESERVED_KEYWORDS** 选项启用
- 通过 **NON_KEYWORDS** 选项禁用

此选项的每个设置都会替换以前的设置。以下语句会清除所有以前的设置：

```
SET OPTION RESERVED_KEYWORDS = ;
```

RETURN_DATE_TIME_AS_STRING 选项

控制在查询时如何将日期、时间或时间戳值传递给客户端应用程序。

允许值

ON、**OFF**

默认值

OFF

范围

在当前连接的整个期间只能设置为临时选项。

描述

RETURN_DATE_TIME_AS_STRING 指示日期、时间和时间戳值是作为日期或时间数据类型还是作为字符串返回应用程序。

当此选项设置为 **ON** 时，服务器会在将日期、时间或时间戳值发送到客户端之前将其转换为字符串，以便保留 **TIMESTAMP_FORMAT**、**DATE_FORMAT** 或 **TIME_FORMAT** 选项设置。

Sybase Central 和 Interactive SQL 会自动将 **RETURN_DATE_TIME_AS_STRING** 选项设置为 **ON**。

另请参见

- **DATE_FORMAT** 选项（第 355 页）
- **TIME_FORMAT** 选项（第 453 页）

- **TIMESTAMP_FORMAT** 选项（第 454 页）

ROW_COUNT 选项

限制从查询返回的行数。

允许值
整型。

默认值
0（不限制返回的行数）

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

描述
当此运行期选项设置为非零值时，查询处理会在返回指定行数后停止。

此选项仅影响带有关键字 **SELECT** 的语句，且不会影响 **UPDATE** 和 **DELETE** 语句。

SELECT 语句的关键字 **FIRST** 和 **TOP** 也限制从查询返回的行数。使用 **FIRST** 等同于将 **ROW_COUNT** 数据库选项设置为 1。使用 **TOP** 等同于将 **ROW_COUNT** 选项设置为相同的行数。如果 **TOP** 和 **ROW_COUNT** 都进行了设置，则 **TOP** 的值优先。

在涉及全局变量、系统函数或代理表的查询中使用 **ROW_COUNT** 选项可能会产生不确定性的结果。此类查询部分使用 **CIS**（组件集成服务）执行。在这种情况下，应使用 **SELECT TOP *n*** 代替设置 **ROW_COUNT**，或将全局变量设置为局部变量并在查询中使用该局部变量。

另请参见

- **QUERY_ROWS_RETURNED_LIMIT** 选项（第 426 页）
- **SELECT** 语句（第 277 页）

SCALE 选项

指定按照最大 **PRECISION** 截断算术结果时小数点后的最少位数，仅适用于对目录存储的查询。

允许值
最大值为 126 的整数。

默认值
38

范围
只能为 **PUBLIC** 设置。

描述

此选项指定按照最大 `PRECISION` 截断算术结果时小数点后的最少位数，适用于对目录存储的查询。

乘、除、加、减和集合函数的结果都可以超过最大精度。

另请参见

- `MAX_CLIENT_NUMERIC_SCALE` 选项（第 397 页）
- `PRECISION` 选项（第 416 页）

SIGNIFICANTDIGITSFORDOUBLEEQUALITY 选项

指定指数记数法中小数点右侧的有效位数，这些位数用于两个复杂算术表达式之间的相等测试。

允许值

0 - 15

默认值

0

范围

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

描述

由于双精度型值以二进制（以 2 为基数）而不是十进制（以 10 为基数）形式存储，因此此设置提供的所用有效小数位数是近似的。如果此选项设置为 0，则使用所有位数。

例如，当 `SIGNIFICANTDIGITSFORDOUBLEEQUALITY` 设置为 12 时，以下两个数字的比较结果相等；当设置为 13 时，其比较结果就不等：

- 1.23456789012345
- 1.23456789012389

`SIGNIFICANTDIGITSFORDOUBLEEQUALITY` 影响两个复杂算术表达式之间的相等测试，而对按指数计算的表达式没有影响。

SORT_COLLATION 选项

允许在 `SORTKEY` 表达式上隐式使用 `ORDER BY` 函数。

允许值

`Internal`、`collation_name` 或 `collation_id`

默认值

Internal

范围

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

描述

当 SORT_COLLATION 的值为 Internal，**ORDER BY** 子句保持不变。

如果将此选项的值设置为有效的归类名称或归类 ID，则 **ORDER BY** 子句中的任何字符串表达式作为 **SORTKEY** 函数已调用处理。

有关函数，请参见《参考：构件块、表和过程》。

示例

将排序归类设置为二进制：

```
SET TEMPORARY OPTION sort_collation='binary';
```

将排序归类设置为二进制将转换以下查询：

```
SELECT Name, ID
FROM Products
ORDER BY Name, ID;
SELECT Name, ID
FROM Products
ORDER BY 1, 2;
```

查询将被转换为：

```
SELECT Name, ID
FROM Products
ORDER BY SORTKEY(Name, 'binary'), ID;
```

SORT_PINNABLE_CACHE_PERCENT 选项

指定排序对象尝试固定的缓冲区可占当前可用缓冲区的最大百分比。

允许值

0 - 100

默认值

20

范围

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

描述

对于非常大的排序，选择较大的值可能有助于减少排序所需的合并阶段数。但较大的值可能会影响在系统上运行的其他用户的排序和散列。如果更改此选项，请尝试找到最佳值以提高性能，因为选择错误的值可能会导致性能降低。Sybase 建议您使用 SORT_PINNABLE_CACHE_PERCENT 的缺省值。

此选项主要供 Sybase 技术支持部门使用。如要更改 SORT_PINNABLE_CACHE_PERCENT 的值，则请谨慎操作。

SQL_FLAGGER_ERROR_LEVEL 选项 [TSQL]

控制对未包含在指定标准中的任何 SQL 代码进行响应的行为。

允许值

- OFF
- SQL:1992/Entry
- SQL:1992/Intermediate
- SQL:1992/Full
- SQL:1999/Core
- SQL:1999/Package
- SQL:2003/Core
- SQL:2003/Package

默认值

OFF

描述

将所有不是指定标准组成部分的 SQL 代码都标记为错误。例如，指定 SQL:2003/Package 会导致数据库服务器标记不是完整级 SQL/2003 语法的语法。为了与以前的 Sybase IQ 版本兼容，也接受该表中的值，而且按照指定进行映射。

表 29. SQL_FLAGGER_ERROR_LEVEL 兼容性值

值	操作
E	标记不是入门级 SQL92 语法的语法。对应于 SQL:1992/Entry。
I	标记不是中间级 SQL92 语法的语法。对应于 SQL:1992/Intermediate。
F	标记不是完整级 SQL92 语法的语法。对应于 SQL:1992/Full。
W	允许所有支持的语法。对应于 OFF。

SQL_FLAGGER_WARNING_LEVEL 选项 [TSQL]

控制对任何未包含在指定标准中的 SQL 进行的响应。

允许值

- OFF
- SQL:1992/Entry
- SQL:1992/Intermediate
- SQL:1992/Full
- SQL:1999/Core
- SQL:1999/Package
- SQL:2003/Core
- SQL:2003/Package

默认值

OFF

描述

将所有不是指定标准组成部分的 SQL 代码都标记为警告。例如，指定 SQL:2003/Package 会导致数据库服务器标记不是完整级 SQL/2003 语法的语法。

缺省行为 OFF 会关闭警告标记。

为了与以前的 Sybase IQ 版本兼容，也接受该表中的值，而且按照指定进行映射。

表 30. SQL_FLAGGER_WARNING_LEVEL 兼容性值

值	操作
E	标记不是入门级 SQL92 语法的语法。对应于 SQL:1992/Entry。
I	标记不是中间级 SQL92 语法的语法。对应于 SQL:1992/Intermediate。
F	标记不是完整级 SQL92 语法的语法。对应于 SQL:1992/Full。
W	允许所有支持的语法。对应于 OFF。

STRING_RTRUNCATION 选项 [TSQL]

确定当 INSERT 或 UPDATE 截断 CHAR 或 VARCHAR 字符串时是否会出现错误。

允许值

ON、OFF

默认值

ON

描述

如果截断的字符均为空格，则不会引发例外。如果设置为 ON，则与 SQL92 行为相对应。如果 STRING_RTRUNCATION 为 OFF，则不会引发异常，并以无提示方式截断字符串。如果该选项为 ON，则会引发错误，并执行 ROLLBACK。

在 Sybase IQ 15.0 之前，缺省情况下该选项为 OFF。可以将其安全地设置为 OFF 以便向后兼容。但是，ON 设置比标识在其中进行截断可能导致数据丢失的语句更可取。

SUBQUERY_CACHING_PREFERENCE 选项

控制用于处理相关子查询谓词的算法。

允许值

-3 到 3

值	操作
1	将基于排序的处理用于第一个子查询谓词。其它没有相同排序键的子查询谓词将使用散列表处理以高速缓存子查询结果。
2	使用散列表高速缓存合法时所有子查询谓词的结果。如果可用的临时高速缓存不能容纳所有子查询结果，性能可能会下降。
3	高速缓存一个以前的子查询结果。切勿使用 SORT 和 HASH。
0	让优化程序选择。
-1	不要使用 SORT。如果合法，IQ 优化程序将选择 HASH。
-2	不要使用 HASH。如果合法，IQ 优化程序将选择 SORT 或高速缓存的值。
-3	不要使用高速缓存的值。如果合法，IQ 优化程序将选择 HASH 或 SORT。

默认值

0

范围

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

描述

对于相关子查询谓词，IQ 优化程序提供了高速缓存外部引用和降低子查询执行开销的子查询结果的选项。SUBQUERY_CACHING_PREFERENCE 允许您在选择要使用的算法时覆盖优化程序的开销决策。它不会覆盖用于确定算法在查询引擎内是否合法的内部规则。

非零值的设置将影响查询中的每个子查询谓词。无法选择性地将非零值用于查询中的某一个子查询谓词。

请参见《参考：构件块、表和过程》>“SQL 语言元素”>“搜索条件”>“搜索条件中的子查询”>“分离子查询谓词”。

SUBQUERY_CACHING_PREFERENCE 通常用于内部测试，只有经验丰富的 DBA 才能使用。它不适用于 IN 子查询。

另请参见

- IN_SUBQUERY_PREFERENCE 选项（第 383 页）

SUBQUERY_FLATTENING_PERCENT 选项

允许用户更改优化程序决定将标量子查询转换为连接处的阈值。

允许值

0 到 (2³² -1)

值	操作
0	让优化程序成本模型决定
1 到 (2 ³² -1)	平展的引用百分比

默认值

100

范围

此选项仅适用于相关标量子查询。设置 SUBQUERY_FLATTENING_PERCENT 无需 DBA 权限。此选项可由任何用户在任何级别进行设置，并会立即生效。如果将 SUBQUERY_FLATTENING_PERCENT 设置为非缺省值，查询中的每个标量子查询谓词都将受影响；无法选择性地将此选项用于查询中的某一个标量子查询谓词。

描述

Sybase IQ 查询优化程序可以将相关标量 子查询转换为等价的连接操作以提高查询性能。 SUBQUERY_FLATTENING_PERCENT 选项 选项允许用户调整 执行优化处的阈值。

SCALAR_FLATTENING_PERCENT 表示标量子查询中估计内部非重复值与估计外部非重复值的百分比。因为估计的百分比接近 100%，所以评估子查询作为连接的开销可能小于使用单独的索引探查。可以将该值设置为大于 100%，因为不保证估计内部重复值小于估计外部重复值。

另请参见

- SUBQUERY_FLATTENING_PREFERENCE 选项（第 437 页）

SUBQUERY_FLATTENING_PREFERENCE 选项

允许用户覆盖优化程序将标量或 **EXISTS** 子查询转换（平展）为连接时的决定。

允许值

-3 到 3

值	操作
-3	避免将 EXISTS 和标量子查询平展为连接操作。
-2	避免将标量子查询平展为连接操作。
-1	避免将 EXISTS 子查询平展为连接操作。
0	允许 IQ 优化程序决定平展子查询。
1	忽略平展 EXIST 的开销（如有可能）。
2	忽略平展标量的开销（如有可能）。
3	忽略 EXISTS 和标量子查询的开销。

默认值

0

范围

设置此选项无需 **DBA** 权限。**SUBQUERY_FLATTENING_PREFERENCE** 可由任何用户在任何级别进行设置。此选项会立即生效。如果将此选项设置为非零值，查询中的每个子查询谓词都将受影响；无法选择性地将此选项用于查询中的某一个子查询谓词。

描述

Sybase IQ 优化程序可以将相关标量子查询或 **EXISTS** 或 **NOT EXISTS** 子查询 转换为等同的连接操作以提高查询性能。该优化称为 *子查询平展*。
SUBQUERY_FLATTENING_PREFERENCE 允许您在选择要使用的算法时 覆盖优化程序的开销决策。

将 **SUBQUERY_FLATTENING_PREFERENCE** 设置为 0（允许 IQ 优化程序决定平展子查询）等同于将 Sybase IQ 早期版本中目前已不受支持的 **FLATTEN_SUBQUERIES** 选项设置为 **ON**。

另请参见

- **SUBQUERY_FLATTENING_PERCENT** 选项 （第 436 页）

SUBQUERY_PLACEMENT_PREFERENCE 选项

控制查询计划中相关子查询谓词运算符的位置。

允许值

-1 到 1

值	操作
-1	选择查询计划中尽可能靠前的位置，从而尽可能先执行查询中的子查询。
0	让优化程序选择。
1	选择查询计划中尽可能靠后的位置，从而尽可能地延迟执行查询中的子查询。

默认值

0

范围

可针对任何范围、任何用户进行设置，设置将立即生效。

描述

对于查询中的相关子查询运算符，IQ 优化程序可以在相应查询计划中选择多个不同的有效位置。SUBQUERY_PLACEMENT_PREFERENCE 允许在选择放置位置时覆盖优化程序的基于开销的决策。它不能覆盖用于确定位置是否有效的内部规则，而在有些查询中，可能只有一个位置可供选择。如果将此选项设置为非零值，则会影响查询中的每个相关子查询谓词；此选项不能用于从查询的多个子查询中有选择性地修改一个子查询的位置。

此选项通常用于内部测试，只有经验丰富的 DBA 才能使用。

此选项的缺省设置几乎总是适用的。有时，Sybase 技术支持部门可能会要求您更改该值。

SUPPRESS_TDS_DEBUGGING 选项

确定是否在服务器窗口中显示 TDS 调试信息。

允许值

ON、OFF

默认值

OFF

描述

在以 **-z** 选项启动服务器时，会在服务器窗口中显示调试信息，其中包括有关 TDS 协议的调试信息。

SUPPRESS_TDS_DEBUGGING 会限制出现在数据库服务器窗口中的有关 TDS 的调试信息。当此选项设置为 OFF（缺省值）时，会在服务器窗口中显示 TDS 调试信息。

SWEEPER_THREADS_PERCENT 选项

指定用于清理 缓冲区高速缓存的 Sybase IQ 线程所占百分比。

允许值

1 - 40

默认值

10

范围

只能为 PUBLIC 组设置。设置此选项需要具有 DBA 授权。必须关闭并重新启动数据库服务器，才能使更改生效。

描述

Sybase IQ 使用一小部分处理线程 作为清理程序线程。这些清理程序线程用于清除主和临时缓冲区高速缓存中的脏页。

在 IQ 监视器 **-cache** 报告中，GDirty 列显示在“脏”（已修改）状态下发生的 LRU 缓冲区争夺次数。如果 GDirty 已有一段时间值大于 0，可能需要增加 SWEEPER_THREADS_PERCENT 或 WASH_AREA_BUFFERS_PERCENT。

此选项的缺省设置几乎总是适用的。有时，Sybase 技术支持部门可能会要求您增加该值。

另请参见

- WASH_AREA_BUFFERS_PERCENT 选项（第 458 页）

TDS_EMPTY_STRING_IS_NULL 选项 [database]

控制为 TDS 连接返回的空字符串为 NULL 还是包含一个空白字符的字符串。

允许值

ON, OFF

默认值

OFF

描述

TDS_EMPTY_STRING_IS_NULL 缺省情况下，被设置为 OFF，并且 TDS 连接会将空字符串作为包含一个空白字符的字符串返回。当此选项设置为 ON 时，为 TDS 连接返回的空字符串为 NULL 字符串。非 TDS 连接会对空字符串和 NULL 字符串加以区分。

TEMP_EXTRACT_APPEND 选项

指定将数据提取功能所提取的所有行添加到输出文件的末尾。

允许值

ON、OFF

默认值

OFF

范围

可针对单个连接进行设置。设置将立即生效。

描述

此选项指定将数据提取功能所提取的所有行 添加到输出文件的末尾。可在 您对其具有 **WRITE/EXECUTE** 权限的目录中创建输出文件，并为用于启动 **Sybase IQ** 的用户名（例如，**sybase**） 设置对该目录和输出文件的 **WRITE** 权限。您可以根据需要为其他用户授予对输出文件的权限。在 **TEMP_EXTRACT_NAME1** 选项中指定输出文件的名称。如果尚不存在输出文件，则数据提取功能会创建该文件。

TEMP_EXTRACT_APPEND 与 **TEMP_EXTRACT_SIZE**_n 选项不兼容。如果您尝试限制提取追加输出文件的大小，**Sybase IQ** 将报告 错误。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

另请参见

- **TEMP_EXTRACT_NAME**_n 选项（第 443 页）

TEMP_EXTRACT_BINARY 选项

与 **TEMP_EXTRACT_SWAP** 选项合并使用，用于指定数据提取功能所执行的提取的类型。

允许值

ON、OFF

默认值

OFF

范围

可针对单个连接进行设置。设置将立即生效。

描述
此选项与 TEMP_EXTRACT_SWAP 选项一起使用，用于指定数据提取功能所执行的提取的类型。

表 31. 与提取类型对应的提取选项设置

提取类型	TEMP_EXTRACT_BINARY	TEMP_EXTRACT_SWAP
binary	ON	OFF
二进制/交换	ON	ON
ASCII	OFF	OFF

缺省提取类型为 ASCII。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》> “导入和导出数据” > “从数据库导出数据的方法” > “数据提取功能”。

另请参见

- TEMP_EXTRACT_SWAP 选项（第 450 页）

TEMP_EXTRACT_COLUMN_DELIMITER 选项

指定用于 ASCII 提取的数据提取功能的输出中各列之间的分隔符。

允许值
字符串

默认值
,

范围
可针对单个连接进行设置。设置将立即生效。

描述
使用 TEMP_EXTRACT_COLUMN_DELIMITER 可指定数据提取功能的输出中各列之间的分隔符。就 ASCII 提取而言，缺省用逗号分隔各列值。缺省情况下，字符串不带引号。

该分隔符必须占 1 至 4 字节；如果使用的是多字节归类顺序，该分隔符必须在当前使用的归类顺序中有效。应选择一个任何数据输出字符串自身中不存在的分隔符。

对于 ASCII 提取，如果将此选项设置为空字符串 "，则将用不带列分隔符的固定宽度 ASCII 写入所提取的数据。数值数据类型和二进制数据类型在具有 *n* 个空白的字段上右对齐，其中 *n* 为该类型的任何值所需的最大字节数。字符数据类型在具有 *n* 个空白的字段上左对齐。

注意：固定宽度 ASCII 提取中的最小列宽为 4 字节，以允许字符串“NULL”表示空值。例如，如果所提取的列为 CHAR(2) 并且 TEMP_EXTRACT_COLUMN_DELIMITER 设置为空字符串"，则提取的数据后面会有两个空格。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

另请参见

- TEMP_EXTRACT_QUOTE 选项（第 447 页）
- TEMP_EXTRACT_QUOTES 选项（第 447 页）
- TEMP_EXTRACT_QUOTES_ALL 选项（第 448 页）
- TEMP_EXTRACT_ROW_DELIMITER 选项（第 449 页）

TEMP_EXTRACT_DIRECTORY 选项

控制是否允许用户使用数据提取功能。还控制放置临时提取文件的目录并覆盖在 TEMP_EXTRACT_NAME_n 选项中指定的目录路径。

允许值

string

默认值

" (空字符串)

范围

可针对单个连接或 PUBLIC 组进行临时设置。设置此选项需要具有 DBA 授权。此选项会立即生效。

描述

如果针对某一用户将 TEMP_EXTRACT_DIRECTORY 选项设置为字符串 FORBIDDEN (不区分大小写)，则禁止该用户执行数据提取。该用户尝试使用数据提取功能将生成错误: You do not have permission to perform Extracts (您无权执行提取)。

如果 TEMP_EXTRACT_DIRECTORY 组的 PUBLIC 设置为 FORBIDDEN，则任何人均无法运行数据提取。

如果将 TEMP_EXTRACT_DIRECTORY 设置为有效目录路径，则将临时提取文件放置到该目录中，并覆盖在 TEMP_EXTRACT_NAME_n 选项中指定的路径。

如果将 TEMP_EXTRACT_DIRECTORY 设置为无效目录路径，则会发生错误: Files does not exist File: <invalid path> (文件不存在文件: <无效路径>)

如果 TEMP_EXTRACT_DIRECTORY 为空，则根据在 TEMP_EXTRACT_NAME_n 中的相应指定，将临时提取文件放置到目录中。如果 TEMP_EXTRACT_NAME_n 中未指定任何路径，缺省情况下，提取文件将放置到服务器启动目录中。

此选项可限制在用户对其具有写入访问权限的目录中创建大型数据提取文件的操作，从而增强了安全性并有助于控制磁盘管理。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

另请参见

- TEMP_EXTRACT_NAME_n 选项（第 443 页）

TEMP_EXTRACT_ESCAPE_QUOTES 选项

指定在用于 ASCII 提取的数据提取工具的输出中，是否包含引号的字段中的所有引号都进行转义。

允许值

ON、OFF

默认值

OFF

范围

可针对单个连接进行设置。设置将立即生效。

描述

除非以下条件成立，否则将忽略此选项：**TEMP_EXTRACT_QUOTE** 为缺省值或设置为 ""（双引号）值，**TEMP_EXTRACT_BINARY** 为 OFF，且 **TEMP_EXTRACT_QUOTES** 或 **TEMP_EXTRACT_QUOTES_ALL** 为 ON。

另请参见

- TEMP_EXTRACT_BINARY 选项（第 440 页）
- TEMP_EXTRACT_QUOTES 选项（第 447 页）
- TEMP_EXTRACT_QUOTES_ALL 选项（第 448 页）

TEMP_EXTRACT_NAME_n 选项

指定数据提取功能所使用的输出文件或命名管道的名称。有 8 个选项：**TEMP_EXTRACT_NAME₁** 通过 **TEMP_EXTRACT_NAME₈**。

允许值

string

默认值

"（空字符串）

范围

可针对单个连接进行设置。设置将立即生效。

描述

TEMP_EXTRACT_NAME1 到 TEMP_EXTRACT_NAME8 指定数据提取功能所使用的输出文件的名称。必须按顺序使用这些选项。例如，除非事先已设置 TEMP_EXTRACT_NAME3 和 TEMP_EXTRACT_NAME1 选项，否则 TEMP_EXTRACT_NAME2 将无效。

在这些选项中，最重要的选项就是 TEMP_EXTRACT_NAME1。如果 TEMP_EXTRACT_NAME1 设置为其缺省设置（空字符串），则将禁用提取并且不重定向输出。要启用提取，请将 TEMP_EXTRACT_NAME1 设置为路径名称。提取功能即开始将相应数据提取至具有该名称的文件中。为该文件选择一个未在别处使用的路径名称。Sybase 建议将 TEMP_EXTRACT_NAME1 选项设置为 TEMPORARY。

当 TEMP_EXTRACT_APPEND 选项设置为 ON 时，You can also use TEMP_EXTRACT_NAME1 指定输出文件的名称。在这种情况下，请在执行 **SELECT** 语句之前，为用于启动 Sybase IQ 的用户名（例如，**sybase**）设置对包含该已命名文件的目录或文件夹以及该已命名文件的 **WRITE** 权限。在附加模式中，数据提取功能向文件末尾添加提取的行并且不会覆盖文件中的已有数据。如果尚不存在输出文件，则数据提取功能会创建该文件。

警告！ 如果选择一个现有文件的路径名称，并将 TEMP_EXTRACT_APPEND 选项设置为 OFF（缺省值），则该文件的内容将被覆盖。例如，如果该文件是周报告文件，而不是某一数据库文件，则这可能正是您所需要的。

除了 TEMP_EXTRACT_NAME2 之外，还可以使用 TEMP_EXTRACT_NAME8 到 TEMP_EXTRACT_NAME1 选项指定多个输出文件的名称。

如果要提取到单个磁盘文件或单个命名管道，请将选项 TEMP_EXTRACT_NAME2 到 TEMP_EXTRACT_NAME8 以及 TEMP_EXTRACT_SIZE1 到 TEMP_EXTRACT_SIZE8 保留为缺省值。

如果设置了 TEMP_EXTRACT_NAME1，则不能执行下列这些操作：针对连接中顶层表的

- **LOAD、DELETE、INSERT, 或 INSERT...LOCATION** 操作
- **SYNCHRONIZE JOIN INDEX**（显式发出或作为 **CREATE JOIN INDEX** 的一部分执行）
- **INSERT...SELECT**

此外，还应注意对数据提取功能的下列限制：

- 仅能对 IQ 存储中存储的数据进行提取操作。
- 对系统表或跨数据库连接无法进行提取。
- 对于使用用户定义的函数或除 **suser_id()** 和 **suser_name()** 以外的系统函数的查询，不能使用提取功能。

- 如果运行带有 **-q**（安静模式）选项的 **Interactive SQL**，并且数据提取命令位于命令文件中，则必须首先永久设置 **Interactive SQL** 选项“显示多个结果集”（**Show multiple result sets**）。如果此选项未设置，则不会创建输出文件。
要设置“显示多个结果集”选项，请在 **Interactive SQL** 窗口中依次选择“工具”>“选项”，然后选中“显示多个结果集”（**Show multiple result sets**）框并单击“设为永久”（**Make permanent**）。

可用 **TEMP_EXTRACT_NAME_n** 选项覆盖使用 **TEMP_EXTRACT_DIRECTORY** 选项指定的目录路径。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

另请参见

- **TEMP_EXTRACT_APPEND** 选项（第 440 页）
- **TEMP_EXTRACT_DIRECTORY** 选项（第 442 页）
- **TEMP_EXTRACT_SIZE_n** 选项（第 449 页）

TEMP_EXTRACT_NULL_AS_EMPTY 选项

控制在用于 **ASCII** 提取的数据提取功能的输出中，空值的表示形式。

允许值

ON、OFF

默认值

OFF

范围

可针对单个连接进行设置。设置将立即生效。

描述

TEMP_EXTRACT_NULL_AS_EMPTY 控制在用于 **ASCII** 提取的数据提取功能的输出中，空值的表示形式。当 **TEMP_EXTRACT_NULL_AS_EMPTY** 选项设置为 **ON** 时，对于所有数据类型，空值的表示形式均为 "（空字符串）。

提取输出文件中不包含上述引号。当 **TEMP_EXTRACT_NULL_AS_EMPTY** 选项设置为 **OFF** 时，则在所有情况下均使用字符串 'NULL' 表示空值。**OFF** 为缺省值。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

TEMP_EXTRACT_NULL_AS_ZERO 选项

控制在用于 ASCII 提取的数据提取功能的输出中，空值的表示形式。

允许值

ON、OFF

默认值

OFF

范围

可针对单个连接进行设置。设置将立即生效。

描述

TEMP_EXTRACT_NULL_AS_ZERO 控制在用于 ASCII 提取的数据提取功能的输出中，空值的表示形式。当 TEMP_EXTRACT_NULL_AS_ZERO 设置为 ON，空值的表示形式如下：

- '0'，表示算术类型
- "（空字符串），表示 CHAR 和 VARCHAR 字符类型
- "（空字符串），表示日期
- "（空字符串），表示时间
- "(空字符串)，表示时间戳

提取输出文件中不包含上述引号。当 TEMP_EXTRACT_NULL_AS_ZERO 选项设置为 OFF 时，则在所有情况下均使用字符串 'NULL' 表示空值。OFF 为缺省值。

注意：在 Sybase IQ 12.5 中，从表的 CHAR 或 VARCHAR 列执行的 ASCII 提取始终至少向输出文件返回四个字符串。如果 TEMP_EXTRACT_NULL_AS_ZERO 设置为 OFF，则需要这样做，因为 Sybase IQ 需要向包含空值的列中的所有行写出词语 NULL。如果 TEMP_EXTRACT_NULL_AS_ZERO 设置为 ON，则无需保留四个字符的空间。

在 Sybase IQ 12.6 中，如果 TEMP_EXTRACT_NULL_AS_ZERO 设置为 ON，则 ASCII 提取向文件的 CHAR 或 VARCHAR 列所写入的字符数等于该列中的字符数，即使该数字小于 4 也是如此。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

TEMP_EXTRACT_QUOTE 选项

在 TEMP_EXTRACT_QUOTES 选项或 TEMP_EXTRACT_QUOTES_ALL 选项设置为 ON 时指定要用作引号的字符串，该引号括起用于 ASCII 提取的数据提取功能的输出中的字段。

允许值
字符串

默认值
" (空字符串)

范围
可针对单个连接进行设置。设置将立即生效。

描述
此选项在缺省值不适用时指定要用作引号的字符串，引号括起用于 ASCII 提取的数据提取功能的输出中的字段。TEMP_EXTRACT_QUOTE 与 TEMP_EXTRACT_QUOTES 和 TEMP_EXTRACT_QUOTES_ALL 选项一起使用。在 TEMP_EXTRACT_QUOTE 选项中指定的引号字符串与行和列分隔符具有相同的限制。此选项的缺省值是空字符串，Sybase IQ 会将其转换为单引号。

TEMP_EXTRACT_QUOTE 选项中所指定的字符串必须占 1 至 4（最大值）字节；如果使用的是多字节归类顺序，则该字符串必须在当前使用的归类顺序中有效。请务必选择一个任何数据输出字符串自身中不存在的字符串。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

另请参见

- TEMP_EXTRACT_COLUMN_DELIMITER 选项（第 441 页）
- TEMP_EXTRACT_QUOTES 选项（第 447 页）
- TEMP_EXTRACT_QUOTES_ALL 选项（第 448 页）
- TEMP_EXTRACT_ROW_DELIMITER 选项（第 449 页）

TEMP_EXTRACT_QUOTES 选项

指定在用于 ASCII 提取的数据提取功能的输出中，字符串字段用引号括起。

允许值
ON、OFF

默认值
OFF

范围

可针对单个连接进行设置。设置将立即生效。

描述

此选项指定在用于 ASCII 提取的数据提取功能的输出中，字符串字段用引号括起。如果缺省值不适用，则在 TEMP_EXTRACT_QUOTE 选项中指定用作引号的字符串。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

另请参见

- TEMP_EXTRACT_COLUMN_DELIMITER 选项（第 441 页）
- TEMP_EXTRACT_QUOTES_ALL 选项（第 448 页）
- TEMP_EXTRACT_ROW_DELIMITER 选项（第 449 页）

TEMP_EXTRACT_QUOTES_ALL 选项

指定在用于 ASCII 提取的数据提取功能的输出中，所有字段都用引号括起。

允许值

ON、OFF

默认值

OFF

范围

可针对单个连接进行设置。设置将立即生效。

描述

TEMP_EXTRACT_QUOTES_ALL 指定在用于 ASCII 提取的数据提取功能的输出中，所有字段都用引号括起。如果缺省值不适用，则在 TEMP_EXTRACT_QUOTE 中指定用作引号的字符串。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

另请参见

- TEMP_EXTRACT_COLUMN_DELIMITER 选项（第 441 页）
- TEMP_EXTRACT_QUOTES 选项（第 447 页）
- TEMP_EXTRACT_QUOTES_ALL 选项（第 448 页）
- TEMP_EXTRACT_ROW_DELIMITER 选项（第 449 页）

TEMP_EXTRACT_ROW_DELIMITER 选项

指定用于 ASCII 提取的数据提取功能的输出中，各行之间的分隔符。

允许值
字符串

默认值
" (空字符串)

范围
可针对单个连接进行设置。设置将立即生效。

描述

TEMP_EXTRACT_ROW_DELIMITER 指定数据提取功能的输出中各行之间的分隔符。就 ASCII 提取而言，缺省情况下以换行符作为行结尾（对于 UNIX 平台）或以回车/换行符对作为行结尾（对于 Windows 平台）。

该分隔符必须占 1 至 4 字节；如果使用的是多字节归类顺序，该分隔符必须在当前使用的归类顺序中有效。应选择一个任何数据输出字符串中不存在的分隔符。

TEMP_EXTRACT_ROW_DELIMITER 选项的缺省值是空字符串。Sybase IQ 会将此选项的缺省空字符串转换为换行符（对于 UNIX 平台）或回车/换行符对（对于 Windows 平台）。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

另请参见

- TEMP_EXTRACT_COLUMN_DELIMITER 选项（第 441 页）
- TEMP_EXTRACT_QUOTES 选项（第 447 页）
- TEMP_EXTRACT_QUOTES_ALL 选项（第 448 页）

TEMP_EXTRACT_SIZE_n 选项

指定数据提取功能所使用的对应输出文件的最大大小。

有 8 个选项：TEMP_EXTRACT_SIZE1 到 TEMP_EXTRACT_SIZE8。

默认值
0

范围
可针对单个连接进行设置。设置将立即生效。

描述

TEMP_EXTRACT_SIZE1 到 TEMP_EXTRACT_SIZE8 用于指定数据提取功能所使用的对应输出文件的最大大小。TEMP_EXTRACT_SIZE1 指定 TEMP_EXTRACT_NAME1 所指定的输出文件的最大大小，TEMP_EXTRACT_SIZE2 指定 TEMP_EXTRACT_NAME2 所指定的输出文件的最大大小，依此类推。

数据提取大小选项的缺省值为 0。Sybase IQ 可将此缺省值转换为以下各值：

设备类型	Size
磁盘文件	AIX 和 HP-UX: 0 - 64GB Sun Solaris 和 Linux: 0 - 512GB Windows: 0 - 128GB
磁带*	524288 KB (0.5 GB)
其它	9007199254740992KB (8192 PB “无限制”)

*目前不支持磁带设备。

如果大型文件系统（如 JFS2）所支持的文件大小大于缺省值，请将 TEMP_EXTRACT_SIZE n 设置为该文件系统允许的值。例如，要支持 1 TB 设置选项：

```
TEMP_EXTRACT_SIZE1 = 1073741824 KB
```

如果要提取到单个磁盘文件或单个命名管道，请将选项 TEMP_EXTRACT_NAME2 到 TEMP_EXTRACT_NAME8 以及 TEMP_EXTRACT_SIZE1 到 TEMP_EXTRACT_SIZE8 保留为缺省值。

TEMP_EXTRACT_SIZE n 与 TEMP_EXTRACT_APPEND 选项 不兼容。如果您尝试限制提取追加输出文件的大小，Sybase IQ 将报告 错误。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

另请参见

- TEMP_EXTRACT_NAME n 选项（第 443 页）

TEMP_EXTRACT_SWAP 选项

与 TEMP_EXTRACT_BINARY 选项合并使用，用于指定数据提取功能所执行的提取的类型。

允许值

ON、OFF

默认值
OFF

范围
可针对单个连接进行设置。设置将立即生效。

描述
此选项与 TEMP_EXTRACT_BINARY 选项一起使用，用于指定数据提取功能所执行的提取的类型。

表 32. 与提取类型对应的提取选项设置

提取类型	TEMP_EXTRACT_BINARY	TEMP_EXTRACT_SWAP
binary	ON	OFF
二进制/交换	ON	ON
ASCII	OFF	OFF

缺省提取类型为 ASCII。

有关数据提取功能及如何使用提取选项的详细信息，请参见《系统管理指南第一卷》>“导入和导出数据”>“从数据库导出数据的方法”>“数据提取功能”。

另请参见

- TEMP_EXTRACT_BINARY 选项（第 440 页）

TEMP_RESERVED_DBSPACE_MB 选项

控制 Sybase IQ 在临时 IQ 存储 中保留的空间量。

允许值
大于或等于 200 的整数（以兆字节为单位）

默认值
200; Sybase IQ 实际保留 IQ_SYSTEM_TEMP 中最后一个读写文件的 最多 50% 和最少 1%

范围
只能为 PUBLIC 组设置。设置此选项需要具有 DBA 授权。设置将立即生效。不需要重新启动服务器即可更改保留空间大小。

描述
TEMP_RESERVED_DBSPACE_MB 用于 控制 Sybase IQ 在临时 IQ 存储中， 为释放保存点、提交和检查点操作期间使用的某些小型 但极为重要的数据结构所留出的空间量。

对于生产数据库，请将该值设置为介于 200 MB 和 1 GB 之间的值。IQ 页大小和并发连接数越大，需要的保留空间就越多。

保留的空间大小是通过计算 IQ_SYSTEM_TEMP 中最后一个读写文件的最多 50% 和最少 1% 得出的。

TEMP_SPACE_LIMIT_CHECK 选项

对每个连接检查目录存储临时空间。

允许值

ON、OFF（不执行限制检查）

默认值

ON

范围

只能为 PUBLIC 组设置。需要具有 DBA 权限。

描述

当 TEMP_SPACE_LIMIT_CHECK 为 ON 时，数据库服务器会检查连接所使用的目录存储临时文件空间量。当此选项设置为 OFF 时，如果连接请求的临时文件空间量超出限额，则会发生致命错误。当此选项设置为 ON 时，如果连接请求的临时文件空间量超出限额，请求将失败并返回“超出了临时空间限制”错误。

有两个因素可用来确定连接的临时文件限额：临时文件的最大大小和活动数据库连接的数量。临时文件的最大大小是该文件当前大小与该文件所在分区上可用磁盘空间量之和。打开限制检查功能后，如果临时文件已达到其最大大小的 80% 或更多，且连接请求更多的临时文件空间，则服务器会检查连接是否超出了其限额。一旦发生这种情况，如果任何连接使用的临时文件空间量超出最大临时文件空间量和活动连接数的比值，该连接将会失败。

注意：此选项与 IQ 临时存储空间无关。要对 IQ 临时空间的增大进行约束，请使用 QUERY_TEMP_SPACE_LIMIT 选项和 MAX_TEMP_SPACE_PER_CONNECTION 选项。

您可以使用 **sa_disk_free_space system** 过程来获取有关临时文件的可用空间信息。有关详细信息，请参见“SQL Anywhere Server - SQL 参考”>“系统对象”>“系统过程”>“按字母顺序排列的系统过程列表”>“sa_disk_free_space 系统过程”。

注意：此参考指向 SQL Anywhere 文档。

示例

数据库启动时，其临时文件所在的驱动器上尚有 100MB 的可用空间，且该驱动器上没有任何其它活动文件。因此，该临时文件的可用空间为 100MB。DBA 输入：

```
SET OPTION PUBLIC.TEMP_SPACE_LIMIT_CHECK = 'ON'
```

只要临时文件保持不超过 80MB，服务器的行为就会和以前一样。一旦该文件达到 80MB，就可能会产生新的行为。如果运行 10 个查询，则临时文件将需要增大。当服务器发现有一个查询在使用超过 8MB 的临时文件空间时，该查询将失败。

TEXT_DELETE_METHOD 选项

指定在 **TEXT** 索引中进行删除时所使用的算法。

用户必须获得非结构化数据分析选项 的使用许可才能使用 **TEXT** 索引。有关 **TEXT_DELETE_METHOD** 语法 和完整说明，请参见《Sybase IQ 中的非结构化数据分析》。

TIME_FORMAT 选项

设置从数据库中检索到的时间所使用的格式。

允许值

由符号 **HH**、**NN**、**MM**、**SS** 组成的一个字符串，这些符号之间用冒号分隔。

默认值

'HH:NN:SS.SSS'

对于 Open Client 和 JDBC 连接，缺省值也设置为 **HH:NN:SS.SSS**。

描述

该格式是使用以下符号的字符串：

- **hh** – 两位数表示的小时数（24 小时制）。
- **nn** – 两位数表示的分钟数。
- **mm** – 如果跟在冒号后面（如 “**hh:mm**” 这样的格式），则为两位数表示的分钟数。
- **ss[s...s]** – 两位数表示的秒数加可选的小数部分。

每个符号都会替换为与要设置格式的时间对应的数据。任何表示字符而不是数字输出的格式符号都可以采用大写形式，这样会使替换的字符也显示为大写。对于数字，在格式字符串中使用混合的大小写会取消前导零的显示。

不能在格式字符串中使用多字节字符。即使数据库的归类顺序是多字节归类顺序（如 932JPN），也只能使用单字节字符。

另请参见

- **DATE_FORMAT** 选项 （第 355 页）
- **RETURN_DATE_TIME_AS_STRING** 选项 （第 429 页）

TIMESTAMP_FORMAT 选项

设置从数据库中检索到的时间戳所使用的格式。

允许值

由下面所列符号组成的字符串。

默认值

'YYYY-MM-DD HH:NN:SS.SSS'

描述

该格式是使用以下符号的字符串：

表 33. TIMESTAMP_FORMAT 字符串符号

符号	描述
yy	2 位数表示的年份。
yyyy	4 位数表示的年份。
mm	2 位数表示的月份，如果跟在冒号后面（如 'hh:mm' 这样的格式）则为两位数表示的分钟数。
mmm	月份名称的 3 字符简写形式
mmmm[m...]	月份名称的字符长格式：有多少个 m，就有多少个字符，直至指定的 m 个数超过相应月份名称中的字符数为止。
dd	两位数表示的日期。
ddd	星期天数的 3 字符简写形式
dddd[d...]	星期名称的字符长格式：有多少个 d，就有多少个字符，直至指定的 d 个数超过相应星期名称中的字符数为止。
hh	2 位数表示的小时。
nn	2 位数表示的分钟。
ss.SSS	秒 (ss) 和秒的小数部分 (SSS)，最多可有六个小数位。并非所有的平台都支持精确到小数点后第六位的时间戳。
aa	上午或下午（12 小时制）。
pp	根据需要表示下午（12 小时制）。

每个符号都会替换为与要设置格式的日期对应的数据。任何表示字符而不是数字输出的格式符号都可以采用大写形式，这样会使替换的字符也显示为大写。对于数字，在格式字符串中使用混合的大小写会取消前导零的显示。

不能在格式字符串中使用多字节字符。即使数据库的归类顺序是多字节归类顺序（如 932JPN），也只能使用单字节字符。

另请参见

- DATE_FORMAT 选项（第 355 页）
- RETURN_DATE_TIME_AS_STRING 选项（第 429 页）

TOP_NSORT_CUTOFF_PAGES 选项

设置结果大小阈值，以选择 **TOP N** 算法。

允许值

1 - 1000

默认值

1

描述

TOP_NSORT_CUTOFF_PAGES 选项设置阈值（以页为单位），在此种情况下，对包含 **TOP** 子句和 **ORDER BY** 子句的查询所执行的评估将算法从基于有序列表的处理切换到基于排序的处理。当 **TOP N** 值小于结果行数时，有序列表处理的执行效果更佳。对于较大的 **TOP N** 值，基于排序的处理的执行效果更佳。

在某些情况下，增加 TOP_NSORT_CUTOFF_PAGES 可避免基于排序的处理，进而提高性能。

另请参见

- SELECT 语句（第 277 页）

TRIM_PARTIAL_MBC 选项

允许自动剪裁部分多字节字符数据。

允许值

ON、OFF

默认值

OFF

范围

设置此选项无需 DBA 权限。只能为 PUBLIC 组设置。设置将立即生效。

描述

为同时包含单字节和多字节字符的归类提供一致的数据装载。如果 TRIM_PARTIAL_MBC 为 ON:

- 在装载到 CHAR 列时，将部分多字节字符替换为空格。
- 在装载到 VARCHAR 列时，截断部分多字节字符。

如果 TRIM_PARTIAL_MBC 为 OFF，则常规 CONVERSION_ERROR 语义有效。

另请参见

- CONVERSION_ERROR 选项 [TSQL] (第 346 页)

TSQL_VARIABLES 选项 [TSQL]

控制 @ 符号是否可以用作嵌入式 SQL 宿主变量名的前缀。

允许值

ON、OFF

ON (对于 Open Client 和 JDBC 连接)

默认值

OFF

描述

当 TSQL_VARIABLES 设置为 ON 时，可以使用 @ 符号代替冒号作为嵌入式 SQL 中宿主变量名的前缀。此选项主要用于 Open Server Gateway。

USER_RESOURCE_RESERVATION 选项

调整用于当前用户数的内存使用量。

允许值

整数

范围

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

默认值

1

描述

Sybase IQ 可跟踪打开的游标数，并相应地分配内存。在某些情况下，可以使用此选项调整 Sybase IQ 认为当前在使用产品的当前游标的最小数量，从而节省从临时高速缓存中分配的内存。

只有经过仔细分析后确定确实有此需要时，才应设置此选项。如果需要设置此参数，请与 Sybase 技术支持部门联系以了解详细信息。

VERIFY_PASSWORD_FUNCTION 选项

指定可用于执行口令规则的用户提供的鉴定函数。

允许值
字符串

范围
可针对单个连接或 PUBLIC 组进行临时设置。设置此选项需要具有 DBA 授权。此选项会立即生效。

默认值
" (空字符串)。(不对 GRANT CONNECT 调用函数。)

描述
当 VERIFY_PASSWORD_FUNCTION 选项值设置为有效字符串时，语句 **GRANT CONNECT TO***userid***IDENTIFIED BY***password* 将调用该选项值所指定的函数。

该选项值需要使用 *owner.function_name* 格式，以防止用户覆盖该函数。

该函数具有两个参数：

- *user_name* VARCHAR(128)
- *new_pwd* VARCHAR(255)

返回的值类型为 VARCHAR(255)。

注意： 请对该函数执行 **ALTER FUNCTION***function-name***SET HIDDEN**，以确保用户不会使用过程调试工具单步执行该函数。

如果设置了 VERIFY_PASSWORD_FUNCTION，则无法使用 **GRANT CONNECT** 语句指定多个用户 ID 和口令。

示例
下面的语句创建一个要求口令不同于用户名的函数：

```
CREATE FUNCTION DBA.f_verify_pwd
( user_name varchar(128),
  new_pwd varchar(255) )
RETURNS varchar(255)
BEGIN
  -- enforce password rules
  IF new_pwd = user_name then
    RETURN('Password cannot be the same as the user name' );
  END IF;
  -- return success
  RETURN( NULL );
END;
ALTER FUNCTION DBA.f_verify_pwd set hidden;
```

```
GRANT EXECUTE on DBA.f_verify_pwd to PUBLIC;  
SET OPTION PUBLIC.VERIFY_PASSWORD_FUNCTION = 'DBA.f_verify_pwd';
```

要关闭该选项，请将其设置为空字符串：

```
SET OPTION PUBLIC.VERIFY_PASSWORD_FUNCTION = ''
```

另请参见

- ALTER FUNCTION 语句（第 14 页）
- GRANT 语句（第 202 页）

WASH_AREA_BUFFERS_PERCENT 选项

指定清洗标记上方的缓冲区高速缓存所占百分比。

允许值

1 - 100

默认值

20

范围

只能为 PUBLIC 组设置。设置此选项需要具有 DBA 授权。关闭并重新启动数据库服务器，以使更改生效。

描述

Sybase IQ 缓冲区高速缓存组织为一个长 MRU/LRU 链。在清洗标记上方的区域用于将脏页清理（即写入）到磁盘。

在 IQ 监视器 **-cache** 报告中，GDirty 列显示在“脏”（已修改）状态下发生的 LRU 缓冲区争夺次数。如果 GDirty 已有一段时间值大于 0，可能需要增加 SWEEPER_THREADS_PERCENT 或 WASH_AREA_BUFFERS_PERCENT。

此选项的缺省设置几乎总是适用的。有时，Sybase 技术支持部门可能会要求您增加该值。

另请参见

- SWEEPER_THREADS_PERCENT 选项（第 439 页）

WAIT_FOR_COMMIT 选项

确定在处理数据时检查外键完整性的时间。

允许值

ON、OFF

默认值

OFF

范围

可针对单个连接或 `PUBLIC` 组进行设置。设置将立即生效。

描述

如果此选项设置为 `ON`，则在下一条 `COMMIT` 语句之前，数据库并不检查外键完整性。否则，在插入、更新或删除所有不是使用 `CHECK ON COMMIT` 选项创建的外键时，都会对其进行检查。

WD_DELETE_METHOD 选项

指定在 `WD` 索引中进行删除时所使用的算法。

允许值

0 - 3

- 0: 开销模型选择删除方法。开销模型仅选择中型删除方法或大型删除方法。
- 1: 强制小型删除方法。当要删除的行数在表的总行数中所占的百分比非常小时，小型删除方法非常有用。少量删除算法可以随机访问索引，导致带有大型数据库的高速缓存抖动。
- 2: 强制大型删除方法。该算法将扫描整个索引搜索要删除的行。当要删除的行数在表的总行数中所占的百分比非常大时，大型删除方法非常有用。
- 3: 强制中型删除方法。中型删除方法是按顺序访问索引的小型删除方法的变化形式，通常快于小型删除方法。

默认值

0

范围

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

描述

`WD_DELETE_METHOD` 指定在 `WD` 索引中进行删除时所使用的算法。当该选项未设置或设置为 0 时，由开销模型选择删除方法。开销模型在选择相应的删除算法时会考虑 CPU 相关开销以及 I/O 相关开销。开销模型考虑下列各项：

- 删除行
- 索引大小
- 索引数据类型的宽度
- 索引数据的基数
- 可用临时高速缓存
- 与计算机相关的 I/O 和 CPU 特性
- 可用 CPU 和线程

示例

要在 **WD** 索引中强制大型删除方法，请执行以下操作：

```
SET TEMPORARY OPTION WD_DELETE_METHOD = 2
```

索引

A

AES 加密算法
 CREATE DATABASE 语句 74
 AGGREGATION_PREFERENCE 选项 332
 ALL
 SELECT 语句中的关键字 279
 ALLOCATE DESCRIPTOR 语句
 语法 5
 ALLOW_NULLS_BY_DEFAULT 选项 333
 ALLOW_READ_CLIENT_FILE 选项 337
 ALTER DATABASE UPGRADE 语句 7
 ALTER DATABASE 语句
 语法 7
 ALTER DBSPACE 语句
 语法 8
 ALTER DOMAIN 语句
 语法 12
 ALTER EVENT 语句
 语法 13
 ALTER FUNCTION 语句
 语法 14
 ALTER INDEX 语句
 错误 17
 语法 16
 ALTER LOGICAL SERVER 语句
 语法 17
 ALTER LOGIN POLICY 语句
 语法 19
 ALTER LS POLICY 语句
 语法 22
 ALTER MULTIPLEX RENAME statement 23
 ALTER MULTIPLEX RENAME 语句 23
 ALTER MULTIPLEX SERVER 语句 24
 ALTER PROCEDURE 语句
 语法 25
 ALTER SERVER 语句
 语法 27
 ALTER SERVICE 语句
 语法 29
 ALTER TABLE 语句
 语法 31
 ALTER USER 语句 39
 ALTER VIEW 语句
 RECOMPILE 35

 语法 41, 43
 ANSI_CLOSE_CURSORS_AT_ROLLBACK 选项 333
 ANSI_PERMISSIONS 选项 334
 ANSI_SUBSTRING 选项 335
 ANSI_UPDATE_CONSTRAINTS 选项 335
 ANSINULL 选项 334
 APPEND_LOAD 选项 337
 ASE_BINARY_DISPLAY
 数据库选项 338
 ASE_FUNCTION_BEHAVIOR
 使用 HEXTOINT 338
 使用 INTTOHEX 338
 数据库选项 338
 AT 子句
 CREATE EXISTING TABLE 88
 AUDITING 选项 339
 AUTOINCREMENT 列缺省值 141
 安全
 审计 339
 最小口令长度 406

B

B-tree 页 341
 BACKUP 语句
 存档设备的数量 47
 语法 44
 BEGIN DECLARE SECTION 语句
 语法 153
 BEGIN PARALLEL IQ 语句 52
 BEGIN TRANSACTION 语句
 Transact-SQL 53
 BEGIN……END 语句
 语法 49
 BLOB 变量
 数据类型转换 368
 BLOCKING 选项 340, 341
 BREAK 语句
 Transact-SQL 312
 BT_PREFETCH_MAX_MISS 选项 341
 BTREE_PAGE_SPLIT_PAD_PERCENT 选项 342
 BYE 语句
 语法 187

索引

绑定变量

- DESCRIBE 语句 166
- EXECUTE 语句 183
- OPEN 语句 246

保存点

- RELEASE SAVEPOINT 语句 259
- ROLLBACK TO SAVEPOINT 语句 273
- ROLLBACK TRANSACTION 语句 274
- SAVE TRANSACTION 语句 276
- 名称 3

备份

- 速度 47
- 验证 266

变量

- 121, 129
- BLOB 转换 368
- LONG BINARY 转换 368
- select into 281
- SET VARIABLE 语句 285
- 创建 149
- 二进制大对象转换 368
- 删除 181
- 声明 154

标签

- 用于语句 201
- 语句 3

标题名

标准

- SQL 1992 遵从性 433, 434
- SQL 1999 遵从性 433, 434
- SQL 2003 遵从性 433, 434

表

- 199
- GLOBAL TEMPORARY 136
- 创建 136
- 创建代理 87
- 更改 31
- 更改定义 35
- 将数据导出到文件 247
- 截断 302
- 临时 117, 146, 161
- 删除 170
- 锁定 239
- 重命名 37
- 装载 223

表约束

别名

- 列 280
- 在 DELETE 语句中 163
- 在 SELECT 语句中 279, 280

并发

- 锁定表 239

并行度

- 备份设备 47

- 不建议使用的数据库选项 320

C

- CACHE_PARTITIONS 选项 343

CALL 语句

- Transact-SQL 184
- 语法 55

CASE 语句

- 语法 57

- Catalog 存储 199

- CHAINED 选项 344

- CHECK ON COMMIT 子句

- 参照完整性 144

CHECK 条件

- 关于 142, 144

CHECKPOINT 语句

- 语法 58

- CHECKPOINT_TIME 选项 344

CIS

- 远程数据访问 345

- CIS_ROWSET_SIZE 选项

- 关于 345

CLEAR 语句

- 语法 59

CLOB 变量

- 数据类型转换 368

CLOSE 语句

- 语法 59

- CLOSE_ON_ENDTRANS 选项 345

- COMMENT ON LOGICAL SERVER 语句 62

- COMMENT ON LOGIN POLICY 语句

- 语法 61

COMMENT 语句

- 语法 61

- COMMIT TRANSACTION 语句

- Transact-SQL 63

COMMIT 语句

- 语法 63

CONFIGURE 语句

- 语法 64

CONNECT 权限

- GRANT 语句 205

- CONNECT 语句

- 语法 65

- connection_property 函数
 - 关于 316
- contains-expression 198
- CONTINUE 语句
 - Transact-SQL 312
- CONTINUE_AFTER_RAISE_ERROR 选项 345
- CONVERSION_ERROR 选项 346
- CONVERSION_MODE 选项 346
- CONVERT_VARCHAR_TO_1242 选项 352
- COOPERATIVE_COMMIT_TIMEOUT 选项 353
- COOPERATIVE_COMMITS 选项 353
- CREATE DATABASE 语句
 - 语法 68
- CREATE DBSPACE 语句
 - 语法 77
- CREATE DOMAIN 语句
 - 语法 80
- CREATE EVENT 语句
 - 语法 82
- CREATE EXISTING TABLE 语句
 - 代理表 87
- CREATE EXTERNLOGIN 语句
 - INSERT...LOCATION 213
 - 语法 89
- CREATE FUNCTION 语句
 - Java 96
 - UDF 96
 - 外部环境 96
 - 语法 90
- CREATE INDEX 语句 52
 - 表使用 101
 - 语法 97
- CREATE JOIN INDEX 语句
 - 语法 104
- CREATE LOGICAL SERVER 语句 107
- CREATE LOGIN POLICY 语句
 - 语法 108
- CREATE MESSAGE 语句
 - Transact-SQL 110
- CREATE MULTIPLEX SERVER 语句 111
- CREATE PROCEDURE 语句
 - Transact-SQL 118
 - 语法 112
- CREATE SCHEMA 语句
 - 语法 131
- CREATE SERVER 语句
 - INSERT...LOCATION 213
 - 语法 132
- CREATE SERVICE 语句
 - 语法 133
- CREATE TABLE 语句
 - 语法 136
- CREATE TEXT CONFIGURATION 语句 147
- CREATE TEXT INDEX 语句 147
- CREATE USER 语句 147
- CREATE VARIABLE 语句
 - 语法 149
- CREATE VIEW 语句
 - 语法 151
- CUBE 运算符 283
 - SELECT 语句 283
- CURSOR_WINDOW_ROWS 选项 354
- 插入
 - 使用游标插入行 255
- 查询 199
 - SELECT 语句 277
 - 可更新游标 158
 - 提高性能 341
 - 由 SQL Anywhere 规则处理 281
- 查询表 199, 281
- 查找页
 - 最大 371
- 处理查询而不 199, 281
- 创建
 - 存储过程 112
 - 代理表 87
 - 数据类型 80
 - 外部存储过程 120, 127, 128
- 创建为组 52
- 创建者 3
- 磁盘空间 83
- 从表中删除所有行 302
- 存储过程
 - sa_dependent_views 43
 - 创建 112
 - 代理 116
 - 选入结果集 279
- 存储空间
 - 最小化 405
- 存档设备
 - 并行备份的最大值 47
- 错误
 - RAISERROR 语句 256
 - SIGNAL 语句 295
 - Transact-SQL 过程 412
 - 在字符转换期间 411

错误处理

Transact-SQL 过程 412

D

DATE_FIRST_DAY_OF_WEEK 选项 354

DATE_FORMAT 选项 355

DATE_ORDER 选项 357

DBA 权限

帐户锁定 399

DBCC_LOG_PROGRESS

数据库选项 357

DBCC_PINNABLE_CACHE_PERCENT

数据库选项 358

dbisql

连接到数据库 66

选项 293

dbo 用户 ID

拥有的视图 171

DEALLOCATE DESCRIPTOR

语法 152

DEBUG_MESSAGES 选项

描述 358

DECLARE CURSOR 语句

Transact-SQL 语法 160

语法 155

DECLARE LOCAL TEMPORARY TABLE 语句

语法 161

DECLARE TEMPORARY TABLE 语句

语法 161

DECLARE 语句

语法 49, 154

DEDICATED_TASK 选项

描述 359

DEFAULT_DBSPACE 选项 359

DEFAULT_DISK_STRIPING 选项 361

DEFAULT_HAVING_SELECTIVITY_PPM 选项
361

DEFAULT_ISQL_ENCODING 选项

说明 362

DEFAULT_KB_PER_STRIPE 选项 363

DEFAULT_LIKE_MATCH_SELECTIVITY_PP
M 选项 363

DEFAULT_LIKE_RANGE_SELECTIVITY_PPM
选项 364

DEFAULT_PROXY_TABLE_ROW_COUNT 选
项 365

DEFAULT_TABLE_UDF_ROW_COUNT option
365

DELAYED_COMMIT_TIMEOUT 选项 365

DELAYED_COMMITS 选项 366

DELETE 语句

语法 163

DELETE (定位) 语句

SQL 语法 165

DESCRIBE 语句

语法 166

DISCONNECT 语句

语法 169

DISK_STRIPING 选项 366

DISTINCT 关键字 279

DIVIDE_BY_ZERO_ERROR 选项 366

DQP_ENABLED 选项 367

DROP CONNECTION 语句

语法 172

DROP DATABASE 语句

语法 173

DROP DATATYPE 语句

语法 170

DROP DBSPACE 语句

语法 170

DROP DOMAIN 语句

语法 170

DROP EVENT

语法 170

DROP EXTERNLOGIN 语句

语法 174

DROP FUNCTION 语句

语法 170

DROP INDEX 语句

语法 170

DROP LOGICAL SERVER 语句 176

DROP LOGIN POLICY 语句

语法 175

DROP MESSAGE

语法 170

DROP MULTIPLEX SERVER 语句 176

DROP PROCEDURE 语句

语法 170

DROP SERVER 语句

语法 177

DROP SERVICE 语句

语法 178

DROP STATEMENT 语句

语法 179

DROP TABLE

IDENTITY_INSERT 选项 171

DROP TABLE 语句
 语法 170
DROP TEXT CONFIGURATION 语句 180
DROP TEXT INDEX 语句 180
DROP USER 语句 180
DROP VARIABLE 语句
 语法 181
DROP VIEW 语句
 限制 171
 语法 170
DROP 语句
 语法 170
DUMMY 199
DYNAMIC SCROLL 游标 155
 代码页
 DEFAULT_ISQL_ENCODING 选项 362
 档案存档
 恢复 265
 导出数据
 SELECT 语句 277
 从表到文件 247
 登录
 口令有效期警告 416
 外部 89
 另请参见 连接
 登录策略
 创建 108
 更改 19, 21
 删除 175
 注释 61
 登录策略选项 393, 398, 399
 登录处理 416
 登录管理
 POST_LOGIN_PROCEDURE 选项 416
 登录管理功能 416
 低磁盘空间 83
 调度
 WAITFOR 309
 调度事件
 WAITFOR 语句 309
 调试
 DEBUG_MESSAGES 选项 358
 控制 **MESSAGE** 语句行为 242
 定位的 **DELETE** 语句
 SQL 语法 165
 多列索引 99, 102
 删除 36
 多行插入 183

多行读取
 FETCH 语句 190

E

EARLY_PREDICATE_EXECUTION 选项 367
ENABLE_LOB_VARIABLES 选项 368
END DECLARE STATEMENT
 语法 153
END PARALLEL IQ 语句 52
END 关键字 49
ESCAPE_CHARACTER 选项 328
EXCEPTION 语句
 语法 49
EXECUTE IMMEDIATE 语句
 语法 185
EXECUTE 语句
 Transact-SQL 184
 语法 182
EXIT 语句
 语法 187
EXTENDED_JOIN_SYNTAX 选项 368
 二进制大对象变量
 数据类型转换 368
 二进制数据
 控制隐式转换 346

F

FETCH 语句
 语法 188
FIRST
 返回一行 280
FLATTEN_SUBQUERIES 选项 437
FOR 语句
 语法 191
FORCE_DROP 选项 369
FORCE_NO_SCROLL_CURSORS 选项 369
FORCE_UPDATABLE_CURSORS 选项 370
FORWARD TO 语句
 语法 193
FP 索引
 分配的高速缓存 371
FP_LOOKUP_SIZE 选项 370
FP_LOOKUP_SIZE_PPM 选项 371
FP_PREDICATE_WORKUNIT_PAGES 选项 372
FPL_EXPRESSION_MEMORY_KB 选项 372
FROM contains-expression 198

FROM 子句 198, 199, 281
 SELECT 语句 281
 选自存储过程结果集 279
 语法 194
 FROM 子句的影响 199
 方法签名 122
 放置
 将行放入游标 255
 分隔符
 示例 100
 在 WD 索引中 100
 分离子查询谓词 282
 分区
 删除 36
 分区限制 343
 服务
 添加 133
 服务器
 创建 132
 创建逻辑 107
 更改 Multiplex 24
 更改 Web 服务 29
 逻辑 62
 删除逻辑 176
 符号链接 71
 复合语句
 关于 49
 复制
 过程 25

G

GARRAY_FILL_FACTOR_PERCENT 选项 373
 GARRAY_PAGE_SPLIT_PAD_PERCENT 选项 374
 GARRAY_PREFETCH_SIZE 选项 373, 374
 GET DESCRIPTOR 语句
 语法 200
 GOTO 语句
 Transact-SQL 201
 GRANT 语句
 CONNECT 权限 205
 INTEGRATED LOGIN 205
 KERBEROS LOGIN 205
 语法 202
 GROUP BY 子句
 SELECT 语句 282
 高度加密
 CREATE DATABASE 语句 74

更改

函数 14
 数据库 7

归类

SORT_COLLATION 选项 431
 客户端文件批量装载 229

过程 121, 129, 252

RAISERROR 语句 256
 sa_post_login_procedure 416
 Transact-SQL CREATE PROCEDURE 语句 118

创建 112
 代理 116
 动态 SQL 语句 185
 返回值 269
 复制 25
 结果集 115
 可变结果集 115
 删除 170
 选自结果集 279
 执行 184

H

HASH_THRASHING_PERCENT 选项 375

HEADER SKIP 选项

LOAD TABLE 语句 234

HG 索引

具有空值的多列 102
 空值 102
 提高查询性能 341

HG_DELETE_METHOD 选项 376

HG_SEARCH_RANGE 选项 377

HTTP_SESSION_TIMEOUT 选项 377

函数

创建 90
 更改 14
 删除 170
 用户定义的 269

和数据类型 198

缓冲区

禁用操作系统缓冲 413, 414

缓冲区高速缓存

分区 343

换行符

WD 索引分隔符 100

恢复操作

检验备份 266

恢复数据库
 检验备份 266

I

I/O
 直接 413, 414
IDENTITY 列
 和 DROP TABLE 171
IDENTITY_ENFORCE_UNIQUENESS 选项
 378
IDENTITY_INSERT 选项
 删除表 171
IF 语句
 Transact-SQL 209
 语法 207
IN_SUBQUERY_PREFERENCE 选项 383
INCLUDE 语句
 语法 210
INDENTITY_INSERT 选项 378
INDEX_ADVISOR 选项 379
INDEX_ADVISOR_MAX_ROWS 选项 381
INDEX_PREFERENCE 选项 382
INFER_SUBQUERY_PREDICATES 选项 383
INSERT
 宽 183
 语法 211
INSERT 语句
 ISOLATION LEVEL 215
 WORD SKIP 选项 217
INSTALL JAVA 语句
 语法 218
INTEGRATED LOGIN
 GRANT 语句 205
Interactive SQL
 OUTPUT 语句语法 247
 为读取和写入文件指定代码页 362
Interactive SQL 选项
 DEFAULT_ISQL_ENCODING 362
INTO 子句
 SELECT 语句 280
IQ UNIQUE
 替代方法 405
IQ UNIQUE 列约束 142
IQ UTILITIES 语句
 语法 220
IQ 存储
 保留空间 395
 保留临时空间 451

iq_dummy 199
iq_dummy 表 199
IQGOVERN_PRIORITY 选项 385
IQGOVERN_PRIORITY_TIME 选项 385
ISOLATION LEVEL
 INSERT 语句 215
ISOLATION_LEVEL 选项 386
isysserver 系统表 132

J

jar 文件
 安装 218
 删除 260
Java 122
 安装类 218
 删除类 260
Java VM
 启动 298
 停止 300
Java 表 UDF 127
Java 方法 122
JAVA_LOCATION 选项 386
JAVA_VM_OPTIONS 选项 387
jConnect
 禁用支持 7
 口令加密 214
 启用支持 7
JOIN_EXPANSION_FACTOR 选项 387
JOIN_OPTIMIZATION 选项 388
JOIN_PREFERENCE 选项 389
JOIN_SIMPLIFICATION_THRESHOLD 选项
 391
加密
 TDS 口令 214
加密算法
 CREATE DATABASE 语句 74
兼容性选项
 ASE_FUNCTION_BEHAVIOR 338
 CONTINUE_AFTER_RAISERROR 345
 CONVERSION_ERROR 346
 ON_TSQL_ERROR 412
监控
 启动和停止 220
 在 IQ UTILITIES 语句中 220
监控磁盘空间 83
监视器
 设置输出文件位置 407
剪裁尾随空白 230, 232

索引

检验备份 266
将数据库空间设置为联机状态 10
结果集 121, 129
 SELECT 来自 279
 变量 115, 252
静态游标
 声明 155

K

KERBEROS LOGIN
 GRANT 语句 205
Kerberos 鉴定
 COMMENT ON KERBEROS LOGIN 子句
 61
可变结果集 121, 129
 来自过程 115, 252
客户端文件批量装载
 错误 229
 回退 229
 字符集 229
空白
 剪裁尾随 232
 剪裁尾随空白 230
空间不足状况
 防止 395
空值
 在多列 HG 索引上 102
 在多列 HG 索引中 102
控制语句
 CALL 语句 55
 CASE 语句 57
 IF 语句 207
 LEAVE 语句 221
 LOOP 语句 241
 Transact-SQL GOTO 语句 201
 Transact-SQL IF 语句 209
 Transact-SQL WHILE 语句 312
口令
 TDS 加密 214
 加密 214
 有效期警告 416
 最小长度 406
口令加密
 jConnect 214
 TDS 214
块读取
 FETCH 语句 190
宽插入 183

L

LEAVE 语句
 语法 221
LF_BITMAP_CACHE_KB 选项 392
LIMIT 关键字
 SELECT 语句 279
LOAD TABLE 语句
 HEADER SKIP 选项 234
 ON PARTIAL INPUT ROW 选项 235
 QUOTES 选项 230
 STRIP 关键字 232
 USING 关键字 229
 WORD SKIP 选项 235
 不建议使用 FROM 子句 229
 新语法 232
 性能 232
 语法 223
 语法更改 232
LOAD_ZEROLENGTH_ASNULL 选项 392
LOB 变量
 数据类型转换 368
LOCK TABLE
 语法 239
LOCKED 选项 393
LOG_CONNECT 数据库选项 393
LOGIN_MODE 选项 394
LOGIN_PROCEDURE 选项 395
LONG BINARY 变量
 数据类型转换 368
LONG VARCHAR 变量
 数据类型转换 368
LOOP 语句
 语法 241
来自过程 121, 129
类
 安装 218
 删除 260
连接
 dbisql 169
 DEDICATED_TASK 选项 359
 FROM 子句语法 194
 SELECT 语句 281
 记录 393
 建立 19, 393
 逻辑服务器 20
 删除 163
 优化 387, 388, 391

- 优化连接顺序 401
- 连接列 198
- 连接索引
 - 创建 104
 - 同步 300
- 连接性能 198
- 联机
 - 数据库空间 10
- 链接
 - 符号 71
- 列
 - 别名在整个查询中都可以使用 280
 - 更改 31
 - 命名 3
 - 约束 142
 - 重命名 37
- 临时表 117, 146
 - 创建 136
 - 声明 161
 - 填充 281
- 临时空间
 - 为 IQ 存储保留 451
- 临时数据库空间
 - 创建 78
- 临时文件（目录）
 - TEMP_SPACE_LIMIT_CHECK 452
- 临时选项 315
- 路径
 - 相对 71
- 逻辑服务器
 - 更改 17
 - 连接 20
- 逻辑服务器策略
 - 更改 22

M

- MAIN_RESERVED_DBSPACE_MB 选项 395
- MAX_CARTESIAN_RESULT 选项 396–398
- MAX_CURSOR_COUNT 选项 399
- MAX_DAYS_SINCE_LOGIN 选项 399
- MAX_FAILED_LOGIN_ATTEMPTS 选项 399
- MAX_HASH_ROWS 选项 400
- MAX_IQ_GOVERN_PRIORITY 选项 384
- MAX_IQ_THREADS_PER_CONNECTION 选项 400
- MAX_IQ_THREADS_PER_TEAM 选项 401
- MAX_JOIN_ENUMERATION 选项 401
- MAX_PREFIX_PER_CONTAINS_PHRASE 选项 402
- MAX_QUERY_PARALLELISM 选项 402
- MAX_QUERY_TIME 选项 403
- MAX_STATEMENT_COUNT 选项 403
- MAX_TEMP_SPACE_PER_CONNECTION 选项 404
- MAX_WARNINGS 选项 405
- MDSR 加密算法
 - CREATE DATABASE 语句 74
- MESSAGE 语句
 - SQL 语法 242
 - 设置 DEBUG_MESSAGES 选项 358
- MIN_PASSWORD_LENGTH 选项 406
- MINIMIZE_STORAGE 选项 405
- MONITOR_OUTPUT_DIRECTORY 选项 407
- MPX_AUTOEXCLUDE_TIMEOUT 选项 408
- MPX_HEARTBEAT_FREQUENCY 选项 408
- MPX_IDLE_CONNECTION_TIMEOUT 选项 408
- MPX_MAX_CONNECTION_POOL_SIZE 选项 408
- MPX_MAX_UNUSED_POOL_SIZE 选项 408
- multiplex
 - renaming 23
- Multiplex
 - 命名存储 23
- Multiplex 数据库
 - 创建 71
 - 添加数据库空间 79
- 密码
 - 更改 205
- 描述符
 - DESCRIBE 语句 166
 - EXECUTE 语句 182
 - FETCH 语句 188
 - PREPARE 语句 251
 - 分配内存 5
 - 获取 200
 - 设置 290
 - 释放 152
- 描述符区
 - UPDATE（定位）语句 308
- 命令文件
 - 参数 250
- 命名管道 237
- 模式
 - 创建 131
 - 目录存储 281
 - 目录临时文件
 - 防止连接超出限额 452

N

NEAREST_CENTURY 选项 409
 NO RESULT SET 子句 115, 121
 NO SCROLL 游标 155
 NOEXEC 选项 409
 NON_ANSI_NULL_VARCHAR 选项 410
 NON_KEYWORDS 数据库选项 410
 NOTIFY_MODULUS 选项 411
 内存
 预取 341

O

ODBC
 ODBC_DISTINGUISH_CHAR_AND_VAR
 CHAR 选项 411
 静态游标 155
 ODBC_DISTINGUISH_CHAR_AND_VARCHA
 R 选项
 描述 411
 ON EXCEPTION RESUME 子句
 存储过程 412
 ON_CHARSET_CONVERSION_FAILURE 选项
 说明 411
 ON_ERROR 选项
 说明 412
 ON_TSQL_ERROR
 数据库选项 412
 OPEN 语句
 语法 245
 ORDER BY 子句 284
 OS_FILE_CACHE_BUFFERING 选项 413
 OS_FILE_CACHE_BUFFERING_TEMPDB 选项
 414
 OUTPUT 语句
 SQL 语法 247

P

PARAMETERS 语句
 语法 250
 PASSWORD_EXPIRY_ON_NEXT_LOGIN 选项
 415
 PASSWORD_GRACE_TIME 选项 415
 PASSWORD_LIFE_TIME 选项 416
 POST_LOGIN_PROCEDURE 选项 416
 PRECISION 选项 416
 PREFETCH 选项 417

PREFETCH_BUFFER_LIMIT 选项 417
 PREFETCH_BUFFER_PERCENT 选项 418
 PREFETCH_GARRAY_PERCENT 选项 418
 PREFETCH_SORT_PERCENT 选项 419
 PREPARE 语句
 语法 251
 PRESERVE_SOURCE_FORMAT 选项
 说明 419
 PRINT 语句
 Transact-SQL 语法 253
 PURGE 子句
 FETCH 语句 190
 PUT 语句
 SQL 语法 255
 批量装载 223

Q

QUERY_DETAIL 选项 420
 QUERY_NAME 选项 421
 QUERY_PLAN 选项 421
 QUERY_PLAN_AFTER_RUN 选项 422
 QUERY_PLAN_AS_HTML 选项 422
 QUERY_PLAN_AS_HTML_DIRECTORY 选项
 423
 QUERY_PLAN_TEXT_ACCESS 选项 424
 QUERY_PLAN_TEXT_CACHING 选项 425
 QUERY_ROWS_RETURNED_LIMIT 选项 426
 QUERY_TEMP_SPACE_LIMIT 选项 426
 QUERY_TIMING 选项 427
 QUIT 语句
 语法 187
 QUOTED_IDENTIFIER 选项 427
 启动
 Java VM 298
 数据库 295
 数据库服务器 297
 签名 122
 嵌入式 SQL
 DELETE (定位) 语句语法 165
 PUT 语句语法 255
 权限
 ALTER 205
 BACKUP 204
 CONNECT 权限 205
 DBA 204
 DBA 权限 206
 DELETE 205
 EXECUTE 205

- GRANT 语句 202
- GROUP 权限 204
- INSERT 205
- MEMBERSHIP 204
- MULTIPLEX ADMIN 204
- OPERATOR 204
- PERMS ADMIN 204
- PROFILE 204
- READCLIENTFILE 204
- READFILE 204
- REFERENCES 205
- RESOURCE 204
- RESOURCE 权限 204
- SELECT 205
- SPACE ADMIN 204
- UPDATE 205
- USER ADMIN 204
- VALIDATE 204
- WRITECLIENTFILE 204
- 撤销 270
- 对数据库空间的 CREATE 权限 206

R

- RAISERROR 语句
 - CONTINUE_AFTER_RAISERROR 选项 345
 - 语法 256
- READ 语句
 - 语法 258
- RECOVERY_TIME 选项 428
- REFERENCES 子句 35
- RELEASE SAVEPOINT 语句
 - 语法 259
- REMOVE 语句
 - 语法 260
- RESERVED_KEYWORDS 选项 428
- RESIGNAL 语句
 - 语法 261
- RESTORE 语句
 - COMPATIBLE 子句 266
 - VERIFY 子句 266
 - 检验备份 266
 - 提高速度 47
 - 语法 262
- RESTRICT 操作 144
- RESUME 语句
 - 语法 267
- RETURN 语句
 - 语法 269

- RETURN_DATE_TIME_AS_STRING 选项
 - 说明 429
- REVOKE 语句
 - 语法 270
- Rigndael 加密算法
 - CREATE DATABASE 语句 74
- ROLLBACK TO SAVEPOINT 语句
 - 语法 273
- ROLLBACK TRANSACTION 语句
 - Transact-SQL 274
 - 语法 274
- ROLLBACK 语句
 - 语法 272
- ROLLUP 运算符 282
 - SELECT 语句 282
- root 逻辑服务器策略 22
- ROW_COUNT 选项 430
- 软件包
 - 安装 218
 - 删除 260

S

- sa_conn_properties
 - 使用 316
- sa_dependent_views 系统过程 43
- sa_post_login_procedure 416
- SAVE TRANSACTION 语句
 - Transact-SQL 276
 - 语法 276
- SAVEPOINT 语句
 - 语法 275
- SCALE 选项 430
- SCROLL 游标 155
- SELECT * 35
- SELECT INTO
 - 在基表中返回结果 279
 - 在临时表中返回结果 279
 - 在宿主变量中返回结果 279
- SELECT 语句
 - FIRST 280
 - FROM 子句语法 194
 - LIMIT 279
 - TOP 280
 - 语法 277
- SET CONNECTION 语句
 - 语法 289
- SET DESCRIPTOR 语句
 - 语法 290

- SET OPTION 语句
 - dbisql 语法 330
 - 使用 315
 - 语法 291, 293
- SET SQLCA 语句
 - 语法 294
- SET TEMPORARY OPTION 语句
 - dbisql 语法 330
 - 使用 315
 - 语法 291, 293
- SET 语句
 - Transact-SQL 287
 - 语法 285
- SIGNAL 语句
 - 语法 295
- SIGNIFICANTDIGITSFORDOUBLEEQUALITY 选项 431
- SORT_COLLATION
 - 数据库选项 431
- sp_addmessage 110
- sp_dropuser 过程 271
- sp_iqcheckoptions 系统过程 316
- sp_login_environment 过程 395
- sp_tsql_environment 过程 395
- SQL
 - 常见语法元素 3
 - 语法约定 4
 - 语句指示符 4
- SQL 变量
 - SET VARIABLE 语句 285
 - 创建 149
 - 删除 181
- SQL 标准
 - 合规性 433, 434
- SQL 描述符区
 - 使用游标插入行 255
- SQL 语句
 - ALTER FUNCTION 语法 14
 - DELETE (定位) 语法 165
 - MESSAGE 语法 242
 - OUTPUT 语法 247
 - PUT 语法 255
 - UPDATE (定位) 语法 308
 - WAITFOR 语法 309
- SQL_FLAGGER_ERROR_LEVEL 选项 433
- SQL_FLAGGER_WARNING_LEVEL 选项 434
- SQLCA
 - INCLUDE 语句 210
 - SET SQLCA 语句 294
- SQLDA
 - DESCRIBE 语句 166
 - INCLUDE 语句 210
 - UPDATE (定位) 语句 308
 - 分配内存 5
 - 设置 290
 - 使用游标插入行 255
 - 释放 152
 - 执行语句 182
- START DATABASE 语句
 - 语法 295
- START ENGINE 语句
 - 语法 297
- START JAVA 语句
 - 语法 298
- STOP DATABASE 语句
 - 语法 298
- STOP ENGINE 语句
 - 语法 299
- STOP JAVA 语句
 - 语法 300
- STRING_RTRUNCATION 选项 434
- STRIP
 - LOAD TABLE 关键字 232
- STRIP 选项 230, 232
- SUBQUERY_CACHING_PREFERENCE 选项 435
- SUBQUERY_FLATTENING_PERCENT 选项 436
- SUBQUERY_FLATTENING_PREFERENCE 选项 437
- SUBQUERY_PLACEMENT_PREFERENCE 数据库选项 438
- SUPPRESS_TDS_DEBUGGING 选项
 - 说明 438
- SWEEPER_THREADS_PERCENT 选项 439
- SYNCHRONIZE JOIN INDEX 语句
 - 语法 300
- SYSTEM dbspace 199
- SYSTEM 数据库空间 281
- SYSWEBSERVICE 系统表
 - 添加服务器 29
- 删除 271
 - 从游标中删除行 165
 - 视图 171
 - 用户 270
- 删除分区 36

- 升级数据库 7
- 声明部分 153
- 实用程序语句 220
- 事件 83
 - 触发 301
 - 创建 82
 - 更改 13
 - 删除 170
- 事件处理程序
 - 触发 301
 - 创建 82
 - 更改 13
- 事务
 - ROLLBACK TO SAVEPOINT 语句 273
 - ROLLBACK TRANSACTION 语句 274
 - ROLLBACK 语句 272
 - SAVE TRANSACTION 语句 276
 - SAVEPOINT 语句 275
 - 提交 63
- 事务管理 63
 - BEGIN TRANSACTION 语句 53
 - 在 Transact-SQL 中 63
- 事务日志
 - TRUNCATE TABLE 语句 303
- 视图
 - 创建 151
 - 更改 41, 43
 - 关于 151
 - 删除 170, 171
 - 索引 101
 - 无效 43
 - 依赖性 43
 - 已更改的表 35
 - 重新编译无效的视图 43
- 数据
 - 从表导出到文件 247
- 数据库
 - 创建 68
 - 更改 7
 - 将数据装入 223
 - 禁用 jConnect 支持 7
 - 启动 295
 - 启用 jConnect 支持 7
 - 删除文件 173
 - 升级 7
 - 停止 298
- 数据库服务器
 - 启动 297
 - 停止 299
- 数据库空间
 - CREATE 权限 206
 - 创建 77
 - 更改 8
 - 删除 170
 - 设置脱机状态 10
 - 虚拟备份 45
- 数据库文件
 - 创建 77
 - 更改 8
- 数据库选项
 - DEBUG_MESSAGES 选项 358
 - DEDICATED_TASK 359
 - ENABLE_LOB_VARIABLES 368
 - ESCAPE_CHARACTER 328
 - FLATTEN_SUBQUERIES 437
 - FORCE_DROP 369
 - FP_LOOKUP_SIZE_PPM 371
 - ODBC_DISTINGUISH_CHAR_AND_VAR CHAR 411
 - ON_CHARSET_CONVERSION_FAILURE 411
 - POST_LOGIN_PROCEDURE 416
 - PRESERVE_SOURCE_FORMAT 419
 - RETURN_DATE_TIME_AS_STRING 429
 - SUBQUERY_FLATTENING_PERCENT 436
 - SUBQUERY_FLATTENING_PREFERENC E 437
 - SUPPRESS_TDS_DEBUGGING 438
 - TDS_EMPTY_STRING_IS_NULL 439
 - 持续时间 317
 - 初始设置 319
 - 游标 317
 - 字符串最大长度 292, 315
- 数据类型 198
 - 创建 80
 - 更改用户定义数据类型 12
 - 删除用户定义数据类型 170
- 数据类型转换
 - CONVERSION_MODE 选项 346
 - LONG BINARY 变量 368
 - 错误 346
- 宿主变量
 - 语法 3
- 所有者 3
- 索引 52
 - 表使用 101

- 查找页 371
- 创建 97
- 多列 102
- 多列 HG 和空值 102
- 命名 101
- 删除 170
- 所有者 101
- 唯一 99
- 锁
 - 使用 ROLLBACK 释放 272
- 锁定
 - 表 239
 - 自动 399
- T**
 - TDS
 - 口令加密 214
 - TDS_EMPTY_STRING_IS_NULL 选项
 - 说明 439
 - TEMP_EXTRACT_APPEND 选项 440
 - TEMP_EXTRACT_BINARY 选项 440
 - TEMP_EXTRACT_COLUMN_DELIMITER 选项 441
 - TEMP_EXTRACT_DIRECTORY 选项 442
 - TEMP_EXTRACT_ESCAPE_QUOTES 选项 443
 - TEMP_EXTRACT_NAME1 选项 443
 - TEMP_EXTRACT_NAME2 选项 443
 - TEMP_EXTRACT_NAME3 选项 443
 - TEMP_EXTRACT_NAME4 选项 443
 - TEMP_EXTRACT_NAME5 选项 443
 - TEMP_EXTRACT_NAME6 选项 443
 - TEMP_EXTRACT_NAME7 选项 443
 - TEMP_EXTRACT_NAME8 选项 443
 - TEMP_EXTRACT_NAME_n 选项 443
 - TEMP_EXTRACT_NULL_AS_EMPTY 选项 445
 - TEMP_EXTRACT_NULL_AS_ZERO 选项 446
 - TEMP_EXTRACT_QUOTE 选项 447
 - TEMP_EXTRACT_QUOTES 选项 447
 - TEMP_EXTRACT_QUOTES_ALL 选项 448
 - TEMP_EXTRACT_ROW_DELIMITER 选项 449
 - TEMP_EXTRACT_SIZE1 选项 449
 - TEMP_EXTRACT_SIZE2 选项 449
 - TEMP_EXTRACT_SIZE3 选项 449
 - TEMP_EXTRACT_SIZE4 选项 449
 - TEMP_EXTRACT_SIZE5 选项 449
 - TEMP_EXTRACT_SIZE6 选项 449
 - TEMP_EXTRACT_SIZE7 选项 449
 - TEMP_EXTRACT_SIZE8 选项 449
 - TEMP_EXTRACT_SIZE_n 选项 449
 - TEMP_EXTRACT_SWAP 选项 450
 - TEMP_RESERVED_DBSPACE_MB
 - 数据库选项 451
 - TEMP_SPACE_LIMIT_CHECK
 - 数据库选项 452
 - TEXT_DELETE_METHOD 选项 453
 - TIME_FORMAT 选项 453
 - TIMESTAMP_FORMAT 选项 454
 - TOP
 - 指定行数 280
 - TOP_NSORT_CUTOFF_PAGES 选项 455
 - Transact-SQL
 - BEGIN TRANSACTION 语句 53
 - COMMIT TRANSACTION 63
 - CREATE MESSAGE 110
 - CREATE PROCEDURE 语句 118
 - CREATE SCHEMA 语句 131
 - ROLLBACK TRANSACTION 语句 274
 - SAVE TRANSACTION 语句 276
 - SET 语句 287
 - 处理时出错 256
 - 过程 118
 - 兼容性选项 327
 - 执行存储过程 184
 - TRIGGER EVENT
 - 语法 301
 - TRIM_PARTIAL_MBC 选项 455
 - TRUNCATE TABLE 语句
 - 语法 302
 - TSQL_VARIABLES 选项 456
 - 停止
 - Java VM 300
 - 停止数据库 298
 - 投影
 - SELECT 语句 279
 - 脱机
 - 数据库空间 10
- U**
 - UNION 操作 303
 - UPDATE (定位) 语句
 - SQL 语法 308
 - USER_RESOURCE_RESERVATION 选项 456
 - USING
 - LOAD TABLE 关键字 229

USING FILE 子句
LOAD TABLE 语句 229

V

VARCHAR 数据类型
转换为压缩格式 352
VERIFY_PASSWORD_FUNCTION 选项 457

W

WAIT_FOR_COMMIT 选项 458
WAITFOR 语句
SQL 语法 309
WASH_AREA_BUFFERS_PERCENT 数据库选项 458
WD 索引
CHAR 列 101
分隔符 100
WD_DELETE_METHOD 选项 459
WHENEVER 语句
语法 311
WHERE 子句
SELECT 语句 281
WHILE 语句
Transact-SQL 312
语法 241
WITH HOLD 子句
OPEN 语句 245
WORD SKIP 选项
INSERT 语句 217
LOAD TABLE 语句 235
外部存储过程
创建 120, 127, 128
外部过程
创建 120, 127, 128
外键
完整性约束 143
未命名 143
唯一
约束 141
唯一索引 99
尾随空白
剪裁 230, 232
谓词
分离 282
文本搜索 198
文件
将数据从表导出到 247

设置联机状态 10
设置脱机状态 10
数据库空间 8, 77

X

系统表 199
PRESERVE_SOURCE_FORMAT 419
SYSFILE 267
源列 419
系统过程
sa_dependent_views 43
显示
消息 242
相对路径 71
相关名
在 DELETE 语句中 163
消息
创建 110
删除 170
显示 242
行
从游标中删除行 165
使用游标插入 255
性能 199
获取更多内存 341
虚拟 IQ 表 199
选项 199
AGGREGATION_PREFERENCE 332
ASE_FUNCTION_BEHAVIOR 338
CIS_ROWSET_SIZE 345
CONTINUE_AFTER_RAISERROR 345
CONVERSION_ERROR 346
DEBUG_MESSAGES 选项 358
DEDICATED_TASK 359
DEFAULT_ISQL_ENCODING 362
DQP_ENABLED 367
ENABLE_LOB_VARIABLES 368
ESCAPE_CHARACTER 328
EXTENDED_JOIN_SYNTAX 368
FLATTEN_SUBQUERIES 437
FORCE_DROP 369
FP_LOOKUP_SIZE 370
FP_LOOKUP_SIZE_PPM 371
MAX_PREFIX_PER_CONTAINS_PHRASE 402
MAX_TEMP_SPACE_PER_CONNECTION 404
MPX_AUTOEXCLUDE_TIMEOUT 408

MPX_HEARTBEAT_FREQUENCY 408
 MPX_IDLE_CONNECTION_TIMEOUT
 408
 MPX_MAX_CONNECTION_POOL_SIZE
 408
 MPX_MAX_UNUSED_POOL_SIZE 408
 NON_ANSI_NULL_VARCHAR 410
 ODBC_DISTINGUISH_CHAR_AND_VAR
 CHAR 411
 ON_CHARSET_CONVERSION_FAILURE
 411
 ON_ERROR 412
 ON_TSQL_ERROR 412
 POST_LOGIN_PROCEDURE 416
 PRESERVE_SOURCE_FORMAT 419
 RETURN_DATE_TIME_AS_STRING 429
 SORT_COLLATION 431
 sp_iqcheckoptions 316
 SUBQUERY_CACHING_PREFERENCE
 435
 SUBQUERY_FLATTENING_PERCENT
 436
 SUBQUERY_FLATTENING_PREFERENC
 E 437
 SUPPRESS_TDS_DEBUGGING 438
 SYSOPTIONDEFAULTS 系统表 316
 TDS_EMPTY_STRING_IS_NULL 439
 TEXT_DELETE_METHOD 453
 Transact-SQL 287
 不建议使用 320
 查找值 316
 常规数据库 320
 持续时间 317
 初始设置 319
 登录策略 21
 范围 317
 兼容性 327
 简介 315
 列表 332
 设置 291, 315
 设置 dbisql 选项 64
 设置 DBISQL 选项 64
 设置临时 293, 330
 意外行为 281
 优先级 317
 游标 317
 选项值
 截断 292, 315

选择列表

DESCRIBE 语句 166
 SELECT 语句 280

Y

意外行为 199
 用户 271
 创建 147
 更改 39
 删除 180, 270
 用户 ID
 撤消 270
 更改口令 205
 用户定义的函数
 RETURN 语句 269
 用户定义的数据类型
 更改 12
 用户定义数据类型
 CREATE DOMAIN 语句 80
 删除 170
 用于外部过程的 CREATE PROCEDURE 语句
 语法 120, 127, 128
 用于组件集成服务的远程服务器 132
 优化
 MAX_HASH_ROWS 选项 400
 MAX_JOIN_ENUMERATION 选项 401
 定义现有表和 87
 由 SQL Anywhere 进行处理 199
 游标
 DESCRIBE 166
 FOR UPDATE 子句 157
 INSENSITIVE 155
 OPEN 语句 245
 WITH HOLD 子句 246
 读取 188
 关闭 59
 敏感性 158
 删除行 165
 声明 155, 160
 使用游标插入 255
 数据库选项 317
 循环 191
 语法
 常见元素 3
 语法错误
 连接 368
 语法定义 4

语句

- ALTER FUNCTION 语法 14
- DELETE (定位) 语法 165
- MESSAGE 语法 242
- OUTPUT 语法 247
- PUT 语法 255
- UPDATE (定位) 语法 308
- WAITFOR 语法 309

- 语句标签 3, 201

- 语句指示符 4

预取

- BT_PREFETCH_MAX_MISS 341

预准备语句

- EXECUTE 语句 182

- 删除 179

域 80

- 更改 12

原始设备

- 命名 71

远程服务器

- 连接 213

- 远程数据访问 17, 28, 307

- CIS_ROWSET_SIZE 345

Z

- 在不足时通知 83

- 直接 I/O 413, 414

只读

- 锁定表 239

- 指示符变量 3

制表符

- WD 索引分隔符 100

主机变量

- 声明 153

主键

- 完整性约束 143

主控台

- 显示消息 242

转义字符

- OUTPUT SQL 语句 247

装载

- 可伸缩性 343

子查询

- 分离 282

字符串

- 数据库选项的长度 292, 315

字符大对象变量

- 数据类型转换 368

字符集

- 客户端文件批量装载 229

- 转换错误 411

自动增量

- 主键值 183

组合 52

