



Adaptive Server[®] Anywhere Quick Reference

Part number: 38178-01-0900-01

Last modified: June 2003

Copyright © 1989–2003 Sybase, Inc. Portions copyright © 2001–2003 iAnywhere Solutions, Inc. All rights reserved.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of iAnywhere Solutions, Inc. iAnywhere Solutions, Inc. is a subsidiary of Sybase, Inc.

For a list of Sybase trademarks, see other books in the SQL Anywhere Studio documentation set.

Contents

SQL Language Elements	1
SQL Data Types	3
SQL Functions	5
SQL Statements	13

SQL Language Elements

Expressions

expression:

case-expression
| *constant*
| [*correlation-name*].*column-name*
| - *expression*
| *expression operator expression*
| (*expression*)
| *function-name* (*expression*, ...)
| *if-expression*
| *special value*
| (*subquery*)
| *variable-name*

case-expression:

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

alternative form of case-expression:

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

constant:

integer | *number* | *string* | *host-variable*

special-value:

CURRENT { **DATE** | **TIME** | **TIMESTAMP** }
| **NULL**
| **SQLCODE**
| **SQLSTATE**
| **USER**

if-expression:

IF *condition*
THEN *expression*
[**ELSE** *expression*]
ENDIF

operator:

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

Search conditions

search-condition:

```
expression compare expression
| expression compare { [ ANY | SOME ] | ALL } ( subquery )
| expression IS [ NOT ] NULL
| expression [ NOT ] BETWEEN expression AND expression
| expression [ NOT ] LIKE expression [ ESCAPE expression ]
| expression [ NOT ] IN ( { expression
    | subquery
    | value-expr1 , value-expr2 [ , value-expr3 ] ... } )
| EXISTS ( subquery )
| NOT condition
| search-condition AND search-condition
| search-condition OR search-condition
| ( search-condition )
| ( search-condition , estimate )
| search-condition IS [ NOT ] { TRUE | FALSE | UNKNOWN }
| trigger-operation
```

compare:

```
= | > | < | >= | <= | <> | != | !< | !>
```

trigger-operation:

```
INSERTING | DELETING
| UPDATING( column-name-string ) | UPDATE( column-name )
```

SQL Data Types

Character data types

{ CHAR | CHARACTER } [(*max-length*)]

{ VARCHAR | CHARACTER VARYING } [(*max-length*)]

LONG VARCHAR

TEXT

UNIQUEIDENTIFIER

Numeric data types

[UNSIGNED] BIGINT

{ DECIMAL | DEC } [(*precision* [, *scale*])]

DOUBLE [PRECISION]

FLOAT [(*precision*)]

[UNSIGNED] { INT | INTEGER }

NUMERIC [(*precision* [, *scale*])]

REAL

[UNSIGNED] SMALLINT

[UNSIGNED] TINYINT

Money data types

MONEY

SMALLMONEY

BIT data type

BIT

Date and time data types

DATE

DATETIME

SMALLDATETIME

TIME

TIMESTAMP

Binary data types

BINARY [(*max-length*)]

LONG BINARY

IMAGE

UNIQUEIDENTIFIER

VARBINARY [(*max-length*)]

SQL Functions

ABS (*numeric-expression*)

ACOS (*numeric-expression*)

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

ASCII (*string-expression*)

ASIN (*numeric-expression*)

ATAN (*numeric-expression*)

{ ATN2 | ATAN2 } (*numeric-expression1*, *numeric-expression2*)

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

BYTE_LENGTH (*string-expression*)

BYTE_SUBSTR (*string-expression*, *start* [, *length*])

CAST (*expression* **AS** *data type*)

CEILING (*numeric-expression*)

CHAR (*integer-expression*)

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

CHAR_LENGTH (*string-expression*)

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

COMPARE (
 string-expression-1,
 string-expression-2
 [, *collation-name* | , *collation-id*])

CONNECTION_PROPERTY (
 { *integer-expression-1* | *string-expression* }
 [, *integer-expression-2*])

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

COS (*numeric-expression*)

COT (*numeric-expression*)

COUNT (
 *
 | *expression*
 | **DISTINCT** { *expression* | *column-name* })

CSCONVERT (
 string-expression,
 '*target-charset*'
 [, '*source-charset*'])

DATALength (*expression*)

DATE (*expression*)

DATEADD (*date-part*, *numeric-expression*, *date-expression*)

date-part :

year | **quarter** | **month** | **week** | **day** | **hour** | **minute** | **second** | **millisecond**

DATEDIFF (*date-part*, *date-expression1*, *date-expression2*)

date-part :

year | **quarter** | **month** | **week** | **day** | **hour** | **minute** | **second** | **millisecond**

DATEFORMAT (*datetime-expression*, *string-expression*)

DATENAME (*date-part*, *date-expression*)

DATEPART (*date-part*, *date-expression*)

DATETIME (*expression*)

DAY (*date-expression*)

DAYNAME(*date-expression*)

DAYS ([*datetime-expression*,] *datetime-expression*)

DAYS (*datetime-expression*, *integer-expression*)

DB_ID ([*database-name*])

DB_NAME ([*database-id*])

DB_EXTENDED_PROPERTY (
{ *property_id* | *property_name* },
[, *property-specific_argument*
[, { *database_id* | *database_name* }]])

DB_PROPERTY (
{ *property_id* | *property_name* }
[, { *database_id* | *database_name* }])

DEGREES (*numeric-expression*)

DIFFERENCE (*string-expression-1*, *string-expression-2*)

DOW (*date-expression*)

ERRORMSG ([*sqlstate* | *sqlcode*])

sqlstate: *string*

sqlcode: *integer*

ESTIMATE (*column-name* [, *value* [, *relation-string*]])

ESTIMATE_SOURCE (
column-name
[, *value* [, *relation-string*]])

EVENT_CONDITION (*condition-name*)

EVENT_CONDITION_NAME (*integer*)

EVENT_PARAMETER (*context-name*)

```

context-name:
  'AppInfo'
| 'ConnectionID'
| DisconnectReason
| 'EventName'
| 'Executions'
| 'NumActive'
| 'ScheduleName'
| 'TableName'
| 'User'
| condition-name

EXP ( numeric-expression )

EXPERIENCE_ESTIMATE (
  column-name
  [ , value [ , relation-string ] ] )

EXPLANATION (
  string-expression [ cursor-type ],
  update-status )

EXPRTYPE ( string-expression, integer-expression )

FLOOR ( numeric-expression )

GET_IDENTITY ( [ owner.] table-name [ , num_to_alloc ],... )

GETDATE ()

GRAPHICAL_PLAN (
  string-expression [ statistics-level ],
  [ cursor-type ],
  [ update-status ] )

GRAPHICAL_ULPLAN ( string-expression )

GREATER ( expression1, expression2 )

GROUPING ( group-by-expression )

HEXTOINT ( hexadecimal-string )

HOUR ( datetime-expression )

HOURS ( [ datetime-expression, ] datetime-expression )

HOURS ( datetime-expression, integer-expression )

HTTP_HEADER ( field-name )

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

IDENTITY ( expression )

IFNULL ( expression-1, expression-2 [ , expression-3 ] )

INDEX_ESTIMATE( column-name, number [ , relation-string ] )

INSERTSTR (
  integer-expression,
  string-expression-1,
  string-expression-2 )

INTTOHEX ( integer-expression )

ISDATE ( string )

```

ISNULL (*expression*, *expression* [, ...])

ISNUMERIC (*string*)

LCASE (*string-expression*)

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

LENGTH (*string-expression*)

LESSER (*expression1*, *expression2*)

LIST (
{ *string-expression* | **DISTINCT** *column-name* }
[, *delimiter-string*]
[**ORDER BY** *order-by-expression*])

LOCATE (*string-expression-1*, *string-expression-2* [, *integer-expression*])

LOG (*numeric-expression*)

LOG10 (*numeric-expression*)

LONG_ULPLAN (*string-expression*)

LOWER (*string-expression*)

LTRIM (*string-expression*)

MAX (*expression* | **DISTINCT** *column name*)

MIN (*expression*
| **DISTINCT** *column name*)

MINUTE (*datetime-expression*)

MINUTES ([*datetime-expression*,] *datetime-expression*)

MINUTES (*datetime-expression*, *integer-expression*)

MOD (*dividend*, *divisor*)

MONTH (*date-expression*)

MONTHNAME (*date-expression*)

MONTHS ([*datetime-expression*,] *datetime-expression*)

MONTHS (*datetime-expression*, *integer-expression*)

NEWID()

NEXT_CONNECTION ([*connection-id*] [, *database-id*])

NEXT_DATABASE ({ **NULL** | *database-id* })

NEXT_HTTP_HEADER (*header-name*)

NEXT_HTTP_VARIABLE (*var-name*)

NOW (*)

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

NUMBER (*)

OPENXML (*xml-expression*,
xpath-query [, *flags* [, *namespace-declaration*]])
WITH (*column-name* *column-type* [*xpath-query*],...)

PATINDEX (' %*pattern*% ' , *string-expression*)

PI (*)

PLAN (*string-expression*, [*cursor-type*], [*update-status*])

POWER (*numeric-expression-1*, *numeric-expression-2*)

PROPERTY_DESCRIPTION ({ *property-id* | *property-name* })

PROPERTY ({ *property-id* | *property-name* })

PROPERTY_NAME (*property-id*)

PROPERTY_NUMBER (*property-name*)

QUARTER(*date-expression*)

RADIANS (*numeric-expression*)

RAND ([*integer-expression*])

REMAINDER (*dividend*, *divisor*)

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

REPLACE (*original-string*, *search-string*, *replace-string*)

REPLICATE (*string-expression*, *integer-expression*)

REWRITE (*select-statement* [, ' ANSI '])

RIGHT (*string-expression*, *integer-expression*)

ROUND (*numeric-expression*, *integer-expression*)

RTRIM (*string-expression*)

SECOND (*datetime-expression*)

SECONDS ([*datetime-expression*,] *datetime-expression*)

SECONDS (*datetime-expression*, *integer-expression*)

SHORT_ULPLAN (*string-expression*)

SIGN (*numeric-expression*)

SIMILAR (*string-expression-1*, *string-expression-2*)

SIN (*numeric-expression*)

SORTKEY (*string-expression*
[, *collation-name* | , *collation-id*])

SOUNDEX (*string-expression*)

SPACE (*integer-expression*)

SQLDILECT (*sql-statement-string*)

SQRT (*numeric-expression*)

STDDEV_POP (*numeric-expression*)

STDDEV_SAMP (*numeric-expression*)

STR (*numeric-expression* [, *length* [, *decimal*]])

STRING (*string-expression* [, ...])

STRTOUUID(*string-expression*)

STUFF (*string-expression1*, *start*, *length*, *string-expression2*)

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

SUM (*expression* | **DISTINCT** *column-name*)

TAN (*numeric-expression*)

TEXTPTR (*column-name*)

TODAY (*)

TRACEBACK (*)

TRANSACTSQL(*sql-statement-string*)

TRIM (*string-expression*)

"**TRUNCATE**" (*numeric-expression*, *integer-expression*)

TRUNCNUM (*numeric-expression*, *integer-expression*)

UCASE (*string-expression*)

UPPER (*string-expression*)

UUIDTOSTR(*uuid-expression*)

VAR_POP (*numeric-expression*)

VAR_SAMP (*numeric-expression*)

VAREXISTS (*variable-name-string*)

WATCOMSQL(*sql-statement-string*)

WEEKS ([*datetime-expression*,] *datetime-expression*)

WEEKS (*datetime-expression*, *integer-expression*)

XMLAGG (*value-expression* [**ORDER BY** *order-by-expression*],...)

XMLCONCAT (*xml-value*,...)

XMLELEMENT (**NAME** *element-name-expression*
[, **XMLATTRIBUTES** (*attribute-value-expression*
[**AS** *attribute-name*],...)
[, *element-content-expression*,...])

XMLFOREST (*element-content-expression* [**AS** *element-name*],...)

XMLGEN (*xquery-constructor*, *content-expression* [**AS** *variable-name*],...)

YEARS ([*datetime-expression*,] *datetime-expression*)

YEARS (*datetime-expression*, *integer-expression*)

YMD (
integer-expression,
integer-expression,
integer-expression)

SQL Statements

ALLOCATE DESCRIPTOR statement [ESQL]

ALLOCATE DESCRIPTOR *descriptor-name*
 [**WITH MAX** { *integer* | *hostvar* }]

descriptor-name : *string*

ALTER DATABASE statement

ALTER DATABASE
 [**UPGRADE** [**JAVA** { **ON** | **OFF** | **JDK** { '1.1.8' | '1.3' } }]
 [**JCONNECT** { **ON** | **OFF** }]
 | **REMOVE JAVA**]

ALTER DATABASE
 { **CALIBRATE** [**SERVER**]
 | **CALIBRATE DBSPACE** *dbspace-name*
 | **CALIBRATE DBSPACE TEMPORARY**
 | **RESTORE DEFAULT CALIBRATION**
 }

ALTER DATABASE *dbfile*
MODIFY [**TRANSACTION**] **LOG**
 { { **OFF** | **ON** } { *log-name* | *log-name MIRROR mirror-name* | **MIRROR mirror-name** } }
 [**KEY** *key* [**ALGORITHM** *algorithm*]]

ALTER DBSPACE statement

ALTER DBSPACE { *dbspace-name* | **TRANSLOG** | **TEMPORARY** }
 { **ADD** *number* [**PAGES** | **KB** | **MB** | **GB** | **TB**]
 | **RENAME** *filename-string* }

ALTER EVENT statement

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**]

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*]
 { **START TIME** *start-time* | **BETWEEN** *start-time* **AND** *end-time* }
 [**EVERY** *period* { **HOURS** | **MINUTES** | **SECONDS** }]
 [**ON** { (*day-of-week*, ...) | (*day-of-month*, ...) }]
 [**START DATE** *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

ALTER FUNCTION statement

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

function-definition:

CREATE FUNCTION syntax following the name

ALTER FUNCTION [*owner*.]*function-name* **SET HIDDEN**

ALTER INDEX statement

ALTER { *index-spec* { *rename-clause* | *cluster-clause* }
| *foreign-key-spec* { *rename-clause* | *cluster-clause* }
| *primary-key-spec* *cluster-clause* }

index-spec :

INDEX *index-name* **ON** [*owner*.]*table-name*

foreign-key-spec :

[**INDEX**] **FOREIGN KEY** *role-name* **ON** [*owner*.]*table-name*

primary-key-spec :

[**INDEX**] **PRIMARY KEY** **ON** [*owner*.]*table-name*

rename-clause :

RENAME [**AS** | **TO**] *index-name*

cluster-clause :

CLUSTERED | **NONCLUSTERED**

ALTER PROCEDURE statement

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

procedure-definition:

CREATE PROCEDURE syntax following the name

ALTER PROCEDURE [*owner*.]*procedure-name*

REPLICATE { **ON** | **OFF** }

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

ALTER PUBLICATION statement

ALTER PUBLICATION [*owner*.]*publication-name* *alterpub-clause*, ...

alterpub-clause:

ADD TABLE *article-description*

| **MODIFY TABLE** *article-description*

| { **DELETE** | **DROP** } **TABLE** [*owner*.]*table-name*

| **RENAME** *publication-name*

owner, *publication-name*, *table-name* : *identifier*

```

article-description :
table-name [ ( column-name, ... ) ]
[ WHERE search-condition ]
[ SUBSCRIBE BY expression ]

```

ALTER REMOTE MESSAGE TYPE statement [SQL Remote]

```

ALTER REMOTE MESSAGE TYPE message-system
ADDRESS address

message-system: FILE | FTP | MAPI | SMTP | VIM

address: string

```

ALTER SERVER statement

```

ALTER SERVER server-name
[ CLASS 'server-class' ]
[ USING 'connection-info' ]
[ CAPABILITY 'cap-name' { ON | OFF } ]

server-class :
  ASAJDBC | ASEJDBC
  | ASAODBC | ASEODBC
  | DB2ODBC | MSSODBC
  | ORAODBC | ODBC

connection-info :
machine-name:port-number[/dbname] | data-source-name

```

ALTER SERVICE statement

```

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

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' }

```

ALTER SYNCHRONIZATION SUBSCRIPTION statement [MobiLink]

```

ALTER SYNCHRONIZATION SUBSCRIPTION
TO publication-name
[ FOR ml_username, ... ]
[ TYPE sync-type ]
[ ADDRESS network-parameters ]
[ ADD OPTION option=value, ... ]
[ MODIFY OPTION option=value, ... ]
[ DELETE { ALL OPTION | OPTION option, ... } ]

ml_username: identifier

network-parameters: string

sync-type: http | https | tcpip | ActiveSync

value: string | integer

```

ALTER SYNCHRONIZATION USER statement [MobiLink]

```
ALTER SYNCHRONIZATION USER ml_username
[ TYPE sync-type ]
[ ADDRESS network-parameters ]
[ ADD OPTION option=value, ... ]
[ MODIFY OPTION option=value, ... ]
[ DELETE { ALL OPTION | OPTION option } ]
```

ml_username: identifier

network-parameters: string

sync-type: http | https | tcpip | ActiveSync

value: string | integer

ALTER TABLE statement

```
ALTER TABLE [ owner. ] table-name
{ add-clause | modify-clause | drop-clause | rename-clause }
```

add-clause :

```
    ADD { column-definition | table-constraint }
| { ADD PCTFREE integer | PCTFREE DEFAULT }
```

modify-clause :

```
    MODIFY column-definition
| MODIFY column-name { DEFAULT default-value
| [ NOT | NULL
| [ CONSTRAINT constraint-name
| CHECK { NULL | ( new-condition ) } }
| ALTER column-name column-modification
| ALTER constraint-name CHECK ( new-condition )
```

drop-clause :

```
{ DELETE | DROP } {
    column-name
| CONSTRAINT constraint-name
| CHECK
| UNIQUE ( column-name, ... )
| PRIMARY KEY
| FOREIGN KEY role-name }
```

rename-clause :

```
    RENAME new-table-name
| RENAME column-name TO new-column-name
| RENAME constraint-name TO new-constraint-name
```

column-definition :

```
column-name data-type [ NOT NULL ] [ DEFAULT default-value ] [ column-
constraint ... ]
```

table-constraint :

```
[ CONSTRAINT constraint-name ] { UNIQUE ( column-name, ... )
| PRIMARY KEY [ CLUSTERED ] ( column-name, ... )
| foreign-key-constraint
| CHECK ( condition ) }
```

```

column-constraint :
[ CONSTRAINT constraint-name ] { UNIQUE
| PRIMARY KEY
| REFERENCES table-name
  [ ( column-name ) ] [ actions ] [ CLUSTERED ]
| CHECK ( condition ) }
| COMPUTE ( expression )

column-modification :
SET DEFAULT default-value
| DROP DEFAULT
| ADD [ CONSTRAINT column-constraint-name ] CHECK ( condition )
| { DELETE | DROP } CONSTRAINT column-constraint-name
| { DELETE | DROP } CHECK
| SET COMPUTE ( expression )
| DROP COMPUTE

default-value :
special-value
| string
| global variable
| [ - ] number
| ( constant-expression )
| built-in-function( constant-expression )
| AUTOINCREMENT
| GLOBAL AUTOINCREMENT [ ( partition-size ) ]
| NULL
| TIMESTAMP
| UTC TIMESTAMP
| LAST USER
| USER

special-value:
CURRENT { DATABASE | DATE
| REMOTE USER | TIME
| TIMESTAMP | UTC TIMESTAMP
| USER | PUBLISHER }

foreign-key-constraint :
[ NOT NULL ] FOREIGN KEY [ role-name ] [ ( column-name, ... ) ]
REFERENCES table-name [ ( column-name, ... ) ] [ CLUSTERED ]
[ actions ] [ CHECK ON COMMIT ]

actions :
[ ON UPDATE action ] [ ON DELETE action ]

action :
CASCADE | SET NULL | SET DEFAULT | RESTRICT

ALTER TABLE [ owner. ] table-name REPLICATE { ON | OFF }

```

ALTER TRIGGER statement

```
ALTER TRIGGER trigger-name trigger-definition
```

trigger-definition :

```
CREATE TRIGGER syntax following the trigger name
```

```
ALTER TRIGGER trigger-name ON [ owner. ] table-name SET HIDDEN
```

ALTER VIEW statement

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

ALTER VIEW
[*owner.view-name* **SET HIDDEN**

ALTER WRITEFILE statement

ALTER WRITEFILE *write-file-name*
REFERENCES *db-file-name* [**KEY** *key*]

write-file-name | *db-file-name* : *string*

BACKUP statement

BACKUP DATABASE
DIRECTORY *backup-directory*
[**WAIT BEFORE START**]
[**WAIT AFTER END**]
[**DBFILE ONLY**]
[**TRANSACTION LOG ONLY**]
[**TRANSACTION LOG RENAME** [**MATCH**]]
[**TRANSACTION LOG TRUNCATE**]

backup-directory : *string*

BACKUP DATABASE TO *archive-root*
[**ATTENDED** { **ON** | **OFF** }]
[**WITH COMMENT** *comment string*]

archive-root : *string*

comment-string : *string*

BEGIN statement

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

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

variable-declaration :
DECLARE *variable-name* *data-type*

exception-declaration :
DECLARE *exception-name* **EXCEPTION**
FOR SQLSTATE [**VALUE**] *string*

exception-case :
WHEN *exception-name* [, ...] **THEN** *statement-list*
| **WHEN OTHERS THEN** *statement-list*

BEGIN TRANSACTION statement

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

CALL statement

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

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

CASE statement

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

CASE
WHEN [*search-condition* | **NULL**] **THEN** *statement-list* ...
[**WHEN** [*search-condition* | **NULL**] **THEN** *statement-list*] ...
[**ELSE** *statement-list*]
END CASE

CHECKPOINT statement

CHECKPOINT

CLEAR statement [Interactive SQL]

CLEAR

CLOSE statement [ESQL] [SP]

CLOSE *cursor-name*

cursor-name : *identifier* | *hostvar*

COMMENT statement

COMMENT ON
{
 COLUMN [*owner*.] *table-name.column-name*
 | **EVENT** *event-name*
 | **FOREIGN KEY** [*owner*.] *table-name.role-name*
 | **INDEX** [[*owner*.] *table*.] *index-name*
 | **JAVA CLASS** *java-class-name*
 | **JAVA JAR** *java-jar-name*
 | **LOGIN** *integrated_login_id*
 | **PROCEDURE** [*owner*.] *procedure-name*
 | **SERVICE** *web-service-name*
 | **TABLE** [*owner*.] *table-name*
 | **TRIGGER** [[*owner*.] *tablename*.] *trigger-name*
 | **USER** *userid*
 | **VIEW** [*owner*.] *view-name*
}
IS *comment*

comment : *string* | **NULL**

COMMIT statement

COMMIT [**WORK**]

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

CONFIGURE statement [Interactive SQL]

CONFIGURE

CONNECT statement [ESQL] [Interactive SQL]

CONNECT

[**TO** *engine-name*]

[**DATABASE** *database-name*]

[**AS** *connection-name*]

[**USER**] *userid* **IDENTIFIED BY** *password*

engine-name, *database-name*, *connection-name*, *userid*, *password* :

{ *identifier* | *string* | *hostvar* }

CONNECT USING *connect-string*

connect-string : { *identifier* | *string* | *hostvar* }

CREATE COMPRESSED DATABASE statement

CREATE [**COMPRESSED** | **EXPANDED**] **DATABASE** *new-db-file-name*
FROM *old-db-file-name* [**KEY** *key*]

CREATE DATABASE statement

CREATE DATABASE *db-file-name*

[[**TRANSACTION**] { **LOG OFF** | **LOG ON** } [*log-file-name-string*]
[**MIRROR** *mirror-file-name-string*]]

[**CASE** { **RESPECT** | **IGNORE** }]

[**PAGE SIZE** *page-size*]

[**COLLATION** *collation-label*]

[**ENCRYPTED** { **ON** | **OFF** | *key-spec* }]

[**BLANK PADDING** { **ON** | **OFF** }]

[**ASE** [**COMPATIBLE**]]

[**JAVA** { **ON** | **OFF** | **JDK** { '1.1.8' | '1.3' } }]

[**JCONNECT** { **ON** | **OFF** }]

]

page-size :

1024 | **2048** | **4096** | **8192** | **16384** | **32768**

collation-label : *string*

key-spec:

[**ON**] **KEY** *key* [**ALGORITHM** { 'AES' | 'MDSR' }]

CREATE DBSPACE statement

CREATE DBSPACE *dbspace-name* **AS** *filename*

CREATE DECRYPTED FILE statement

```
CREATE DECRYPTED FILE newfile
FROM oldfile KEY
key
```

CREATE DOMAIN statement

```
CREATE { DOMAIN | DATATYPE } [ AS ] domain-name data-type
[ [ NOT ] NULL ]
[ DEFAULT default-value ]
[ CHECK ( condition ) ]

domain-name : identifier

data-type : built-in data type, with precision and scale
```

CREATE ENCRYPTED FILE statement

```
CREATE ENCRYPTED FILE newfile
FROM oldfile
KEY key
ALGORITHM algorithm
```

CREATE EVENT statement

```
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 ]

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 ]
{ START TIME start-time | BETWEEN start-time AND end-time }
[ EVERY period { HOURS | MINUTES | SECONDS } ]
[ ON { ( day-of-week, ... ) | ( day-of-month, ... ) } ]
[ START DATE 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
```

CREATE EXISTING TABLE statement

CREATE EXISTING TABLE [owner.]table-name
[(column-definition, ...)]
AT location-string

column-definition :
column-name data-type [NOT NULL]

location-string :
remote-server-name.[db-name].[owner].object-name
| remote-server-name:[db-name];[owner];object-name

CREATE EXTERNLOGIN statement

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

CREATE FUNCTION statement

CREATE FUNCTION [owner.]function-name ([parameter, ...])
RETURNS data-type routine-characteristics
{ compound-statement
| **AS** tsq-compound-statement
| external-name }

parameter :
[**IN**] parameter-name data-type

routine-characteristics
ON EXCEPTION RESUME | [**NOT**] **DETERMINISTIC**

tsq-compound-statement :
sql-statement
sql-statement
...

external-name :
EXTERNAL NAME library-call
| **EXTERNAL NAME** java-call **LANGUAGE JAVA**

library-call :
'[operating-system:]function-name@library; ... '

operating-system :
Windows95 | **WindowsNT** | **NetWare** | **UNIX**

java-call :
'[package-name.]class-name.method-name method-signature'

method-signature :
([field-descriptor, ...]) return-descriptor

field-descriptor | return-descriptor :
Z | **B** | **S** | **I** | **J** | **F** | **D** | **C** | **V** | [descriptor | Lclass-name;

CREATE INDEX statement

CREATE [**VIRTUAL**] [**UNIQUE**] [**CLUSTERED**] **INDEX** index-name
ON [owner.]table-name
(column-name [**ASC** | **DESC**], ...
| function-name (argument [,...]) **AS** column-name)
[{ **IN** | **ON** } dbspace-name]

CREATE MESSAGE statement [T-SQL]

CREATE MESSAGE *message-number* **AS** *message-text*

message-number : *integer*

message-text : *string*

CREATE PROCEDURE statement

```
CREATE PROCEDURE [ owner ].procedure-name ( [ parameter, ... ] )
{ [ RESULT ( result-column, ... ) ]
  [ ON EXCEPTION RESUME ]
    compound-statement
  | AT location-string
  | EXTERNAL NAME library-call
  | [ DYNAMIC RESULT SETS integer-expression ]
  | [ EXTERNAL NAME java-call LANGUAGE JAVA ]
}
```

```
CREATE PROCEDURE [ owner ].procedure-name ( [ parameter, ... ] )
compound-statement
```

parameter :

```
parameter_mode parameter-name data-type [ DEFAULT expression ]
| SQLCODE
| SQLSTATE
```

parameter_mode : **IN** | **OUT** | **INOUT**

result-column : *column-name* *data-type*

library-call :

'*operating-system*:*function-name*@*library*; ... '

operating-system :

Windows95 | **WindowsNT** | **NetWare** | **UNIX**

java-call :

'*package-name*.*class-name*.*method-name* *method-signature*'

method-signature :

([*field-descriptor*, ...]) *return-descriptor*

field-descriptor | *return-descriptor* :

Z | **B** | **S** | **I** | **J** | **F** | **D** | **C** | **V** | [*descriptor* | *Lclass-name*];

CREATE PROCEDURE statement [T-SQL]

```
CREATE PROCEDURE [ owner ].procedure_name
[ [ ( ) @parameter_name data-type [ = default ] [ OUTPUT ], ... [ ] ] ]
[ WITH RECOMPILE ] AS statement-list
```

CREATE PUBLICATION statement

```
CREATE PUBLICATION [ owner ].publication-name
( TABLE article-description, ... )
```

owner, *publication-name* : *identifier*

article-description :

table-name [(*column-name*, ...)]

[**WHERE** *search-condition*]

[**SUBSCRIBE BY** *expression*]

CREATE REMOTE MESSAGE TYPE statement [SQL Remote]

```
CREATE REMOTE MESSAGE TYPE message-system
ADDRESS address

message-system: FILE | FTP | MAPI | SMTP | VIM

address: string
```

CREATE SCHEMA statement

```
CREATE SCHEMA AUTHORIZATION userid
[
    create-table-statement
    | create-view-statement
    | grant-statement
], ...
```

CREATE SERVER statement

```
CREATE SERVER server-name
CLASS ' server-class '
USING ' connection-info '
[ READ ONLY ]

server-class :
    ASAJDBC | ASEJDBC
    | ASAODBC | ASEODBC
    | DB2ODBC | MSSODBC
    | ORAODBC | ODBC

connection-info :
    { machine-name:port-number [/ dbname ] | data-source-name }
```

CREATE SERVICE statement

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

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' }
```

CREATE STATISTICS statement

```
CREATE STATISTICS table-name [ ( column-list ) ]
```

CREATE SUBSCRIPTION statement [SQL Remote]

```
CREATE SUBSCRIPTION
TO publication-name [ ( subscription-value ) ]
FOR subscriber-id

publication-name: identifier

subscription-value, subscriber-id: string
```

subscriber-id: string

CREATE SYNCHRONIZATION SUBSCRIPTION statement [MobiLink]

```
CREATE SYNCHRONIZATION SUBSCRIPTION
TO publication-name
[ FOR ml_username, ... ]
[ TYPE sync-type ]
[ ADDRESS network-parameters ]
[ OPTION option=value, ... ]
```

ml_username: identifier

network-parameters: string

sync-type: http | https | tcpip | ActiveSync

value: string | integer

CREATE SYNCHRONIZATION USER statement [MobiLink]

```
CREATE SYNCHRONIZATION USER ml_username
[ TYPE sync-type ]
[ ADDRESS network-parameters ]
[ OPTION option=value, ... ]
```

ml_username: identifier

sync-type: tcpip | http | https | ActiveSync

network-parameters: string

value: string | integer

CREATE TABLE statement

```
CREATE [ GLOBAL TEMPORARY ] TABLE [ owner. ] table-name
( { column-definition | table-constraint | pctfree }, ... )
[ { IN | ON } dbspace-name ]
[ ON COMMIT { DELETE | PRESERVE } ROWS
  | NOT TRANSACTIONAL ]
[ AT location-string ]
```

column-definition :

column-name data-type [NOT NULL]

[DEFAULT *default-value*] [*column-constraint* ...]

default-value :
 special-value
 | *string*
 | *global variable*
 | [-] *number*
 | (*constant-expression*)
 | *built-in-function*(*constant-expression*)
 | **AUTOINCREMENT**
 | **CURRENT DATABASE**
 | **CURRENT REMOTE USER**
 | **CURRENT UTC TIMESTAMP**
 | **GLOBAL AUTOINCREMENT** [(*partition-size*)]
 | **NULL**
 | **TIMESTAMP**
 | **UTC TIMESTAMP**
 | **LAST USER**

special-value:
 CURRENT { **DATE** | **TIME** | **TIMESTAMP**
 | **UTC TIMESTAMP** | **USER** | **PUBLISHER** }
 | **USER**

column-constraint :
 [**CONSTRAINT** *constraint-name*] {
 UNIQUE
 | **PRIMARY KEY** [**CLUSTERED**]
 | **REFERENCES** *table-name*
 [(*column-name*)] [*actions*] [**CLUSTERED**]
 }
 | [**CONSTRAINT** *constraint-name*] **CHECK** (*condition*)
 | **COMPUTE** (*expression*)

table-constraint :
 [**CONSTRAINT** *constraint-name*] {
 UNIQUE (*column-name*, ...)
 | **PRIMARY KEY** [**CLUSTERED**] (*column-name*, ...)
 | **CHECK** (*condition*)
 | *foreign-key-constraint*
 }
}

foreign-key-constraint :
 [**NOT NULL**] **FOREIGN KEY** [*role-name*] [(*column-name*, ...)]
 REFERENCES *table-name* [(*column-name*, ...)] [**CLUSTERED**]
 [*actions*] [**CHECK ON COMMIT**]

action :
 ON { **UPDATE** | **DELETE** }
 ... { **CASCADE** | **SET NULL** | **SET DEFAULT** | **RESTRICT** }

location-string :
 remote-server-name.*db-name*.*owner*.*object-name*
 | *remote-server-name*;*db-name*;*owner*;*object-name*

pctfree : **PCTFREE** *percent-free-space*

percent-free-space : integer

CREATE TRIGGER statement

```
CREATE TRIGGER trigger-name trigger-time trigger-event [, trigger-event, ... ]  
[ ORDER integer ] ON table-name  
[ REFERENCING [ OLD AS old-name ]  
                  [ NEW AS new-name ] ]  
                  [ REMOTE AS remote-name ] ]  
[ FOR EACH { ROW | STATEMENT } ]  
[ WHEN ( search-condition ) ]  
  compound-statement
```

trigger-time : BEFORE | AFTER | RESOLVE

trigger-event :

DELETE | INSERT | UPDATE | UPDATE OF *column-list*

CREATE TRIGGER statement [SQL Remote]

```
CREATE TRIGGER trigger-name trigger-time
trigger-event, ...
[ ORDER integer ] ON table-name
[ REFERENCING [ OLD AS old-name ]
  [ NEW AS new-name ] ]
[ REMOTE AS remote-name ] ]
[ FOR EACH { ROW | STATEMENT } ]
[ WHEN ( search-condition ) ]
[ IF UPDATE ( column-name ) THEN
  [ { AND | OR } UPDATE ( column-name ) ] ... ]
  compound-statement
[ ELSEIF UPDATE ( column-name ) THEN
  [ { AND | OR } UPDATE ( column-name ) ] ... ]
  compound-statement
END IF ] ]
```

trigger-time:

BEFORE | AFTER | RESOLVE

trigger-event:

DELETE | INSERT | UPDATE

| UPDATE OF *column-name* [, *column-name*, ...]

CREATE TRIGGER statement [T-SQL]

```
CREATE TRIGGER [owner.]trigger_name
ON [owner.]table_name
FOR { INSERT, UPDATE, DELETE }
AS statement-list

CREATE TRIGGER [owner.]trigger_name
ON [owner.]table_name
FOR { INSERT, UPDATE }
AS
[ IF UPDATE ( column_name )
  [ { AND | OR } UPDATE ( column_name ) ] ... ]
  statement-list
[ IF UPDATE ( column_name )
  [ { AND | OR } UPDATE ( column_name ) ] ... ]
  statement-list
```

CREATE VARIABLE statement

CREATE VARIABLE *identifier* *data-type*

CREATE VIEW statement

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

CREATE WRITEFILE statement

```
CREATE WRITEFILE write-file-name
FOR DATABASE db-file-name [ KEY key ]
[ LOG OFF | LOG ON [ log-file-name [ MIRROR mirror-file-name ] ] ]

write-file-name | db-file-name | log-file-name | mirror-file-name : string
```

DEALLOCATE statement

```
DEALLOCATE [ CURSOR ] cursor-name

cursor-name : identifier
```

DEALLOCATE DESCRIPTOR statement [ESQL]

```
DEALLOCATE DESCRIPTOR descriptor-name

descriptor-name : string
```

Declaration section [ESQL]

```
EXEC SQL BEGIN DECLARE SECTION;
C declarations
EXEC SQL END DECLARE SECTION;
```

DECLARE statement

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

DECLARE CURSOR statement [ESQL] [SP]

```
DECLARE cursor-name
[ UNIQUE ]
[ NO SCROLL
  | DYNAMIC SCROLL
  | SCROLL
  | INSENSITIVE
  | SENSITIVE
]
CURSOR FOR
{ select-statement
  | statement-name
    [ FOR { UPDATE [ cursor-concurrency ] | READ ONLY } ]
  | call-statement }

DECLARE cursor-name
[ NO SCROLL
  | DYNAMIC SCROLL
  | SCROLL
  | INSENSITIVE
  | SENSITIVE
]
CURSOR FOR
{ select-statement
  | FOR { UPDATE [ cursor-concurrency ] | READ ONLY } ]
  | call-statement
  | USING variable-name }

cursor-name : identifier

statement-name : identifier | hostvar
```


variable-name : *identifier*

cursor-concurrency :
BY { **VALUES** | **TIMESTAMP** | **LOCK** }

DECLARE CURSOR statement [T-SQL]

DECLARE *cursor-name*
CURSOR FOR *select-statement*
 [**FOR** { **READ ONLY** | **UPDATE** }]

cursor-name : *identifier*

select-statement : *string*

DECLARE LOCAL TEMPORARY TABLE statement

DECLARE LOCAL TEMPORARY TABLE *table-name*
 ({ *column-definition* [*column-constraint* ...] | *table-constraint* | *pctfree*
 }, ...)
 [**ON COMMIT** { **DELETE** | **PRESERVE** } **ROWS**
 | **NOT TRANSACTIONAL**]

pctfree : **PCTFREE** *percent-free-space*

percent-free-space : *integer*

DELETE statement

DELETE [**FIRST** | **TOP** *n*]
 [**FROM**] [*owner*.] *table-name*
 [**FROM** *table-list*]
 [**WHERE** *search-condition*]

DELETE (positioned) statement [ESQL] [SP]

DELETE [**FROM** *table-spec*] **WHERE CURRENT OF** *cursor-name*

cursor-name : *identifier* | *hostvar*

table-spec : [*owner*.] *correlation-name*

owner : *identifier*

DESCRIBE statement [ESQL]

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*

long-name-spec :

OWNER.TABLE.COLUMN | **TABLE.COLUMN** | **COLUMN**

statement-name : *identifier* | *hostvar*

cursor-name : *declared cursor*

sqlda-name : *identifier*

DISCONNECT statement [ESQL] [Interactive SQL]

DISCONNECT [*connection-name* | **CURRENT** | **ALL**]

connection-name : *identifier*, *string*, or *hostvar*

DROP statement

DROP
| { **DATATYPE** | **DOMAIN** } *datatype-name*
| **DBSPACE** *dbspace-name*
| **EVENT** *event-name*
| **FUNCTION** [*owner.*] *function-name*
| **INDEX** [[*owner.*] *table-name.*] *index-name*
| **MESSAGE** *msgnum*
| **PROCEDURE** [*owner.*] *procedure-name*
| **TABLE** [*owner.*] *table-name*
| **TRIGGER** [[*owner.*] *table-name.*] *trigger-name*
| **VIEW** [*owner.*] *view-name*

DROP DATABASE statement

DROP DATABASE *database-name* [**KEY** *key*]

DROP CONNECTION statement

DROP CONNECTION *connection-id*

DROP EXTERNLOGIN statement

DROP EXTERNLOGIN *login-name* **TO** *remote-server*

DROP PUBLICATION statement

DROP PUBLICATION [*owner.*] *publication-name*

owner, *publication-name* : *identifier*

DROP REMOTE MESSAGE TYPE statement [SQL Remote]

DROP REMOTE MESSAGE TYPE *message-system*

message-system: **FILE** | **FTP** | **MAPI** | **SMTP** | **VIM**

DROP SERVER statement

DROP SERVER *server-name*

DROP SERVICE statement

DROP SERVICE *service-name*

DROP STATEMENT statement [ESQL]

DROP STATEMENT [*owner.*] *statement-name*

statement-name : *identifier* | *hostvar*

DROP STATISTICS statement

DROP STATISTICS [**ON**] [*owner*.] *table-name* [(*column-list*)]

DROP OPTIMIZER STATISTICS

DROP SUBSCRIPTION statement [SQL Remote]

DROP SUBSCRIPTION TO *publication-name* [(*subscription-value*)]
FOR *subscriber-id*, ...

subscription-value: *string*

subscriber-id: *string*

DROP SYNCHRONIZATION SUBSCRIPTION statement [MobiLink]

DROP SYNCHRONIZATION SUBSCRIPTION
TO *publication-name*
[**FOR** *ml_username*, ...]

DROP SYNCHRONIZATION USER statement [MobiLink]

DROP SYNCHRONIZATION USER *ml_username*, ...

ml_username: *identifier*

DROP VARIABLE statement

DROP VARIABLE *identifier*

EXCEPT operation

select-statement
EXCEPT [**ALL** | **DISTINCT**] *select-statement*
[**EXCEPT** [**ALL** | **DISTINCT**] *select-statement*] ...
[**ORDER BY** *integer* [**ASC** | **DESC**], ...]

EXECUTE statement [ESQL]

EXECUTE *statement*
[**USING** { *hostvar-list* | **DESCRIPTOR** *sqllda-name* }]
[**INTO** { *into-hostvar-list* | **DESCRIPTOR** *into-sqllda-name* }]
[**ARRAY** :*integer*]

statement : { *identifier* | *hostvar* | *string* }

sqllda-name : *identifier*

into-sqllda-name : *identifier*

EXECUTE IMMEDIATE *statement*

statement : { *string* | *hostvar* }

EXECUTE statement [T-SQL]

EXECUTE [@return_status =] [creator.]procedure_name [argument, ...]

argument :

[@parameter-name =] expression

| [@parameter-name =] @variable [output]

EXECUTE (string-expression)

EXECUTE IMMEDIATE statement [SP]

EXECUTE IMMEDIATE [execute-option] string-expression

execute-option:

WITH QUOTES [ON | OFF]

| **WITH ESCAPES** { ON | OFF }

EXECUTE (string-expression)

EXIT statement [Interactive SQL]

{ **EXIT** | **QUIT** | **BYE** } [return-code]

return-code: number

| hostvar

EXPLAIN statement [ESQL]

EXPLAIN PLAN FOR CURSOR cursor-name

{ **INTO** hostvar | **USING DESCRIPTOR** sqlda-name }

cursor-name : identifier or hostvar

sqlda-name : identifier

FETCH statement [ESQL] [SP]

FETCH cursor-position cursor-name

[**INTO** { hostvar-list | variable-list }]

| **USING DESCRIPTOR** sqlda-name]

[**PURGE**]

[**BLOCK** n]

[**FOR UPDATE**]

[**ARRAY** fetch-count]

INTO variable-list [**FOR UPDATE**]

cursor-position :

NEXT | **PRIOR** | **FIRST** | **LAST**

| { **ABSOLUTE** | **RELATIVE** } row-count

row-count : number or hostvar

cursor-name : identifier or hostvar

hostvar-list : may contain indicator variables

variable-list : stored procedure variables

sqlda-name : identifier

fetch-count : integer or hostvar

FOR statement

```
[ statement-label : ]  
FOR for-loop-name AS cursor-name  
CURSOR FOR statement  
[ FOR UPDATE | FOR READ ONLY ]  
DO statement-list  
END FOR [ statement-label ]
```

FORWARD TO statement

```
FORWARD TO server-name SQL-statement  
  
FORWARD TO [ server-name ]
```

FROM clause

```
FROM table-expression, ...
```

table-expression:

```
table  
| view  
| procedure  
| derived-table  
| lateral-derived-table  
| joined table  
| ( table-expression, ... )
```

table or view :

```
[userid.] table-or-view-name [ [ AS ] correlation-name ] [ WITH ( table-  
hint ) ]
```

procedure :

```
[ owner.] procedure-name ( [ parameter, ... ] )  
[ WITH( column-name data-type, ... ) ]  
[ [ AS ] correlation-name ]
```

derived-table :

```
( select-statement )  
[ AS ] correlation-name [ ( column-name, ... ) ]
```

lateral-derived-table :

```
LATERAL ( select-statement | table-expression )  
[ AS ] correlation-name [ ( column-name, ... ) ]
```

joined table:

```
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 ]
```

table-hint:
NOLOCK
| READUNCOMMITTED
| READCOMMITTED
| REPEATABLEREAD
| HOLDLOCK
| SERIALIZABLE
| FASTFIRSTROW

GET DATA statement [ESQL]

GET DATA *cursor-name*
COLUMN *column-num*
OFFSET *start-offset*
[WITH TEXTPTR]
USING DESCRIPTOR *sqlda-name* **| INTO** *hostvar* [, ...]

cursor-name : *identifier*, or *hostvar*

column-num : *integer* or *hostvar*

start-offset : *integer* or *hostvar*

sqlda-name : *identifier*

GET DESCRIPTOR statement [ESQL]

GET DESCRIPTOR *descriptor-name*
{ *hostvar* = **COUNT** | **VALUE** { *integer* | *hostvar* } *assignment* [, ...] }
assignment :
hostvar = **TYPE** | **LENGTH** | **PRECISION** | **SCALE** | **DATA** | **INDICA-**
TOR | **NAME** |
NULLABLE | **RETURNED_LENGTH**

GET OPTION statement [ESQL]

GET OPTION [*userid*.] *option-name*
[INTO *hostvar* **]**
[USING DESCRIPTOR *sqlda-name* **]**
userid : *identifier*, *string*, or *hostvar*
option-name : *identifier*, *string*, or *hostvar*
hostvar : indicator variable allowed
sqlda-name : *identifier*

GOTO statement [T-SQL]

label : **GOTO** *label*

GRANT statement

```
GRANT CONNECT TO userid, ...
[ AT starting-id ]
IDENTIFIED BY password, ...

GRANT {
  DBA,
  GROUP,
  MEMBERSHIP IN GROUP userid, ...,
  [ RESOURCE | ALL ]
}
  TO userid, ...

GRANT {
  ALL [ PRIVILEGES ],
  ALTER,
  DELETE,
  INSERT,
  REFERENCES [ ( column-name, ... ) ],
  SELECT [ ( column-name, ... ) ],
  UPDATE [ ( column-name, ... ) ],
}
  ON [ owner.]table-name
  TO userid, ...
  [ WITH GRANT OPTION ]
  [ FROM userid ]

GRANT EXECUTE ON [ owner.]procedure-name TO userid, ...

GRANT INTEGRATED LOGIN TO user_profile_name, ... AS USER userid
```

GRANT CONSOLIDATE statement [SQL Remote]

```
GRANT CONSOLIDATE
TO userid
TYPE message-system, ...
ADDRESS address-string, ...
[ SEND { EVERY | AT } 'hh:mm:ss' ]

message-system: FILE | FTP | MAPI | SMTP | VIM

address: string
```

GRANT PUBLISH statement [SQL Remote]

```
GRANT PUBLISH TO userid
```

GRANT REMOTE statement [SQL Remote]

```
GRANT REMOTE TO userid, ...
TYPE message-system, ...
ADDRESS address-string, ...
[ SEND { EVERY | AT } send-time ]
```

GRANT REMOTE DBA statement [SQL Remote]

```
GRANT REMOTE DBA
TO userid, ...
IDENTIFIED BY password
```

HELP statement [Interactive SQL]

HELP [*topic*]

IF statement

IF *search-condition* **THEN** *statement-list*
[**ELSEIF** { *search-condition* | *operation-type* } **THEN** *statement-list*] ...
[**ELSE** *statement-list*]
END IF

IF statement [T-SQL]

IF *expression*
statement
[**ELSE**
[**IF** *expression*]
statement]

INCLUDE statement [ESQL]

INCLUDE *filename*

filename : **SQLDA** | **SQLCA** | *string*

INPUT statement [Interactive SQL]

INPUT INTO [*owner.*]*table-name*
[**FROM** *filename* | **PROMPT**]
[**FORMAT** *input-format*]
[**ESCAPE CHARACTER** *character*]
[**BY ORDER** | **BY NAME**]
[**DELIMITED BY** *string*]
[**COLUMN WIDTHS** (*integer*, ...)]
[**NOSTRIP**]
[(*column-name*, ...)]

input-format :
ASCII | **DBASE** | **DBASEII** | **DBASEIII**
| **EXCEL** | **FIXED** | **FOXPRO** | **LOTUS**

INSERT statement

INSERT [**INTO**] [*owner.*]*table-name* [(*column-name*, ...)]
[**ON EXISTING** { **ERROR** | **SKIP** | **UPDATE** }] **VALUES**
(*expression* | **DEFAULT**, ...)

INSERT [**INTO**] [*owner.*]*table-name*
[**ON EXISTING** { **ERROR** | **SKIP** | **UPDATE** }]
[**WITH AUTO NAME**]
select-statement

INSTALL JAVA statement

INSTALL JAVA
[**NEW** | **UPDATE**]
[**JAR** *jar-name*]
FROM { **FILE** *filename* | *expression* }

INTERSECT operation

```
select-statement
INTERSECT { ALL | DISTINCT } select-statement
[ INTERSECT { ALL | DISTINCT } select-statement ] ...
[ ORDER BY integer [ ASC | DESC ], ... ]
```

LEAVE statement

```
LEAVE statement-label
```

LOAD STATISTICS statement

```
LOAD STATISTICS [ [ owner. ] table-name. ] column-name
format-id, density, max-steps, actual-steps, step-values, frequencies
```

LOAD TABLE statement

```
LOAD [ INTO ] TABLE [ owner. ] table-name [ ( column-name, ... ) ]
FROM filename-string
[ load-option ... ]
```

```
load-option :
CHECK CONSTRAINTS { ON | OFF }
| COMPUTES { ON | OFF }
| DEFAULTS { ON | OFF }
| DELIMITED BY string
| ESCAPE CHARACTER character
| ESCAPES { ON | OFF }
| FORMAT { ASCII | BCP }
| HEXADECEIMAL { ON | OFF }
| ORDER { ON | OFF }
| PCTFREE percent-free-space
| QUOTES { ON | OFF }
| STRIP { ON | OFF }
| WITH CHECKPOINT { ON | OFF }
```

LOCK TABLE statement

```
LOCK TABLE table-name
[ WITH HOLD ]
IN { SHARE | EXCLUSIVE } MODE
```

LOOP statement

```
[ statement-label : ]
[ WHILE search-condition ] LOOP
    statement-list
END LOOP [ statement-label ]
```

MESSAGE statement

```
MESSAGE expression, ...
[ TYPE { INFO | ACTION | WARNING | STATUS } ]
[ TO { CONSOLE | CLIENT | LOG } ]
```

OPEN statement [ESQL] [SP]

```
OPEN cursor-name  
[ USING [ DESCRIPTOR sqlda-name | hostvar, ... ] ]  
[ WITH HOLD ]  
[ ISOLATION LEVEL n ]  
[ BLOCK n ]
```

cursor-name : *identifier* or *hostvar*

sqlda-name : *identifier*

OUTPUT statement [Interactive SQL]

```
OUTPUT TO filename  
[ APPEND ]  
[ VERBOSE ]  
[ FORMAT output-format ]  
[ ESCAPE CHARACTER character ]  
[ DELIMITED BY string ]  
[ QUOTE string [ ALL ] ]  
[ COLUMN WIDTHS (integer, ... ) ]  
[ HEXADECIMAL { ON | OFF | ASIS } ]
```

output-format :

ASCII | DBASEII | DBASEIII | EXCEL
| FIXED | FOXPRO | HTML | LOTUS | SQL | XML

PARAMETERS statement [Interactive SQL]

```
PARAMETERS parameter1, parameter2, ...
```

PASSTHROUGH statement [SQL Remote]

```
PASSTHROUGH [ ONLY ] FOR userid, ...
```

```
PASSTHROUGH [ ONLY ] FOR SUBSCRIPTION  
TO [ ( owner ) ].publication-name [ ( constant ) ]
```

```
PASSTHROUGH STOP
```

PREPARE statement [ESQL]

```
PREPARE statement-name  
FROM statement  
[ DESCRIBE describe-type INTO [ [ SQL ] DESCRIPTOR ] descriptor ]  
[ WITH EXECUTE ]
```

statement-name : *identifier* or *hostvar*

statement : *string* or *hostvar*

describe-type :

```
[ ALL | BIND VARIABLES | INPUT | OUTPUT | SELECT LIST ]  
[ LONG NAMES [ [ [ OWNER. ]TABLE. ]COLUMN ]  
| WITH VARIABLE RESULT ]
```

PREPARE TO COMMIT statement

```
PREPARE TO COMMIT
```

PRINT statement [T-SQL]

PRINT *format-string* [, *arg-list*]

PUT statement [ESQL]

PUT *cursor-name*
[**USING DESCRIPTOR** *sqlda-name* | **FROM** *hostvar-list*]
[**INTO** { **DESCRIPTOR** *into-sqlda-name* | *into-hostvar-list* }]
[**ARRAY** :*nnn*]

cursor-name : *identifier* or *hostvar*

sqlda-name : *identifier*

hostvar-list : may contain indicator variables

RAISERROR statement [T-SQL]

RAISERROR *error-number* [*format-string*] [, *arg-list*]

READ statement [Interactive SQL]

READ *filename* [*parameters*]

READTEXT statement [T-SQL]

READTEXT *table-name.column-name*
text-pointer *offset* *size*
[**HOLDLOCK**]

RELEASE SAVEPOINT statement

RELEASE SAVEPOINT [*savepoint-name*]

REMOTE RESET statement [SQL Remote]

REMOTE RESET *userid*

REMOVE JAVA statement

REMOVE JAVA *classes_to_remove*

classes_to_remove :
 CLASS *java_class_name* [, *java_class_name*, ...]
 JAR *jar_name* [, *jar_name*, ...]

REORGANIZE TABLE statement

REORGANIZE TABLE [*owner.*]*table-name*
[{ **PRIMARY KEY**
 | **FOREIGN KEY** *foreign_key_name*
 | **INDEX** *index_name* }
 | **ORDER** {**ON** | **OFF** }
]

RESIGNAL statement

RESIGNAL [*exception-name*]

RESTORE DATABASE statement

RESTORE DATABASE *filename*
FROM *archive_root*
[**CATALOG ONLY** |
[[**RENAME** *dbspace_name* **TO** *new_dbspace_name*] ...]]

RESUME statement

RESUME *cursor-name*

cursor-name : *identifier* or *hostvar*

RETURN statement

RETURN [*expression*]

REVOKE statement

REVOKE *special-priv*, ... **FROM** *userid*, ...

special-priv :

CONNECT
| **DBA**
| **INTEGRATED LOGIN**
| **GROUP**
| **MEMBERSHIP IN GROUP** *userid*, ...
| **RESOURCE**

REVOKE *table-priv*, ... **ON** [*owner.*] *table-name* **FROM** *userid*, ...

table-priv :

ALL [PRIVILEGES]
| **ALTER**
| **DELETE**
| **INSERT**
| **REFERENCES** [(*column-name*, ...)]
| **SELECT** [(*column-name*, ...)]
| **UPDATE** [(*column-name*, ...)]

REVOKE EXECUTE ON [*owner.*] *procedure-name* **FROM** *userid*, ...

REVOKE CONSOLIDATE statement [SQL Remote]

REVOKE CONSOLIDATE FROM *userid*

REVOKE PUBLISH statement [SQL Remote]

REVOKE PUBLISH FROM *userid*

REVOKE REMOTE statement [SQL Remote]

REVOKE REMOTE FROM *userid*, ...

REVOKE REMOTE DBA statement [SQL Remote]

```
REVOKE REMOTE DBA
FROM userid, ...
```

ROLLBACK statement

```
ROLLBACK [ WORK ]
```

ROLLBACK TO SAVEPOINT statement

```
ROLLBACK TO SAVEPOINT [savepoint-name]
```

ROLLBACK TRIGGER statement

```
ROLLBACK TRIGGER [ WITH raiserror-statement ]
```

SAVEPOINT statement

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

SELECT statement

```
[ WITH temporary-views ]
SELECT [ ALL | DISTINCT ] [ row-limitation ] select-list
[ INTO { hostvar-list | variable-list } ]
[ FROM table-expression ]
[ WHERE search-condition ]
[ GROUP BY [ group-by-list | ROLLUP ] (group-by-list) ]
[ HAVING search-condition ]
[ ORDER BY { expression | integer } [ ASC | DESC ], ... ]
[ FOR { UPDATE [ cursor-concurrency ] | READ ONLY } ]
[ FOR XML xml-mode ]
```

temporary-views :

regular-view, ...

| RECURSIVE { *regular-view* | *recursive-view* }, ...

regular-view :

view-name [(*column-name*, ...)]

AS (*subquery*)

recursive-view :

view-name (*column-name*, ...)

AS (*initial-subquery* UNION ALL *recursive-subquery*)

row-limitation :

FIRST | TOP *n* [START AT *m*]

select-list :

{ *column-name* | *expression* } [[AS] *alias-name*], ...

| *

group-by-list :

{ *column-name* | *alias-name* | *function* | *expression* }, ...

cursor-concurrency :

BY { VALUES | TIMESTAMP | LOCK }

xml-mode :

RAW [, ELEMENTS] | AUTO [, ELEMENTS] | EXPLICIT

SET statement

SET *identifier* = *expression*

SET statement [T-SQL]

SET *option-name* *option-value*

SET CONNECTION statement [Interactive SQL] [ESQL]

SET CONNECTION [*connection-name*]

connection-name :
identifier, *string*, or *hostvar*

SET DESCRIPTOR statement [ESQL]

SET DESCRIPTOR *descriptor-name*
{ **COUNT** = { *integer* | *hostvar* }
| **VALUE** { *integer* | *hostvar* } *assignment* [, ...] }

assignment :
{ **TYPE** | **SCALE** | **PRECISION** | **LENGTH** | **INDICATOR** }
= { *integer* | *hostvar* }
| **DATA** = *hostvar*

SET OPTION statement

SET [**EXISTING**] [**TEMPORARY**] **OPTION**
[*userid.* | **PUBLIC.**] *option-name* = [*option-value*]

userid : *identifier* | *string* | *hostvar*

option-name : *identifier* | *string* | *hostvar*

option-value : *hostvar* (indicator allowed)
| *string*
| *identifier*
| *number*

SET OPTION statement [Interactive SQL]

SET [**TEMPORARY**] **OPTION**
[*userid.* | **PUBLIC.**] *option-name* = [*option-value*]

userid : *identifier*, *string* or *hostvar*

option-name : *identifier*, *string* or *hostvar*

option-value : *hostvar* (indicator allowed), *string*, *identifier*, or *number*

SET PERMANENT

SET

SET REMOTE OPTION statement [SQL Remote]

```

SET REMOTE link-name OPTION
[ userid.| PUBLIC.]link-option-name = link-option-value

link-name:
file | ftp | mapi | smtp | vim

link-option-name:
common-option | file-option | ftp-option
| mapi-option | smtp-option | vim-option

common-option:
debug | output_log_send_on_error
| output_log_send_limit | output_log_send_now

file-option:
directory | unlink_delay

ftp-option:
active_mode | host | password | port | root_directory | user

mapi-option:
force_download | ipm_receive | ipm_send | profile

smtp-option:
local_host | pop3_host | pop3_password | pop3_userid
| smtp_host | top_supported

vim-option:
password | path | receive_all | send_vim_mail | userid

link-option-value:
string

```

SET SQLCA statement [ESQL]

```

SET SQLCA sqlca

sqlca : identifier or string

```

SETUSER statement

```

{ SET SESSION AUTHORIZATION | SETUSER }
[ [ WITH OPTIONS ] userid ]

```

SIGNAL statement

```

SIGNAL exception-name

```

START DATABASE statement

```

START DATABASE database-file
[ AS database-name ]
[ ON engine-name ]
[ WITH TRUNCATE AT CHECKPOINT ]
[ FOR READ ONLY ]
[ AUTOSTOP { ON | OFF } ]
[ KEY key ]

```

START ENGINE statement [Interactive SQL]

```

START ENGINE AS engine-name [ STARTLINE command-string ]

```

START JAVA statement

START JAVA

START LOGGING statement [Interactive SQL]

START LOGGING *filename*

START SUBSCRIPTION statement [SQL Remote]

START SUBSCRIPTION
TO *publication-name* [(*subscription-value*)]
FOR *subscriber-id*, ...

START SYNCHRONIZATION DELETE statement [MobiLink]

START SYNCHRONIZATION DELETE

STOP DATABASE statement

STOP DATABASE *database-name*
[ON *engine-name*]
[UNCONDITIONALLY]

STOP ENGINE statement

STOP ENGINE [*engine-name*] [UNCONDITIONALLY]

STOP JAVA statement

STOP JAVA

STOP LOGGING statement [Interactive SQL]

STOP LOGGING

STOP SUBSCRIPTION statement [SQL Remote]

STOP SUBSCRIPTION
TO *publication-name* [(*subscription-value*)]
FOR *subscriber-id*, ...

STOP SYNCHRONIZATION DELETE statement [MobiLink]

STOP SYNCHRONIZATION DELETE

SYNCHRONIZE SUBSCRIPTION statement [SQL Remote]

SYNCHRONIZE SUBSCRIPTION
TO *publication-name* [(*subscription-value*)]
FOR *remote-user*, ...

SYSTEM statement [Interactive SQL]

SYSTEM ' [path] filename '

TRIGGER EVENT statement

TRIGGER EVENT *event-name* [(*parm* = *value*, ...)]

TRUNCATE TABLE statement

TRUNCATE TABLE [*owner.*] *table-name*

UNION operation

select-statement
UNION [**ALL** | **DISTINCT**] *select-statement*
[**UNION** [**ALL** | **DISTINCT**] *select-statement*] ...
[**ORDER BY** *integer* [**ASC** | **DESC**], ...]

UNLOAD statement

UNLOAD *select-statement* **TO** *filename-string* [*unload-option* ...]

unload-option :
| **APPEND** { **ON** | **OFF** }
| **DELIMITED BY** *string*
| **ESCAPE CHARACTER** *character*
| **ESCAPES** { **ON** | **OFF** }
| **FORMAT** { **ASCII** | **BCP** }
| **HEXADECIMAL** { **ON** | **OFF** }
| **QUOTES** { **ON** | **OFF** }

UNLOAD TABLE statement

UNLOAD [**FROM**] **TABLE** [*owner.*] *table-name* **TO** *filename-string*
[*unload-option* ...]

unload-option :
| **APPEND** { **ON** | **OFF** }
| **DELIMITED BY** *string*
| **ESCAPE CHARACTER** *character*
| **ESCAPES** { **ON** | **OFF** }
| **FORMAT** { **ASCII** | **BCP** }
| **HEXADECIMAL** { **ON** | **OFF** }
| **ORDER** { **ON** | **OFF** }
| **QUOTES** { **ON** | **OFF** }

UPDATE statement

```
UPDATE [ FIRST | TOP n ] table-list SET set-item, ...
[ FROM table-list ]
[ WHERE search-condition ]
[ ORDER BY expression [ ASC | DESC ], ... ]

UPDATE table-list
SET set-item, ...
[ VERIFY ( column-name, ... ) VALUES ( expression, ... ) ]
[ WHERE search-condition ]
[ ORDER BY expression [ ASC | DESC ], ... ]

UPDATE table
PUBLICATION publication
{ SUBSCRIBE BY expression
  OLD SUBSCRIBE BY expression NEW SUBSCRIBE BY expression
}
WHERE search-condition

set-item :
  column-name [.field-name...] = expression
| column-name [.field-name...].method-name( [ expression ] )
| @variable-name = expression
```

UPDATE (positioned) statement [ESQL] [SP]

```
UPDATE WHERE CURRENT OF cursor-name
{ USING DESCRIPTOR sqllda-name | FROM hostvar-list }

UPDATE table-list
SET set-item, ...
WHERE CURRENT OF cursor-name

hostvar-list : indicator variables allowed

set-item :
  column-name [.field-name...] = expression
| column-name [.field-name...].method-name( [ expression ] )

sqllda-name : identifier
```

UPDATE statement [SQL Remote]

```
UPDATE table-list
SET column-name = expression, ...
[ VERIFY ( column-name, ... ) VALUES ( expression, ... ) ]
[ WHERE search-condition ]
[ ORDER BY expression [ ASC | DESC ], ... ]

UPDATE table
PUBLICATION publication
{ SUBSCRIBE BY expression |
  OLD SUBSCRIBE BY expression
  NEW SUBSCRIBE BY expression }
WHERE search-condition

expression: value | subquery
```

VALIDATE INDEX statement

```
VALIDATE INDEX [ [ owner.]table-name. ] { index-name | table-name }
```

VALIDATE TABLE statement

```
VALIDATE TABLE [ owner. ] table-name  
[ WITH { DATA | EXPRESS | FULL | INDEX } CHECK ]
```

WAITFOR statement

```
WAITFOR { DELAY time | TIME time }
```

time: string

WHENEVER statement [ESQL]

```
WHENEVER { SQLERROR | SQLWARNING | NOTFOUND }  
GOTO label | STOP | CONTINUE | { C-code; }
```

label : identifier

WHILE statement [T-SQL]

```
WHILE search-condition-statement
```

WRITETEXT statement [T-SQL]

```
WRITETEXT table-name.column-name  
text_pointer [ WITH LOG ] data
```