

CA Datacom®/DB

SQL Quick Reference Guide

Version 14.02



This Documentation, which includes embedded help systems and electronically distributed materials (hereinafter referred to as the "Documentation"), is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2015 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA products:

- CA Datacom/DB
- CA Datacom® SQL (SQL)
- CA Dataquery™ for CA Datacom® (CA Dataquery)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: Introduction 9

Reading Syntax Diagrams	9
Statement Without Parameters	10
Statement with Required Parameters	10
Delimiters Around Parameters	10
Choice of Required Parameters	10
Default Value for a Required Parameter	11
Optional Parameter	11
Choice of Optional Parameters	11
Repeatable Variable Parameter	12
Separator with Repeatable Variable and Delimiter	12
Optional Repeatable Parameters	13
Default Value for a Parameter	13
Variables Representing Several Parameters	14
CA Datacom/DB Extensions	14

Chapter 2: Expressions 17

Expression	17
CASE	17
COALESCE Expression	18
NULLIF Expression	18
CAST WITHOUT CONVERSION	18
CAST	18
Special Registers	18
Labeled Duration	19
Expressions without Arithmetic Operators	19
Expressions with the Concatenation Operator	19
Expressions with Arithmetic Operators	20

Chapter 3: Functions 21

Functions	21
Column Functions	21
Scalar Functions	21

Chapter 4: Predicates 25

Predicates	25
Basic Predicate	25
Quantified Predicate	25
BETWEEN Predicate	26
LIKE Predicate.....	26
EXISTS Predicate.....	26
IN Predicate.....	26
NULL Predicate	26

Chapter 5: Search Conditions 27

Chapter 6: SQL Statements 29

SQL Statements	29
ALTER TABLE Statement.....	29
Assignment Statement.....	29
CALL/EXECUTE PROCEDURE Statement	29
CASE Statement	30
CLOSE Statement	30
COMMENT ON Statement.....	30
COMMIT WORK Statement	30
CREATE INDEX Statement	31
CREATE PROCEDURE Statement	31
CREATE RULE Statement	38
CREATE SCHEMA Statement	38
CREATE SYNONYM Statement.....	38
CREATE TABLE	39
CREATE TRIGGER/RULE	40
CREATE VIEW	41
DECLARE CURSOR.....	41
DECLARE STATEMENT	42
DELETE.....	42
DESCRIBE	42
DROP	42
EXECUTE	43
EXECUTE IMMEDIATE.....	43
FETCH	43
GRANT (Plan Security)	43
GRANT (Non-Plan Security)	44
IF-THEN Statement.....	44

INSERT	44
ITERATE Statement	45
LEAVE Statement.....	45
LOCK TABLE	45
LOOP Statement.....	45
OPEN	45
PREPARE	45
REPEAT-UNTIL Statement	46
REVOKE (Plan Security)	46
REVOKE (Non-Plan Security)	46
ROLLBACK WORK	46
SELECT	46
SET CURRENT SQLID	49
UPDATE	49
WHENEVER.....	49
WHILE Statement	50

Chapter 7: SQL Codes 51

Overview	51
SQL Return Codes 0 through 243	51
SQL Return Codes -1 through -999.....	52

Chapter 1: Introduction

This guide provides a quick reference for experienced users of CA Datacom SQL. This guide is intended for those who write application programs with embedded SQL statements and create (and maintain) personal SQL tables.

For complete information about using SQL statements, is not provided in this quick reference guide. See the *CA Datacom/DB SQL User Guide*.

Reading Syntax Diagrams

Syntax diagrams are used to illustrate the format of statements and some basic language elements. Read syntax diagrams from left to right and top to bottom.

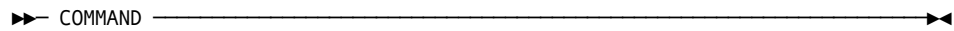
The following terminology, symbols, and concepts are used in syntax diagrams:

- Keywords appear in uppercase letters, for example, COMMAND or PARM. These words must be entered exactly as shown.
- Variables appear in italicized lowercase letters, for example, *variable*.
- Required keywords and variables appear on a main line.
- Optional keywords and variables appear below a main line.
- Default keywords and variables appear above a main line.
- Double arrowheads pointing to the right indicate the beginning of a statement.
- Double arrowheads pointing to each other indicate the end of a statement.
- Single arrowheads pointing to the right indicate a portion of a statement, or that the statement continues in another diagram.
- Punctuation marks or arithmetic symbols that are shown with a keyword or variable must be entered as part of the statement or command. Punctuation marks and arithmetic symbols can include the following:

,	comma	>	greater than symbol
.	period	<=	less than symbol
(	open parenthesis	=	equal sign
)	close parenthesis	≠	not sign
+	addition	-	subtraction
*	multiplication	/	division

Statement Without Parameters

The following is a diagram of a statement without parameters:



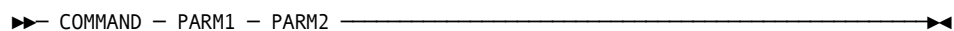
For this statement, you must write the following:

COMMAND

Statement with Required Parameters

Required parameters appear on the same horizontal line, the main path of the diagram, as the command or statement. The parameters must be separated by one or more blanks.

The following is a diagram of a statement with required parameters:



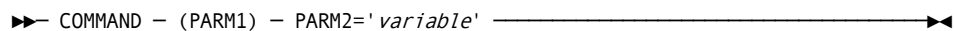
You must write the following:

COMMAND PARAM1 PARAM2

Delimiters Around Parameters

Delimiters, such as parentheses, around parameters or clauses must be included.

The following is a diagram of a statement with delimiters around parameters:



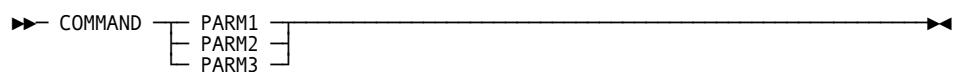
If the word *variable* is a valid entry, you must write the following:

COMMAND (PARAM1) PARAM2='variable'

Choice of Required Parameters

When you see a vertical list of parameters as shown in the following example, you must choose one of the parameters. This indicates that one entry is required, and only one of the displayed parameters is allowed in the statement.

The following is a diagram of a statement with a choice of required parameters:



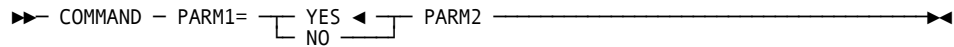
You can choose one of the parameters from the vertical list, such as in the following examples:

```
COMMAND PARM1
COMMAND PARM2
COMMAND PARM3
```

Default Value for a Required Parameter

When a required parameter in a syntax diagram has a default value, the default value appears above the main line, and it indicates the value for the parameter if the command is not specified. If you specify the command, you must code the parameter and specify one of the displayed values.

The following is a diagram of a statement with a default value for a required parameter:



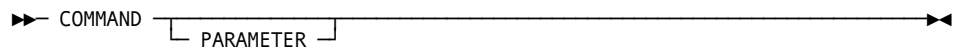
If you specify the command, you must write one of the following:

```
COMMAND PARM1=NO PARM2
COMMAND PARM1=YES PARM2
```

Optional Parameter

A single optional parameter appears below the horizontal line that marks the main path.

The following is a diagram of a statement with an optional parameter:



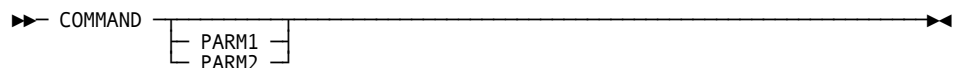
You can choose (or not) to use the optional parameter, as shown in the following examples:

```
COMMAND
COMMAND PARAMETER
```

Choice of Optional Parameters

If you have a choice of more than one optional parameter, the parameters appear in a vertical list below the main path.

The following is a diagram of a statement with a choice of optional parameters:



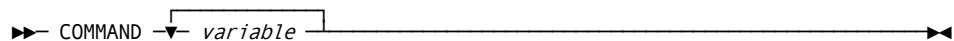
You can choose any of the parameters from the vertical list, or you can write the statement without an optional parameter, such as in the following examples:

```
COMMAND
COMMAND PARM1
COMMAND PARM2
```

Repeatable Variable Parameter

In some statements, you can specify a single parameter more than once. A repeat symbol indicates that you can specify multiple parameters.

The following is a diagram of a statement with a repeatable variable parameter:



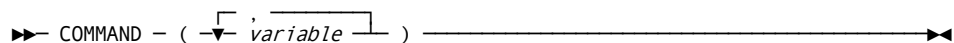
In the preceding diagram, the word *variable* is in lowercase italics, indicating that it is a value you supply, but it is also on the main path, which means that you are required to specify at least one entry. The repeat symbol indicates that you can specify a parameter more than once. Assume that you have three values named VALUEX, VALUEY, and VALUEZ for the variable. The following are some of the statements you might write:

```
COMMAND VALUEX
COMMAND VALUEX VALUEY
COMMAND VALUEX VALUEX VALUEZ
```

Separator with Repeatable Variable and Delimiter

If the repeat symbol contains punctuation such as a comma, you must separate multiple parameters with the punctuation. The following diagram includes the repeat symbol, a comma, and parentheses:

The following is a diagram of a statement with a separator with a repeatable variable and a delimiter:



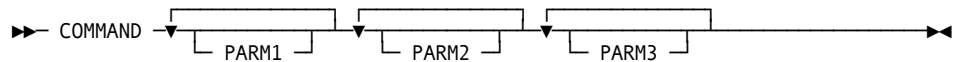
In the preceding diagram, the word *variable* is in lowercase italics, indicating that it is a value you supply. It is also on the main path, which means that you must specify at least one entry. The repeat symbol indicates that you can specify more than one variable and that you must separate the entries with commas. The parentheses indicate that the group of entries must be enclosed within parentheses. Assume that you have three values named VALUEA, VALUEB, and VALUEC for the variable.

The following are some of the statements you can write:

```
COMMAND (VALUEC)
COMMAND (VALUEB, VALUEC)
COMMAND (VALUEB, VALUEA)
COMMAND (VALUEA, VALUEB, VALUEC)
```

Optional Repeatable Parameters

The following diagram shows a list of parameters with the repeat symbol for optional repeatable parameters:



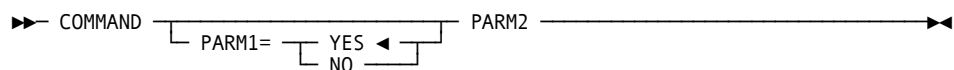
The following are some of the statements you can write:

```
COMMAND PARM1
COMMAND PARM1 PARM2 PARM3
COMMAND PARM1 PARM1 PARM3
```

Default Value for a Parameter

The placement of YES in the following diagram indicates that it is the default value for the parameter. If you do not include the parameter when you write the statement, the result is the same as if you had actually specified the parameter with the default value.

The following is a diagram of a statement with a default value for an optional parameter:

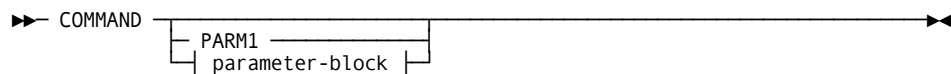


For this command, COMMAND PARM2 is the equivalent of COMMAND PARM1=YES PARM2.

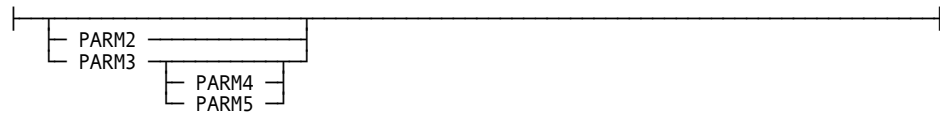
Variables Representing Several Parameters

In some syntax diagrams, a set of several parameters is represented by a single reference.

The following is a diagram of a statement with variables representing several parameters:



Expansion of parameter-block



The *parameter-block* can be displayed in a separate syntax diagram.

Choices you can make from this syntax diagram therefore include, but are not limited to, the following:

```
COMMAND PARM1
COMMAND PARM3
COMMAND PARM3 PARM4
```

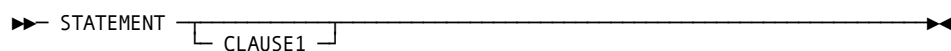
Note: Before you can specify PARM4 or PARM5 in this command, you must specify PARM3.

CA Datacom/DB Extensions

The CA Datacom/DB implementation of SQL conforms to American National Standard Database Language SQL, ANSI X3.135-1989 in the following manner:

1. Full SQL conformance to level 2, except as noted.
2. Includes implementation of the following facilities:
 - a. Direct processing of SQL data manipulation language statements.
 - b. Embedded SQL COBOL and PL/I.

In some instances, the SQL implementation described in this manual has added capabilities beyond the SQL standard. CA Datacom/DB extensions to ANSI standard SQL are indicated for the applicable syntax diagrams as shown in the following example.

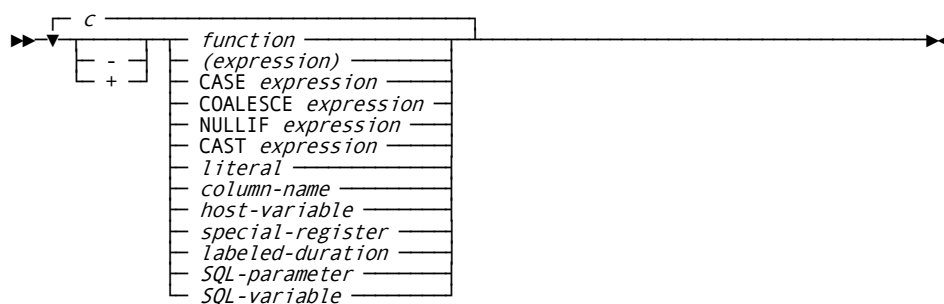


Note: CLAUSE1 is a CA Datacom/DB extension.

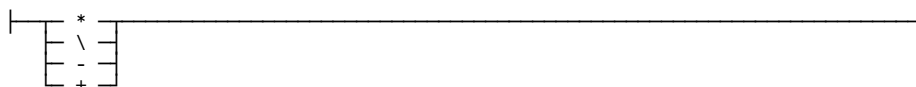
You can choose to use the CA Datacom/DB extension, which is essentially an optional parameter. Some of these extensions are unique to the CA Datacom/DB environment and could function differently from syntactically similar extensions provided by other implementations of SQL.

Chapter 2: Expressions

Expression



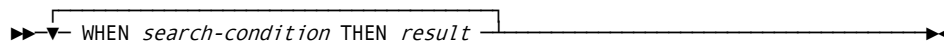
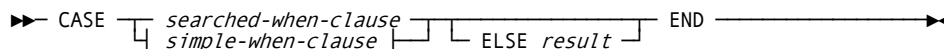
Expansion of *c*



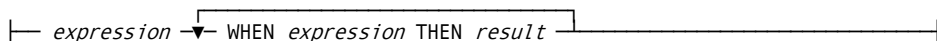
Notes:

- The special-register is a CA Datacom/DB extension. See [Special Registers](#) (see page 18).
- The labeled-duration is a CA Datacom/DB extension. See [Labeled Duration](#) (see page 19).
- For information on *SQL-parameter*, see the *CA Datacom/DB SQL User Guide*.
- For information on *SQL-variable*, see the *CA Datacom/DB SQL User Guide*.

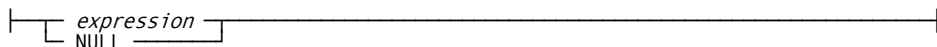
CASE



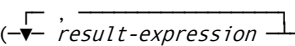
Expansion of *simple-when-clause*



Expansion of *result*



COALESCE Expression

►► COALESCE – ( *result-expression*) ►►

Note: All arguments passed to COALESCE and NULLIF must be of compatible data types.

NULLIF Expression

►► NULLIF – (*result-expression-1*, *result-expression-2*) ►►

Note: All arguments passed to COALESCE and NULLIF must be of compatible data types.

CAST WITHOUT CONVERSION

►► CAST ►►
 ► (*<source-expression>* AS *<target-type>* WITHOUT CONVERSION) ►►

CAST

►► CAST – (*<source-expression>* AS *<target-type>*) ►►

Note: This version of CAST requires that the source value be interpreted first and then stored into the target in converted form.

Special Registers

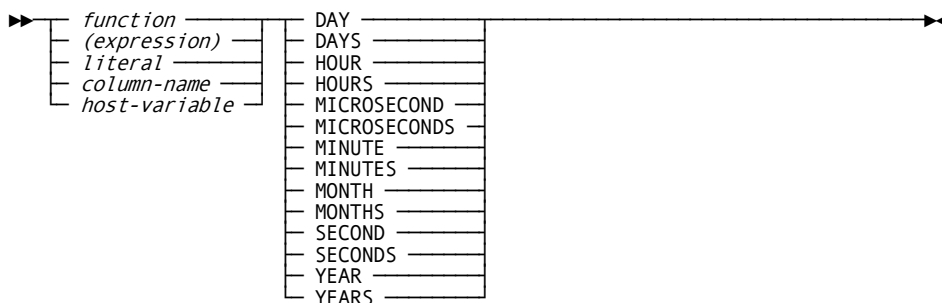
►►

CURRENT DATACOM MUF NAME
CURRENT DATACOM TSN
CURRENT DATE
CURRENT SQLID
CURRENT TIME
CURRENT TIMESTAMP
CURRENT TIMEZONE
SYSTEM USER
USER

 ►►

Labeled Duration

Labeled duration is a CA Datacom/DB extension.



Expressions without Arithmetic Operators

If the arithmetic operators are not used, the result of the expression is the specified value. Some examples of expressions without arithmetic operators are:

- COST -- column-name
- :COST -- host variable
- 'COST' -- string literal
- MIN(COST) -- function performed with column-name as argument

Expressions with the Concatenation Operator

The concatenation operator is represented by two vertical lines (||) or by CONCAT. If the concatenation operator is used in an expression, the result of the expression is a character string. The operands of concatenation must be character strings.

If either operand can be null, the result of the expression can be null. If either operand is null, the result of the expression is the null value. The result otherwise consists of the first operand character string followed by the second operand character string. The length of the result character string is the sum of the lengths of the operand strings.

In a concatenation, if any operand is MIXED, the result of the expression is MIXED. When concatenating a MIXED string ending with Shift-In with a MIXED string beginning with Shift-Out, both Shift characters are omitted in the result.

Note: For more information on MIXED data, see the *CA Datacom/DB SQL User Guide*.

Following is an example of using a concatenation operator:

```
LASTNAME || ' , ' || FIRSTNAME
```

Expressions with Arithmetic Operators

If arithmetic operators are used, the result of an expression is derived by the application of the operators to the value of the operands.

Arithmetic operators must not be applied to character strings. Thus, `USER + 2` is invalid.

The two types of arithmetic operators are prefix operators and infix operators.

Unary Operators

The prefix operator `+` (unary plus) does not change its operand.

The prefix operator `-` (unary minus) reverses the sign of its operand. For example, if the data type of `A` is small integer, then the data type of `-A` is large integer.

The first character of the token following a prefix operator must not be a plus or minus sign. For example, `--10` is not allowed since `10` is preceded by two minus signs.

Note: Unary operators are not allowed with `FLOAT`, `REAL` and `DOUBLE PRECISION` data types.

Infix Operators

The infix operators and the operation they specify are:

Operator	Operation
<code>+</code>	Addition
<code>-</code>	Subtraction
<code>*</code>	Multiplication
<code>/</code>	Division

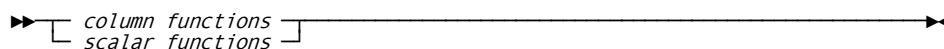
The operand of an infix operator must not be a function that includes the keyword `DISTINCT`.

The value of the second operand of division must not be zero.

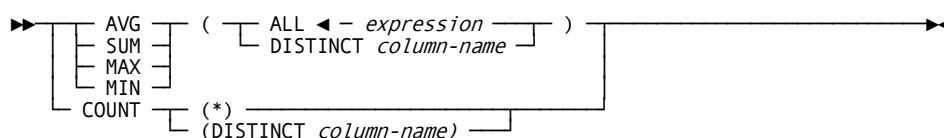
Chapter 3: Functions

Functions

Scalar functions are a CA Datacom/DB extension. See [Scalar Functions](#) (see page 21).

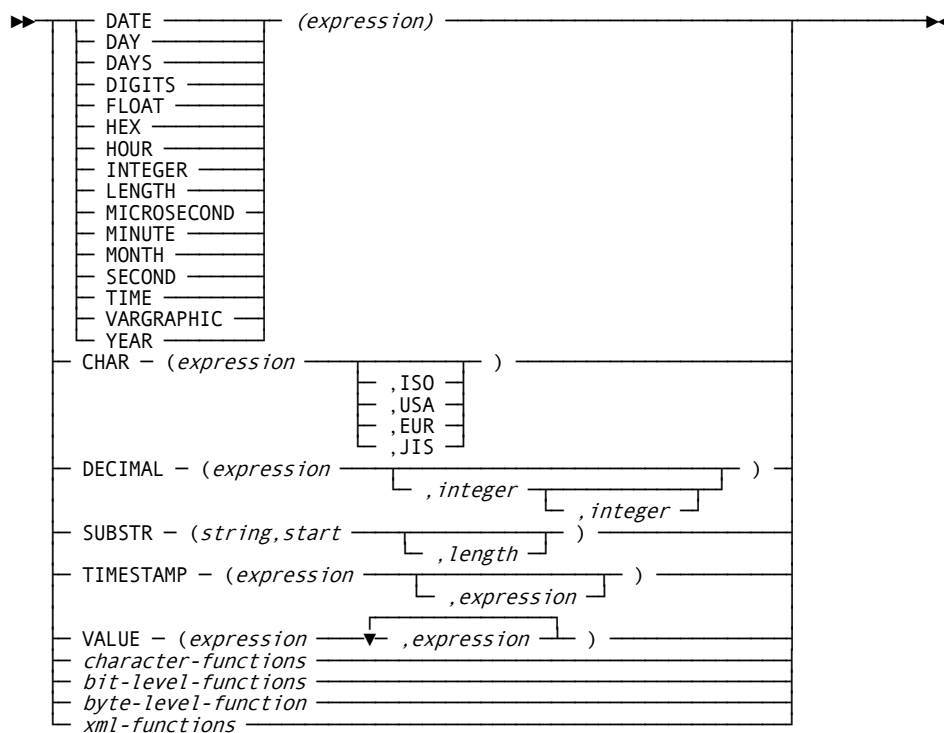


Column Functions



Scalar Functions

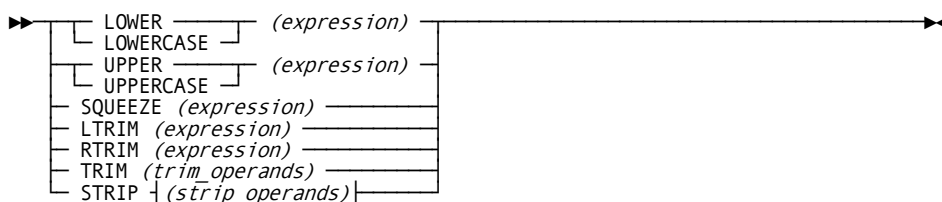
Scalar functions are a CA Datacom/DB extension.



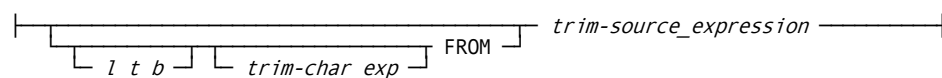
Notes:

- For details on *character-functions*, see [Character Functions](#) (see page 22).
- For details on *bit-level-functions*, see [Bit-Level Functions](#) (see page 22).
- For details on *byte-level-functions*, see [Byte-Level Function](#) (see page 22).
- For details on *xml-functions*, see [XML Functions](#) (see page 22).

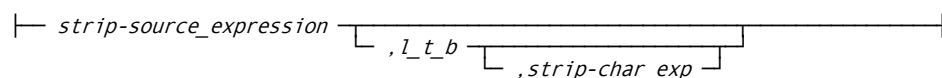
Character Functions



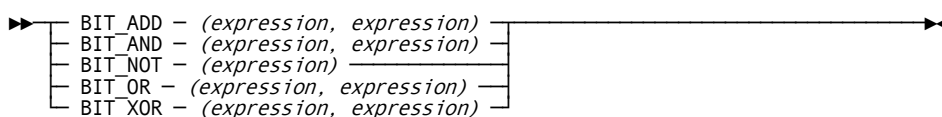
Expansion of trim_operands



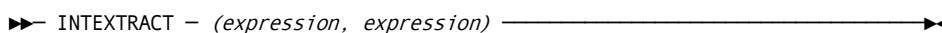
Expansion of strip_operands



Bit-Level Functions

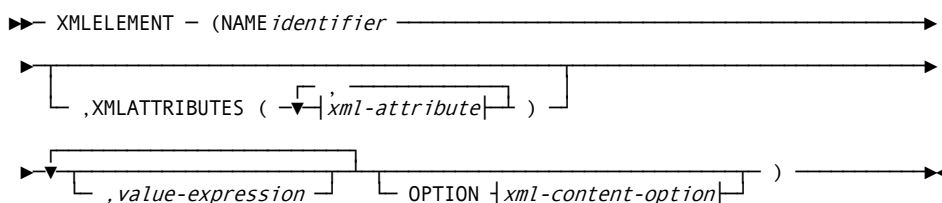


Byte-Level Function



XML Functions

XMLELEMENT and XMLATTRIBUTES



Expansion of xml-attribute

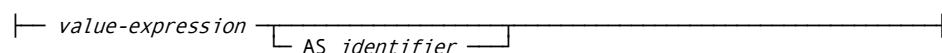


Diagram illustrating the mapping of various null values to a single 'EMPTY ON NULL' value:

- EMPTY ON NULL
- NULL ON NULL
- ABSENT ON NULL
- NIL ON NULL
- NIL ON NO CONTENT

►► XMLFOREST (*forest-element*
 [OPTION *xml-content-option*]) ►

└ *value-expression* ─┬─ *AS identifier* ─┘

Diagram illustrating the mapping of various null values to a single 'NULL ON NULL' value:

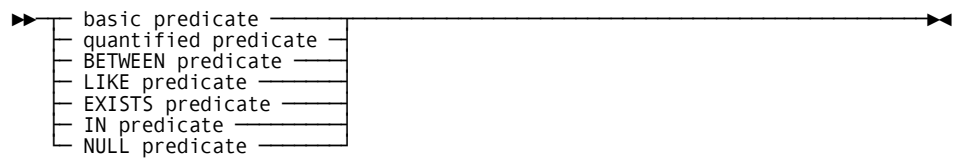
- NULL ON NULL
- EMPTY ON NULL
- ABSENT ON NULL
- NIL ON NULL
- NIL ON NO CONTENT

►► XMLSERIALIZE (DOCUMENT *xml-value-expression AS data-type*) ►►

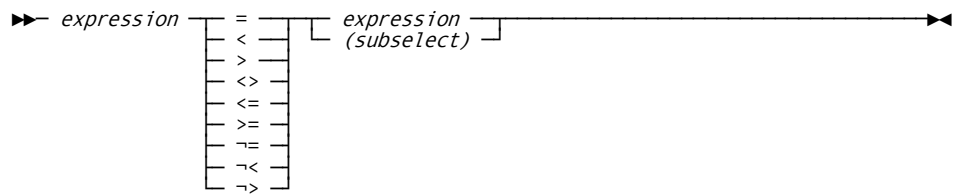
► XMLCONCAT ($\sqrt{}$, $\overline{\text{xml-value-expression}}$) ►

Chapter 4: Predicates

Predicates



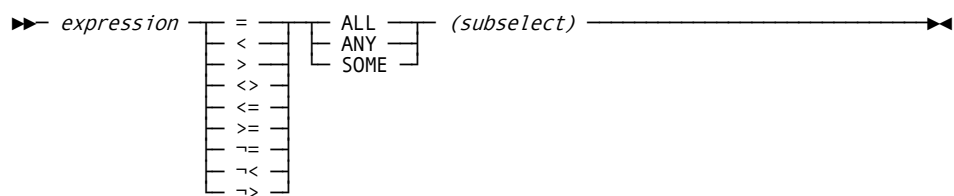
Basic Predicate



The following are CA Datacom/DB extensions:

- <=>
- <=>
- <=>

Quantified Predicate



The following are CA Datacom/DB extensions:

- <=>
- <=>
- <=>

BETWEEN Predicate

►► *expression* NOT BETWEEN *expression* AND *expression* ►►

LIKE Predicate

►► *column-name* NOT LIKE *host-variable*
literal ►►

►► ESCAPE *host-variable*
literal ►►

EXISTS Predicate

►► NOT EXISTS - (*subselect*) ►►

IN Predicate

►► *expression* NOT IN (*subselect*) ►►

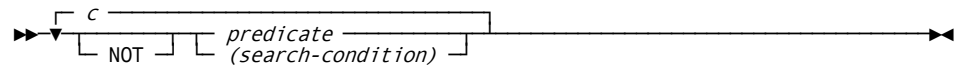
(*host-variable*
literal
special-register)

The *special-register* is a CA Datacom/DB extension. See [Special Registers](#) (see page 18).

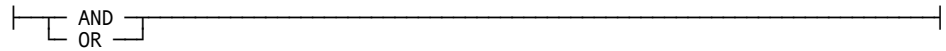
NULL Predicate

►► *column-name* - IS NOT NULL ►►

Chapter 5: Search Conditions



Expansion of c

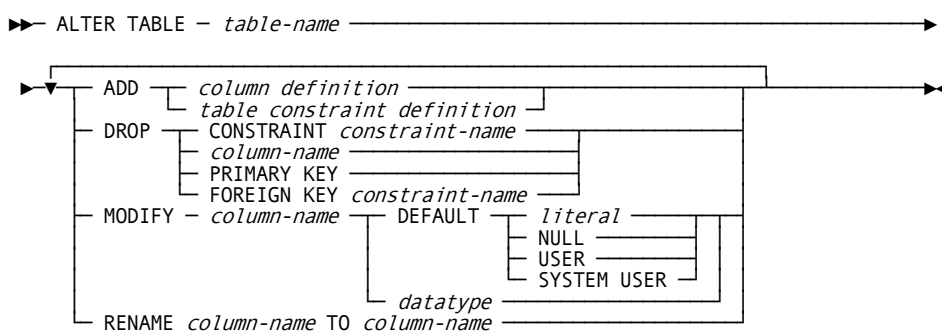


Chapter 6: SQL Statements

SQL Statements

Following are listed the SQL statements in alphabetical order.

ALTER TABLE Statement

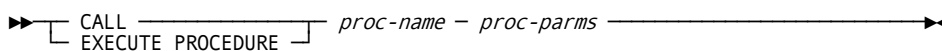


Assignment Statement

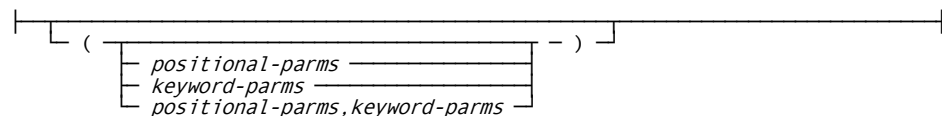
For details about this statement, see [Assignment Statement](#) (see page 33).

This statement is executed from within the Compound Statement. See [Compound Statement Syntax](#) (see page 34).

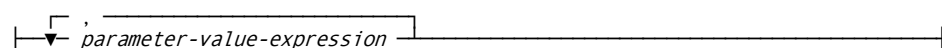
CALL/EXECUTE PROCEDURE Statement



Expansion of *proc-parms*



Expansion of *positional-parm*

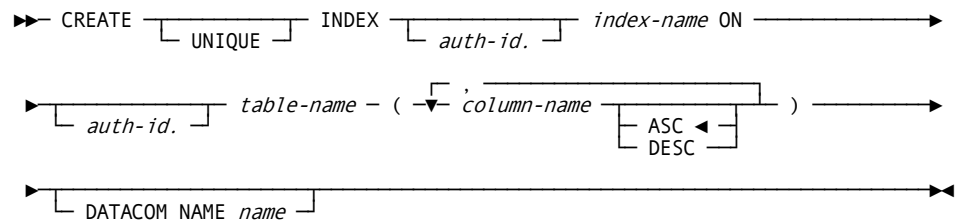


parameter-name=parameter-value-expression

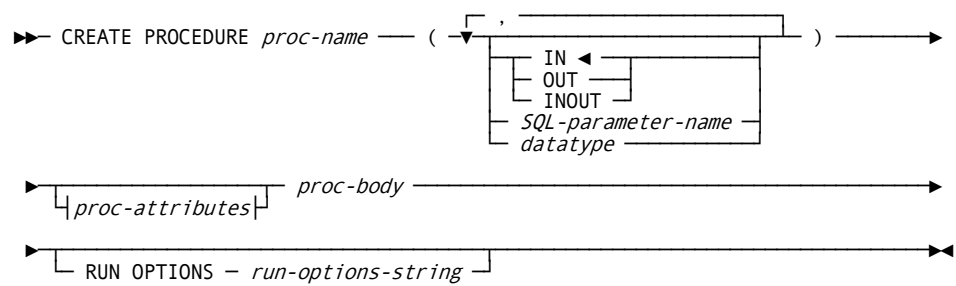
```

graph LR
    A[ ] --- B[literal]
    A --- C[host-variable]
    A --- D["BYREF (host-variable)"]
    A --- E[value-expression]
    A --- F[trigger-column-reference]
    style A width:0px,height:0px
  
```

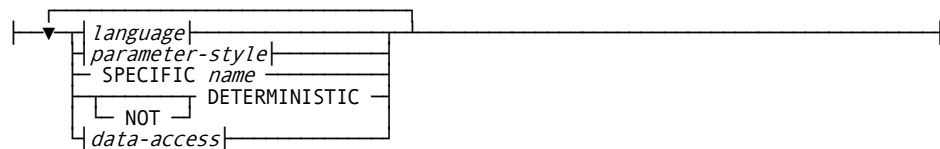
CREATE INDEX Statement



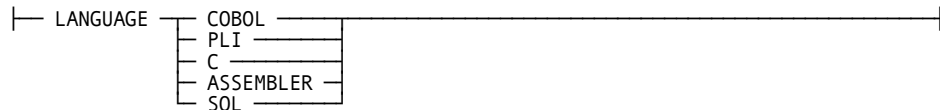
CREATE PROCEDURE Statement



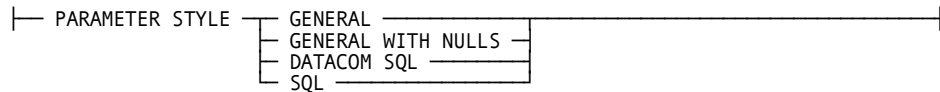
Expansion of proc-attributes



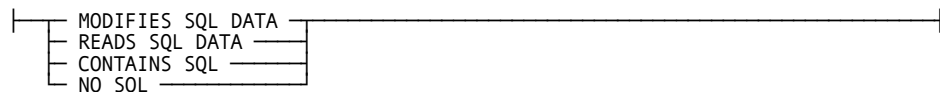
Expansion of language



Expansion of parameter-style



Expansion of data-access



SQL-parameter-name is required for SQL Procedures and, though optional for all others, is recommended for all others as well.

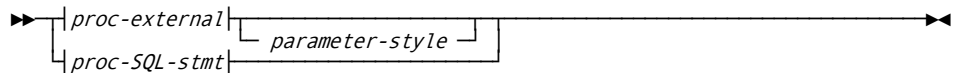
See the separate *proc-body* syntax diagram that follows.

All of the *proc-attributes* choices can be used together, but each can only be used once. If you are using SQL Procedures, *data-access* and *parameter-style* do not apply.

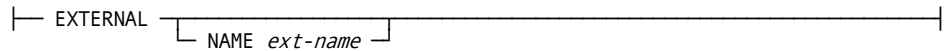
You can specify the *parameter-style* only once, either here as part of the procedure attribute syntax or as part of the EXTERNAL syntax. *Parameter-style* applies only to External Procedures. It does not apply to SQL Procedures. For details, see the *CA Datacom/DB SQL User Guide*.

For LANGUAGE keyword, SQL applies to SQL Procedures. The other choices apply to External Procedures.

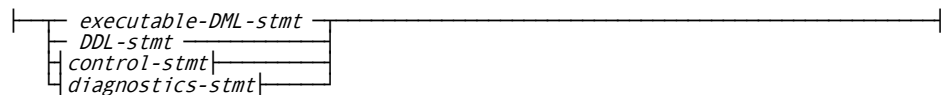
Proc-Body Syntax



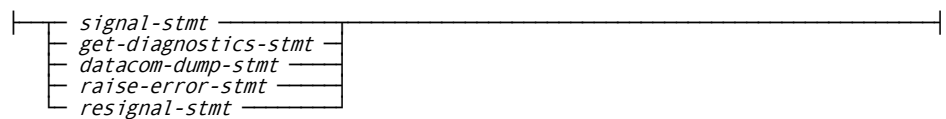
Expansion of *proc_external*



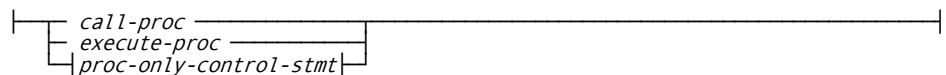
Expansion of *proc-SQL-stmt*



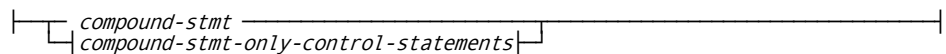
Expansion of *diagnostics-stmt*



Expansion of *control-stmt*



Expansion of *proc-only-control-stmt*



The *proc-external* clause is only used for External Procedures.

The optional *parameter-style* clause is only used for External Procedures.

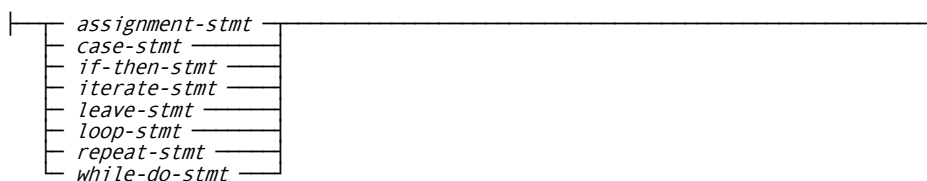
The *proc-SQL-stmt* clause is only used for SQL Procedures.

The executable-DML-stmt is any DML statement except for DECLARE CURSOR (DECLARE CURSOR is supported as part of the compound statement, see [Compound Statement Syntax](#) (see page 34)). For a list of DML statements, see the *CA Datacom/DB SQL User Guide*.

See the separate *diagnostics-stmt* syntax diagram that follows and see the GET DIAGNOSTICS statement on page [GET DIAGNOSTICS Syntax](#) (see page 35).

For the *compound-stmt*, see [Compound Statement Syntax](#) (see page 34).

Expansion of compound-stmt-only-control-statements



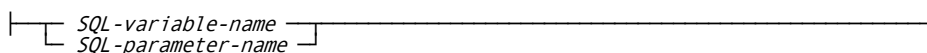
For details on the variables, see the following:

- [Assignment Statement](#) (see page 33).
- [CASE Statement](#) (see page 34).
- [IF-THEN Syntax](#) (see page 36).
- [ITERATE Syntax](#) (see page 36).
- [LEAVE Syntax](#) (see page 36).
- [LOOP Syntax](#) (see page 36).
- [REPEAT-UNTIL Syntax](#) (see page 37).
- [WHILE Statement](#) (see page 38).

Assignment Statement

►► SET ─|assignment-target|─ = proc-value-expression ─────────────────────────────────►►

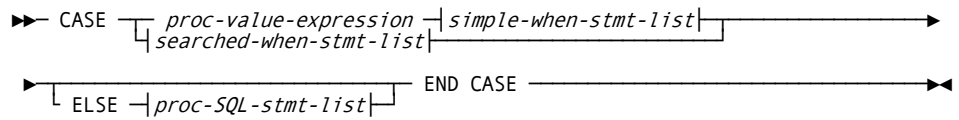
Expansion of assignment-target



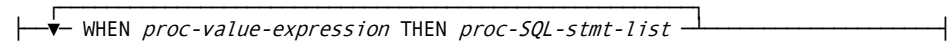
A *proc-value-expression* is a variant form of the expressions used to specify values.

Note: For information about expressions, see the *CA Datacom/DB SQL User Guide*.

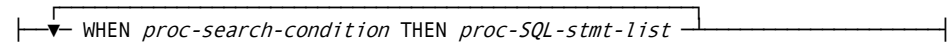
CASE Statement



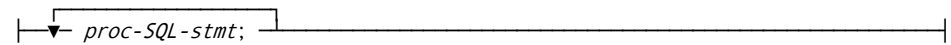
Expansion of *simple-when-stmt-list*



Expansion of *searched-when-stmt-list*



Expansion of *proc-SQL-stmt-list*

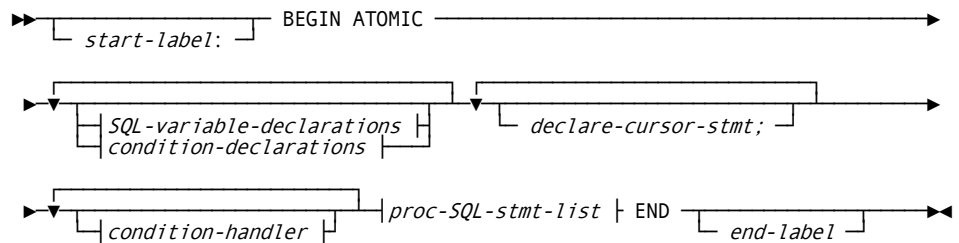


A *proc-value-expression* is a variant form of the expressions used to specify values.

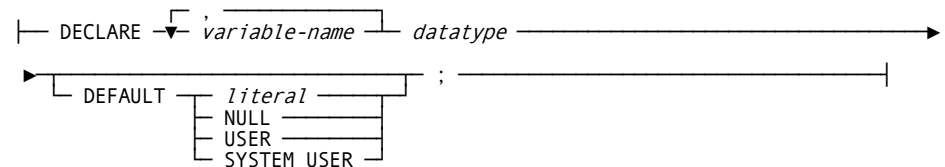
Note: For information about expressions, see the *CA Datacom/DB SQL User Guide*

For *proc-SQL-stmt* syntax, see [Proc-Body Syntax](#) (see page 32).

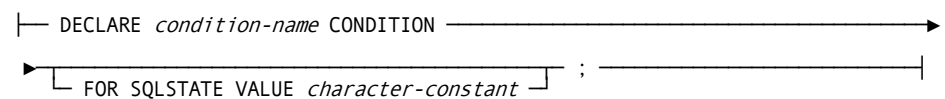
Compound Statement Syntax

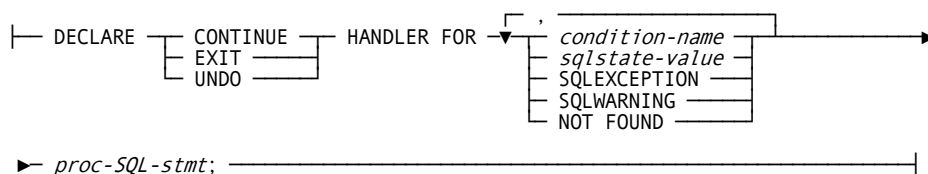
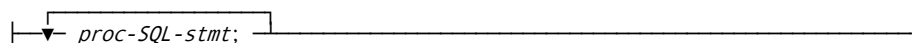


Expansion of *SQL-variable-declarations*



Expansion of *condition-declarations*



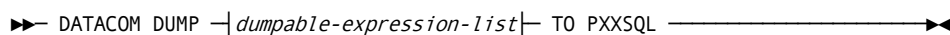
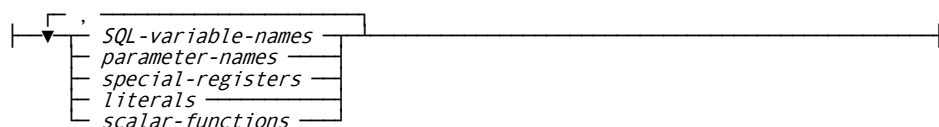
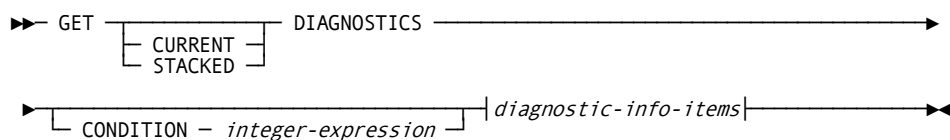
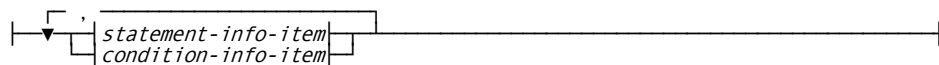
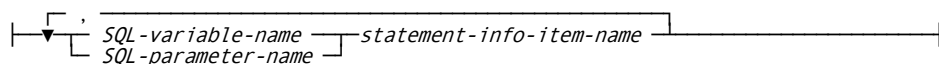
Expansion of condition-handler*Expansion of proc-SQL-stmt-list*

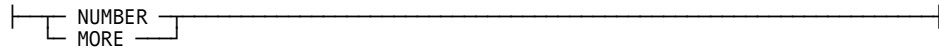
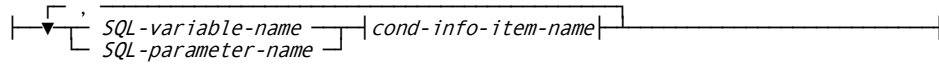
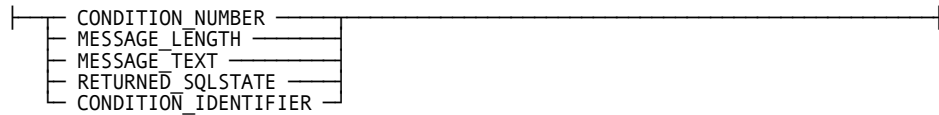
Combinations and multiples of these two optional choices can be used.

The syntax contains a list of DECLARE CURSOR statements, each terminated by a semicolon. For DECLARE CURSOR information, see DECLARE CURSOR.

For *proc-SQL-stmt* syntax, see [Proc-Body Syntax](#) (see page 32).

SYSTEM USER is a CA Datacom/DB extension.

DATA COM DUMP Syntax*Expansion of dumpable-expression-list***GET DIAGNOSTICS Syntax***Expansion of diagnostic-info-items**Expansion of statement-info-item*

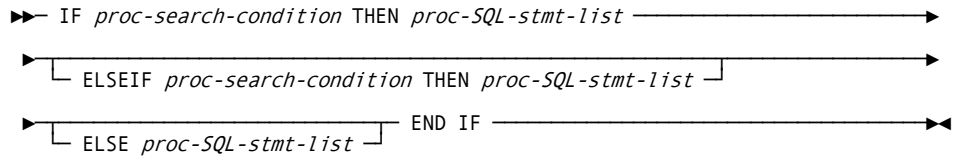
Expansion of statement-info-item-name*Expansion of condition-info-item**Expansion of cond-info-item-name*

CURRENT and STACKED are optional choices for specifying which diagnostics area to use.

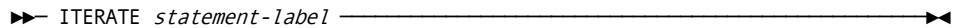
CONDITION followed by an integer is an optional specification for which condition area to use. If you do not specify CONDITION, a default of CONDITION 1 is used.

A specific data type appears in parentheses as shown following each information item name.

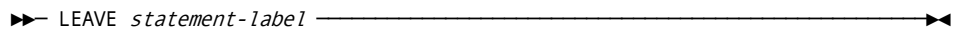
IF-THEN Syntax



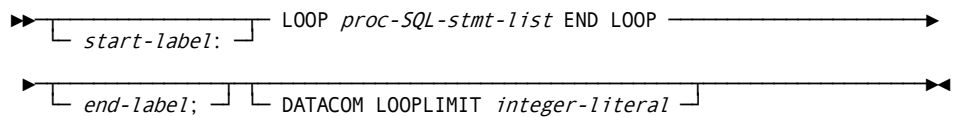
ITERATE Syntax



LEAVE Syntax

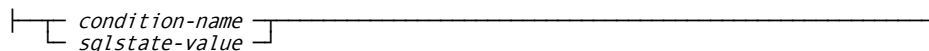


LOOP Syntax



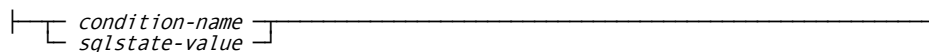
RAISE ERROR Syntax



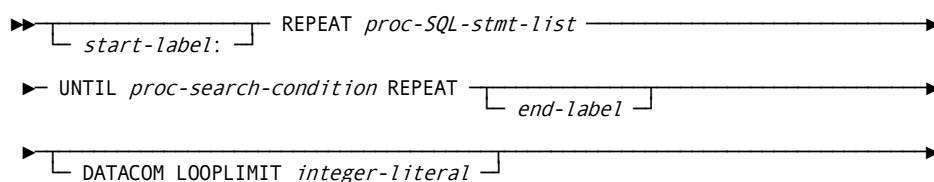
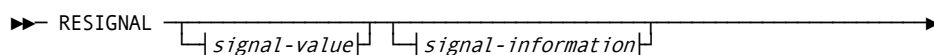
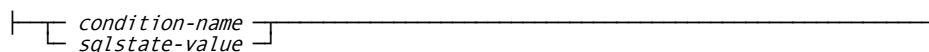
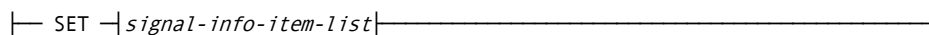
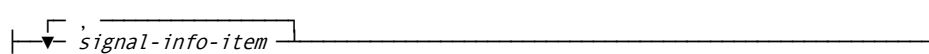
Expansion of signal-value

The RAISE ERROR statement functions identically to the following SIGNAL statement syntax:

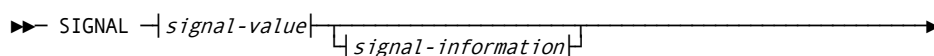
►► SIGNAL ─┤ *signal-value* ─ SET MESSAGE_TEXT = ─ *error-message-text* ─►►

Expansion of signal-value

For more information on the SIGNAL statement, see [SIGNAL Syntax](#) (see page 37).

REPEAT-UNTIL Syntax**RESIGNAL Syntax***Expansion of signal-value**Expansion of signal-information**Expansion of signal-info-item-list**Expansion of signal-info-item*

A *proc-value-expression* is a variant form of the expressions used to specify values (see the information about expressions in the *CA Datacom/DB SQL User Guide*)

SIGNAL Syntax

Expansion of signal-value

| condition-name | _____ |
| sqlstate-value | _____ |

Expansion of signal-information

| SET | signal-info-item-list | _____ |

Expansion of signal-info-item-list

| signal-info-item | _____ |

Expansion of signal-info-item

| condition-info-item-name = proc-value-expression | _____ |

A *proc-value-expression* is a variant form of the expressions used to specify values (see the information about expressions in the *CA Datacom/DB SQL User Guide*)

SIMULATE DATACOM PROCEDURE Syntax

►► SIMULATE DATACOM PROCEDURE *compound-statement* _____ ►►

WHILE Statement

►► start-label: WHILE *proc-search-condition* DO _____ ►►

► *proc-SQL-stmt-list* - END WHILE end-label _____ ►

► DATACOM LOOPLIMIT integer-literal _____ ►►

CREATE RULE Statement

For information about the CREATE RULE statement, see [CREATE TRIGGER/RULE](#) (see page 40).

CREATE SCHEMA Statement

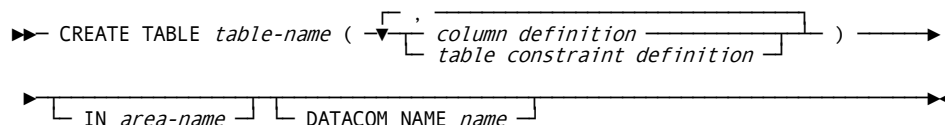
►► CREATE SCHEMA - AUTHORIZATION - *auth-id* _____ ►►

► table-definition view-definition privilege-definition _____ ►►

CREATE SYNONYM Statement

►► CREATE SYNONYM *synonym* FOR auth-id. table-name view-name _____ ►►

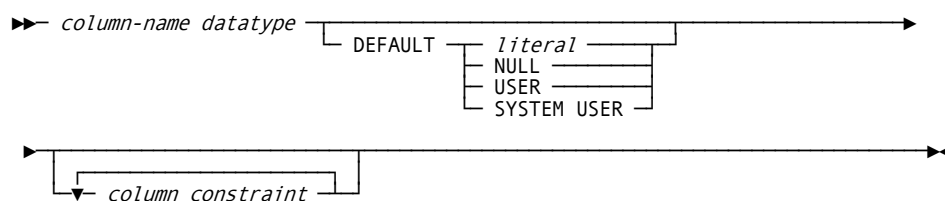
CREATE TABLE



The following are CA Datacom/DB extensions:

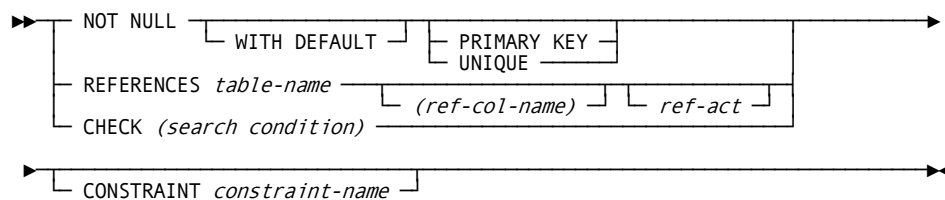
- IN area-name
- DATACOM NAME name

Column Definition



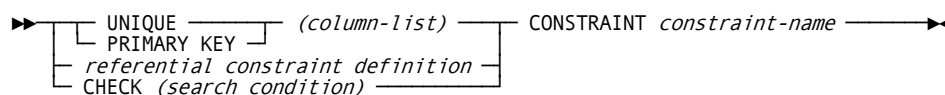
SYSTEM USER is a CA Datacom/DB extension.

Column Constraint Definition

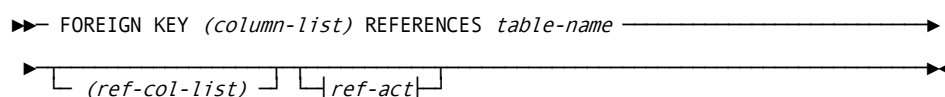


WITH DEFAULT is a CA Datacom/DB extension.

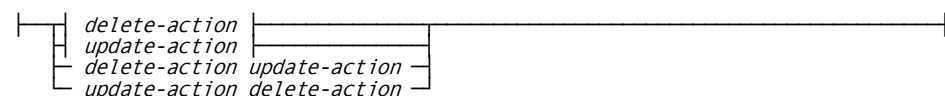
Table Constraint Definition

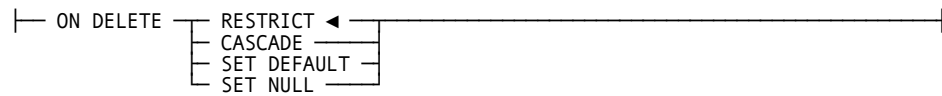
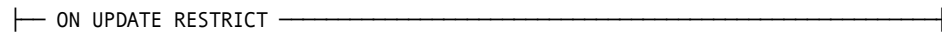
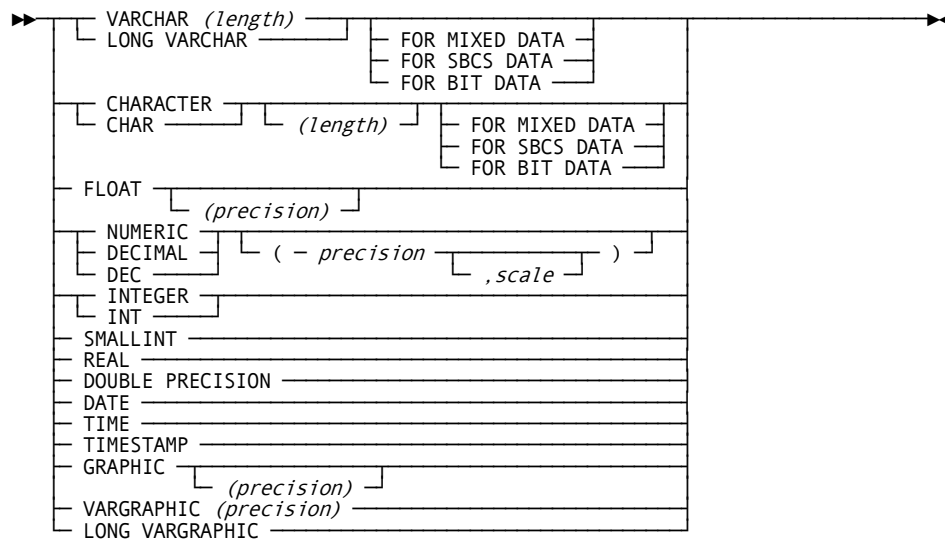


Referential Constraint Definition



Expansion of ref-act (referential action)



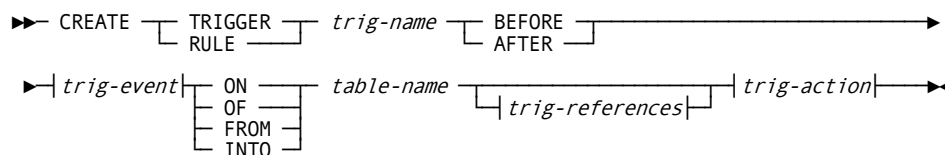
Expansion of delete-action*Expansion of update-action***Data Types**

VARCHAR and LONG VARCHAR are CA Datacom/DB extensions.

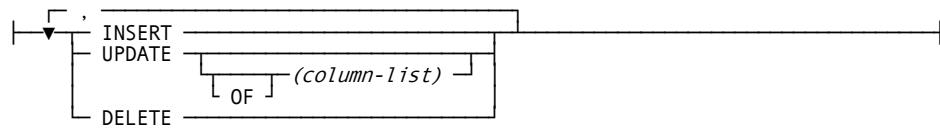
FOR MIXED DATA, FOR SBCS DATA, and FOR BIT DATA are CA Datacom/DB extensions.

DATE, TIME, and TIMESTAMP are CA Datacom/DB extensions.

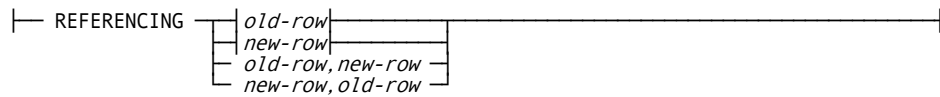
GRAPHIC, VARGRAPHIC, and LONG VARGRAPHIC are CA Datacom/DB extensions.

CREATE TRIGGER/RULE

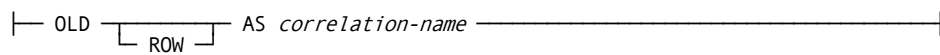
Expansion of trig-event



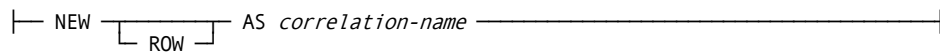
Expansion of trig-references



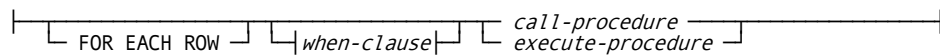
Expansion of old-row



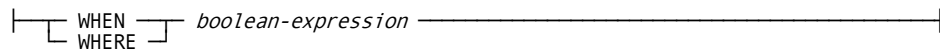
Expansion of new-row



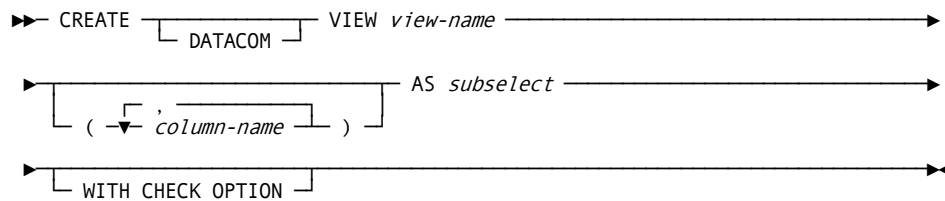
Expansion of trig-action



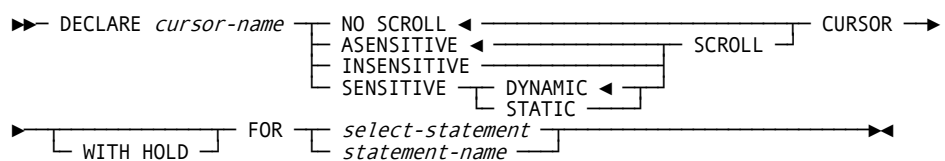
Expansion of when-clause



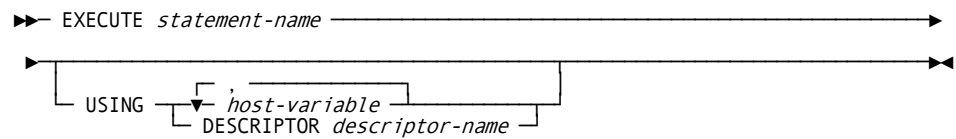
CREATE VIEW



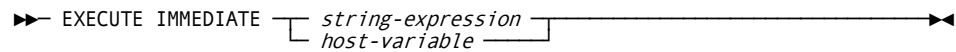
DECLARE CURSOR



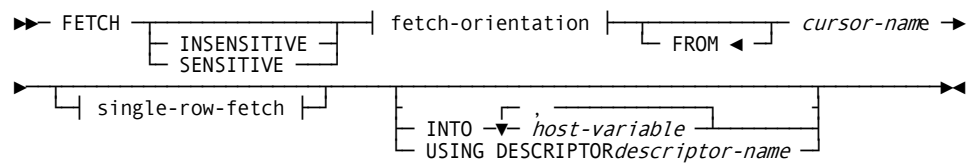
EXECUTE



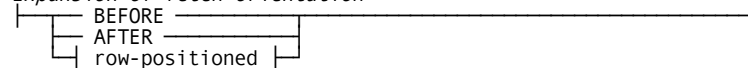
EXECUTE IMMEDIATE



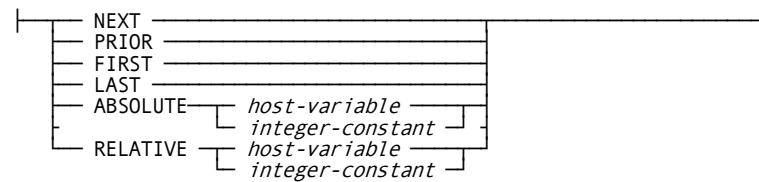
FETCH



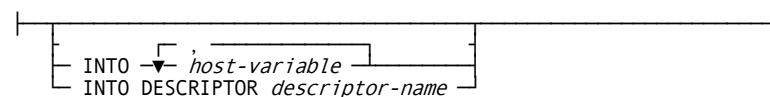
Expansion of *fetch-orientation*



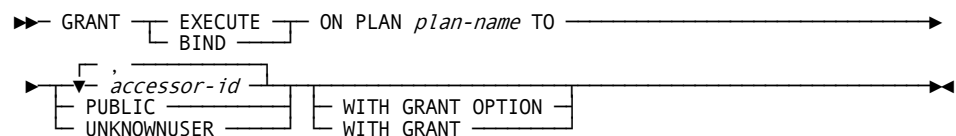
Expansion of *row-positioned*



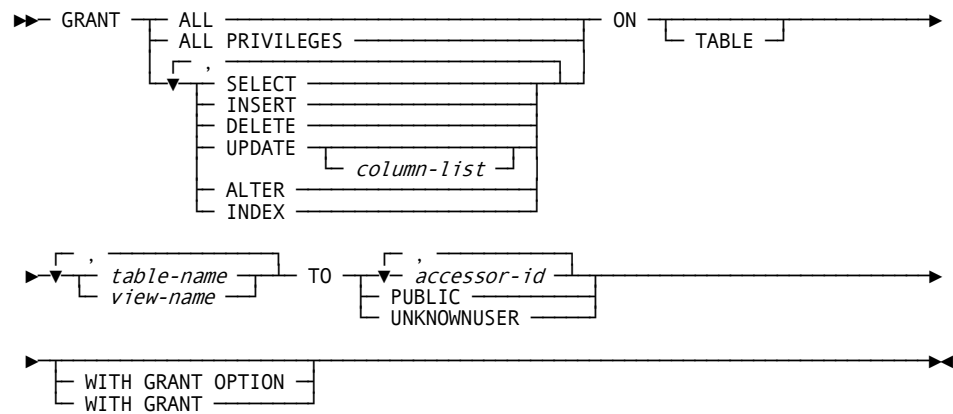
Expansion of *single-row-fetch*



GRANT (Plan Security)



GRANT (Non-Plan Security)



The following are CA Datacom/DB extensions:

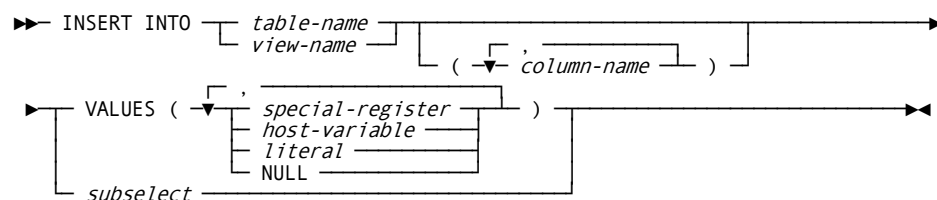
- **ALL**
- **TABLE**
- **ALTER**
- **INDEX**
- **UNKNOWNUSER**
- **WITH GRANT** (without the keyword **OPTION**)

IF-THEN Statement

For details about this statement, see [IF-THEN Syntax](#) (see page 36).

This statement is executed from within the Compound Statement. See [Compound Statement Syntax](#) (see page 34).

INSERT



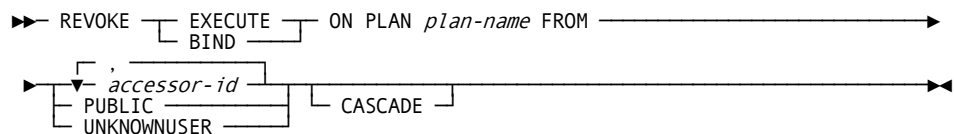
The *special-register* is a CA Datacom/DB extension. See [Special Registers](#) (see page 18).

REPEAT-UNTIL Statement

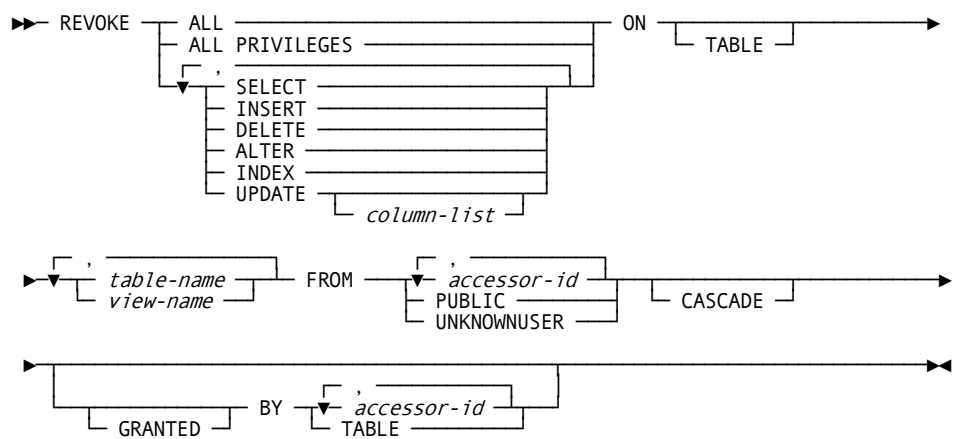
For details about this statement, see [REPEAT-UNTIL Syntax](#) (see page 37).

This statement is executed from within the Compound Statement. See [Compound Statement Syntax](#) (see page 34).

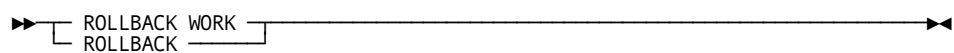
REVOKE (Plan Security)



REVOKE (Non-Plan Security)

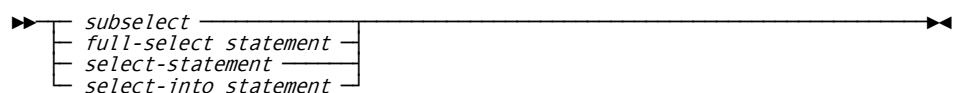


ROLLBACK WORK



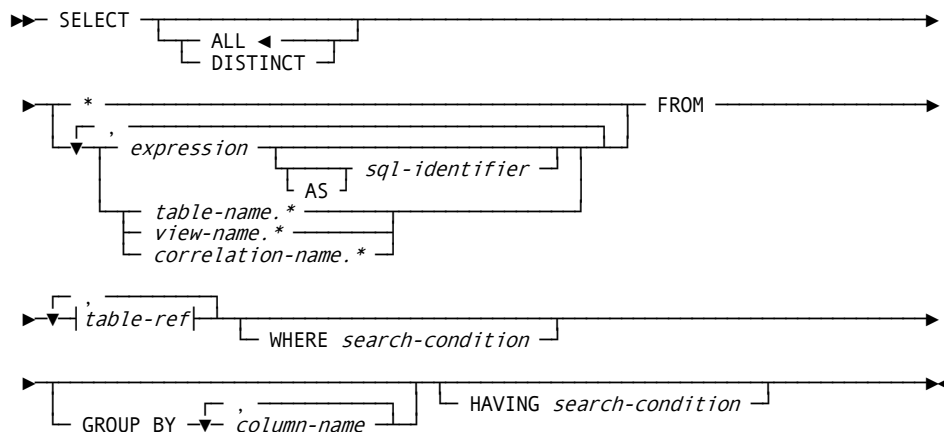
ROLLBACK (without the keyword WORK) is a CA Datacom/DB extension.

SELECT

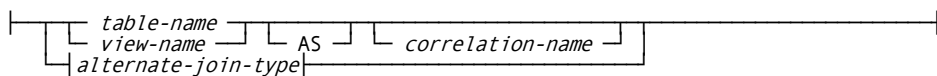


The *select-into* statement is not used by CA Dataquery.

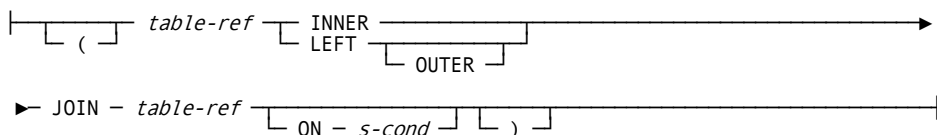
Subselect



Expansion of table-ref

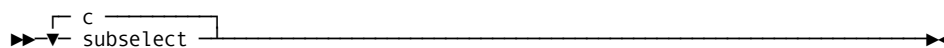


Expansion of alternate-join-type

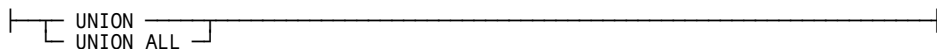


The *s-cond* (search-condition) in this optional ON clause differs from the one in the WHERE clause in that the ON clause defines the join conditions that determine which rows contain nulls, as opposed to the WHERE clause, which eliminates rows from the result entirely. Also, if you use the optional parentheses, they must be balanced, that is, if you use an open parenthesis, you must also use a close parenthesis.

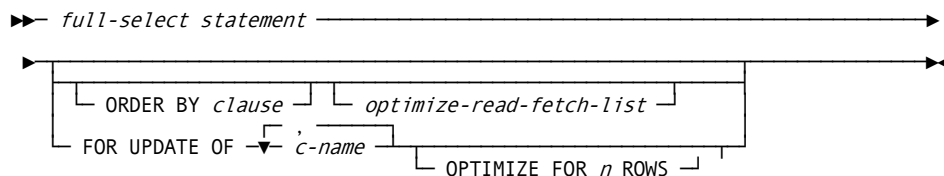
Full-Select Statement

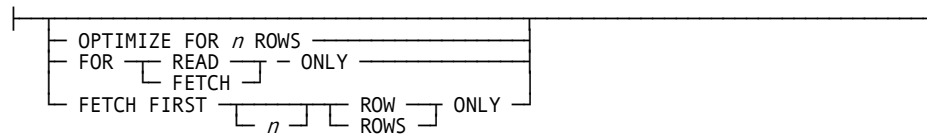


Expansion of c



Select-Statement



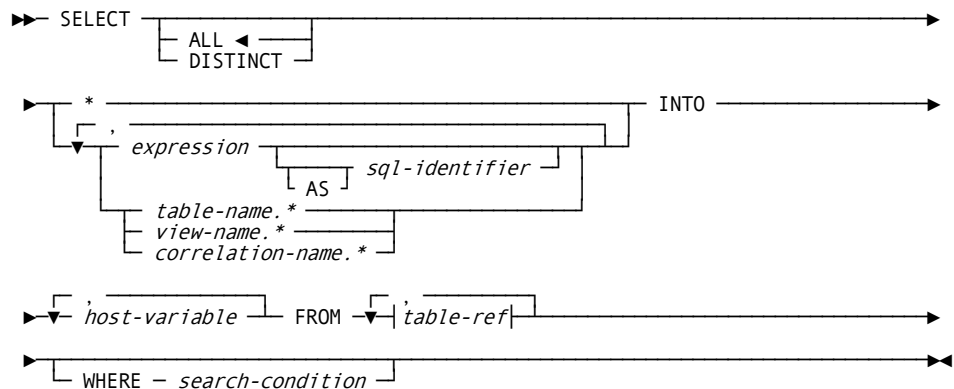
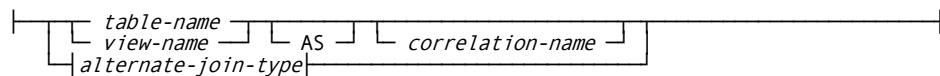
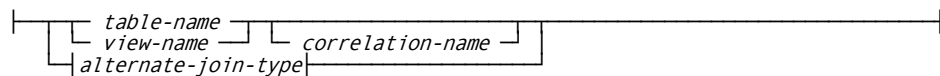
Expansion of optimize-read-fetch-list

Only one use of each *optimize-read-fetch-list* clause is allowed.

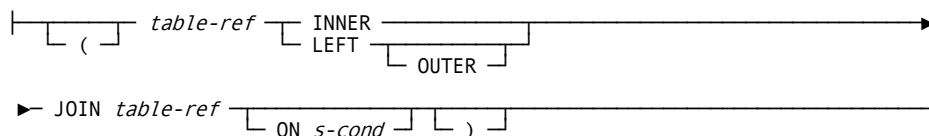
The following are CA Datacom/DB extensions:

- FOR UPDATE OF
- OPTIMIZE FOR *n* ROWS
- FOR READ/FETCH ONLY
- FETCH FIRST

For the syntax of the *full-select* statement, see the *CA Datacom/DB SQL User Guide*.

ORDER BY Clause**Select-Into Statement***Expansion of table-ref**Expansion of table-ref*

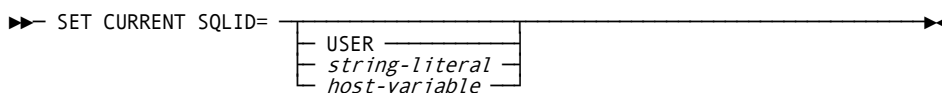
Expansion of alternate-join-type



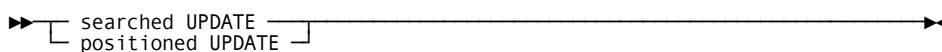
The *s-cond* (search-condition) in this optional ON clause differs from the one in the WHERE clause in that the ON clause defines the join conditions that determine which rows contain nulls, as opposed to the WHERE clause, which eliminates rows from the result entirely. Also, if you use the optional parentheses, they must be balanced, that is, if you use an open parenthesis, you must also use a close parenthesis.

The *select-into* statement is not used by CA Dataquery.

SET CURRENT SQLID

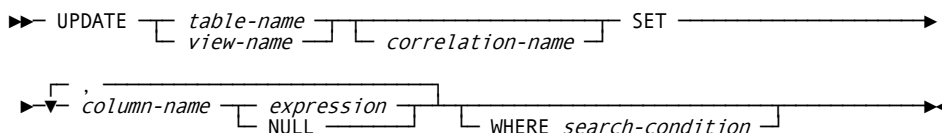


UPDATE

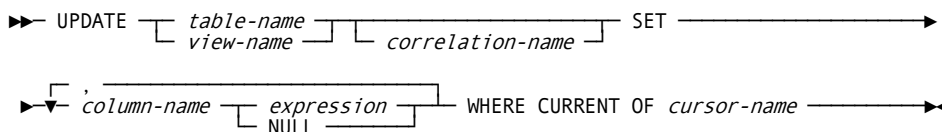


The positioned UPDATE statement is not used by CA Dataquery.

Searched UPDATE

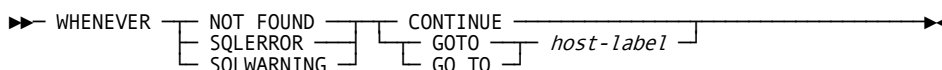


Positioned UPDATE



The positioned UPDATE statement is not used by CA Dataquery.

WHENEVER



SQLWARNING is a CA Datacom/DB extension.

WHILE Statement

For details about this statement, see [WHILE Statement](#) (see page 38).

This statement is executed from within the Compound Statement. See [Compound Statement Syntax](#) (see page 34).

Chapter 7: SQL Codes

Overview

If you are using the CA Datacom/DB SQL Preprocessor or the Interactive SQL Service Facility, errors can generate non-SQL return codes or messages as described in the *CA Datacom/DB Message Reference Guide*.

The SQL Manager passes a value to the SQLCODE field of the SQL Communications Area after each SQL statement is processed during preprocessing or during program execution. Most SQLCODE values indicate that the error was detected within the SQL Manager. These have negative values. If no error is indicated, the value is zero. If the message is a warning, the code has a positive value.

Two special cases are indicated when the values -117 or -118 are placed in the SQLCODE field.

- If SQLCODE contains -117, this indicates the error was detected by CA Datacom/DB and you need to check other SQL Communication Area (SQLCA) fields for the internal and/or external CA Datacom/DB return code to determine what action you should take.
- If SQLCODE contains -118, this indicates the error was detected by the Datadictionary Service Facility and you need to check other SQL Communication Area (SQLCA) fields for a specific Datadictionary Service Facility (DSF) return code to determine what caused the error.
- CA Datacom/DB SQL can also pass warning information in the SQLWARN section of the SQLCA.

Your program should print or display all return code information in the SQLCA.

Note: For information about the SQLCA fields, see the *CA Datacom/DB SQL User Guide*.

The following pages list all of the SQLCODE values and their meanings. For detailed explanations and appropriate user responses for each, see the *CA Datacom/DB Message Reference Guide*.

SQL Return Codes 0 through 243

0	THE COMMAND WAS SUCCESSFUL
100	NO ROW FOUND
170	CONSTRAINT(S) DID NOT CONFIRM - TABLE PLACED IN CHECK STATUS

243 ANSI EXTENSION

SQL Return Codes -1 through -999

-1	NULL INDICATOR VARIABLE REQUIRED FOR OUTPUT HOST VARIABLE NUMBER nnn
-2	EMPTY QUERY STRING
-3	CANNOT REVOKE FROM YOURSELF
-4	OVERFLOW:DATATYPE <x> OPERATION <x>
-5	TABLE tablename IN DDD AND NOT IN CXX
-6	STRING CONSTANT TOO LONG
-7	INCOMPLETE STRING. CHECK FOR MISSING QUOTES
-8	CANNOT GRANT TO YOURSELF
-9	ILLEGAL COLUMN NAME: aaaa
-10	UNABLE TO ALLOCATE MEMORY
-12	COLUMN FUNCTIONS NOT ALLOWED IN SET CLAUSE OF QUERY
-14	TABLE/VIEW DOES NOT EXIST:(table/new-name) AUTHID:(auth-id)
-15	AUTHID (auth-id) NOT DEFINED
-16	DEPENDENT PLAN (auth-id.plan-name) IN USE
-17	MODIFICATION OF COLUMN TO INCOMPATIBLE DATATYPE IS NOT ALLOWED
-18	CHECK OPTION IS NOT ALLOWED ON NON-UPDATABLE VIEWS
-19	INDEX AND TABLE AUTHIDS DO NOT MATCH: aaaa, bbbb
-20	SYNTAX-LAST TOKEN(S): description
-21	DEFAULT CHARACTER STRING TOO LONG
-24	VIEWS CONTAINING FUNCTIONS OR GROUP BY CANNOT BE JOINED
-25	INCOMPLETE COMMENT
-26	EXCEEDS MAX NUMBER OF TABLES/VIEWS PERMITTED IN SUBSELECT
-27	CANNOT USE GROUP BY IN QUERY THAT REFERENCES A GROUPED VIEW
-30	SQL IDENTIFIERS MAY NOT EXCEED 32 BYTES IN LENGTH
-31	CANNOT REFERENCE A VIEW CONTAINING SELECT DISTINCT IN THIS STATEMENT
-32	STATEMENT IS TOO LARGE TO COMPILE
-33	AMBIGUOUS COLUMN REFERENCE: x...x
-34	ILLEGAL CORRELATION NAME nnnn

-35 TTM IS UNUSABLE: SQLCODE xxx, DB RC yy(zzz) ON INITIALIZATION
-36 BAD FOREIGN KEY: reason code,A,B,C,D,E,F,G
-37 CONSTRAINT REBIND FAILED: A,B,C,D,E,F
-38 DB2 TRANSPARENCY OPTION IS NOT INSTALLED
-39 NEGATIVE RESULT FOR UNSIGNED DECIMAL TARGET
-40 CONSTRAINT AUTO-REBIND IN PROGRESS. PLEASE TRY AGAIN.
-41 DATATYPES OF OPERATION op-string NOT COMPATIBLE
-42 INVALID USE OF ESCAPE CHARACTER IN PATTERN OF LIKE PREDICATE
-43 COLUMN FUNCTION ON INVALID TYPE
-44 INPUT HOST VARIABLE NUMBER <xxx> HAS INVALID DATA TYPE <zzz>
-53 STRING TOO LONG FOR COLUMN nnn
-54 UNEQUAL NUMBER OF COLUMNS AND VALUES
-55 DUPLICATE NAMES IN INSERT/ALTER/CREATE LIST: aaaa
-56 entity-type entity-name ALREADY EXISTS
-57 SPECIFIED MORE COLUMNS OR VALUES THAN THE TABLE HAS
-58 COLUMN aaaa TYPE INCOMPATIBLE WITH bbbb TYPE
-60 nnnn IS GREATER THAN MAXIMUM ALLOWABLE NUMBER OF COLUMNS
-66 ILLEGAL NUMBER nnnn IN ORDER SPECIFICATION
-67 COLUMN FUNCTIONS NOT ALLOWED IN WHERE CLAUSE OF QUERY
-69 TOO MANY SET CLAUSES IN UPDATE COMMAND
-70 ORDERING BY COLUMN aaaa THAT IS NOT SELECTED
-71 SELECTING MORE THAN ONE ITEM IN A SUBQUERY
-72 CANNOT NEST COLUMN FUNCTIONS
-73 SELECTED COLUMNS MUST BE IN THE GROUP BY CLAUSE
-75 CANNOT USE COLUMN FUNCTIONS WITH NO GROUP BY CLAUSE
-76 colName IN HAVING NOT GROUP COLUMN OR COLUMN FUNCTION ARGUMENT
-79 UPDATE/DELETE WHERE CURRENT OF cursorname ON READ ONLY CURSOR
-80 ONLY ONE DISTINCT CLAUSE ALLOWED PER QUERY
-82 EACH QUERY IN A UNION MUST HAVE THE SAME NBR OF PROJECTED COLUMNS
-83 DATA TYPES IN COLUMN aaaa DO NOT MATCH
-84 ORDER BY ITEMS MUST BE CONSTANTS FOR UNION OF QUERIES
-85 ORDER BY MUST FOLLOW ALL QUERIES WHEN PERFORMING A UNION

-86	NULL NOT ALLOWED IN COLUMN aaaa
-87	CANNOT USE <UNION> AND <UNION ALL> IN SAME STATEMENT
-101	A SYNONYM ON A SYNONYM IS NOT ALLOWED
-103	MUST USE COLUMN FUNCTIONS IF HAVING CLAUSE WITH NO GROUP BY
-105	aaa IN bb
-110	ORDER BY IS NOT ALLOWED IN A VIEW DEFINITION
-111	THE VIEW IS NOT UPDATABLE BECAUSE IT CONTAINS A SUBQUERY
-113	THE VIEW'S COLUMN IS NOT UPDATABLE: aaaa
-117	RAAT/CBS ERROR: CMD=xxxxx TBL=ttt DBID=nnnn KEY=kkkkk RC=rr (iii)
-118	DSF dsf-code ERROR OCCURRED DURING SQL DSF PROCESSING
-119	product_name: error_text
-120	PROGRAM TIMESTAMP timestamp-in-decimal NOT = PLAN TIMESTAMP timestamp-in-decimal
-121	STATEMENT REBIND ERROR
-122	STATEMENT aaaa NOT FOUND
-123	PLAN ALREADY EXISTS
-124	PLAN authid.plannname DOES NOT EXIST
-125	PLAN CURRENTLY IN USE
-126	NO MORE WORK SPACE
-127	sql-object-name NOT FOUND
-128	LUW ABORTED
-129	RRS COMMIT ROLLBACK
-130	CURSOR NOT UPDATEABLE
-131	RESULT TABLE LENGTH TOO LONG
-132	INVALID PLAN OPTION: STRING DELIMITER
-133	CURSOR NAME aaaa NOT DECLARED
-134	UNKNOWN DYNAMIC/CURSOR STATEMENT-ID (aaaa)
-135	INVALID CURSOR STATE (STMT-ID=aaaa)
-136	SQLDA TRUNCATED - NEEDED aaaa ENTRIES
-137	PLAN OPTION CBSIO EXCEEDED
-138	INVALID PLAN OPTION: PRIORITY
-139	INVALID PLAN OPTION: TIMEMIN
-140	INVALID PLAN OPTION: TIMESEC

-141	INVALID PLAN OPTION: CLOSE OPTION
-142	MORE THAN 1 ROW SELECTED
-143	INVALID PLAN OPTION: DECPOINT
-144	TRX ISOLATION LEVELS UNMATCHED
-145	INVALID PLAN OPTION: ISOLEVEL
-146	AUTHORIZATION ID LENGTH > 18
-147	OP. NOT ALLOWED AT ISOL. LEVEL U
-148	SUBQUERY AT LVL aaaa NBR bbbb RETURNED MORE THAN 1 ROW
-149	PLAN CAN'T SPECIFY DATACOM SQLMODE WHEN MUF OPTION SPECIFIES ANSI
-150	AUTHORIZATION ID MUST MATCH SCHEMA'S AUTHORIZATION ID
-151	INVALID PRECISION OR SCALE SPECIFICATION ON aaaa
-152	MAXIMUM NUMBER OF TABLES EXCEEDED FOR OPERATION
-153	DUPLICATE CURSOR NAME DETECTED
-155	NUMBER OF COLUMNS IN VIEW NOT EQUAL TO NUMBER IN QUERY
-156	VIEW xxx WITH CHECK OPTION VIOLATION
-157	TTM INIT. FAILED. SQL CODE xxx, DB RC yy(zzz).
-158	CANNOT CREATE SYNONYM FOR INVALID VIEW aaaa
-159	HOST VARIABLE INCOMPATIBLE WITH COLUMN AT POSITION aaaa
-161	colname datatype(prec,scale) INVALID DATA: hh
-162	CANNOT INSERT AND SELECT ON THE SAME TABLE
-163	MORE THAN 1 DEFAULT SPECIFIED FOR A COLUMN
-164	CHARACTER DEFAULT MORE THAN 20 BYTES
-165	DUPLICATE CONSTRAINT NAME: nnnnnnnn mmmmmmmm
-166	TBL authid.tblname CHECK PENDING
-167	DOMAIN CONSTRAINT nnnnnnnn.mmmmmmmm VIOLATION
-168	THE TABLE ALREADY HAS A PRIMARY KEY, ONLY ONE MAY BE DEFINED
-169	THE REFERENCED COLUMNS DID NOT REFER TO A CANDIDATE KEY
-171	THE REFERENCED TABLE DOES NOT CONTAIN A PRIMARY KEY
-172	THE NUMBER OF REFERENCING AND REFERENCED COLUMNS DOES NOT MATCH
-173	THE REFERENCING AND REFERENCED COLUMNS ARE NOT EQUIVALENT
-174	UPDATE/DELETE CANNOT HAVE CORRELATED SUBQRY
-175	FOREIGN KEY <authid.constraintname> REFERENCE VIOLATION

- 176 FOREIGN KEY <authid.constraintname> VALUE VIOLATION
- 177 PRIMARY/UNIQUE CONSTRAINT CANNOT BE DROPPED DUE TO REFERENCE
- 178 WITH CHECK OPTION CANNOT BE DROPPED FROM A VIEW DEFINITION
- 179 CONSTRAINT NOT FOUND mmmmm nnnnn
- 180 ROW HELD BY PREVIOUS LOGICAL UNIT OF WORK WAS DELETED
- 181 ROW HELD BY PREVIOUS LOGICAL UNIT OF WORK WAS UPDATED
- 182 UNIQUE/PRIMARY CANNOT BE SPECIFIED DUE TO NULL ON nnnnnn
- 183 MUST SPECIFY DATACOM SQLMODE FOR CURSORS TO SURVIVE UNITS OF WORK
- 184 COLUMN NAME LIST REQUIRED ON CREATE VIEW CONTAINING NON-SIMPLE COLUMN
- 185 accessor-id HAS NO action-type AUTHORITY ON object
- 188 TABLE CANNOT BE DROPPED BECAUSE IT IS REFERENCED
- 189 NO RESULT FORMAT SPECIFIED
- 190 OPERAND 1 OR RESULT FORMAT MUST BE INTERNAL TO DO CONVERSION
- 191 BOTH OPERANDS MUST BE IN INTERNAL FORMAT
- 192 RESULT FORMAT MUST BE INTERNAL TO DO VERIFICATION
- 193 CENTURY nn IS LARGER THAN 99
- 194 IT IS INVALID TO ADD xx AND zz DATATYPES
- 195 IT IS INVALID TO SUBTRACT xx AND zz DATATYPES
- 198 YEAR yy IS LARGER THAN 99
- 199 MONTH mm IS NOT BETWEEN 1 AND 12 INCLUSIVE
- 201 HOUR hh OUT OF RANGE
- 202 MINUTE mm OUT OF RANGE
- 203 SECOND ss OUT OF RANGE
- 204 MICROSECOND ssssss GREATER THAN 99
- 205 INVALID DATATYPE xx FOR SCALAR FUNCTION nn
- 206 OP2 NOT ALLOWED IF OP1 IS A TIMESTAMP FOR SCALAR FUNCTION CHAR
- 207 RESULT DATATYPES IN CASE/COALESCE/NULLIF EXPRESSION ARE INCOMPATIBLE
- 208 CASE EXPRESSION MUST CONTAIN AT LEAST ONE NON-NULL RESULT EXPRESSION
- 209 operation DISALLOWED ON object
- 210 ROLLBACK INCOMPATIBLE WITH USE OF RESOURCE RECOVERY SERVICES
- 214 ARRAY INDEX MUST BE A LITERAL IN THIS CONTEXT
- 215 <COLUMN-NAME> INDEX INDEX-value MUST BE BETWEEN lower-limit AND upper-limit INCLUSIVE

- 216 BAD ARRAY VALUE: reason
- 217 disallowed-language-element NOT ALLOWED disallowed-context
- 219 JULIAN DATE HAS GONE NEGATIVE: dddd
- 223 A TIME STRING CANNOT CONTAIN MORE THAN ONE A: aa
- 224 A TIME STRING CANNOT CONTAIN MORE THAN ONE P: pp
- 225 A TIME STRING CANNOT CONTAIN MORE THAN ONE BLANK BEFORE THE A OR P
- 226 NO M AFTER THE A OR P OR TOO MANY COLONS IN A TIME STRING
- 227 THERE MUST BE AN M AFTER THE A OR P IN A TIME STRING
- 228 TOO MANY NODES IN CHARACTER STRING: node count is n
- 229 TOO MANY DIGITS PER NODE IN STRING: xxxx
- 230 THE VALUE OF A NODE IS OUT OF RANGE: xxxx
- 231 NUMBER OF SLASHES IS n, ONLY TWO ALLOWED: xxxx
- 232 NO AM OR PM FOUND IN TIME CHARACTER STRING WITH ONE COLON: xxxx
- 233 NUMBER OF COLONS MUST BE 1 OR 2. FOUND n
- 234 FOUND AN M BUT NO A OR P TO GO WITH IT: xxxx
- 235 NUMBER OF PERIODS CAN ONLY BE 2. FOUND n
- 236 NUMBER OF DASHES CAN ONLY BE 3. FOUND n
- 237 NUMBER OF DIGITS FOR MICROSECONDS CANNOT EXCEED 6, FOUND n: xxxx
- 238 UNRECOGNIZABLE DATE/TIME FORMAT: xxxx
- 241 SQL REFERENTIAL INTEGRITY IS NOT ALLOWED ON A DL1/T TABLE
- 242 CONFLICT ALTERING COLUMN xxx
- 243 ANSI EXTENSION
- 244 REJECT OF NON-ANSI SQL-86 FEATURE (SQLMODE=ANSI/FIPS)
- 245 CONFLICTING SECURITY MODES
- 246 INVALID AREA NAME
- 247 accessor-id HAS NO CREATE SCHEMA AUTHORITY
- 248 accessor-id HAS NO AUTHORITY TO USE "GRANTED BY" FORM OF REVOKE
- 249 REFERENCED TABLE aaa.ttt PLUS n OTHERS IN CHECK STATUS
- 250 EXCEPTION TABLE authid.tblname NOT FOUND
- 251 TABLE HAS NO CONSTRAINTS
- 252 TABLE NOT IN A CHECK STATE
- 254 REVOKE WOULD CASCADE BUT "CASCADE" NOT SPECIFIED

-255	COLUMN xxx NOT NULL WITHOUT DEFAULT
-256	RQA TOO LARGE FOR TABLE ttt
-257	FOREIGN KEY COLUMN xxx NOT FOUND
-258	TABLE FULL - DBID=nnnn INTERNAL NAME aaa
-259	INDEX FULL - DBID=nnnn
-260	UNIQUE CONSTRAINT constraint-name DUPLICATES KEY key-name
-261	UNIQUE constraint-name COLUMN column-name NOT FOUND
-262	REFERENCING TABLE xxx.xxx IN CHECK-RELATED STATE
-263	DUPLICATE KEY <keyname> VALUE REJECTED DBID <dbid> TBL <tbl>
-264	INDEX ALREADY EXISTS xxx
-265	INDEX DOES NOT EXIST xxx
-269	FOREIGN KEY LOOP
-272	ERROR DURING AUTOMATIC REBINDING OF VIEW authid.viewname
-273	GRANT AND REVOKE STATEMENTS NOT ALLOWED
-274	CANNOT CREATE AN INDEX ON A VIEW
-275	TOO MANY HOST VARIABLES nnn
-276	TOO FEW HOST VARIABLES nnn
-277	aaaa RECEIVED UNKNOWN DURATION TYPE
-278	INVALID PREC(xxx) OR SCALE(zzz) FOR DECIMAL DATATYPE
-279	INVALID DATE TABLE INDEX (xxx)
-280	INVALID TIME TABLE INDEX (xxx)
-281	INVALID TIMESTAMP TABLE INDEX (xxx)
-282	NO OPERAND NODE RECEIVED
-283	NO OPERAND VALUE RECEIVED
-284	NON-NUMERIC ITEM FOUND <x> IN STRING <aaaa>
-286	SUBSTR FUNCTION ARGUMENTS <START=aaaa, LENGTH=bbbb> ARE OUT OF RANGE
-287	DL1/T CONSTRAINT ENFORCEMENT MODULE DLSERV NOT FOUND
-289	<x> LITERAL <y> TOO LARGE/SMALL
-290	DATATYPE OF FIRST OPERAND OF LIKE NOT CHARACTER STRING
-291	INDEX <xxxxx> NOT IN SEQUENCE
-292	<nnn> HOST VARIABLES FOR <nnn> SELECT LIST COLUMNS
-293	PREDICATE INVALID BECAUSE HOST VARIABLE <nnn> HAS THE NULL VALUE

-294	INVALID MIXED DATA IN VALUE ASSIGNED TO column_name
-296	INVALID MIXED DATA IN COLUMN column_name
-297	INVALID MIXED DATA IN LIKE PATTERN
-298	INVALID MIXED DATA IN VARGRAPHIC OPERAND
-299	LIKE ESCAPE OPTION NOT ALLOWED WITH MIXED DATA
-300	CANNOT EXECUTE UNPREPARED/INVALID STATEMENT statement_name
-301	PREPARE OF SELECT REJECTED - CURSOR cursor_name IN OPEN STATE
-302	INVALID USE OF PARAMETER MARKER
-303	STATEMENT-NAME OF CURSOR cursor_name NOT PREPARED SELECT STATEMENT
-304	CANNOT DESCRIBE UNPREPARED STATEMENT statement_name
-305	SQLDA HAS count HOST VARIABLES - STATEMENT HAS count PARAMETER MARKERS
-306	STATEMENT TYPE CANNOT BE PREPARED
-307	STATEMENT-NAME statement_name CAN ONLY BE USED WITH ONE CURSOR
-308	PREPARE HOST VARIABLE SOURCE STATEMENT LENGTH INVALID
-309	HOST VARIABLES CANNOT BE USED IN DYNAMICALLY PREPARED STATEMENTS
-310	COMBINATION OF CHECKPLAN=checkplan_value, CHECKWHO=checkwho_value, AND CHECKWHEN=checkwhen_value IS INVALID
-311	EMBEDDED UPDATE OR DELETE REFERENCES DYNAMIC CURSOR
-312	NO SQLDA FOR EXECUTE OF A STATEMENT WITH PARAMETER MARKERS
-313	NO SUCH PRIVILEGE IS grantable or revokable ON object_name
-314	EXTERNALLY SECURED PLANS REQUIRE AUTHID.PLAN-NAME < 26 BYTES
-315	ONLY THE PLAN CREATOR MAY MODIFY PLAN SECURITY OPTIONS DURING REBIND
-316	%s.%s OUTSIDE SCOPE OF JOINED TABLE
-317	CURSOR cursor-name ERROR: MULTIPLE stmt-type STATEMENTS ATTEMPTED
-317	CURSOR REQUIRED ERROR: CURSOR REQUIRED FOR NON-DYNAMIC SELECT
-318	URI=YES IS REQUIRED FOR SQL TO USE TABLE (DATACOM-NAME) ttt DBID nnnny
-319	LUW IS TIMING OUT - TRY AGAIN (LUWID=luw-id-string, TASK=task-nbr, JOB=job-name)
-320	message describing user error
-321	INVALID SQLCODE sqlcode HAS BEEN GENERATED
-322	INVALID SQLSTATE
-323	LUW STATEMENT LIMIT OF x STATEMENTS EXCEEDED - TRY A COMMIT
-324	INVALID NUMERIC LITERAL n
-326	MAXIMUM NUMBER (number) OF DIAGNOSTICS AREAS EXCEEDED

-435	USER CONDITION user-cond-name SIGNALLED WITH NO ATTACHED ERROR MESSAGE
-436	RESIGNAL FOR SQLSTATE sqlstate REQUESTED WHEN HANDLER NOT ACTIVE
-437	NO HANDLER EXISTS FOR USER-DEFINED CONDITION NAME name-string
-438	SQLSTATE sqlstate-value WAS SIGNALLED WITH NO ATTACHED ERROR MESSAGE
-439	sqlstate-value IN handler-name:error-description
-502	CURSOR ALREADY OPEN
-503	SQL REJECTED INVALID INPUT FOR REASON
-504	PLAN OPTION ERROR: error-string
-505	ON-CLAUSE ERROR: x NOT ALLOWED
-530	PROC authid.name: msg-string
-531	PROC authid.name: msg-string
-532	TRIG authid.name: msg-string
-533	TRIG authid.name: msg-string
-534	msg-string
-535	PROC authid.name: msg-string
-537	msg-string
-559	SECURITY NOT ENABLED
-560	TEMPORARY TABLE AREA (TTM) FULL
-561	TTM TOO SMALL, SEE ERROR ACTION
-562	INVALID DATACOM NAME
-563	INVALID ORDER: <aaa>---USE 'ASC' OR 'DESC'
-564	ONLY ONE PRIMARY KEY MAY BE DEFINED
-565	HEX LITERALS MUST HAVE EVEN NUMBER OF HEX DIGITS: xxx
-566	INVALID HEX DIGIT IN HEX LITERAL <xxxx>--USE 0-9, A-F
-985 -998	SQLnnnn: xxxx ...
-999	INTERNAL ERROR (file-name LINE line-number): specific-error