



Teradata Database

SQL Quick Reference

Release 13.0
B035-1510-098B
April 2009

The product or products described in this book are licensed products of Teradata Corporation or its affiliates.

Teradata, BYNET, DBC/1012, DecisionCast, DecisionFlow, DecisionPoint, Eye logo design, InfoWise, Meta Warehouse, MyCommerce, SeeChain, SeeCommerce, SeeRisk, Teradata Decision Experts, Teradata Source Experts, WebAnalyst, and You've Never Seen Your Business Like This Before are trademarks or registered trademarks of Teradata Corporation or its affiliates.

Adaptec and SCSISelect are trademarks or registered trademarks of Adaptec, Inc.

AMD Opteron and Opteron are trademarks of Advanced Micro Devices, Inc.

BakBone and NetVault are trademarks or registered trademarks of BakBone Software, Inc.

EMC, PowerPath, SRDF, and Symmetrix are registered trademarks of EMC Corporation.

GoldenGate is a trademark of GoldenGate Software, Inc.

Hewlett-Packard and HP are registered trademarks of Hewlett-Packard Company.

Intel, Pentium, and XEON are registered trademarks of Intel Corporation.

IBM, CICS, RACF, Tivoli, and z/OS are registered trademarks of International Business Machines Corporation.

Linux is a registered trademark of Linus Torvalds.

LSI and Engenio are registered trademarks of LSI Corporation.

Microsoft, Active Directory, Windows, Windows NT, and Windows Server are registered trademarks of Microsoft Corporation in the United States and other countries.

Novell and SUSE are registered trademarks of Novell, Inc., in the United States and other countries.

QLogic and SANbox are trademarks or registered trademarks of QLogic Corporation.

SAS and SAS/C are trademarks or registered trademarks of SAS Institute Inc.

SPARC is a registered trademark of SPARC International, Inc.

Sun Microsystems, Solaris, Sun, and Sun Java are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.

Symantec, NetBackup, and VERITAS are trademarks or registered trademarks of Symantec Corporation or its affiliates in the United States and other countries.

Unicode is a collective membership mark and a service mark of Unicode, Inc.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN “AS-IS” BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. IN NO EVENT WILL TERADATA CORPORATION BE LIABLE FOR ANY INDIRECT, DIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS OR LOST SAVINGS, EVEN IF EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The information contained in this document may contain references or cross-references to features, functions, products, or services that are not announced or available in your country. Such references do not imply that Teradata Corporation intends to announce such features, functions, products, or services in your country. Please consult your local Teradata Corporation representative for those features, functions, products, or services available in your country.

Information contained in this document may contain technical inaccuracies or typographical errors. Information may be changed or updated without notice. Teradata Corporation may also make improvements or changes in the products or services described in this information at any time without notice.

To maintain the quality of our products and services, we would like your comments on the accuracy, clarity, organization, and value of this document. Please e-mail: teradata-books@lists.teradata.com

Any comments or materials (collectively referred to as “Feedback”) sent to Teradata Corporation will be deemed non-confidential. Teradata Corporation will have no obligation of any kind with respect to Feedback and will be free to use, reproduce, disclose, exhibit, display, transform, create derivative works of, and distribute the Feedback and derivative works thereof without limitation on a royalty-free basis. Further, Teradata Corporation will be free to use any ideas, concepts, know-how, or techniques contained in such Feedback for any purpose whatsoever, including developing, manufacturing, or marketing products or services incorporating Feedback.

Copyright © 2000 – 2009 by Teradata Corporation. All Rights Reserved.

Preface

Purpose

This book is a quick reference for the SQL dialect supported by the Teradata Database.

Audience

All users of Teradata SQL who need information about how to structure an SQL statement.

Supported Software Release

This book supports Teradata® Database 13.0.

Prerequisites

You should be familiar with basic computer technology, the Teradata Database, and the Teradata SQL language.

It may be helpful to review the following books:

- *Introduction to Teradata*
- The SQL book set

Changes to This Book

Date	Description
Teradata Database 13.0 April 2009	<ul style="list-style-type: none">• Updated syntax diagrams throughout the book• Added the following new syntax diagrams to Chapter 1:<ul style="list-style-type: none">• Period Literals• PERIOD(DATE) Data Type• PERIOD(TIME) Data Type• PERIOD(TIME WITH TIME ZONE) Data Type• PERIOD(TIMESTAMP) Data Type

Date	Description
Teradata Database 13.0 (Continued)	<ul style="list-style-type: none"> Added the following new syntax diagrams to Chapter 1: <ul style="list-style-type: none"> PERIOD(TIMESTAMP WITH TIME ZONE) Data Type VARIANT_TYPE UDT Geospatial Data Types Added the following new syntax diagrams to Chapter 2: <ul style="list-style-type: none"> CURRENT_ROLE CURRENT_USER CONTAINS IS UNTIL_CHANGED/IS NOT UNTIL_CHANGED MEETS PRECEDES SUCCEEDS BEGIN END LAST INTERVAL PRIOR NEXT P_INTERSECT LDIFF RDIFF P_NORMALIZE Period Value Constructor Arithmetic Operators Scalar UDF Expression Aggregate UDF Expression NEW VARIANT_TYPE Added the following new syntax diagrams to Chapter 3: <ul style="list-style-type: none"> COLLECT STATISTICS (Recollect Statistics) CREATE GLOP SET CREATE REPLICATION RULESET/REPLACE REPLICATION RULESET DROP GLOP SET DROP REPLICATION RULESET SHOW QUERY LOGGING Added the following new syntax diagrams to Chapter 4: <ul style="list-style-type: none"> GRANT CONNECT THROUGH REVOKE CONNECT THROUGH Added the following new syntax diagrams to Chapter 5: <ul style="list-style-type: none"> HASH BY Clause LOCAL ORDER BY Clause

Date	Description
Teradata Database 13.0 (Continued)	<ul style="list-style-type: none"> Added the following new syntax diagrams to Chapter 7: <ul style="list-style-type: none"> SIGNAL RESIGNAL GET DIAGNOSTICS
Teradata Database 12.0 September 2007	<ul style="list-style-type: none"> Updated existing syntax diagrams in Chapters 2, 3, 4, 5, 6, 7, 8, 9 and 10 Added the following new syntax diagrams to Chapter 1: <ul style="list-style-type: none"> Hexadecimal Name Literals Unicode Character String Literals Unicode Delimited Identifier Added the following new syntax diagrams to Chapter 2: <ul style="list-style-type: none"> DEGREES, RADIANS STRING_CS Modified the syntax diagrams for Window Aggregate Functions in Chapter 2 to support the following aggregate functions: <ul style="list-style-type: none"> CORR COVAR_POP COVAR_SAMP REGR_AVGX REGR_AVGY REGR_COUNT REGR_INTERCEPT REGR_R2 REGR_SLOPE REGR_SXX REGR_SXY REGR_SYY STDDEV_POP STDDEV_SAMP VAR_POP VAR_SAMP Added the following new syntax diagrams to Chapter 3: <ul style="list-style-type: none"> CREATE ERROR TABLE Java Simple and Object Mapped Data Types DROP ERROR TABLE LOGGING ONLINE ARCHIVE OFF LOGGING ONLINE ARCHIVE ON SET QUERY_BAND

Date	Description
Teradata Database 12.0 (Continued)	<ul style="list-style-type: none"> Added the following new syntax diagrams to Chapter 5: <ul style="list-style-type: none"> INITIATE PARTITION ANALYSIS DIAGNOSTIC COSTPRINT DIAGNOSTIC DUMP COSTS DIAGNOSTIC HELP COSTS DIAGNOSTIC SET COSTS DIAGNOSTIC HELP PROFILE DIAGNOSTIC SET PROFILE Added the following new syntax diagrams to Chapter 6: <ul style="list-style-type: none"> HELP COLUMN - Syntaxes 6, 7 and 8 HELP ERROR TABLE

Additional Information

URL	Description
http://www.info.teradata.com/	<p>Use the Teradata Information Products Publishing Library site to:</p> <ul style="list-style-type: none"> View or download a manual: <ol style="list-style-type: none"> Under Online Publications, select General Search. Enter your search criteria and click Search. Download a documentation CD-ROM: <ol style="list-style-type: none"> Under Online Publications, select General Search. In the Title or Keyword field, enter <i>CD-ROM</i>, and click Search. Order printed manuals: <p>Under Print & CD Publications, select How to Order.</p>
http://www.teradata.com	<p>The Teradata home page provides links to numerous sources of information about Teradata. Links include:</p> <ul style="list-style-type: none"> Executive reports, case studies of customer experiences with Teradata, and thought leadership Technical information, solutions, and expert advice Press releases, mentions and media resources
http://teradatauniversitynetwork.com	<p>Teradata University Network fosters education on data warehousing, business intelligence (BI) and database administration (DBA).</p>

To maintain the quality of our products and services, we would like your comments on the accuracy, clarity, organization, and value of this document. Please e-mail: teradata-books@lists.teradata.com

References to Microsoft Windows and Linux

This book refers to “Microsoft Windows” and “Linux.” For Teradata Database 13.0, these references mean:

- “Windows” is Microsoft Windows Server 2003 64-bit.
- “Linux” is SUSE Linux Enterprise Server 9 and SUSE Linux Enterprise Server 10.

Table of Contents

Preface	3
Purpose	3
Audience	3
Supported Software Release.....	3
Prerequisites	3
Changes to This Book.....	3
Additional Information	6
References to Microsoft Windows and Linux	7

Chapter 1: Data Types and Literals

	21
Byte and BLOB Data Types	21
Character and CLOB Data Types	22
Data Literals.....	23
DateTime and Interval Data Types	29
Decimal/Numeric Data Types	32
PERIOD Data Types.....	33
UDT Data Types	34
Geospatial Data Types	35
Default Value Control Phrases.....	37
Output Format Phrases	38

Chapter 2: SQL Functions and Expressions

Aggregate Functions	39
Arithmetic Operators and Functions/Trigonometric/Hyperbolic Functions	42
Trigonometric Functions.....	45
Hyperbolic Functions.....	45
Attribute Functions.....	46
Built-In Functions.....	47

CASE Expressions	49
Comparison Operators	50
Data Type Conversions	51
Byte Conversion	51
Character-to-Character Conversion	52
Character-to-DATE Conversion	53
Character-to-INTERVAL Conversion	53
Character-to-Period Conversion	54
Character-to-Numeric Conversion	54
Character-to-TIME Conversion	54
Character-to-TIMESTAMP Conversion	55
Character-to-UDT Conversion	56
DATE-to-Character Conversion	56
DATE-to-DATE Conversion	57
DATE-to-Numeric Conversion	57
DATE-to-Period Conversion	58
DATE-to-TIMESTAMP Conversion	58
DATE-to-UDT Conversion	59
INTERVAL-to-Character Conversion	59
INTERVAL-to-INTERVAL Conversion	60
INTERVAL-to-Numeric Conversion	61
INTERVAL-to-UDT Conversion	61
Numeric-to-Character Conversion	61
Numeric-to-DATE Conversion	62
Numeric-to-INTERVAL Conversion	62
Numeric-to-Numeric Conversion	63
Numeric-to-UDT Conversion	63
Period-to-Character Conversion	64
Period-to-DATE Conversion	64
Period-to-Period Conversion	64
Period-to-TIME Conversion	65
Period-to-TIMESTAMP Conversion	65
TIME-to-Character Conversion	65
TIME-to-Period Conversion	66
TIME-to-TIME Conversion	66
TIME-to-TIMESTAMP Conversion	67
TIME-to-UDT Conversion	68
TIMESTAMP-to-Character Conversion	68

TIMESTAMP-to-DATE Conversion	69
TIMESTAMP-to-Period Conversion	69
TIMESTAMP-to-TIME Conversion	69
TIMESTAMP-to-TIMESTAMP Conversion	70
TIMESTAMP-to-UDT Conversion	71
UDT-to-Byte Conversion	71
UDT-to-Character Conversion	72
UDT-to-DATE Conversion	72
UDT-to-INTERVAL Conversion	73
UDT-to-Numeric Conversion	73
UDT-to-TIME Conversion	74
UDT-to-TIMESTAMP Conversion	75
UDT-to-UDT Conversion	75
DateTime and Interval Functions and Expressions	75
Hash-Related Functions	78
Logical Predicates	79
Ordered Analytical Functions	83
Period Functions and Operators	87
Set Operators	89
String Operator and Functions	90
UDF Expressions	92
UDT Expressions and Methods	93

Chapter 3: SQL Data Definition Language..... 95

ALTER FUNCTION	95
ALTER METHOD	95
ALTER PROCEDURE (External Form)	96
ALTER PROCEDURE (SQL Form)	96
ALTER REPLICATION GROUP	96
ALTER TABLE	97
ALTER TRIGGER	102
ALTER TYPE	103
BEGIN LOGGING	104
BEGIN QUERY LOGGING	105
COLLECT STATISTICS (Optimizer Form)	106
COLLECT STATISTICS (Alternate Optimizer Form)	107

COLLECT STATISTICS (Recollect Statistics)	107
COMMENT (Comment Placing Form)	108
CREATE AUTHORIZATION/ REPLACE AUTHORIZATION	108
CREATE CAST/ REPLACE CAST	109
CREATE DATABASE	110
CREATE ERROR TABLE	110
CREATE FUNCTION/ REPLACE FUNCTION	111
CREATE FUNCTION (Table Form)	115
CREATE GLOBAL TEMPORARY TRACE TABLE	118
CREATE GLOP SET	121
CREATE HASH INDEX	121
CREATE INDEX	122
CREATE JOIN INDEX	123
CREATE MACRO/ REPLACE MACRO	125
CREATE METHOD	126
CREATE ORDERING/ REPLACE ORDERING	128
CREATE PROCEDURE (External Form)/ REPLACE PROCEDURE (External Form)	129
CREATE PROCEDURE (SQL Form)/ REPLACE PROCEDURE	131
CREATE PROFILE	138
CREATE RECURSIVE VIEW/ REPLACE RECURSIVE VIEW	139
CREATE REPLICATION GROUP	140
CREATE REPLICATION RULESET/ REPLACE REPLICATION RULESET	141
CREATE ROLE	141
CREATE TABLE	142
CREATE TABLE (Queue Table Form)	151
CREATE TRANSFORM/ REPLACE TRANSFORM	156
CREATE TRIGGER/ REPLACE TRIGGER	157
CREATE TYPE (Distinct Form)	158
CREATE TYPE (Structured Form)	160
CREATE USER	164

CREATE VIEW/ REPLACE VIEW	165
DATABASE	166
DELETE DATABASE DELETE USER	166
DROP AUTHORIZATION	166
DROP CAST	166
DROP DATABASE	168
DROP ERROR TABLE	168
DROP FUNCTION	169
DROP GLOB SET	171
DROP HASH INDEX	171
DROP INDEX	172
DROP JOIN INDEX	172
DROP MACRO/ DROP PROCEDURE/ DROP TABLE/ DROP TRIGGER/ DROP VIEW	172
DROP ORDERING	173
DROP PROFILE	173
DROP REPLICATION GROUP	174
DROP REPLICATION RULESET	174
DROP ROLE	174
DROP STATISTICS (Optimizer Form)	175
DROP TRANSFORM	176
DROP TYPE	176
DROP USER	176
END LOGGING	177
END QUERY LOGGING	178
LOGGING ONLINE ARCHIVE OFF	178
LOGGING ONLINE ARCHIVE ON	178
MODIFY DATABASE	179
MODIFY PROFILE	180
MODIFY USER	181
RENAME FUNCTION	182
RENAME MACRO/ RENAME PROCEDURE/ RENAME TABLE/	

RENAME TRIGGER/	
RENAME VIEW.....	184
REPLACE METHOD	185
SET QUERY_BAND.....	187
SET ROLE	188
SET SESSION	188
SET SESSION ACCOUNT.....	188
SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL.....	189
SET SESSION COLLATION	189
SET SESSION DATABASE	189
SET SESSION DATEFORM.....	189
SET SESSION FUNCTION TRACE	190
SET SESSION OVERRIDE REPLICATION.....	190
SET SESSION SUBSCRIBER.....	190
SET TIME ZONE	190
HELP	191
HELP CAST.....	193
HELP COLUMN.....	193
HELP CONSTRAINT.....	194
HELP DATABASE/	
HELP USER	195
HELP ERROR TABLE	195
HELP FUNCTION	195
HELP HASH INDEX	197
HELP INDEX	197
HELP JOIN INDEX	198
HELP MACRO/	
HELP TABLE/	
HELP VIEW	198
HELP METHOD.....	198
HELP PROCEDURE.....	199
HELP REPLICATION GROUP.....	199
HELP SESSION.....	199
HELP STATISTICS (Optimizer Form).....	199
HELP STATISTICS (QCD Form)	200
HELP TRANSFORM	200
HELP TRIGGER	200
HELP TYPE	201
HELP VOLATILE TABLE	201

HELP (Online Form)	201
SHOW	202
SHOW CAST/	
SHOW ERROR TABLE/	
SHOW FUNCTION/	
SHOW HASH INDEX/	
SHOW JOIN INDEX/	
SHOW MACRO/	
SHOW METHOD/	
SHOW PROCEDURE/	
SHOW REPLICATION GROUP/	
SHOW TABLE/	
SHOW TRIGGER/	
SHOW TYPE/	
SHOW VIEW	202
SHOW QUERY LOGGING	208

Chapter 4: SQL Data Control Language 209

GIVE	209
GRANT	209
GRANT CONNECT THROUGH	212
GRANT LOGON	213
REVOKE	213
REVOKE CONNECT THROUGH	216
REVOKE LOGON	217

Chapter 5: SQL Data Manipulation Language 219

SELECT	220
SELECT AND CONSUME	223
WITH [RECURSIVE] Request Modifier	224
DISTINCT, ALL, and .ALL Options	226
TOP <i>n</i> Operator	227
FROM Clause	227
HASH BY Clause	228
LOCAL ORDER BY Clause	228
WHERE Clause	228
Subqueries in Search Conditions	228

GROUP BY Clause	229
HAVING Clause	229
QUALIFY Clause.....	230
SAMPLE Clause	230
SAMPLEID Expression.....	230
ORDER BY Clause	231
WITH Clause.....	231
Outer Join	231
Null.....	231
ABORT.....	232
BEGIN TRANSACTION	232
CALL	232
CHECKPOINT	233
COMMENT (Comment-Retrieving Form)	233
COMMIT.....	233
DELETE.....	234
ECHO.....	235
END TRANSACTION	235
EXECUTE	235
INSERT/INSERT ... SELECT	235
LOCKING Request Modifier.....	236
MERGE	237
ROLLBACK.....	238
UPDATE	238
USING Request Modifier.....	239
COLLECT DEMOGRAPHICS	240
COLLECT STATISTICS (QCD Form)	240
DROP STATISTICS (QCD Form)	241
DUMP EXPLAIN	241
EXPLAIN Request Modifier.....	241
INITIATE INDEX ANALYSIS.....	242
INITIATE PARTITION ANALYSIS	242
INSERT EXPLAIN	243
RESTART INDEX ANALYSIS.....	243
DIAGNOSTIC COSTPRINT.....	244
DIAGNOSTIC DUMP COSTS	244
DIAGNOSTIC HELP COSTS	244
DIAGNOSTIC SET COSTS.....	245

DIAGNOSTIC HELP PROFILE	245
DIAGNOSTIC SET PROFILE	245
DIAGNOSTIC DUMP SAMPLES	246
DIAGNOSTIC HELP SAMPLES	246
DIAGNOSTIC SET SAMPLES	246
DIAGNOSTIC “Validate Index”	247

Chapter 6: SQL Cursor Control 249

CLOSE	249
DECLARE CURSOR	249
DELETE	250
FETCH	251
OPEN	252
POSITION	252
PREPARE	252
REWIND	253
SELECT ... INTO	253
SELECT AND CONSUME ... INTO	254
UPDATE (Positioned Form)	254

Chapter 7: SQL Stored Procedures: Control Statements and Condition Handling 255

BEGIN - END Statement	255
CASE	255
DECLARE	260
FOR	260
IF	265
ITERATE	269
LEAVE	269
LOOP	269
REPEAT	273
SET	274
WHILE	274
DECLARE CONDITION	278
DECLARE HANDLER (Basic Syntax)	279

SIGNAL	279
RESIGNAL.....	279
GET DIAGNOSTICS	280

Chapter 8: Static Embedded SQL Statements281

BEGIN DECLARE SECTION	281
COMMENT.....	281
DATABASE	281
DECLARE STATEMENT.....	282
DECLARE TABLE.....	282
END DECLARE SECTION	282
END-EXEC Statement Terminator.....	282
EXEC	283
EXEC SQL Statement Prefix	283
INCLUDE	283
INCLUDE SQLCA	283
INCLUDE SQLDA	284
WHENEVER.....	284

Chapter 9: Dynamic Embedded SQL Statements.....285

DESCRIBE.....	285
EXECUTE	285
EXECUTE IMMEDIATE.....	286
PREPARE.....	286

Chapter 10: SQL Client-Server Connectivity Statements ...287

CONNECT	287
GET CRASH	287
LOGOFF	287
LOGON	288
SET BUFFERSIZE.....	288
SET CHARSET	288
SET CONNECTION.....	288

SET CRASH.....	288
SET ENCRYPTION	289

Chapter 11: Multisession Asynchronous Programming With Embedded SQL..... 291

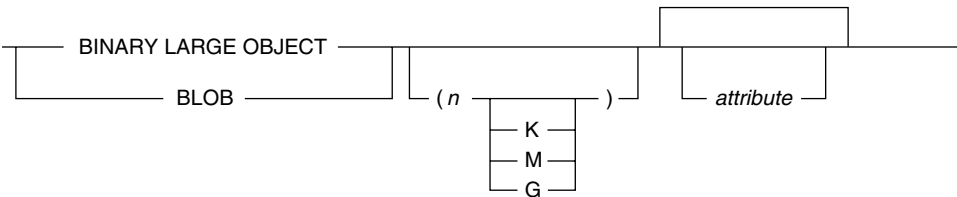
ASYNC Statement Modifier.....	291
TEST.....	291
WAIT	291

Appendix A: How to Read Syntax Diagrams 293

Syntax Diagram Conventions	293
----------------------------------	-----

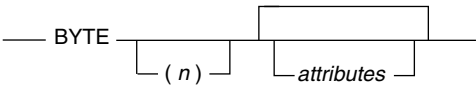
Byte and BLOB Data Types

BLOB Data Type



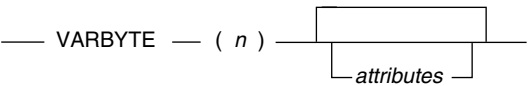
1101B095

BYTE Data Type



1101A430

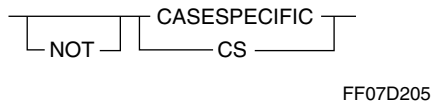
VARBYTE Data Type



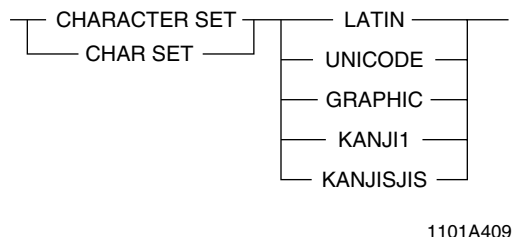
1101E171

Character and CLOB Data Types

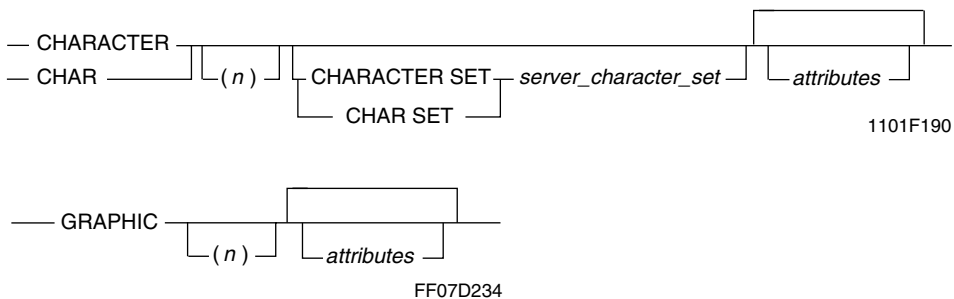
CASESPECIFIC Phrase



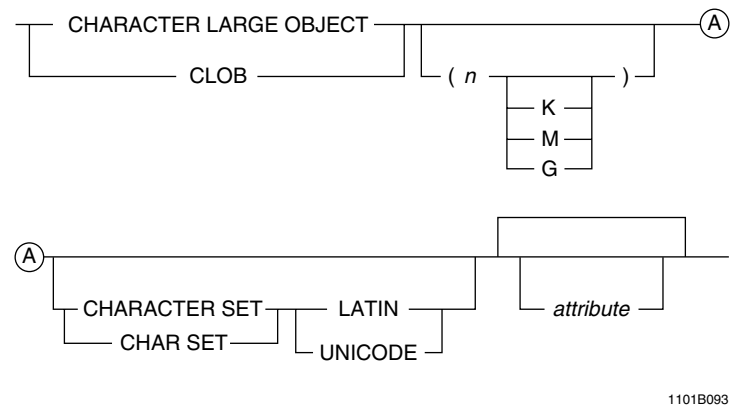
CHARACTER SET Phrase



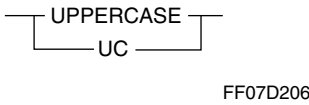
CHARACTER Data Type



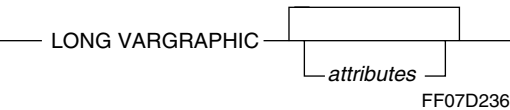
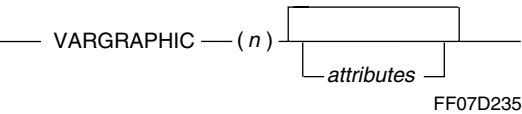
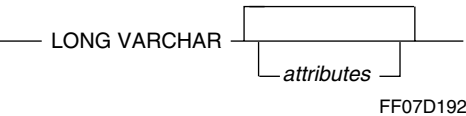
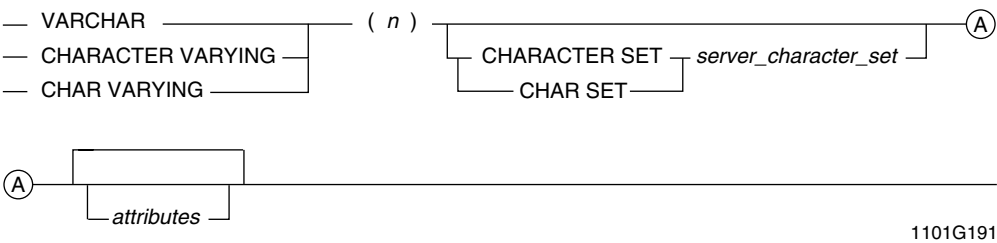
CLOB Data Type



UPPERCASE Phrase

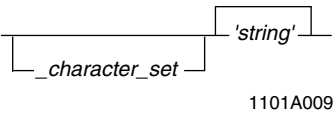


VARCHAR Data Type



Data Literals

CHARACTER String Literals

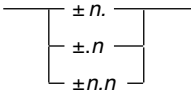


DATE Literals

DATE 'string'

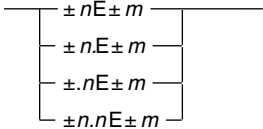
1101A036

DECIMAL Literals



KR01A143

FLOATING POINT Literals



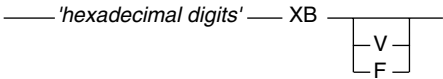
KR01A142

GRAPHIC Literals

G '< ABC >'

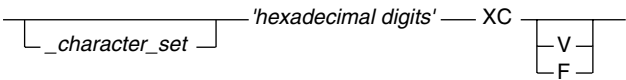
HH01A090

Hexadecimal Byte Literals



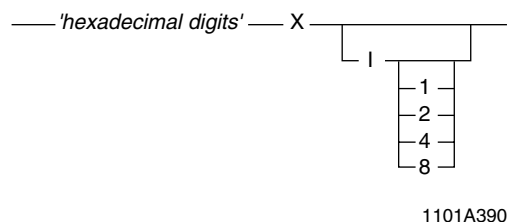
1101A391

Hexadecimal Character Literals

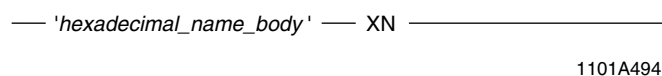


1101A392

Hexadecimal Integer Literals



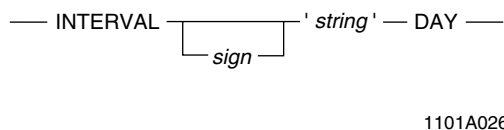
Hexadecimal Name Literals



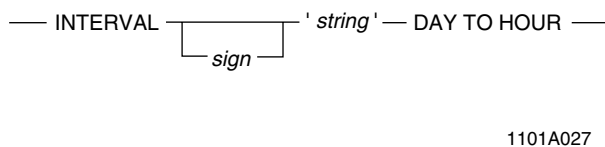
INTEGER Literals



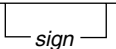
INTERVAL DAY Literals



INTERVAL DAY TO HOUR Literals

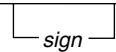


INTERVAL DAY TO MINUTE Literals

— INTERVAL  'string' — DAY TO MINUTE —

1101A028

INTERVAL DAY TO SECOND Literals

— INTERVAL  'string' — DAY TO SECOND —

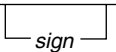
1101A029

INTERVAL HOUR Literals

— INTERVAL  'string' — HOUR —

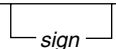
1101A030

INTERVAL HOUR TO MINUTE Literals

— INTERVAL  'string' — HOUR TO MINUTE —

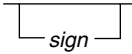
1101A037

INTERVAL HOUR TO SECOND Literals

— INTERVAL  'string' — HOUR TO SECOND —

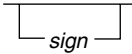
1101A038

INTERVAL MINUTE Literals

— INTERVAL  ' *string* ' — MINUTE —

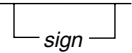
1101A031

INTERVAL MINUTE TO SECOND Literals

— INTERVAL  ' *string* ' — MINUTE TO SECOND —

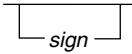
1101A032

INTERVAL MONTH Literals

— INTERVAL  ' *string* ' — MONTH —

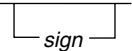
1101A025

INTERVAL SECOND Literals

— INTERVAL  ' *string* ' — SECOND —

1101A033

INTERVAL YEAR Literals

— INTERVAL  ' *string* ' — YEAR —

1101A023

INTERVAL YEAR TO MONTH Literals

— INTERVAL sign 'string' — YEAR TO MONTH —

1101A024

Period Literals

— PERIOD — (beginning_bound - ending_bound) —
, UNTIL_CHANGED

1101A591

Time Literals

— TIME — 'string' —

1101A021

Timestamp Literals

— TIMESTAMP — 'string' —

1101A022

Unicode Character String Literals

_character_set U& 'Unicode_string_body' U& '' 'Unicode_string_body' '' UESCAPE 'Unicode_esc_char' (A)

1101A493

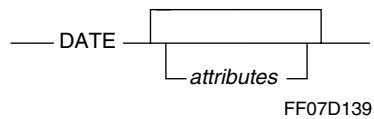
Unicode Delimited Identifier

— U&" *Unicode_delimiter_body*" — UESCAPE —'*Unicode_esc_char*'—

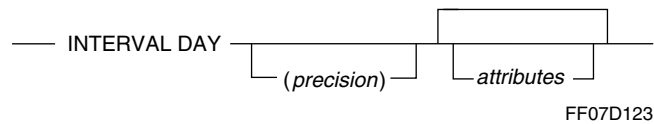
1101A495

DateTime and Interval Data Types

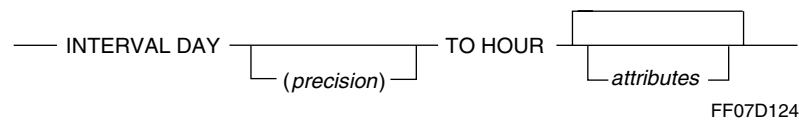
DATE Data Type



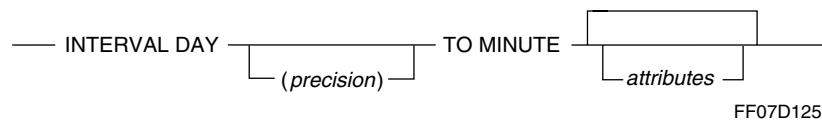
INTERVAL DAY Data Type



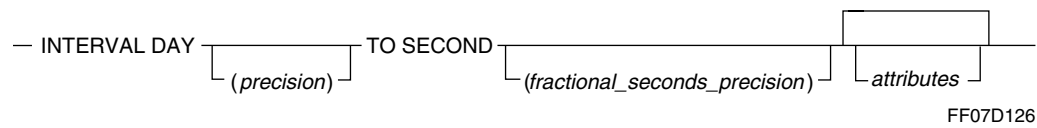
INTERVAL DAY TO HOUR Data Type



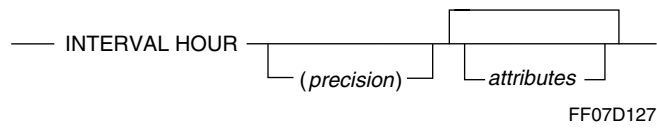
INTERVAL DAY TO MINUTE Data Type



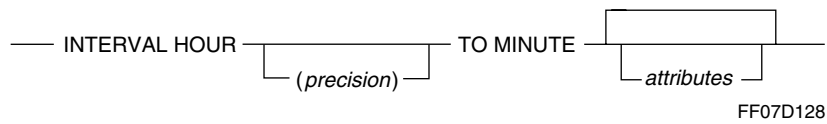
INTERVAL DAY TO SECOND Data Type



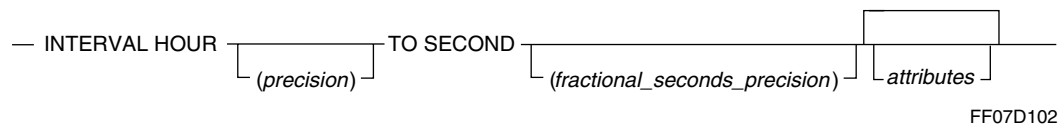
INTERVAL HOUR Data Type



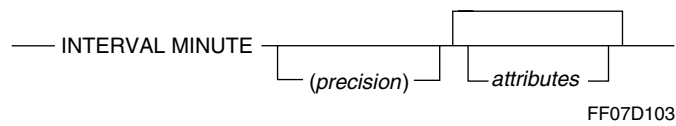
INTERVAL HOUR TO MINUTE Data Type



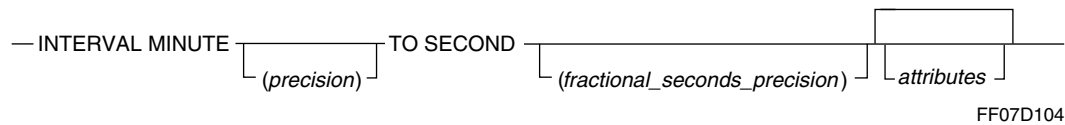
INTERVAL HOUR TO SECOND Data Type



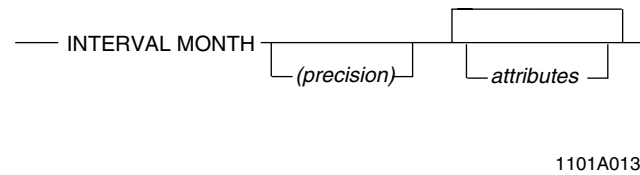
INTERVAL MINUTE Data Type



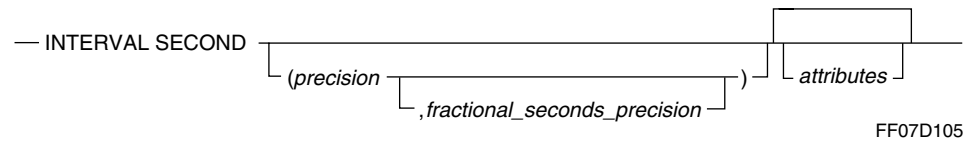
INTERVAL MINUTE TO SECOND Data Type



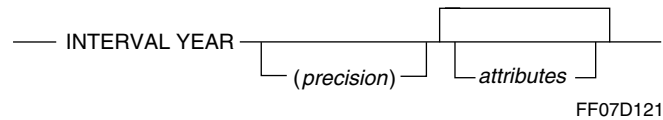
INTERVAL MONTH Data Type



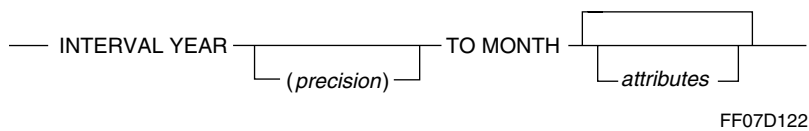
INTERVAL SECOND Data Type



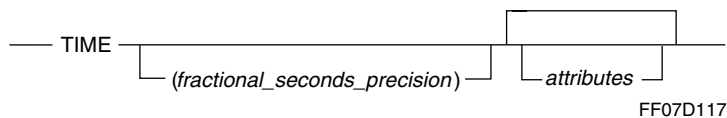
INTERVAL YEAR Data Type



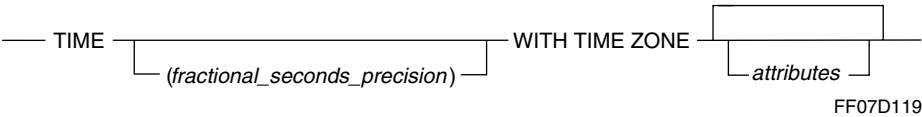
INTERVAL YEAR TO MONTH Data Type



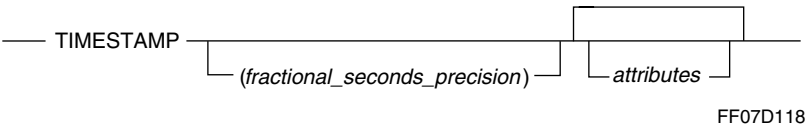
TIME Data Type



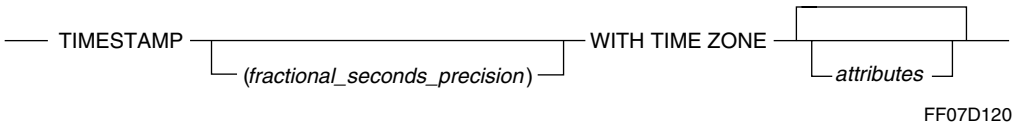
TIME WITH TIME ZONE Data Type



TIMESTAMP Data Type

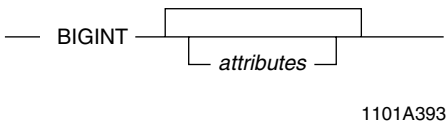


TIMESTAMP WITH TIME ZONE Data Type

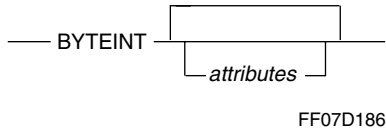


Decimal/Numeric Data Types

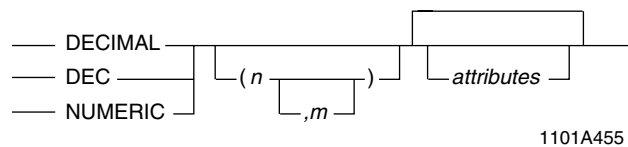
BIGINT Data Type



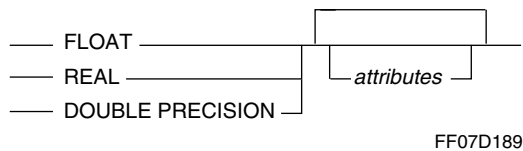
BYTEINT Data Type



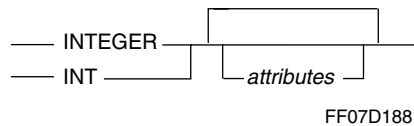
DECIMAL/NUMERIC Data Types



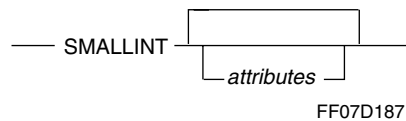
FLOAT/REAL/DOUBLE PRECISION Data Types



INTEGER Data Type

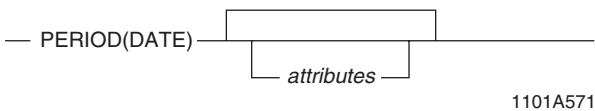


SMALLINT Data Type

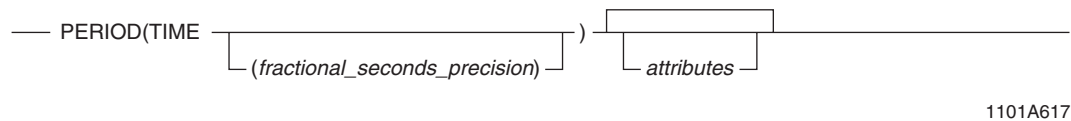


PERIOD Data Types

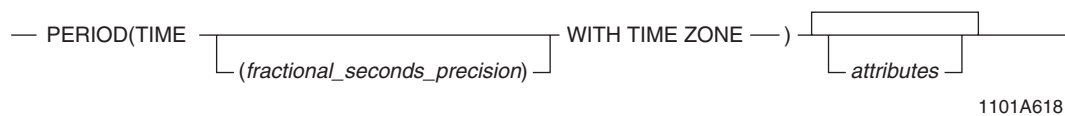
PERIOD(DATE) Data Type



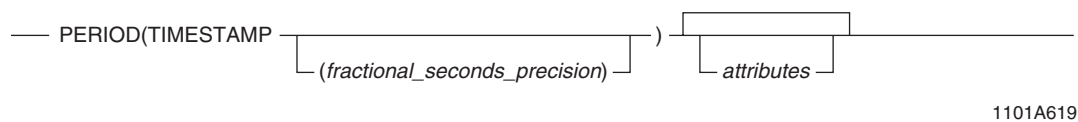
PERIOD(TIME) Data Type



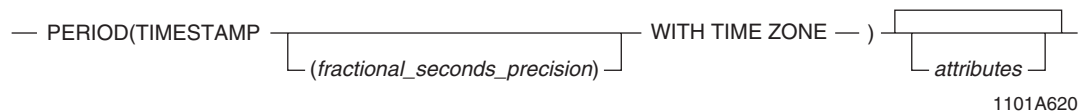
PERIOD(TIME WITH TIME ZONE) Data Type



PERIOD(TIMESTAMP) Data Type

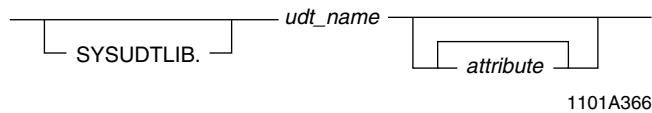


PERIOD(TIMESTAMP WITH TIME ZONE) Data Type

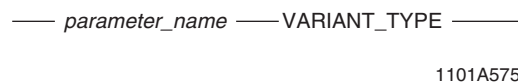


UDT Data Types

UDT Data Type

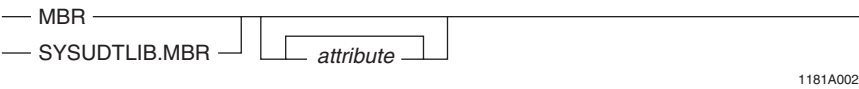


VARIANT_TYPE UDT

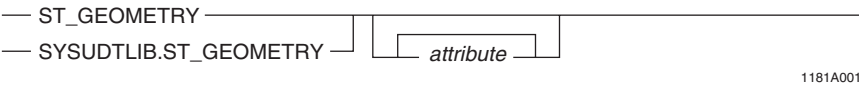


Geospatial Data Types

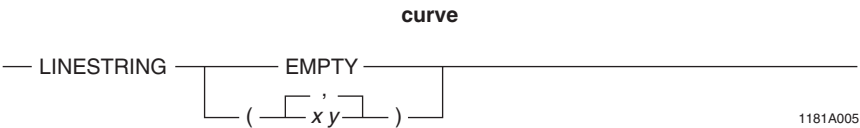
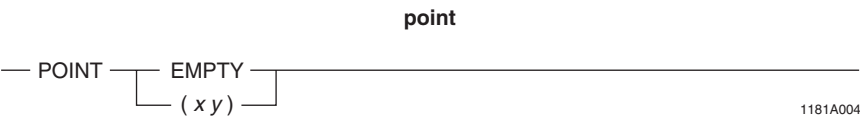
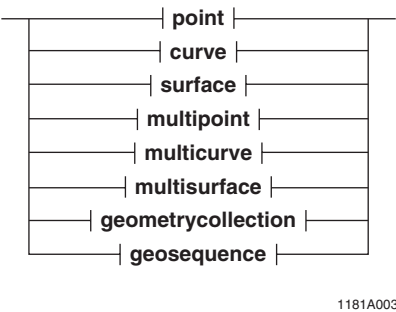
MBR Type



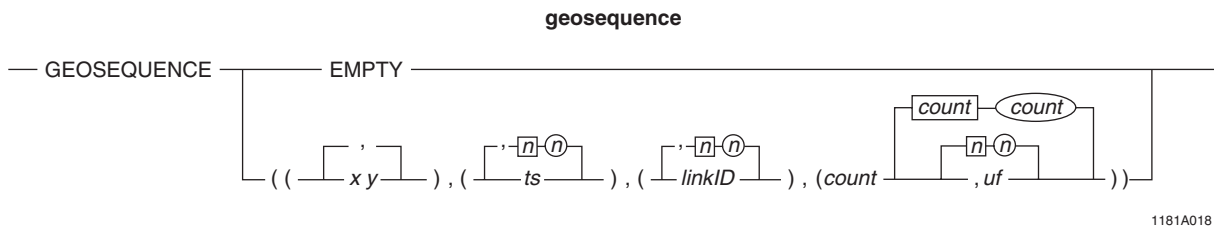
ST_Geometry Type



Well-Known Text Representation

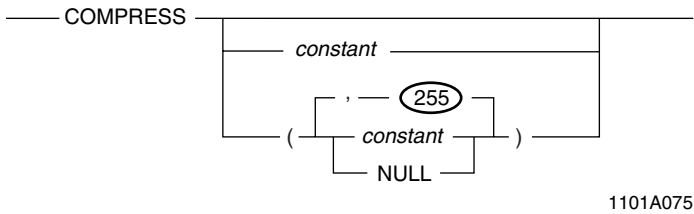




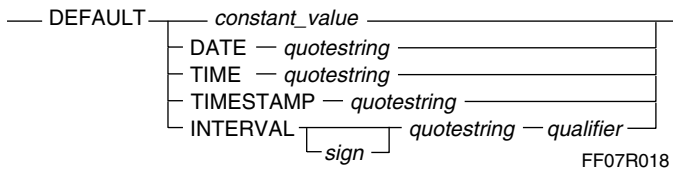


Default Value Control Phrases

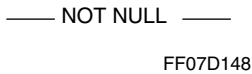
COMPRESS Phrase



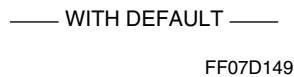
DEFAULT Phrase



NOT NULL Phrase

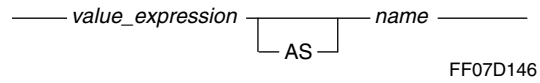


WITH DEFAULT Phrase

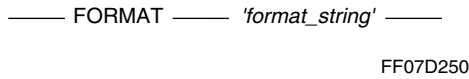


Output Format Phrases

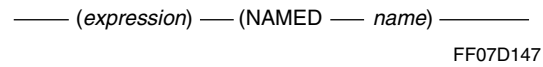
AS



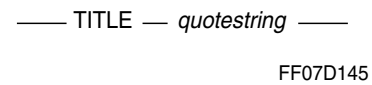
FORMAT



NAMED



TITLE



CHAPTER 2 SQL Functions and Expressions

Aggregate Functions

AVG

— AVERAGE — (— DISTINCT — value_expression) —
— AVG —
— AVE —
— ALL —

1101B410

CORR

— CORR — (value_expression_1, value_expression_2) —

1101B217

COUNT

— COUNT — (— DISTINCT — value_expression —) —
— ALL —
— * —

1101A411

COVAR_POP

— COVAR_POP — (value_expression_1, value_expression_2) —

1101B216

COVAR_SAMP

— COVAR_SAMP — (value_expression_1, value_expression_2) —

1101A456

GROUPING

— GROUPING — (expression) —

1101A461

KURTOSIS

— KURTOSIS — ((DISTINCT | ALL) *value_expression*) —
1101B215

MAX

— MAXIMUM — ((DISTINCT | ALL) *value_expression*) —
— MAX —
1101B412

MIN

— MINIMUM — ((DISTINCT | ALL) *value_expression*) —
— MIN —
1101B413

REGR_AVGX

—— REGR_AVGX — (*dependent_variable_expression*, *independent_variable_expression*) ——
1101B414

REGR_AVGY

—— REGR_AVGY — (*dependent_variable_expression*, *independent_variable_expression*) ——
1101B415

REGR_COUNT

— REGR_COUNT — (*dependent_variable_expression*, *independent_variable_expression*) ——
1101B416

REGR_INTERCEPT

—— REGR_INTERCEPT —— (*dependent_variable_expression*, *independent_variable_expression*) ——
1101B417

REGR_R2

—— REGR_R2 —— (*dependent_variable_expression*, *independent_variable_expression*) ——
1101B418

REGR_SLOPE

—— REGR_SLOPE —— (*dependent_variable_expression*, *independent_variable_expression*) ——
1101B419

REGR_SXX

—— REGR_SXX —— (*dependent_variable_expression*, *independent_variable_expression*) ——
1101B420

REGR_SXY

—— REGR_SXY —— (*dependent_variable_expression*, *independent_variable_expression*) ——
1101B421

REGR_SYY

—— REGR_SYY —— (*dependent_variable_expression*, *independent_variable_expression*) ——
1101B422

SKEW

—— SKEW —— (

DISTINCT
<u>ALL</u>

value_expression) ——
1101B428

STDDEV_POP

—— STDDEV_POP —— (

DISTINCT
<u>ALL</u>

value_expression) ——
1101B424

STDDEV_SAMP

—— STDDEV_SAMP —— (

DISTINCT
<u>ALL</u>

value_expression) ——
1101B425

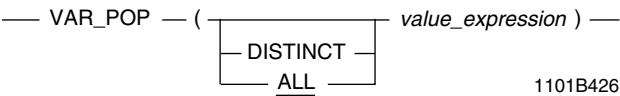
SUM

—— SUM —— (

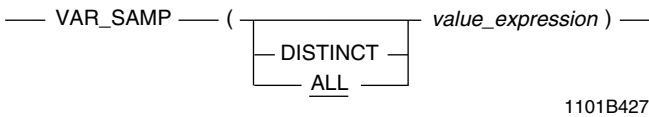
DISTINCT
<u>ALL</u>

value_expression) ——
1101B423

VAR_POP



VAR_SAMP



Arithmetic Operators and Functions/
Trigonometric/Hyperbolic Functions

Arithmetic Operators

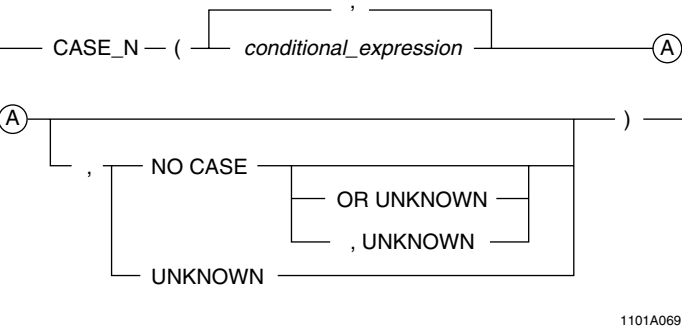
Teradata Database supports the following arithmetic operators:

Operator	Function
**	Exponentiate This is a Teradata extension to the ANSI SQL-99 standard.
*	Multiply
/	Divide
MOD	Modulo (remainder). MOD calculates the remainder in a division operation. For example, 60 MOD 7 = 4: 60 divided by 7 equals 8, with a remainder of 4. The result takes the sign of the dividend, thus: -17 MOD 4 = -1 -17 MOD -4 = -1 17 MOD -4 = 1 17 MOD 4 = 1 This is a Teradata extension to the ANSI SQL-99 standard.
+	Add
-	Subtract
+	Unary plus (positive value)
-	Unary minus (negative value)

ABS

— ABS — (*arg*) —
1101A480

CASE_N



**DEGREES/
RADIANS**

— DEGREES — (*arg*) —
— RADIANS —
1101A481

EXP

— EXP — (*arg*) —
1101A484

LN

— LN — (*arg*) —
1101A485

LOG

— LOG — (*arg*) —
1101A486

NULLIFZERO

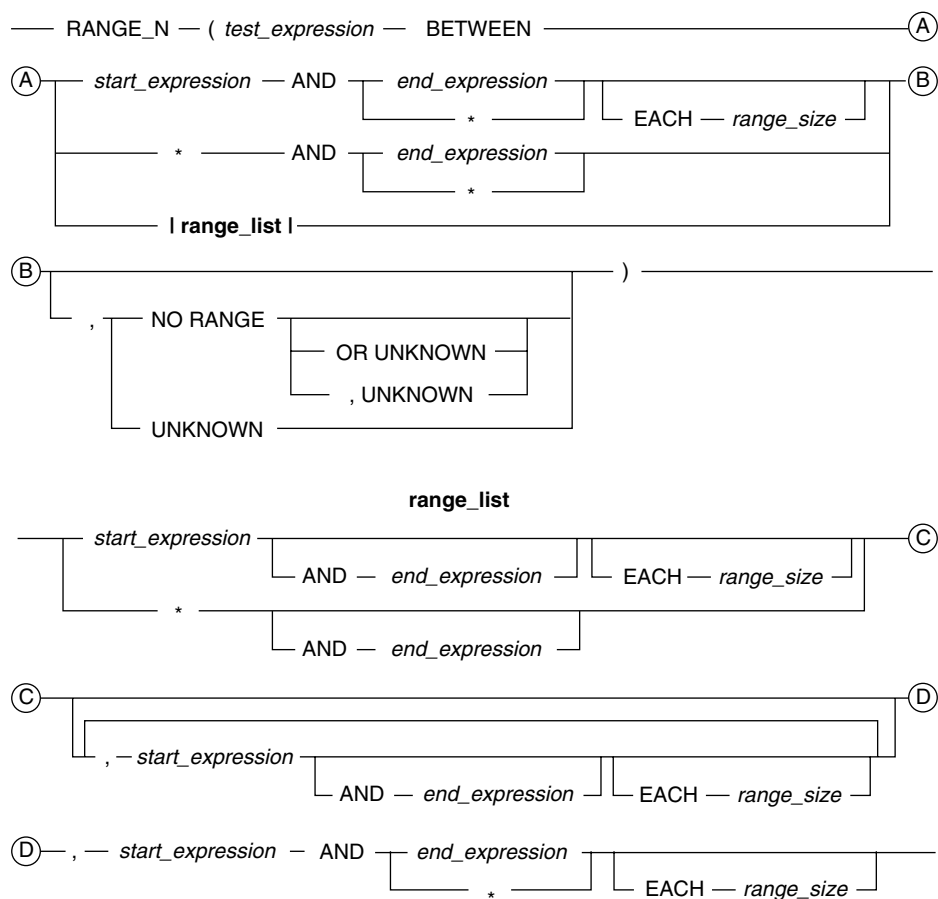
— NULLIFZERO — (*arg*) —
1101F225

RANDOM

— RANDOM — (*lower_bound*, *upper_bound*) —

1101C025

RANGE_N



1101B068

SQRT

— SQRT — (*arg*) —

1101A487

WIDTH_BUCKET

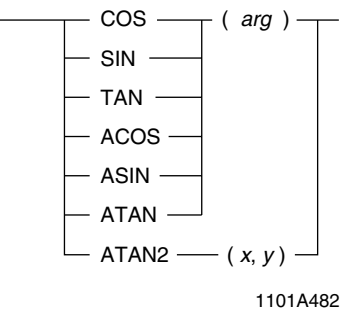
— WIDTH_BUCKET — (*value_expression*, *lower_bound*, *upper_bound*, *partition_count*) —
1101A492

ZEROIFNULL

— ZEROIFNULL — (*arg*) —
1101F226

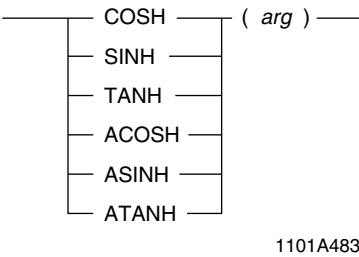
Trigonometric Functions

COS, SIN, TAN, ACOS, ASIN, ATAN, ATAN2



Hyperbolic Functions

COSH, SINH, TANH, ACOSH, ASINH, ATANH



Attribute Functions

BYTES

— BYTE — (*byte_expression*) —
— BYTES —
1101F174

CHARACTERS

— CHARACTERS — (*string_expression*) —
— CHARS —
— CHAR —
1101A488

CHARACTER_LENGTH

— CHARACTER_LENGTH — (*string_expression*) —
— CHAR_LENGTH —
FF07D088

DEFAULT

— DEFAULT — (*column_name*) —
1101A394

FORMAT

— FORMAT — (*column_name*) —
1101A489

OCTET_LENGTH

— OCTET_LENGTH — (— *string_expression* — (— *character_set_name*) —) —
1101A513

TITLE

— TITLE — (*expression*) —
1101B039

TYPE

— TYPE —(*expression*) —
1101A491

Built-In Functions

ACCOUNT

—— ACCOUNT ——
FF07R001

CURRENT_DATE

—— CURRENT_DATE ——
FF07D135

CURRENT_ROLE

—— CURRENT_ROLE ——
1101A565

CURRENT_TIME

—— CURRENT_TIME ————
└── (fractional_precision) ──┘
FF07D136

CURRENT_TIMESTAMP

—— CURRENT_TIMESTAMP ————
└── (fractional_precision) ──┘
FF07D137

CURRENT_USER

—— CURRENT_USER ——
1101A564

DATABASE

_____ DATABASE _____
FF07R002

DATE

_____ DATE _____
FF07D134

PROFILE

_____ PROFILE _____
KZ01A006

ROLE

_____ ROLE _____
KZ01A007

SESSION

_____ SESSION _____
FF07R003

TIME

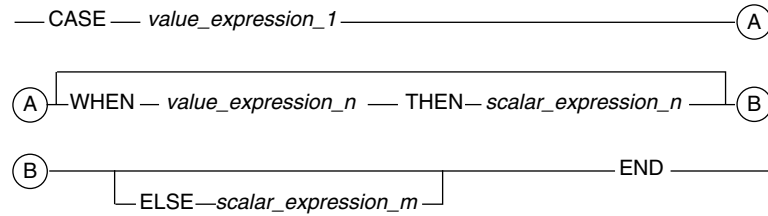
_____ TIME _____
FF07D271

USER

_____ USER _____
FF07D272

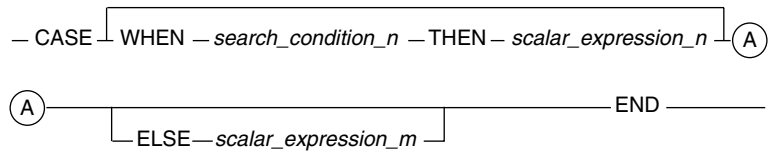
CASE Expressions

Valued CASE Expression



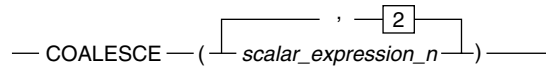
1101A012

Searched CASE Expression



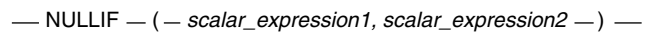
FF07D224

COALESCE Expression



1101E227

NULLIF Expression



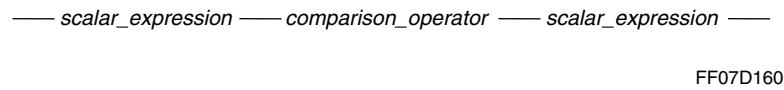
HH01B094

Comparison Operators

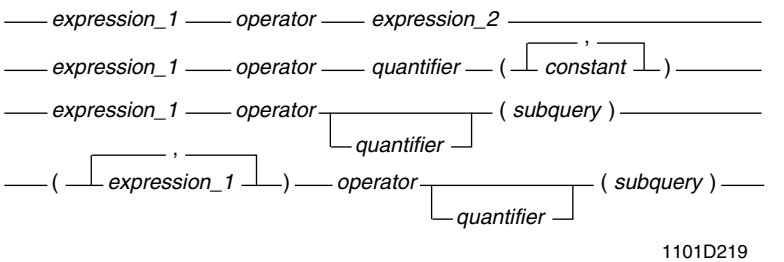
Teradata Database supports the following comparison operators:

ANSI Operator	Teradata Extensions	Function
=	EQ	Tests for equality.
<>	^= NE NOT=	Tests for inequality.
<	LT	Tests for less than.
<=	LE	Tests for less than or equal.
>	GT	Tests for greater than.
>=	GE	Tests for greater than or equal.

Comparison Operators

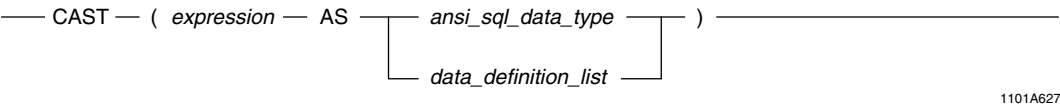


Comparison Operators in Logical Expressions

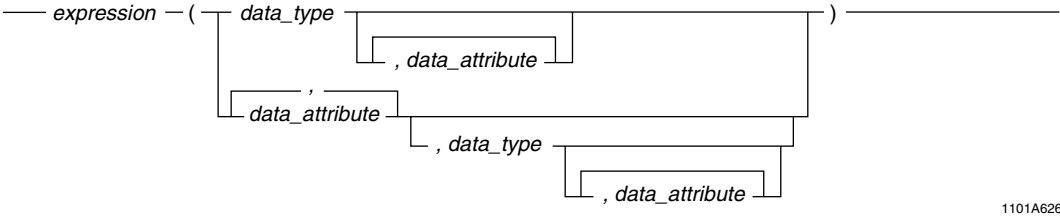


Data Type Conversions

CAST

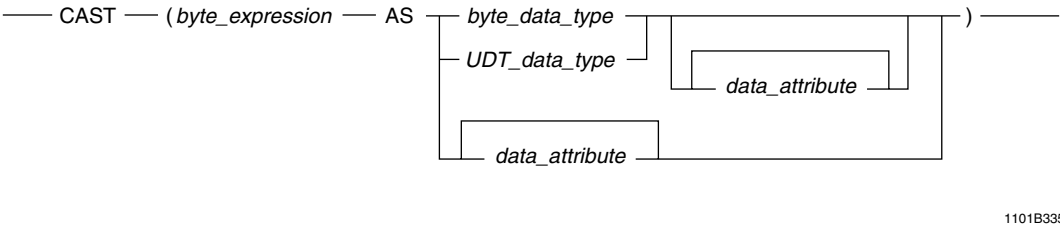


Teradata Conversion Syntax

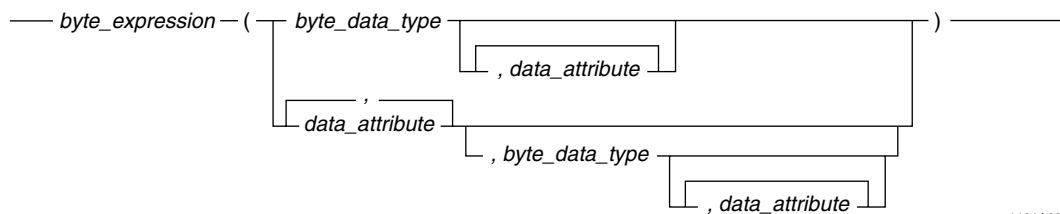


Byte Conversion

CAST



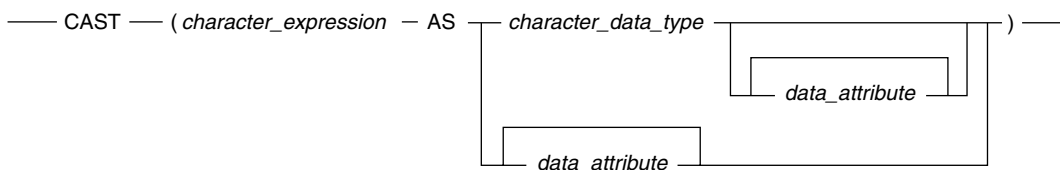
Teradata Conversion



1101A623

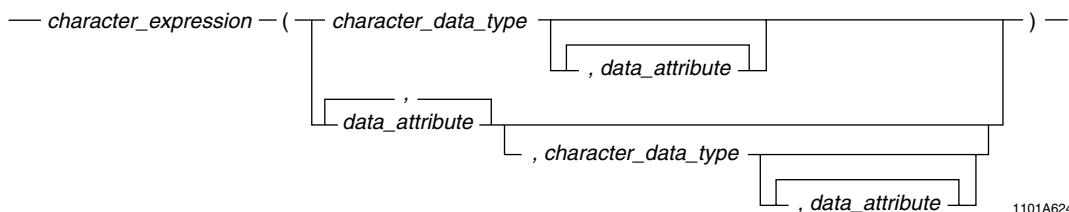
Character-to-Character Conversion

CAST



1101A625

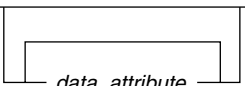
Teradata Conversion



1101A624

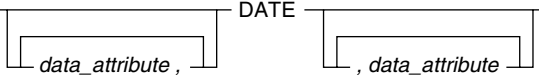
Character-to-DATE Conversion

CAST

— CAST — (*character_expression* — AS — DATE —) —


1101B244

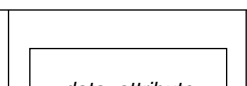
Teradata Conversion

— *character_expression* — (— DATE —) —


1101B255

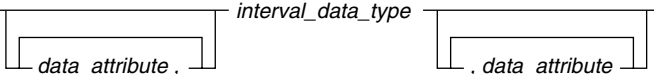
Character-to-INTERVAL Conversion

CAST

— CAST — (*character_expression* — AS — *interval_data_type* —) —


1101B245

Teradata Conversion

— *character_expression* — (— *interval_data_type* —) —


1101B256

Character-to-Period Conversion

CAST

— CAST — (*character_expression* — AS — *period_data_type* —) —
data_attribute

1101A587

Character-to-Numeric Conversion

CAST

— CAST — (*character_expression* — AS — *numeric_data_type* —) —
data_attribute

1101A628

Teradata

— *character_expression* — (—
data_attribute , *numeric_data_type* , data_attribute) —

1101A629

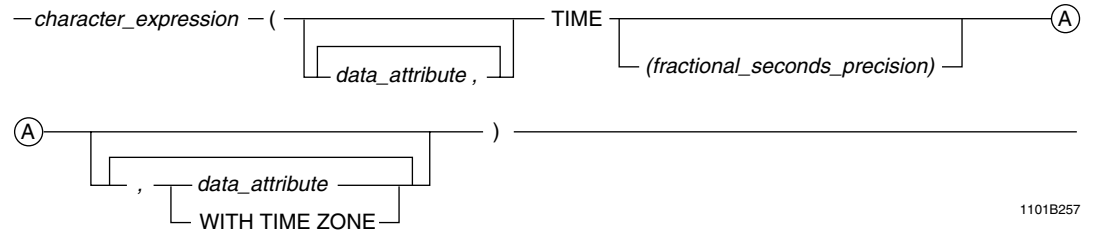
Character-to-TIME Conversion

CAST

— CAST — (— *character_expression* — AS —
(fractional_seconds_precision) —
WITH TIME ZONE —
time_data_attribute) —

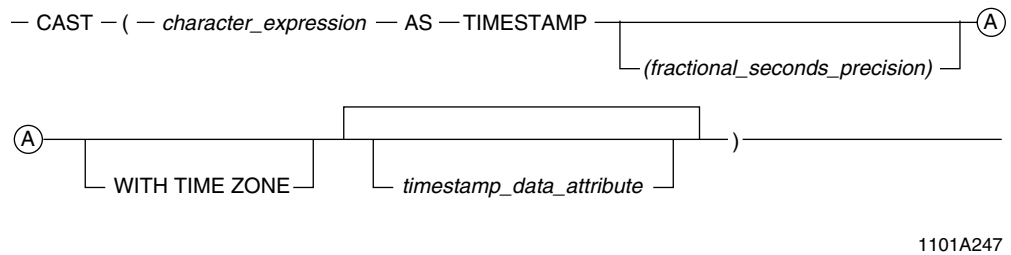
1101A246

Teradata Conversion

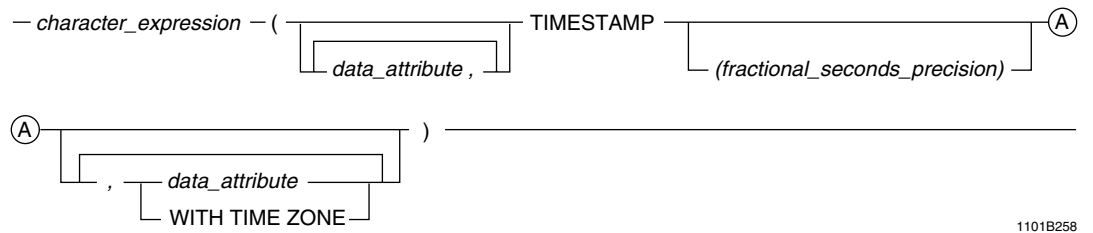


Character-to-TIMESTAMP Conversion

CAST



Teradata Conversion



Character-to-UDT Conversion

CAST

— CAST — (*character_expression* — AS — *UDT_data_definition*) —

1101A336

DATE-to-Character Conversion

CAST

— CAST — (— *date_expression* — AS — *character_data_type* ————— (A)

Diagram illustrating the structure of a character set definition:

A horizontal line represents the definition. A circled 'A' is at the start. Below the line, the text "CHARACTER SET — *server_character_set*" is shown. To the right, a rectangular box is shown, with the text "*character_data_attribute*" below it. The line ends with a closing parenthesis ")".

1101A248

Teradata Conversion

$\text{--data_expression} - (\text{data_attribute . } \text{character_data_type})$ ————— (A)

(A)

data_attribute

CHARACTER SET — server_character_set

1101B259

DATE-to-DATE Conversion

CAST

Diagram illustrating the syntax for the `CAST` function:

```

CAST( ( date_expression AS DATE ) )

```

The diagram shows the components of the `CAST` function:

- `CAST`: The function name.
- `(date_expression AS DATE)`: The expression being cast, followed by the target data type `DATE`.
- `date_data_attribute`: An annotation indicating the data attribute for the `date_expression`.

1101A249

Teradata Conversion

— *date_expression* — (DATE , *data_attribute*) —

data_attribute , DATE

, data_attribute

1101B26

1101B260

DATE-to-Numeric Conversion

CAST

$$- \text{CAST}(-\text{date_expression} \text{ AS } \text{numeric_data_type} \text{ } \boxed{\text{numeric_data_attribute}}) -$$

1101A250

Teradata Conversion

$$- \text{date_expression} - (\overbrace{\text{data_attribute},}^{\text{numeric_data_type}} \overbrace{\text{data_attribute}}^{\text{numeric_data_type}}) -$$

1101B261

DATE-to-Period Conversion

CAST

— CAST — (— *date_expression* — AS — *period_data_type* — *period_data_attribute* —) —

1101A589

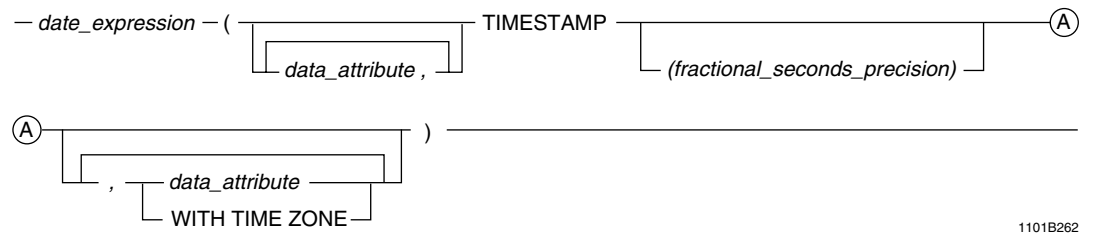
DATE-to-TIMESTAMP Conversion

CAST

— CAST — (— *date_expression* — AS — TIMESTAMP — A —) —
(fractional_seconds_precision) —
(A) —
WITH TIME ZONE —
timestamp_data_attribute —) —

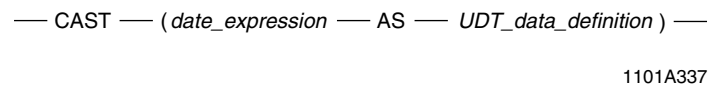
1101A251

Teradata Conversion



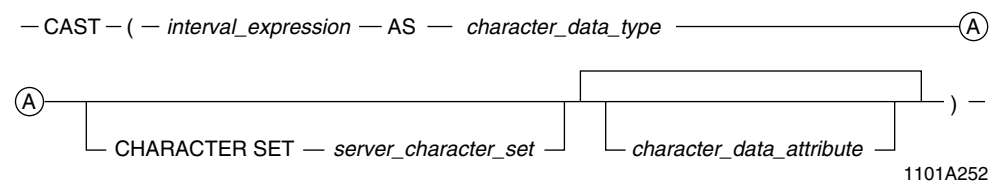
DATE-to-UDT Conversion

CAST

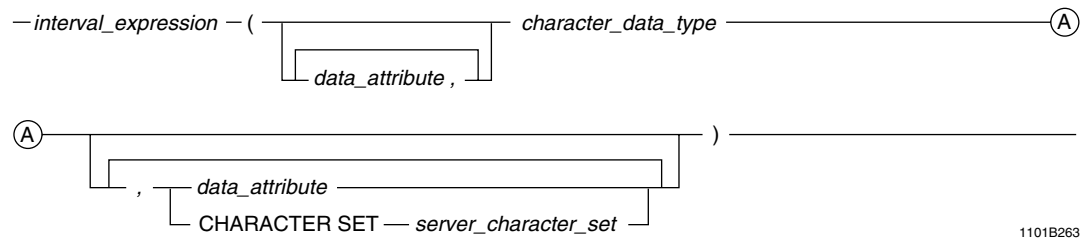


INTERVAL-to-Character Conversion

CAST

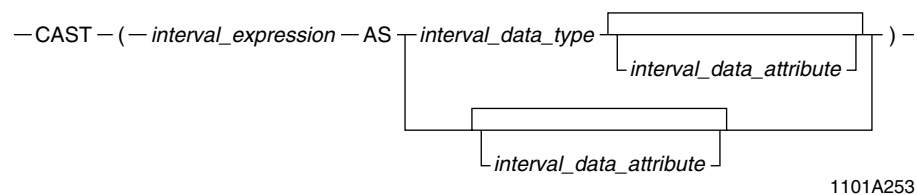


Teradata Conversion

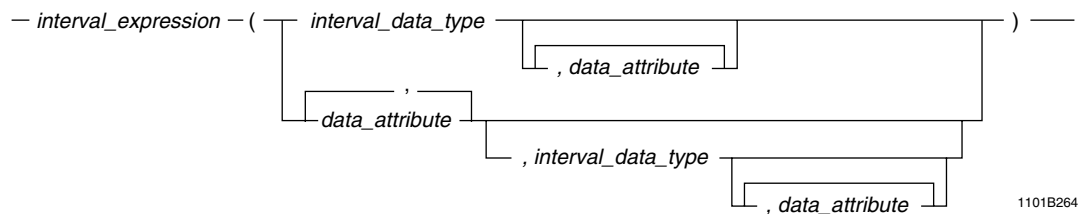


INTERVAL-to-INTERVAL Conversion

CAST

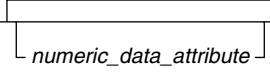


Teradata Conversion

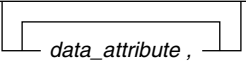
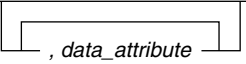


INTERVAL-to-Numeric Conversion

CAST

— CAST — (— *interval_expression* — AS — *numeric_data_type* — ) —
1101A254

Teradata Conversion

— *interval_expression* — ( *numeric_data_type* ) —
1101B265

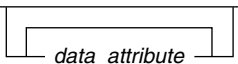
INTERVAL-to-UDT Conversion

CAST

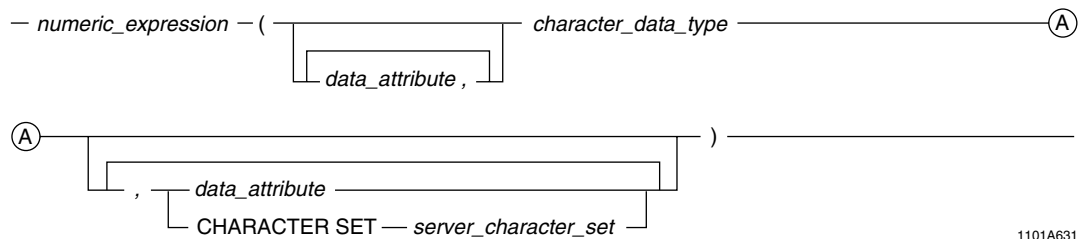
— CAST — (*interval_expression* — AS — *UDT_data_definition*) —
1101A338

Numeric-to-Character Conversion

CAST

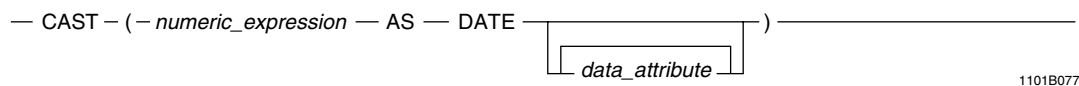
— CAST — (*numeric_expression* — AS — *character_data_type* — ) —
1101A630

Teradata Conversion

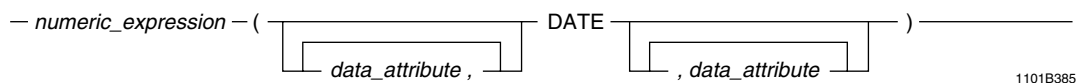


Numeric-to-DATE Conversion

CAST

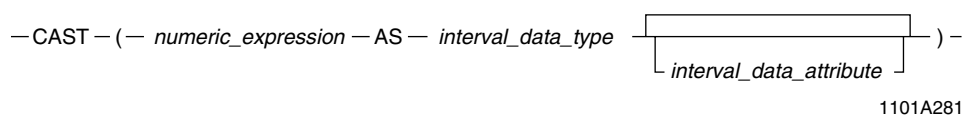


Teradata Conversion



Numeric-to-INTERVAL Conversion

CAST



Teradata Conversion

— *numeric_expression* — (— *interval_data_type* — *data_attribute* , — *data_attribute* —) —

1101B273

Numeric-to-Numeric Conversion

CAST

— CAST — (— *numeric_expression* — AS — *numeric_data_type* — *numeric_data_attribute* —) —

1101A632

Teradata Conversion

— *numeric_expression* — (— *numeric_data_type* — *data_attribute* , — *data_attribute* —) —

1101A633

Numeric-to-UDT Conversion

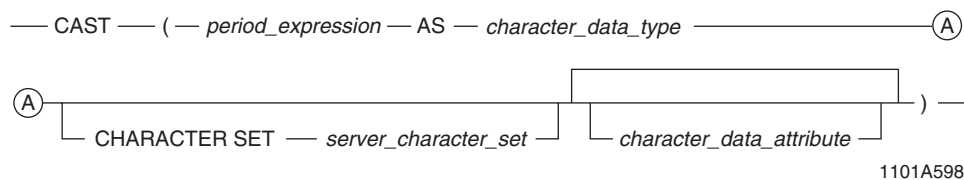
CAST

— CAST — (*numeric_expression* — AS — *UDT_data_definition*) —

1101A334

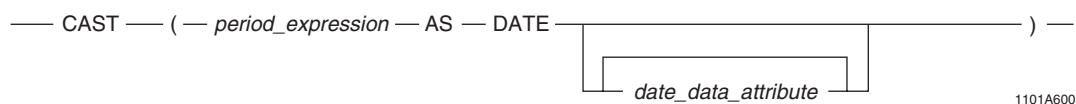
Period-to-Character Conversion

CAST



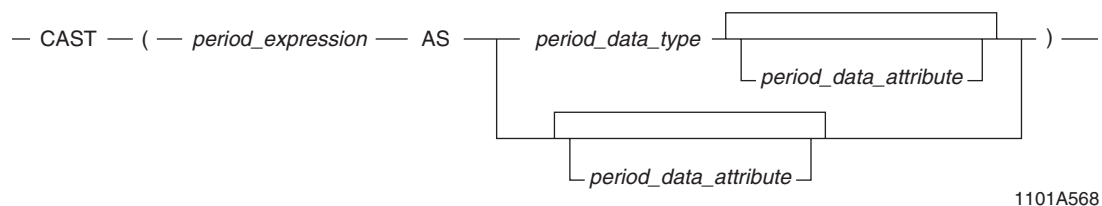
Period-to-DATE Conversion

CAST



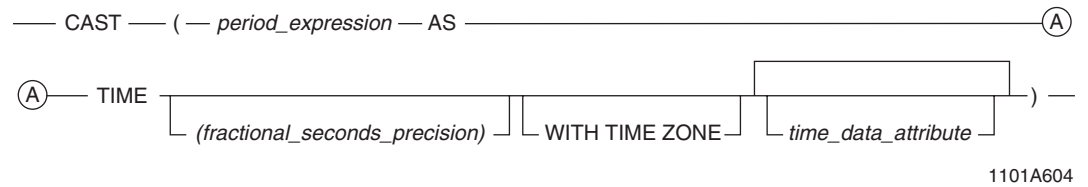
Period-to-Period Conversion

CAST



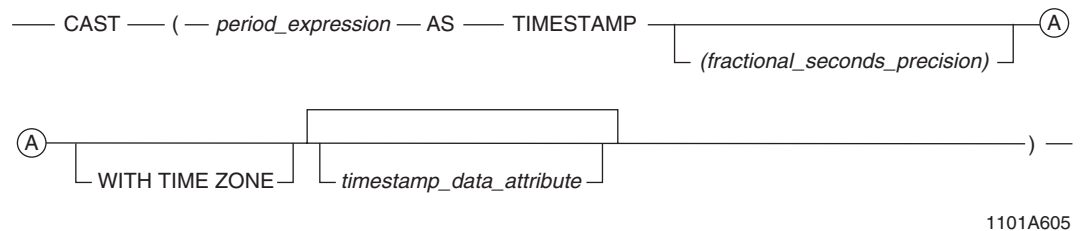
Period-to-TIME Conversion

CAST



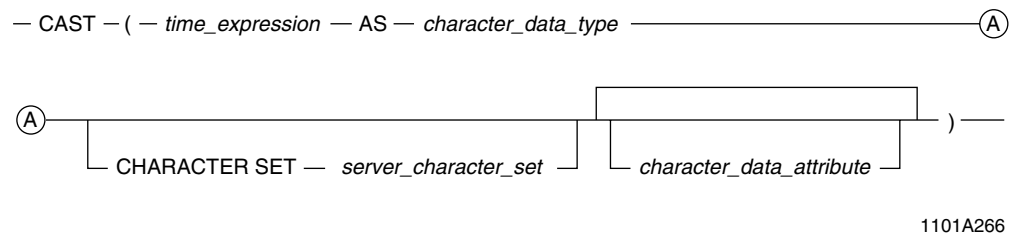
Period-to-TIMESTAMP Conversion

CAST

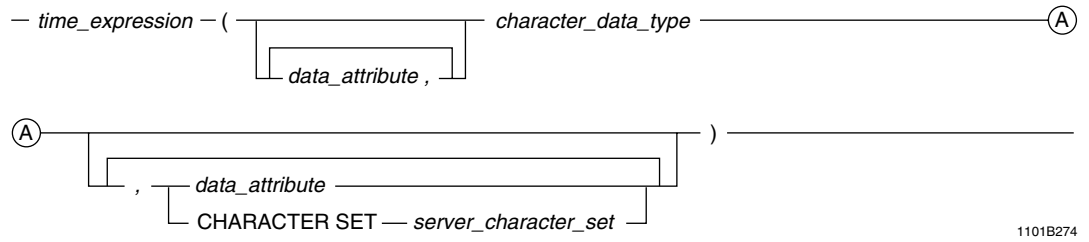


TIME-to-Character Conversion

CAST

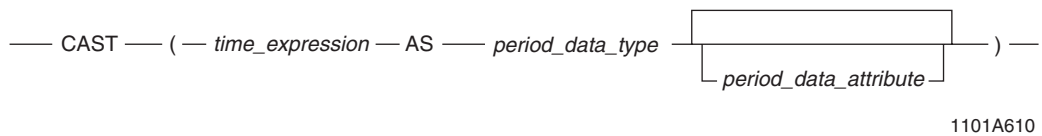


Teradata Conversion



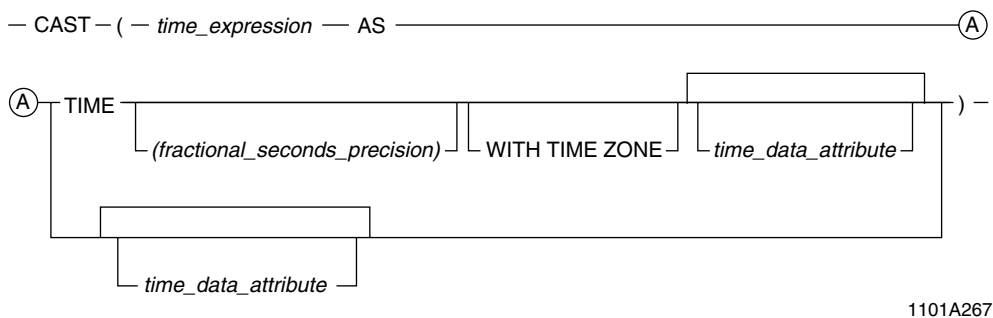
TIME-to-Period Conversion

CAST

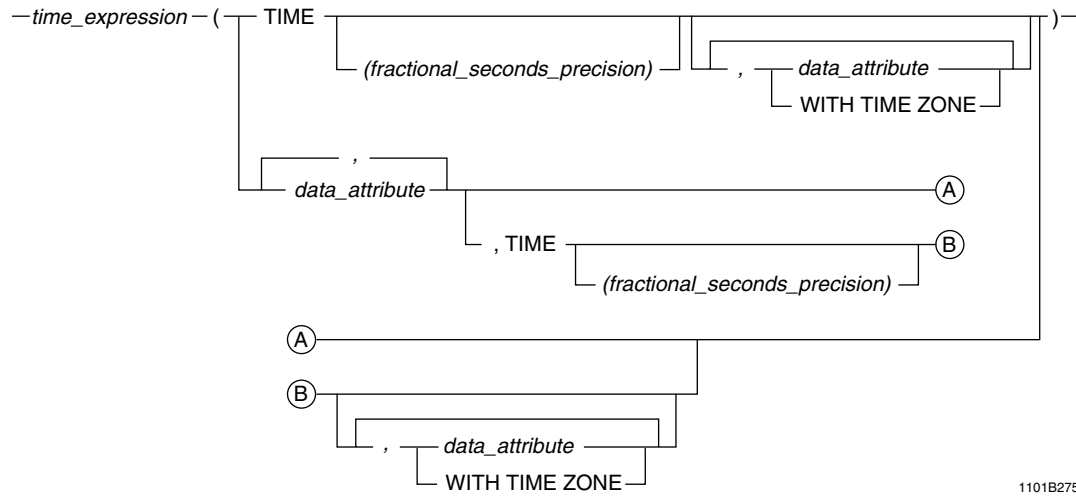


TIME-to-TIME Conversion

CAST

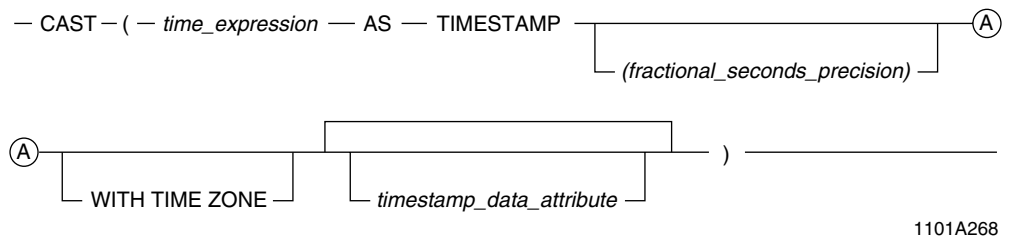


Teradata Conversion

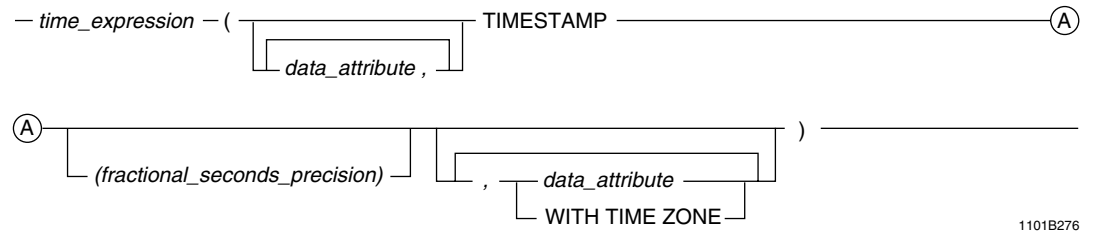


TIME-to-TIMESTAMP Conversion

CAST



Teradata Conversion



TIME-to-UDT Conversion

CAST

— CAST — (*time_expression* — AS — *UDT_data_definition*) —

1101A340

TIMESTAMP-to-Character Conversion

CAST

— CAST — (— *timestamp_expression* — AS — *character_data_type* —————— (A)

(A) ——————) —
 CHARACTER SET — *server_character_set* —————— *character_data_attribute* —

1101A269

Teradata Conversion

— *timestamp_expression* — (—————— *character_data_type* —————— (A)
 —————— *data_attribute* , —

(A) ——————) ——————
 , —————— *data_attribute* ——————
 CHARACTER SET — *server_character_set* —

1101B277

TIMESTAMP-to-DATE Conversion

CAST

— CAST — (— *timestamp_expression* — AS — DATE — date_data_attribute) —
1101A270

Teradata Conversion

— *timestamp_expression* — (data_attribute , DATE , data_attribute) —
1101B278

TIMESTAMP-to-Period Conversion

CAST

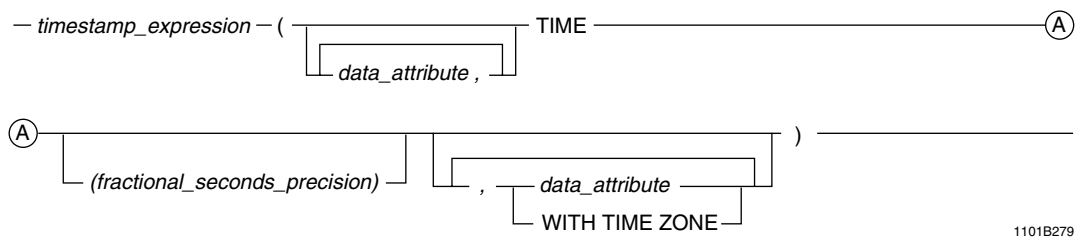
— CAST — (— *timestamp_expression* — AS — *period_data_type* — period_data_attribute) —
1101A608

TIMESTAMP-to-TIME Conversion

CAST

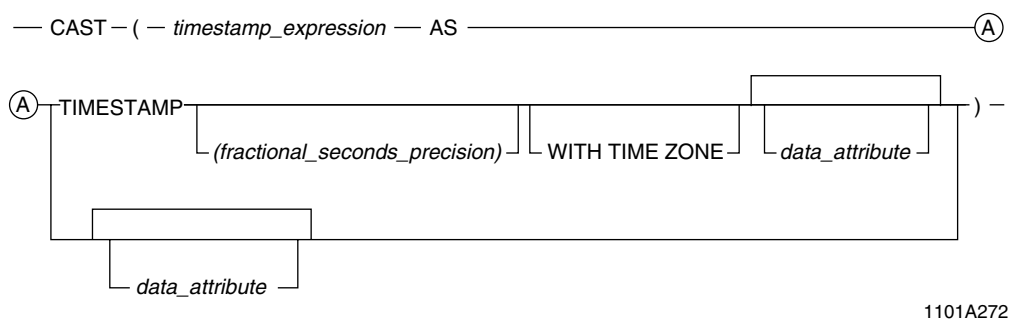
— CAST — (— *timestamp_expression* — AS — (fractional_seconds_precision) WITH TIME ZONE time_data_attribute) —
1101A271

Teradata Conversion

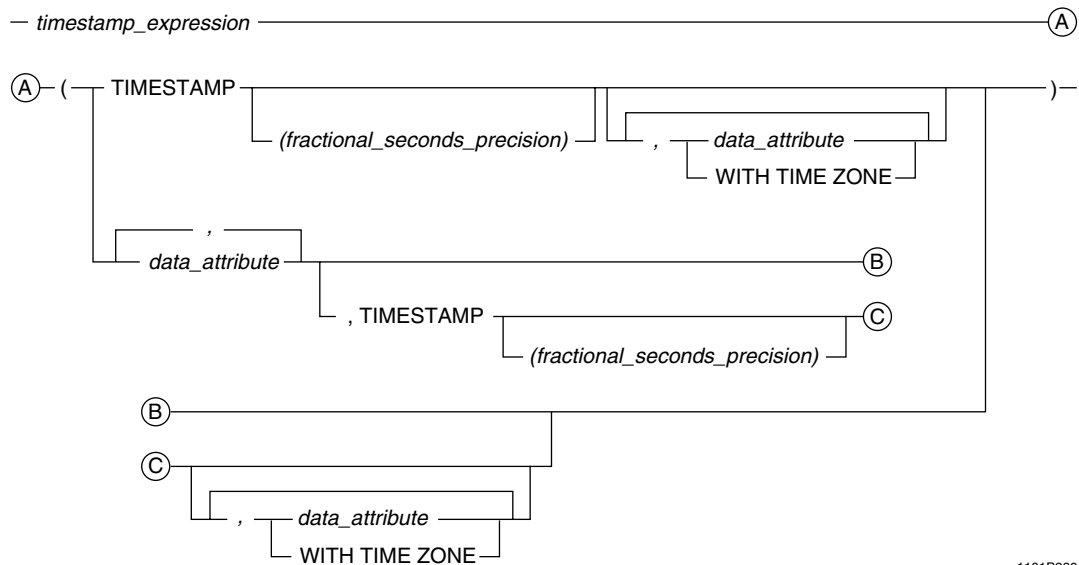


TIMESTAMP-to-TIMESTAMP Conversion

CAST



Teradata Conversion



1101B280

TIMESTAMP-to-UDT Conversion

CAST

— CAST — (*timestamp_expression* — AS — *UDT_data_definition*) —

1101A341

UDT-to-Byte Conversion

CAST

— CAST — (*UDT_expression* — AS — *byte_data_definition* —) —

1101A344

Teradata Conversion

— UDT_expression — (

data_attribute ,

 byte_data_type

, data_attribute

) —

1101B345

UDT-to-Character Conversion

CAST

— CAST — (UDT_expression — AS — character_data_definition —) —

1101A346

Teradata Conversion

— UDT_expression — (

data_attribute ,

 character_data_type

, data_attribute

) —

1101B347

UDT-to-DATE Conversion

CAST

— CAST — (UDT_expression — AS — DATE —

date_data_attribute

) —

1101B348

Teradata Conversion

— UDT_expression — ([data_attribute ,] DATE [, data_attribute])

1101B349

UDT-to-INTERVAL Conversion

CAST

— CAST — (UDT_expression AS interval_data_definition)

1101A350

Teradata Conversion

— UDT_expression — ([data_attribute ,] interval_data_type [, data_attribute])

1101B351

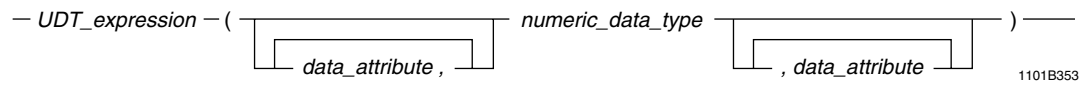
UDT-to-Numeric Conversion

CAST

— CAST — (UDT_expression AS numeric_data_definition)

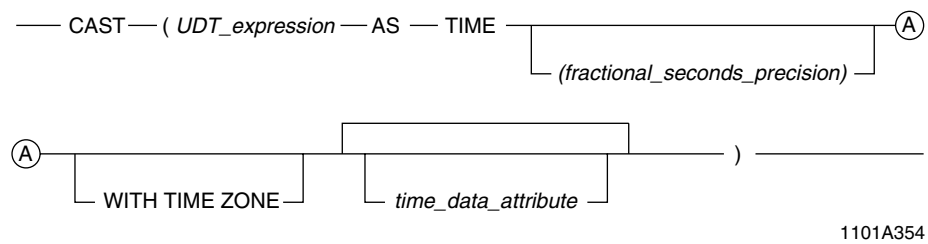
1101A352

Teradata Conversion

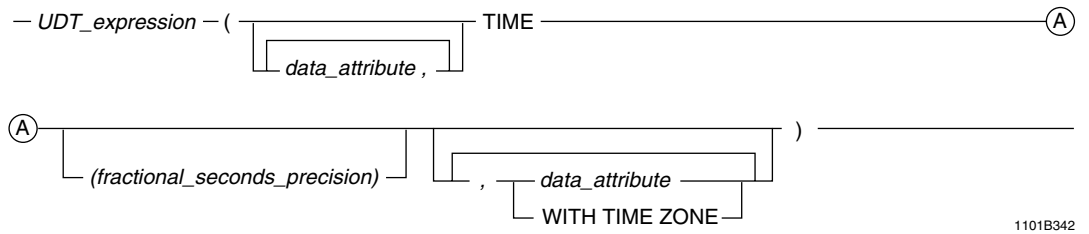


UDT-to-TIME Conversion

CAST

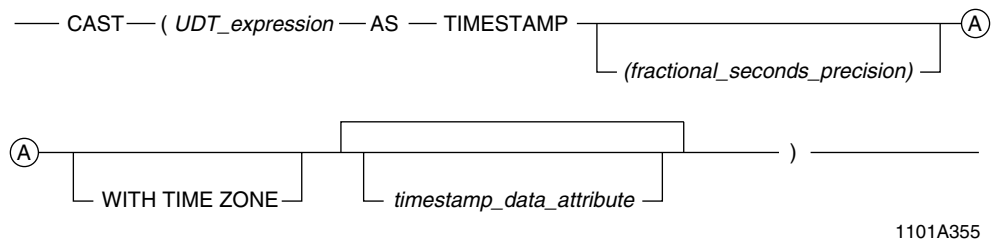


Teradata Conversion

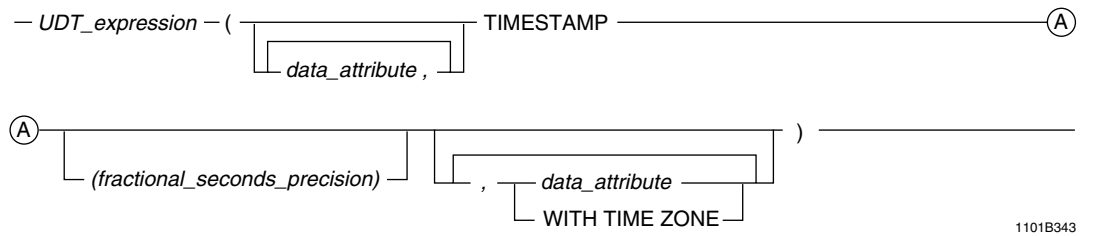


UDT-to-TIMESTAMP Conversion

CAST

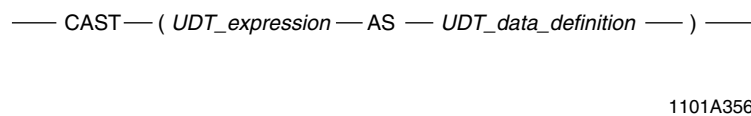


Teradata Conversion



UDT-to-UDT Conversion

CAST



DateTime and Interval Functions and Expressions

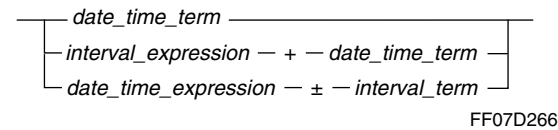
Arithmetic Operators and Result Types

The following arithmetic operations are permitted for DateTime and Interval data types:

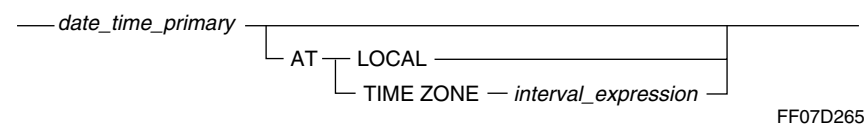
First Value Type	Operator	Second Value Type	Result Type
DateTime	-	DateTime	Interval
DateTime	+	Interval	DateTime
DateTime	-	Interval	DateTime
Interval	+	DateTime	DateTime
Interval	+	Interval	Interval
Interval	-	Interval	Interval
Interval	*	Number	Interval
Interval	/	Number	Interval
Number	*	Interval	Interval

ANSI DateTime Expressions

date_time_expression Syntax

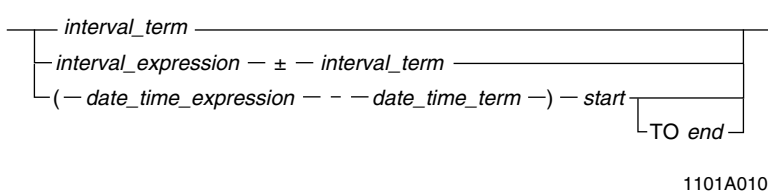


date_time_term Syntax

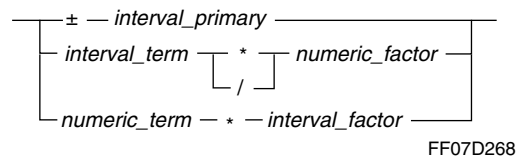


ANSI Interval Expressions

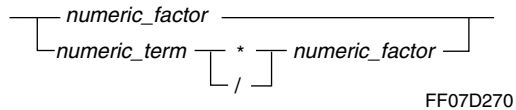
interval_expression Syntax



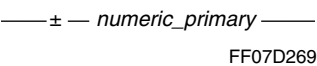
interval_term Syntax



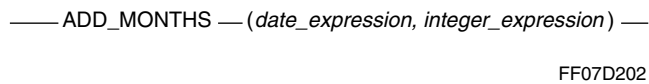
numeric_term Syntax



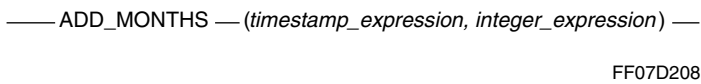
numeric_factor Syntax



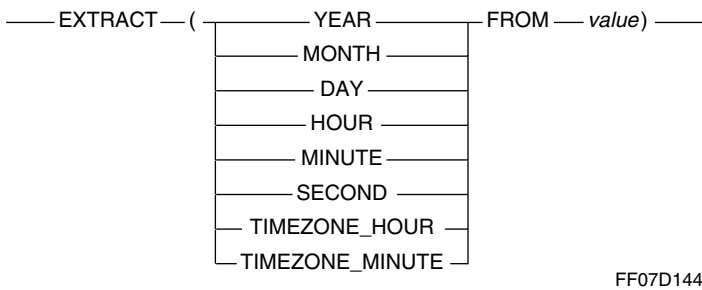
ADD_MONTHS (DATE Syntax)



ADD_MONTHS (TIMESTAMP Syntax)

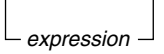


EXTRACT

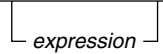


Hash-Related Functions


HASHAMP

— HASHAMP — (—  —) —
HH01A027

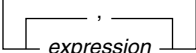
HASHBAKAMP

— HASHBAKAMP — (—  —) —
HH01A028

HASHBUCKET

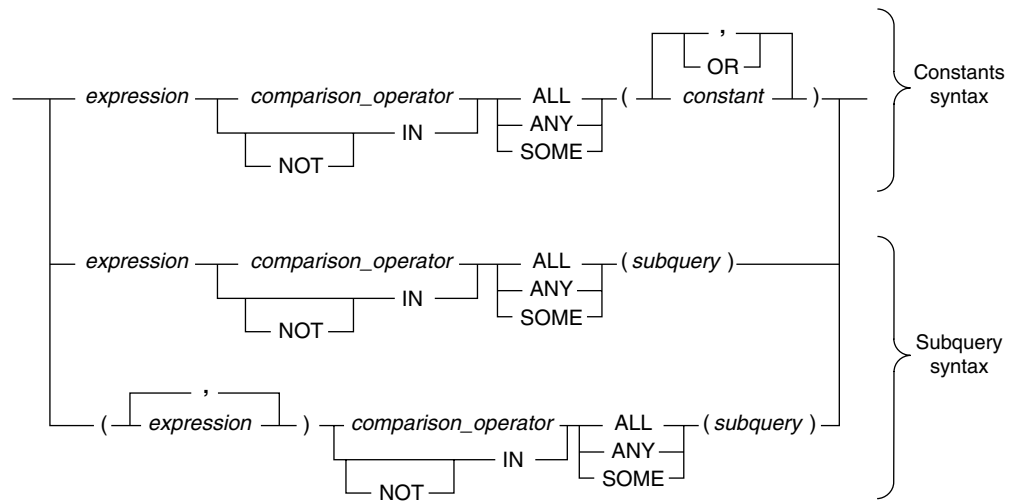
— HASHBUCKET — (—  —) —
HH01A026

HASHROW

— HASHROW — (—  —) —
1101B026

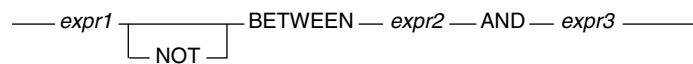
Logical Predicates

ANY/ALL/SOME Quantifiers



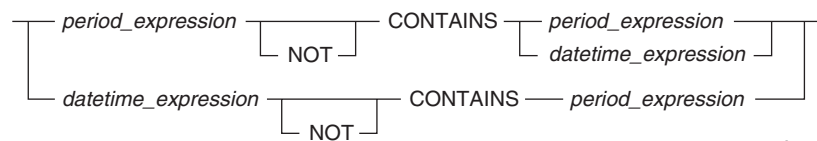
1101B090

BETWEEN/NOT BETWEEN



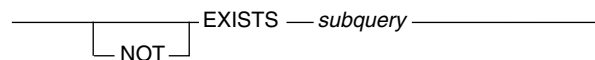
HH01A038

CONTAINS



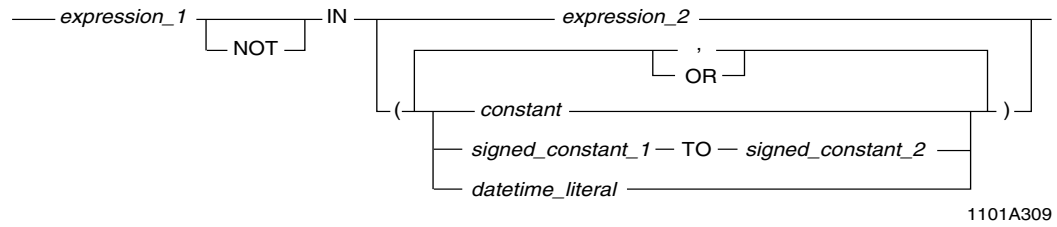
1101A582

EXISTS/NOT EXISTS

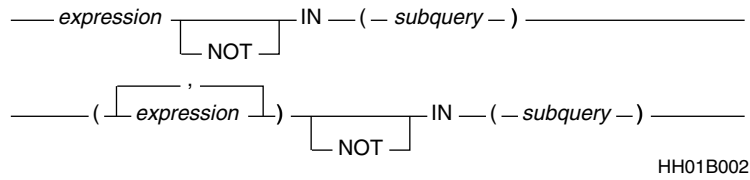


HH01A047

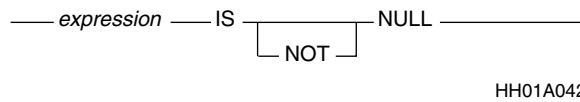
IN/NOT IN (Syntax 1)



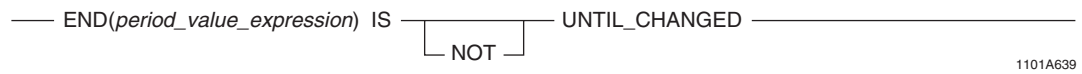
IN/NOT IN (Syntax 2)



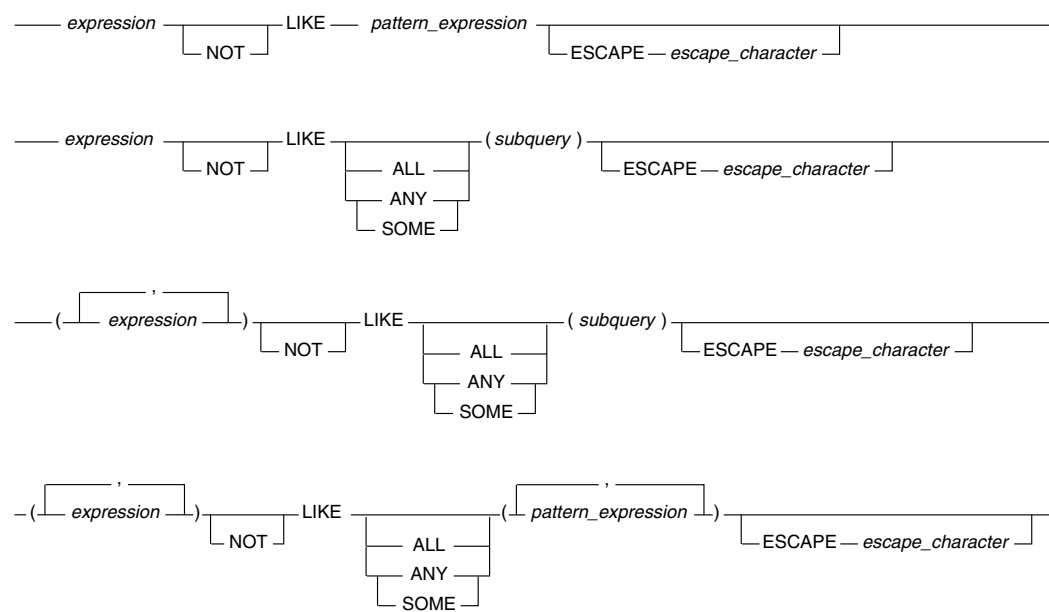
IS NULL/IS NOT NULL



IS UNTIL_CHANGED/IS NOT UNTIL_CHANGED

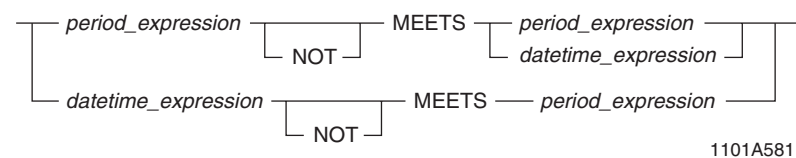


LIKE



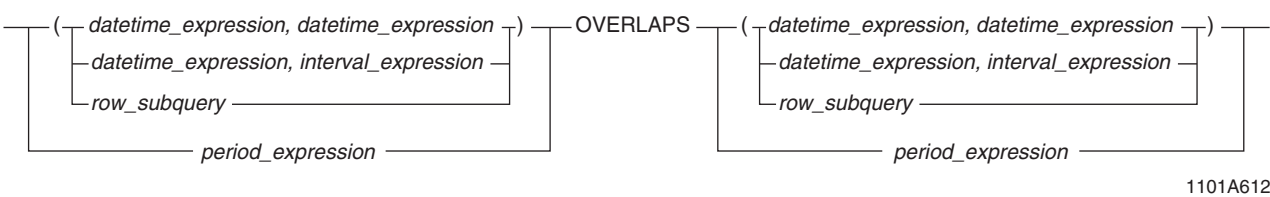
FF07D196

MEETS



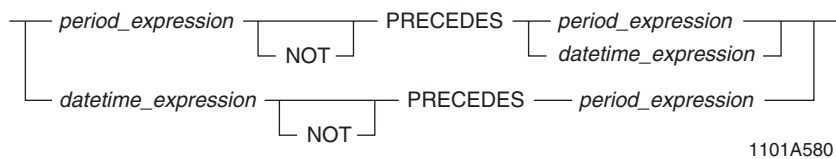
1101A581

OVERLAPS

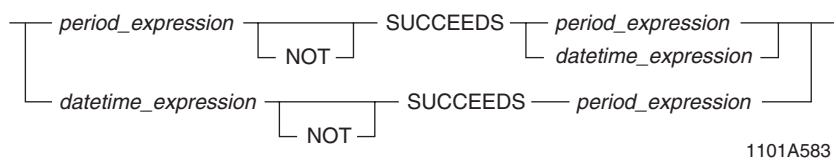


1101A612

PRECEDES



SUCCEEDS

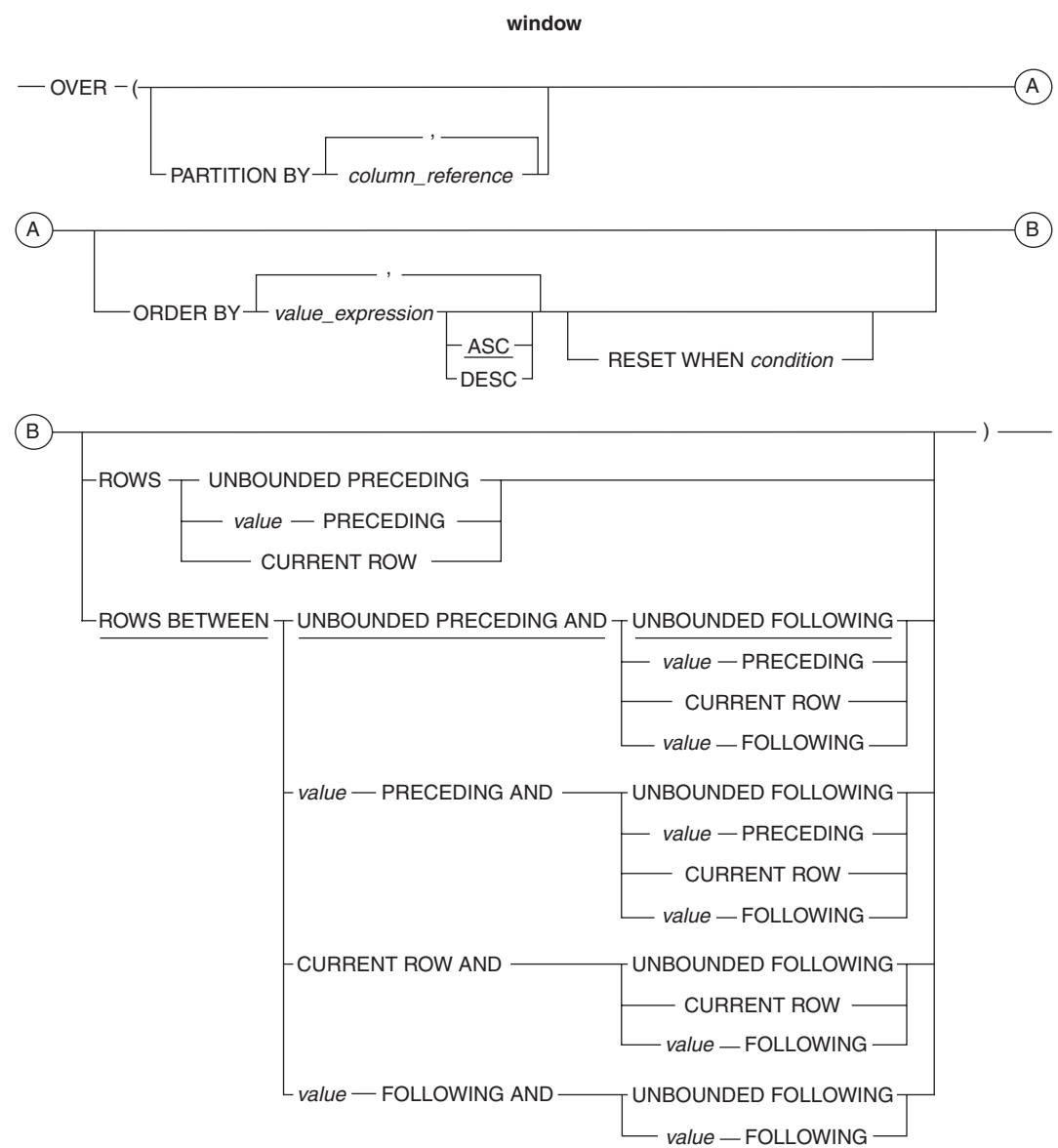


Ordered Analytical Functions

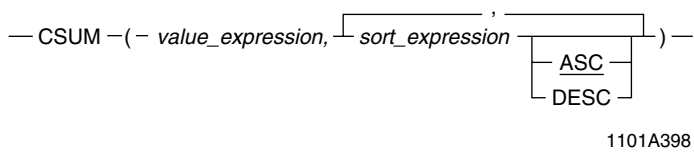
Window Aggregate Functions (AVG, CORR, COUNT, COVAR_POP, COVAR_SAMP, MAX, MIN, REGR_AVGX, REGR_AVGY, REGR_COUNT, REGR_INTERCEPT, REGR_R2, REGR_SLOPE, REGR_SXX, REGR_SXY, REGR_SYY, STDDEV_POP, STDDEV_SAMP, SUM, VAR_POP, VAR_SAMP)

AVG	— (value_expression)	A
COUNT	— (value_expression)	
COVAR_POP	— (value_expression_1, value_expression_2)	
COVAR_SAMP	— (value_expression_1, value_expression_2)	
CORR	— (value_expression_1, value_expression_2)	
MAX	— (value_expression)	
MIN	— (value_expression)	
REGR_AVGX	— (dependent_variable_expression, independent_variable_expression)	
REGR_AVGY	— (dependent_variable_expression, independent_variable_expression)	
REGR_COUNT	— (dependent_variable_expression, independent_variable_expression)	
REGR_INTERCEPT	— (dependent_variable_expression, independent_variable_expression)	
REGR_R2	— (dependent_variable_expression, independent_variable_expression)	
REGR_SLOPE	— (dependent_variable_expression, independent_variable_expression)	
REGR_SXX	— (dependent_variable_expression, independent_variable_expression)	
REGR_SXY	— (dependent_variable_expression, independent_variable_expression)	
REGR_SYY	— (dependent_variable_expression, independent_variable_expression)	
STDDEV_POP	— (value_expression)	
STDDEV_SAMP	— (value_expression)	
SUM	— (value_expression)	
VAR_POP	— (value_expression)	
VAR_SAMP	— (value_expression)	
	— window	

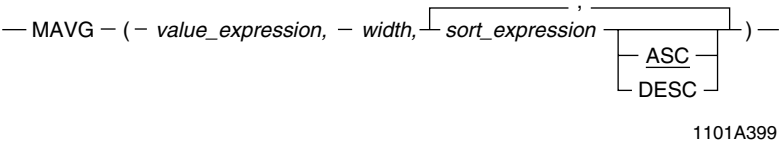
1101A465



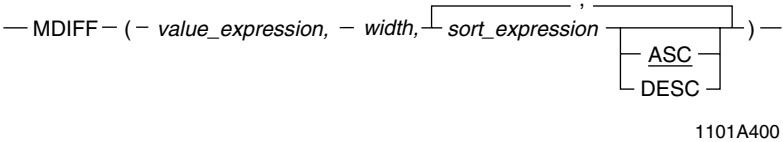
CSUM



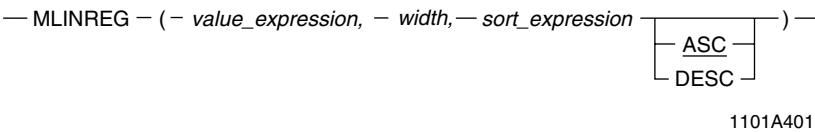
MAVG



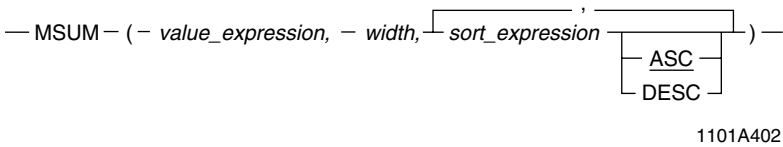
MDIFF



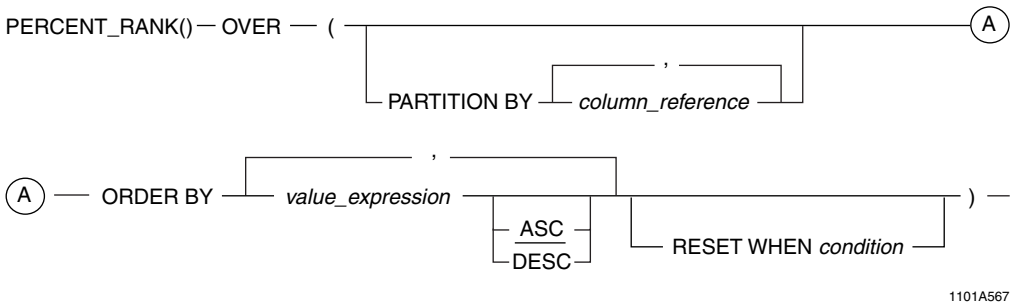
MLINREG



MSUM



PERCENT_RANK



QUANTILE

— QUANTILE — (— *quantile_constant*, — *sort_expression* — *ASC* — *DESC* —) —

1101A403

RANK (Teradata-Specific Function)

— RANK — (— *sort_expression* — *ASC* — *DESC* —) —

1101A404

RANK (SQL:2008 Window Function)

RANK() — OVER — (— *column_reference* —) — ^(A)

^(A) — ORDER BY — *value_expression* — *ASC* — *DESC* — RESET WHEN *condition* —) —

1101A566

ROW_NUMBER

— ROW_NUMBER() — OVER — (— *column_reference* —) — ^(A)

^(A) — ORDER BY — *value_expression* — *ASC* — *DESC* — RESET WHEN *condition* —) —

1101C108

Period Functions and Operators

BEGIN

—— BEGIN(*period_value_expression*) ——
1101A595

END

—— END(*period_value_expression*) ——
1101A596

LAST

—— LAST(*period_value_expression*) ——
1101A597

INTERVAL

—— INTERVAL (*period_expression*) —— *interval_qualifier* ——
1101A577

PRIOR

—— PRIOR (*datetime_expression*) ——
1101A578

NEXT

—— NEXT (*datetime_expression*) ——
1101A579

P_INTERSECT

period_expression — P_INTERSECT — *period_expression* —
1101A584

LDIFF

period_expression — LDIFF — *period_expression* —
1101A592

RDIFF

period_expression — RDIFF — *period_expression* —
1101A593

P_NORMALIZE

period_expression — P_NORMALIZE — *period_expression* —
1101A594

Period Value Constructor

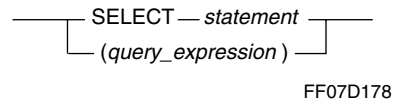
PERIOD (*datetime_expression*) —
PERIOD (*datetime_expression*, *datetime_expression*) —
PERIOD (*datetime_expression*, UNTIL_CHANGED) —
1101A585

Arithmetic Operators

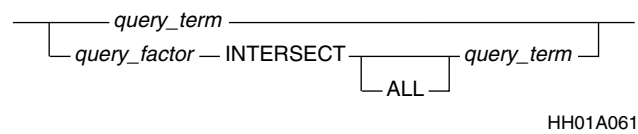
period_expression — + — *interval_expression* —
— - —
interval_expression — + — *period_expression* —
— - —
1101A586

Set Operators

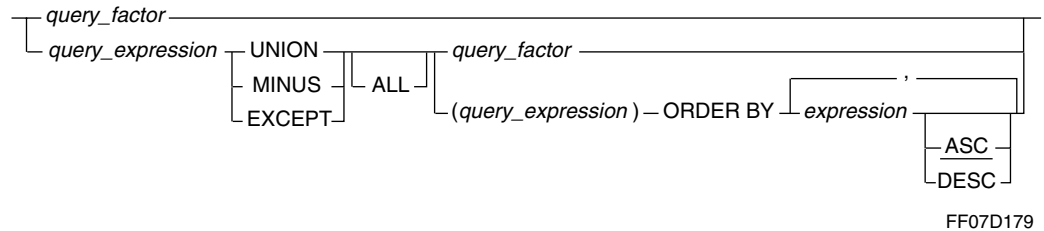
Syntax for *query_term*



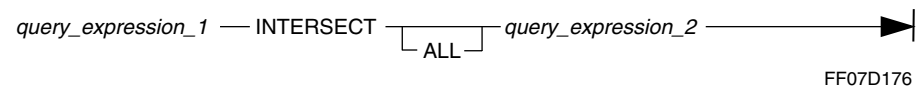
Syntax for *query_factor*



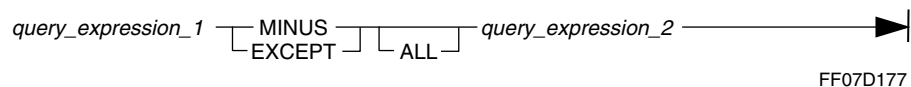
Syntax for *query_expression*



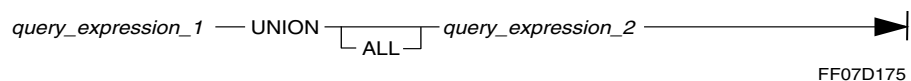
INTERSECT Operator



MINUS/EXCEPT Operator



UNION Operator

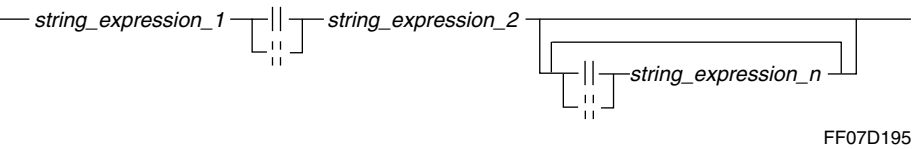


String Operator and Functions

CHAR2HEXINT

— CHAR2HEXINT — (*character_string_expression*) —
1101E173

Concatenation Operator



INDEX

— INDEX — (*string_expression_1* , *string_expression_2*) —
FF07D253

LOWER

— LOWER — (*character_string_expression*) —
FF07D091

POSITION

— POSITION — (*string_expression_1* IN *string_expression_2*) —
FF07D090

SOUNDEX

SOUNDEX — (— *string_expression* —) —
KO01A060

STRING_CS

— STRING_CS — (*string_expression*) —
1101A515

SUBSTRING/SUBSTR (ANSI Syntax)

— SUBSTRING — (*string_expression* — FROM — *n1* — FOR *n2*) —
FF07D256

SUBSTRING/SUBSTR (Teradata Syntax)

— SUBSTR — (*string_expression*, *n1* — , *n2*) —
FF07D257

TRANSLATE

— TRANSLATE — (— *character_string_expression* — USING — *source_repertoire_name* — A —
A _encoding — TO — *target_repertoire_name* — _suffix WITH ERROR) —
1101E198

TRANSLATE_CHK

— TRANSLATE_CHK — (— *character_string_expression* — USING — *source_repertoire_name* — A —
A _encoding — TO — *target_repertoire_name* — _suffix) —
1101E199

TRIM

— TRIM — (— BOTH TRAILING LEADING — FROM — _trim_expression character_set *string_expression* —) —
1101F200

UPPER

— UPPER — (*character_string_expression*) —
FF07D258

VARGRAPHIC

— VARGRAPHIC — (*character_string_expression*) —
1101E197

UDF Expressions

| Scalar UDF Expression

— *udf_name* — (

,

argument

) —
1101A640

| Aggregate UDF Expression

— *udf_name* — (

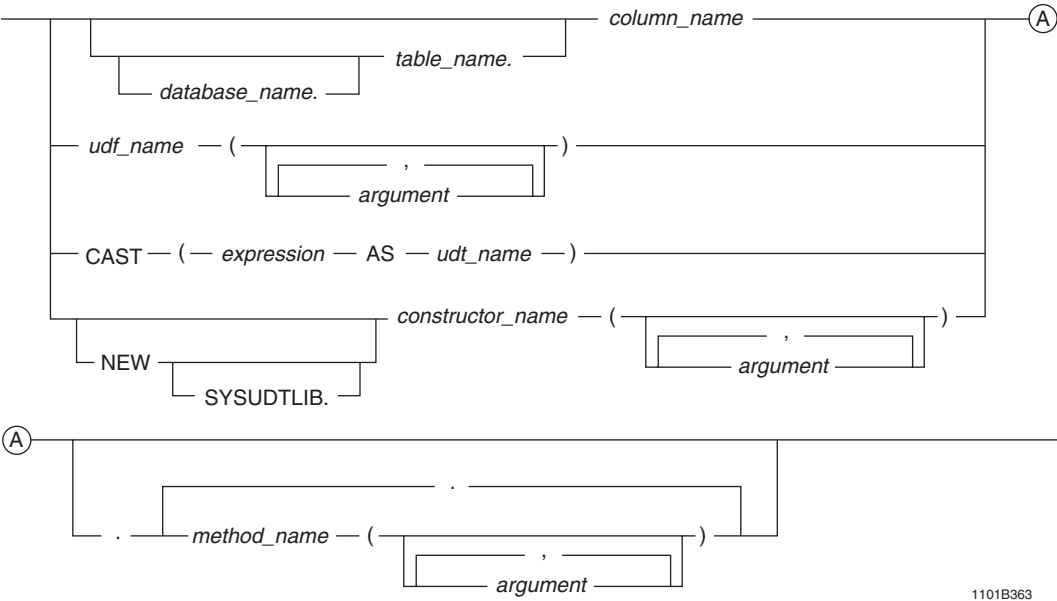
,

argument

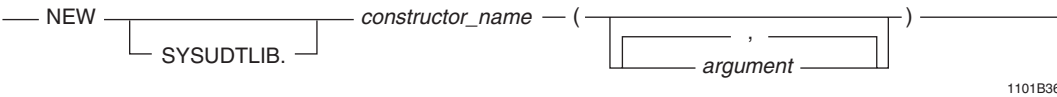
) —
1101A640

UDT Expressions and Methods

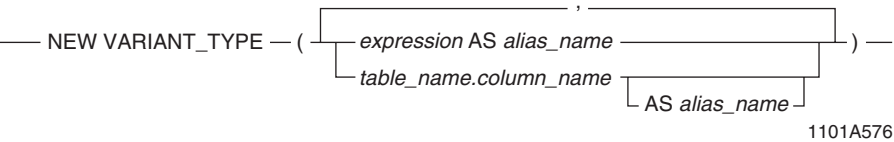
UDT Expression



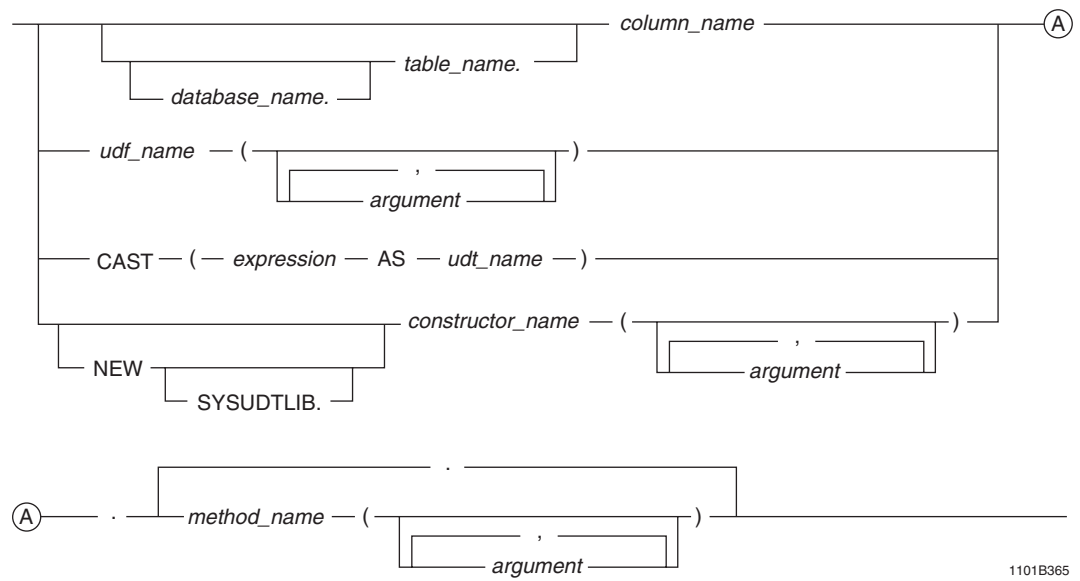
NEW



NEW VARIANT_TYPE

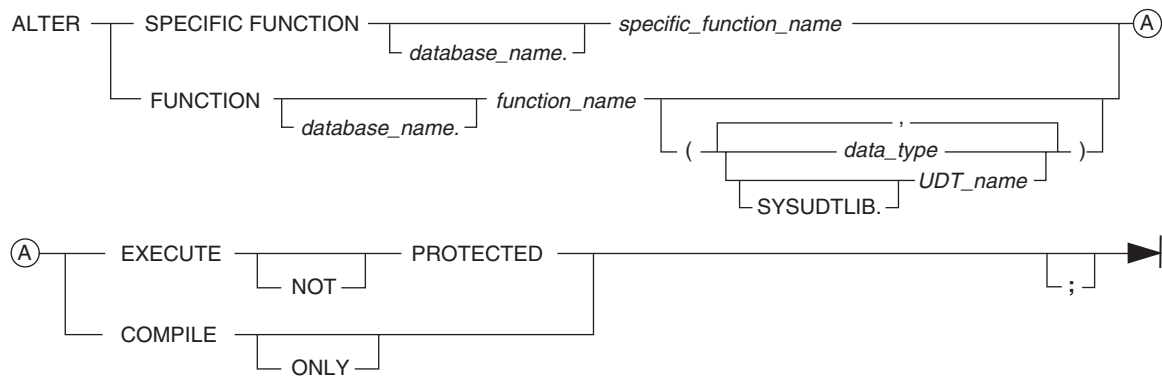


Method Invocation



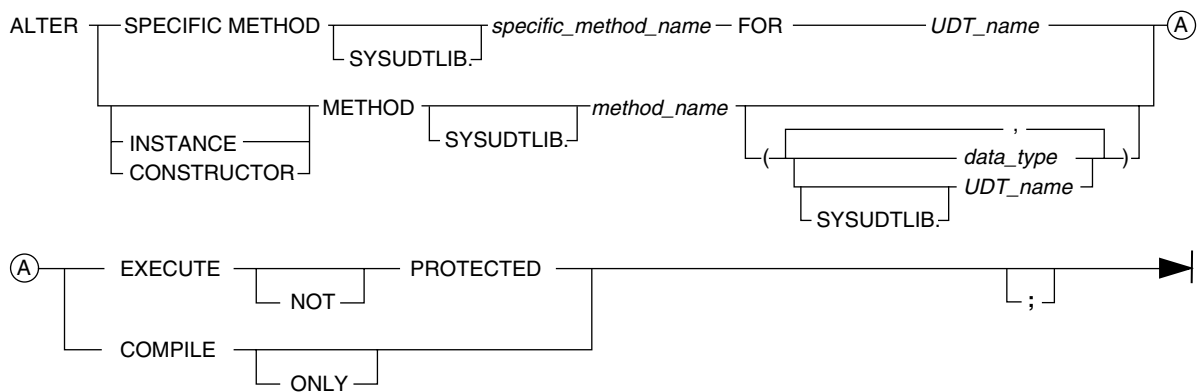
CHAPTER 3 SQL Data Definition Language

ALTER FUNCTION



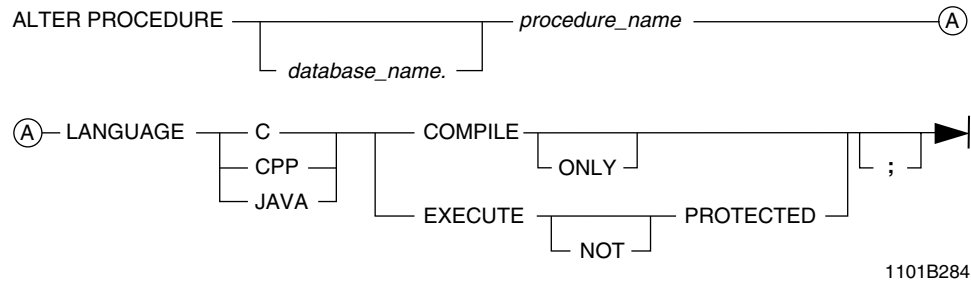
1101D097

ALTER METHOD

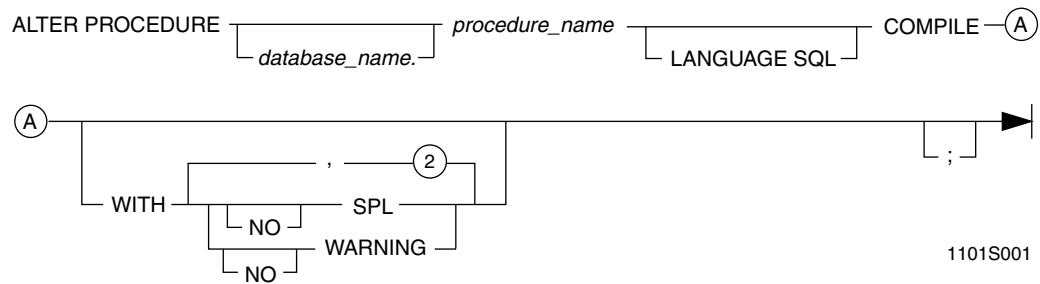


1101B369

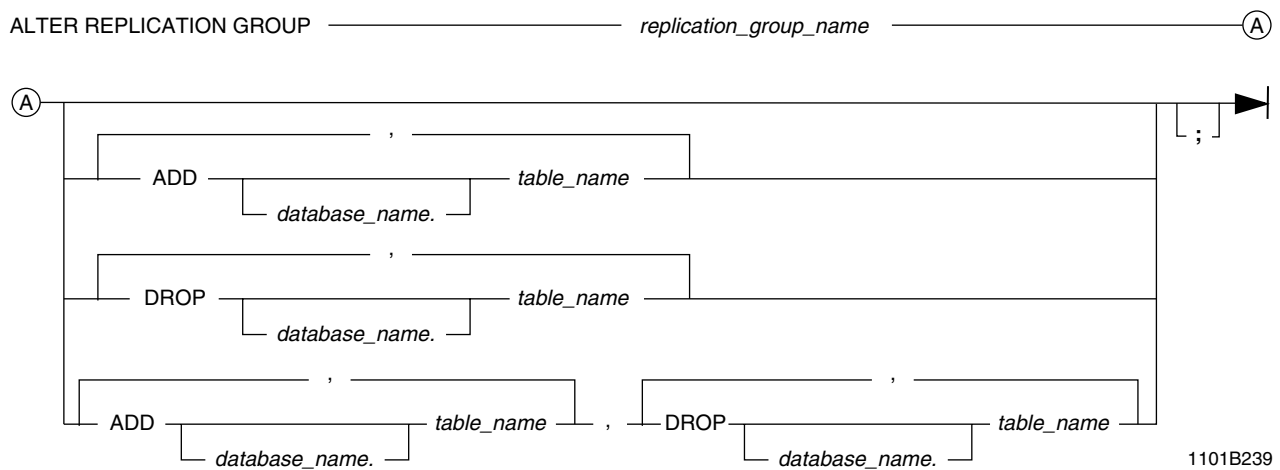
ALTER PROCEDURE (External Form)



ALTER PROCEDURE (SQL Form)

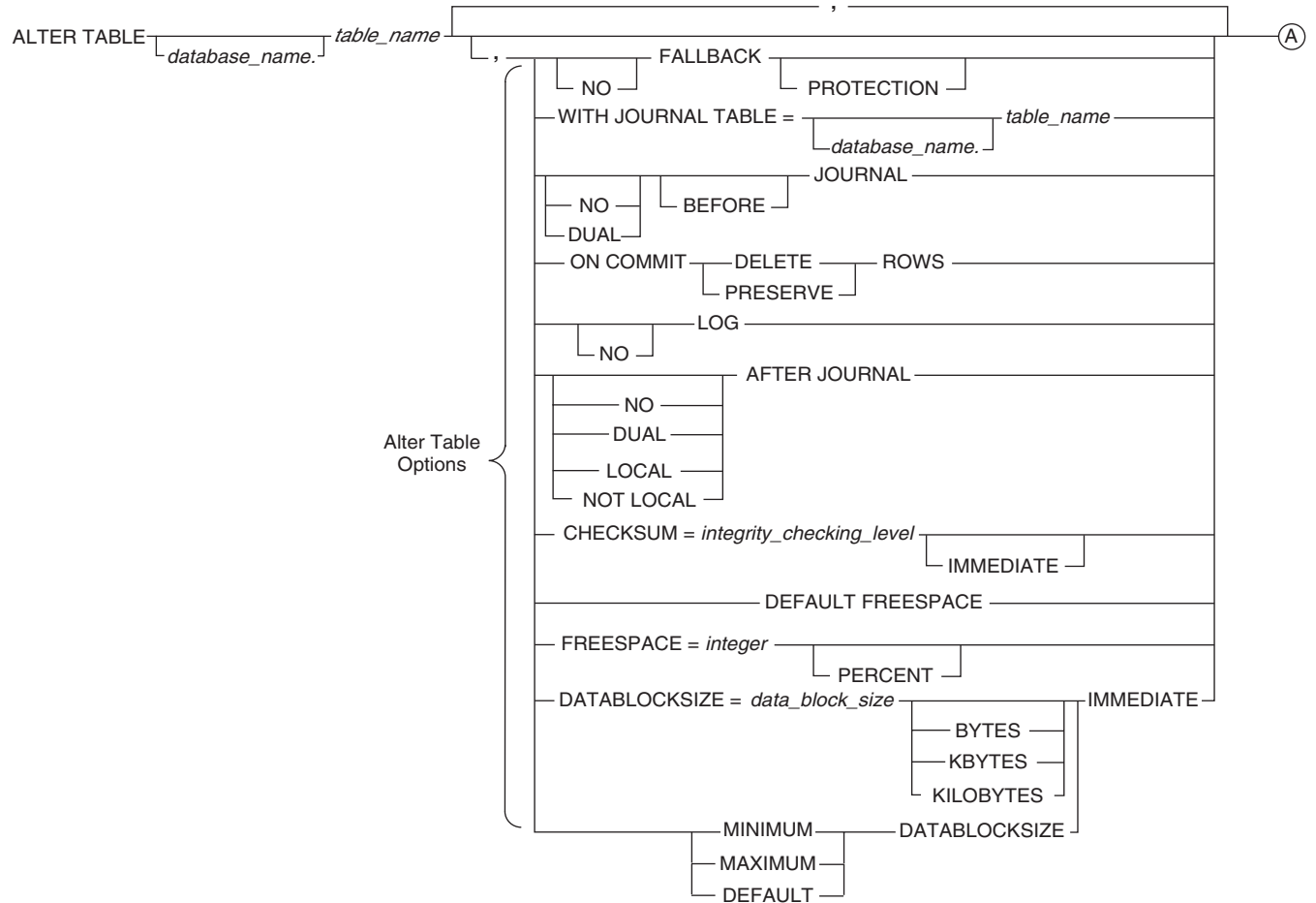


ALTER REPLICATION GROUP

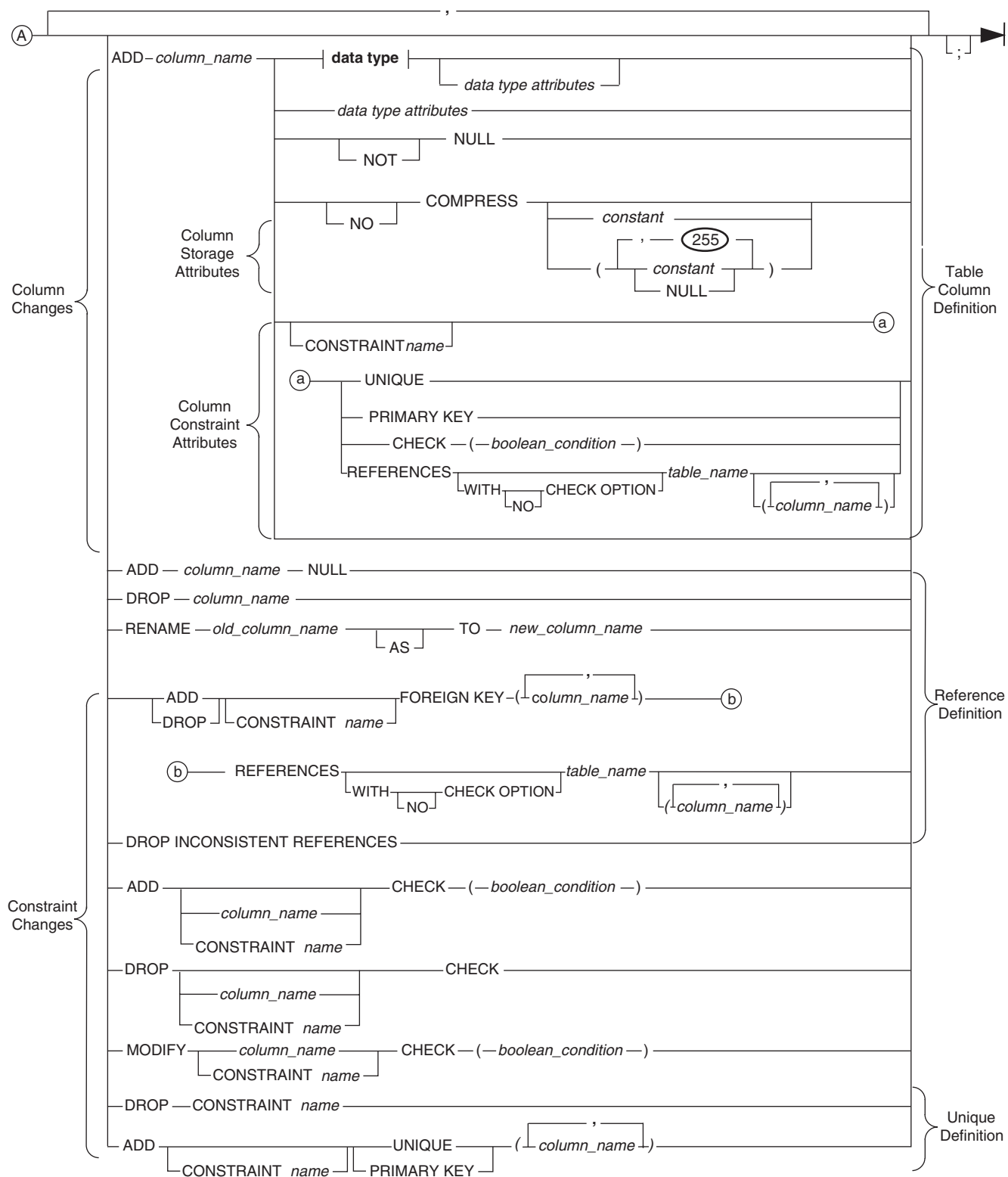


ALTER TABLE

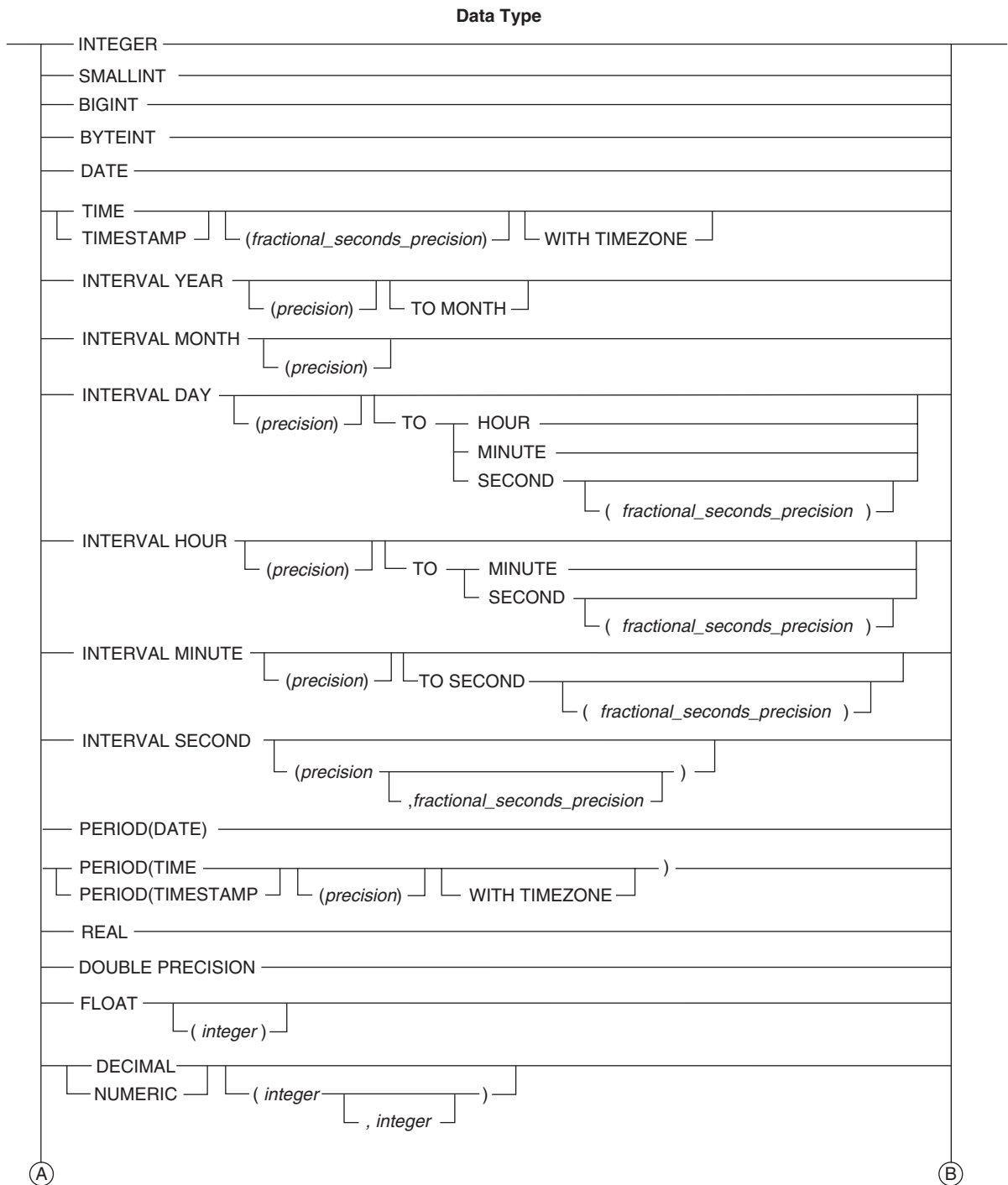
Basic Table Parameters Modification Syntax



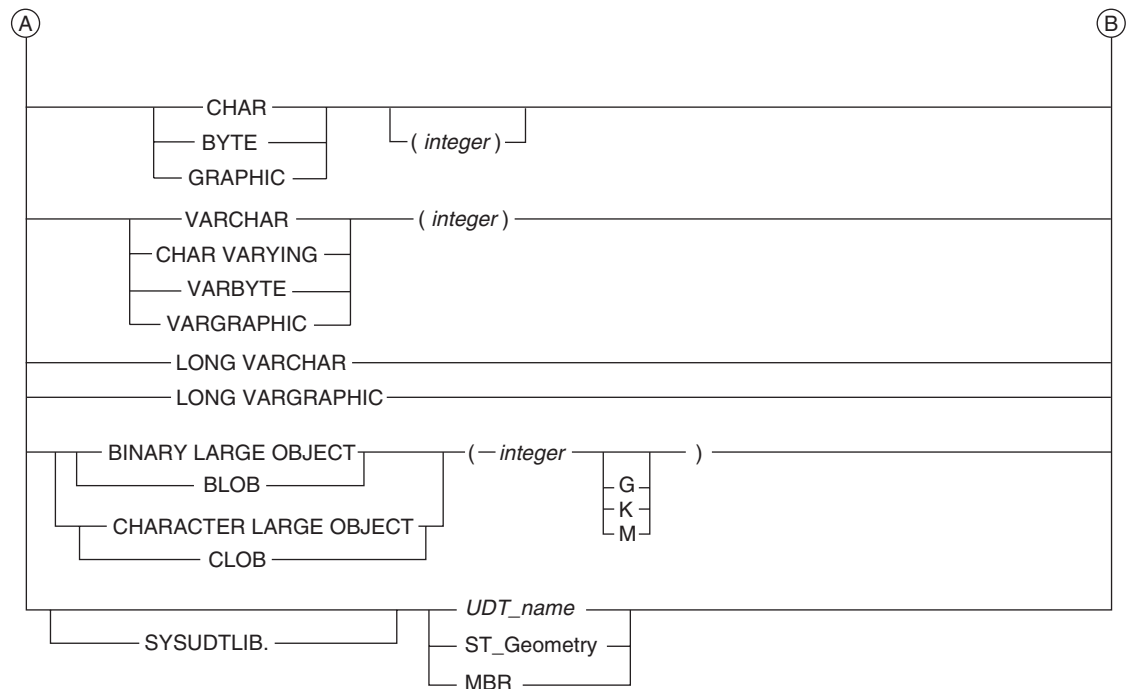
1101A389



1101H034

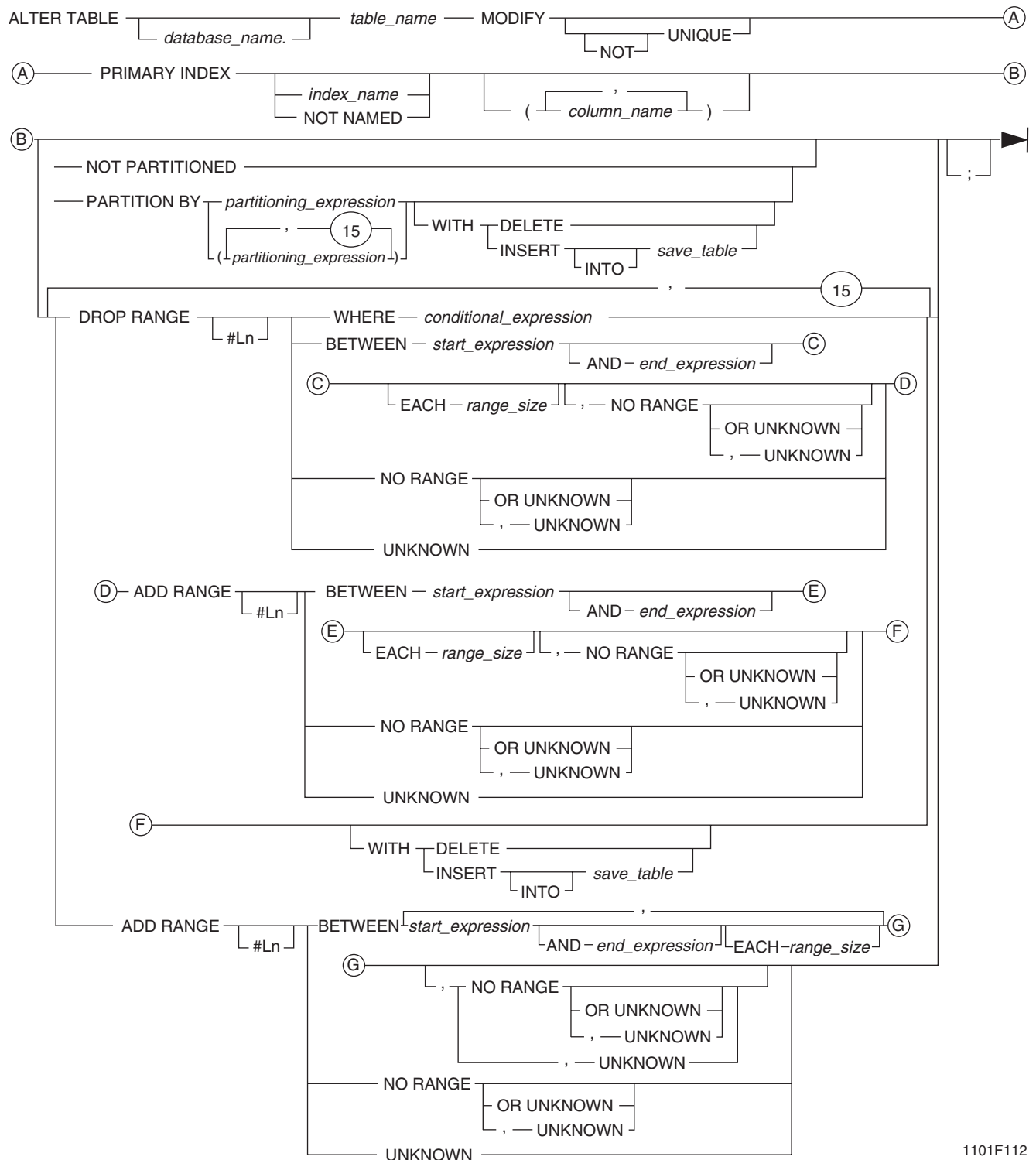


1101A535



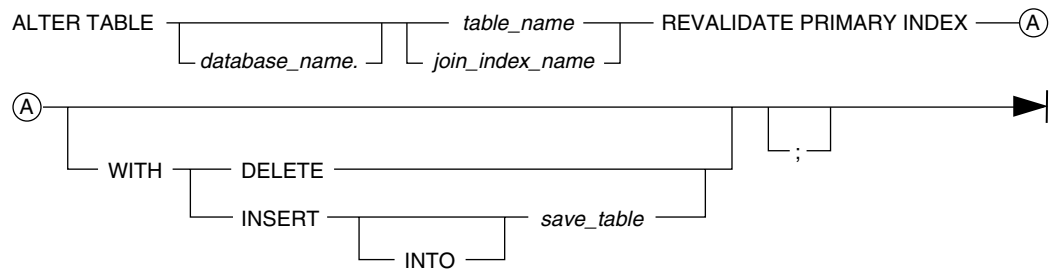
1101A536

Primary Index Modification Syntax



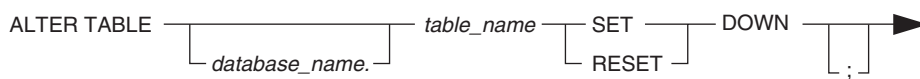
1101F112

Partitioned Primary Index Revalidation Syntax



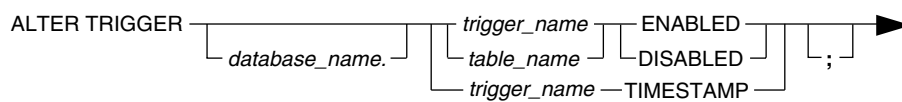
1101A432

Set Down/Reset Down Syntax



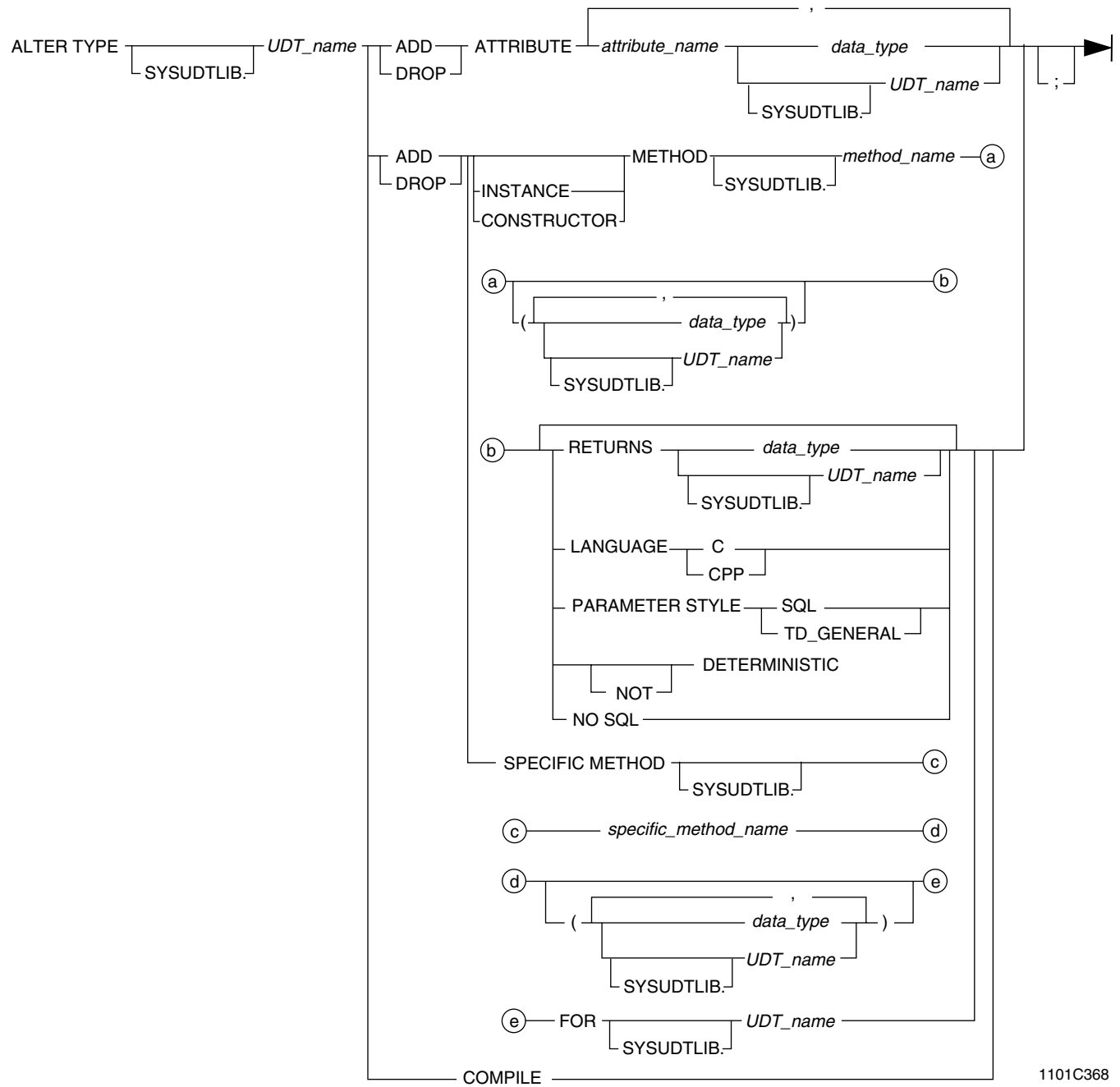
1101B543

ALTER TRIGGER



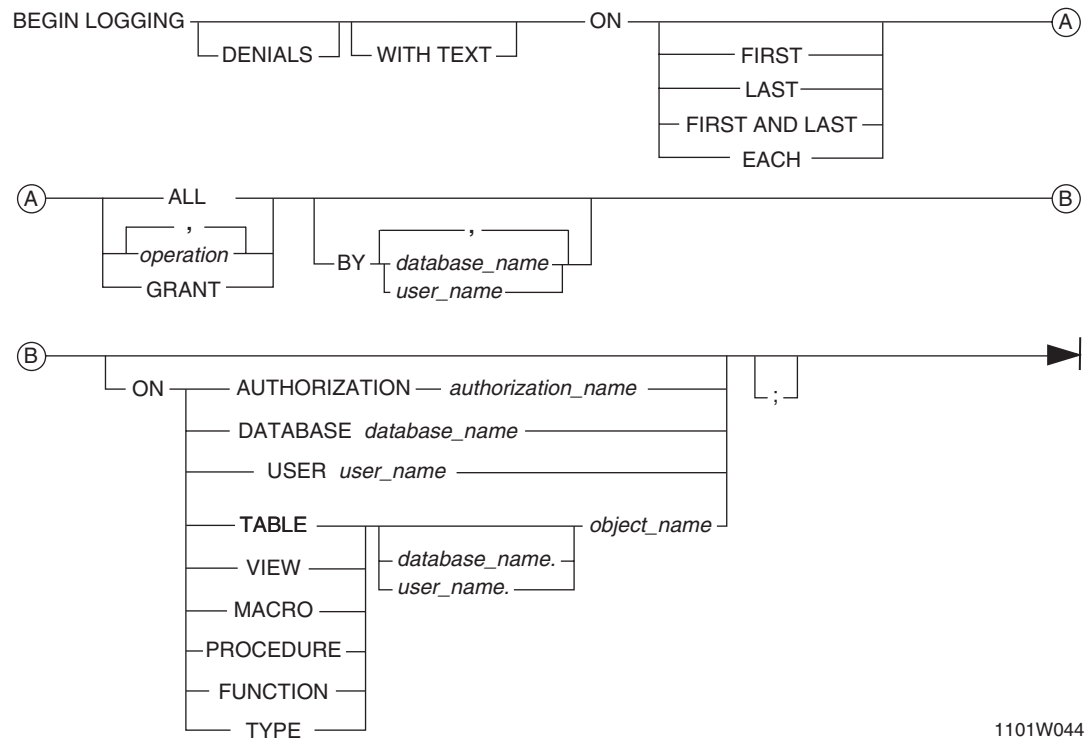
1101A201

ALTER TYPE

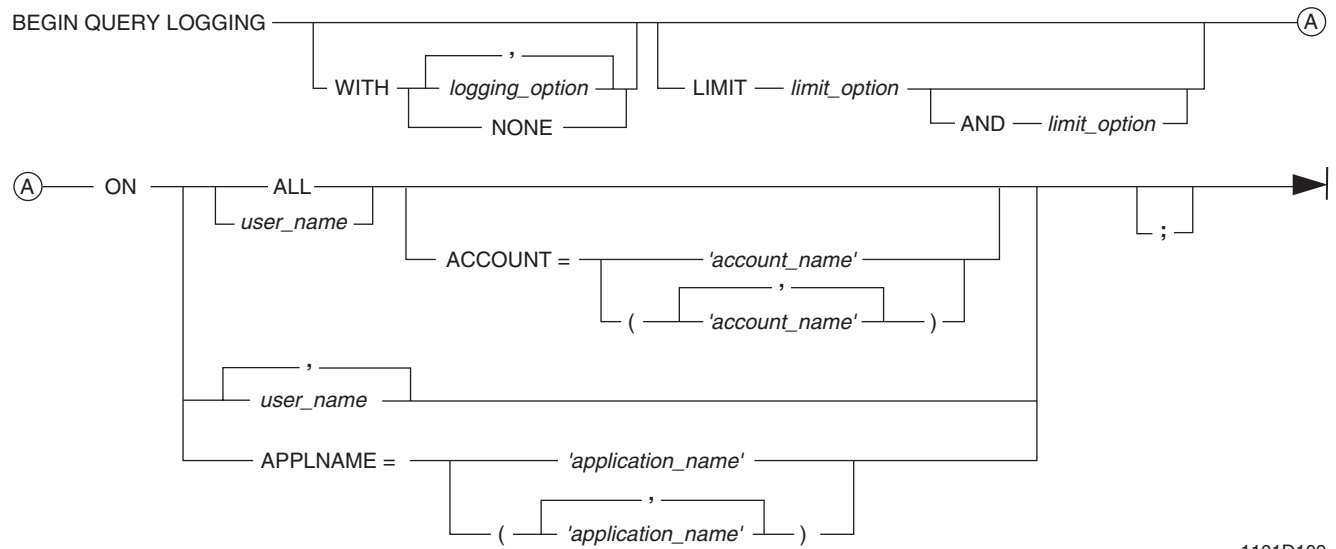


1101C368

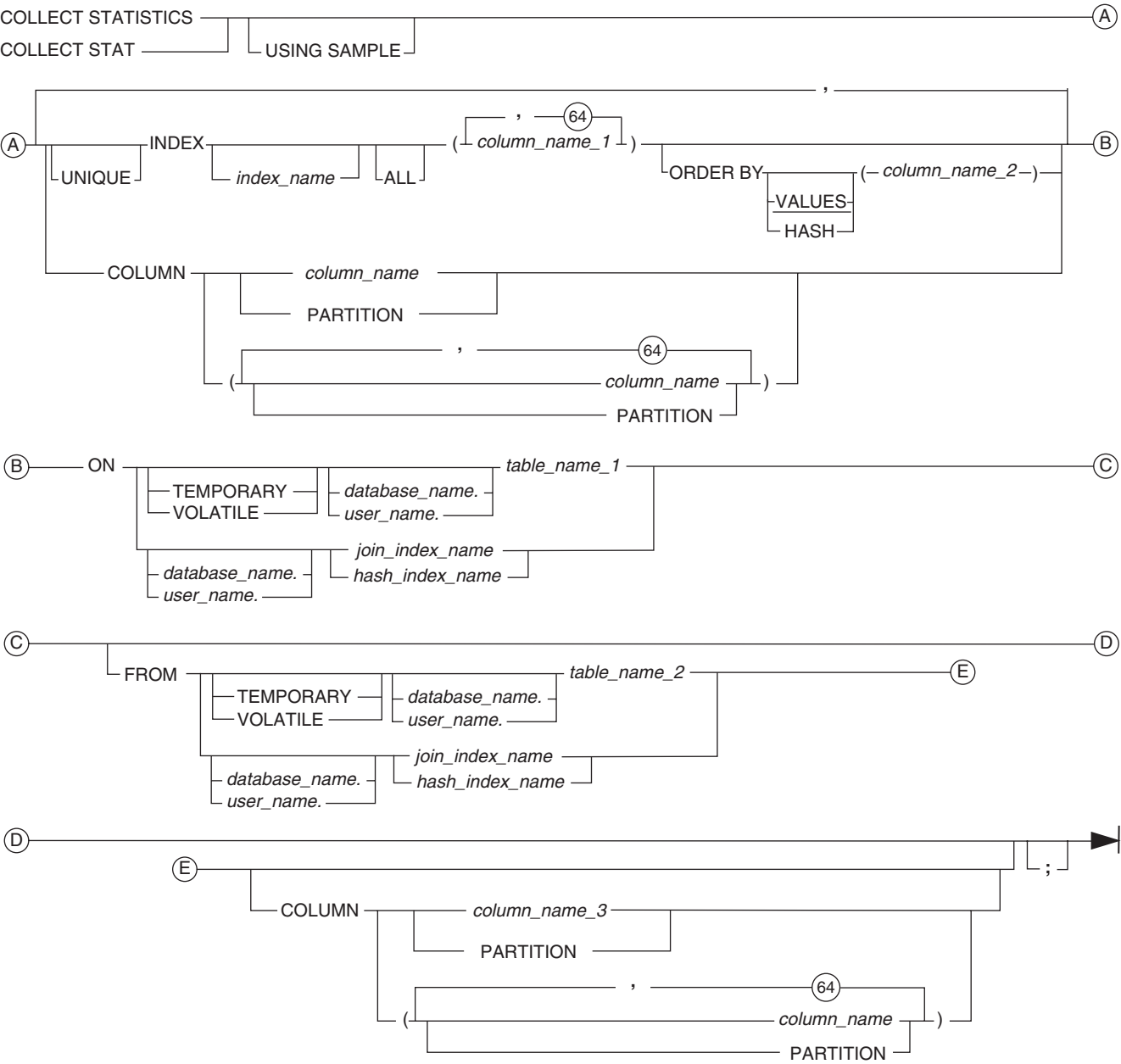
BEGIN LOGGING



BEGIN QUERY LOGGING

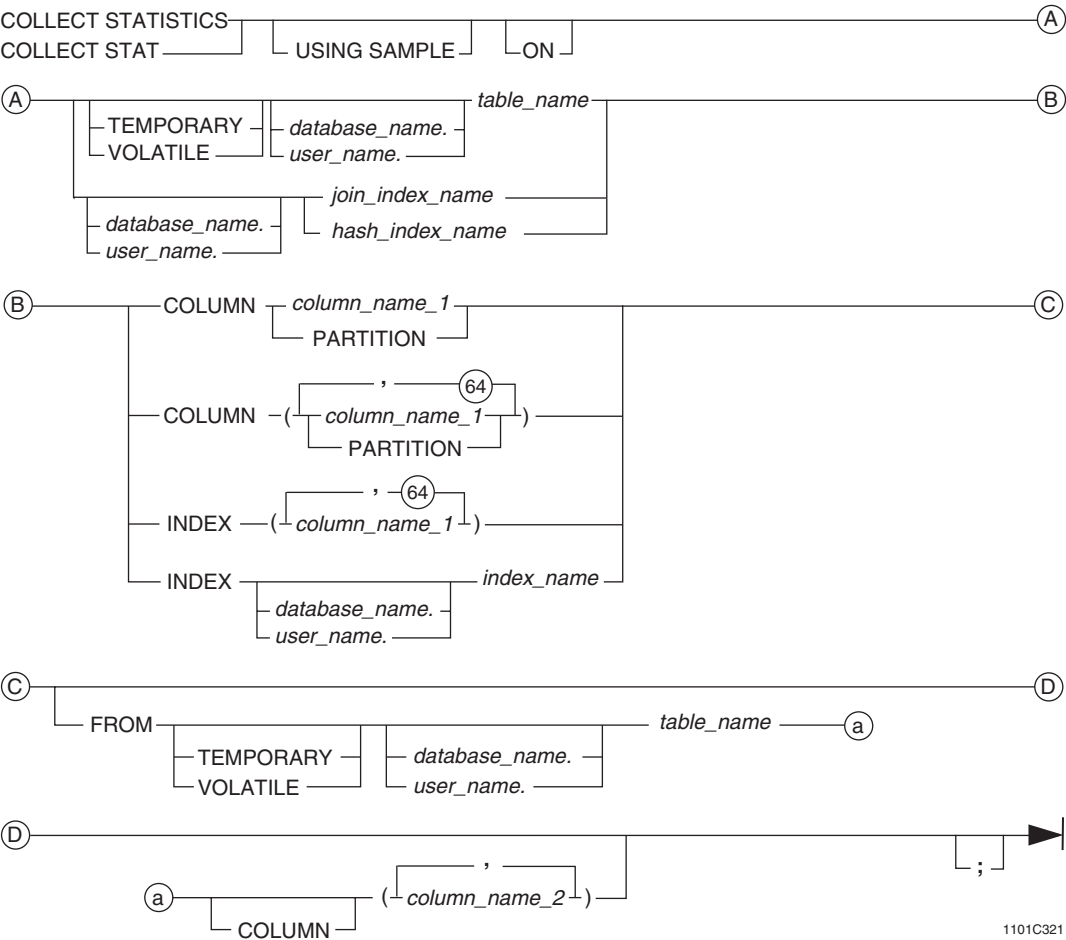


COLLECT STATISTICS (Optimizer Form)

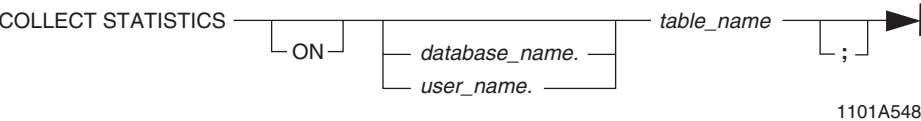


1101C322

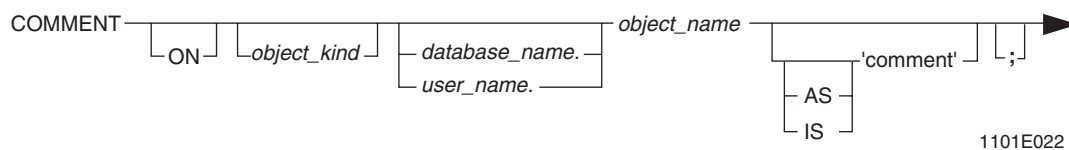
COLLECT STATISTICS (Alternate Optimizer Form)



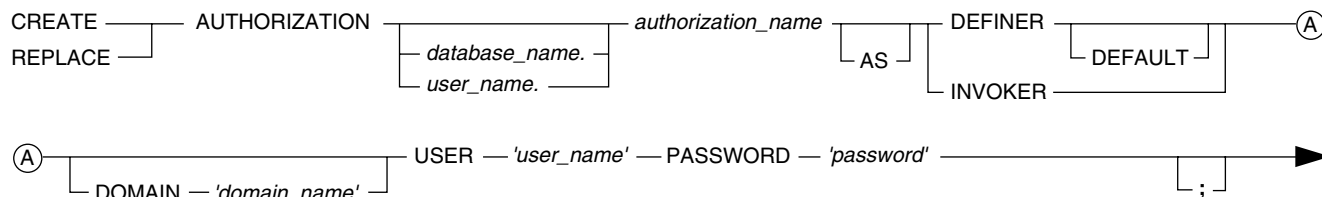
COLLECT STATISTICS (Recollect Statistics)



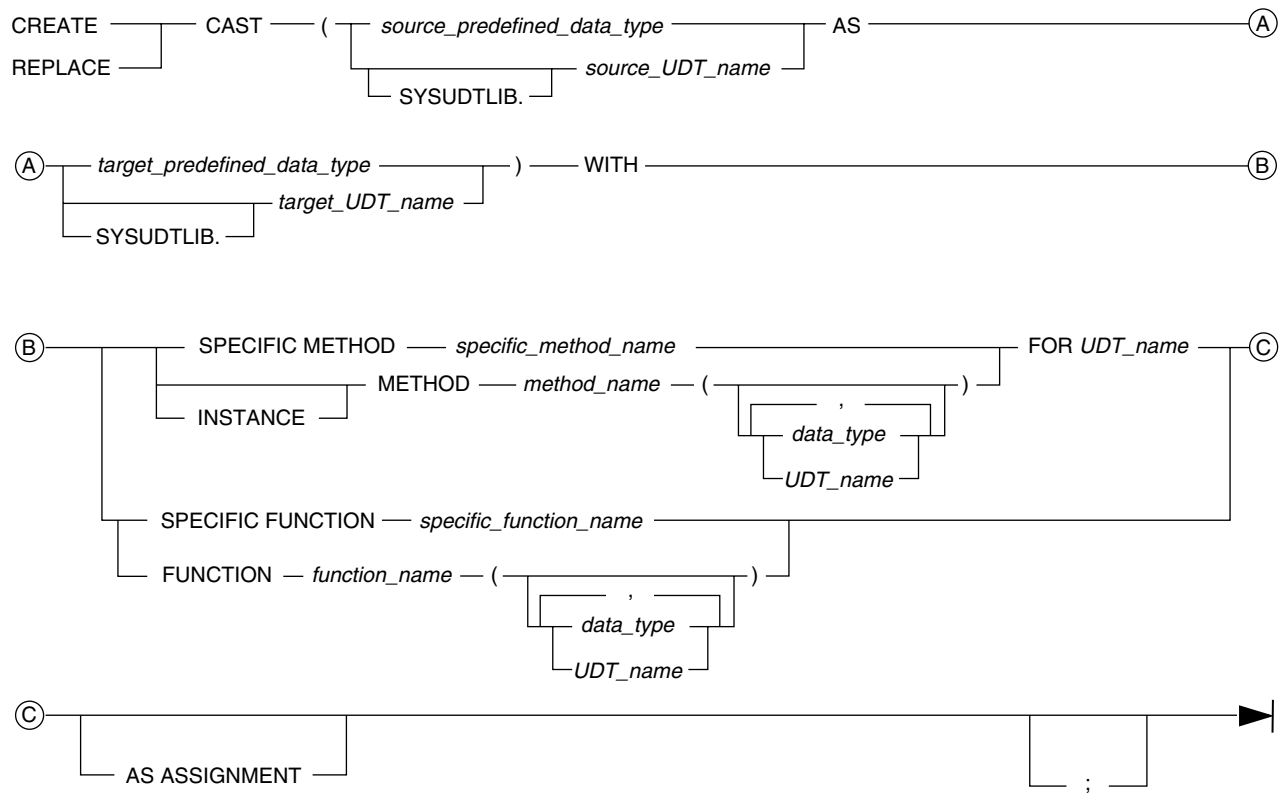
COMMENT (Comment Placing Form)



CREATE AUTHORIZATION/ REPLACE AUTHORIZATION

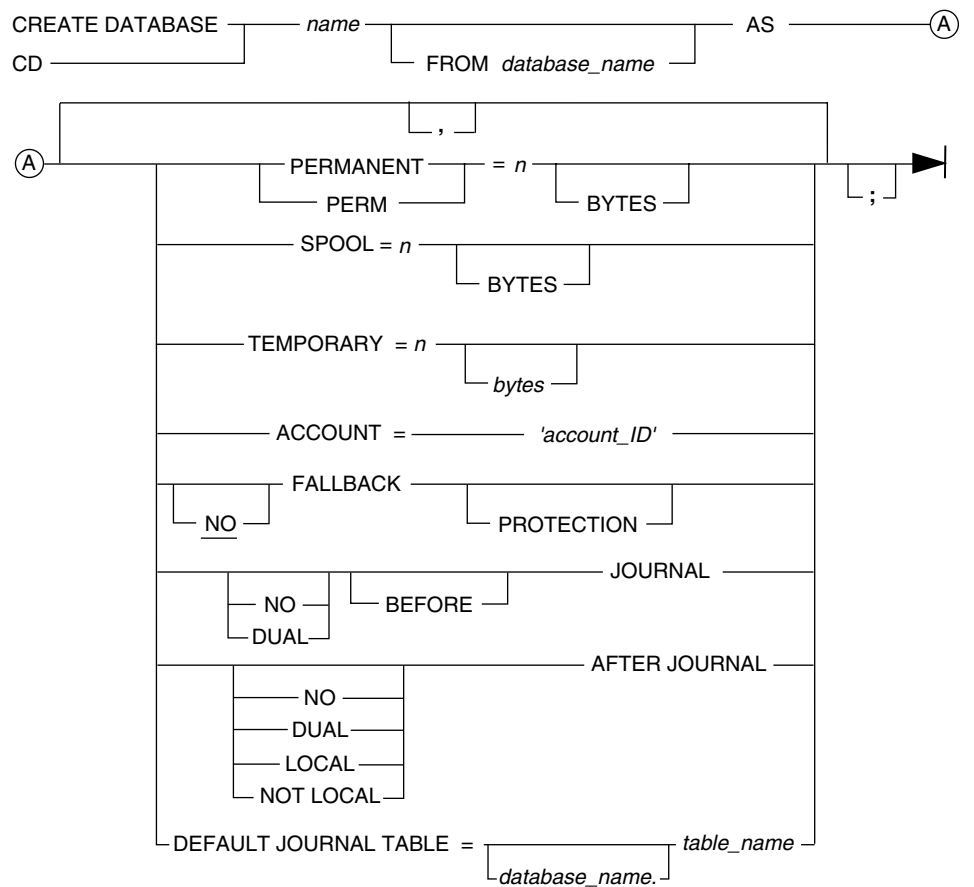


CREATE CAST/ REPLACE CAST



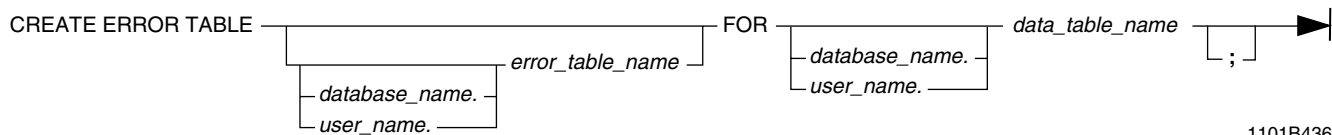
1101A358

CREATE DATABASE



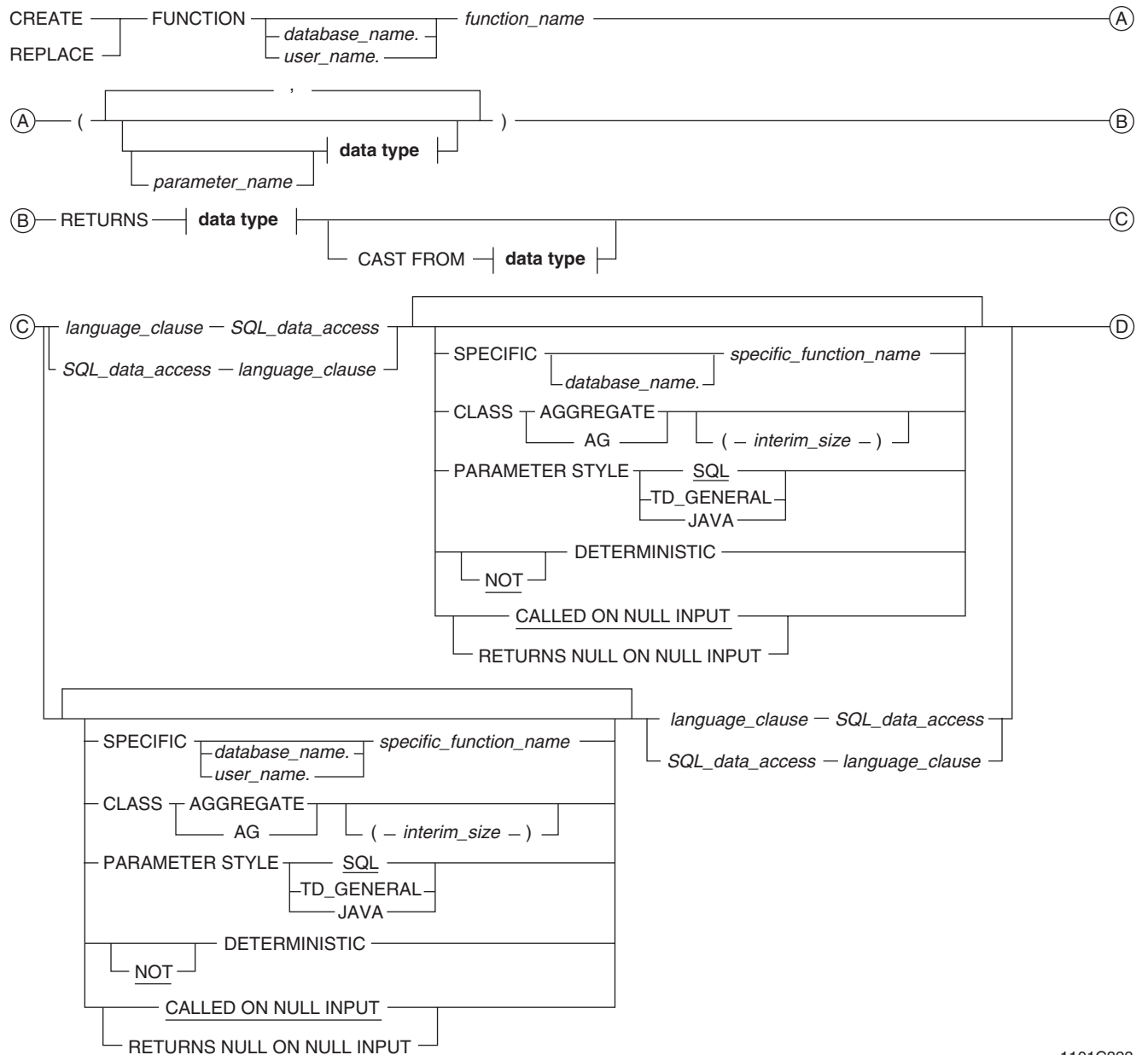
1101E030

CREATE ERROR TABLE

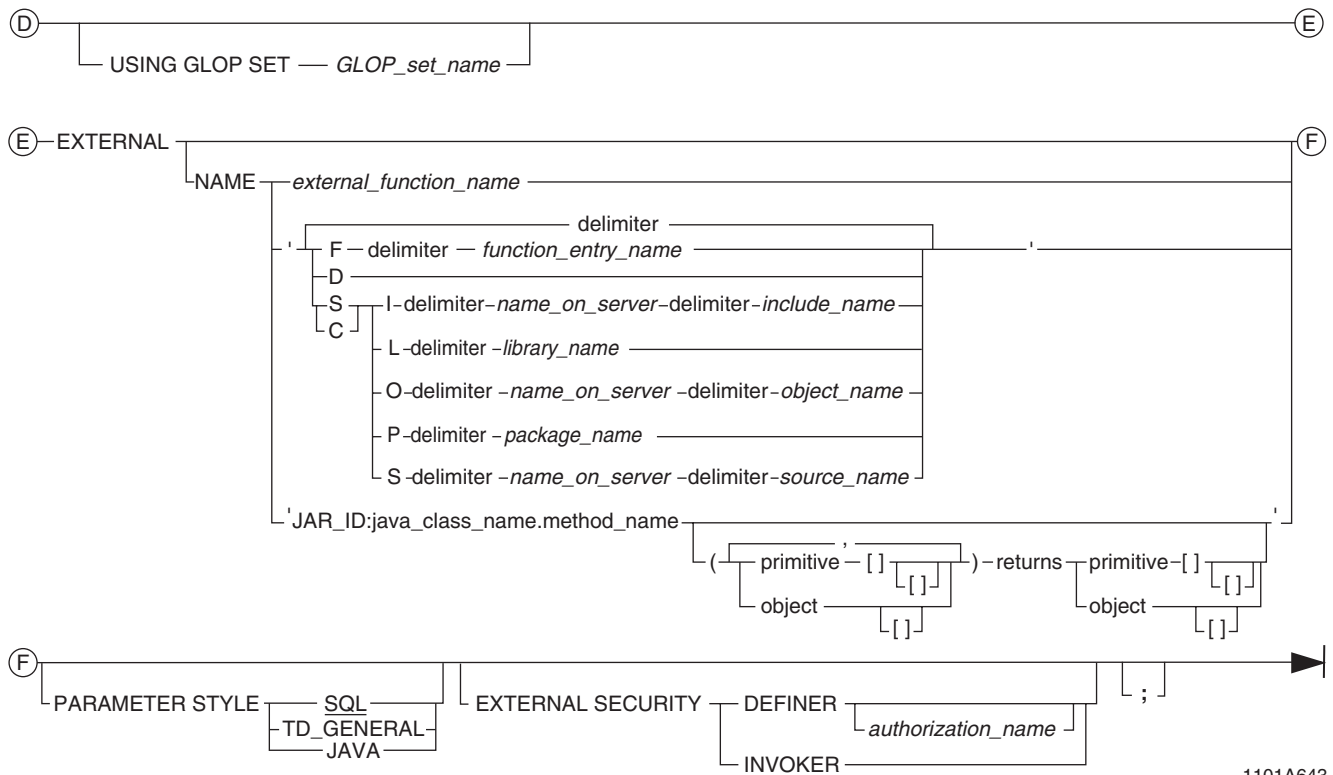


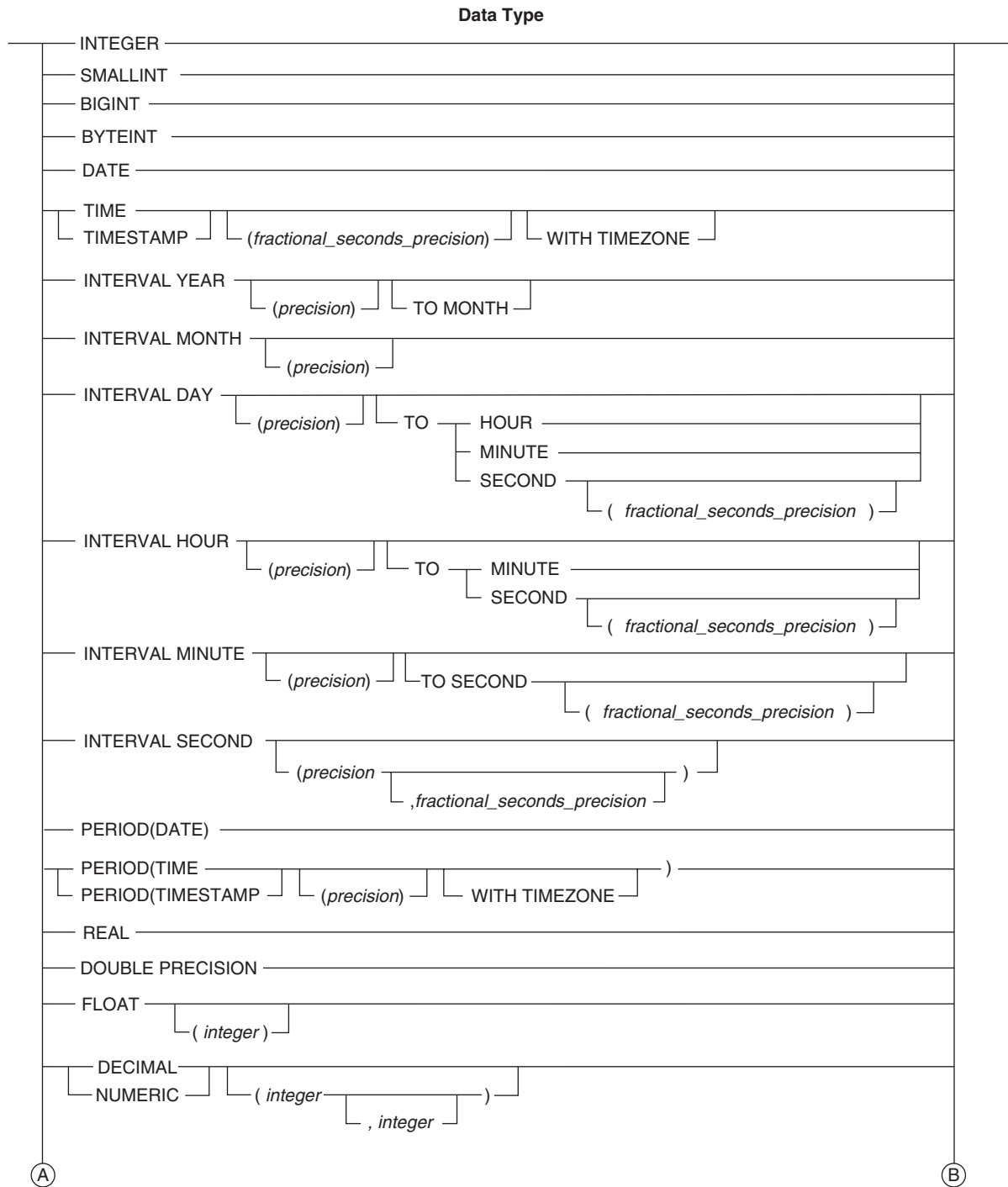
1101B436

CREATE FUNCTION/ REPLACE FUNCTION

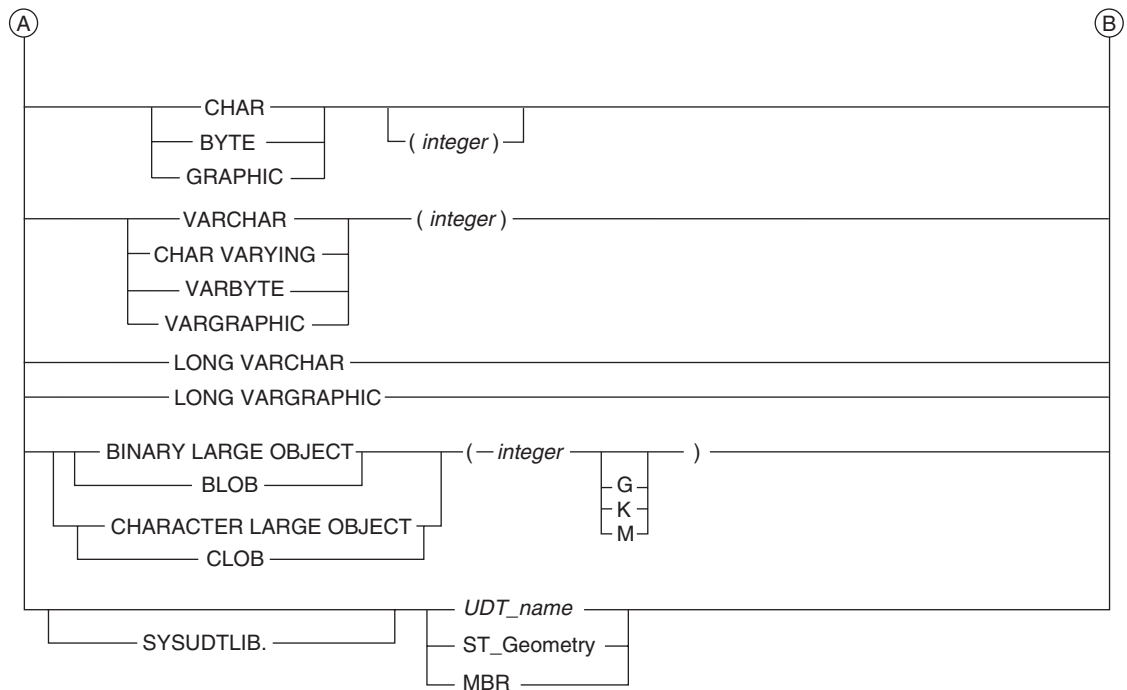


1101C323





1101A535

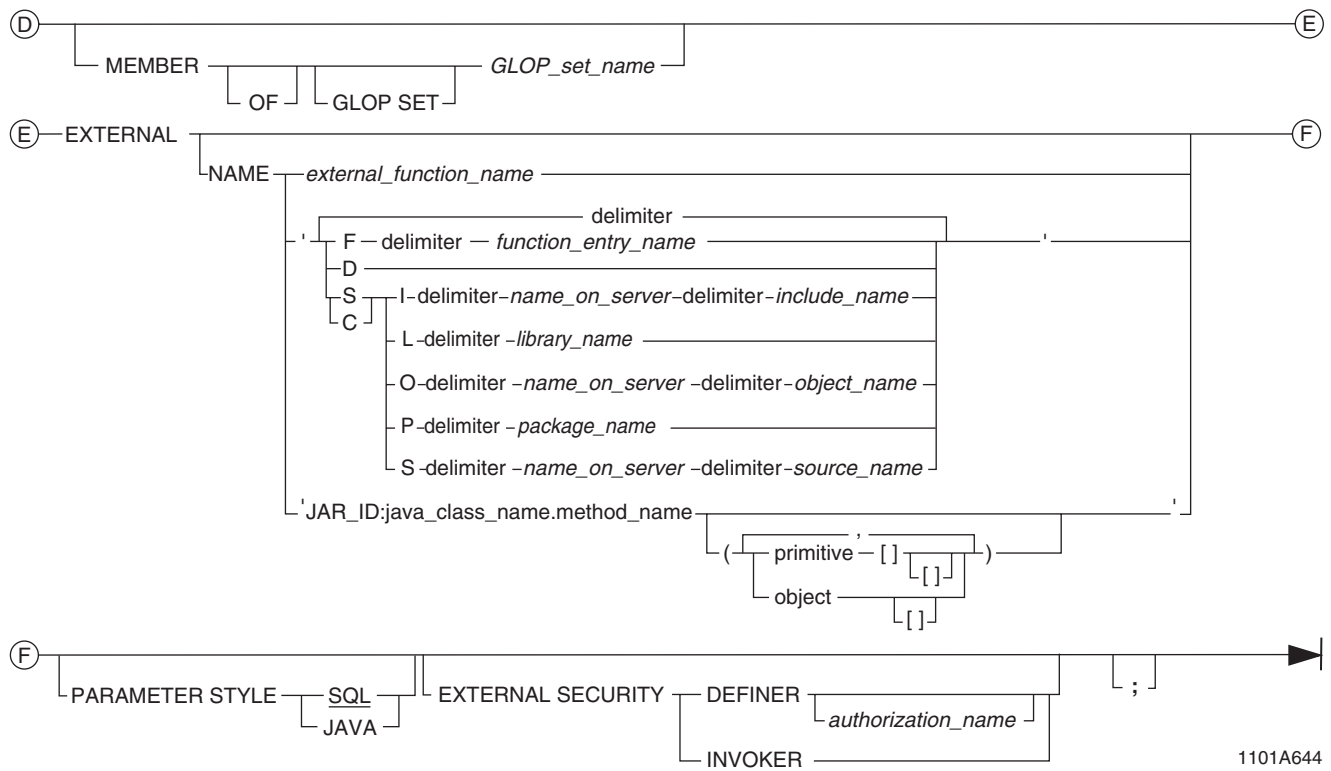


1101A536

CREATE FUNCTION (Table Form)

CREATE	FUNCTION	<div>database_name. user_name.</div>	function_name	(A)	
REPLACE					
(A)	(<div>parameter_name</div>	data type)	(B)
(B)	RETURNS	TABLE	(<div>column_name</div> data_type)	(C)	
			VARYING COLUMNS (— maximum_output_columns —)		
(C)	language_clause — SQL_data_access	SQL_data_access — language_clause	<div> <div>SPECIFIC <div>database_name. user_name.</div> specific_function_name</div> <div>PARAMETER STYLE <div>SQL JAVA</div></div> <div><div>NOT</div> DETERMINISTIC</div> <div>CALLED ON NULL INPUT</div> </div>	(D)	
			<div> <div>SPECIFIC <div>database_name. user_name.</div> specific_function_name</div> <div>PARAMETER STYLE <div>SQL JAVA</div></div> <div><div>NOT</div> DETERMINISTIC</div> <div>CALLED ON NULL INPUT</div> </div>		
			language_clause — SQL_data_access SQL_data_access — language_clause		

1101E228



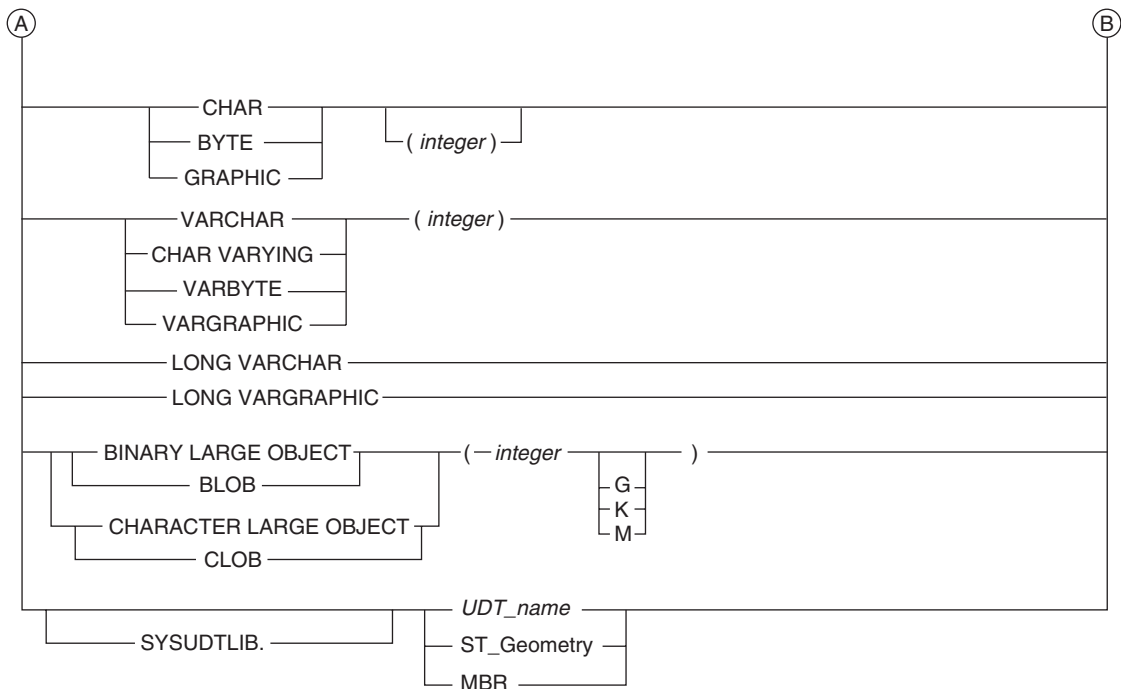
1101A644

Data Type	
INTEGER	
SMALLINT	
BIGINT	
BYTEINT	
DATE	
TIME	
TIMESTAMP	(fractional_seconds_precision) WITH TIMEZONE
INTERVAL YEAR	(precision) TO MONTH
INTERVAL MONTH	(precision)
INTERVAL DAY	(precision) TO HOUR MINUTE SECOND (fractional_seconds_precision)
INTERVAL HOUR	(precision) TO MINUTE SECOND (fractional_seconds_precision)
INTERVAL MINUTE	(precision) TO SECOND (fractional_seconds_precision)
INTERVAL SECOND	(precision, fractional_seconds_precision)
PERIOD(DATE)	
PERIOD(TIME)	
PERIOD(TIMESTAMP)	(precision) WITH TIMEZONE
REAL	
DOUBLE PRECISION	
FLOAT	(integer)
DECIMAL	
NUMERIC	(integer, integer)

(A)

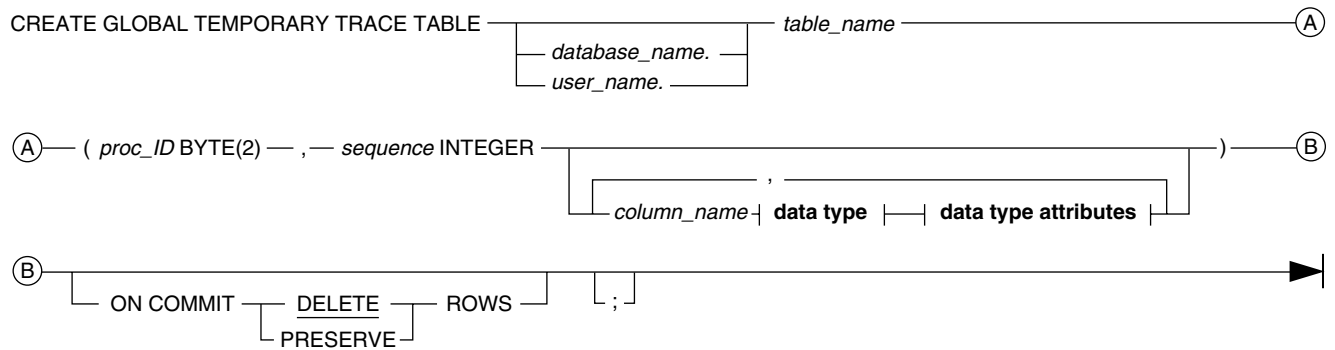
(B)

1101A535

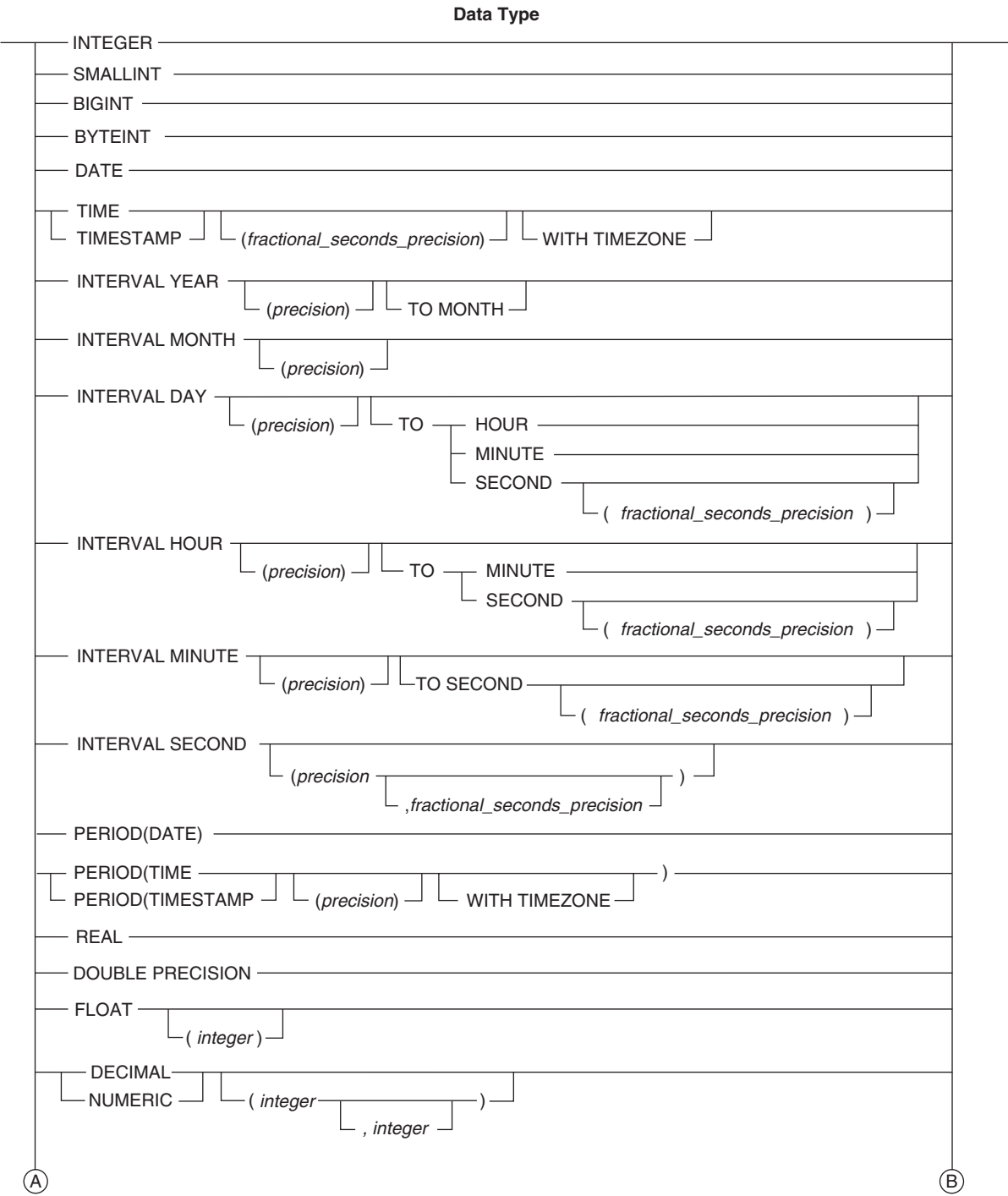


1101A536

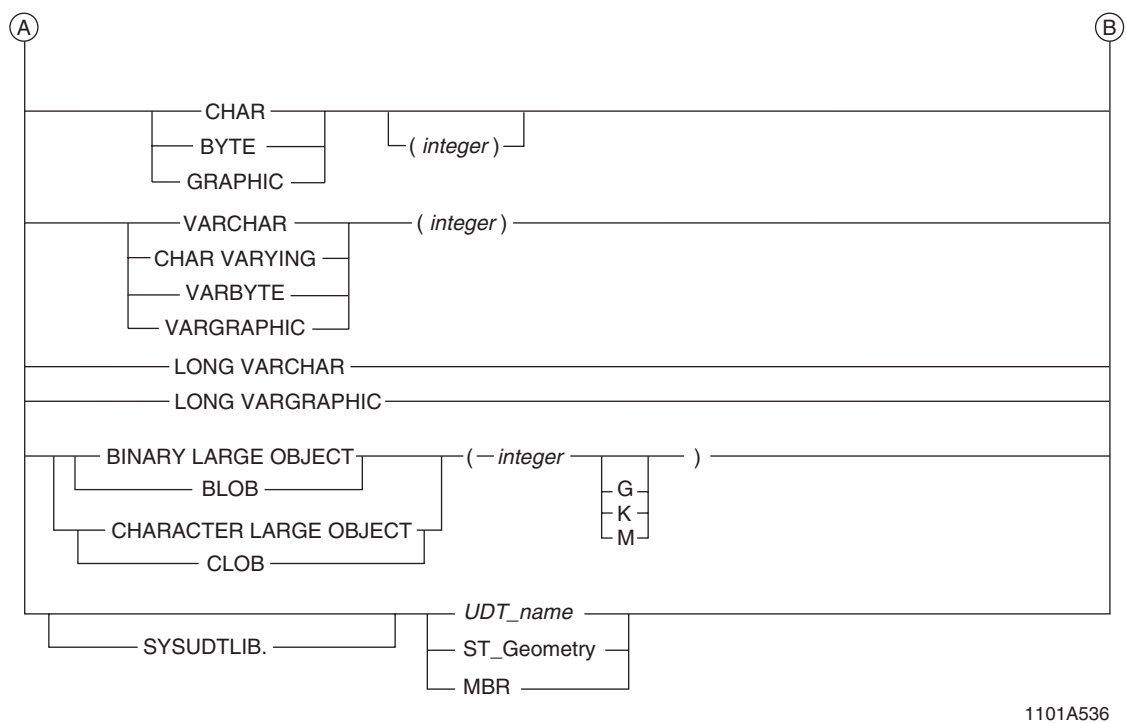
CREATE GLOBAL TEMPORARY TRACE TABLE



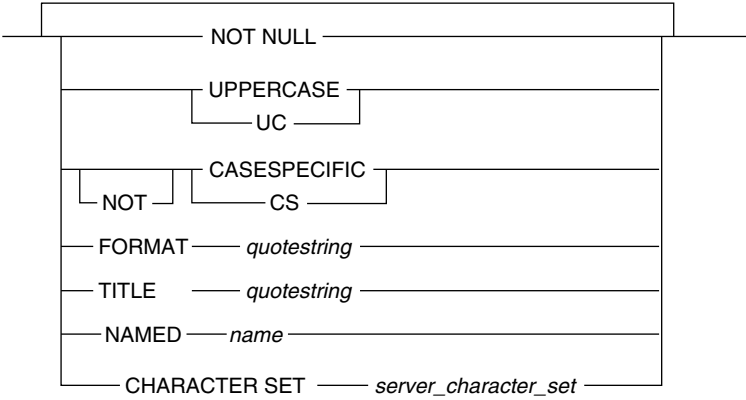
1101B534



1101A535

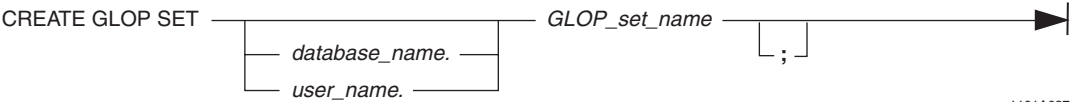


Data Type Attributes



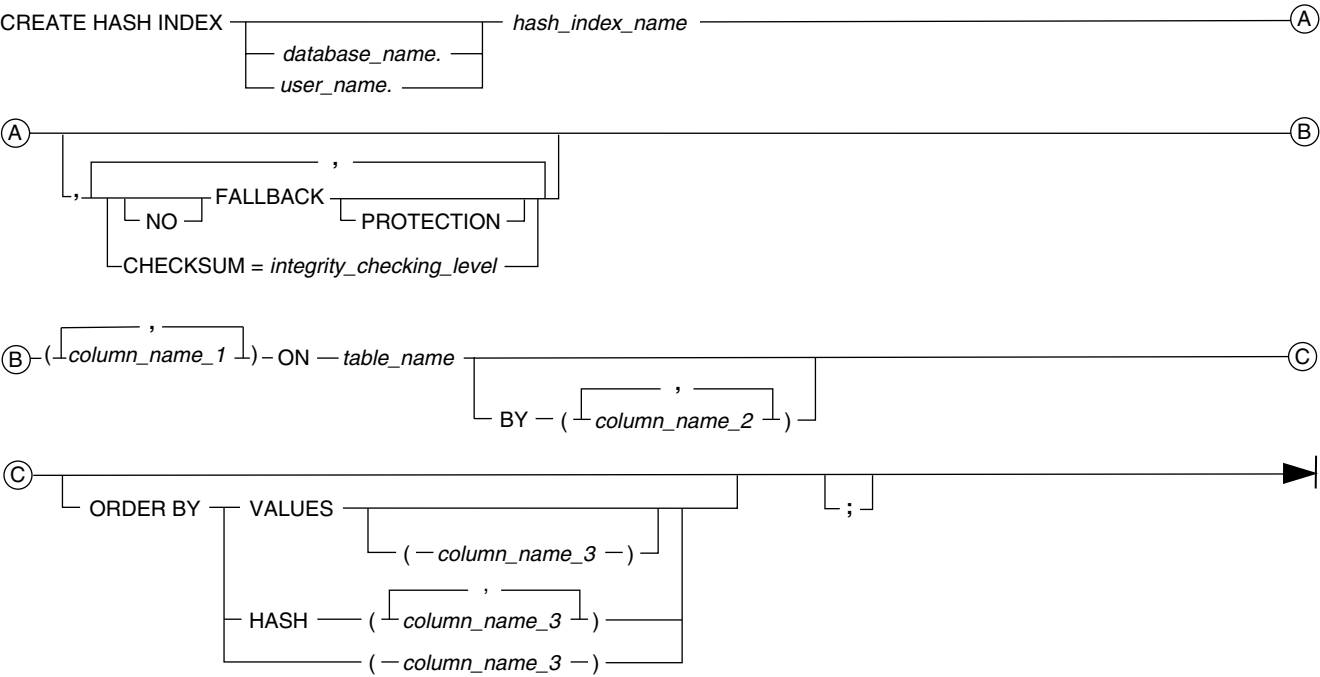
1101B115

CREATE GLOP SET



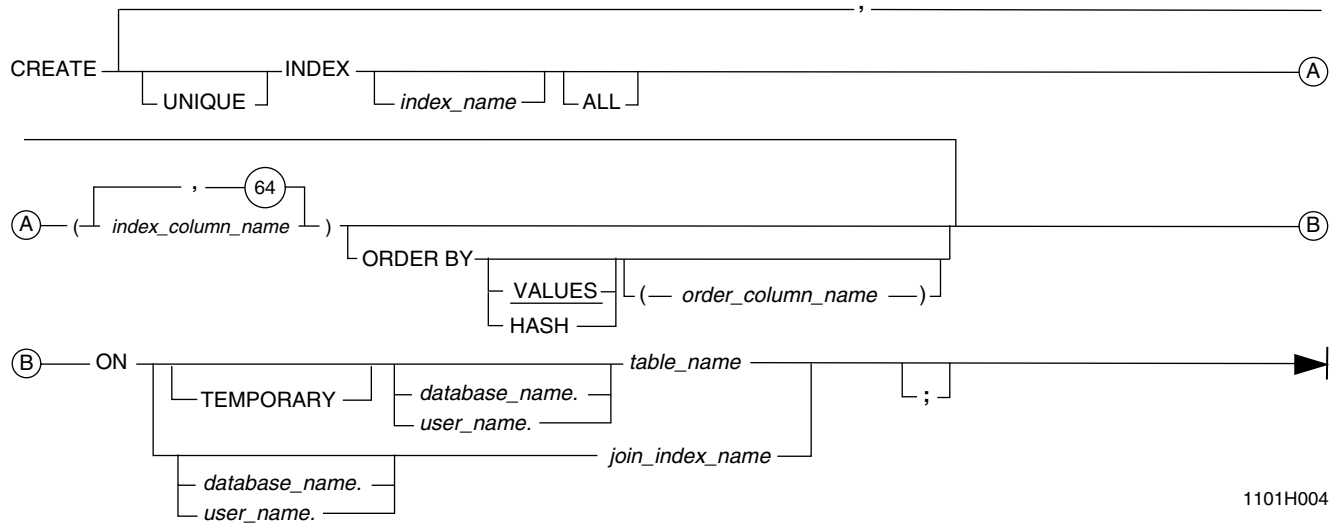
1101A637

CREATE HASH INDEX

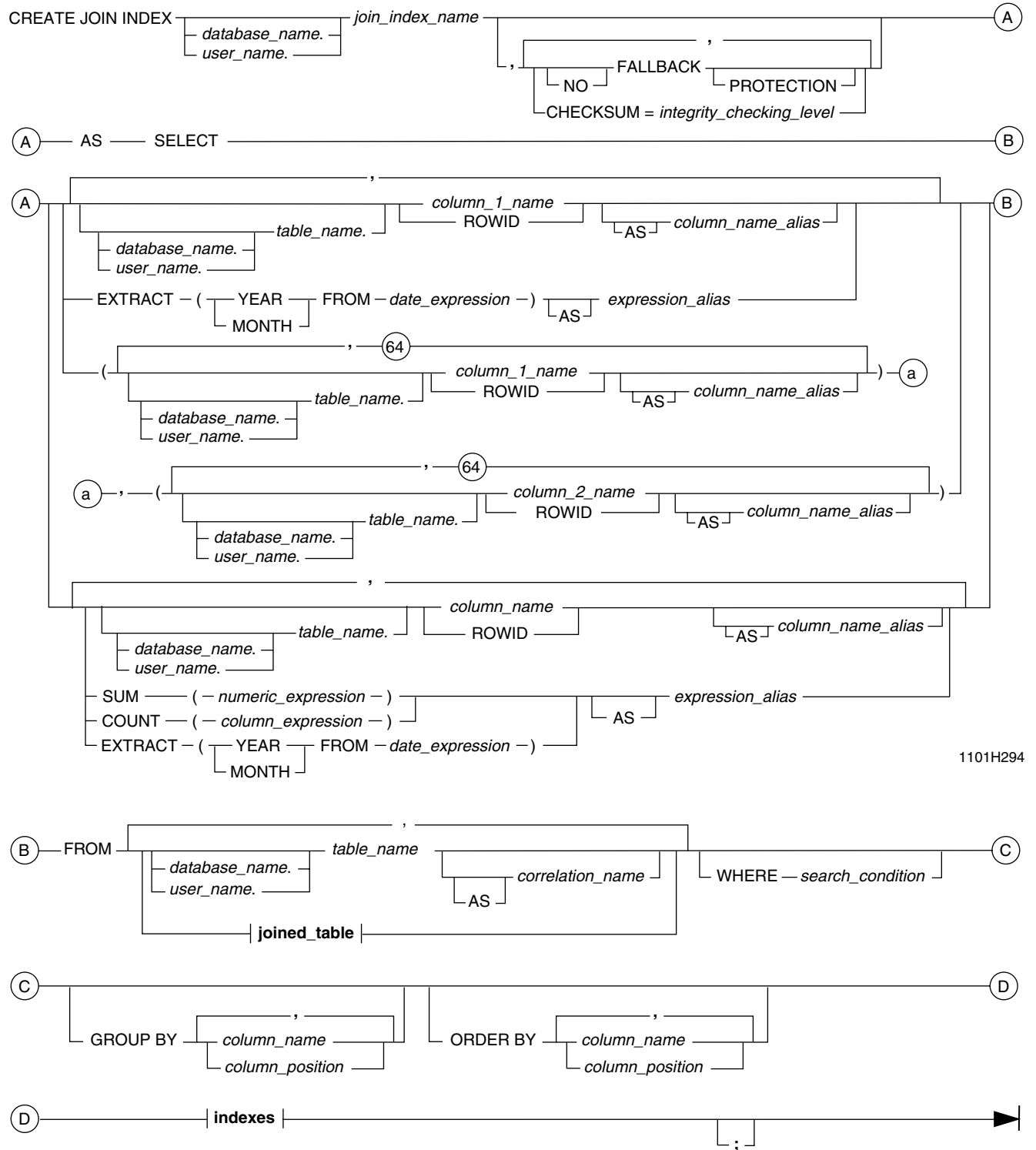


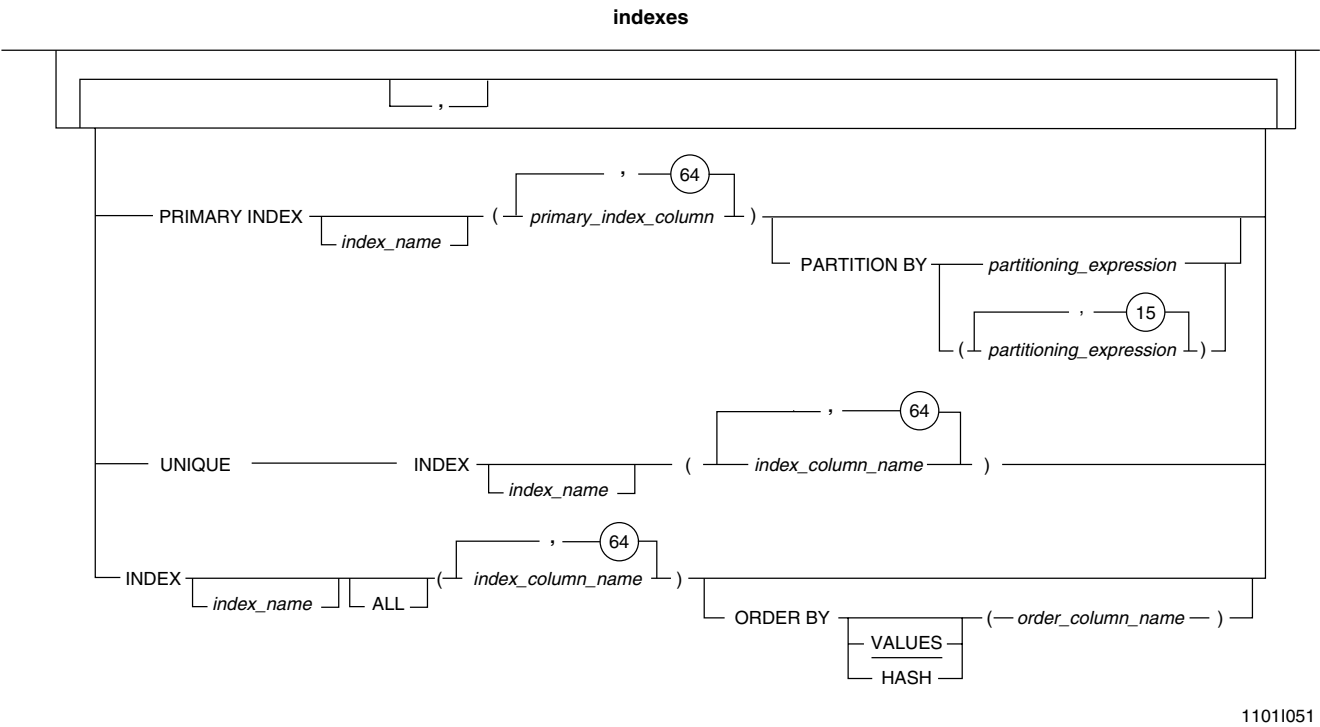
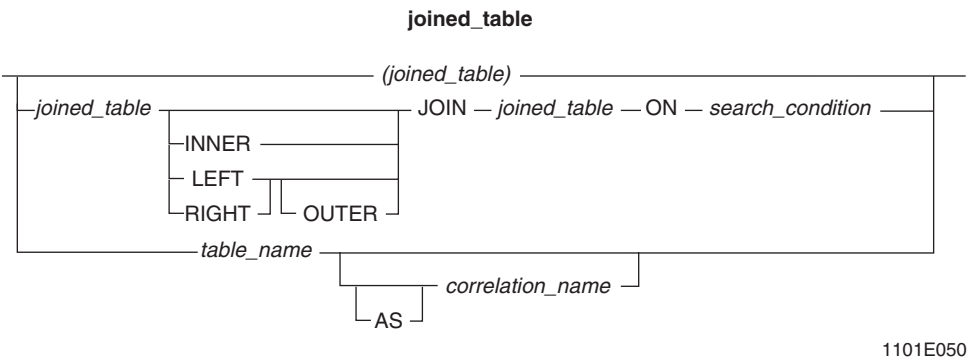
1101B431

CREATE INDEX

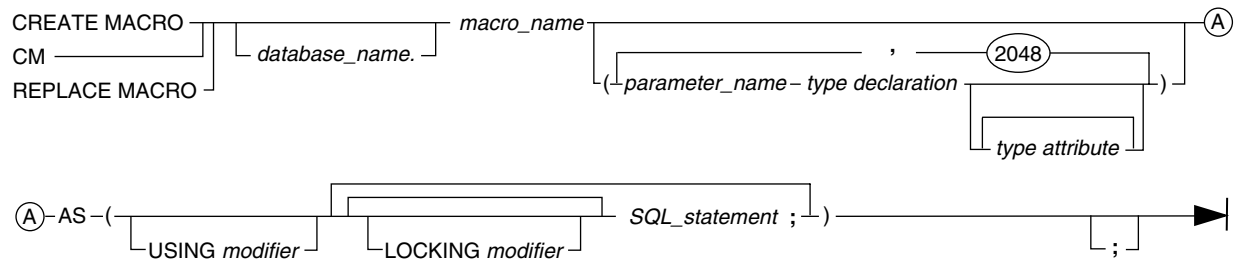


CREATE JOIN INDEX



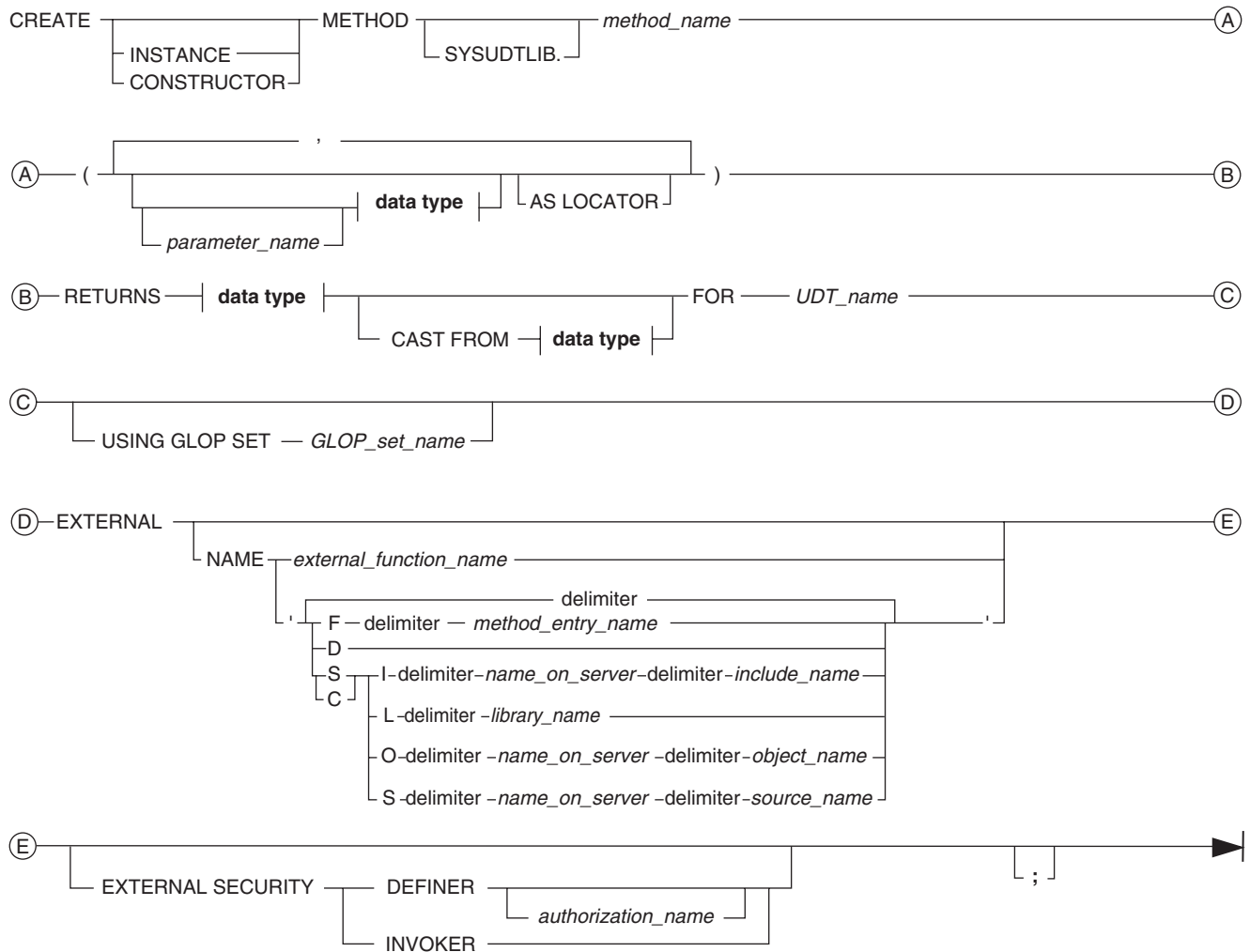


CREATE MACRO/ REPLACE MACRO

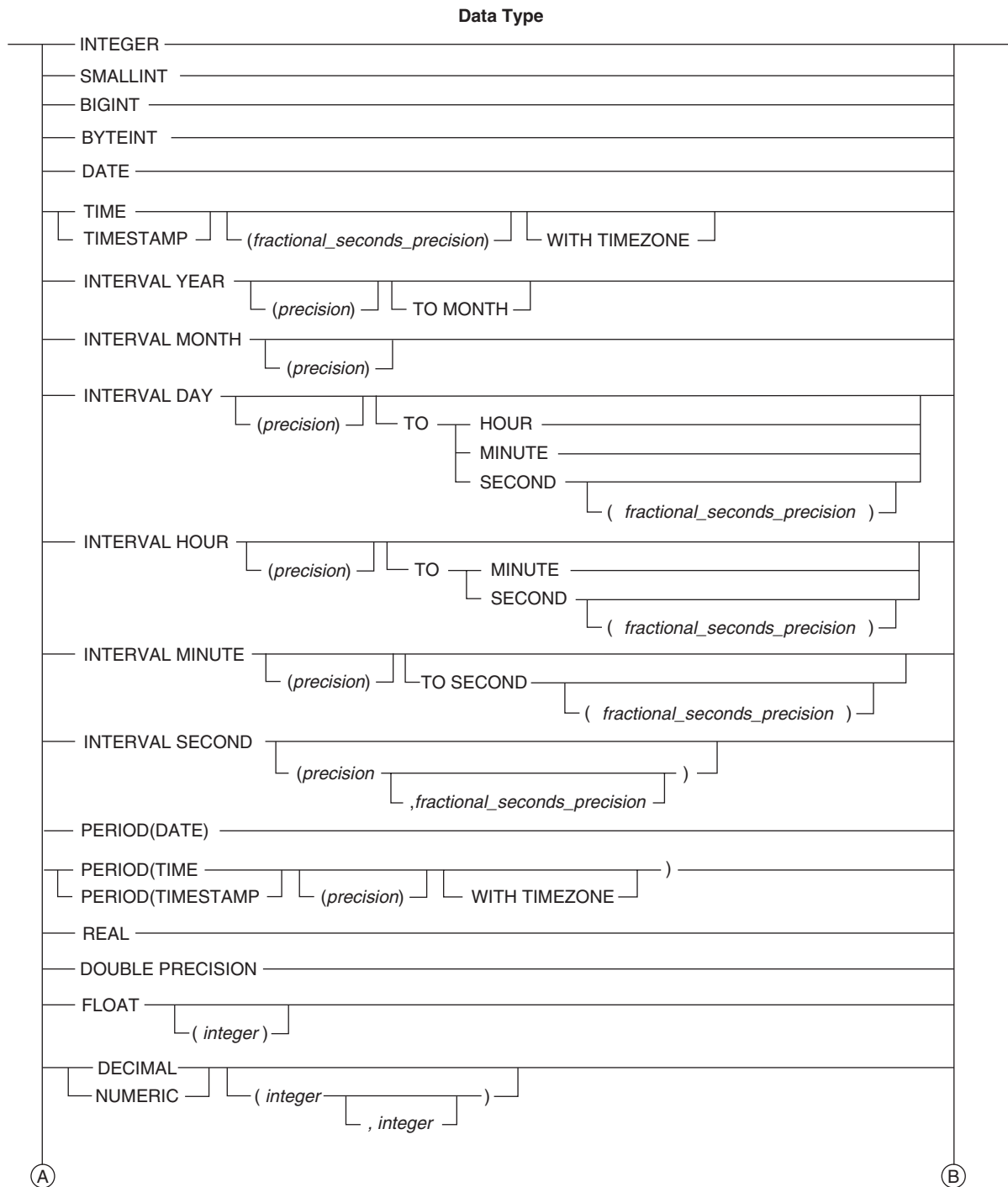


1101G172

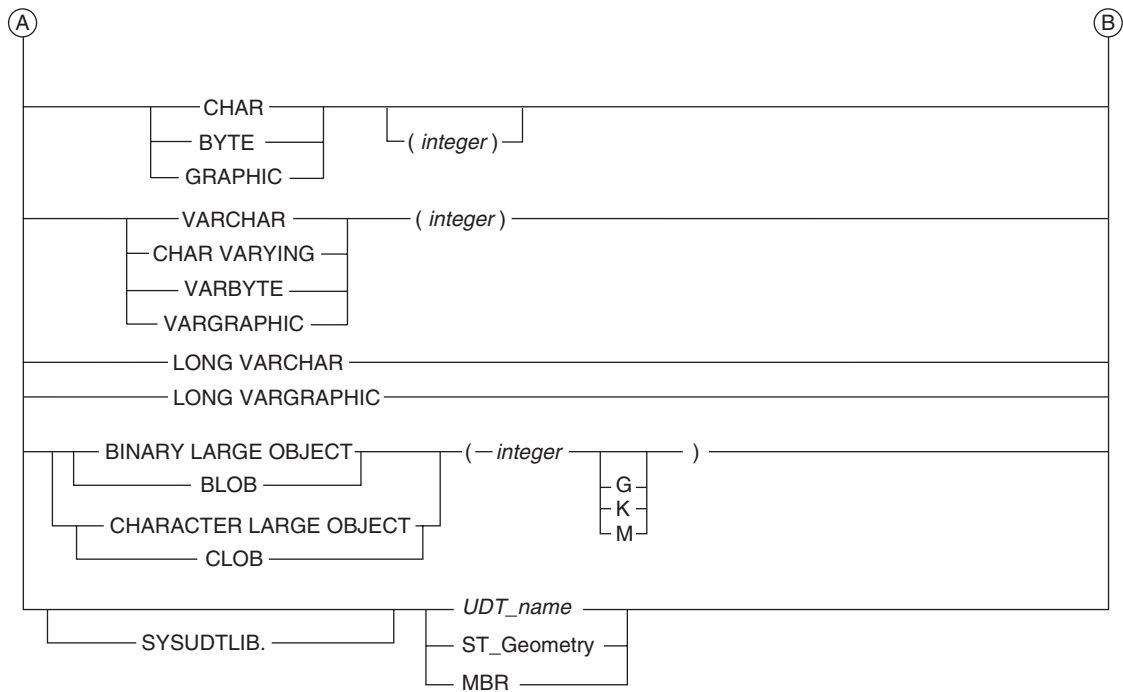
CREATE METHOD



1101C371

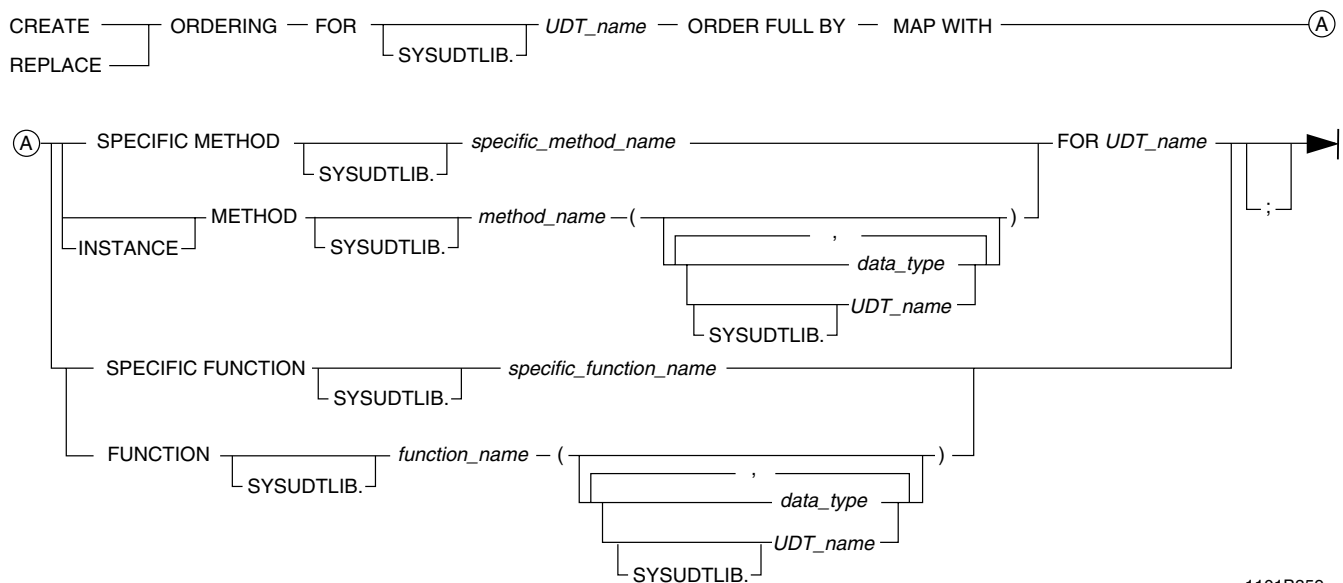


1101A535



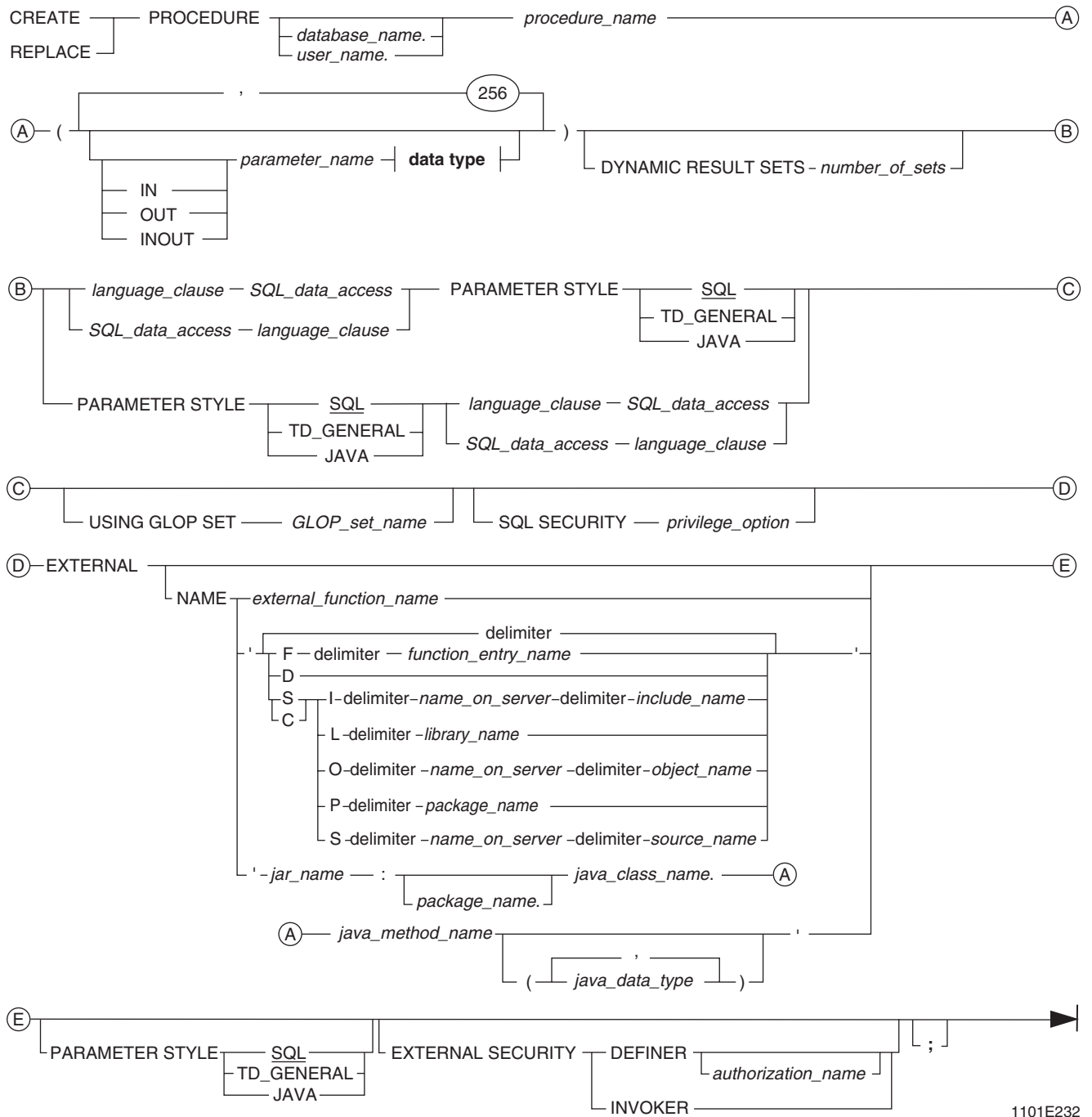
1101A536

CREATE ORDERING/ REPLACE ORDERING

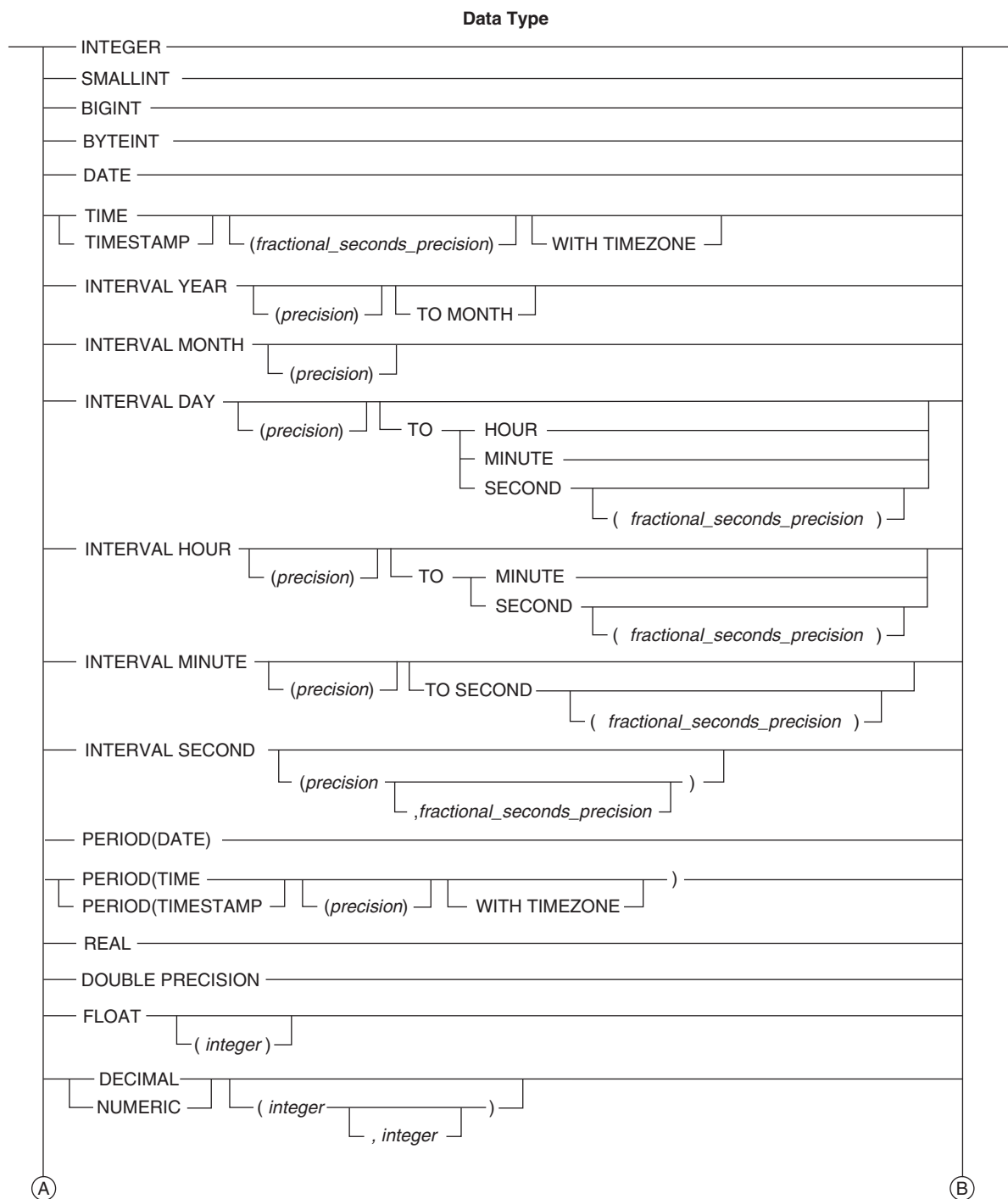


1101B359

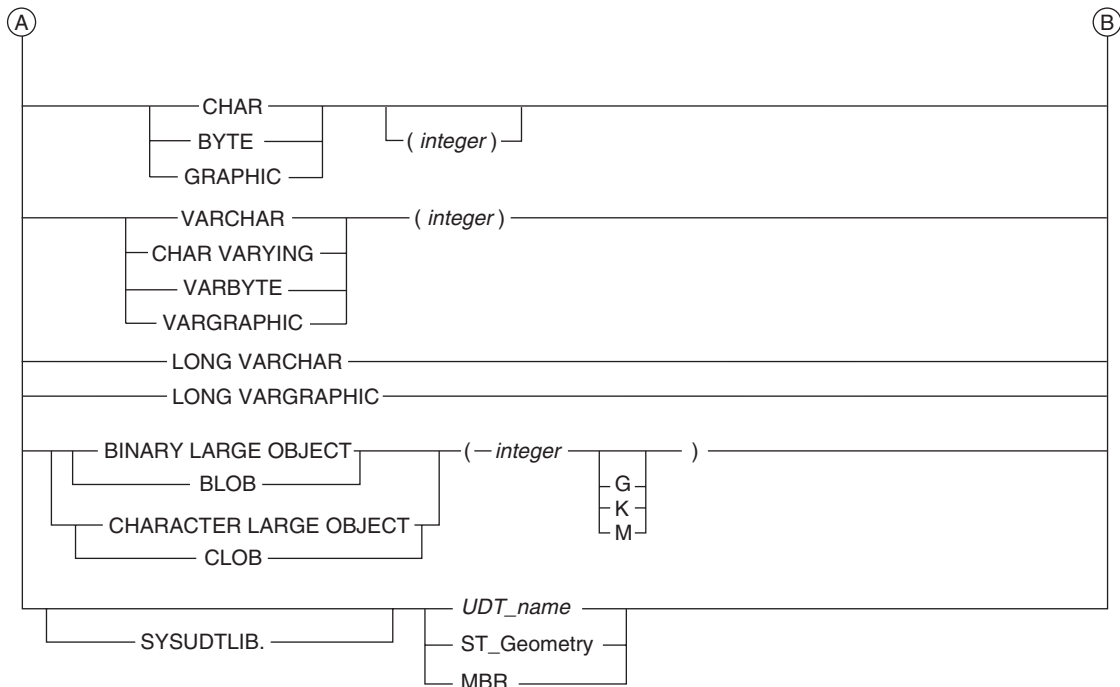
CREATE PROCEDURE (External Form)/ REPLACE PROCEDURE (External Form)



1101E232

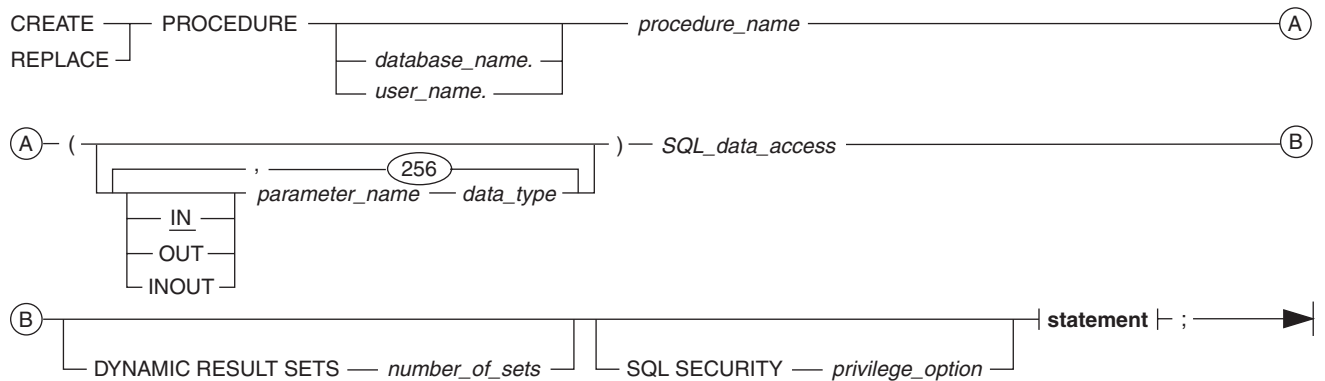


1101A535

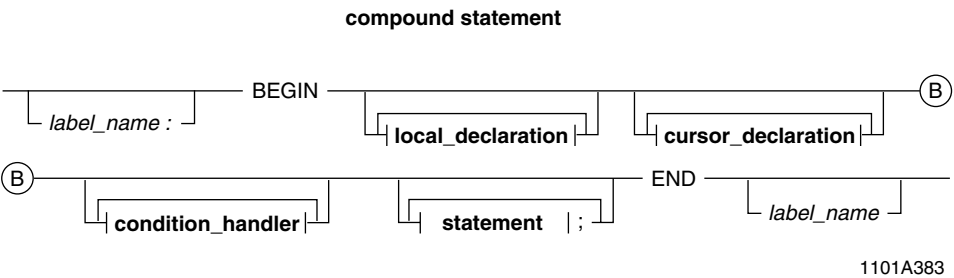
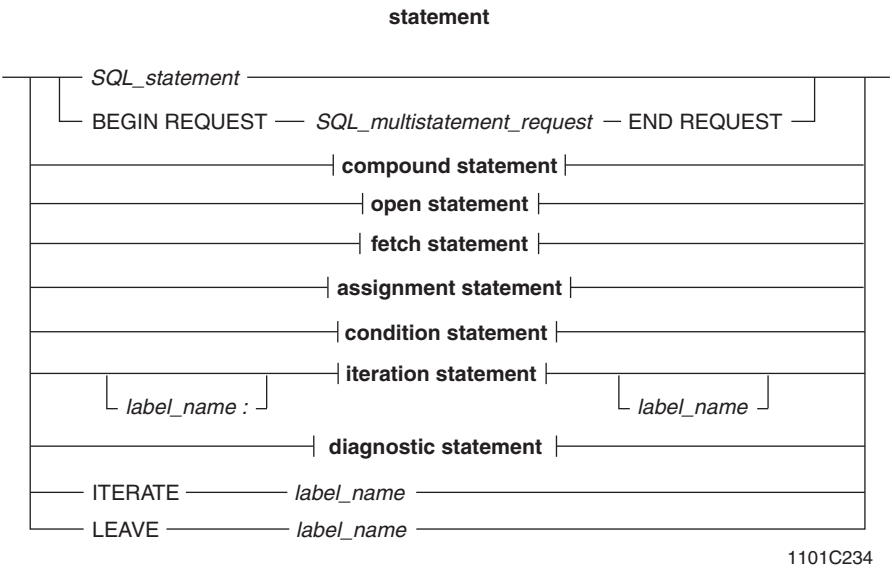


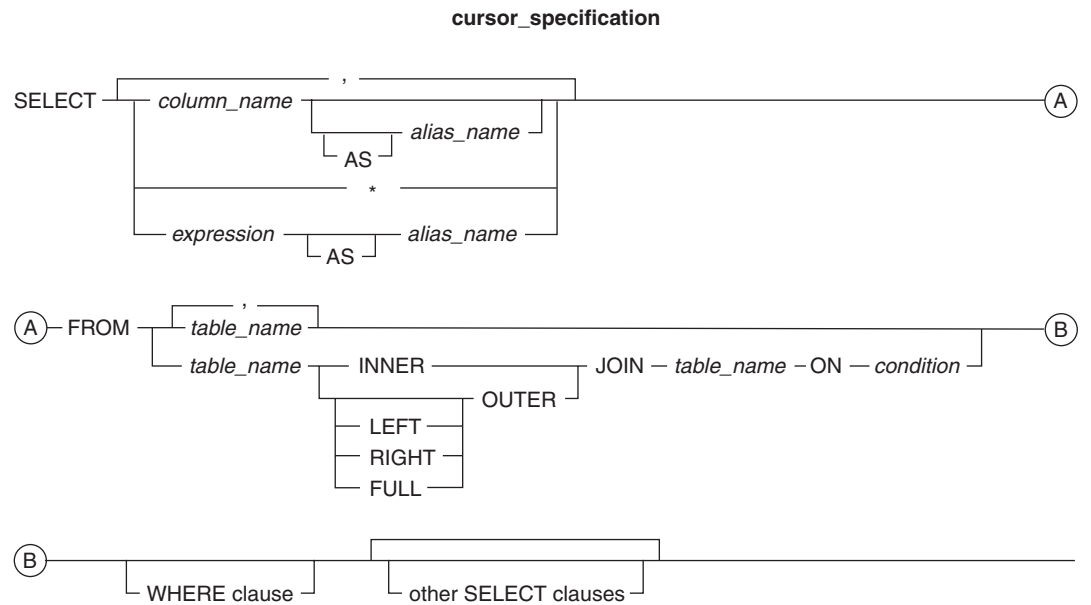
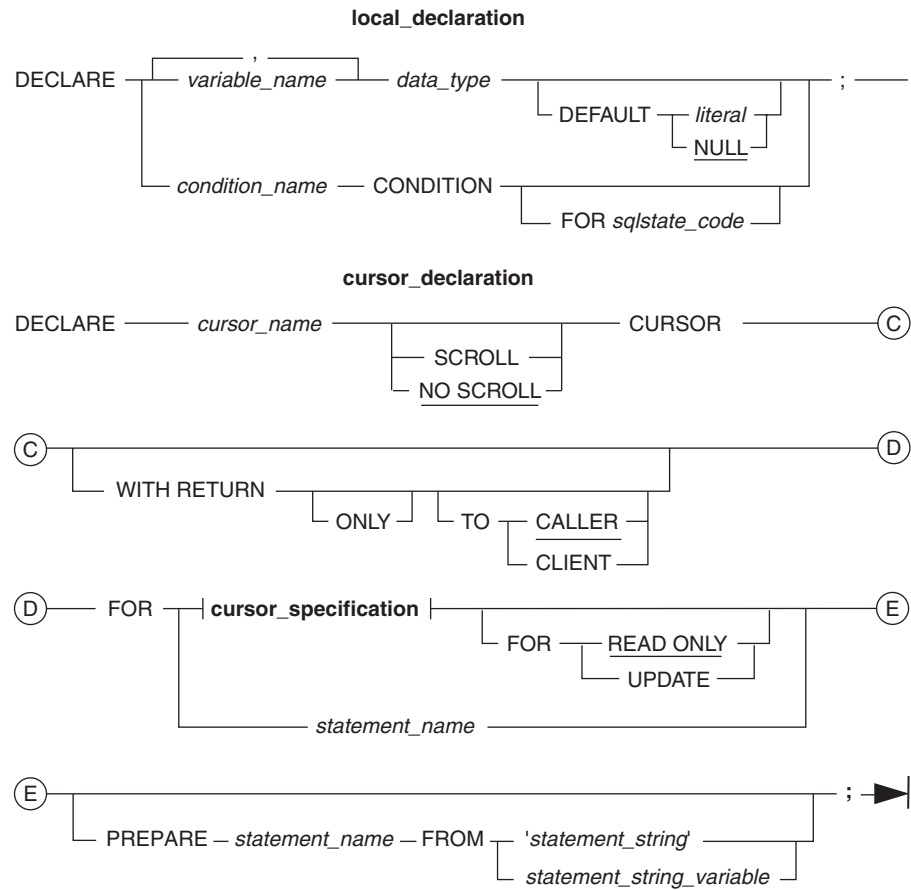
1101A536

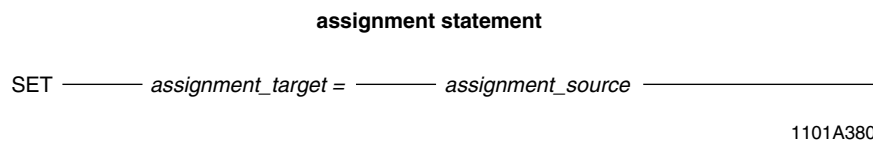
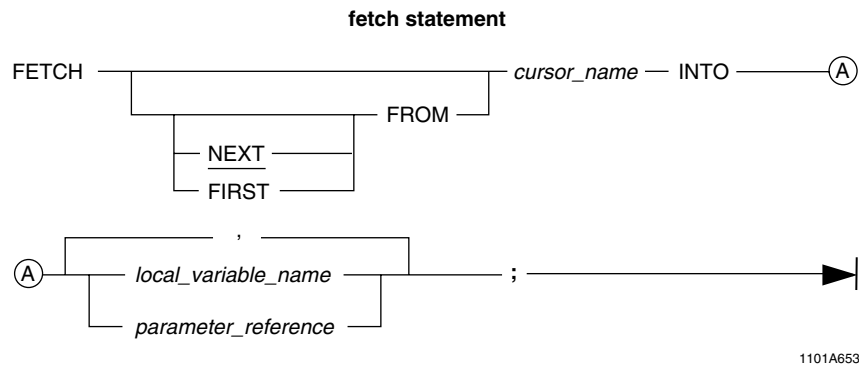
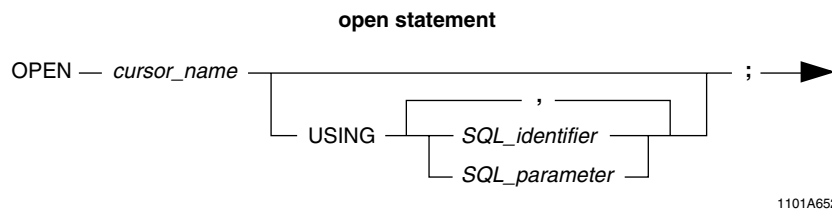
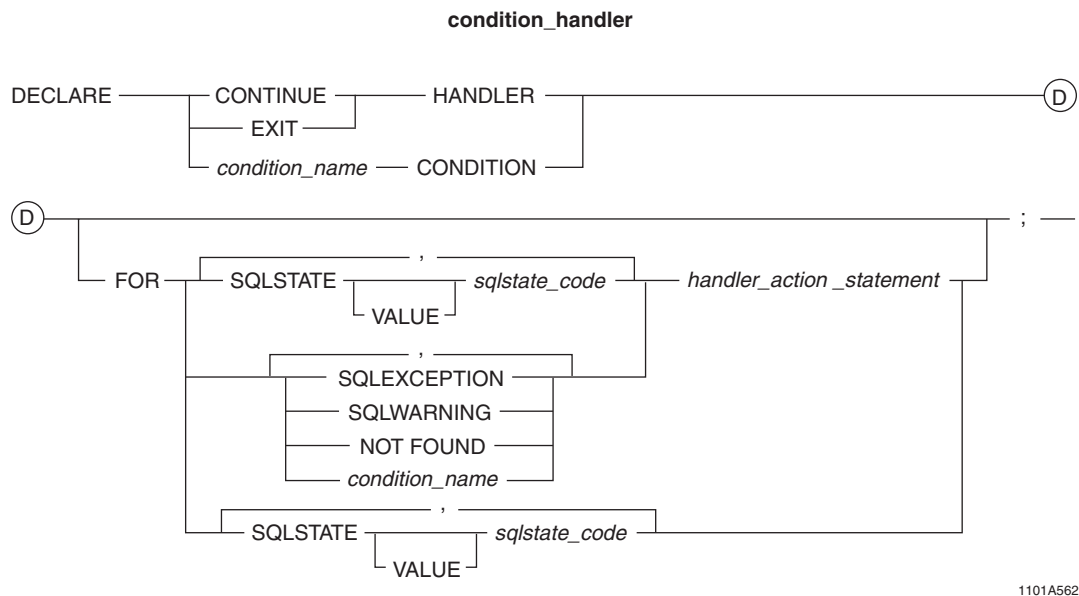
CREATE PROCEDURE (SQL Form)/ REPLACE PROCEDURE



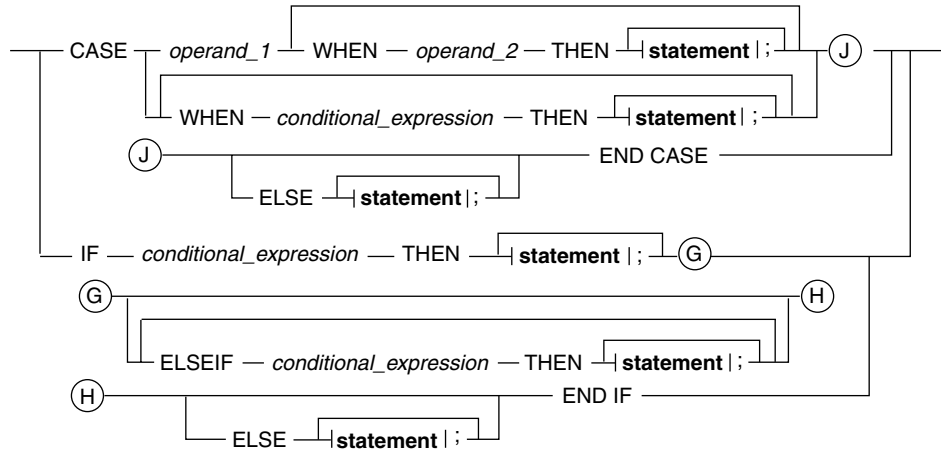
1101T002





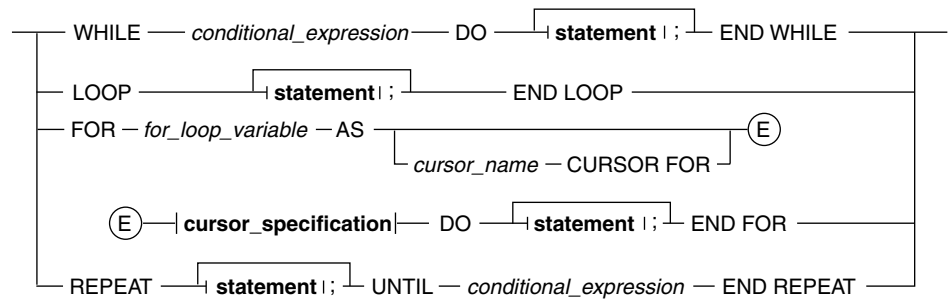


condition statement



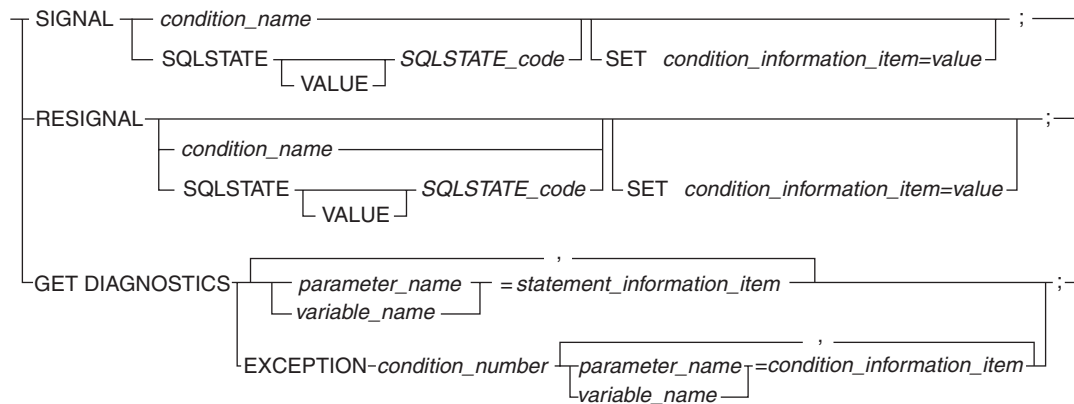
1101A381

iteration statement

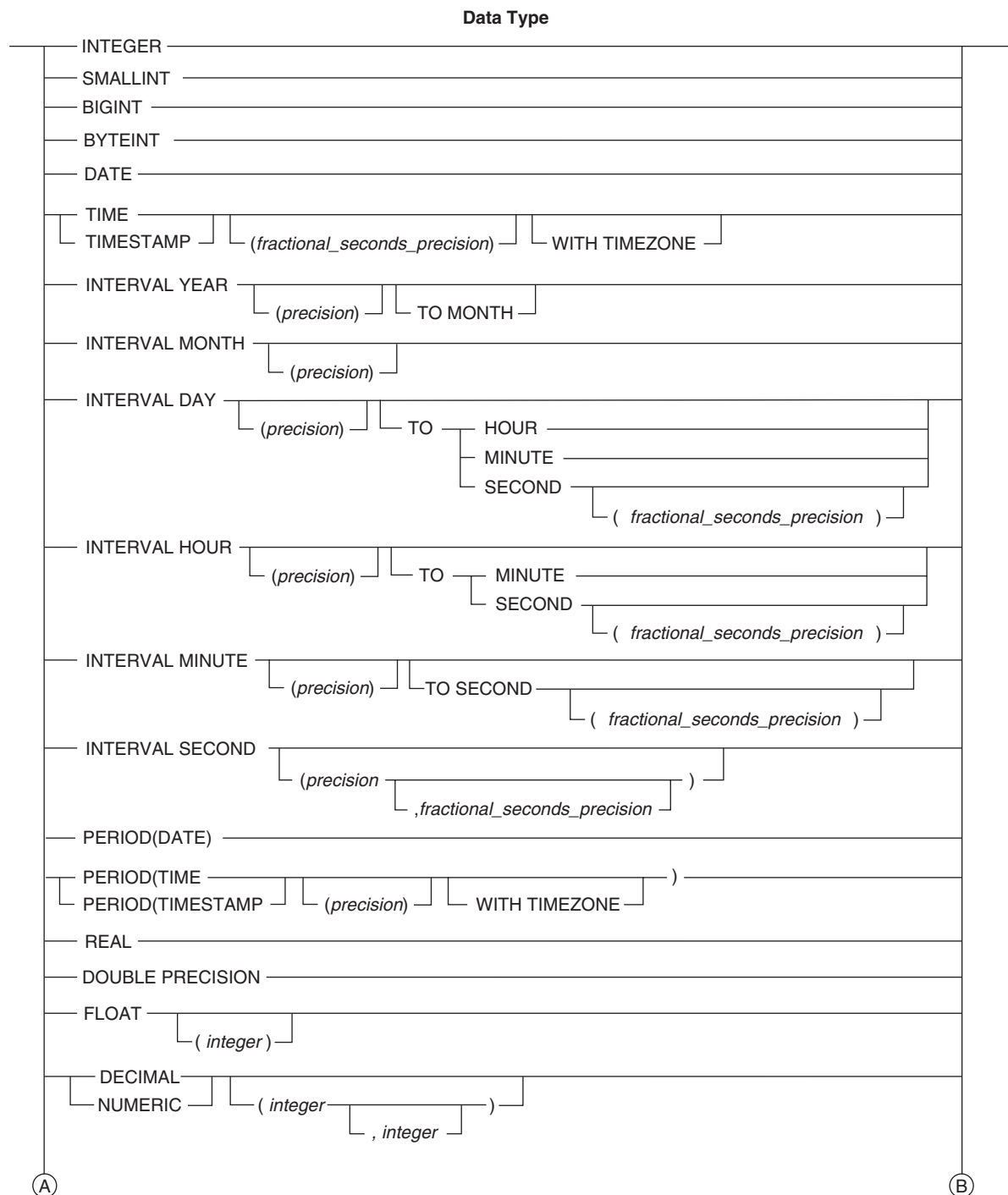


1101A382

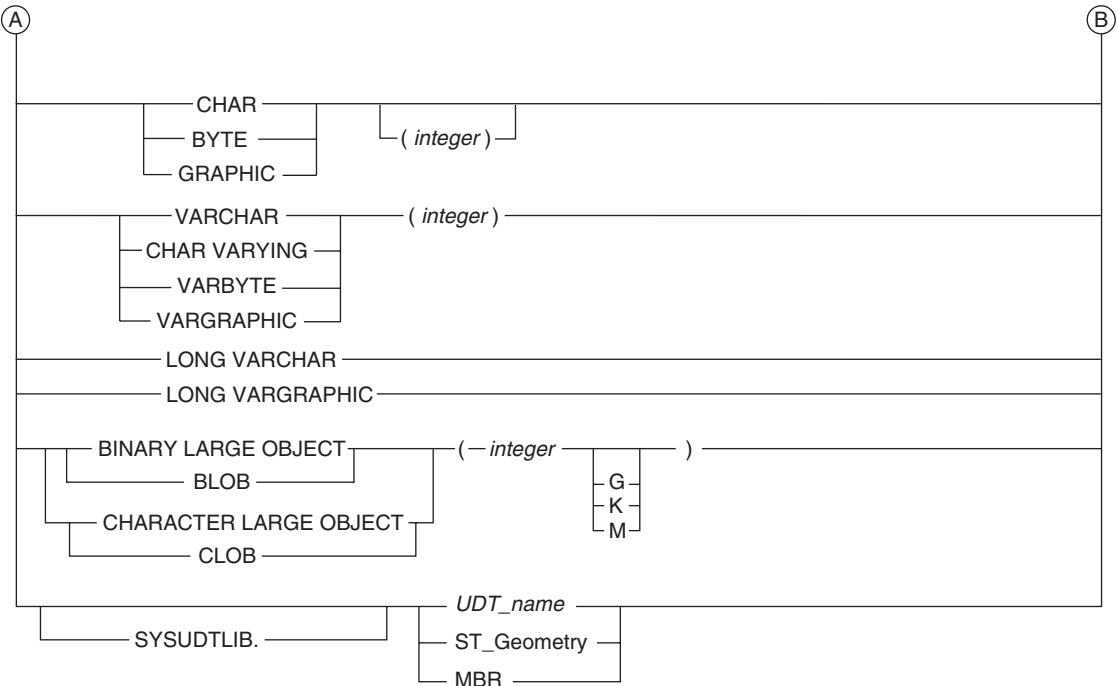
diagnostic statement



1101A616

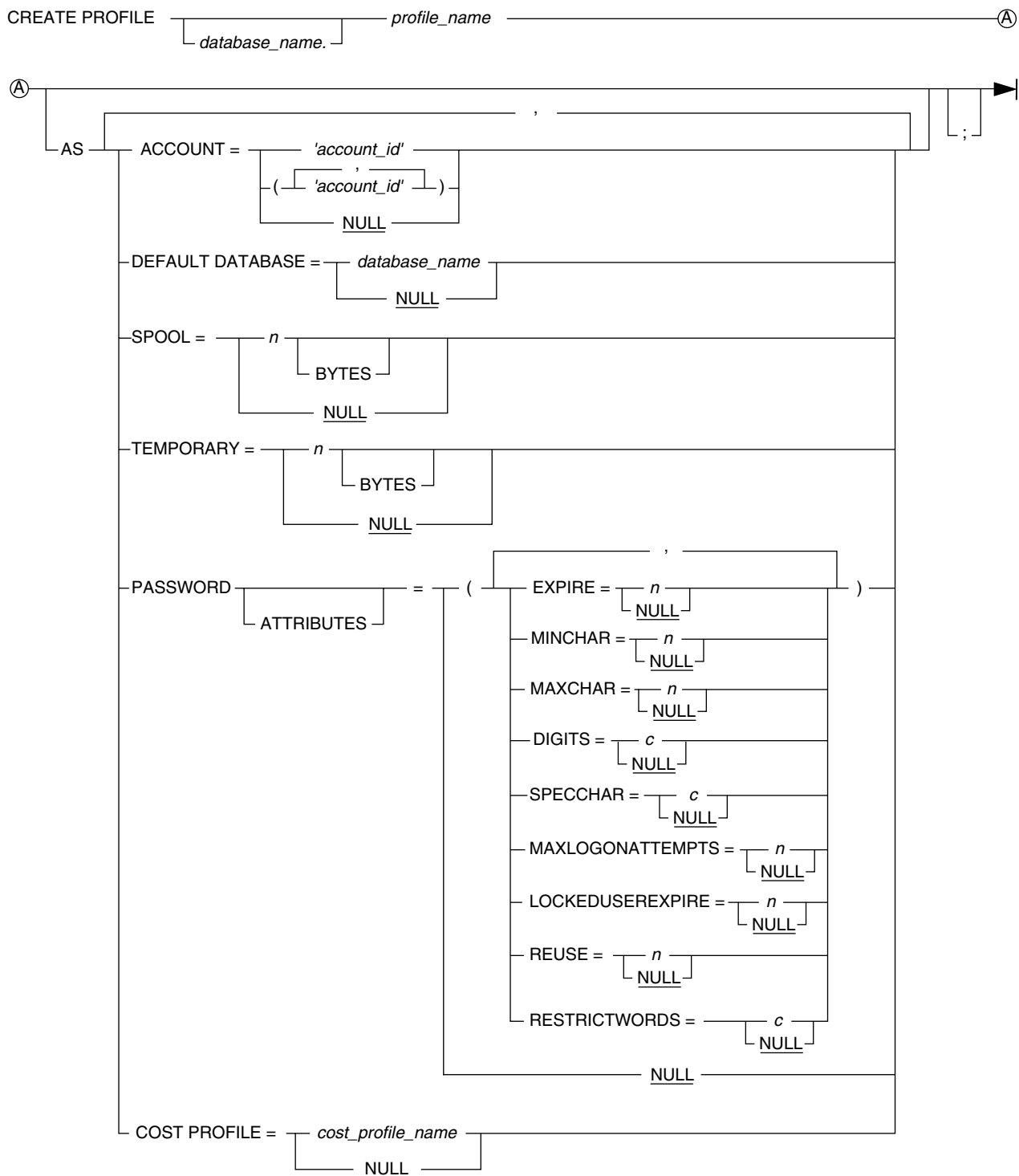


1101A535



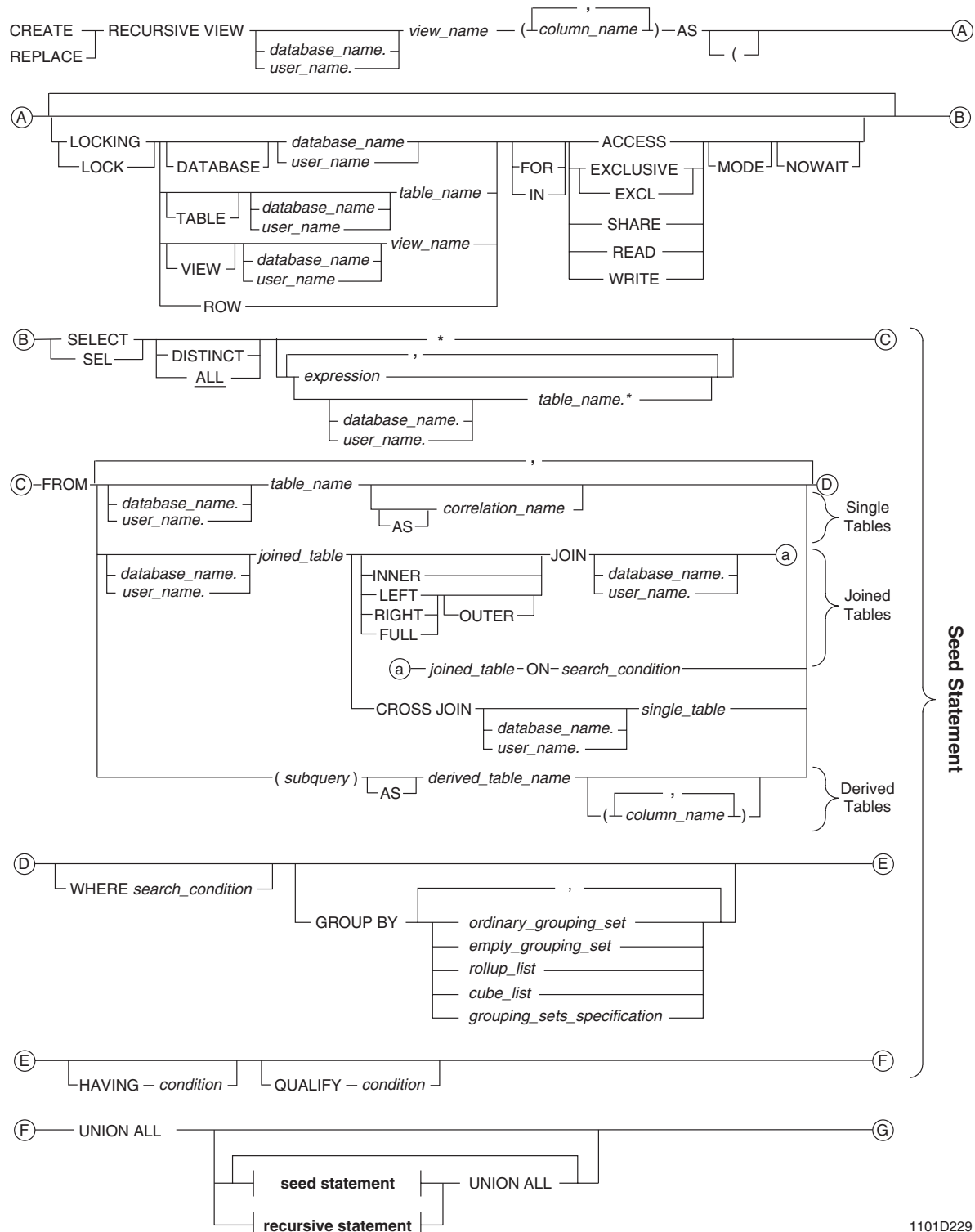
1101A536

CREATE PROFILE

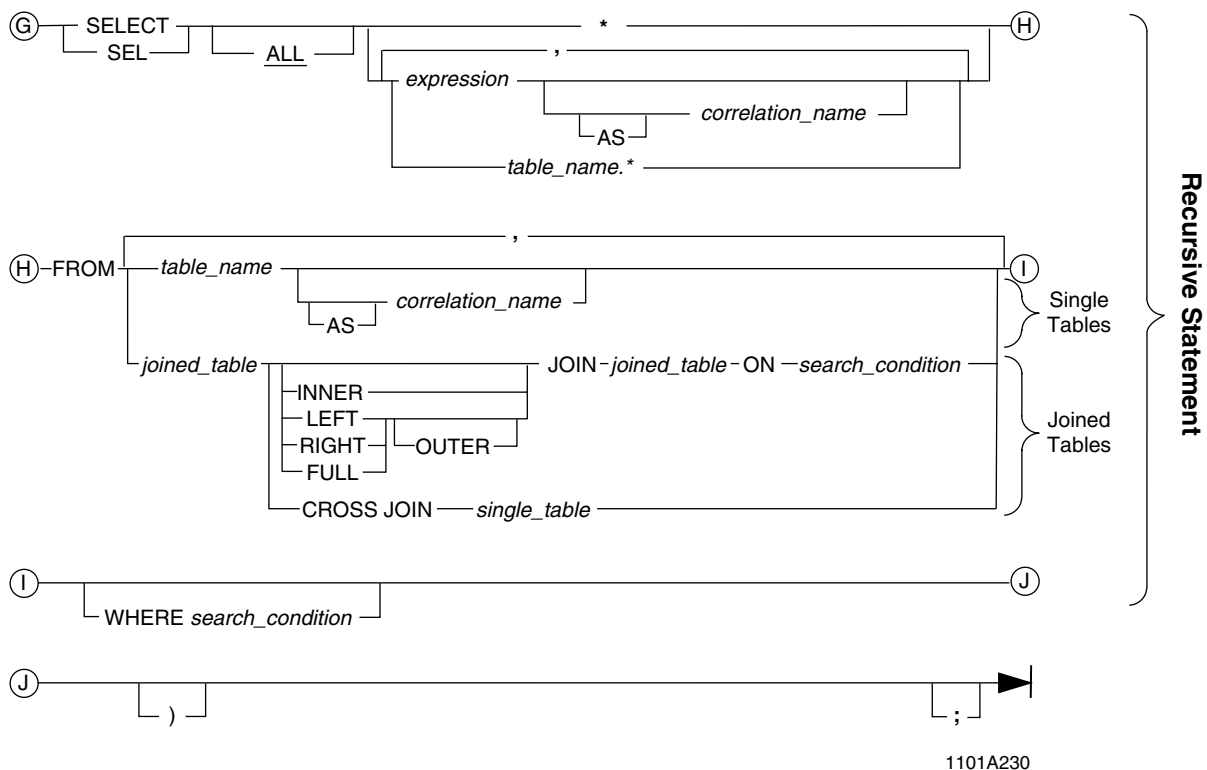


1101B466

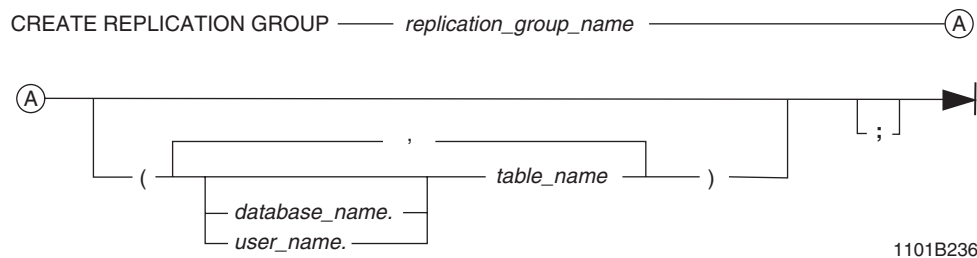
CREATE RECURSIVE VIEW/ REPLACE RECURSIVE VIEW



1101D229



CREATE REPLICATION GROUP



CREATE REPLICATION RULESET/ REPLACE REPLICATION RULESET

CREATE REPLICATION RULESET *rule_set_name* FOR *replication_group_name* (A)
REPLACE , DEFAULT ; ▶

(A) AS - *object_kind* - LIKE - *string_literal* ESCAPE - *character_literal* AND NOT LIKE - *string_literal* ESCAPE - *character_literal* ; ▶

1101A553

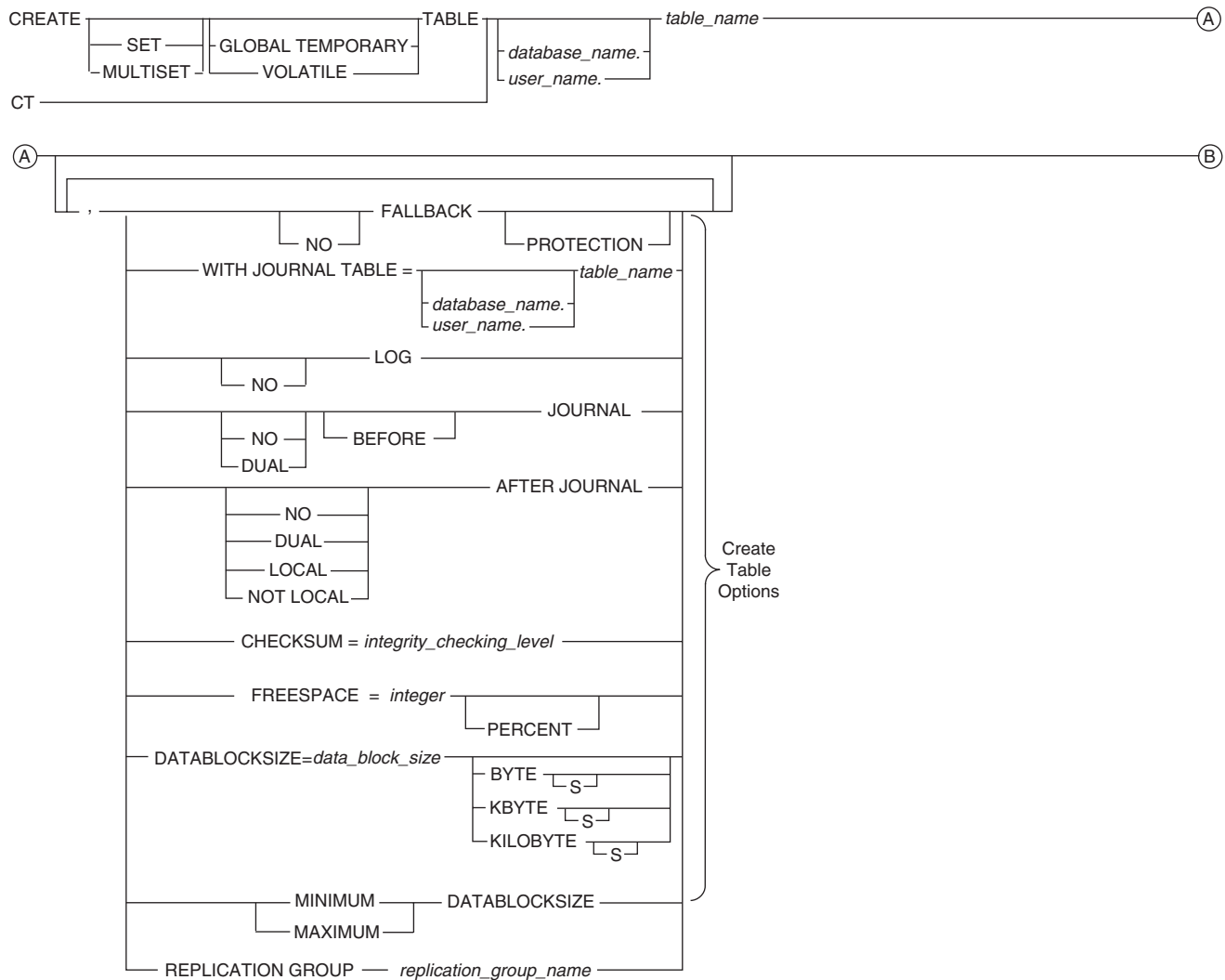
CREATE ROLE

CREATE EXTERNAL ROLE [*database_name*] *role_name* ; ▶

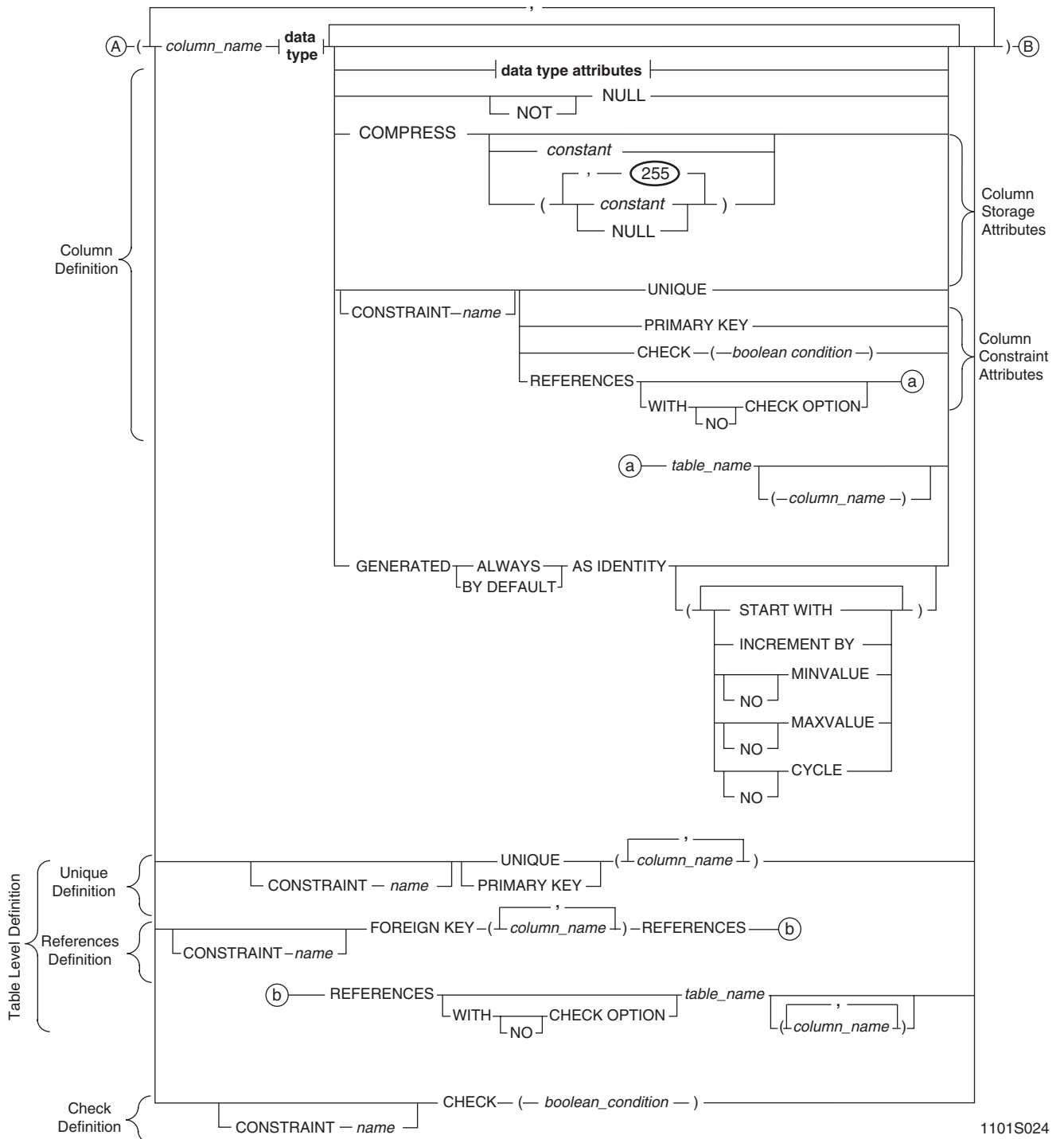
1101B288

CREATE TABLE

Create Table Syntax

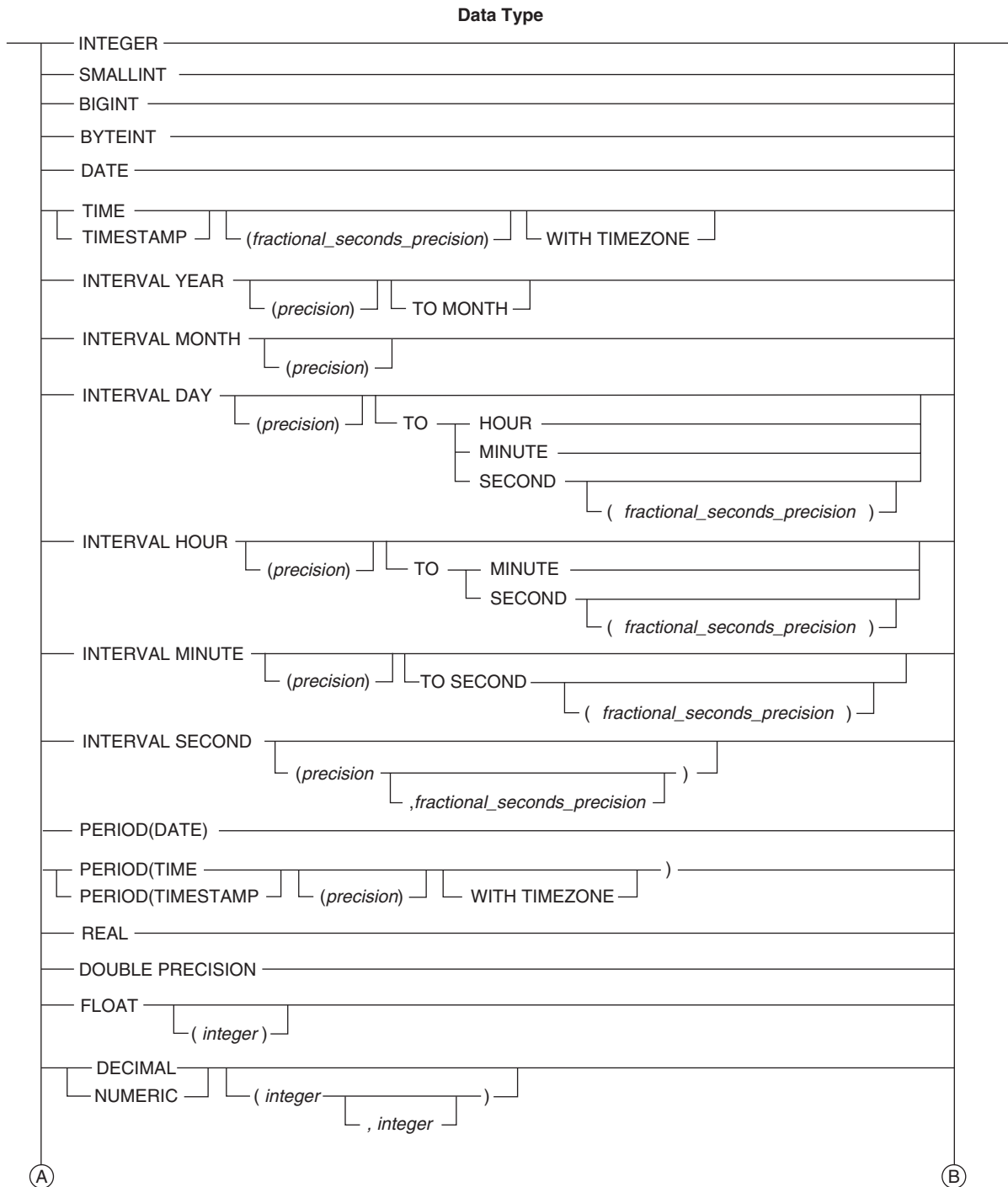


1101U020

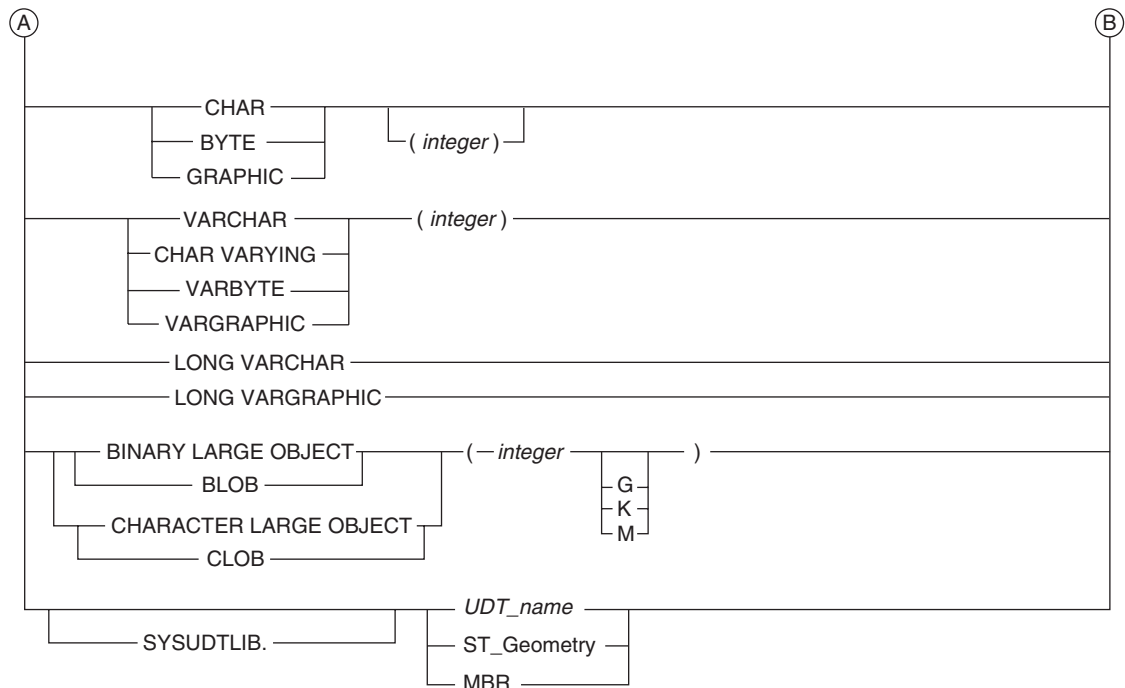


1101S024



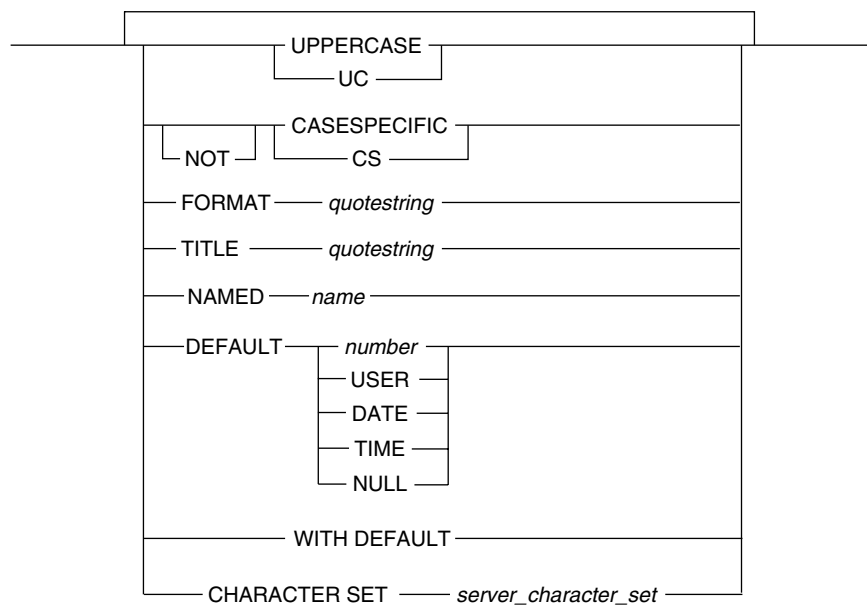


1101A535



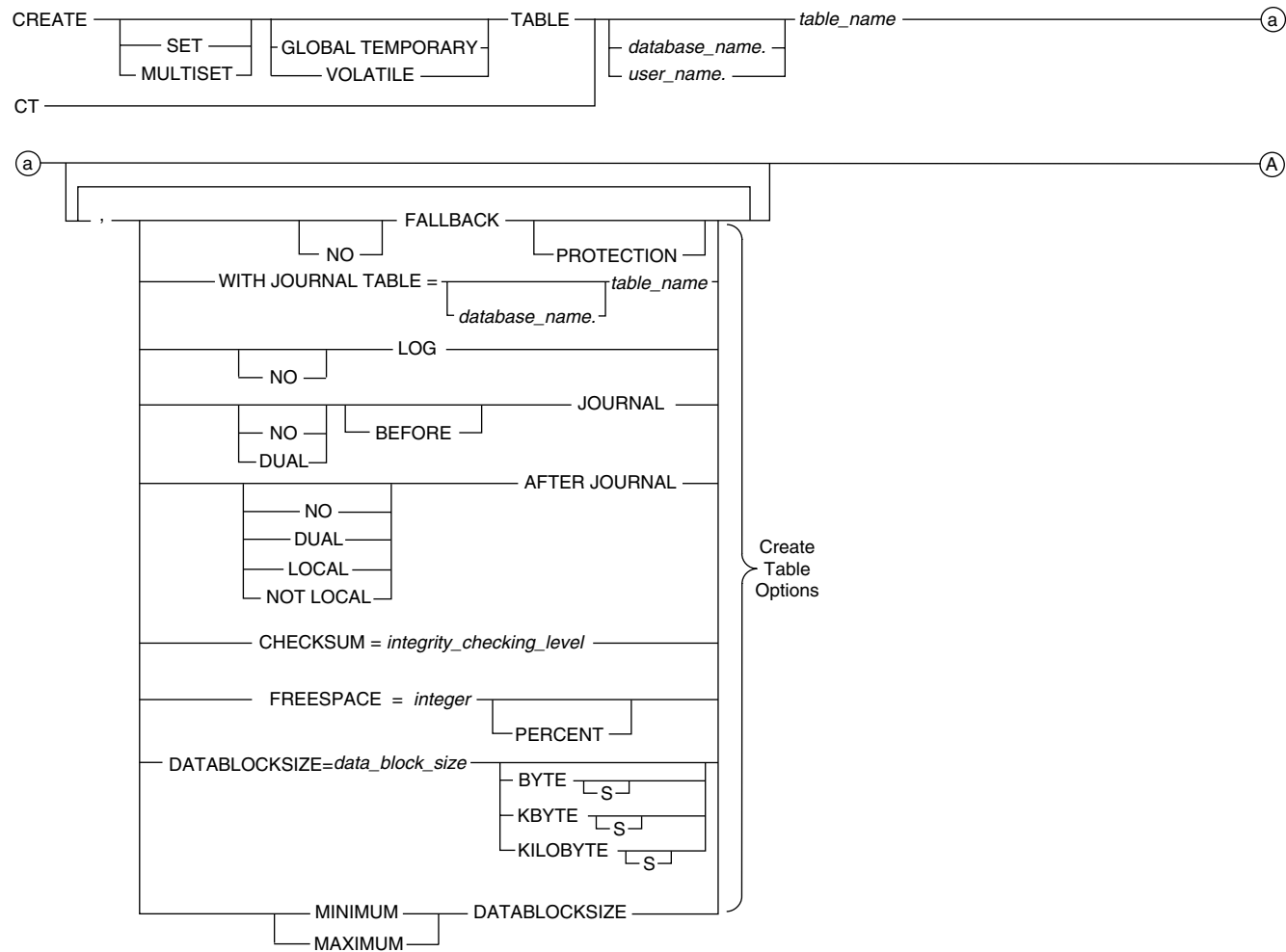
1101A536

Data Type Attributes

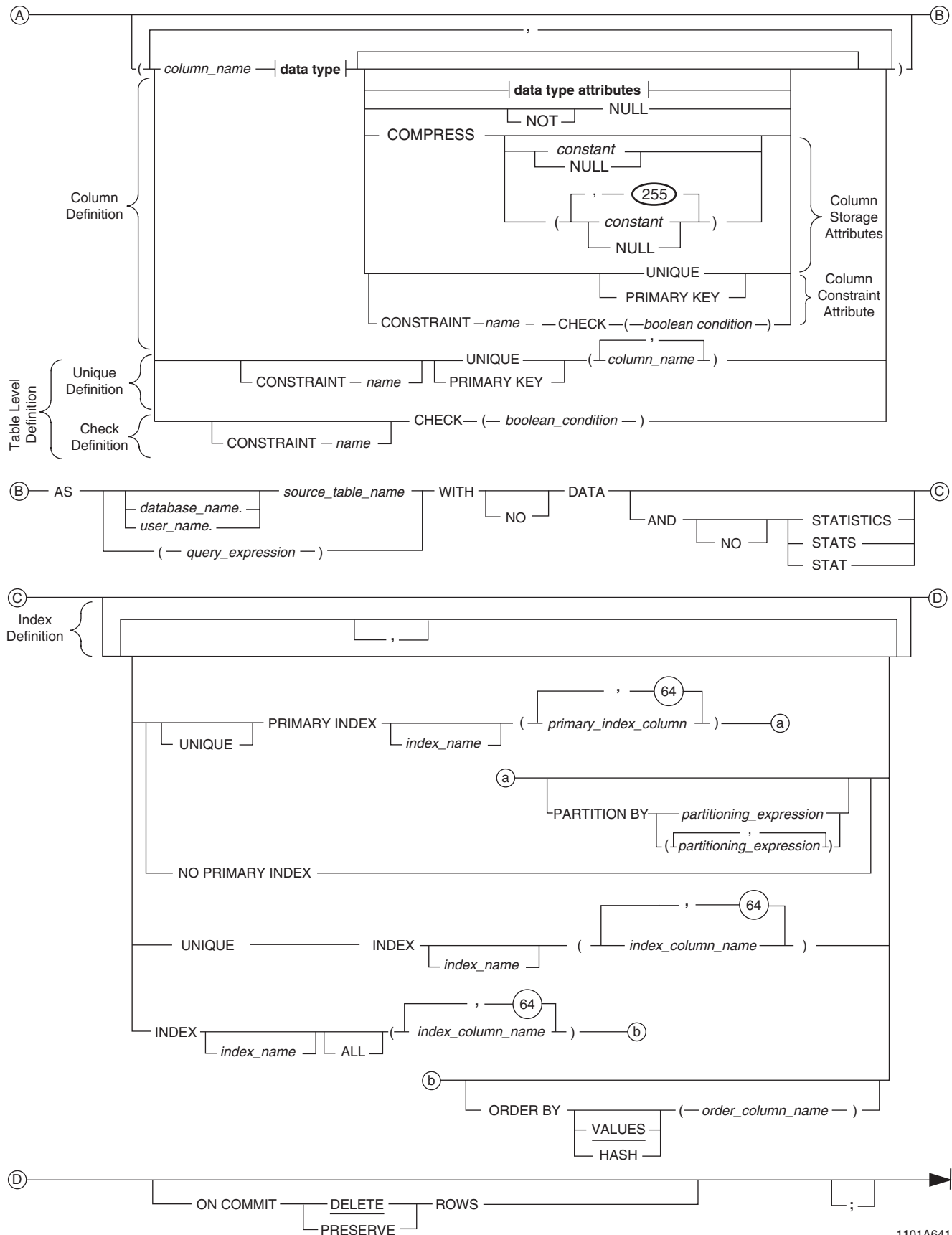


1101G205

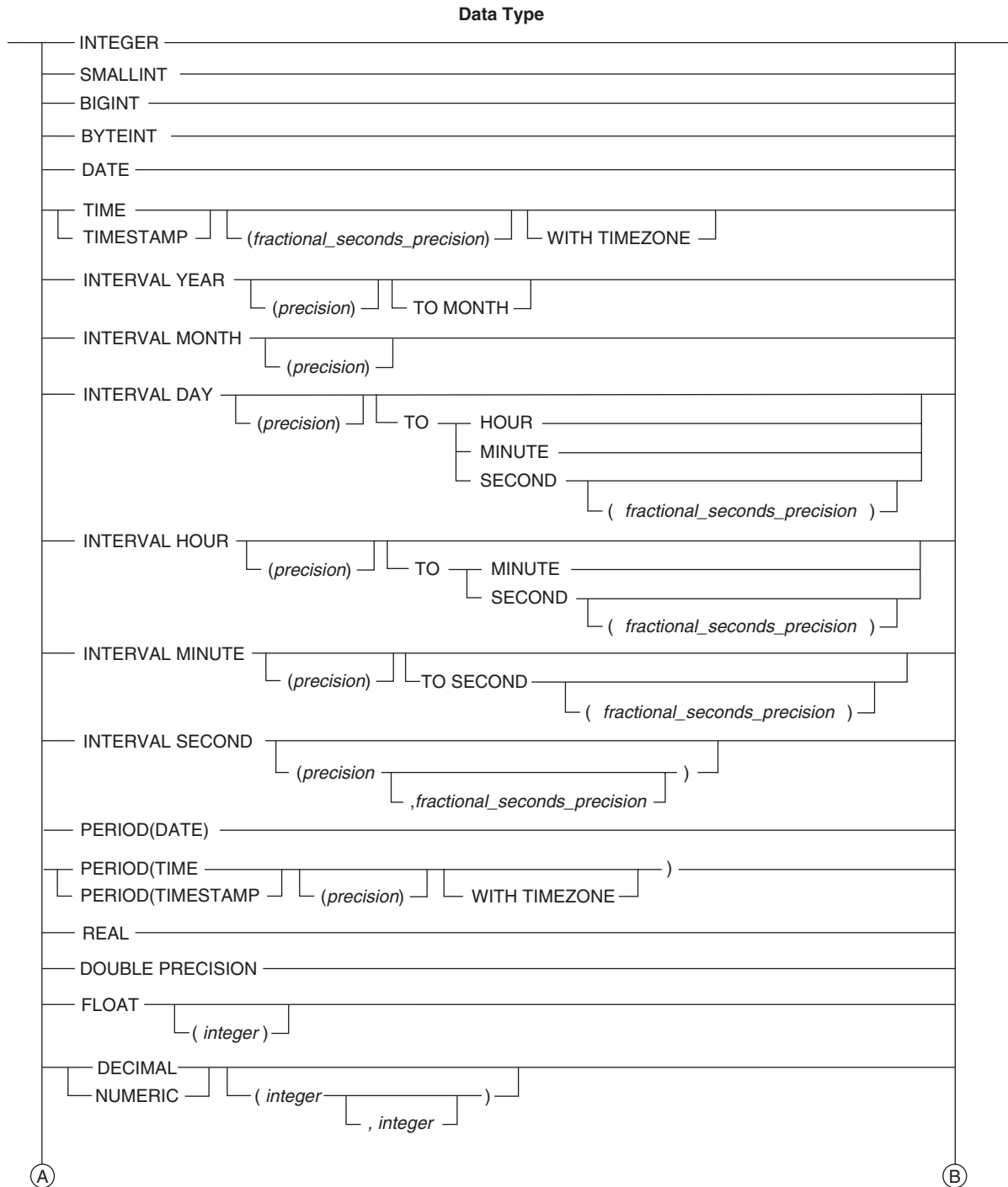
Copy Table Syntax



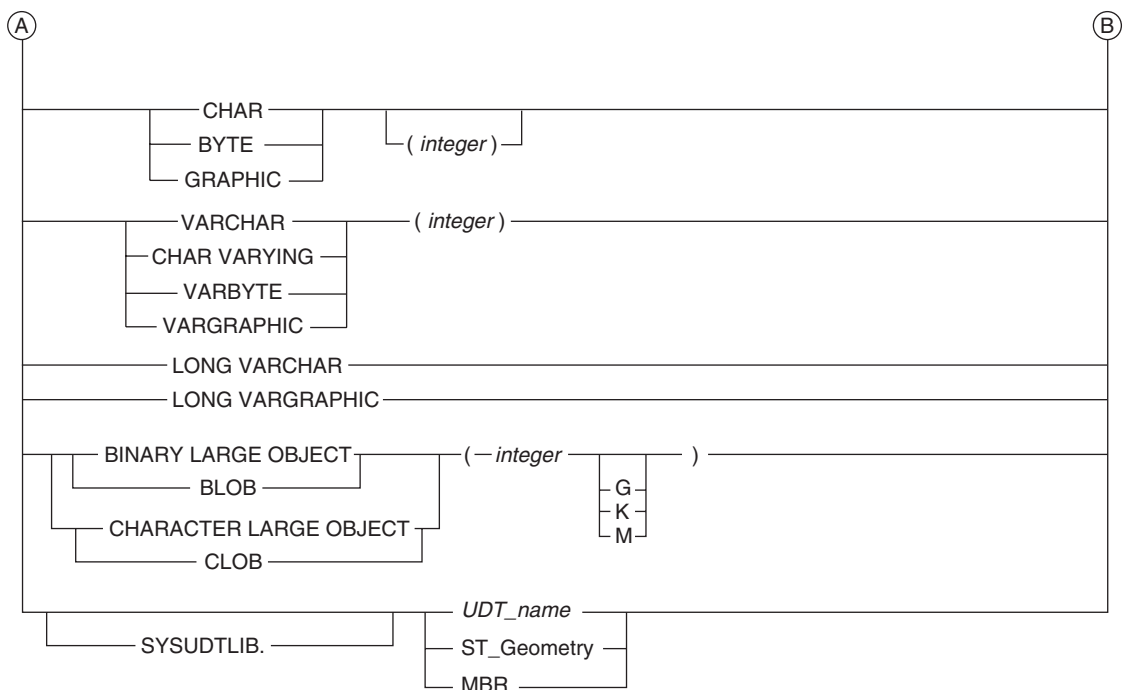
1101T020



1101A641

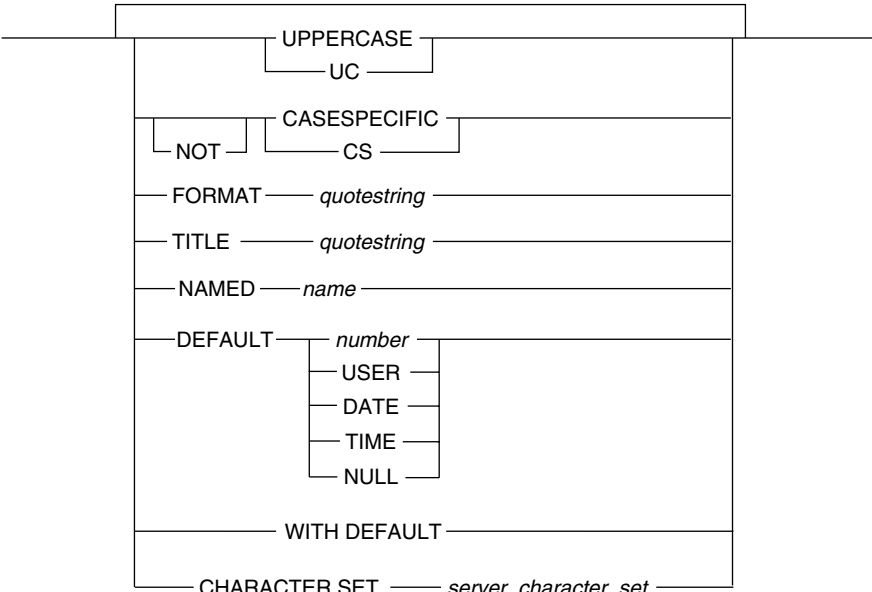


1101A535



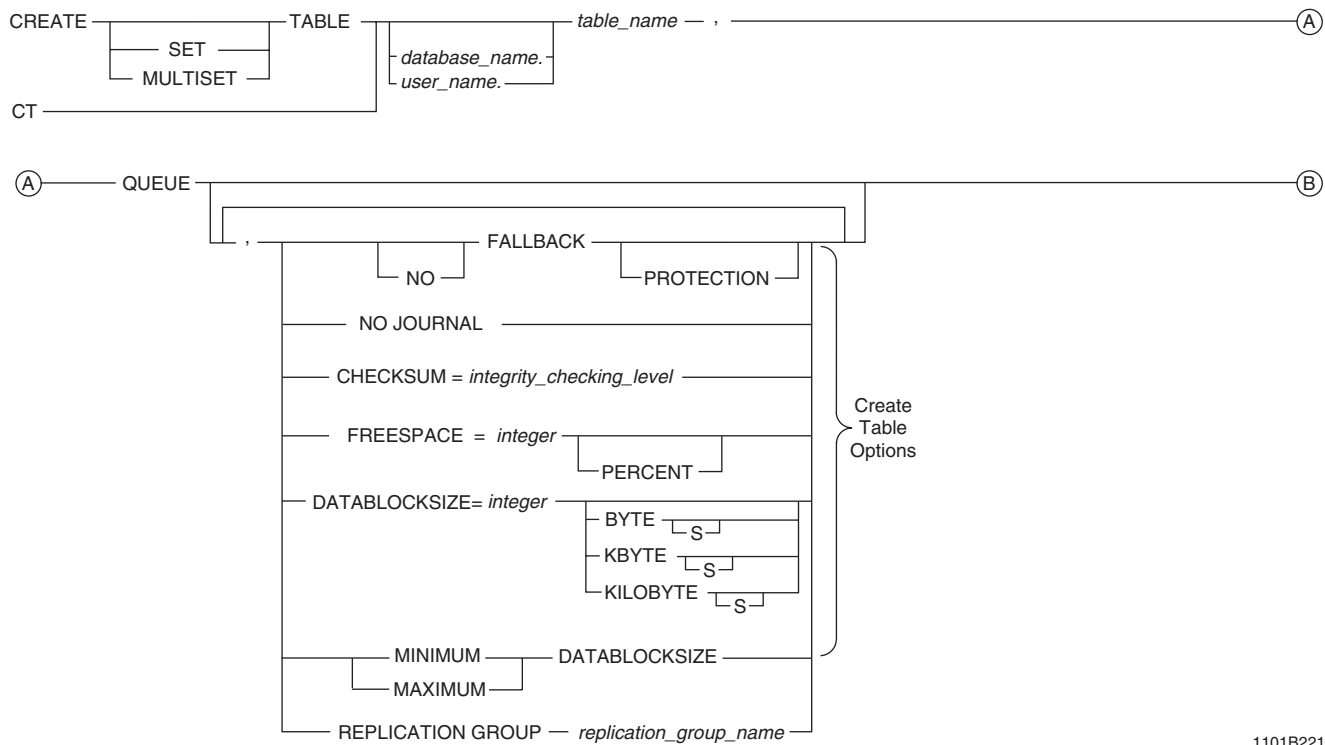
1101A536

Data Type Attributes



1101G205

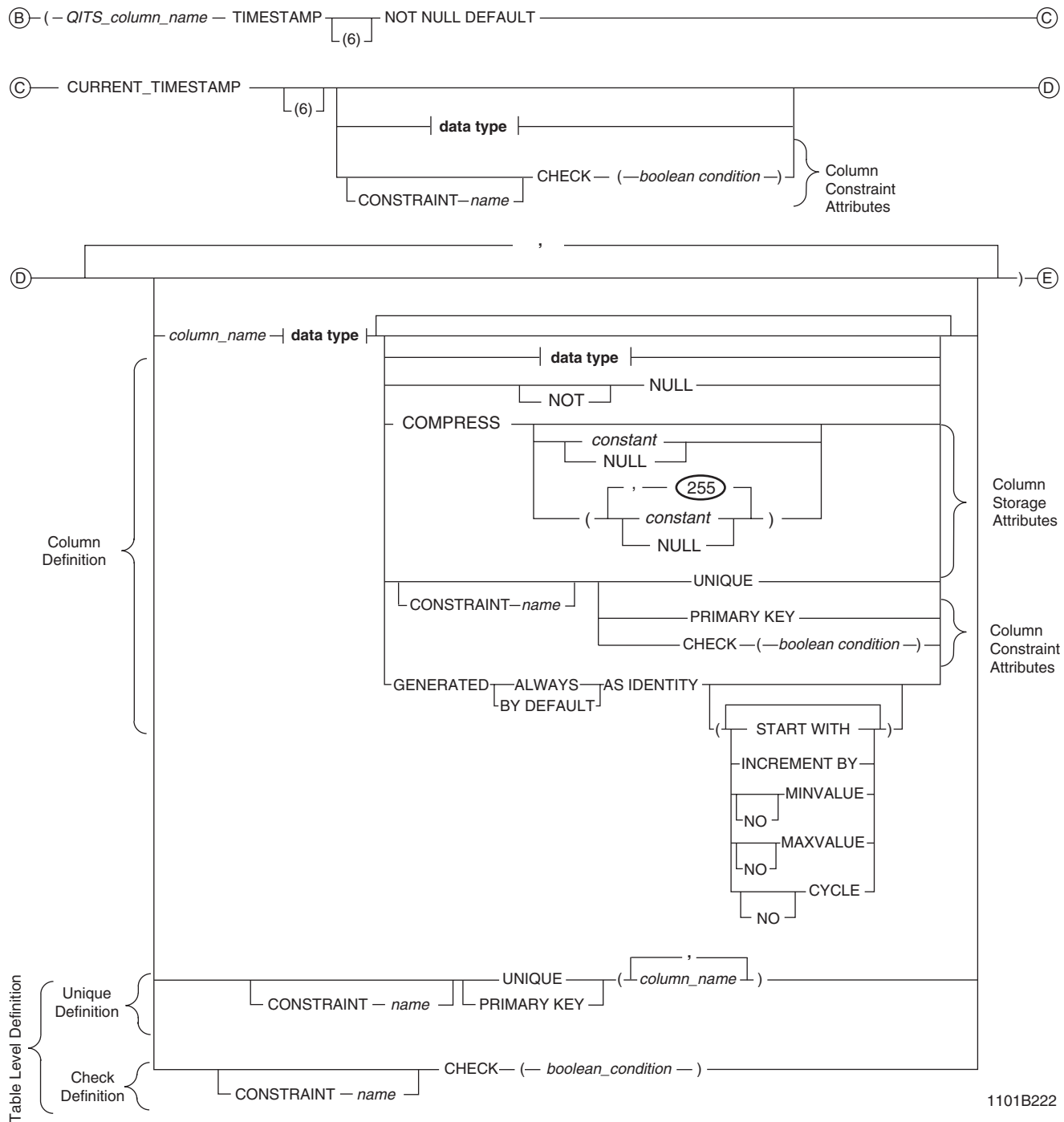
CREATE TABLE (Queue Table Form)

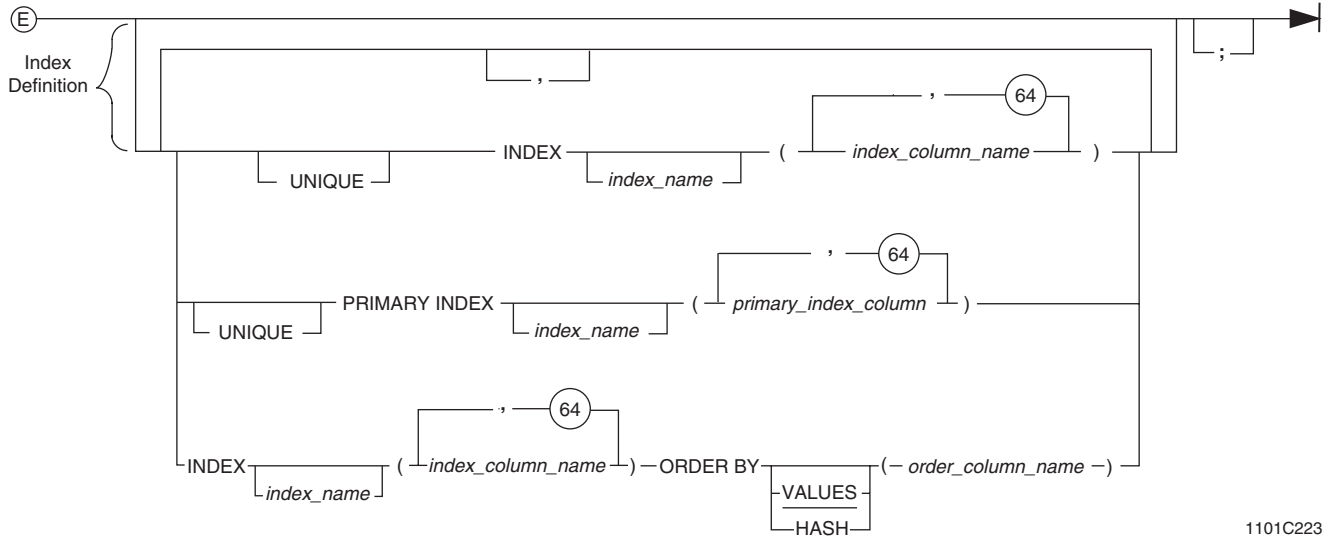


1101B221

Chapter 3: SQL Data Definition Language

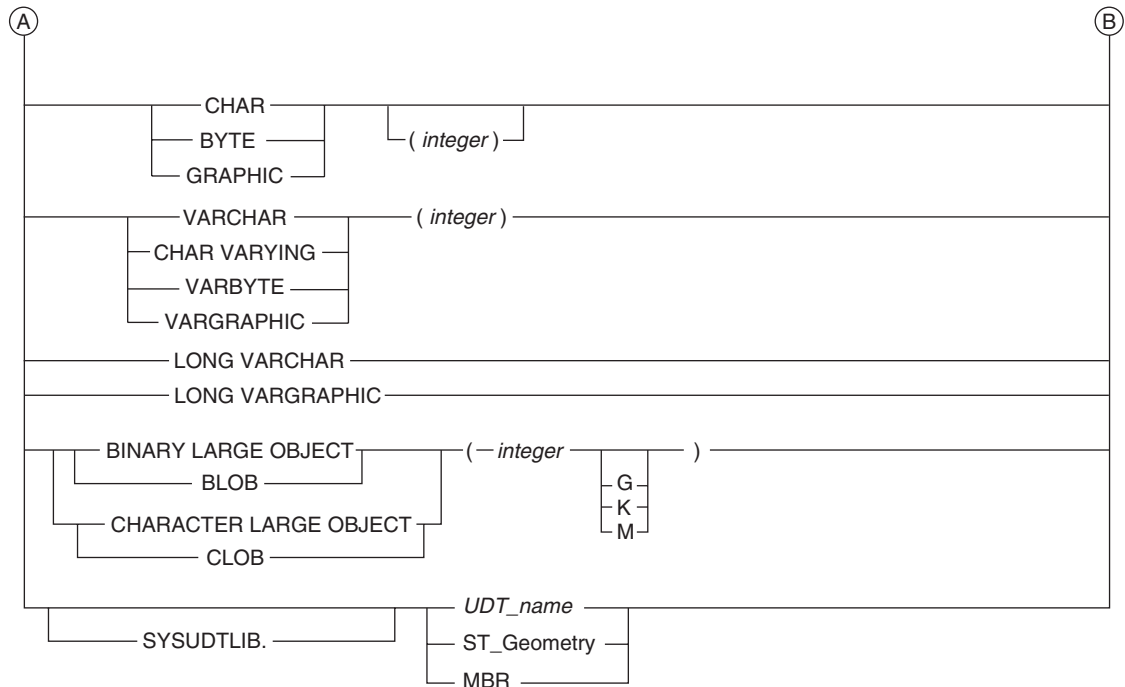
CREATE TABLE (Queue Table Form)





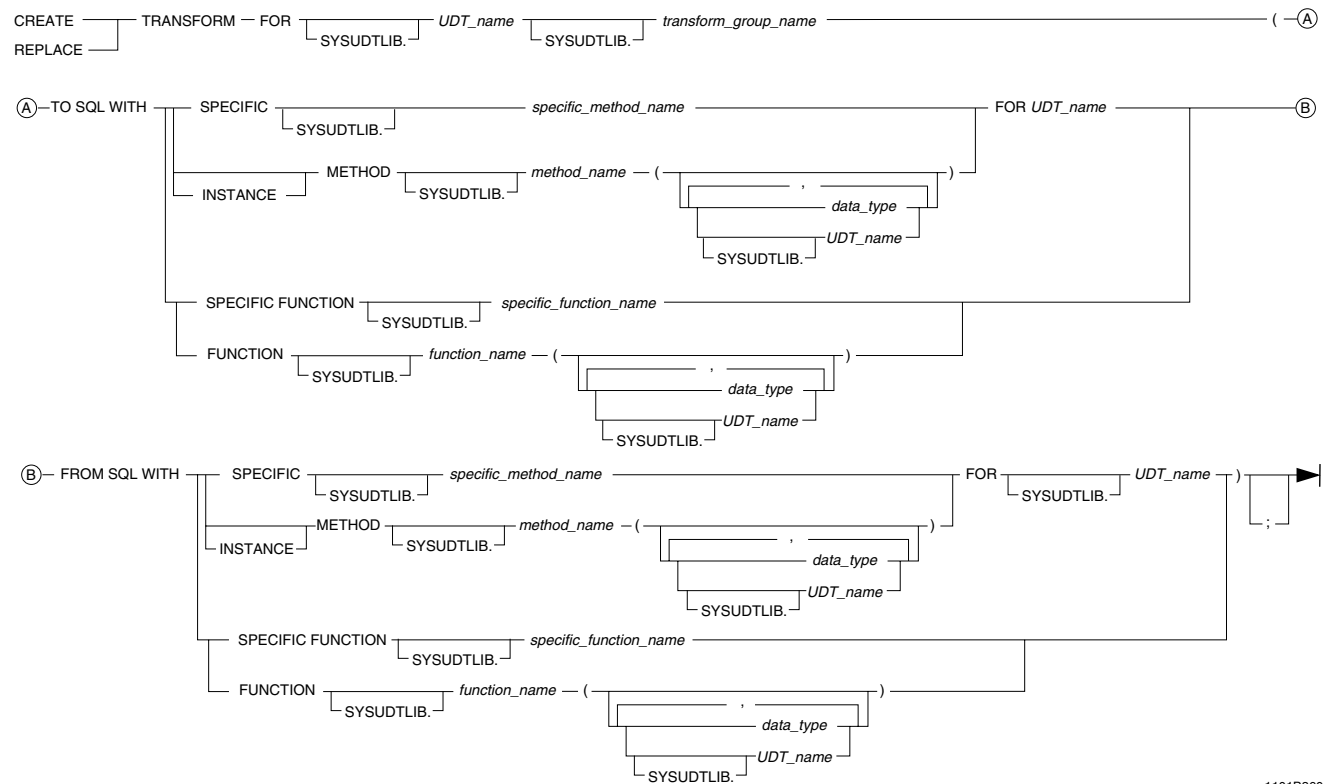
Data Type	
INTEGER	
SMALLINT	
BIGINT	
BYTEINT	
DATE	
TIME	
TIMESTAMP	(fractional_seconds_precision) WITH TIMEZONE
INTERVAL YEAR	(precision) TO MONTH
INTERVAL MONTH	(precision)
INTERVAL DAY	(precision) TO HOUR MINUTE SECOND (fractional_seconds_precision)
INTERVAL HOUR	(precision) TO MINUTE SECOND (fractional_seconds_precision)
INTERVAL MINUTE	(precision) TO SECOND (fractional_seconds_precision)
INTERVAL SECOND	(precision, fractional_seconds_precision)
PERIOD(DATE)	
PERIOD(TIME)	
PERIOD(TIMESTAMP)	(precision) WITH TIMEZONE
REAL	
DOUBLE PRECISION	
FLOAT	(integer)
DECIMAL	
NUMERIC	(integer , integer)

1101A535



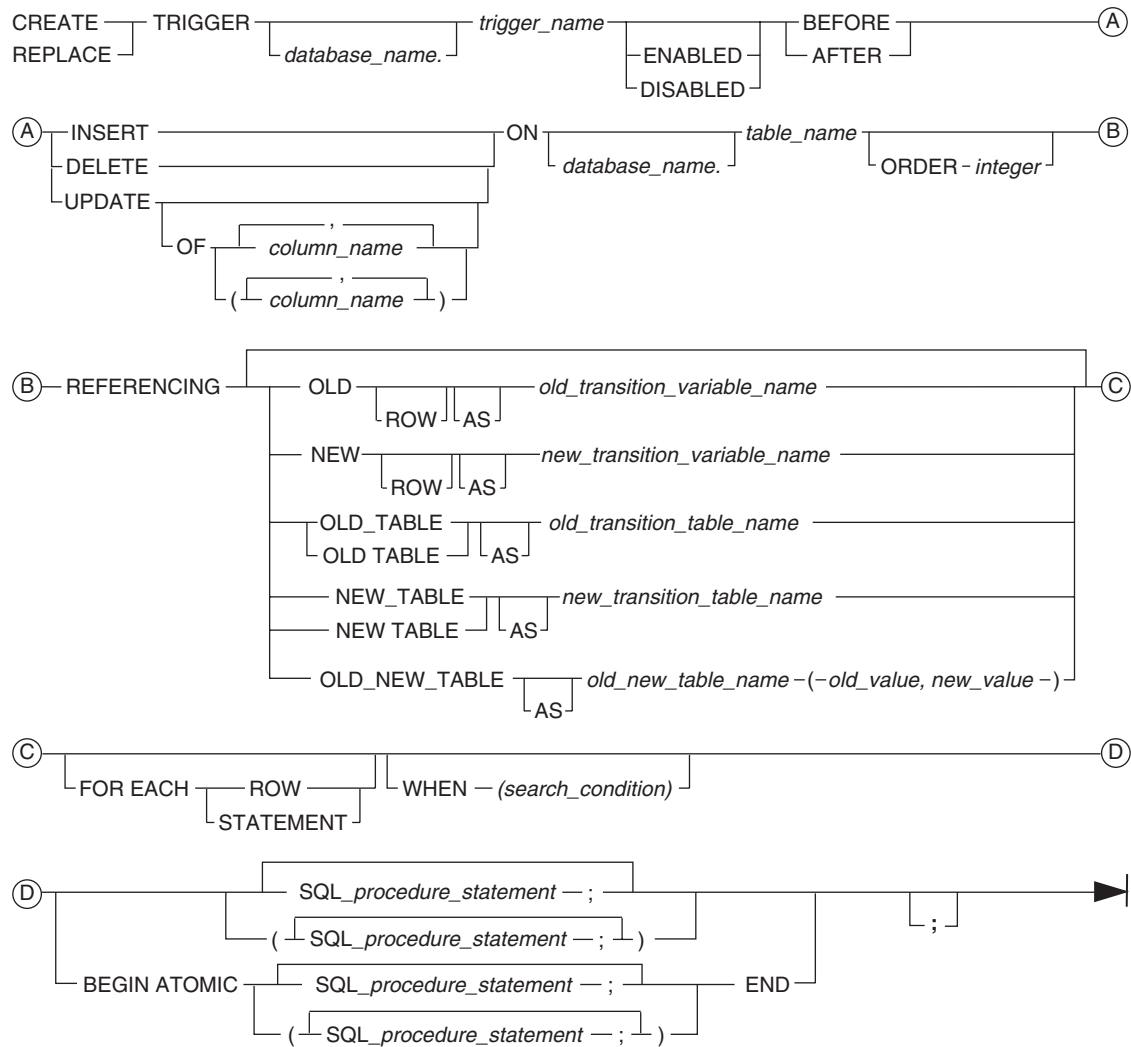
1101A536

CREATE TRANSFORM/ REPLACE TRANSFORM



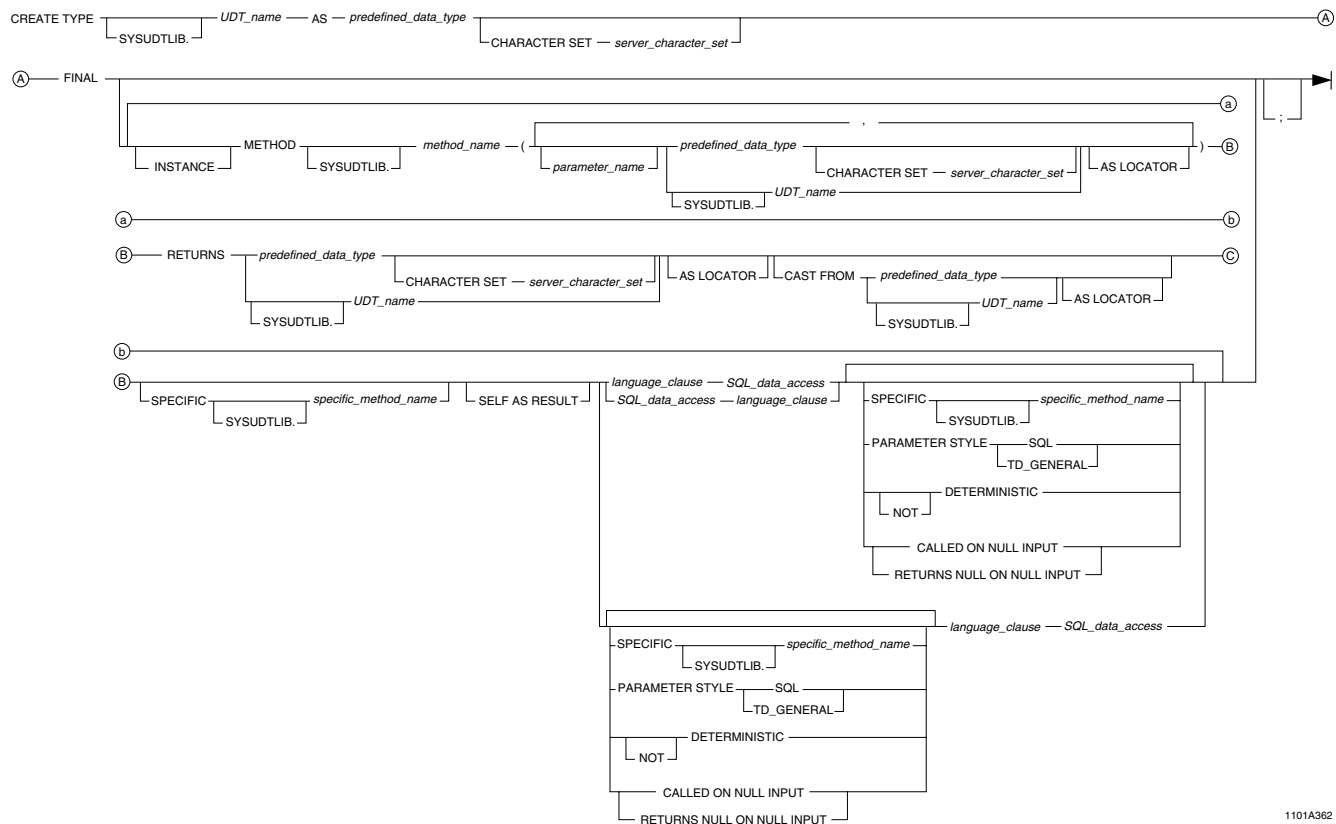
1101B360

CREATE TRIGGER/ REPLACE TRIGGER



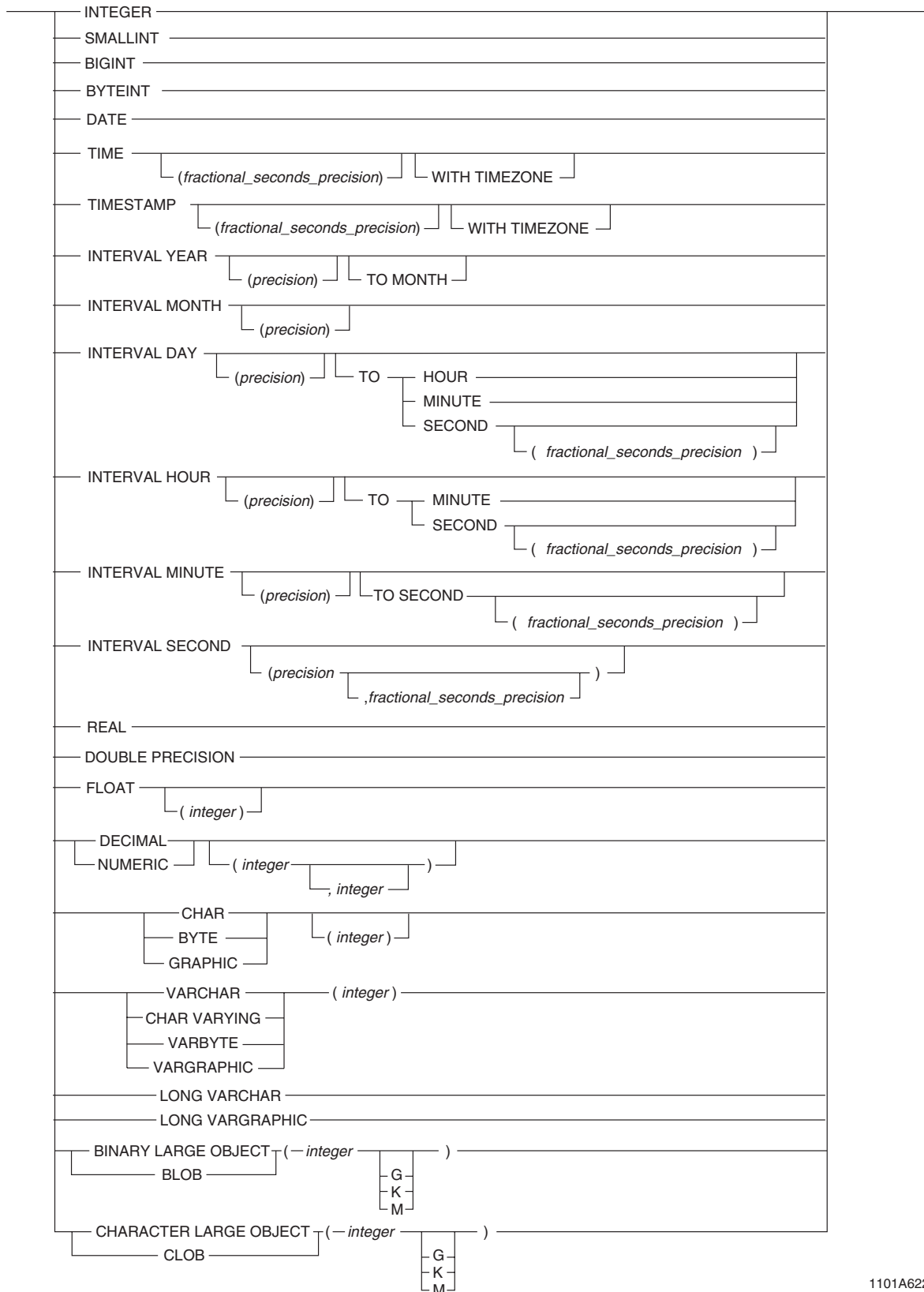
1101D202

CREATE TYPE (Distinct Form)



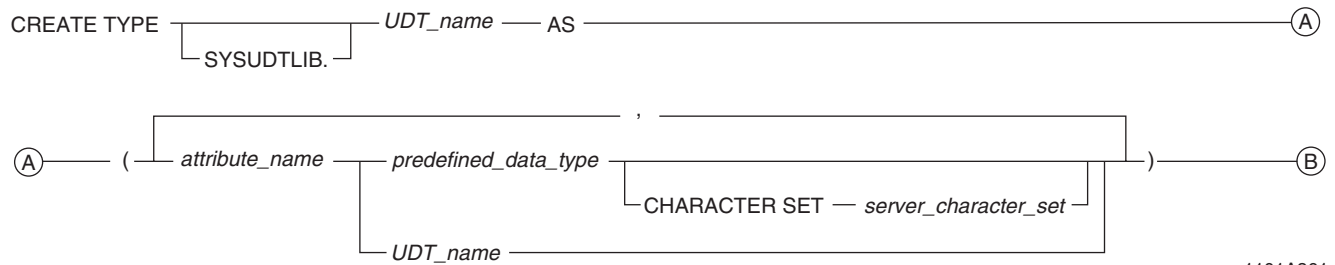
1101A362

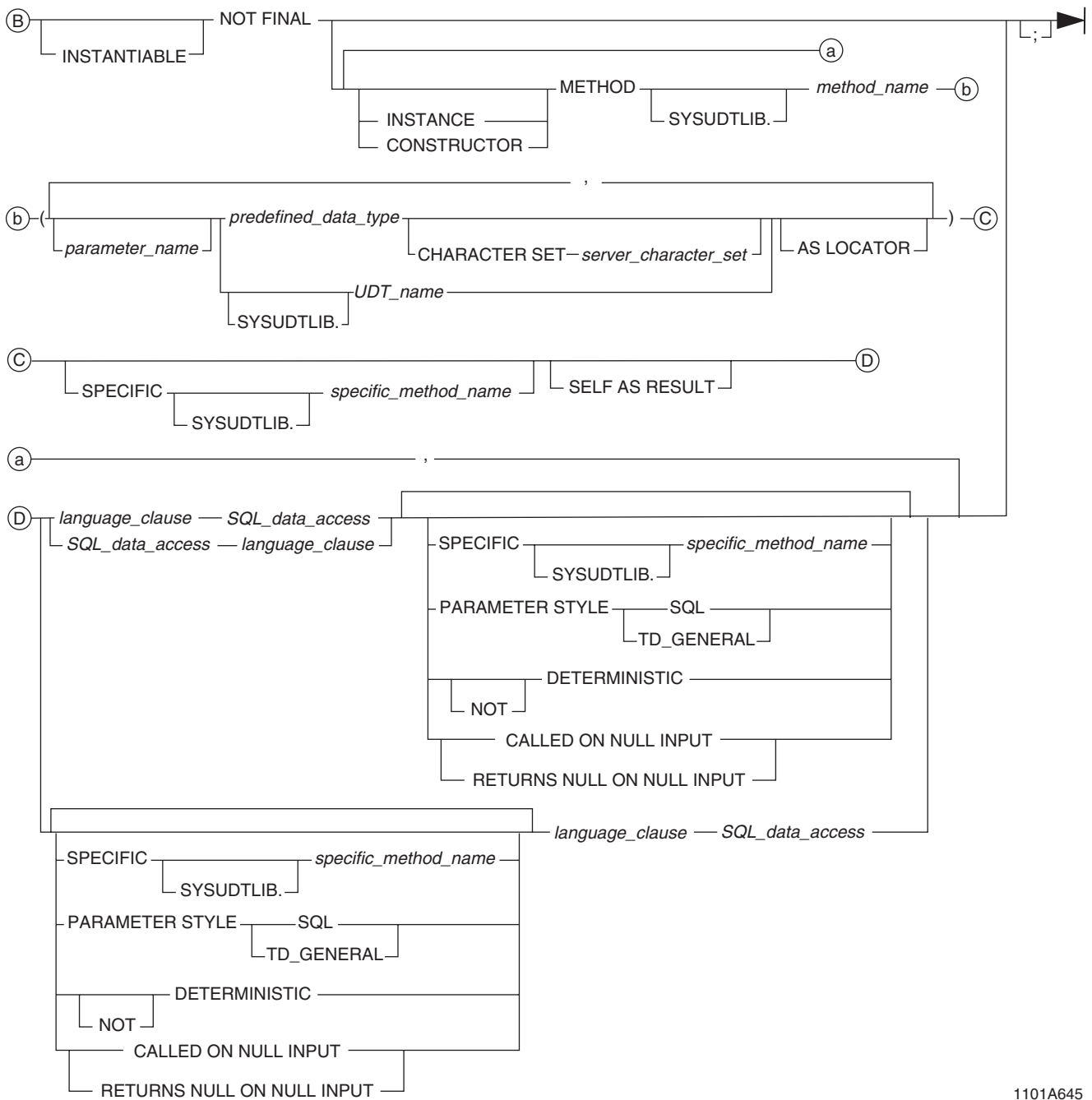
Data Type Declaration



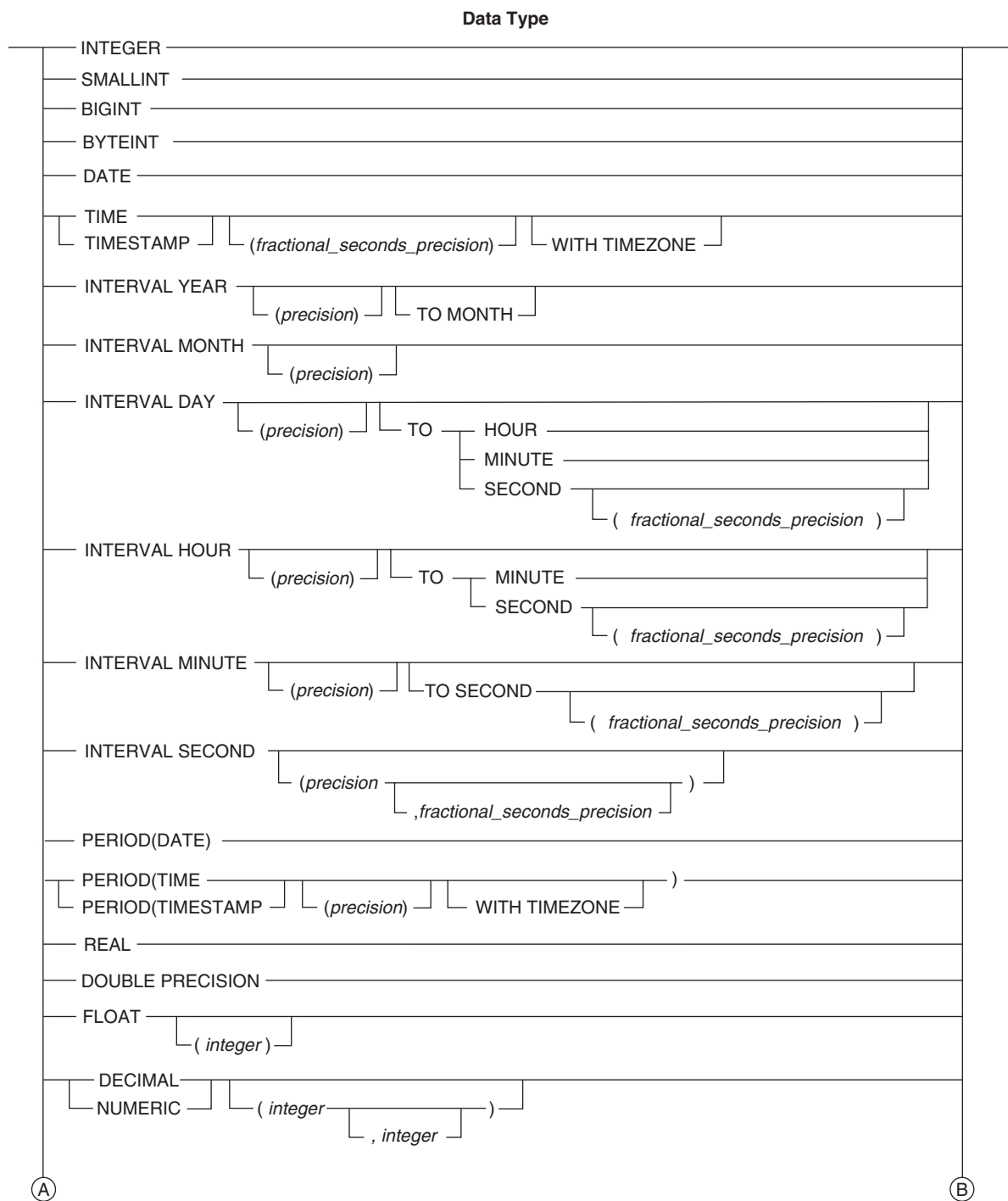
1101A622

CREATE TYPE (Structured Form)

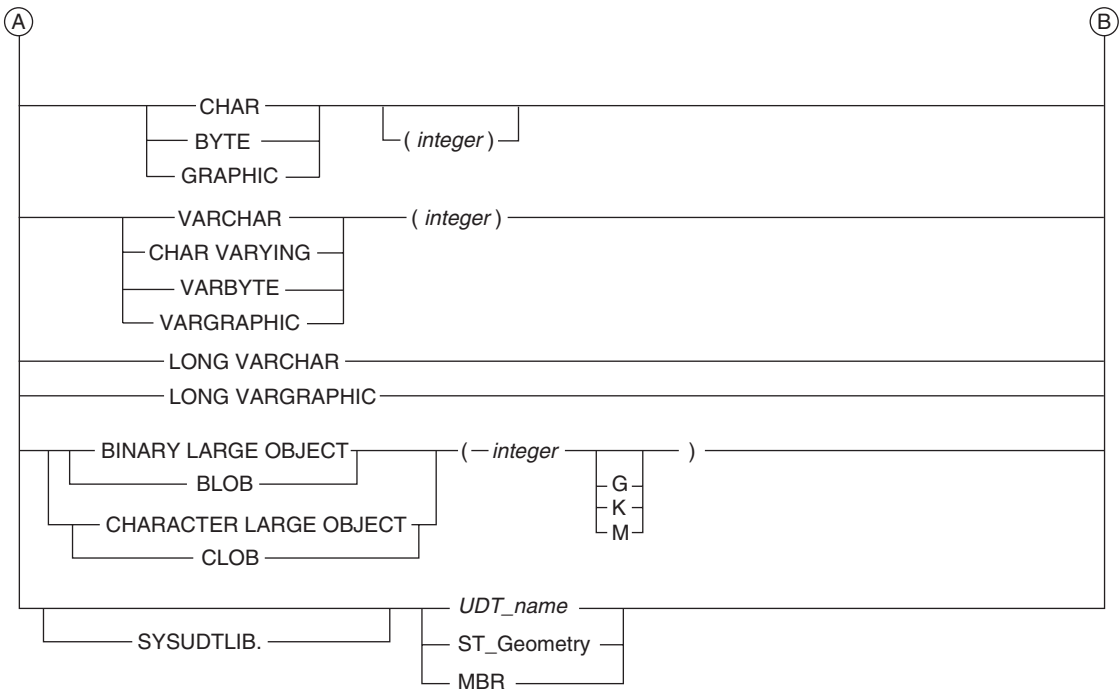




1101A645

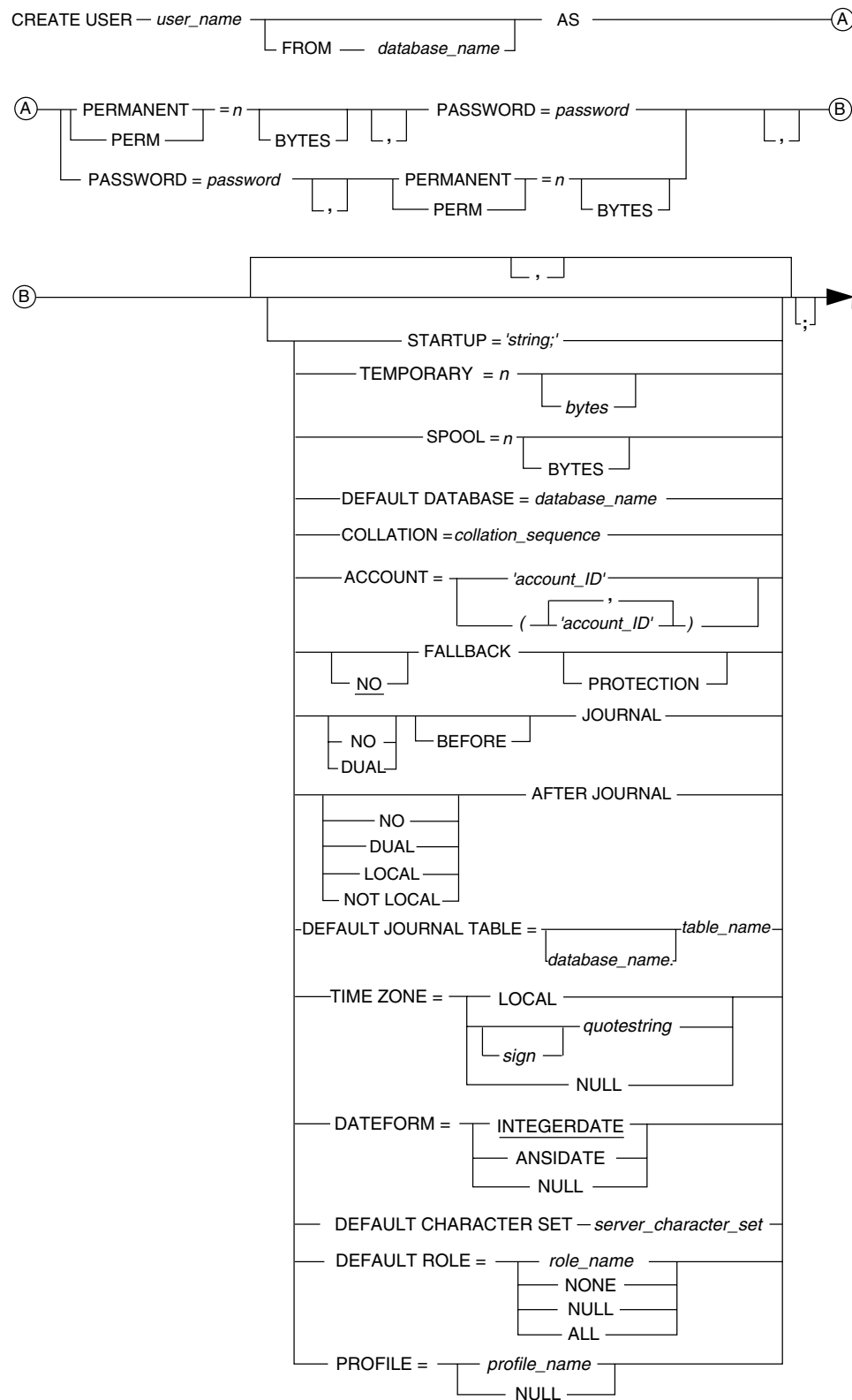


1101A535



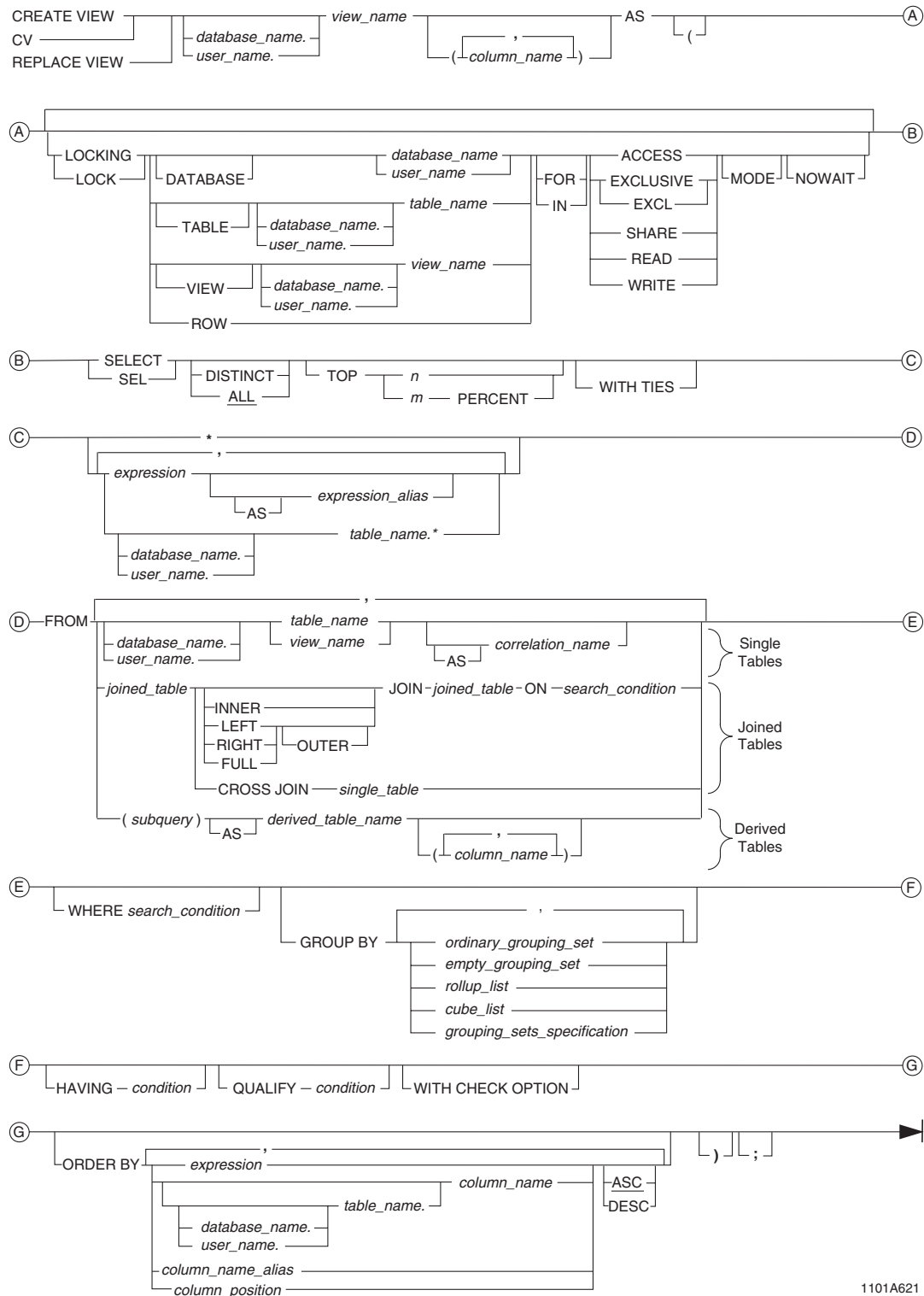
1101A536

CREATE USER



1101C007

CREATE VIEW/ REPLACE VIEW



DATABASE

DATABASE *database_name* ;

FF07A016

DELETE DATABASE DELETE USER

DELETE DATABASE *name* ;
DEL USER *name* ALL ;

FF07D026

DROP AUTHORIZATION

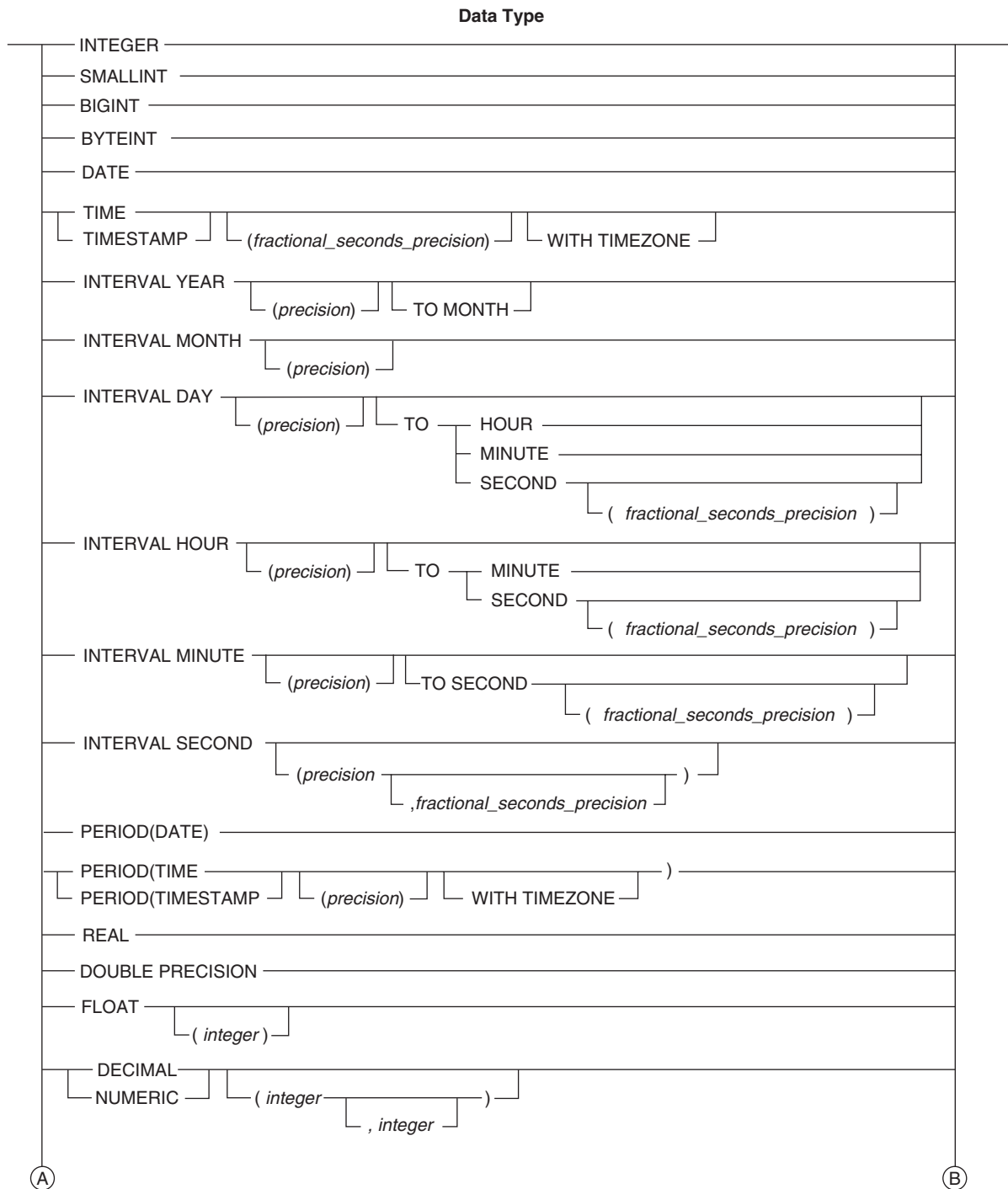
DROP AUTHORIZATION *database_name.* *authorization_name* ;

1101A324

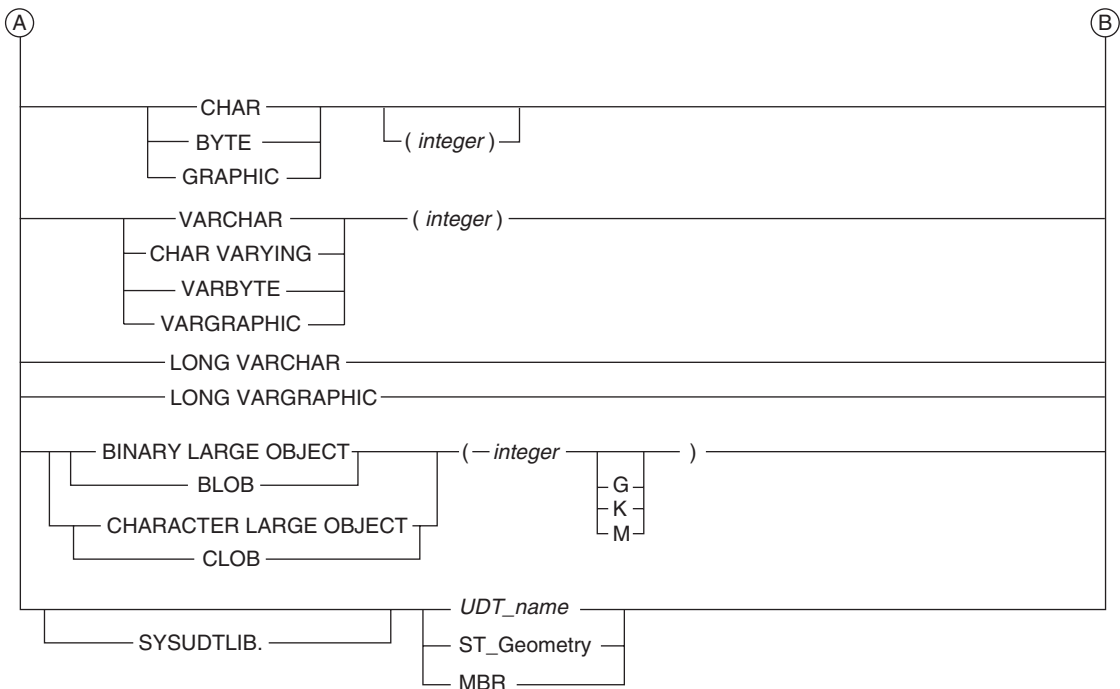
DROP CAST

DROP CAST *database_name.* (*source_data_type* AS *target_data_type*) ;

1101A325



1101A535



1101A536

DROP DATABASE

DROP DATABASE *database_name* [;] ►

1101A070

DROP ERROR TABLE

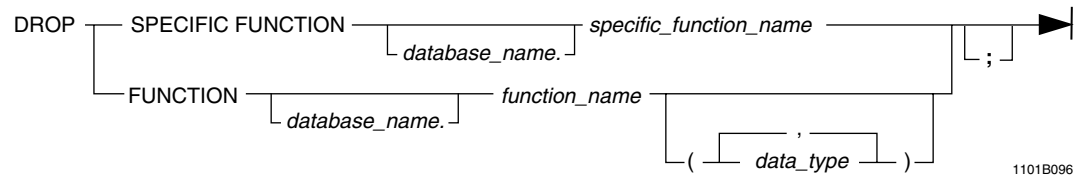
DROP ERROR TABLE FOR *data_table_name* [*database_name.*] [;] ►

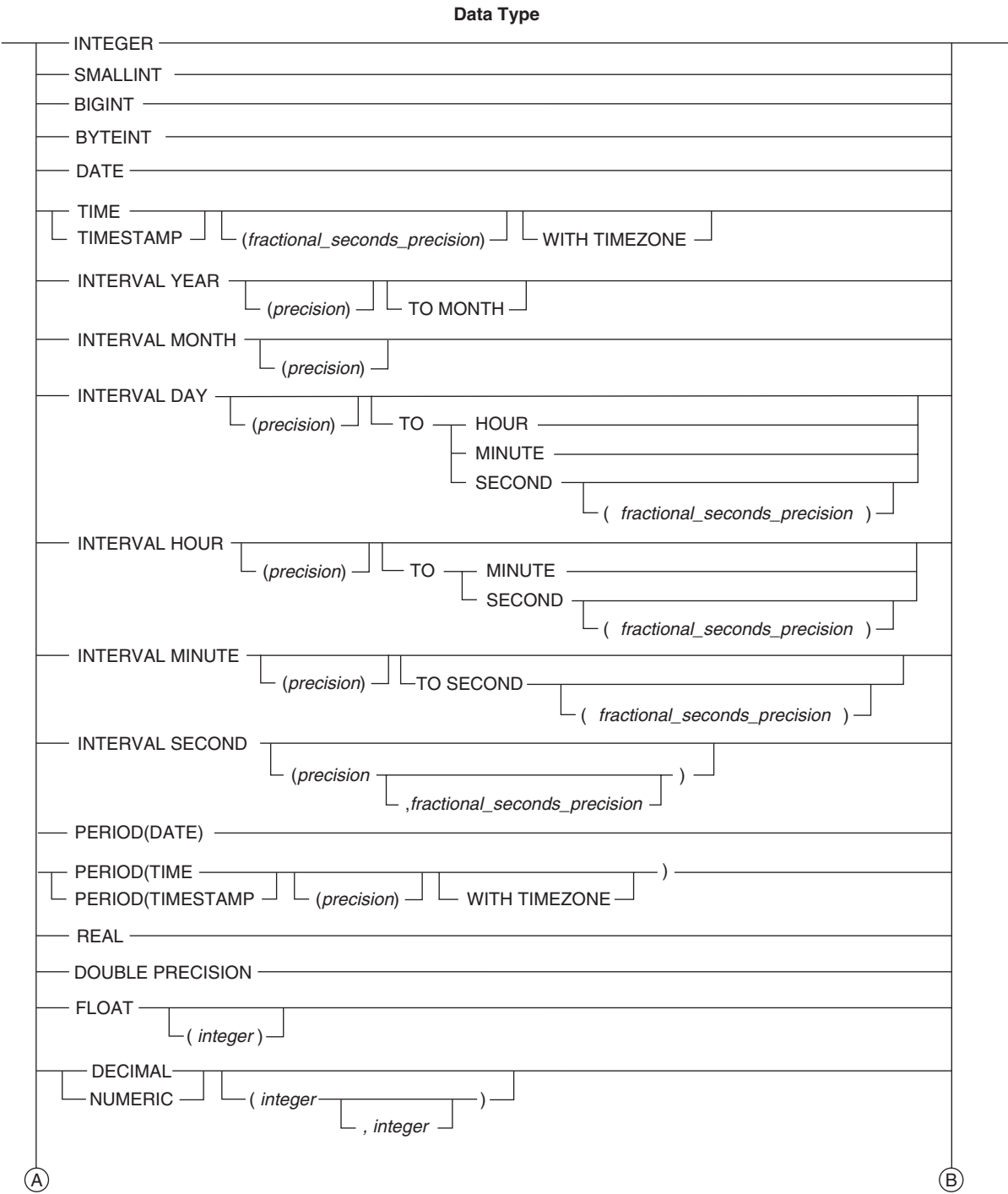
1101A437

DROP TABLE *error_table_name* [*database_name.*] [;] ►

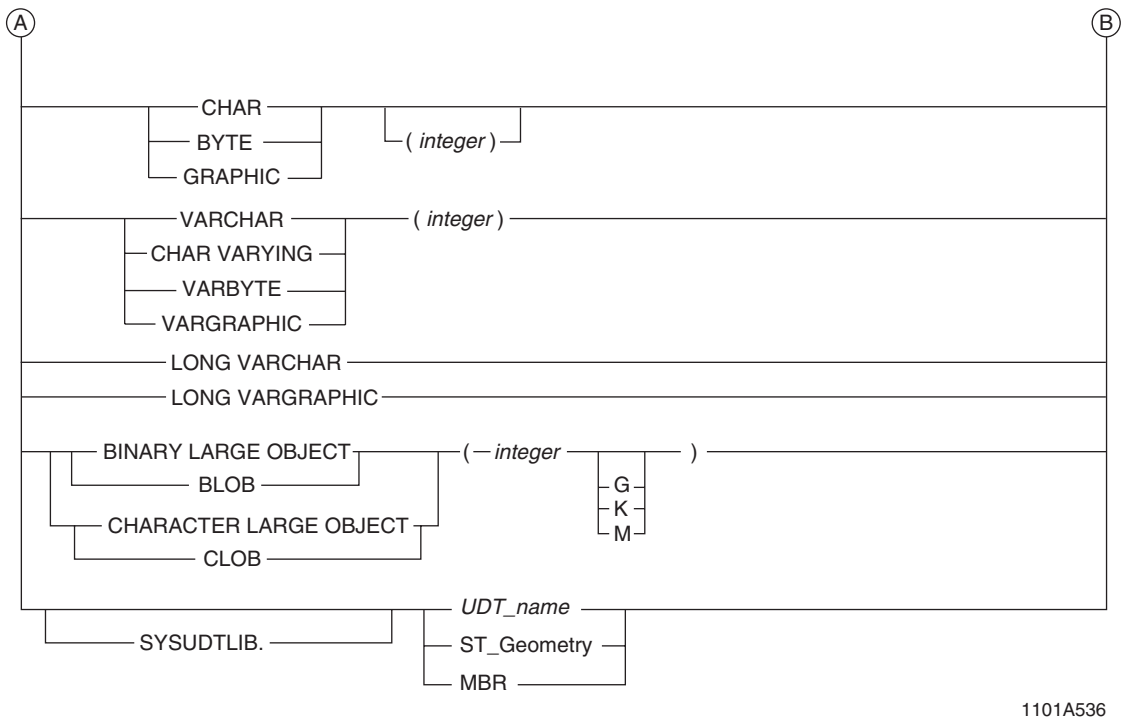
1101A438

DROP FUNCTION

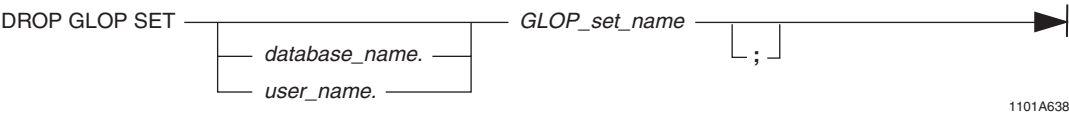




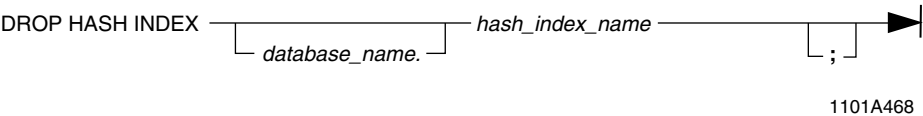
1101A535



DROP GLOP SET

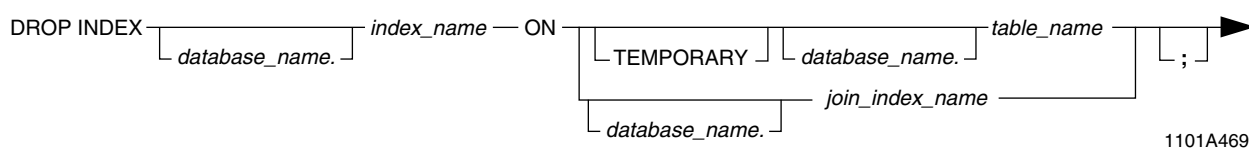


DROP HASH INDEX

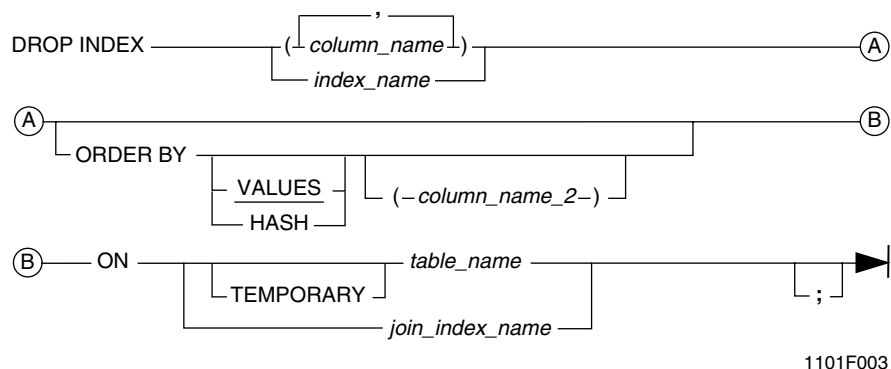


DROP INDEX

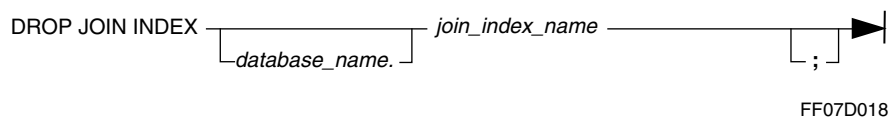
DROP *index_name* Syntax



DROP *index_definition* Syntax

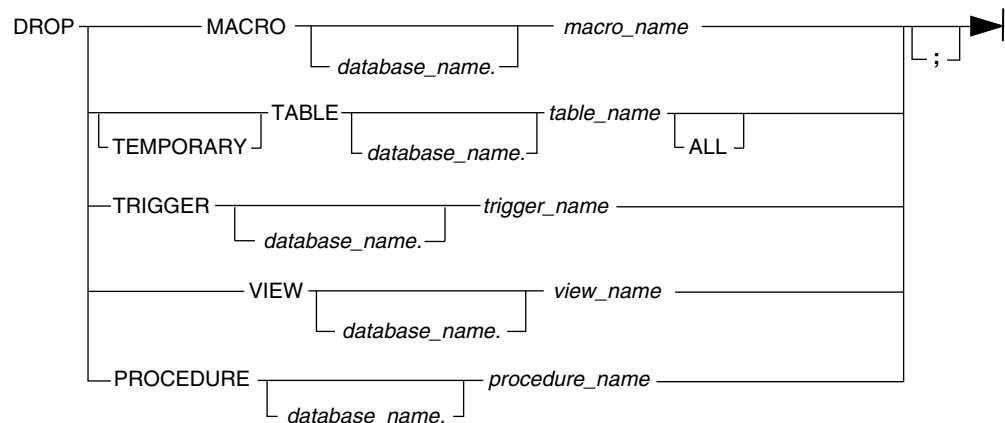


DROP JOIN INDEX



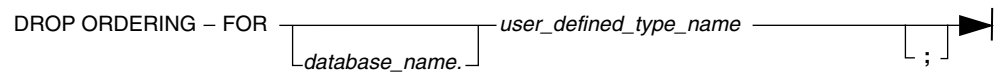
DROP MACRO/ DROP PROCEDURE/ DROP TABLE/

DROP TRIGGER/ DROP VIEW



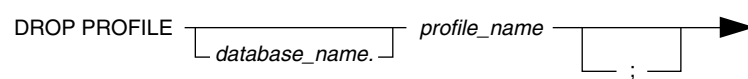
1101E093

DROP ORDERING



1101A326

DROP PROFILE



1101A470

DROP REPLICATION GROUP

DROP REPLICATION GROUP — *replication_group_name* —————▶
[;]

1101A240

DROP REPLICATION RULESET

DROP REPLICATION RULESET — *rule_set_name* — FOR — *replication_group_name* —▶
[;]

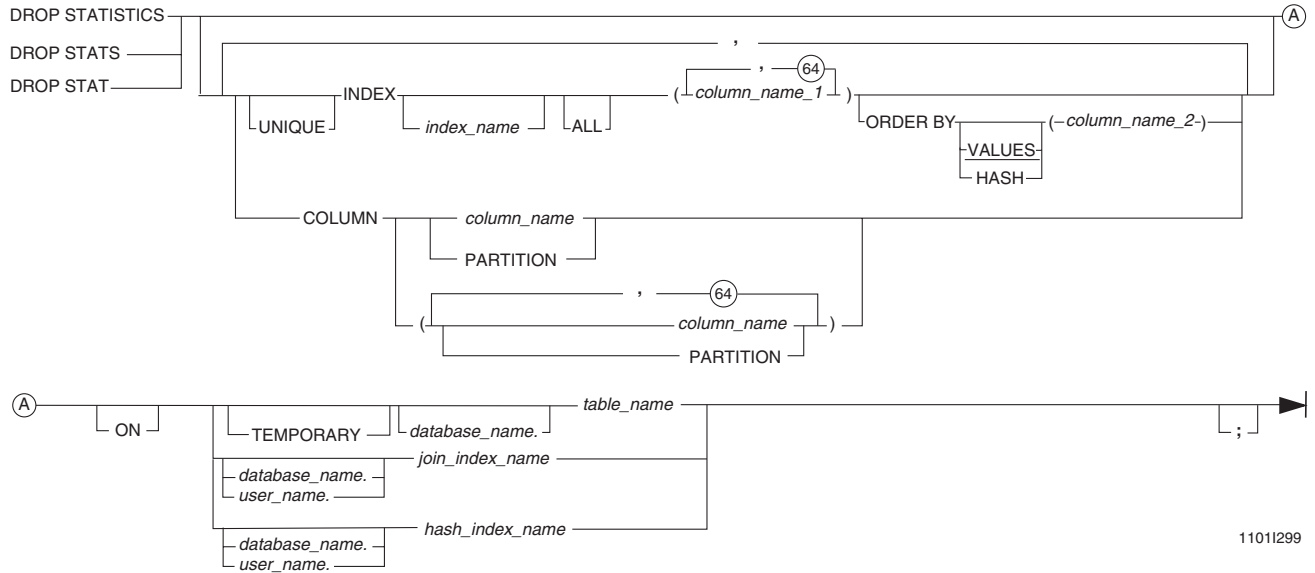
1101A554

DROP ROLE

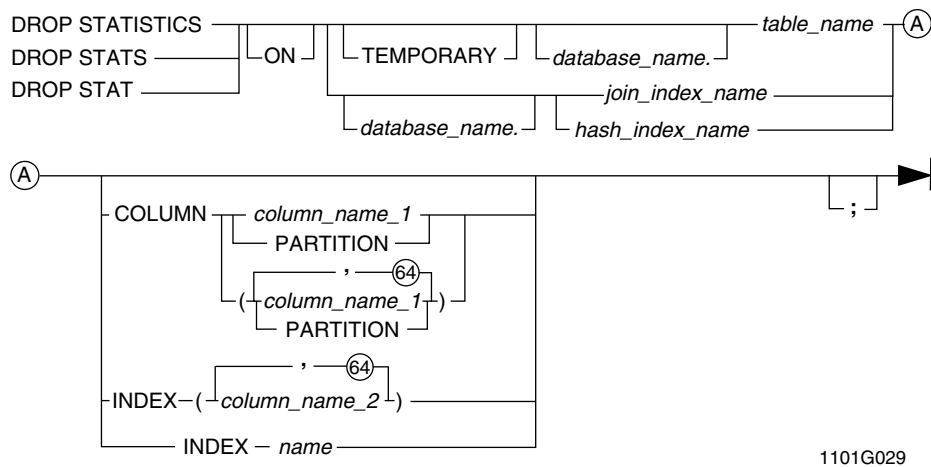
DROP ——— ROLE ——— *role_name* —▶
[EXTERNAL] [*database_name.*] [;]

1101B289

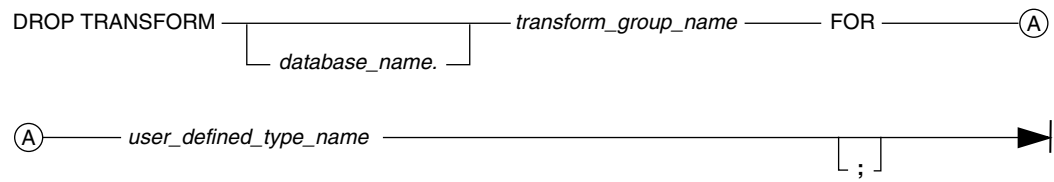
DROP STATISTICS (Optimizer Form)



Syntax (Alternate)

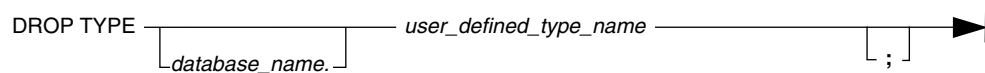


DROP TRANSFORM



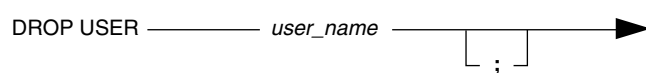
1101A327

DROP TYPE



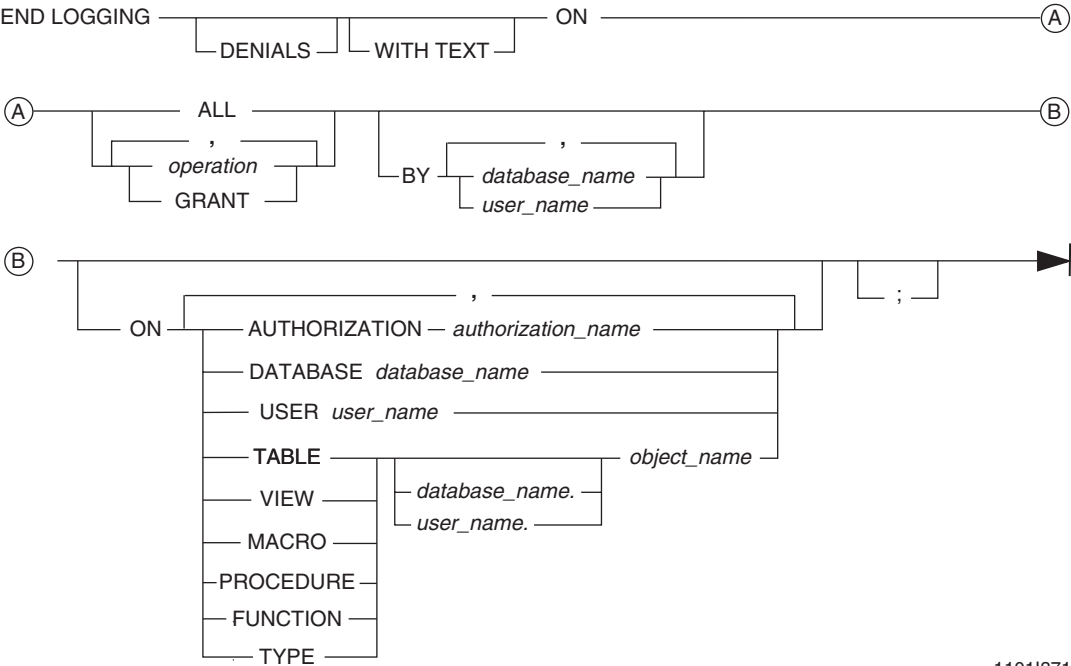
1101A328

DROP USER

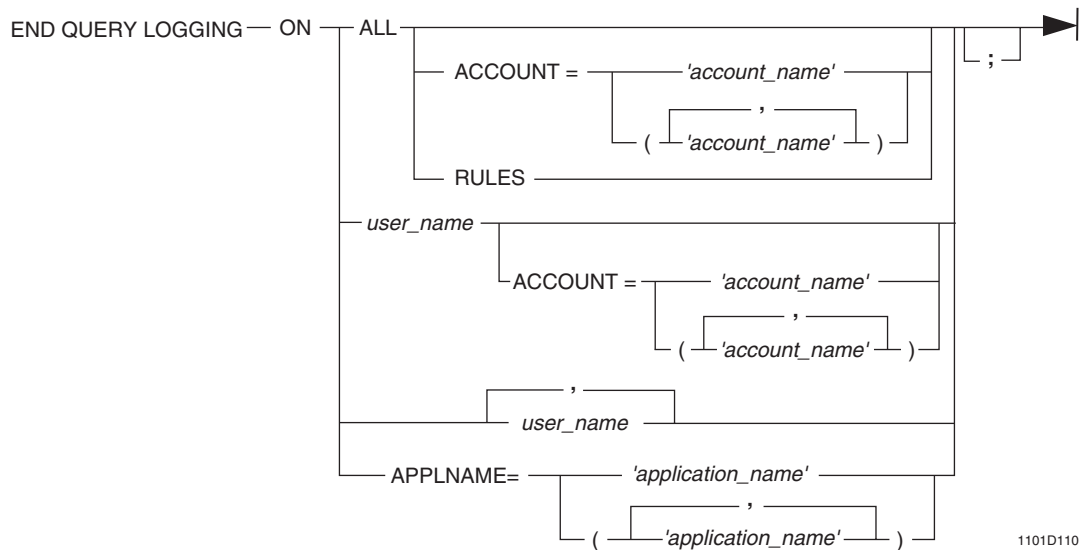


1101A071

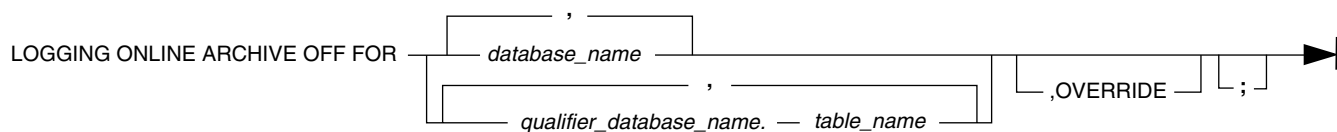
END LOGGING



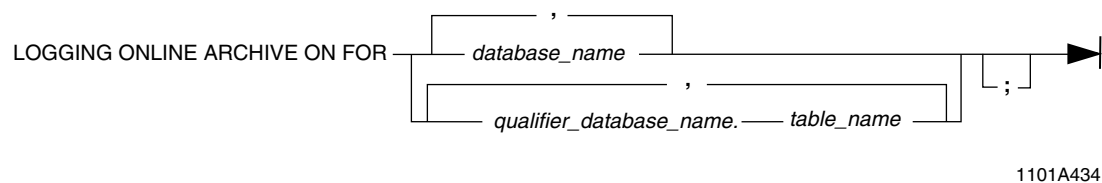
END QUERY LOGGING



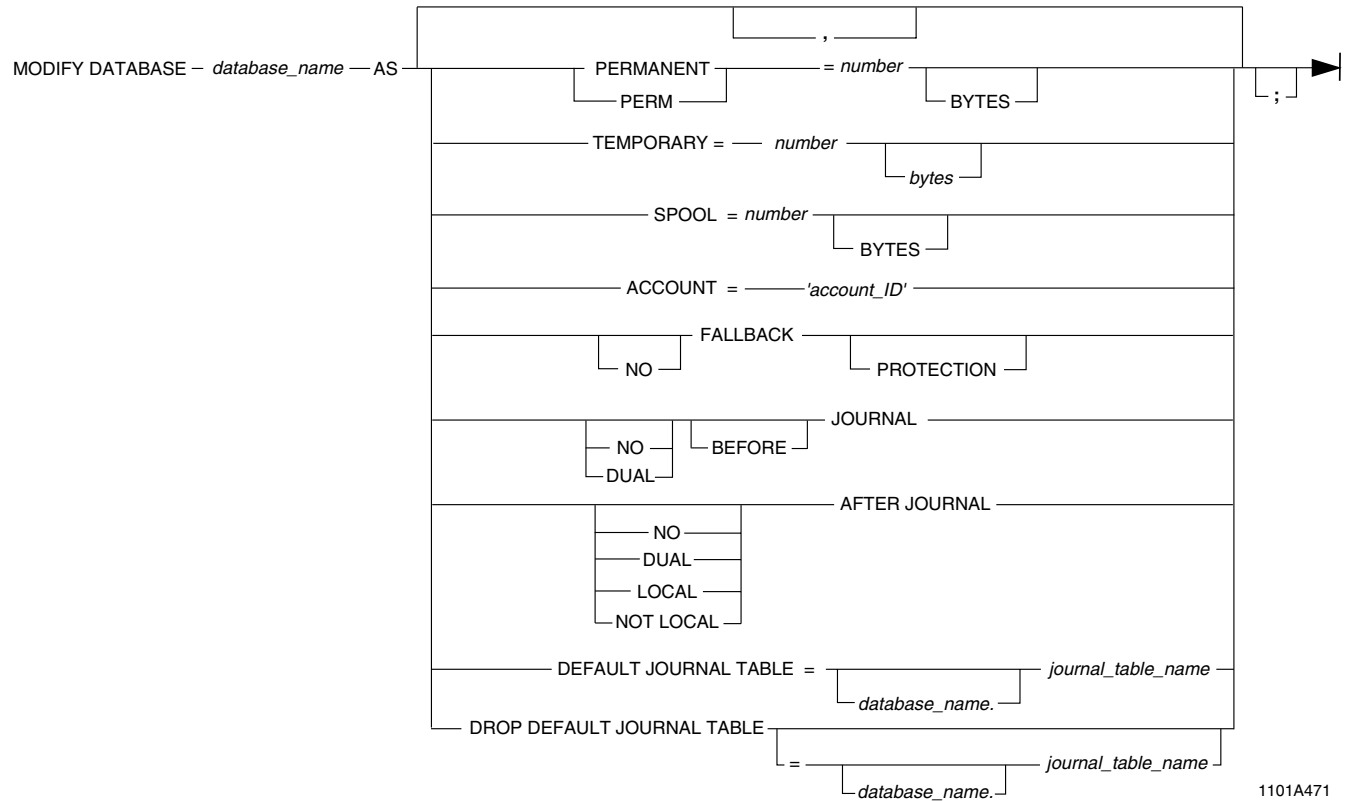
LOGGING ONLINE ARCHIVE OFF



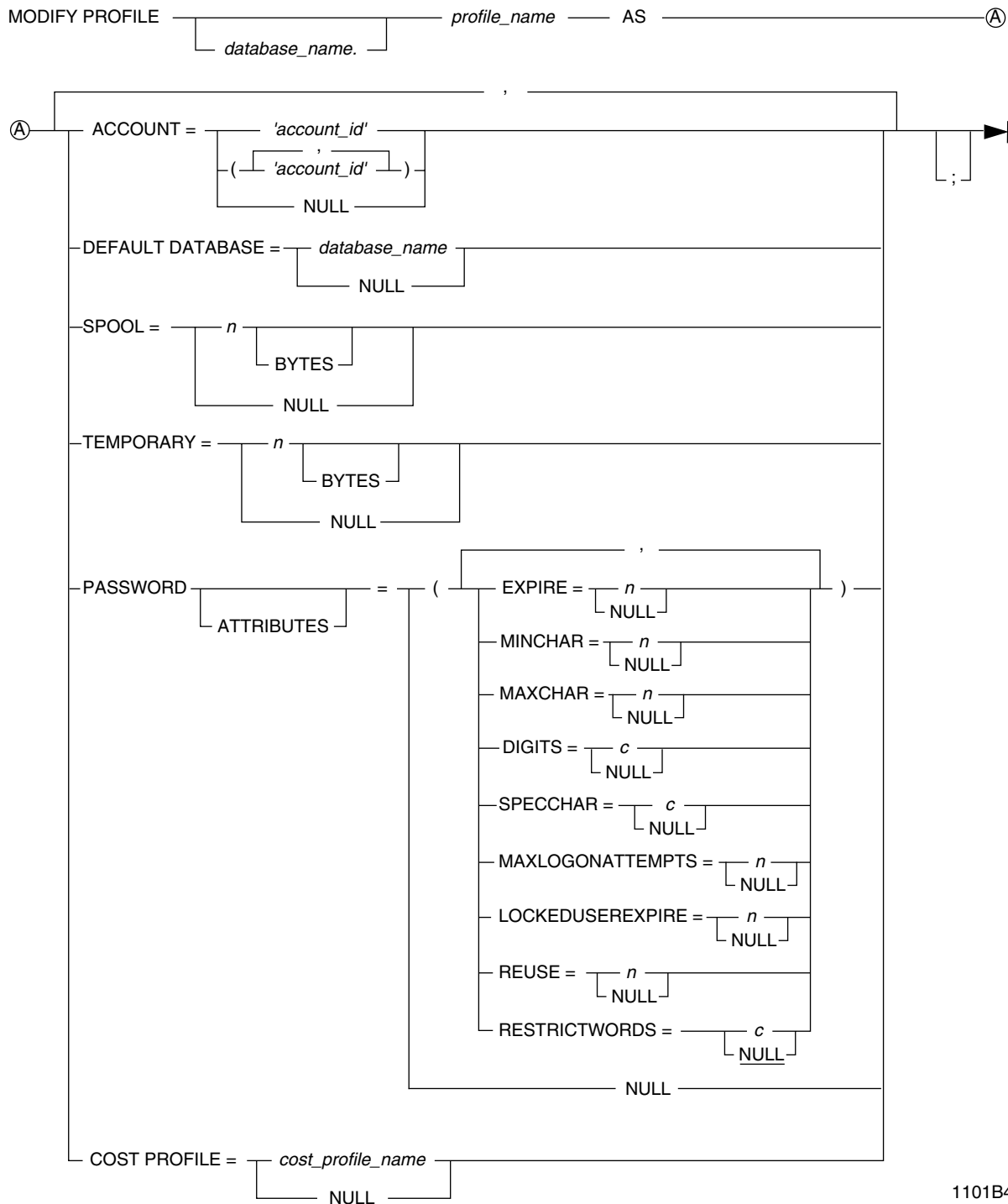
LOGGING ONLINE ARCHIVE ON



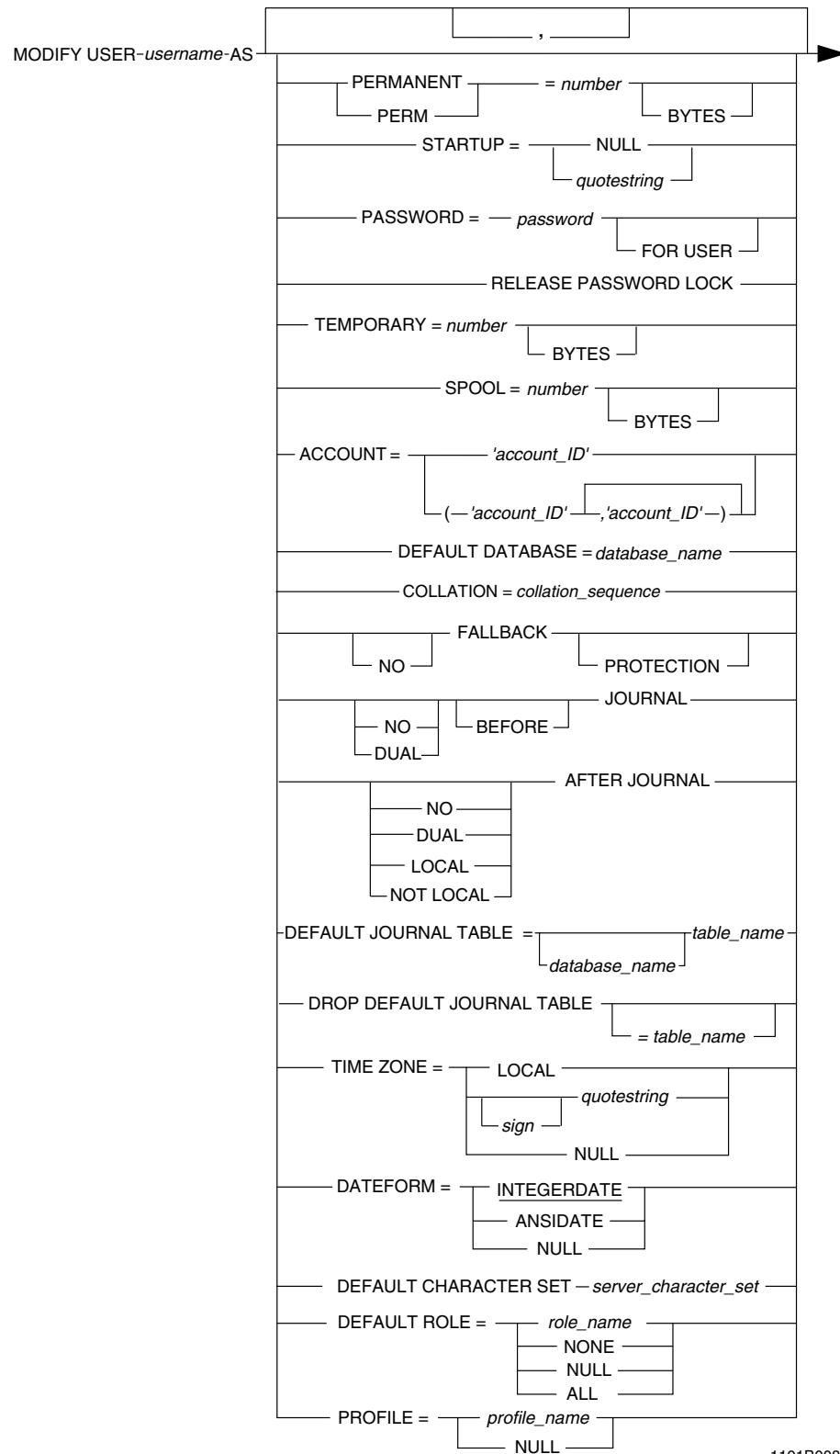
MODIFY DATABASE



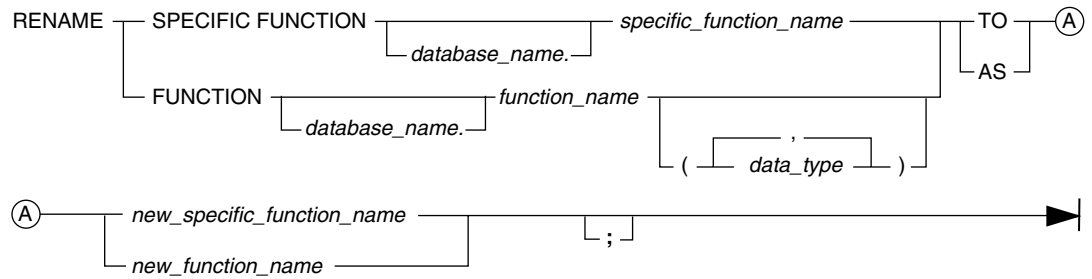
MODIFY PROFILE



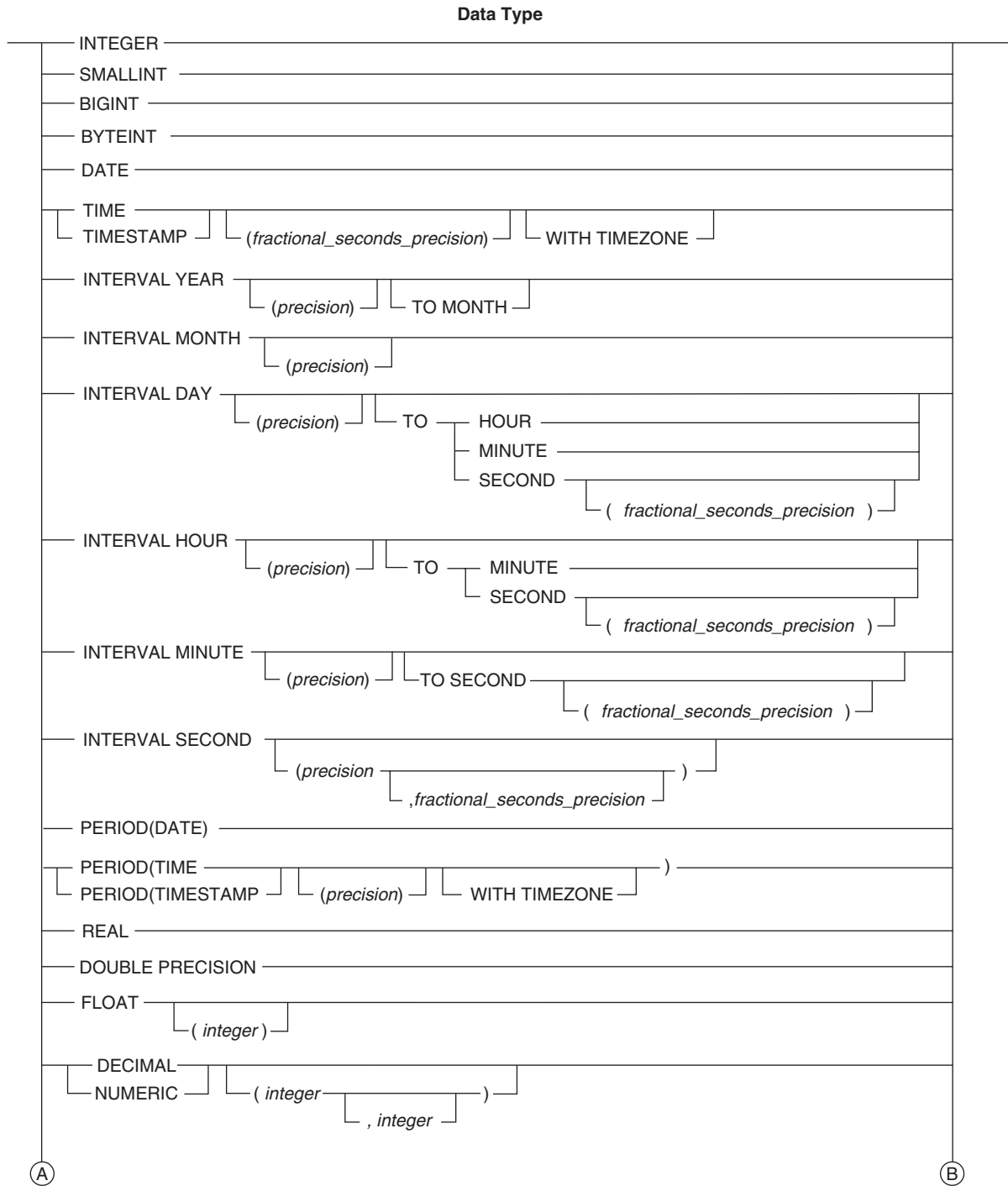
MODIFY USER



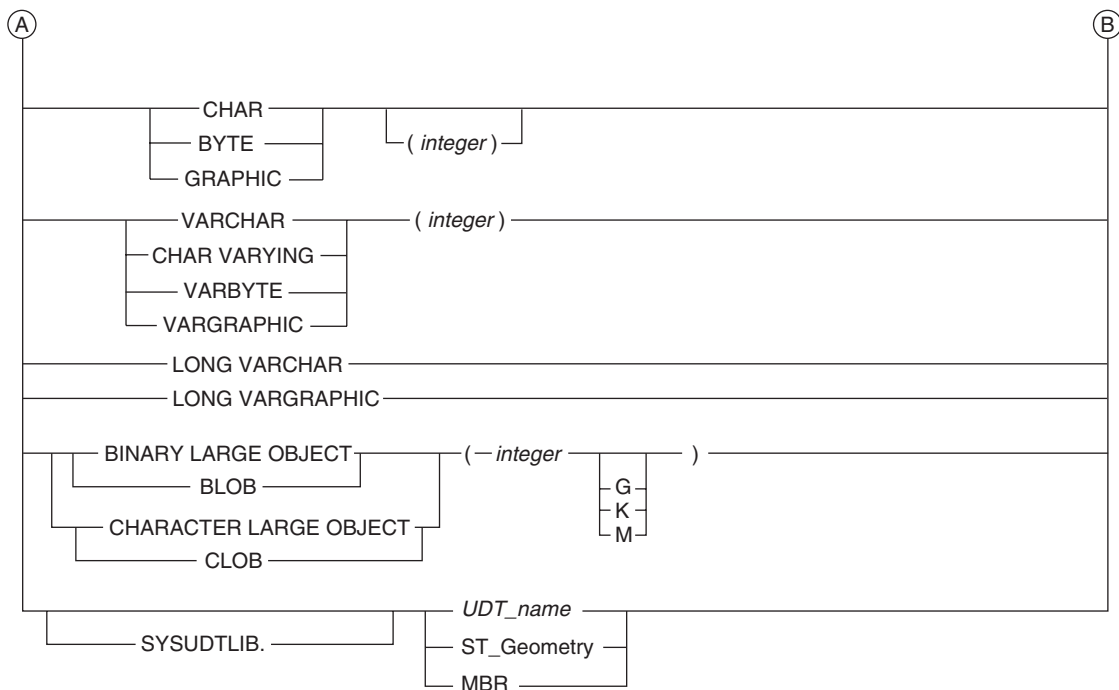
RENAME FUNCTION



1101B105

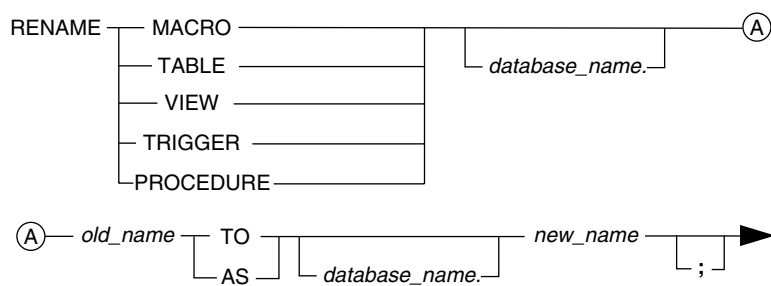


1101A535



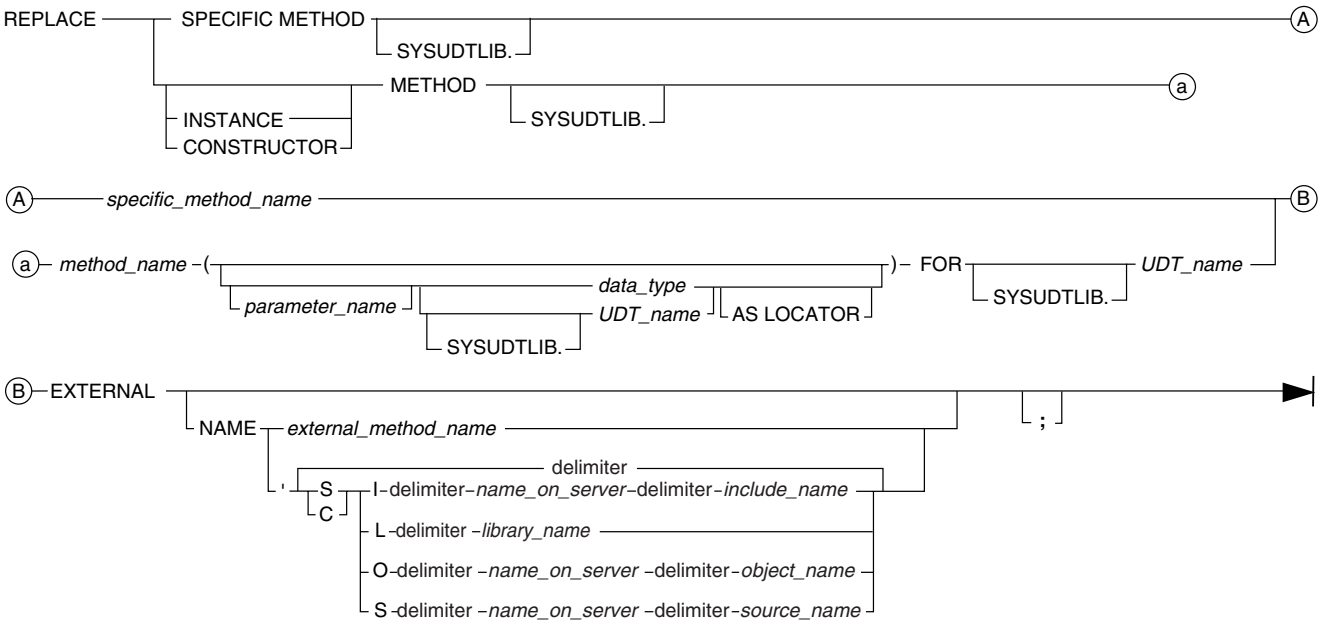
1101A536

RENAME MACRO/ RENAME PROCEDURE/ RENAME TABLE/ RENAME TRIGGER/ RENAME VIEW

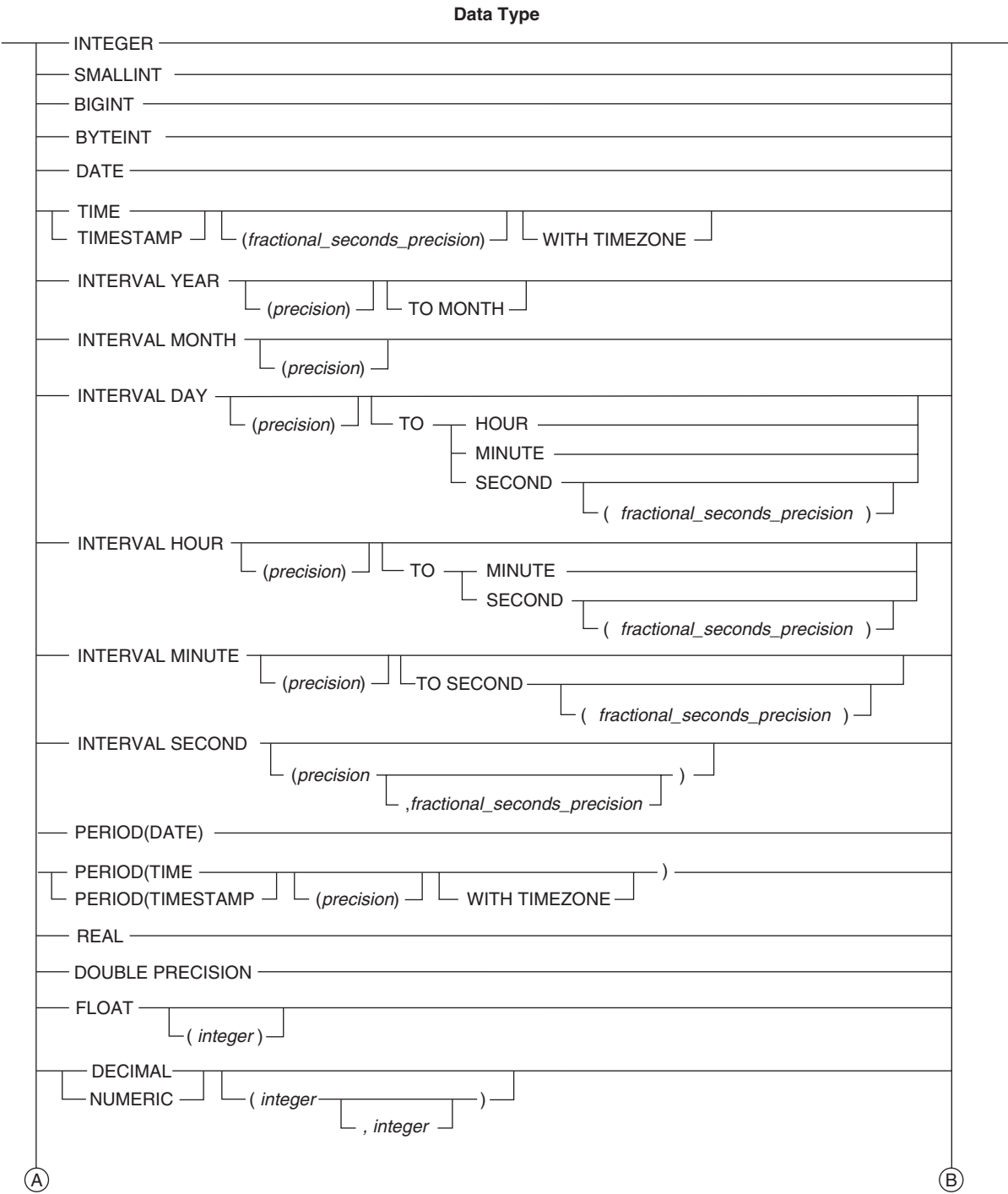


1101G159

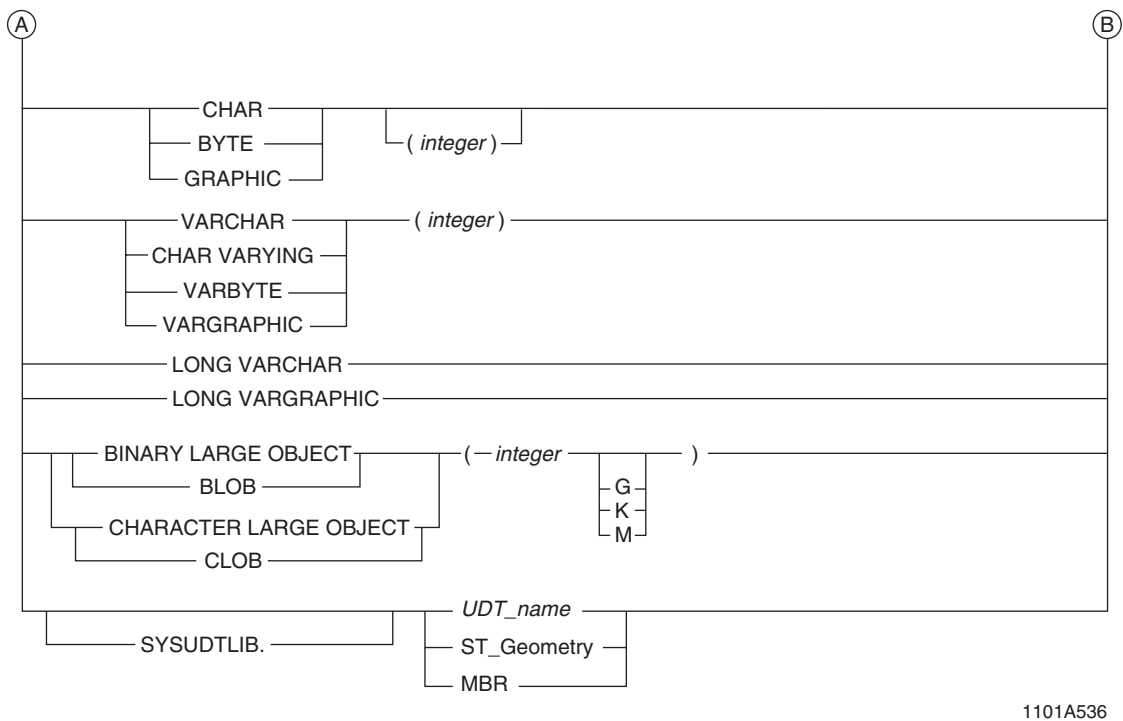
REPLACE METHOD



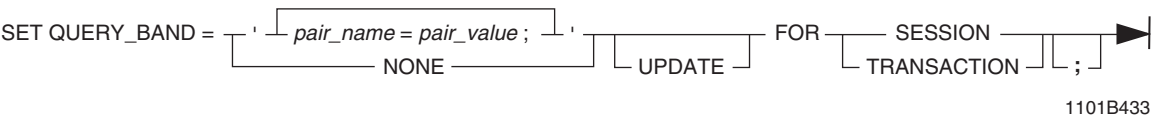
1101B370



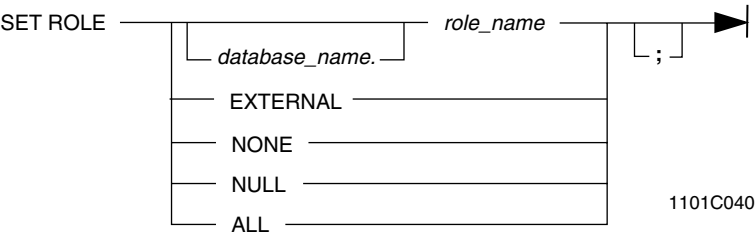
1101A535



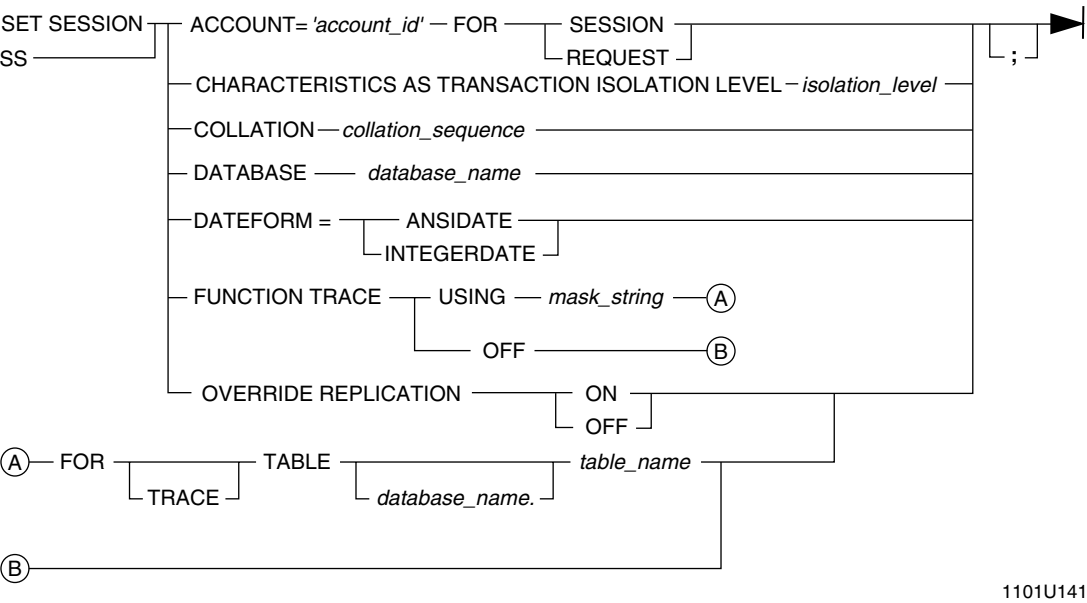
SET QUERY_BAND



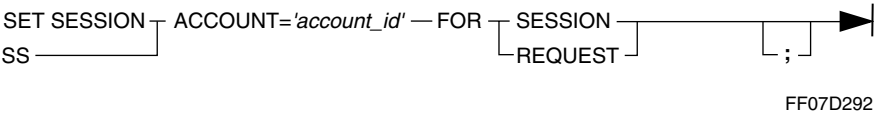
SET ROLE



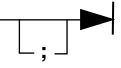
SET SESSION



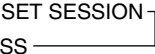
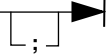
SET SESSION ACCOUNT



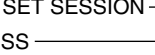
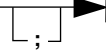
SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL

SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL—*isolation_level* 
1101A315

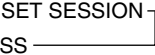
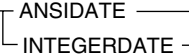
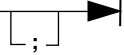
SET SESSION COLLATION

SET SESSION  COLLATION—*collation_sequence* 
FF07D291

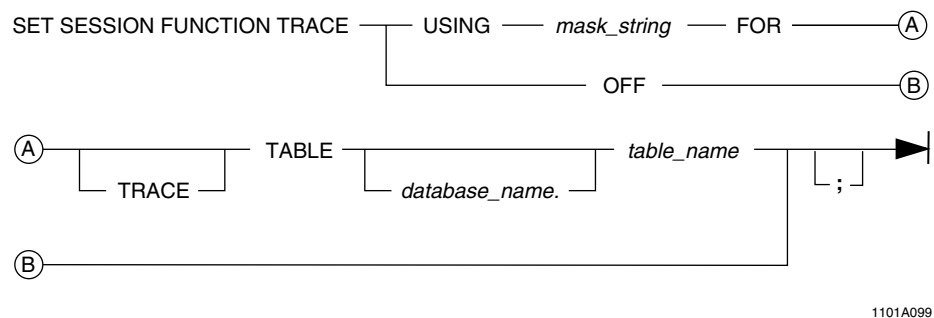
SET SESSION DATABASE

SET SESSION  DATABASE — *database_name* 
1101A076

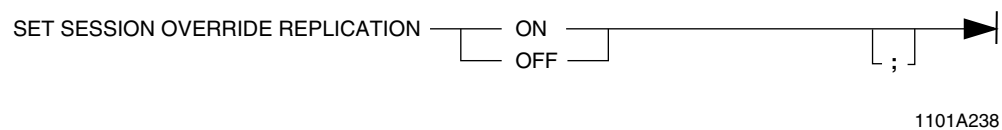
SET SESSION DATEFORM

SET SESSION  DATEFORM=  
FF07D293

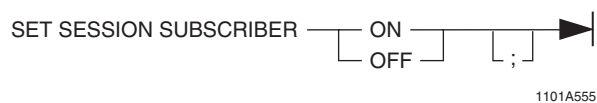
SET SESSION FUNCTION TRACE



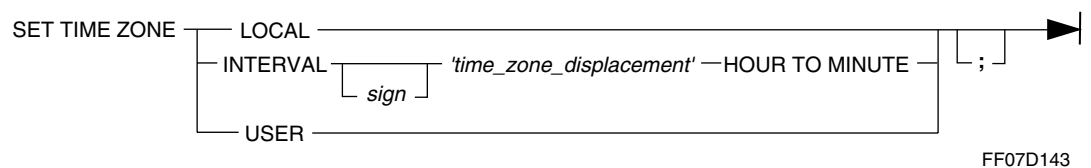
SET SESSION OVERRIDE REPLICATION



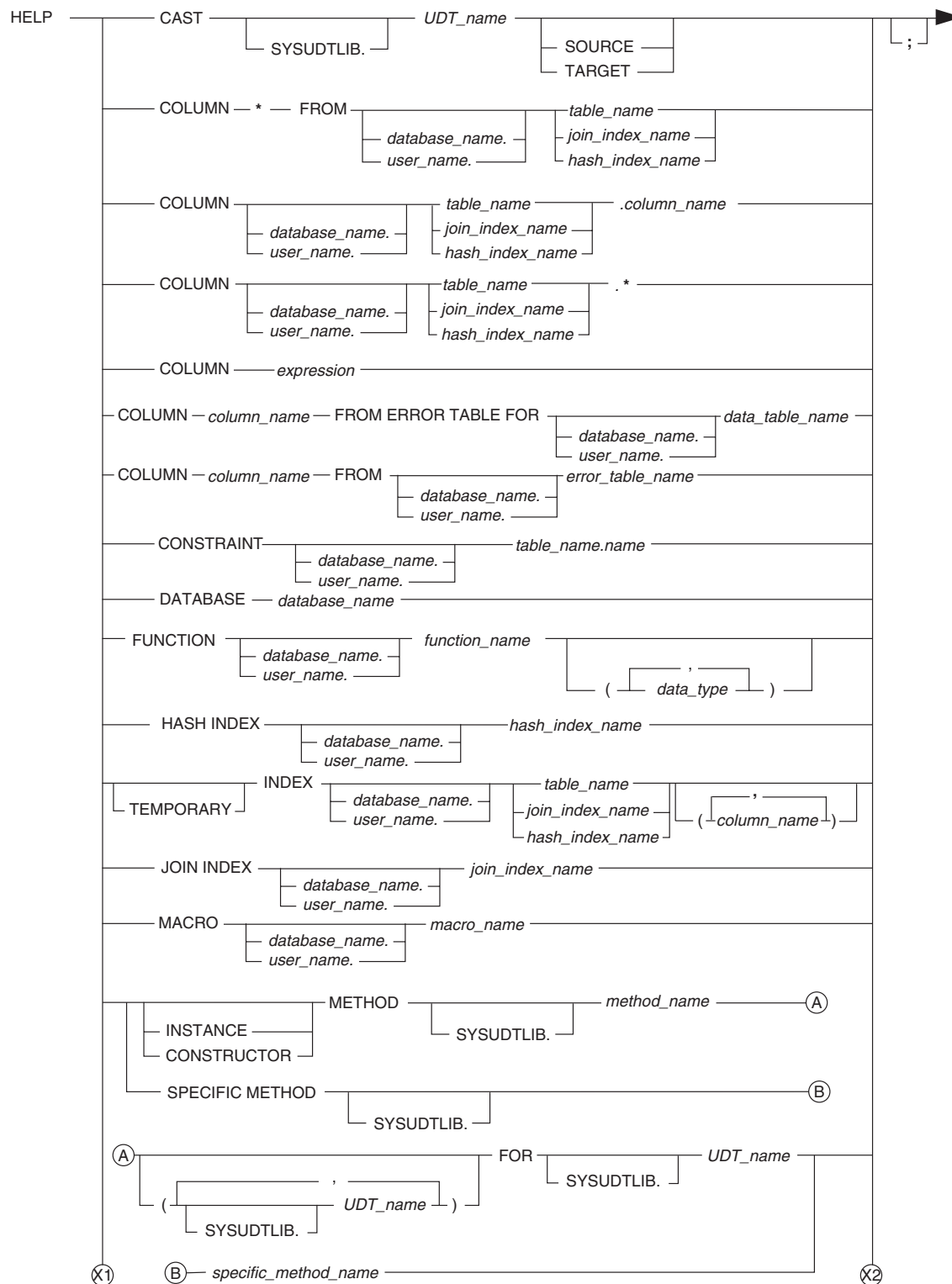
SET SESSION SUBSCRIBER



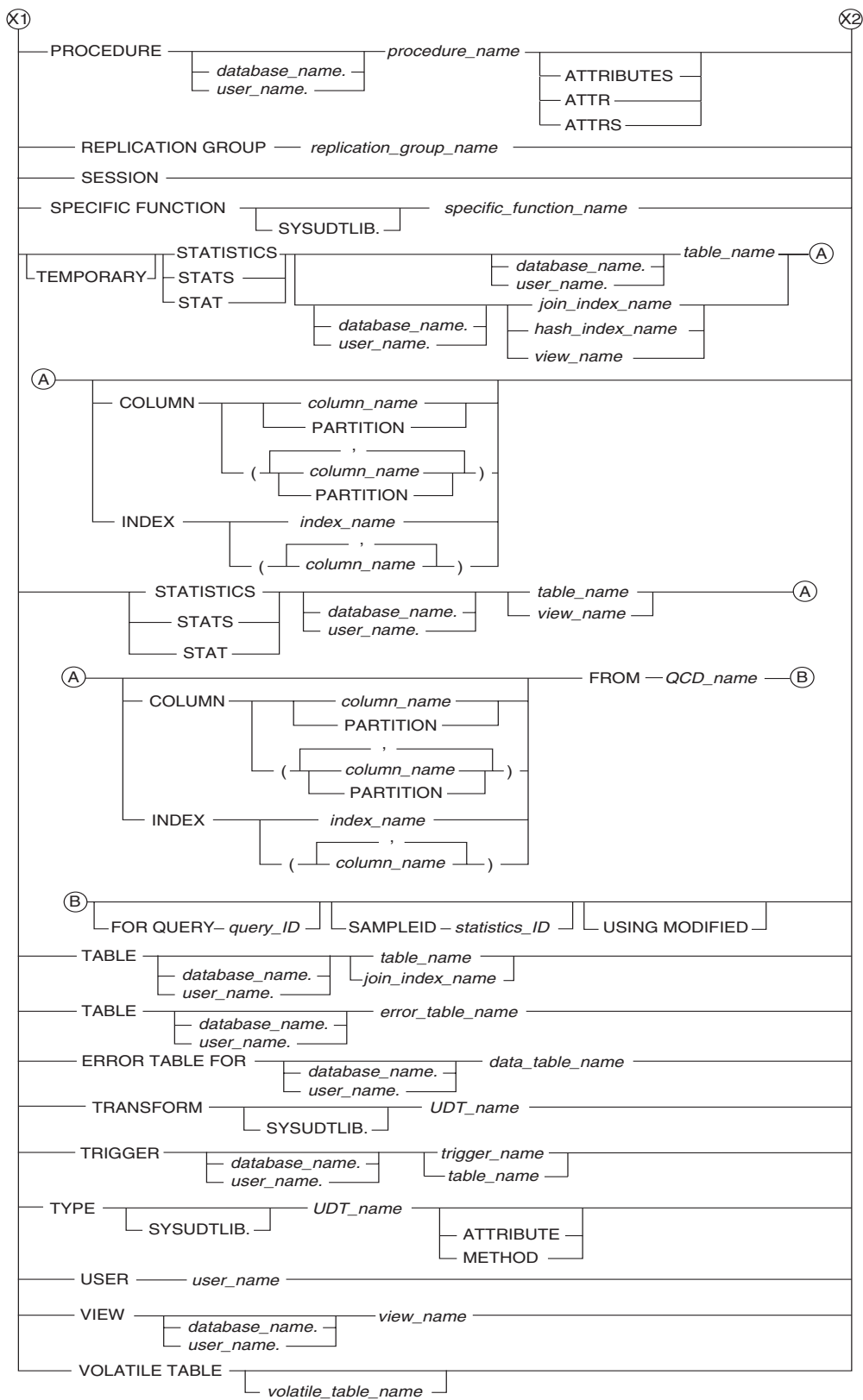
SET TIME ZONE



HELP

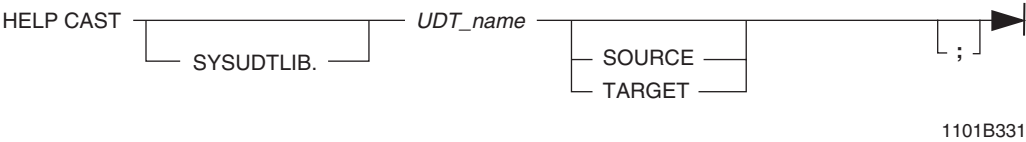


1101C387



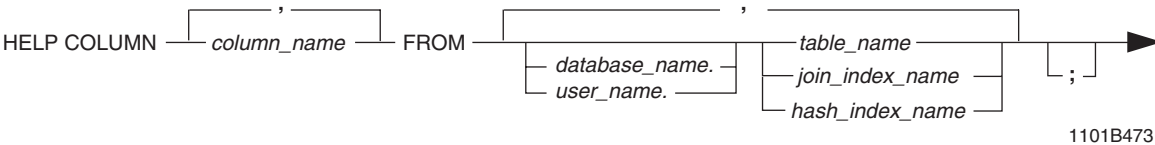
1101F388

HELP CAST

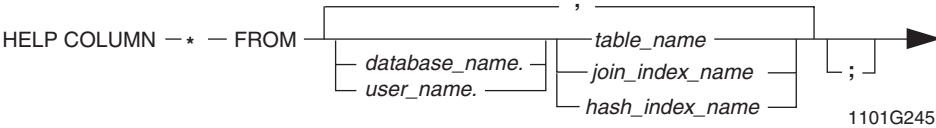


HELP COLUMN

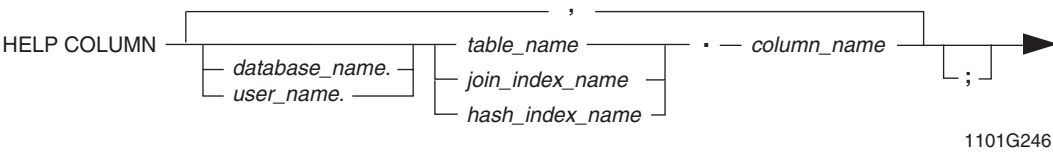
Syntax 1



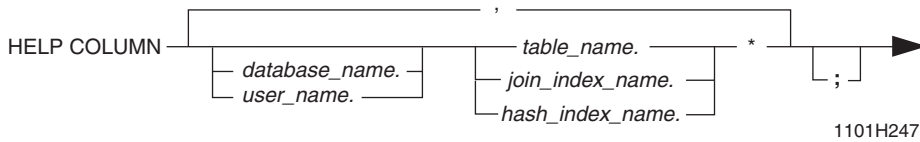
Syntax 2



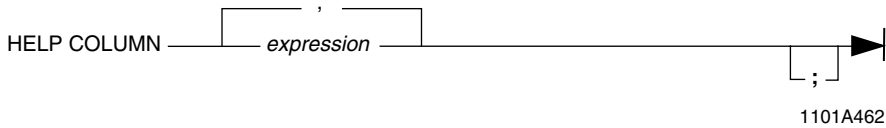
Syntax 3



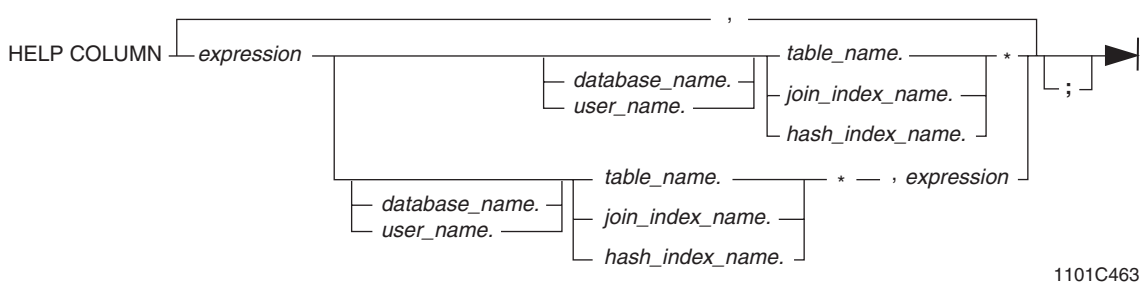
Syntax 4



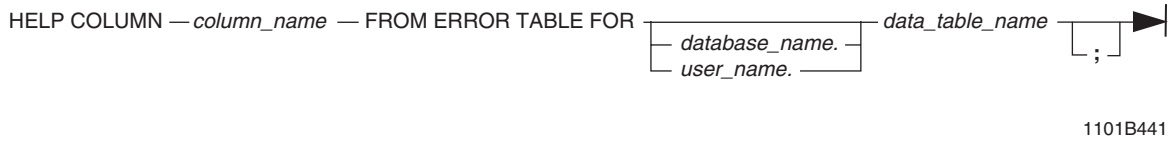
Syntax 5



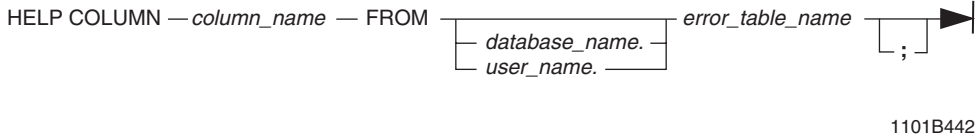
Syntax 6



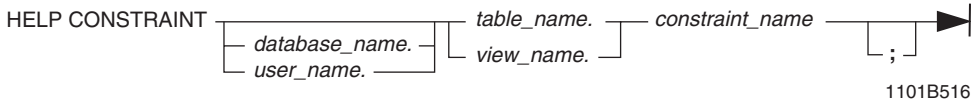
Syntax 7



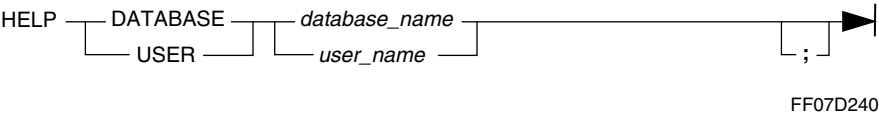
Syntax 8



HELP CONSTRAINT

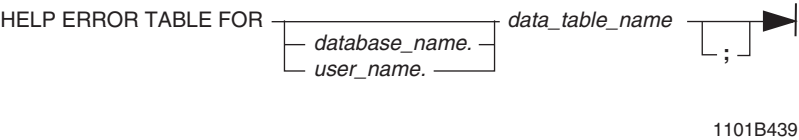


HELP DATABASE/ HELP USER

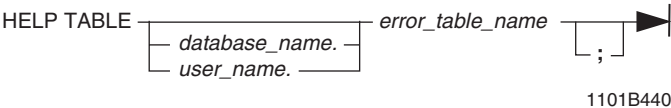


HELP ERROR TABLE

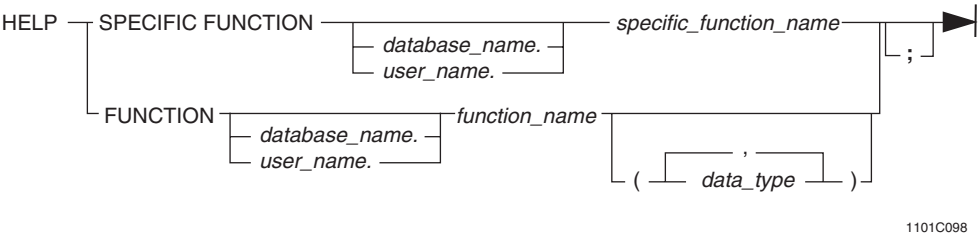
Syntax 1

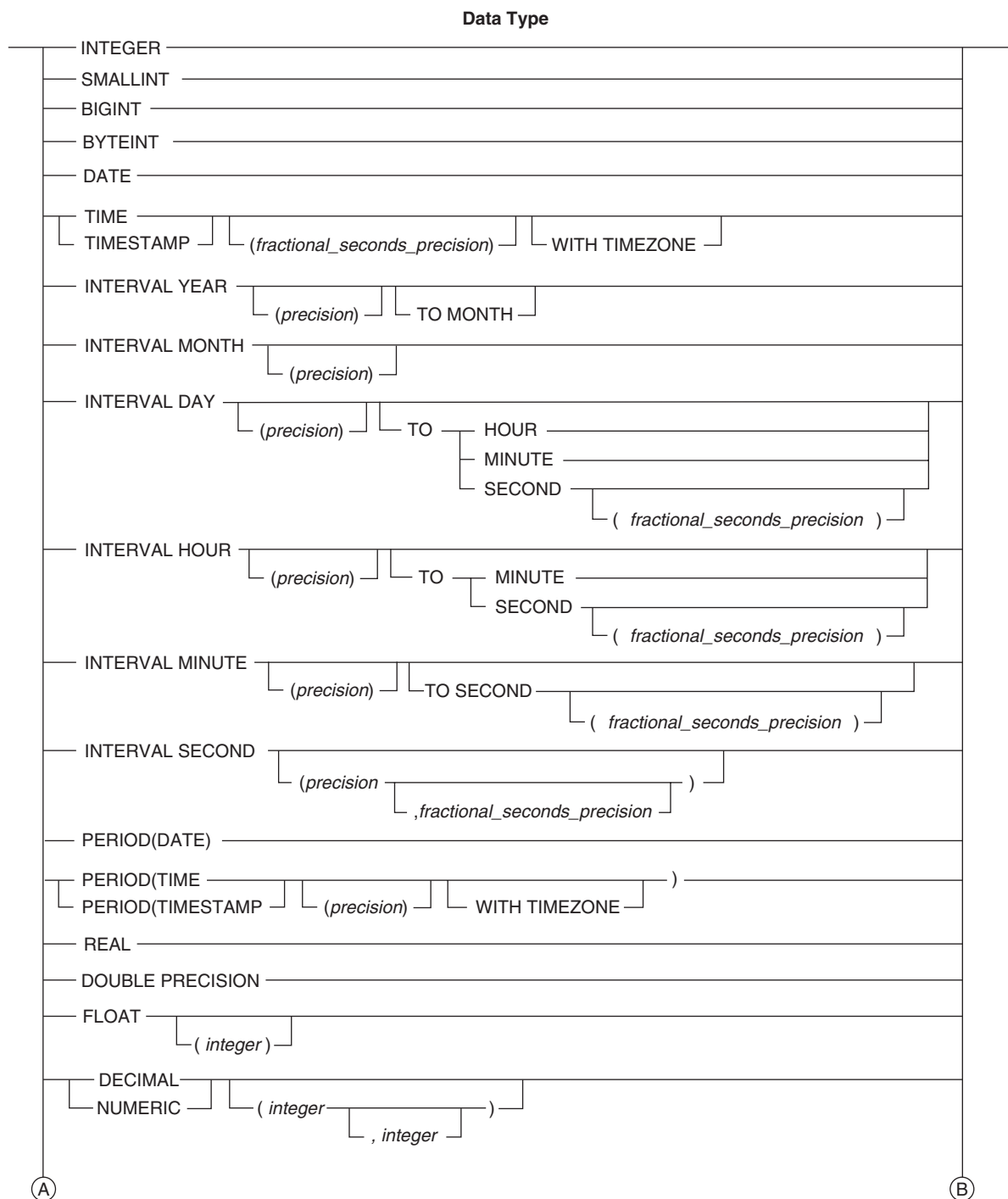


Syntax 2

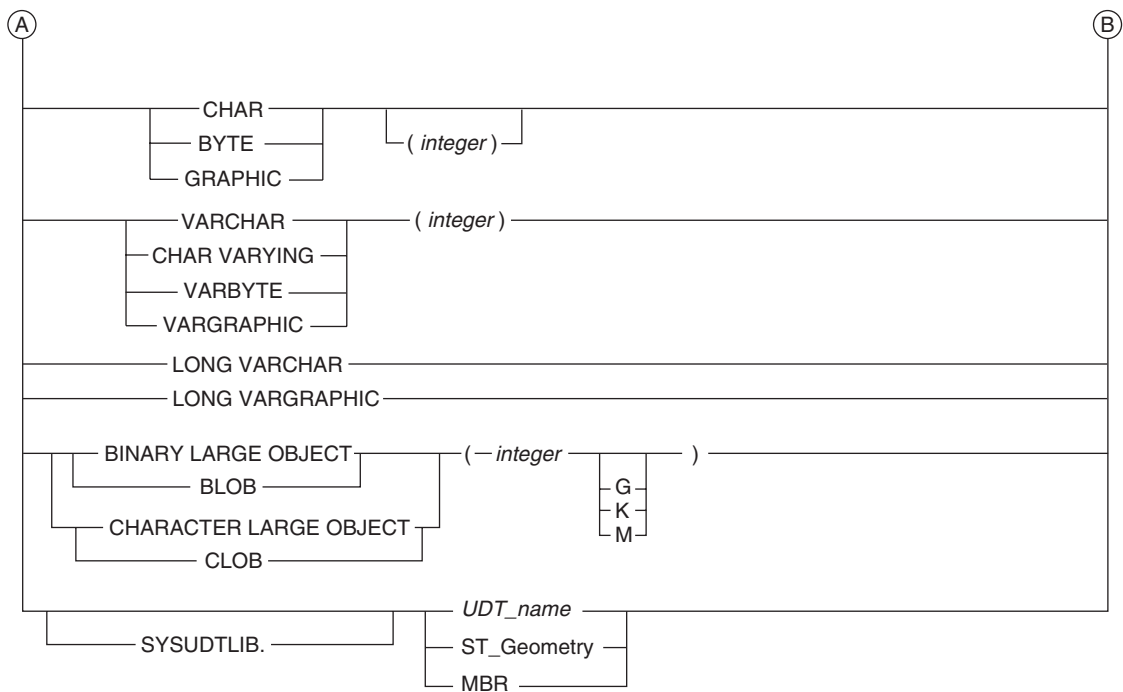


HELP FUNCTION



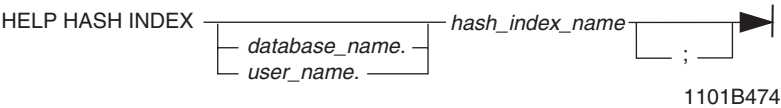


1101A535



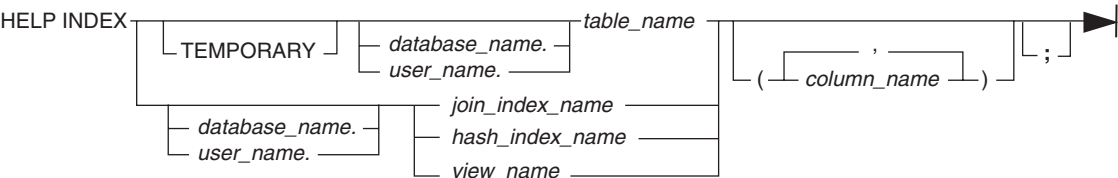
1101A536

HELP HASH INDEX



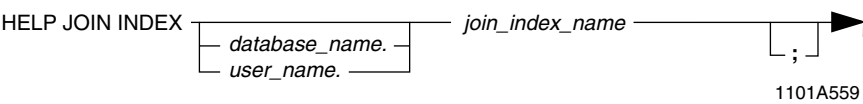
1101B474

HELP INDEX

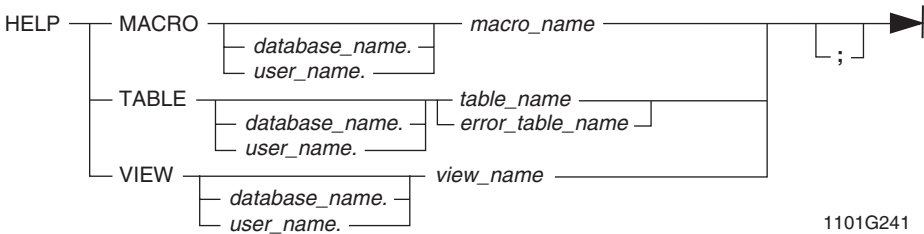


1101B477

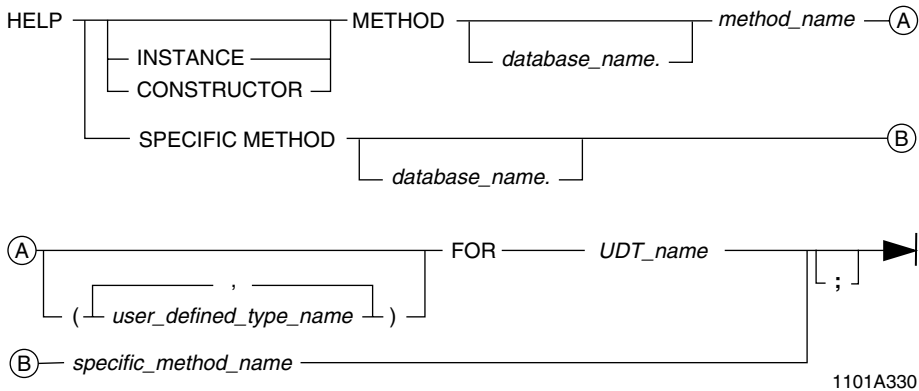
HELP JOIN INDEX



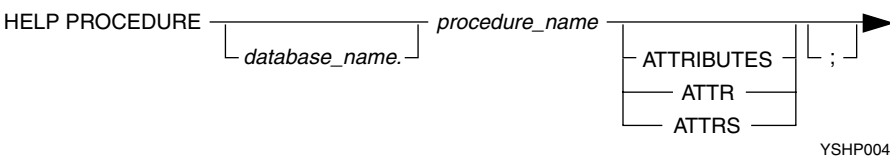
HELP MACRO/ HELP TABLE/ HELP VIEW



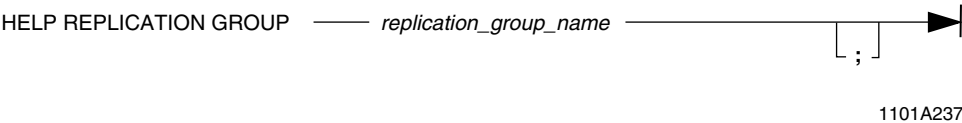
HELP METHOD



HELP PROCEDURE



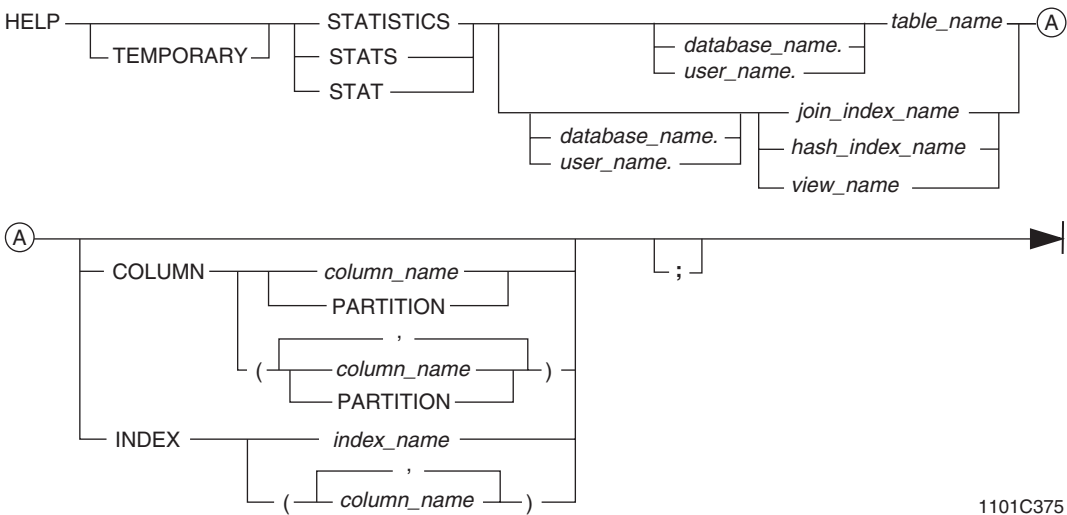
HELP REPLICATION GROUP



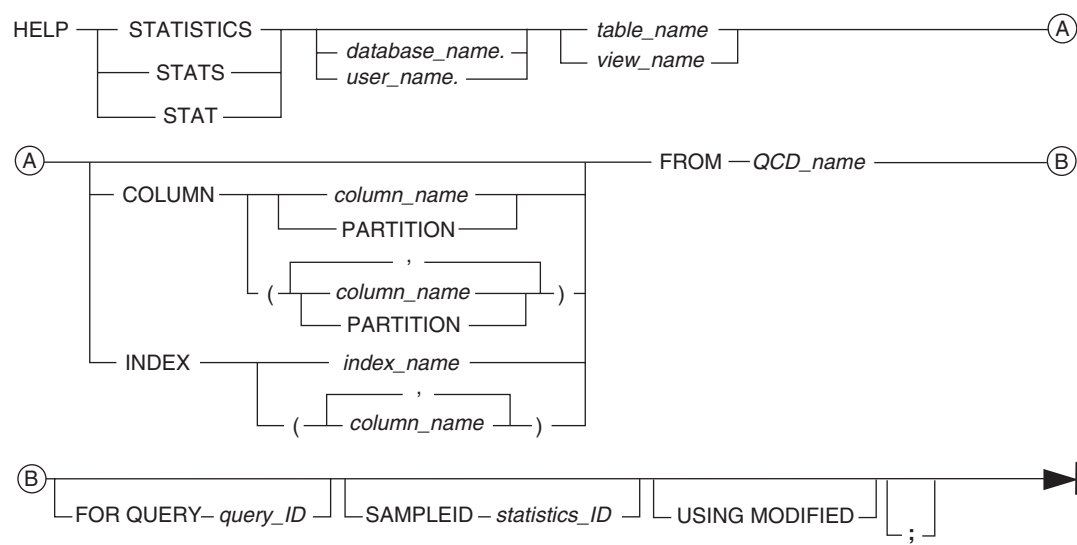
HELP SESSION



HELP STATISTICS (Optimizer Form)

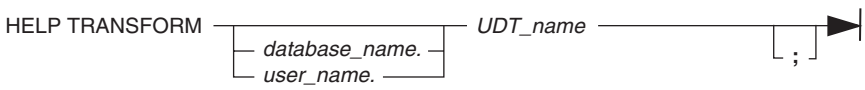


HELP STATISTICS (QCD Form)



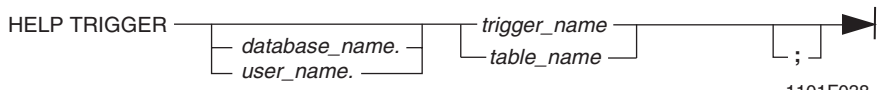
1101J243

HELP TRANSFORM



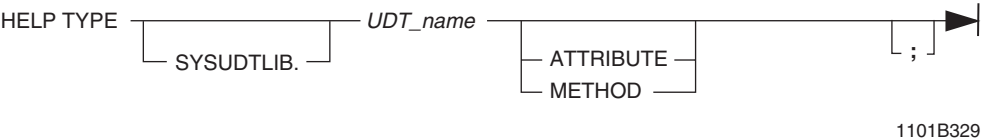
1101B332

HELP TRIGGER

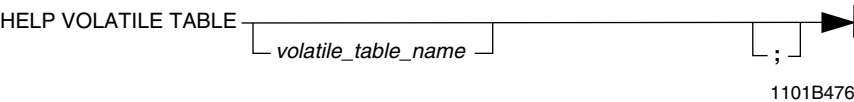


1101F038

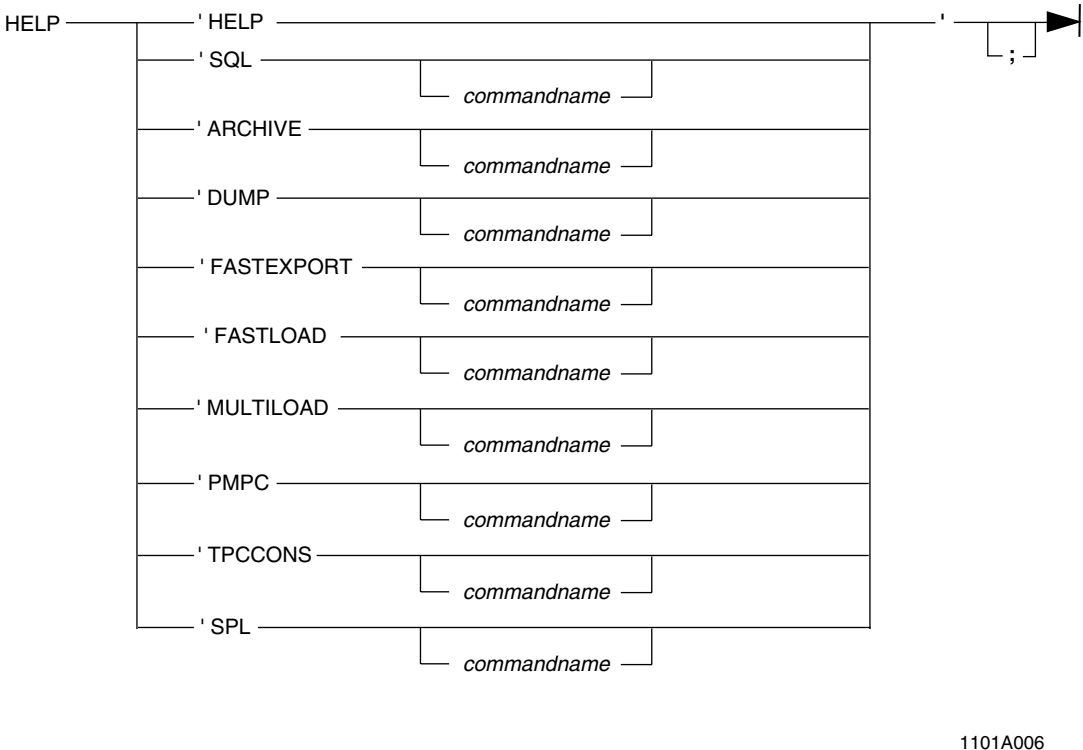
HELP TYPE



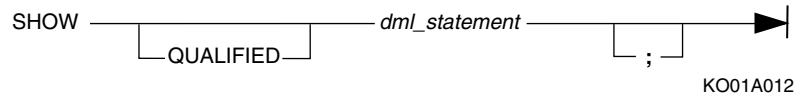
HELP VOLATILE TABLE



HELP (Online Form)

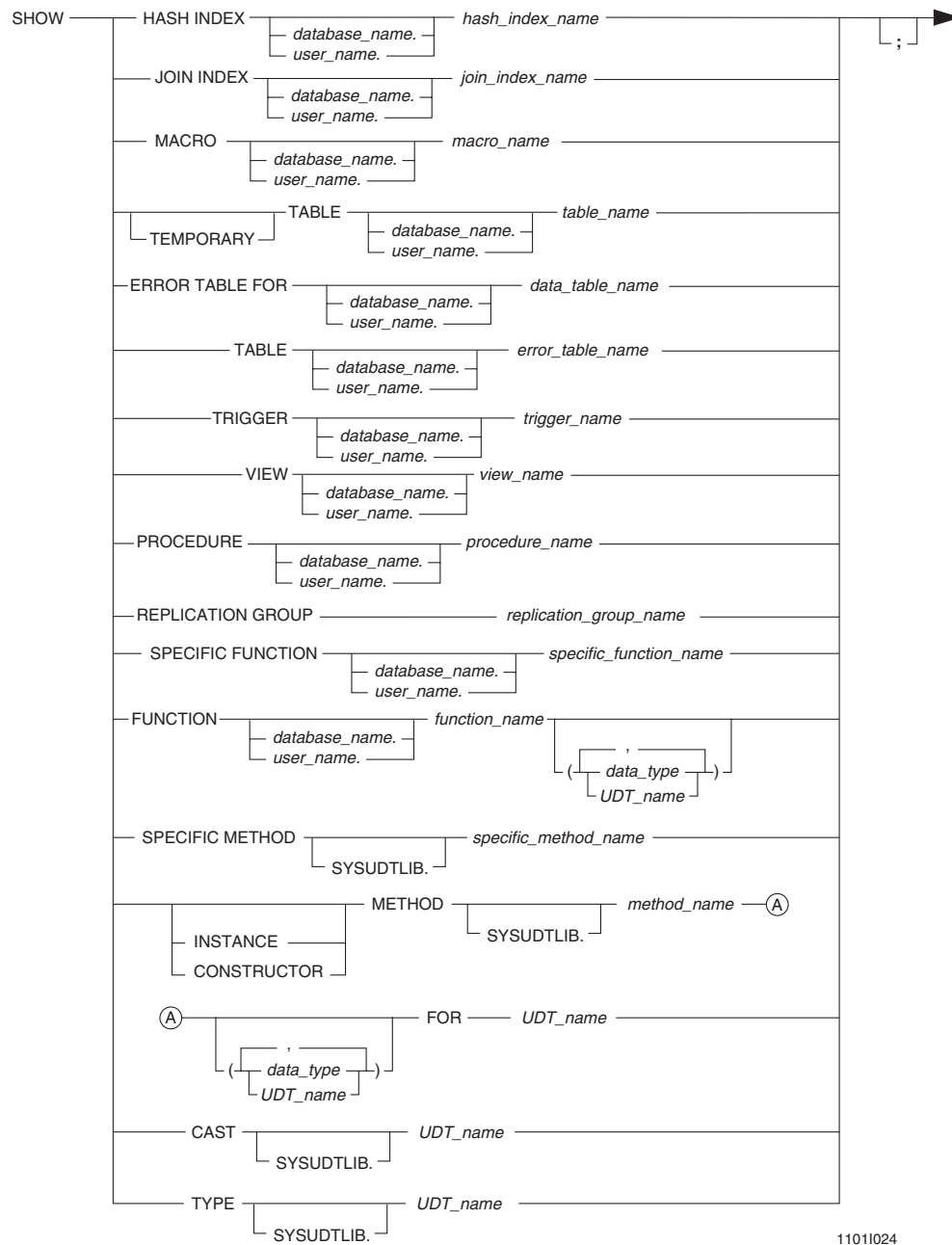


SHOW

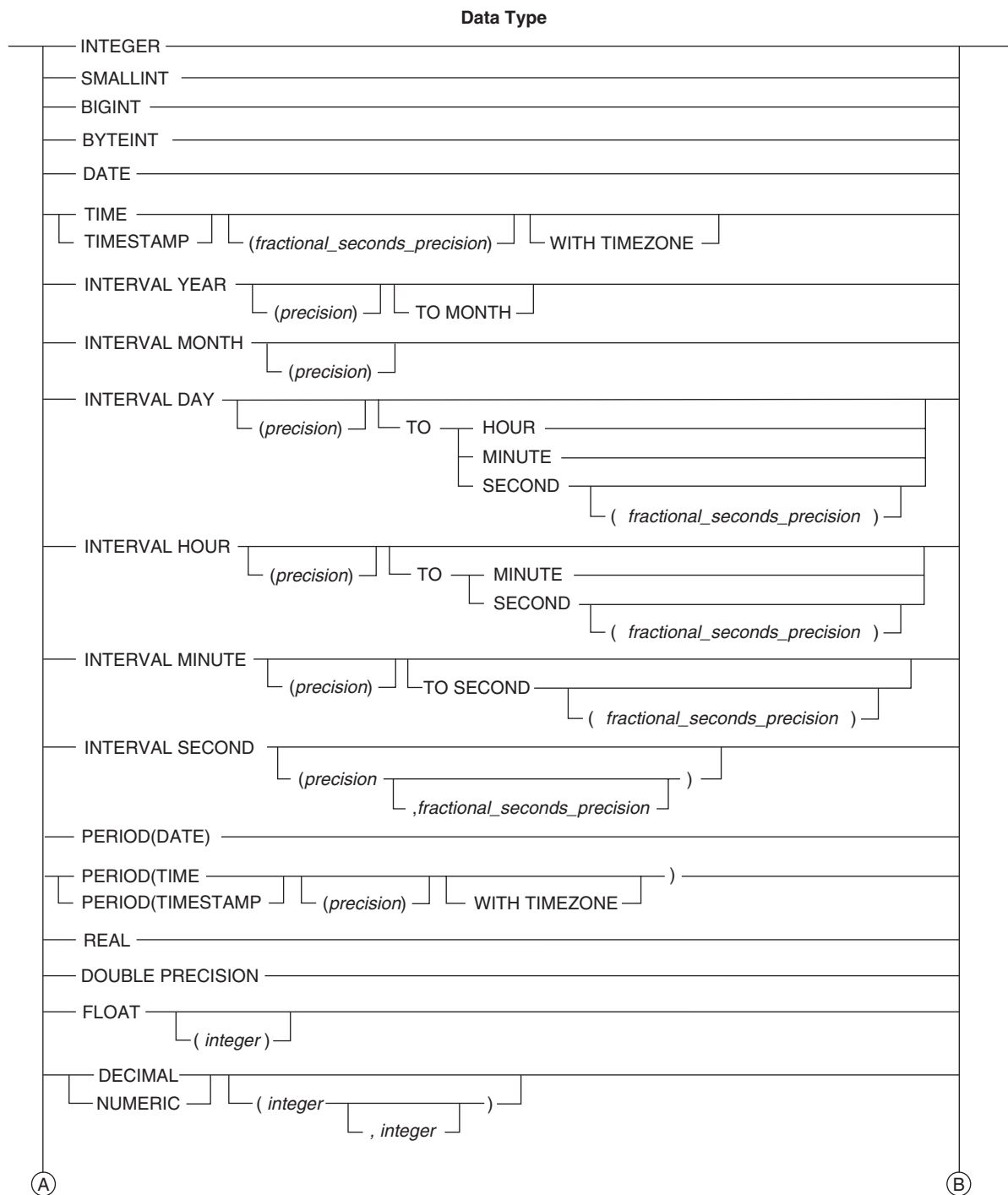


**SHOW CAST/
SHOW ERROR TABLE/
SHOW FUNCTION/
SHOW HASH INDEX/
SHOW JOIN INDEX/
SHOW MACRO/
SHOW METHOD/
SHOW PROCEDURE/
SHOW REPLICATION GROUP/
SHOW TABLE/
SHOW TRIGGER/
SHOW TYPE/
SHOW VIEW**

General Syntax

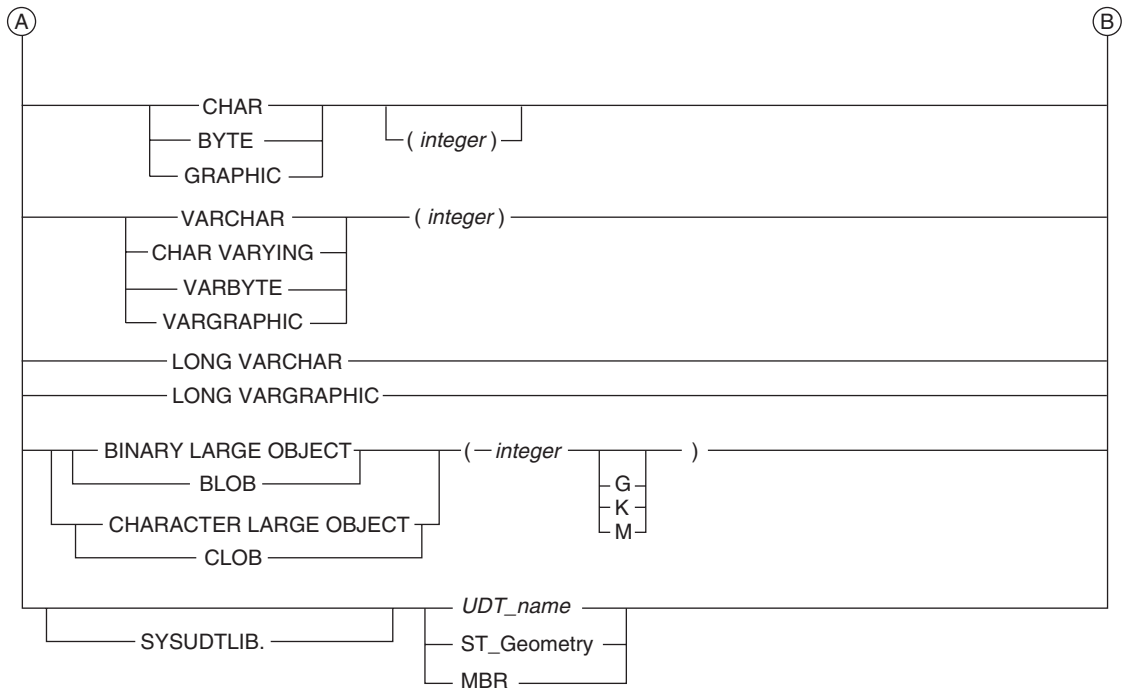


11011024



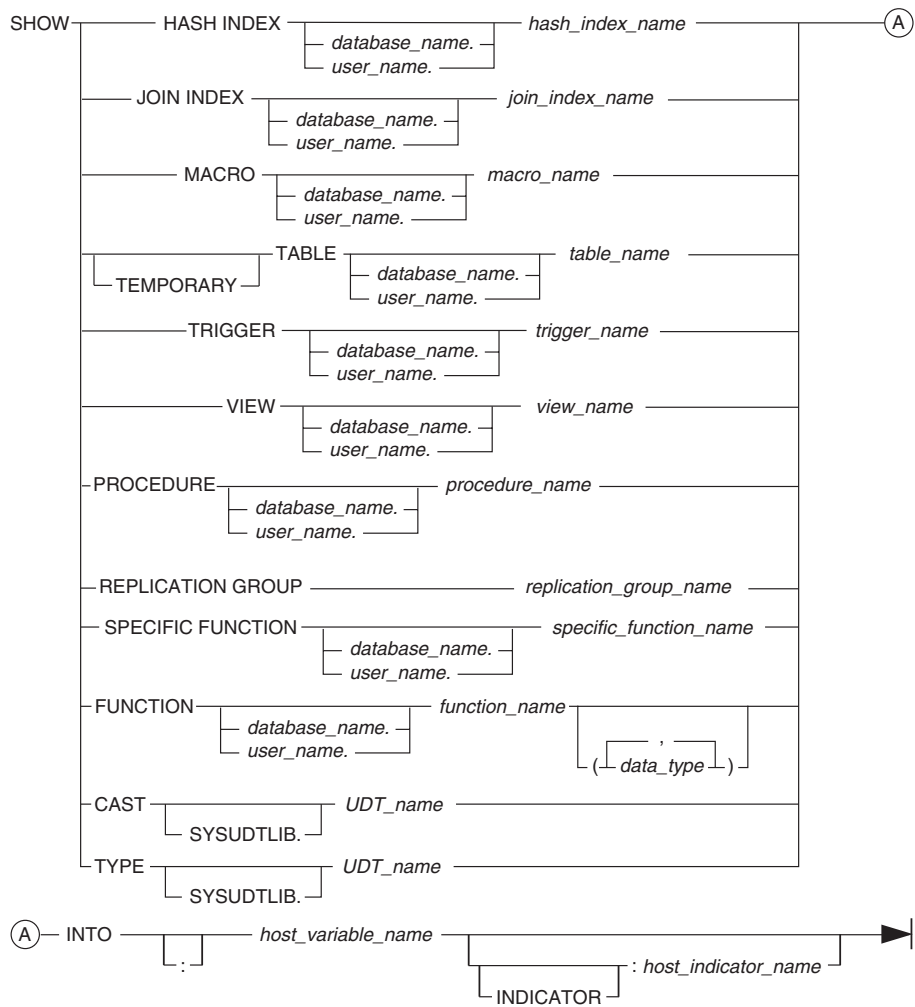
1101A535

SHOW CAST/ SHOW ERROR TABLE/ SHOW FUNCTION/ SHOW HASH INDEX/ SHOW JOIN INDEX/ SHOW MACRO/ SHOW METHOD/



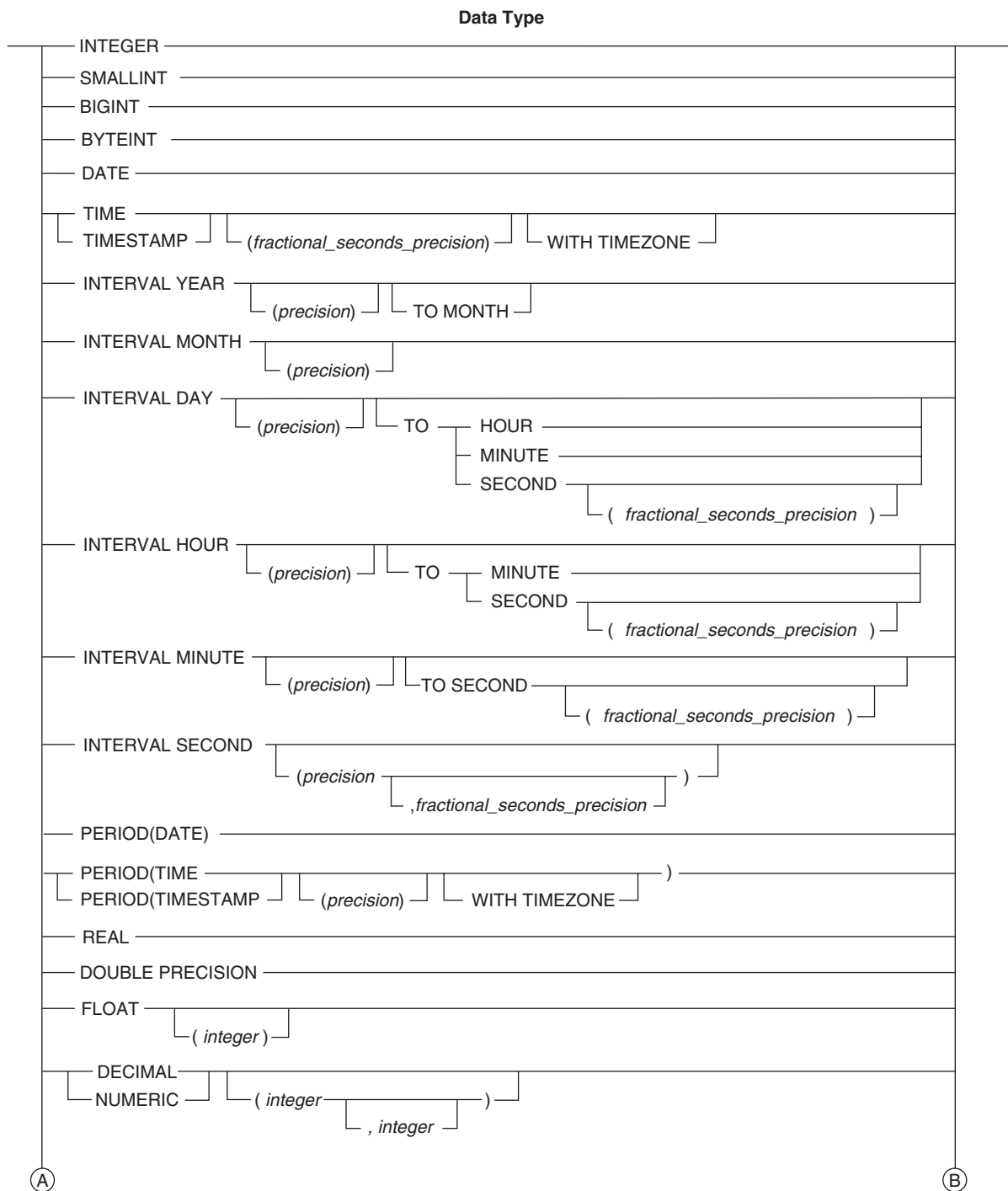
1101A536

Embedded SQL Syntax

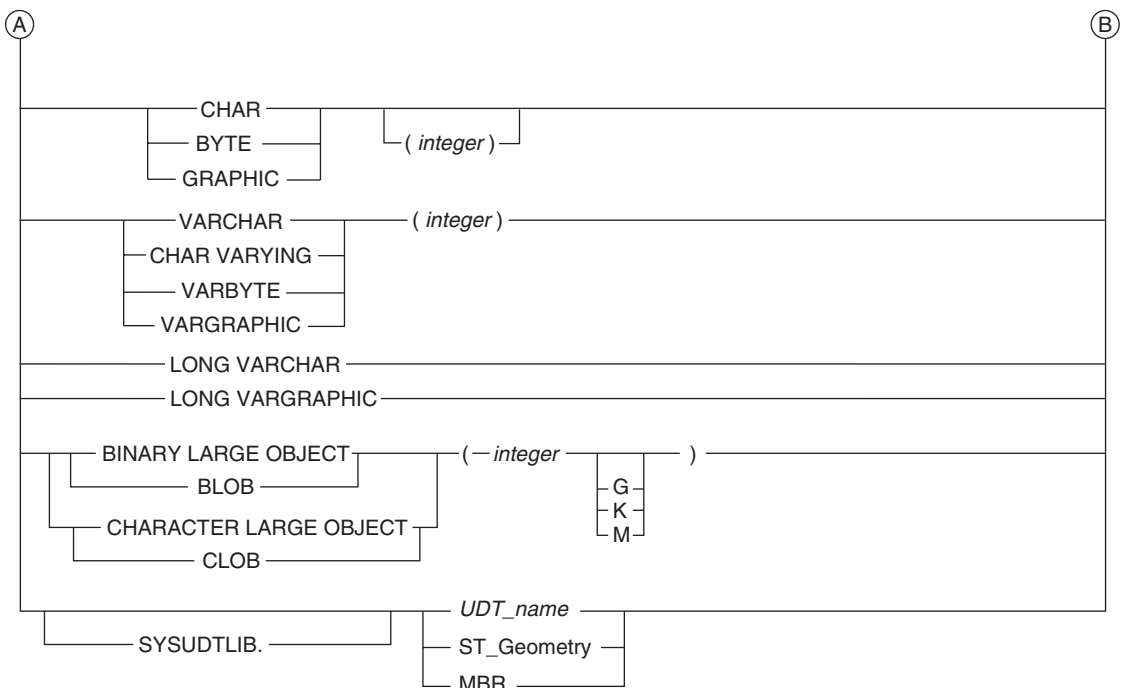


11011023

SHOW CAST/ SHOW ERROR TABLE/ SHOW FUNCTION/ SHOW HASH INDEX/ SHOW JOIN INDEX/ SHOW MACRO/ SHOW METHOD/

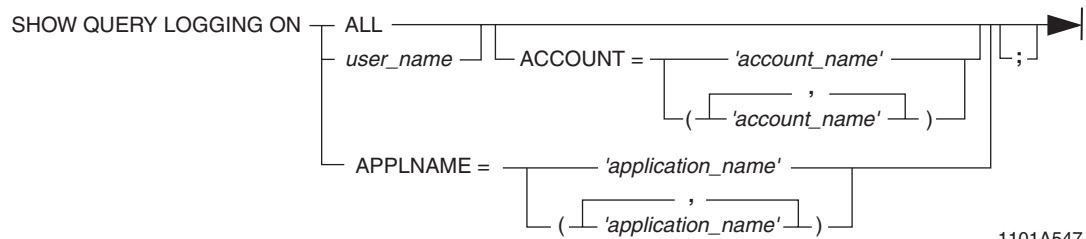


1101A535



1101A536

SHOW QUERY LOGGING



1101A547

CHAPTER 4 SQL Data Control Language

GIVE

GIVE database_name TO recipient_name ;

FF07A025

GRANT

Monitor Form

GRANT MONITOR TO (A)

PRIVILEGES

BUT NOT monitor_privilege

monitor_privilege

(A) ALL user_name PUBLIC WITH GRANT OPTION ;

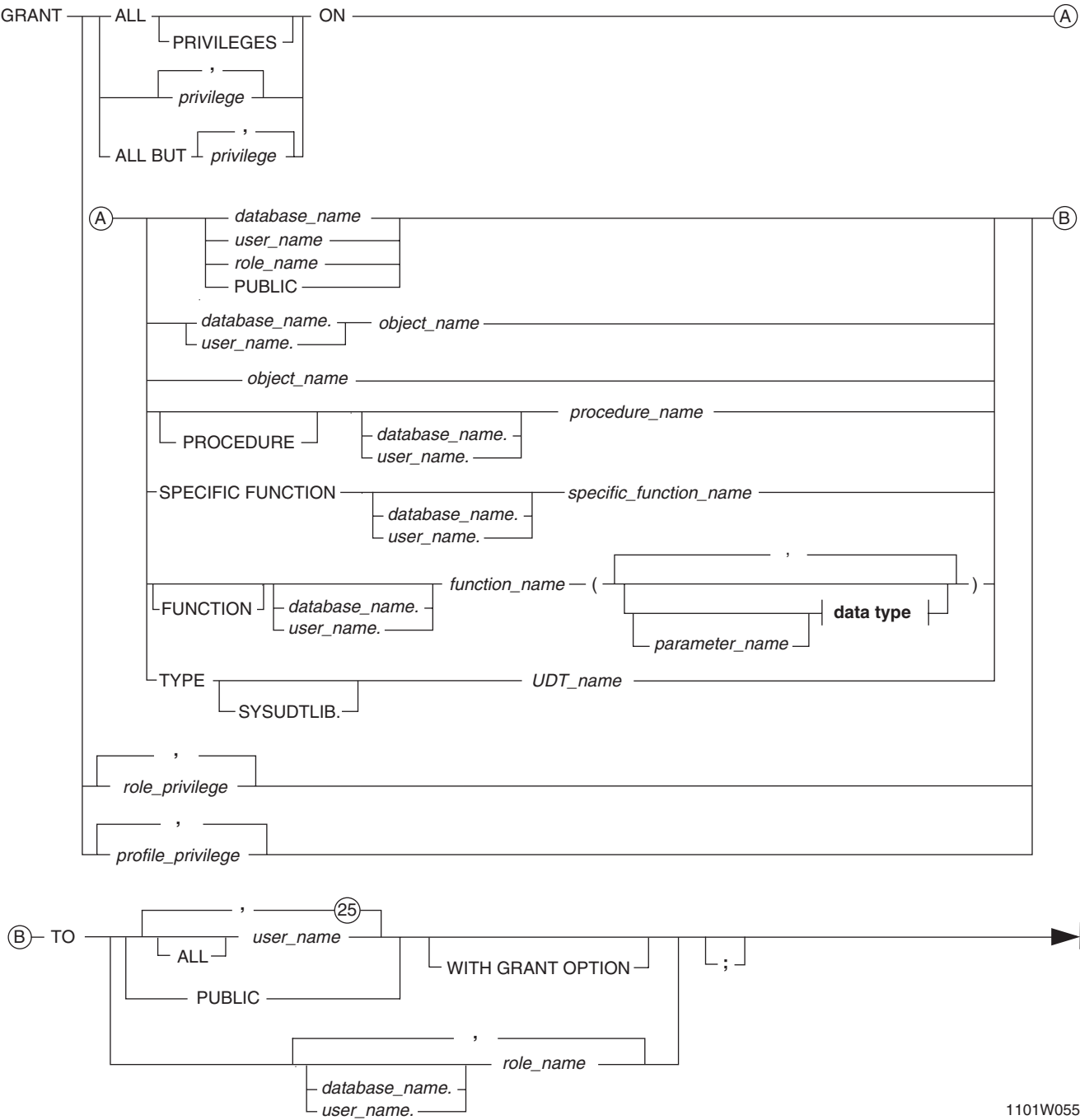
FF07A056

Role Form

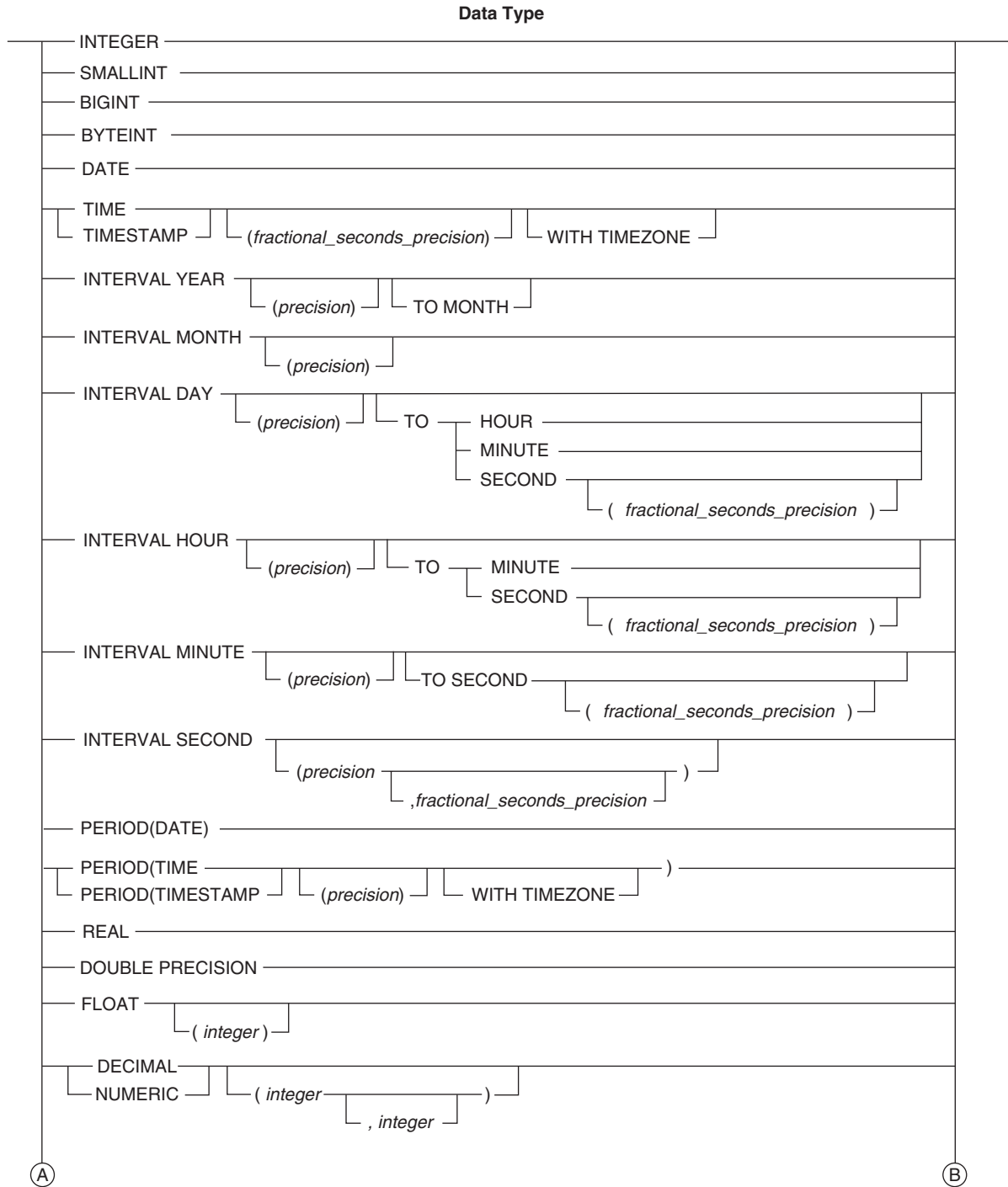
GRANT role_name TO user_name role_name WITH ADMIN OPTION ;

KZ01a008

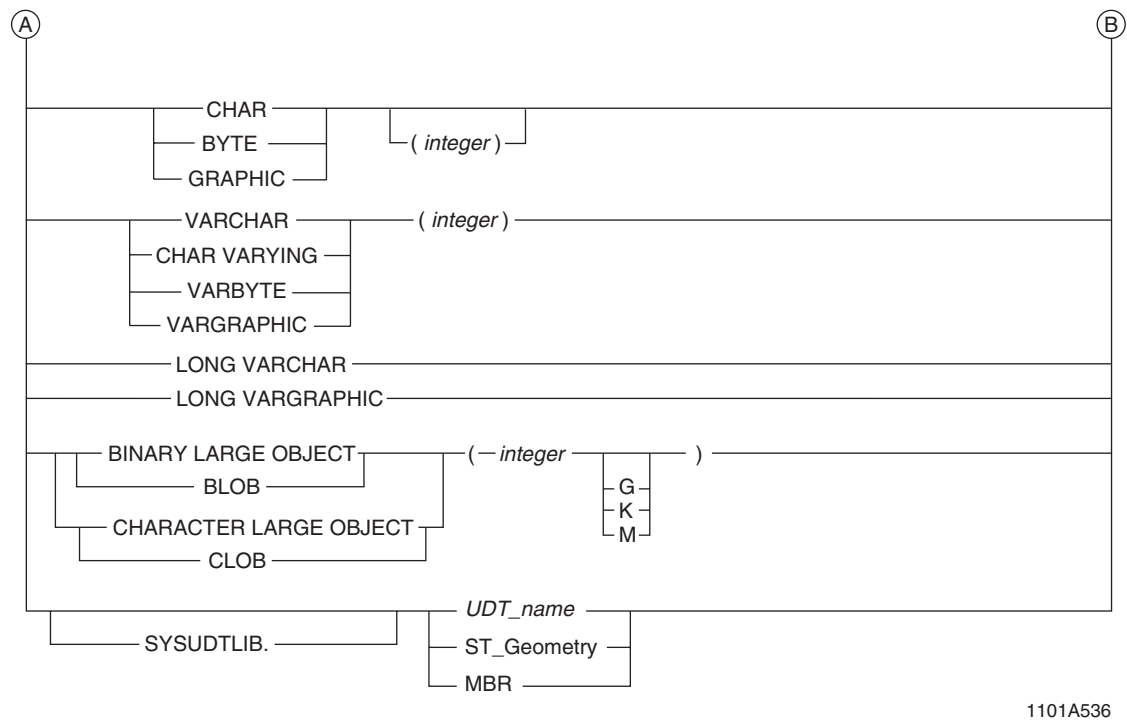
SQL Form



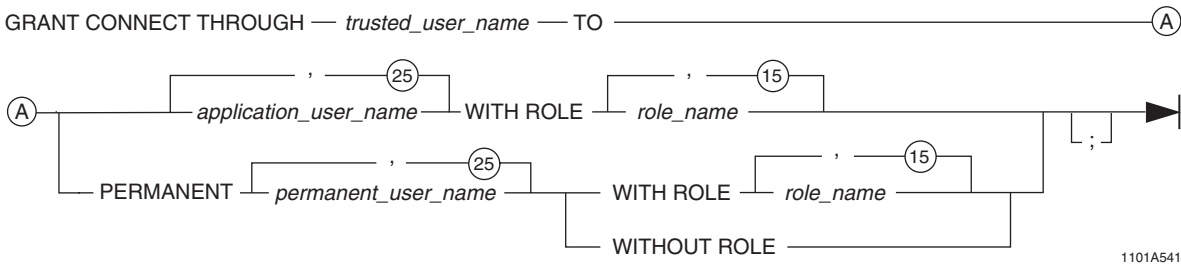
1101W055



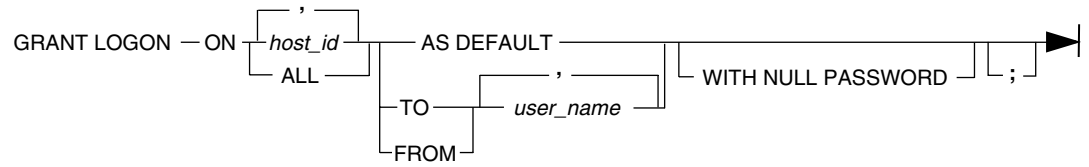
1101A535



GRANT CONNECT THROUGH



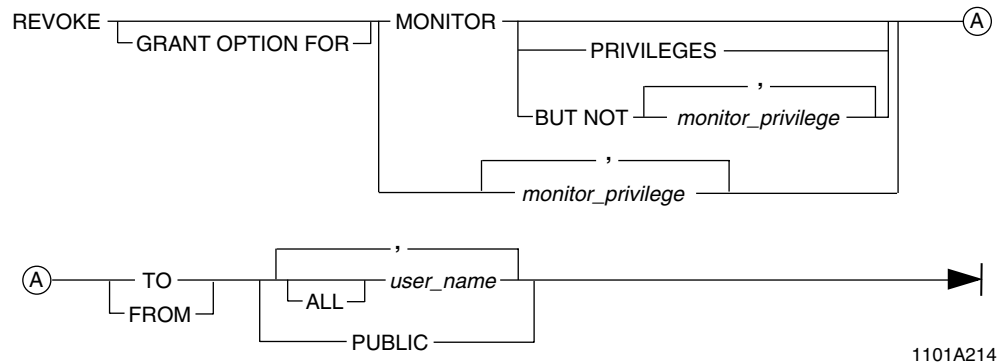
GRANT LOGON



1101C027

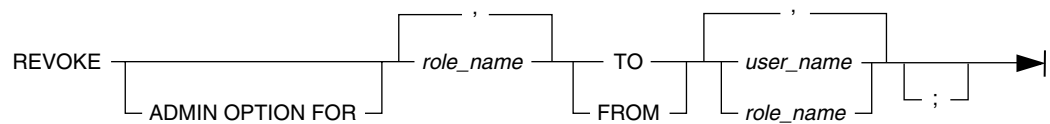
REVOKE

Monitor Form



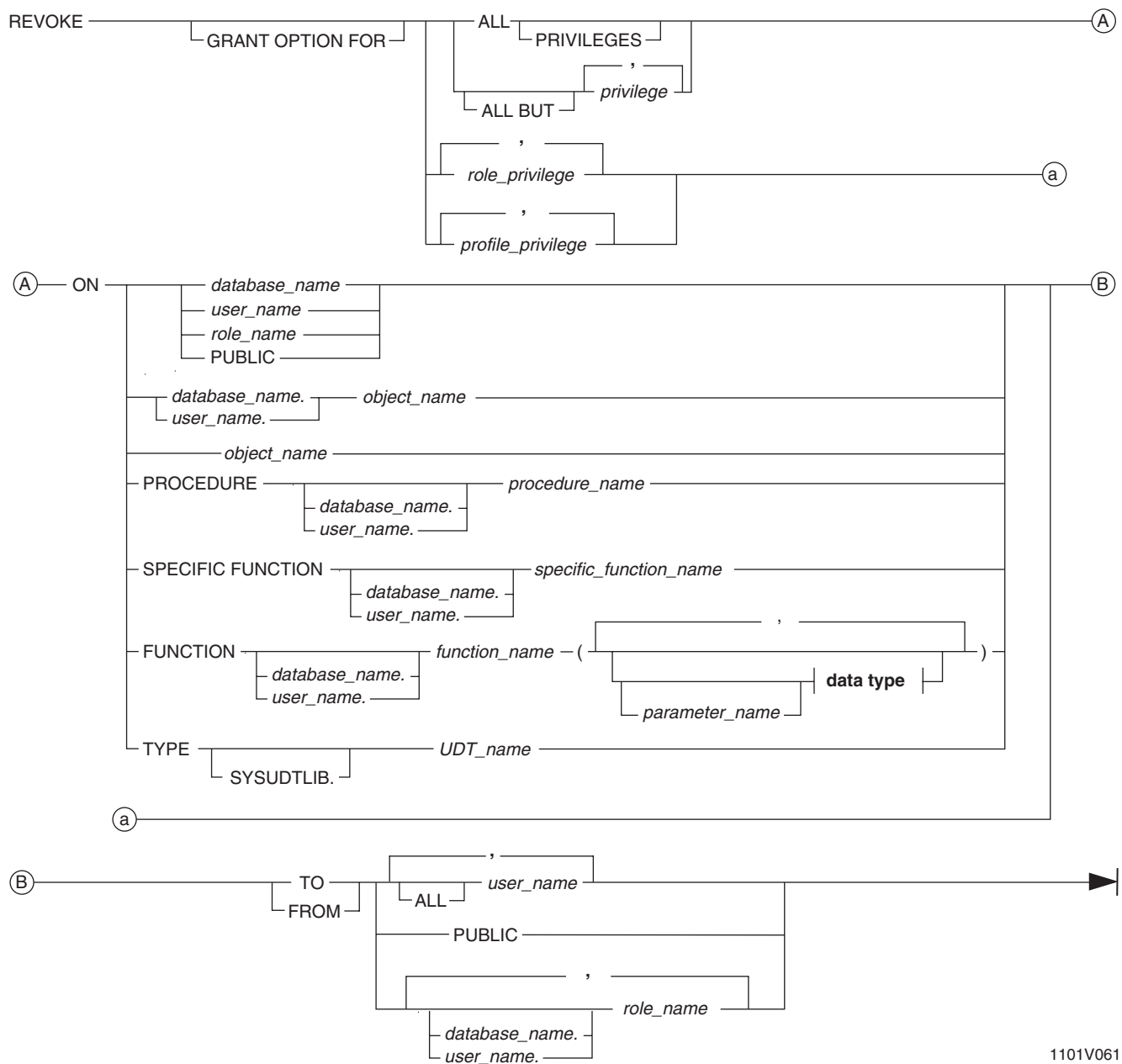
1101A214

Role Form

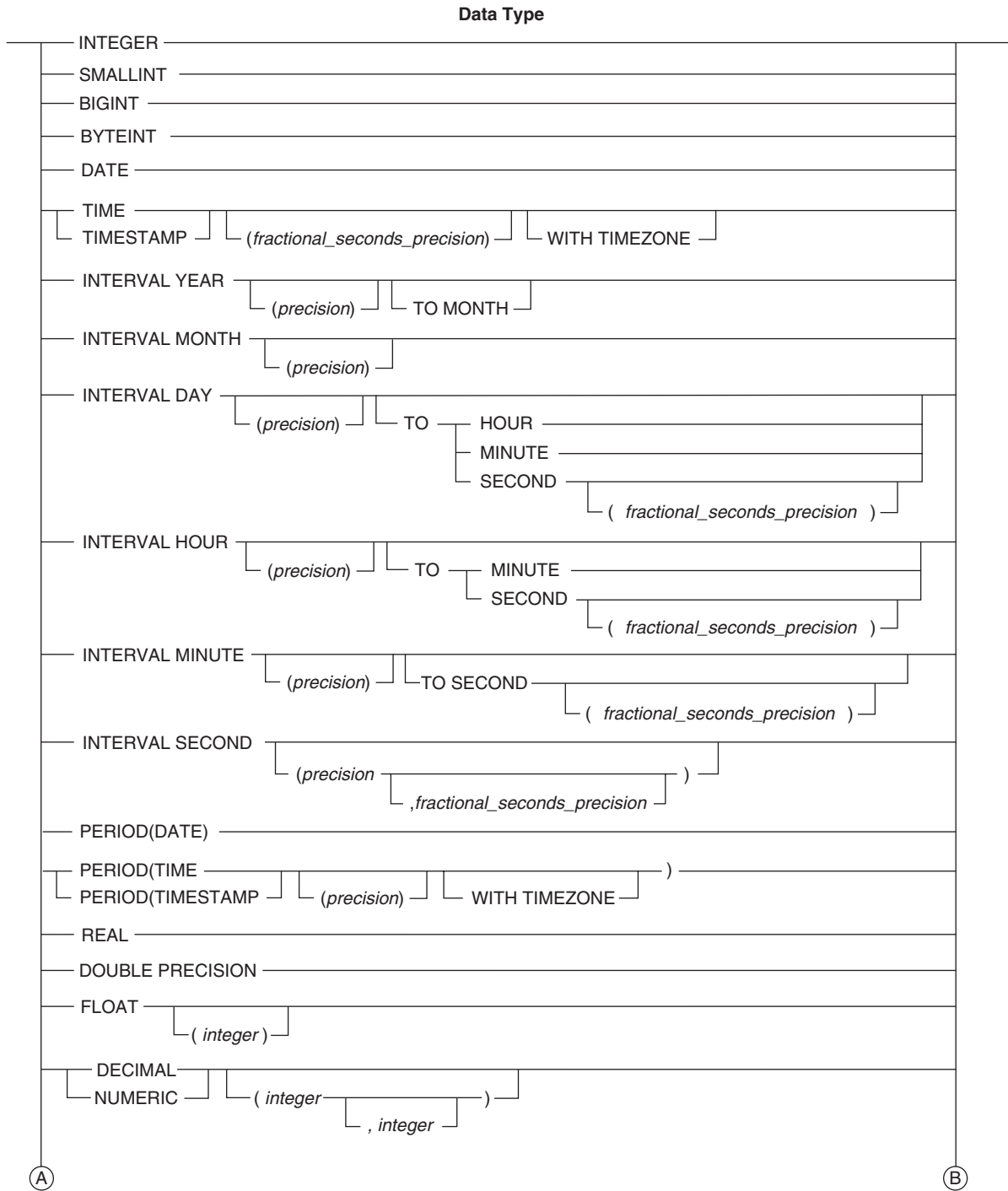


KZ01a009

SQL Form



1101V061



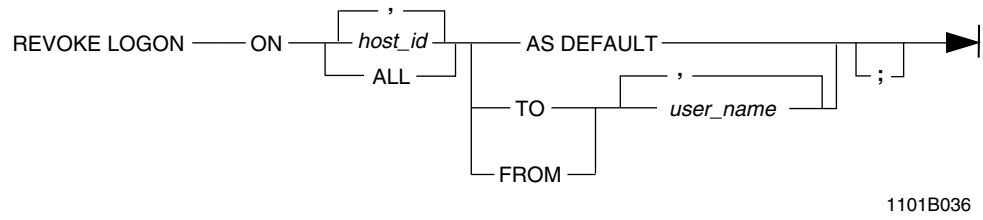
1101A535



REVOKE CONNECT THROUGH

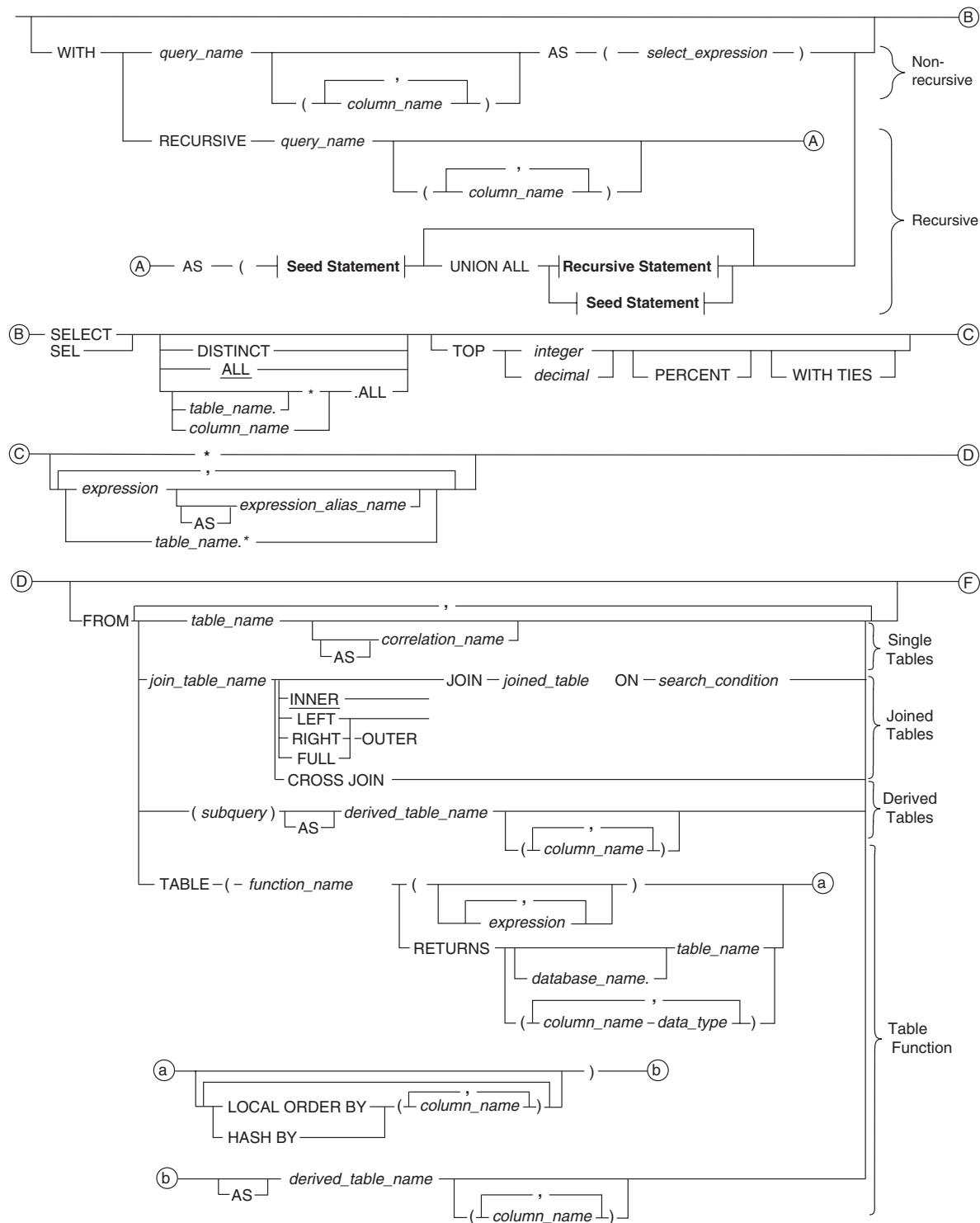


REVOKE LOGON

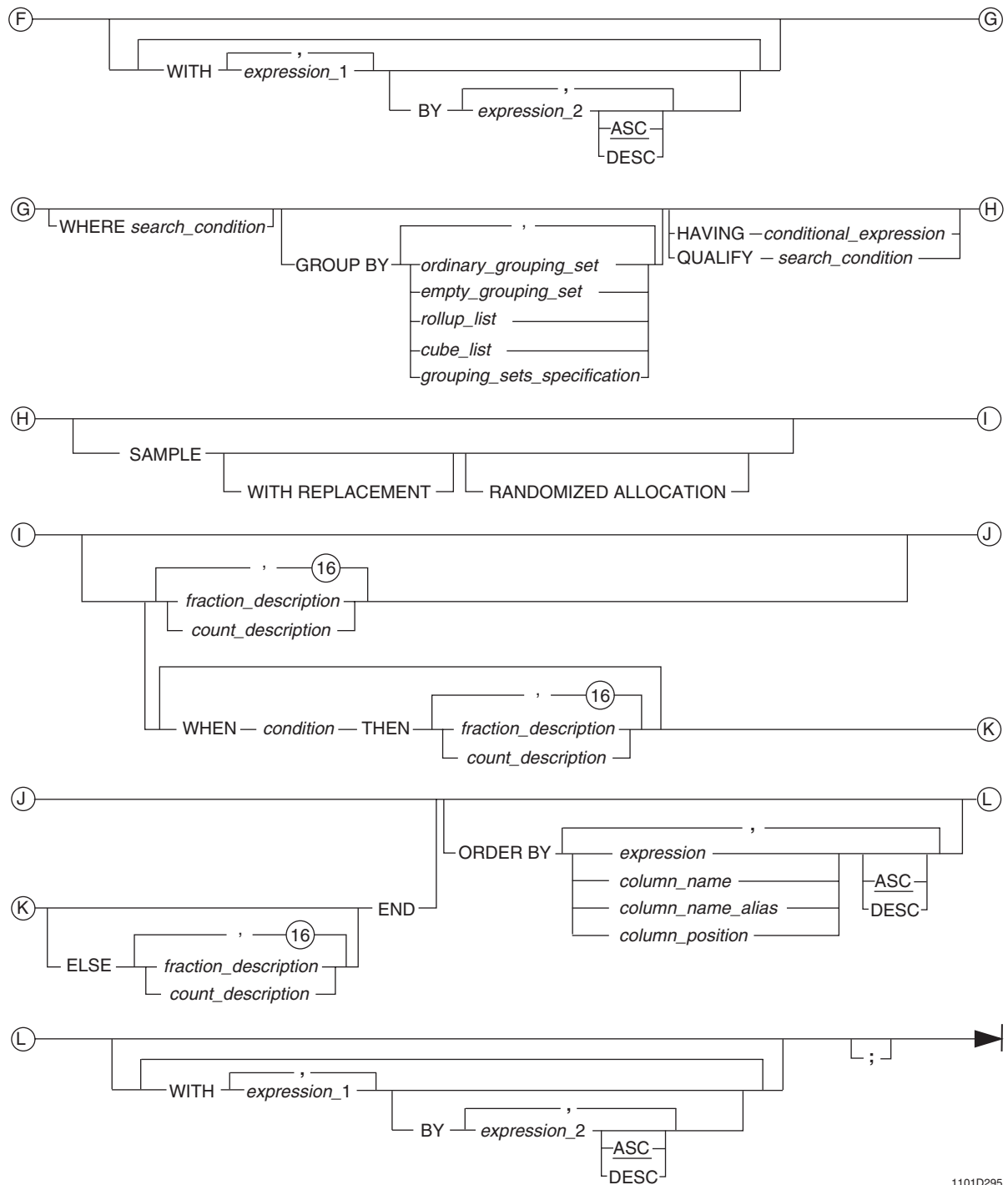


CHAPTER 5 **SQL Data Manipulation Language**

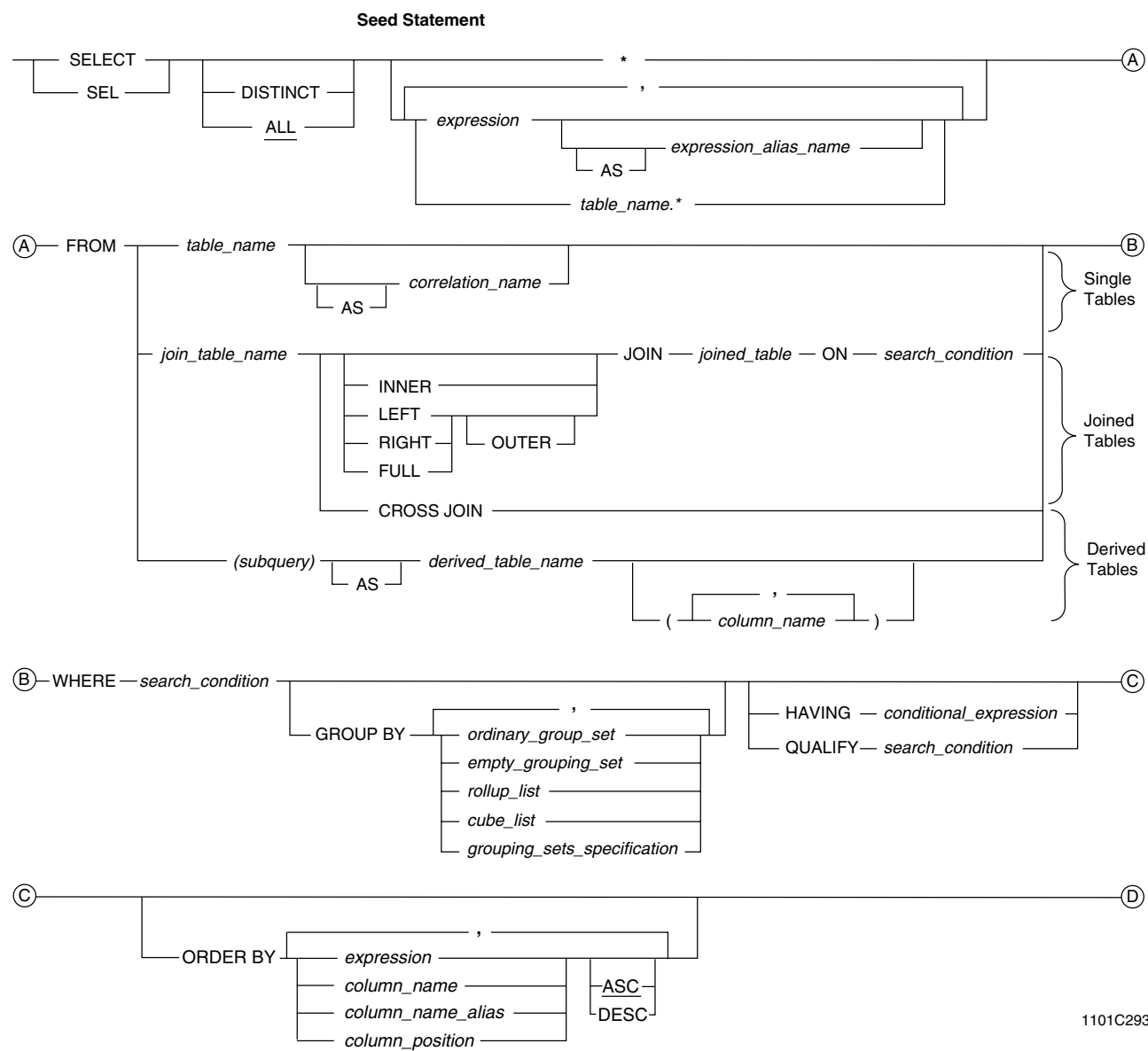
SELECT

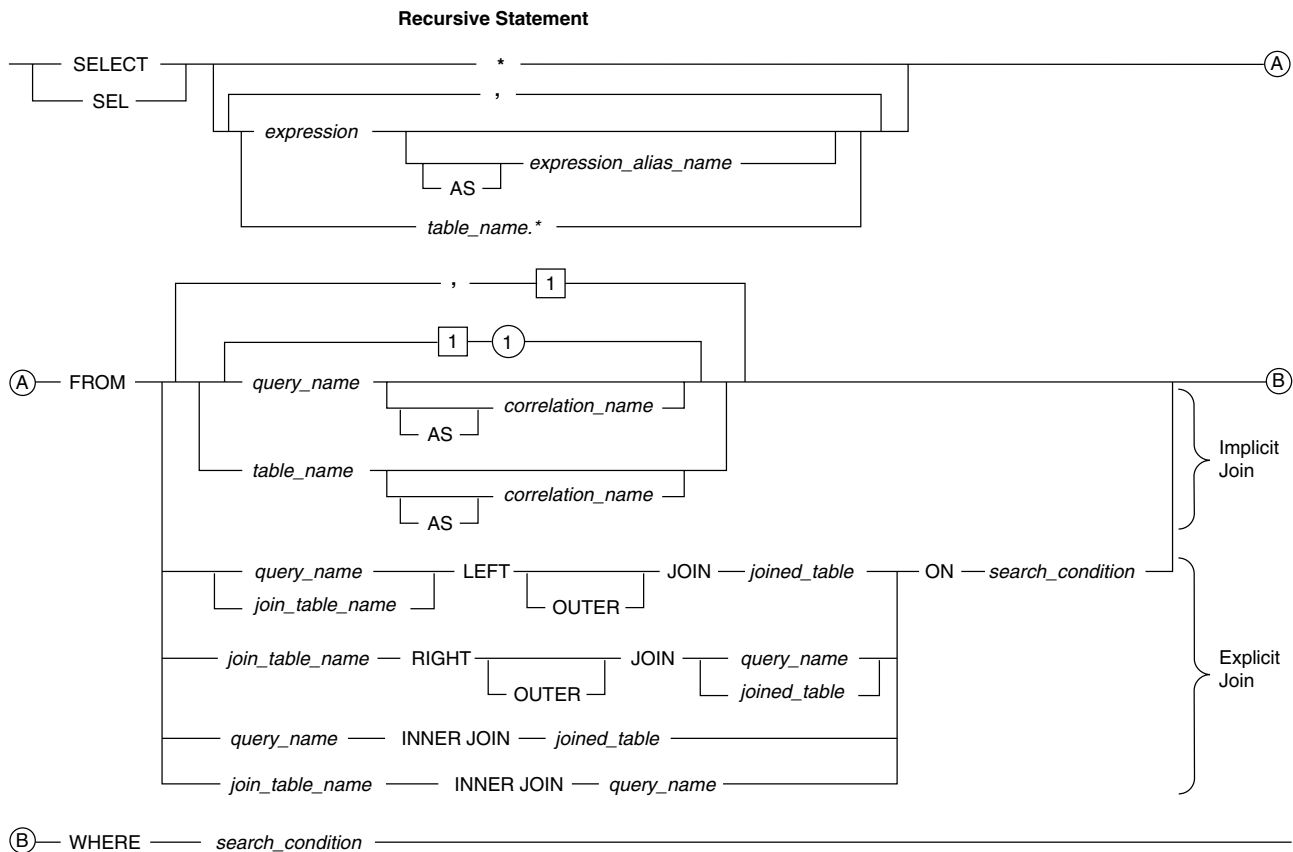


1101B545



1101D295





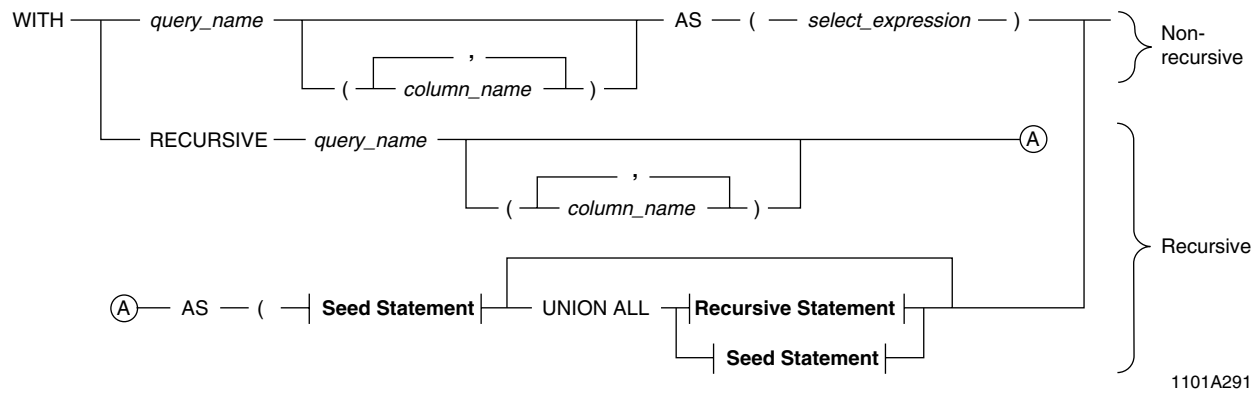
1101B292

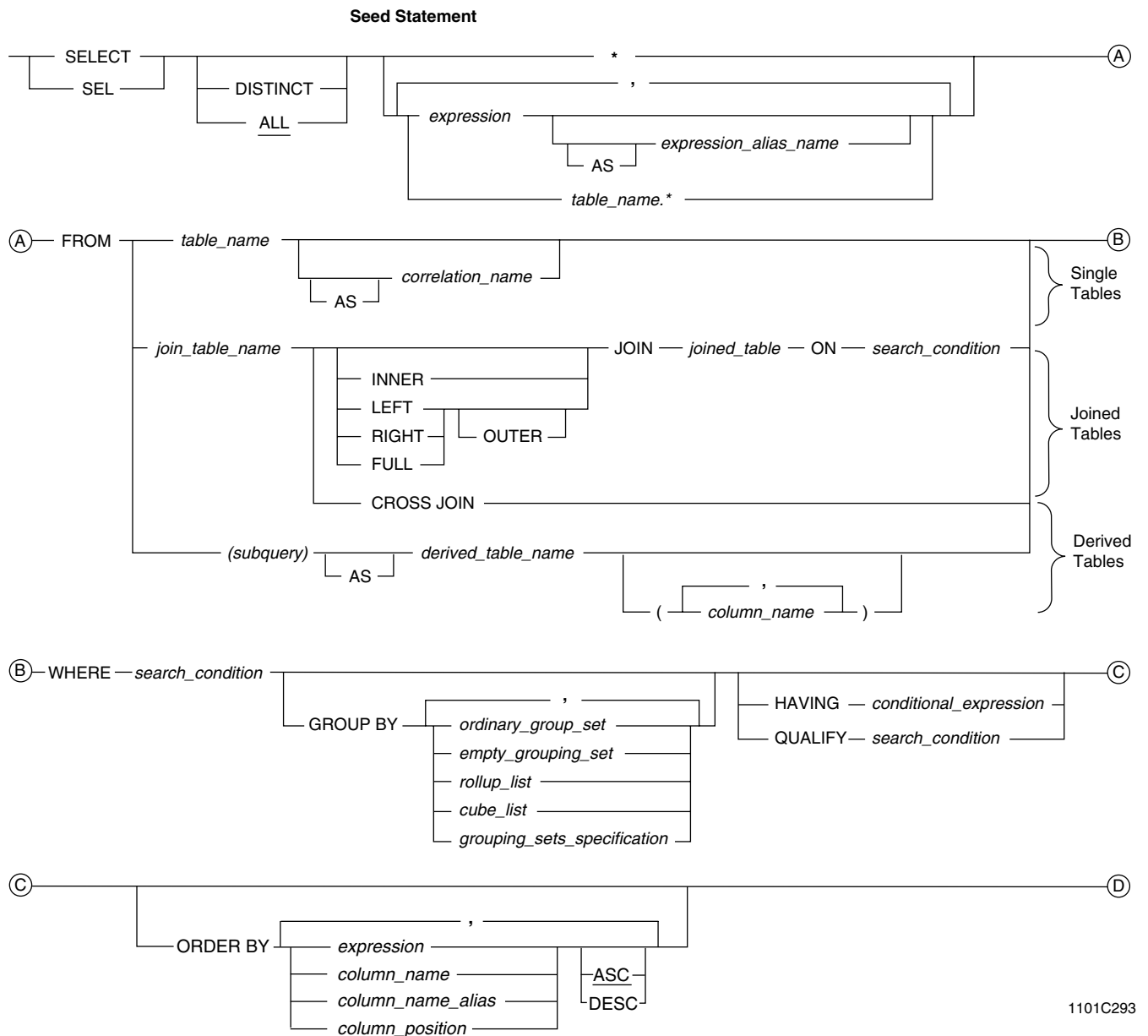
SELECT AND CONSUME

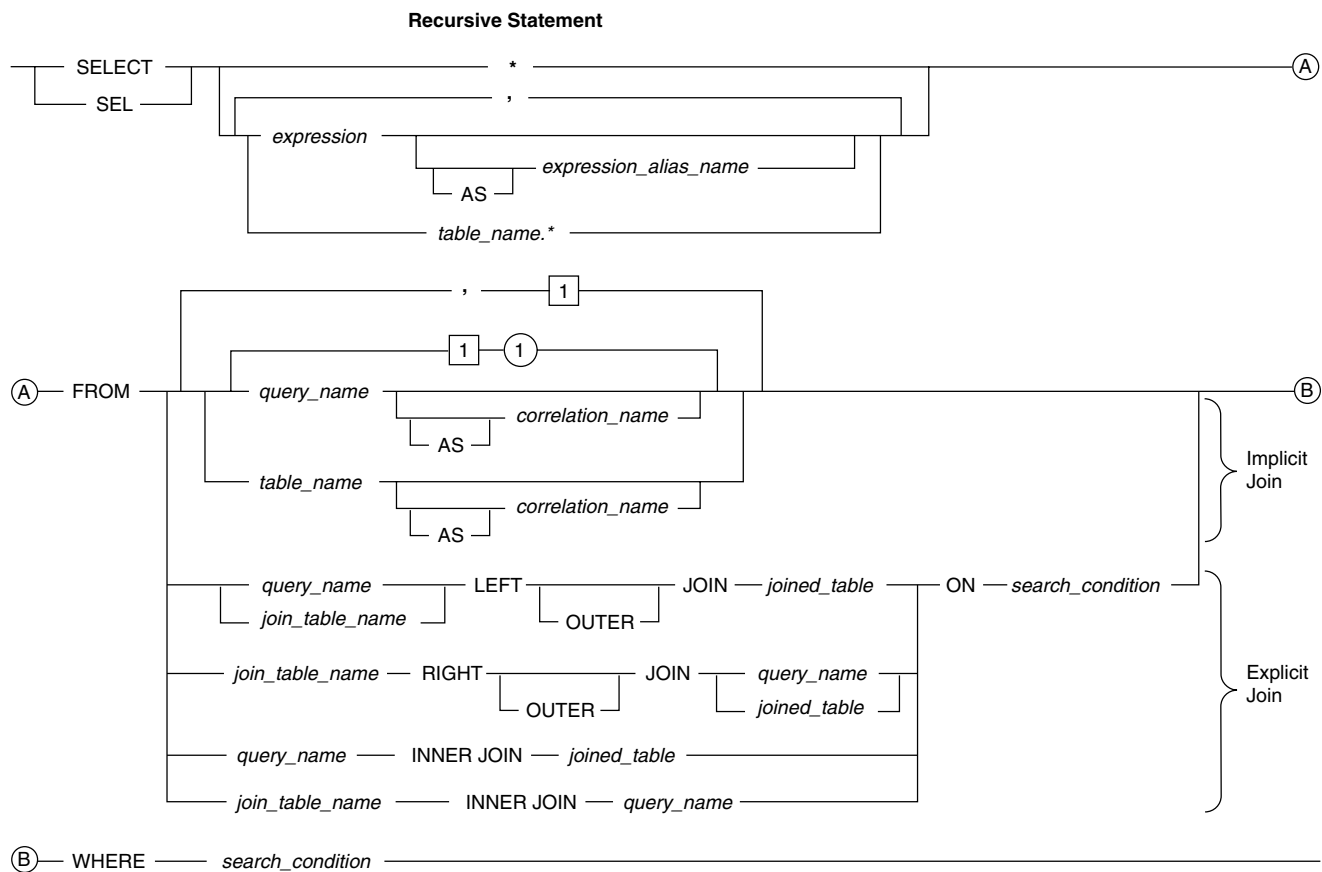
SELECT AND CONSUME TOP 1 — *select_list* — FROM — *queue_table_name* — ;

1101A220

WITH [RECURSIVE] Request Modifier

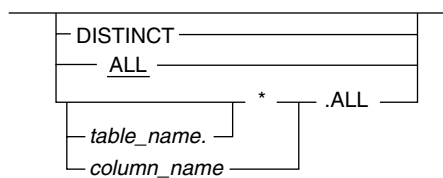






1101B292

DISTINCT, ALL, and .ALL Options



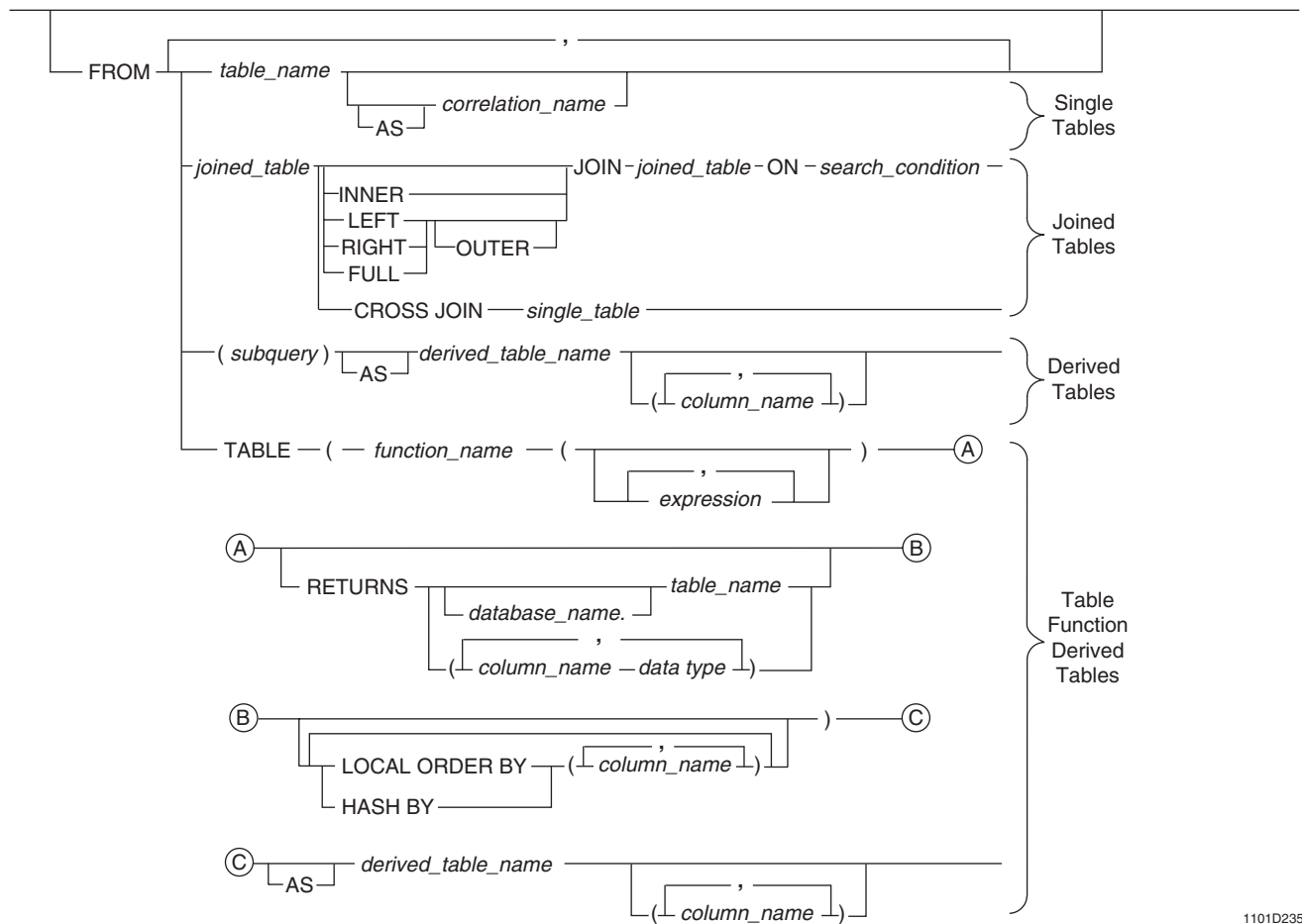
1101A357

TOP n Operator



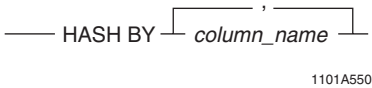
1101A231

FROM Clause

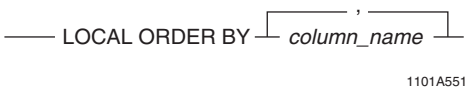


1101D235

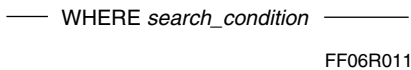
HASH BY Clause



LOCAL ORDER BY Clause

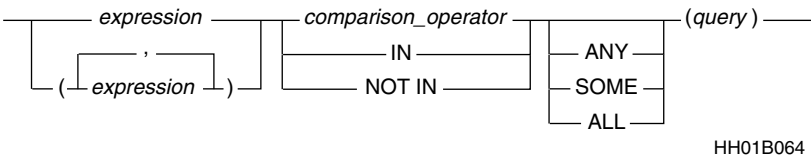


WHERE Clause

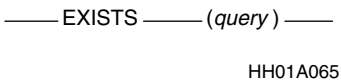


Subqueries in Search Conditions

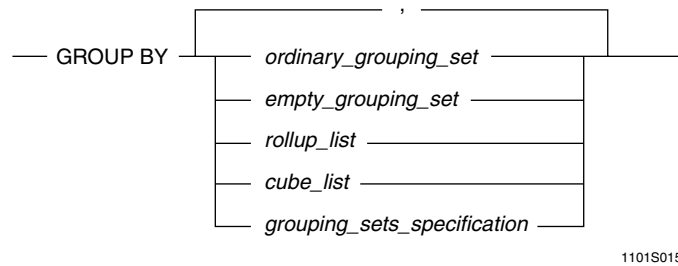
Syntax 1



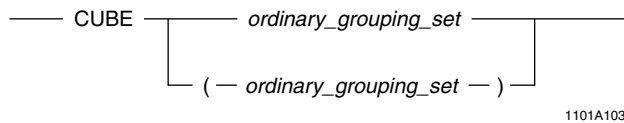
Syntax 2: Logical Expressions



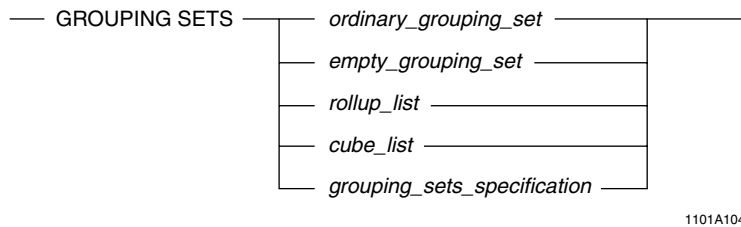
GROUP BY Clause



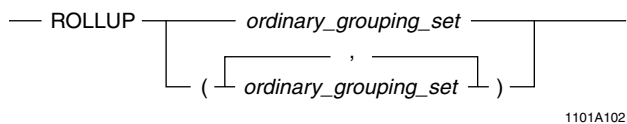
CUBE Option



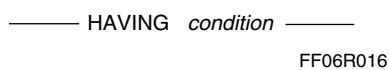
GROUPING SETS Option



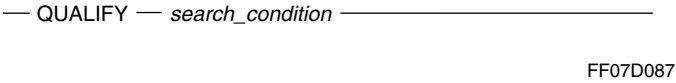
ROLLUP Option



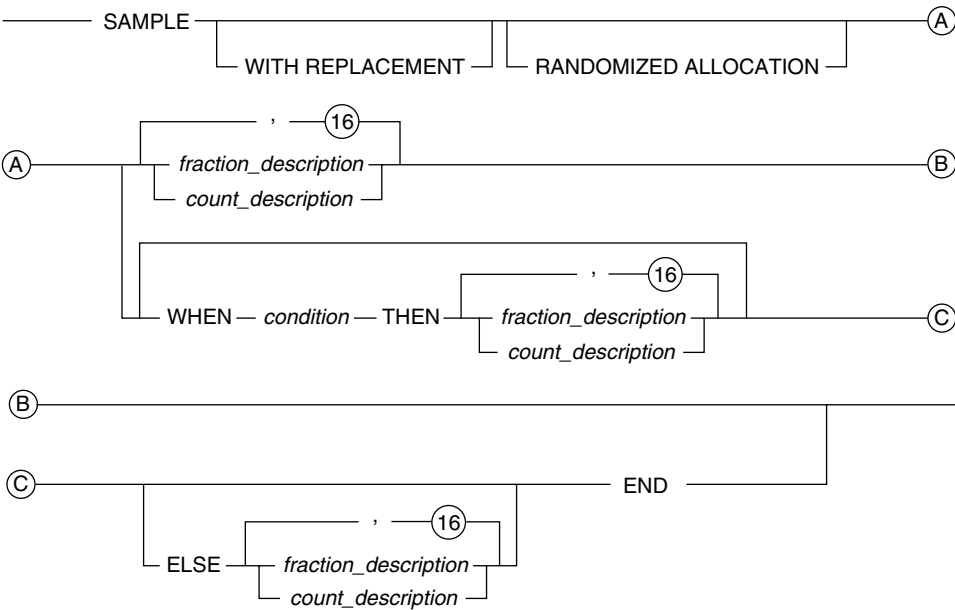
HAVING Clause



QUALIFY Clause

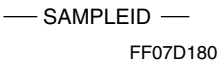


SAMPLE Clause

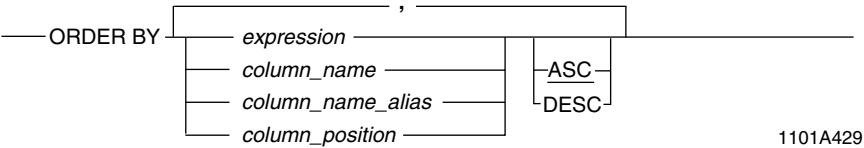


1101B065

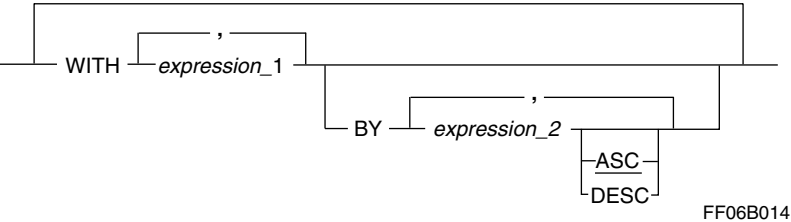
SAMPLEID Expression



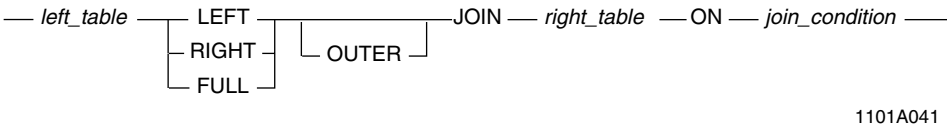
ORDER BY Clause



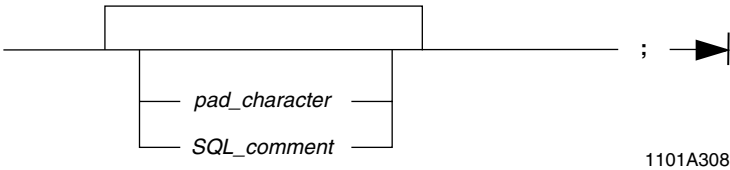
WITH Clause



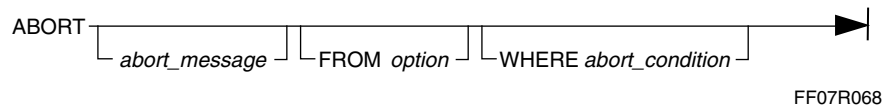
Outer Join



Null



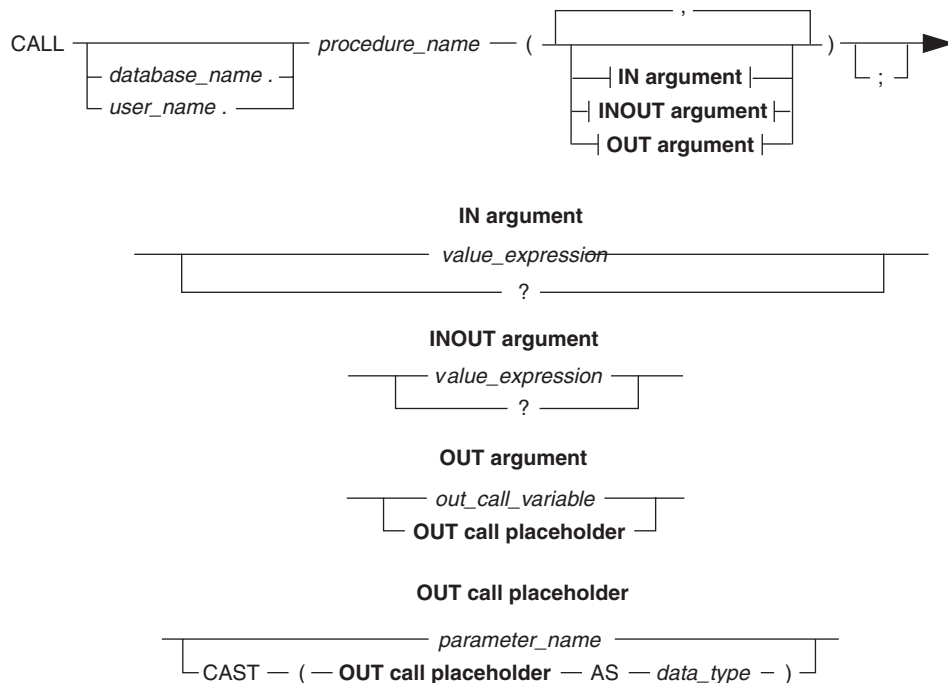
ABORT



BEGIN TRANSACTION



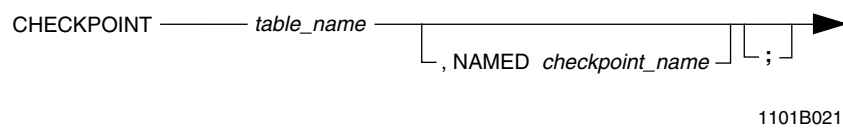
CALL



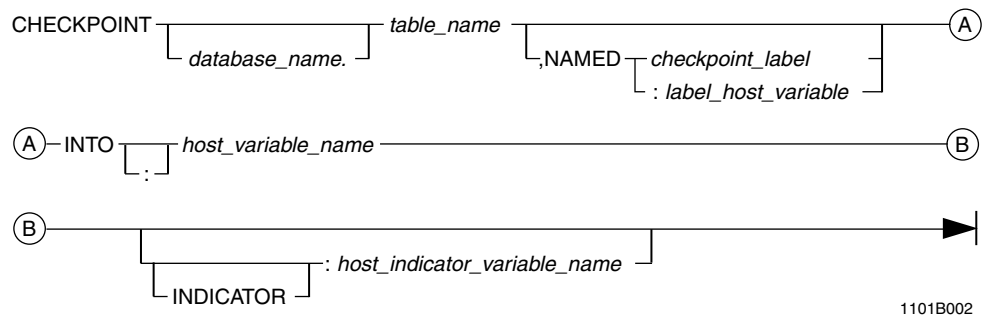
1101B042

CHECKPOINT

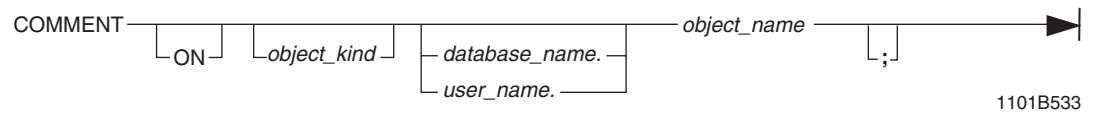
Interactive Syntax



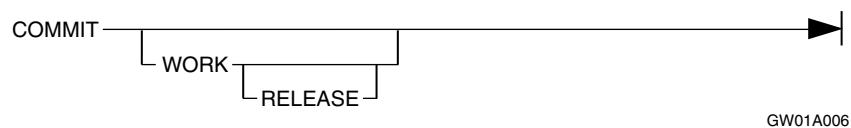
Embedded SQL and Stored Procedure Syntax



COMMENT (Comment-Retrieving Form)

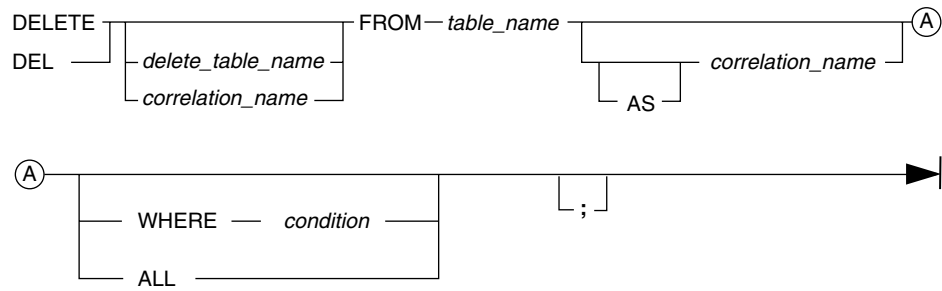


COMMIT



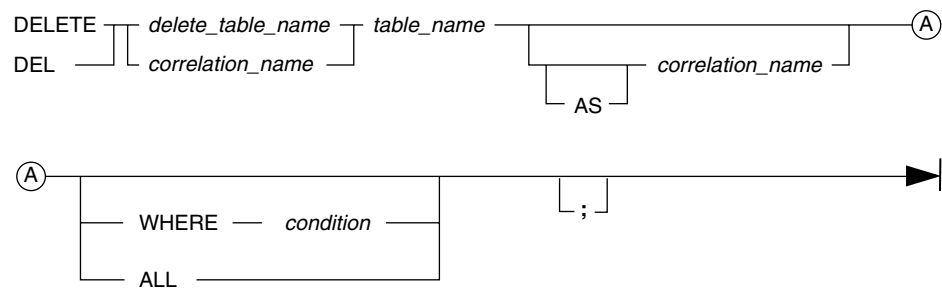
DELETE

Basic/Searched Form

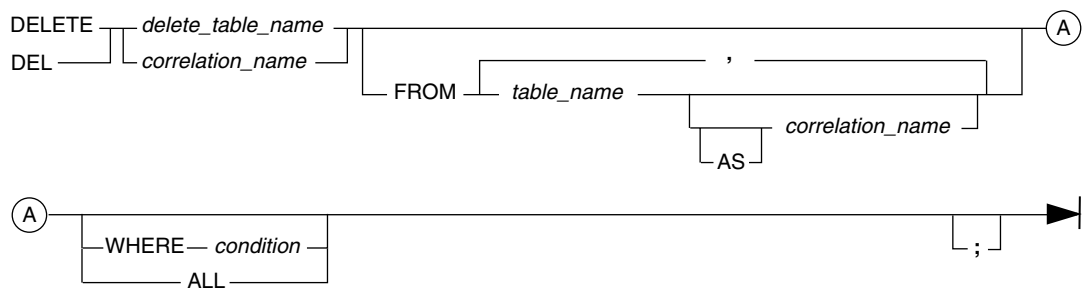


1101C074

Join Condition Form

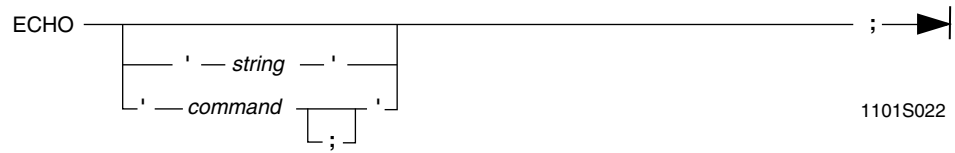


1101C073



1101S079

ECHO

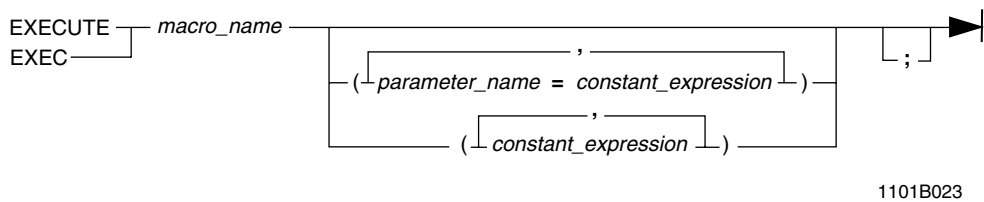


END TRANSACTION

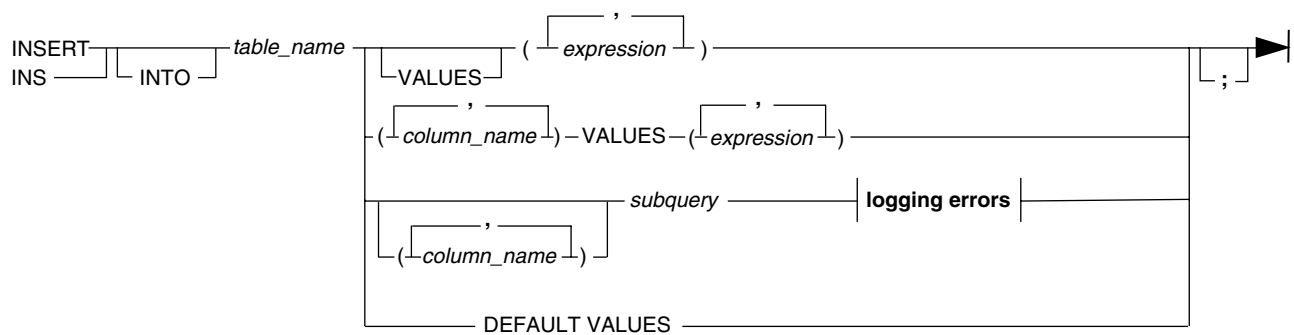


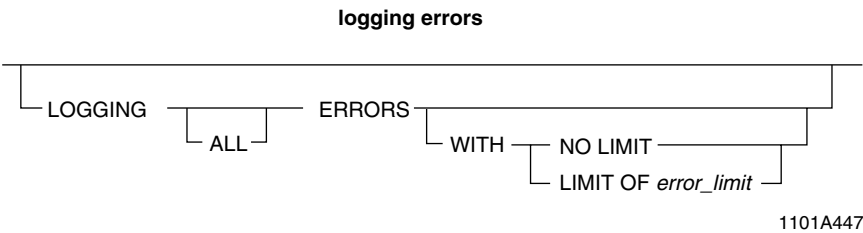
EXECUTE

Macro Form

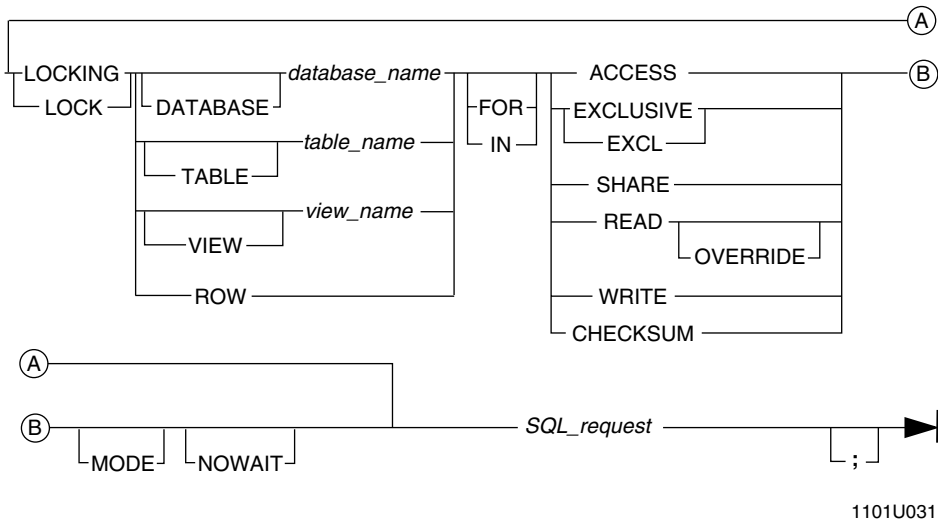


INSERT/INSERT . . . SELECT

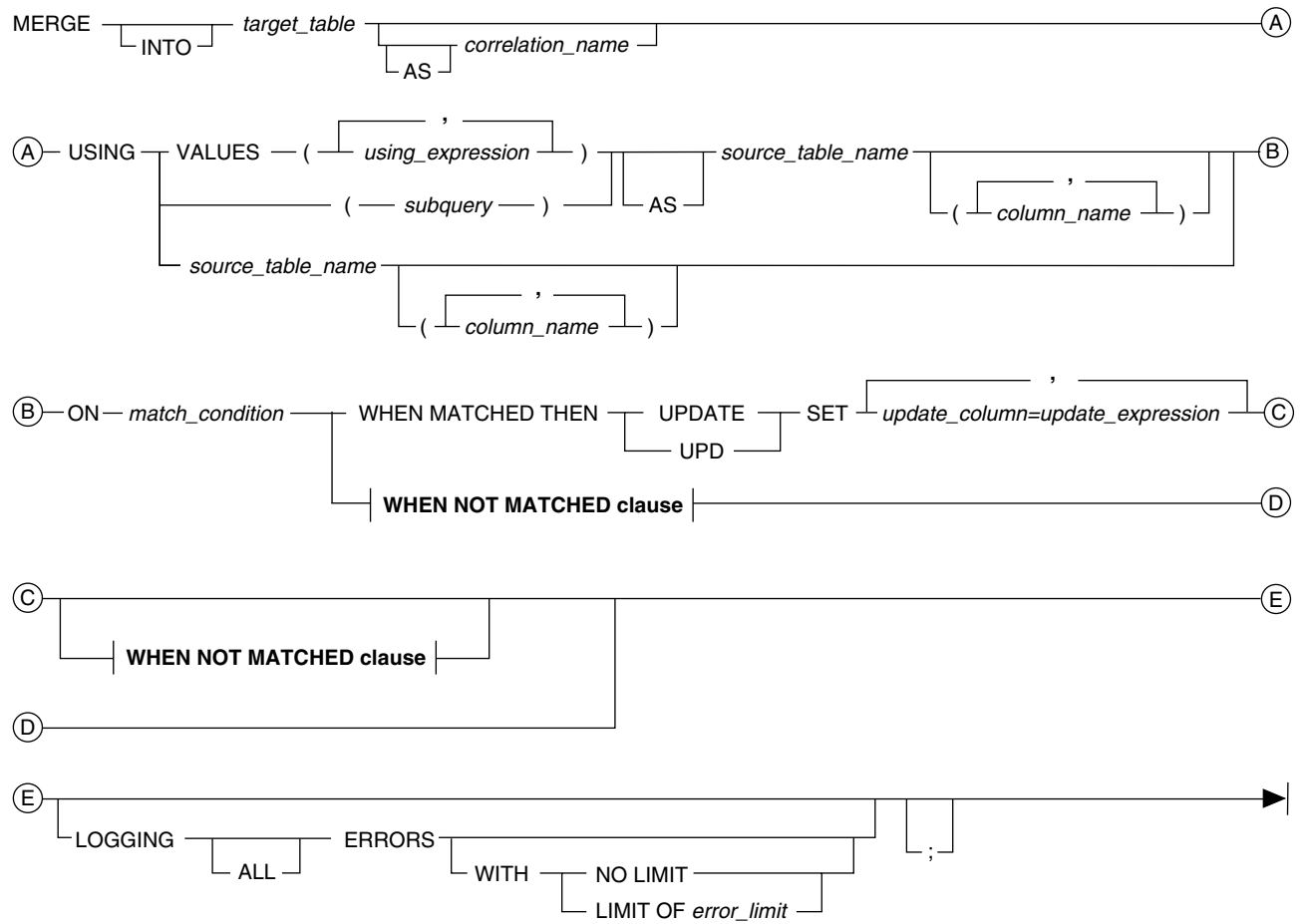




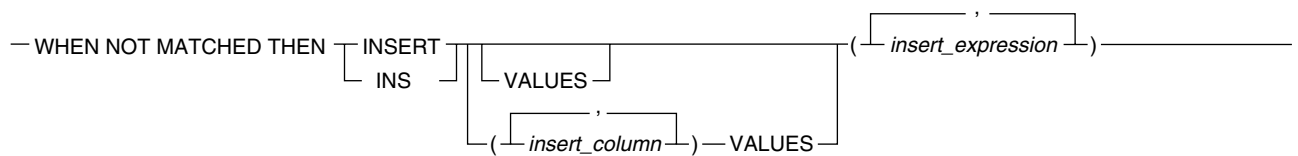
LOCKING Request Modifier



MERGE

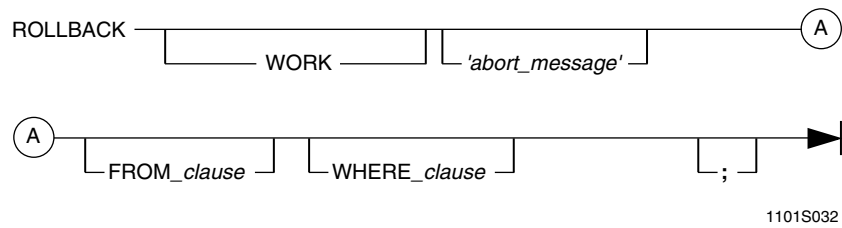


WHEN NOT MATCHED clause



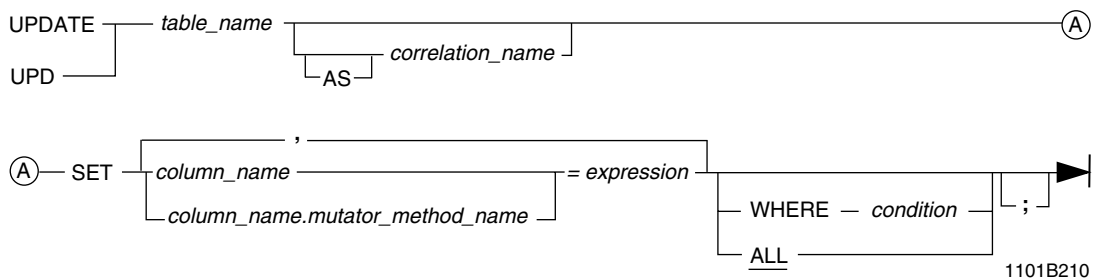
1101B445

ROLLBACK

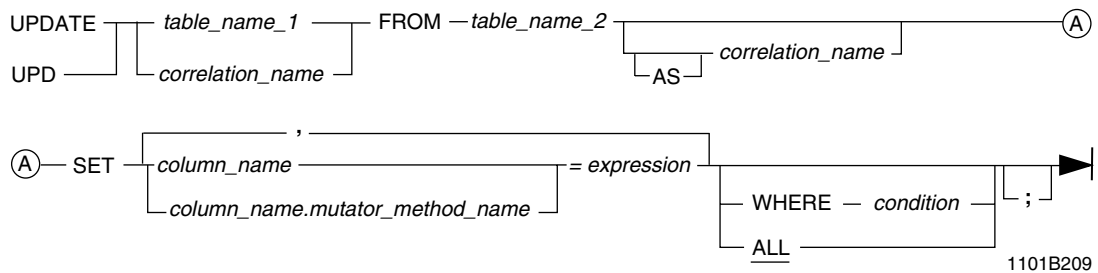


UPDATE

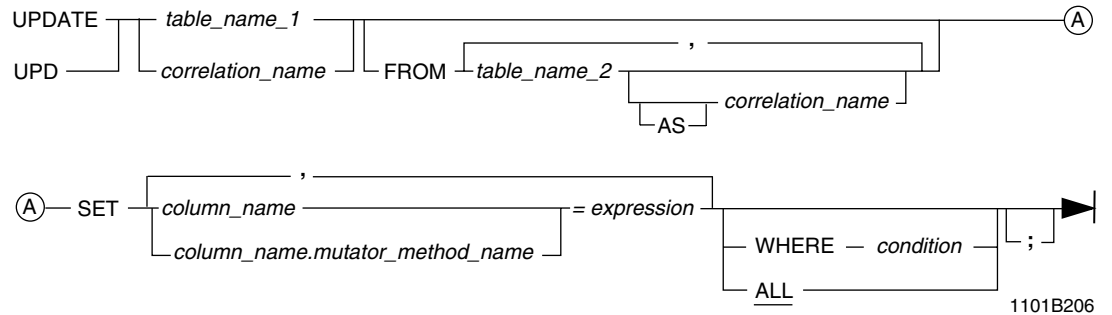
Basic Form, No FROM Clause Syntax



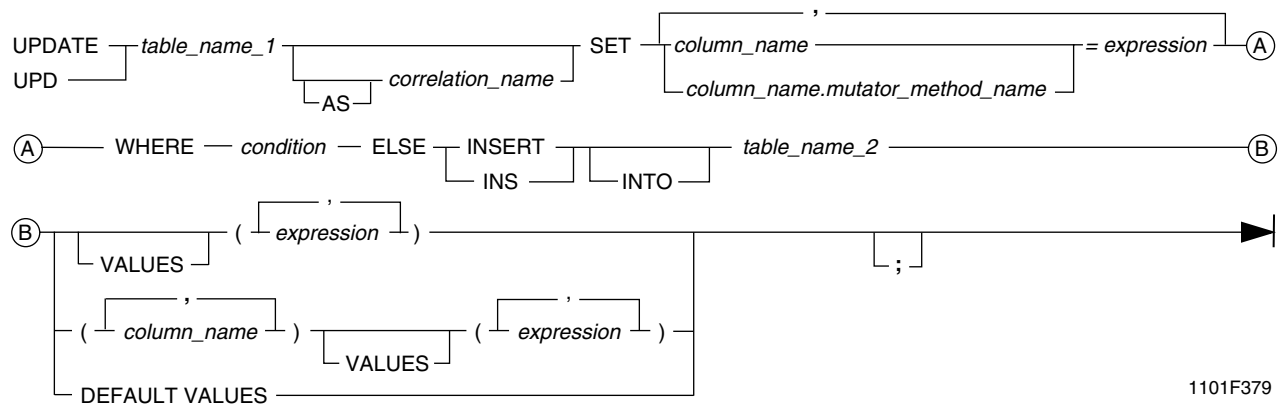
Basic Form, FROM Clause Syntax



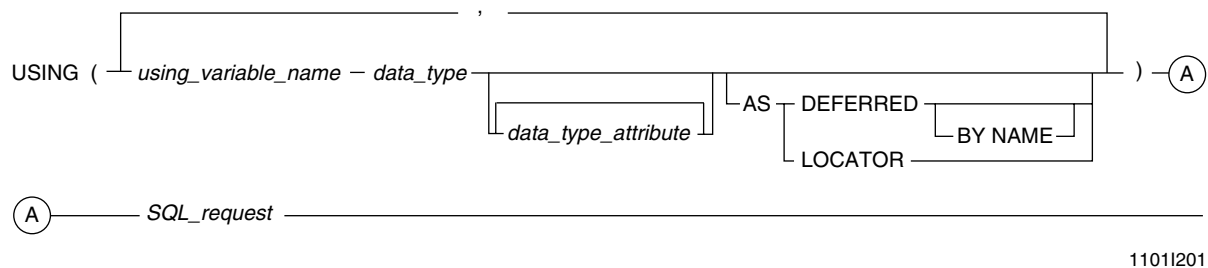
Joined Tables Syntax



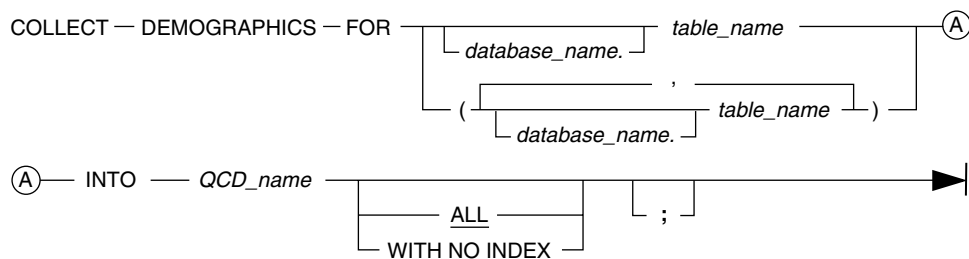
Upsert Form



USING Request Modifier

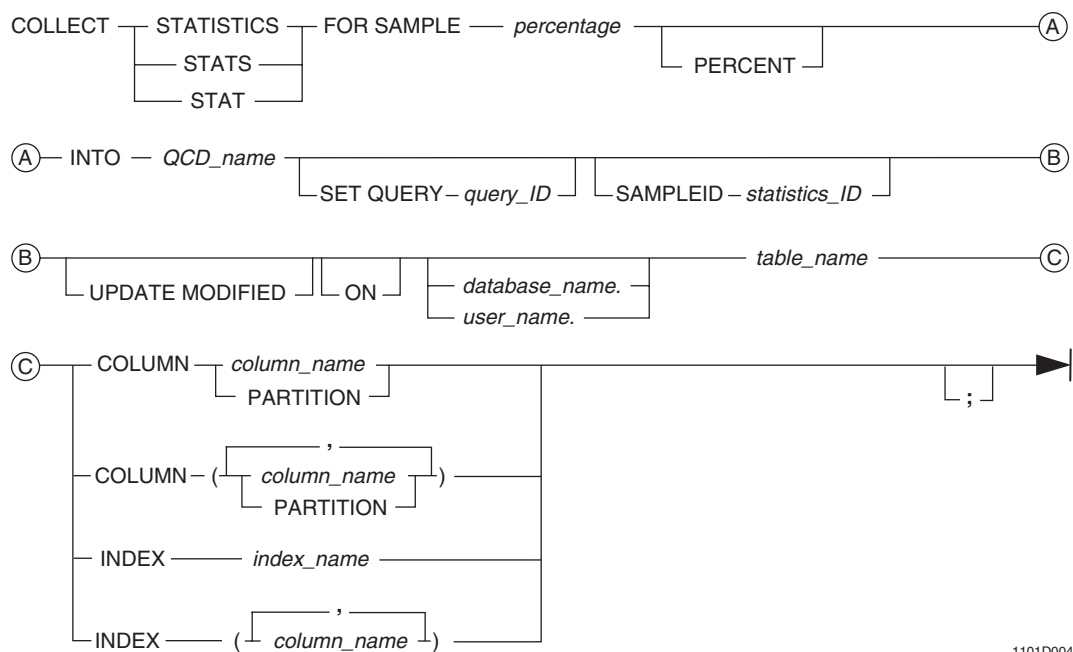


COLLECT DEMOGRAPHICS



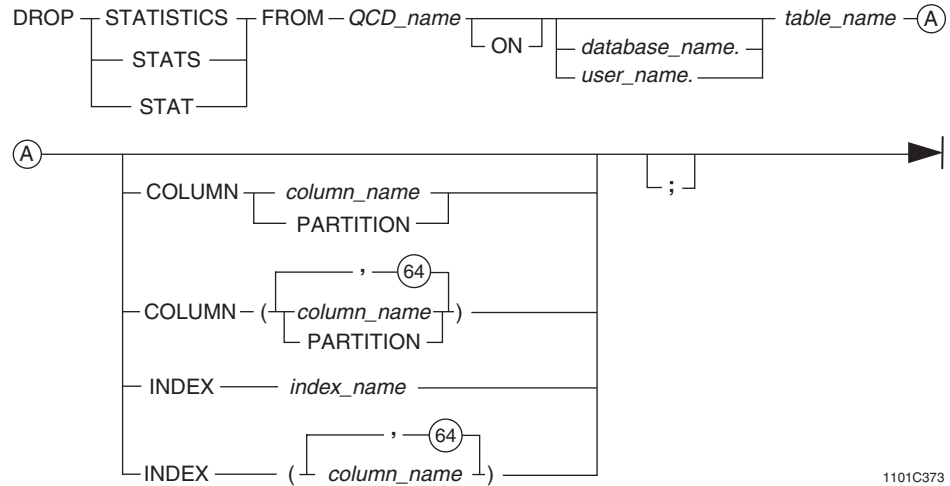
1101A472

COLLECT STATISTICS (QCD Form)

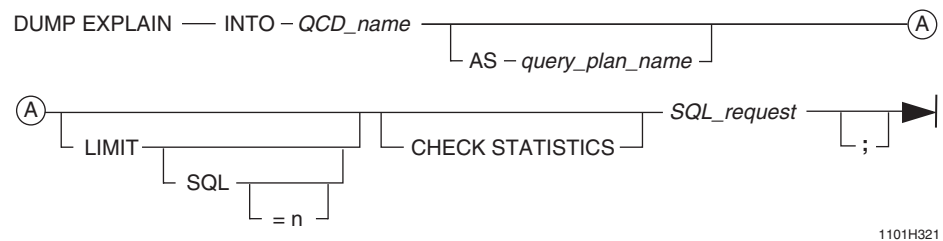


1101D004

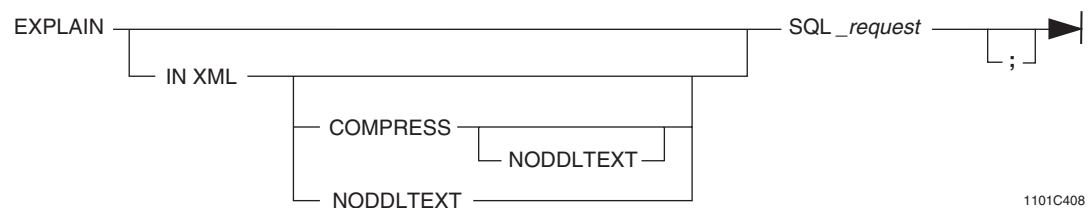
DROP STATISTICS (QCD Form)



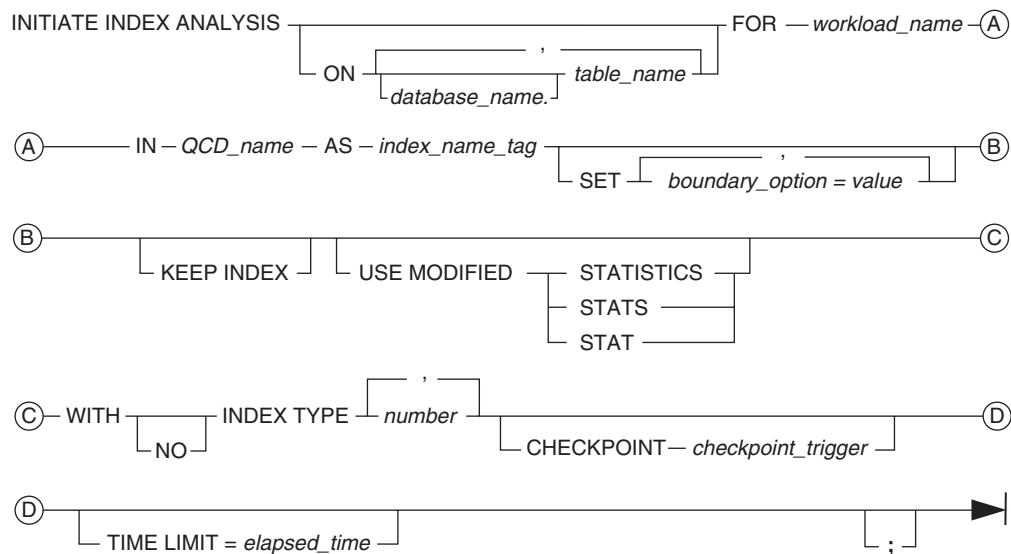
DUMP EXPLAIN



EXPLAIN Request Modifier

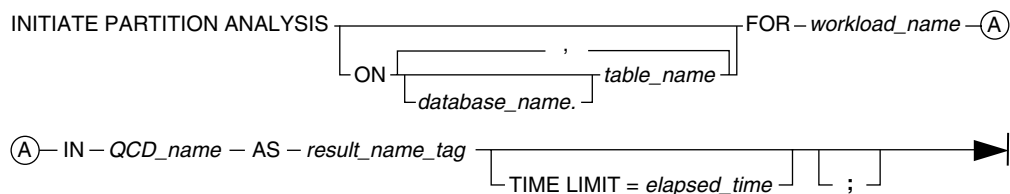


INITIATE INDEX ANALYSIS



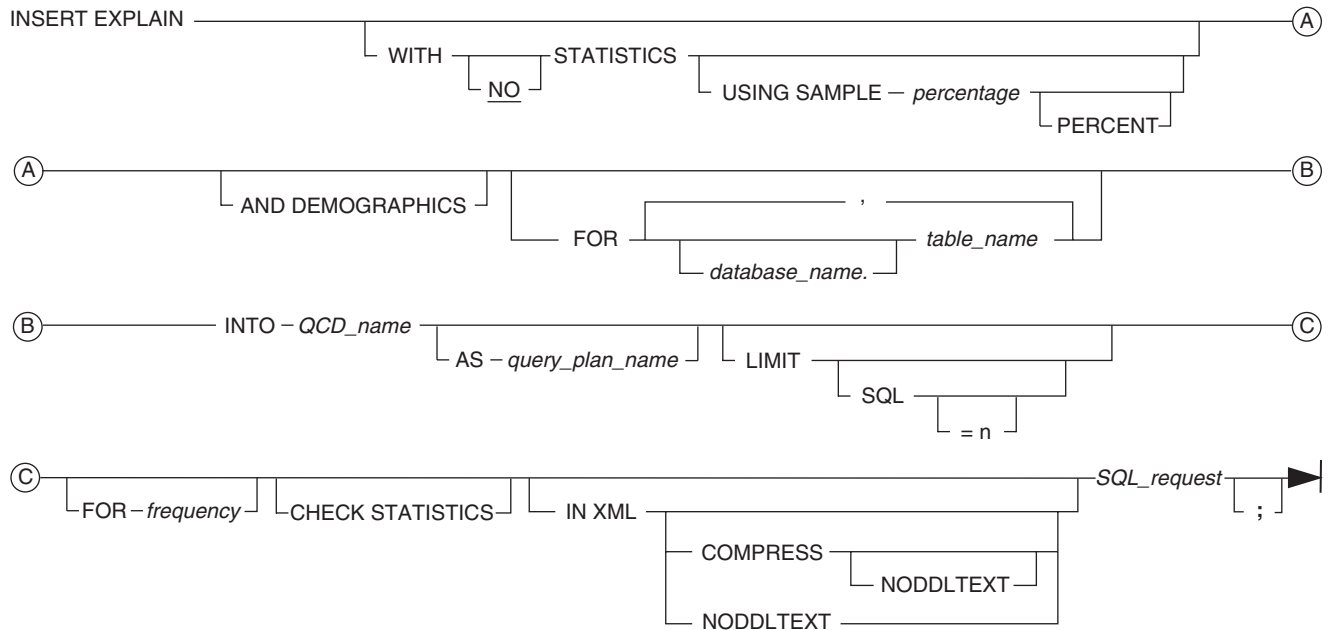
1101F013

INITIATE PARTITION ANALYSIS



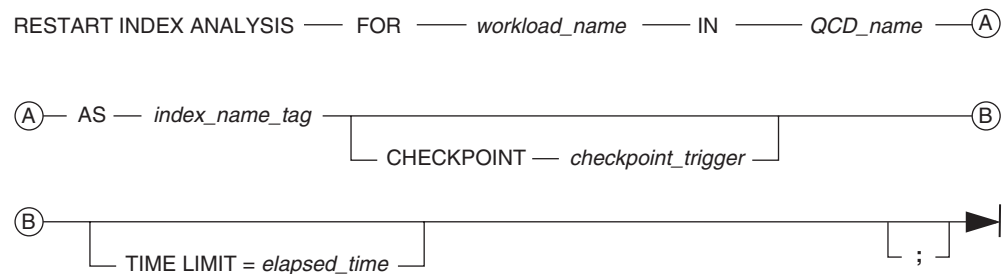
1101B443

INSERT EXPLAIN



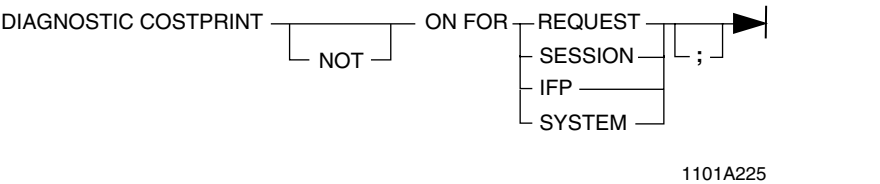
1101I322

RESTART INDEX ANALYSIS

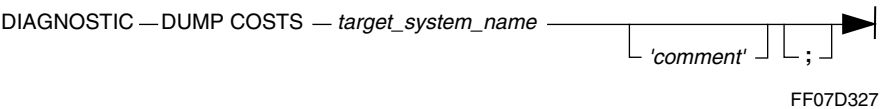


1101C111

DIAGNOSTIC COSTPRINT



DIAGNOSTIC DUMP COSTS

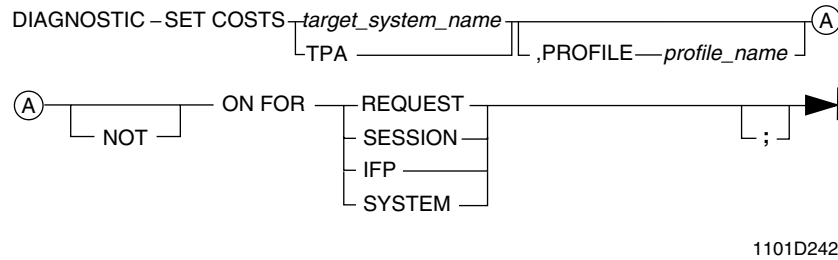


DIAGNOSTIC HELP COSTS

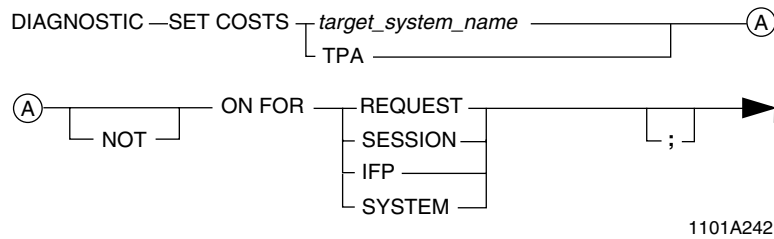


DIAGNOSTIC SET COSTS

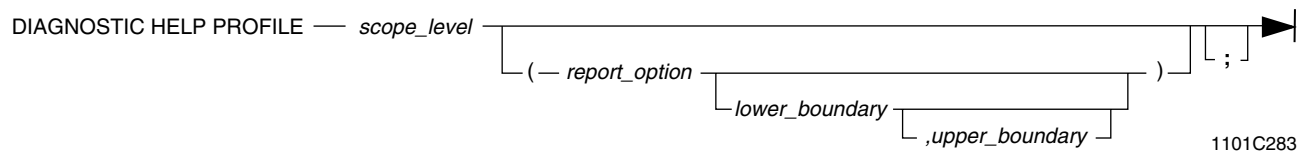
Syntax (Full)



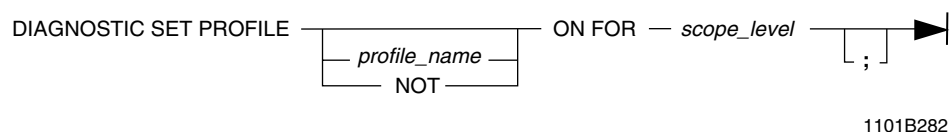
Syntax (Restricted)



DIAGNOSTIC HELP PROFILE



DIAGNOSTIC SET PROFILE



DIAGNOSTIC DUMP SAMPLES

DIAGNOSTIC — DUMP SAMPLES — *target_system_name* — (A)

(A) — TABLE — *database_name.* — *table_name* — ; — ►

GO01A001

DIAGNOSTIC HELP SAMPLES

DIAGNOSTIC — HELP SAMPLES — *target_system_name* — ; — ►

GO01A002

DIAGNOSTIC SET SAMPLES

General Syntax

DIAGNOSTIC — SET SAMPLES — *target_system_name* — [NOT] — ON FOR — (A)

(A) — [SESSION
SYSTEM] — TABLE — *database_name.* — *table_name* — ; — ►

GO01A004

Disable All Samples Syntax

DIAGNOSTIC SET ALL SAMPLES NOT ON FOR — [SESSION
SYSTEM] — ; — ►

GO01A003

DIAGNOSTIC "Validate Index"

DIAGNOSTIC — *"validate index"* [NOT] ON — FOR — SESSION [;] ►

TW01A002

CLOSE

CLOSE — *cursor_name* —————▶|

GW01A003 ³

DECLARE CURSOR

Dynamic SQL Form

DECLARE — *cursor_name* ————┐
 └ SCROLL ┘ — CURSOR FOR — *statement_name* —▶|

1101A307

Macro Form

DECLARE — *cursor_name* — CURSOR FOR EXEC ————┐
 └ *database_name.* ┘ (A)
(A) — *macroname* ————┐
 └ (*parameter_list*) ┘ —————▶|

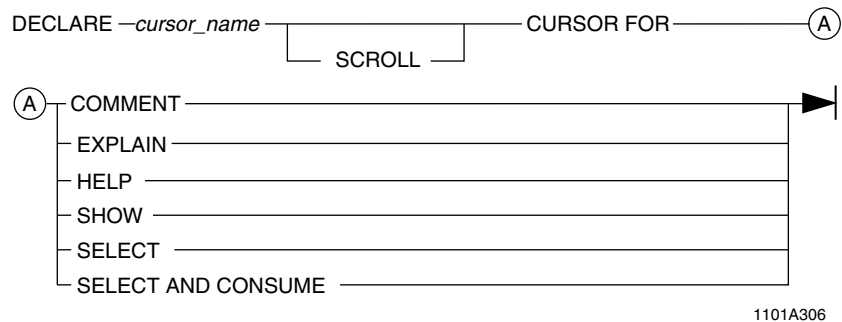
1101B011

Request Form

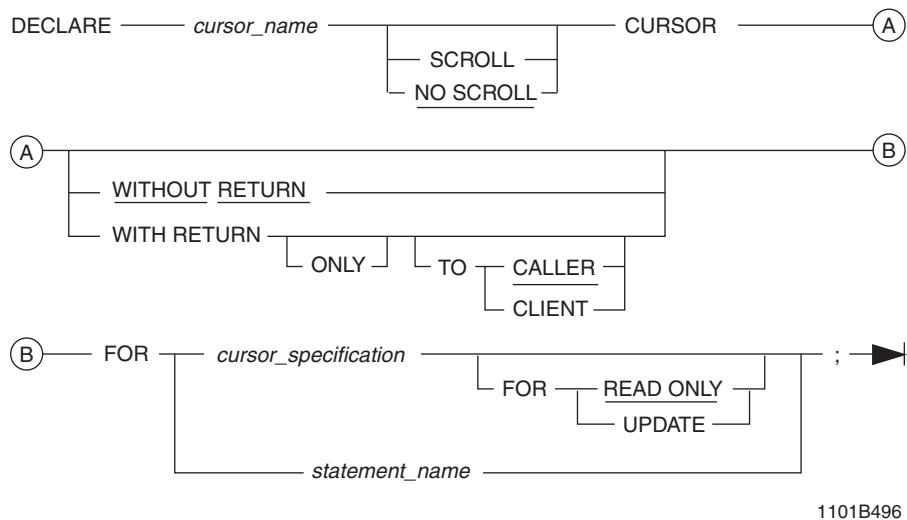
DECLARE — *cursor_name* ————— CURSOR FOR — *'request_specification'* —————▶|

1101B301

Selection Form

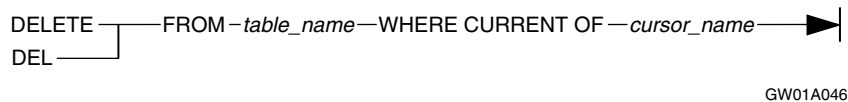


Stored Procedures Form



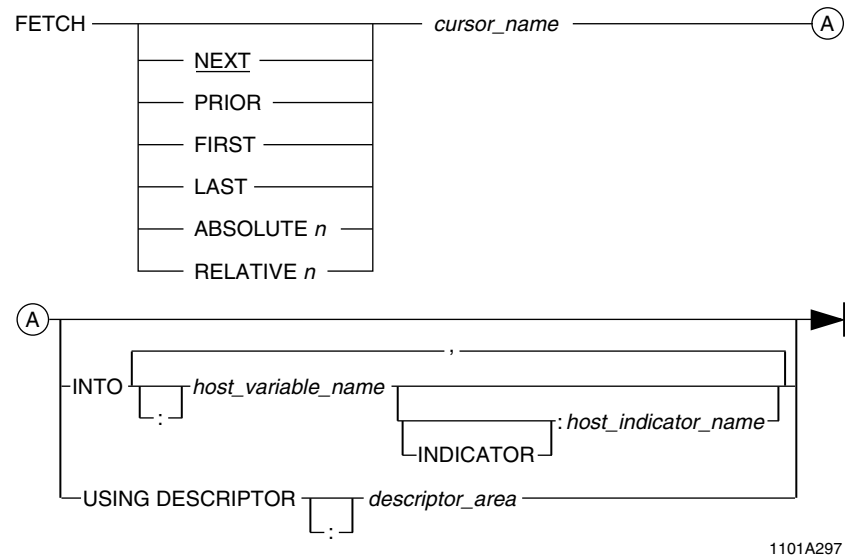
DELETE

Positioned Form

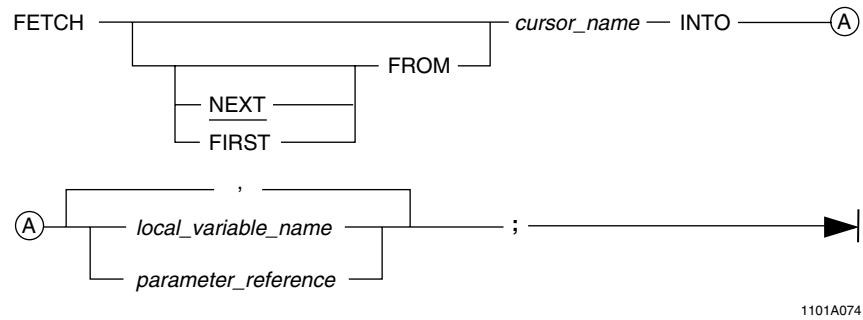


FETCH

Embedded SQL Form

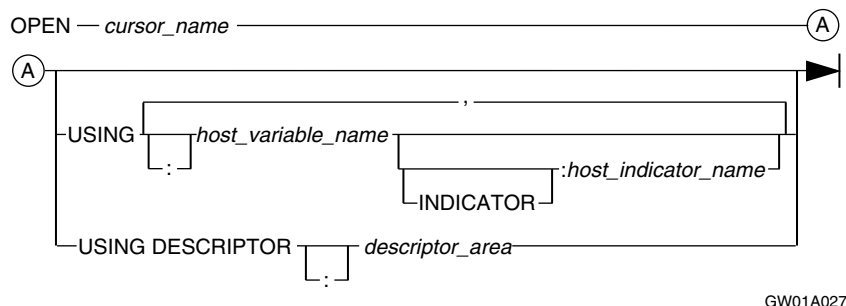


Stored Procedures Form

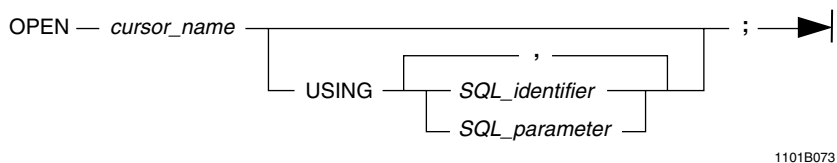


OPEN

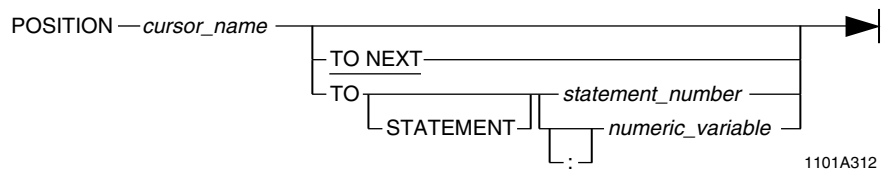
Embedded SQL Form



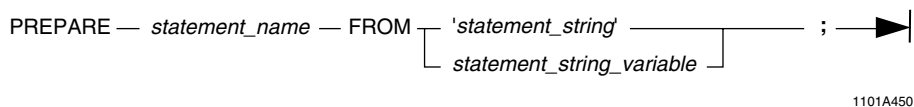
Stored Procedures Form



POSITION



PREPARE



REWIND

REWIND — *cursor_name* —————▶

GW01A030

SELECT ... INTO

Stored Procedures Only

[*with_[recursive]_modifier*] SELECT *select_list* INTO (A)
 SEL [ALL]
 [DISTINCT]
 (A) [*local_variable_name* , *parameter_name*] [*from_clause*] [*where_clause*] —————▶

1101B296

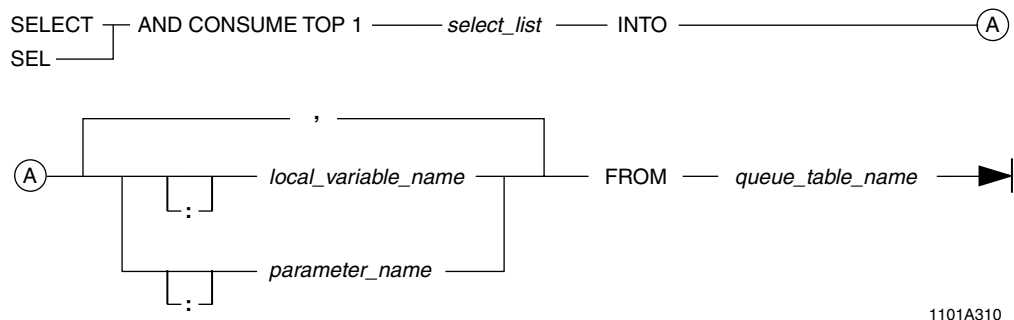
Embedded SQL Only

[*with_[recursive]_modifier*] SELECT *select_list* INTO (A)
 SEL
 (A) [*host_variable_name* , *host_variable_name*] (B)
 [:] [*host_indicator_name*]
 [INDICATOR]
 (B) [*from_clause*] [*where_clause*] —————▶

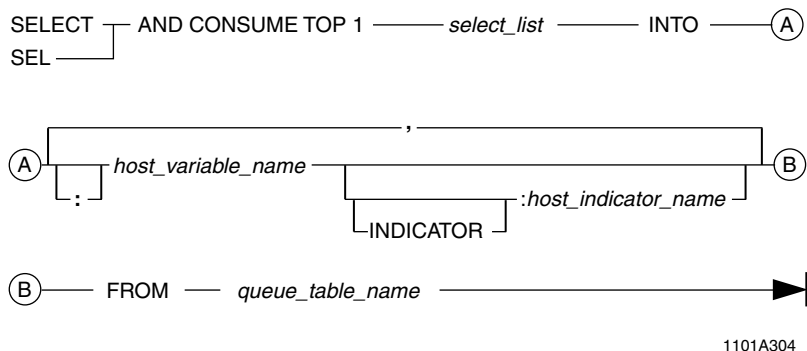
1101B305

SELECT AND CONSUME ... INTO

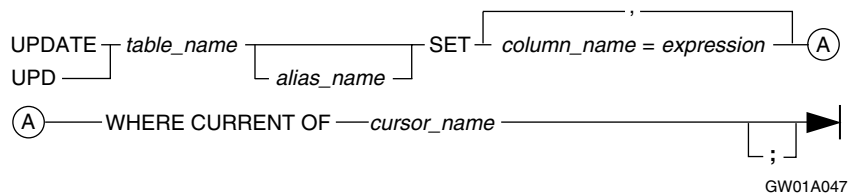
Stored Procedures Only



Embedded SQL Only

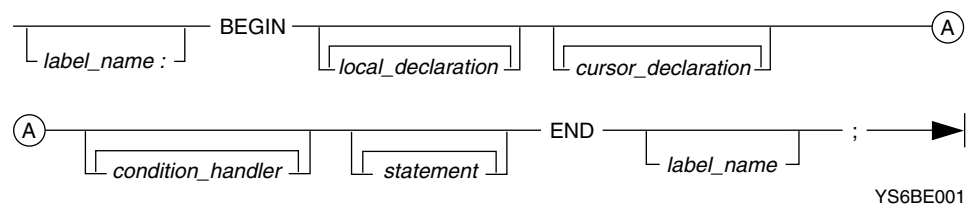


UPDATE (Positioned Form)



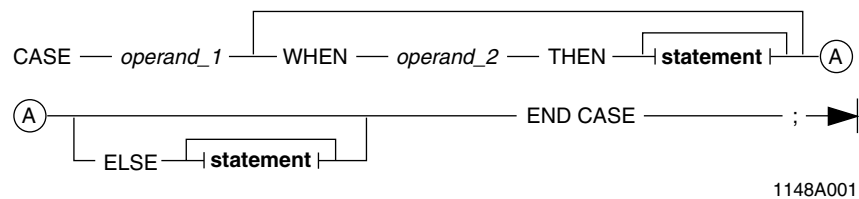
CHAPTER 7 SQL Stored Procedures: Control Statements and Condition Handling

BEGIN - END Statement

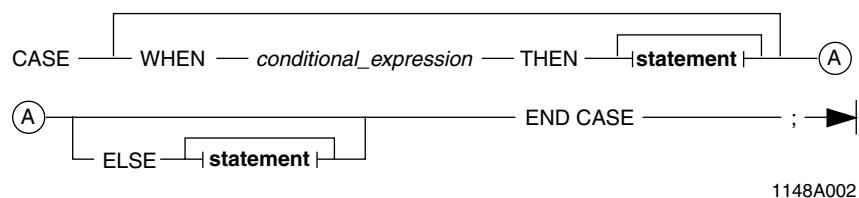


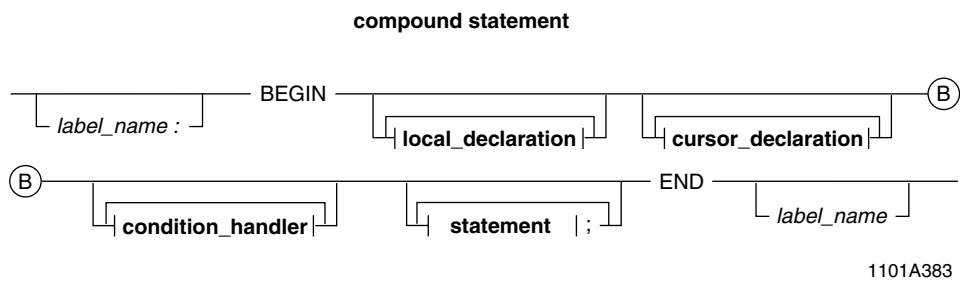
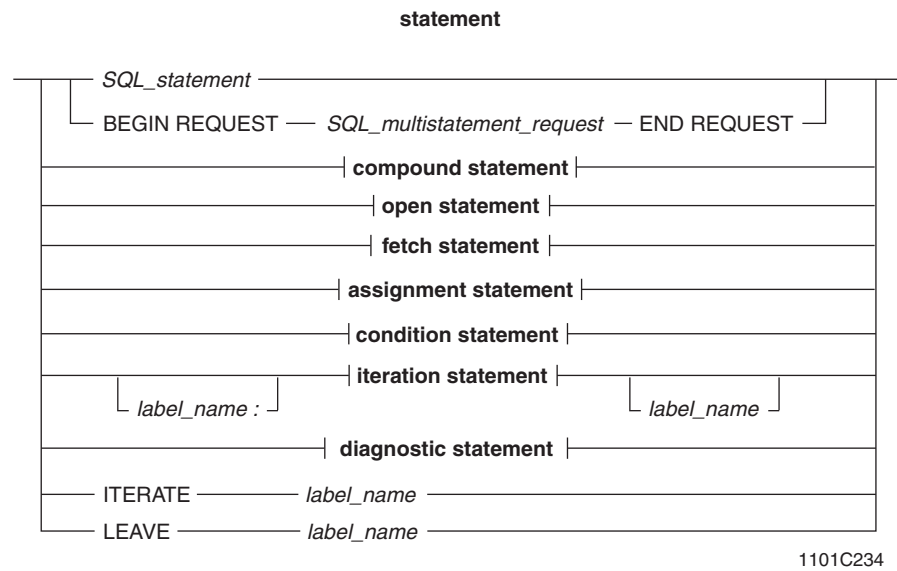
CASE

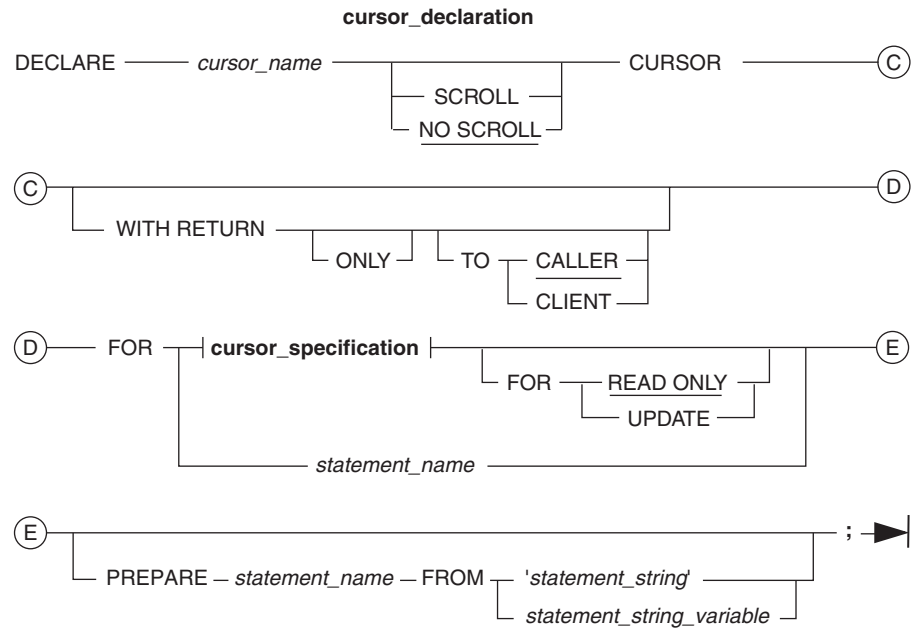
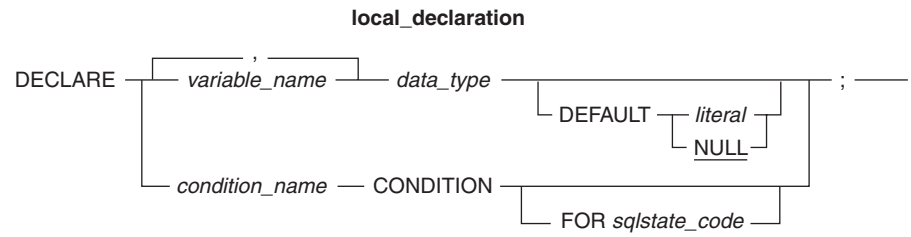
Syntax 1



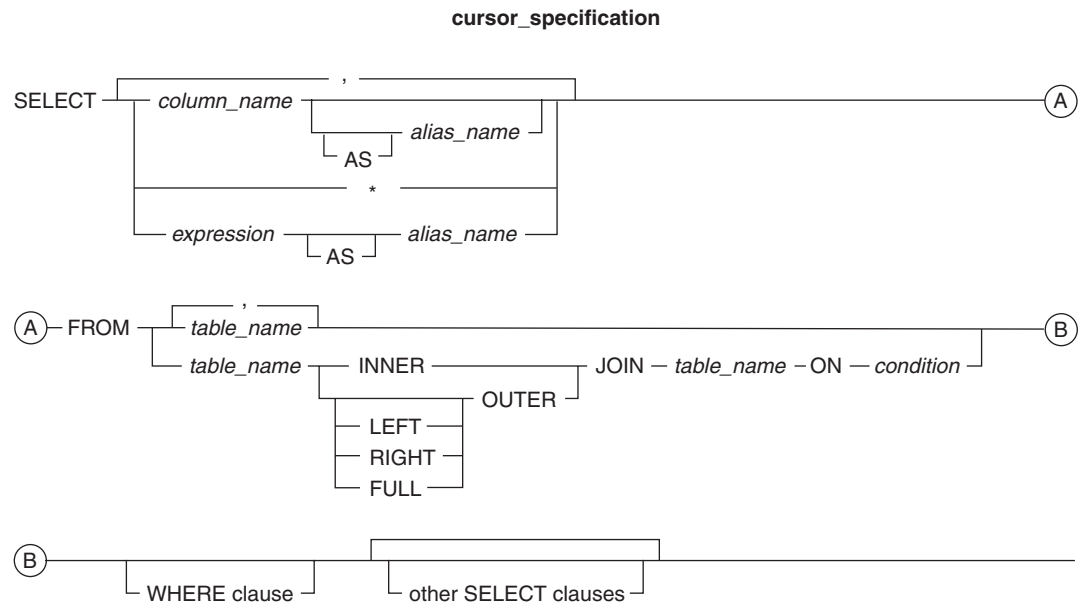
Syntax 2



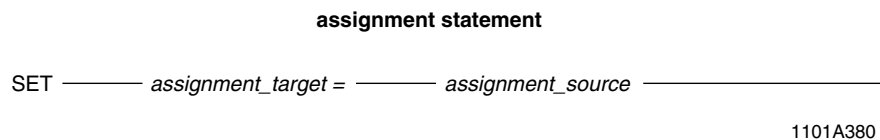
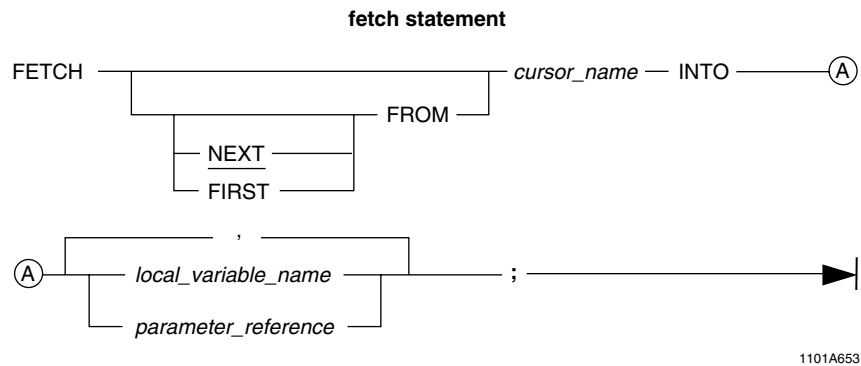
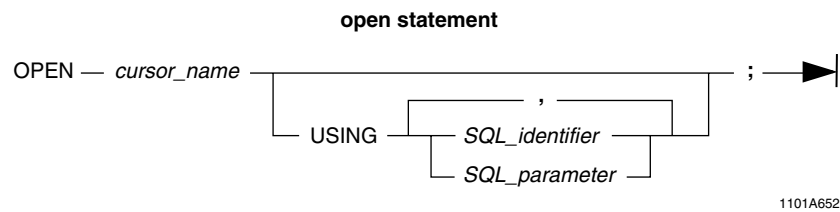
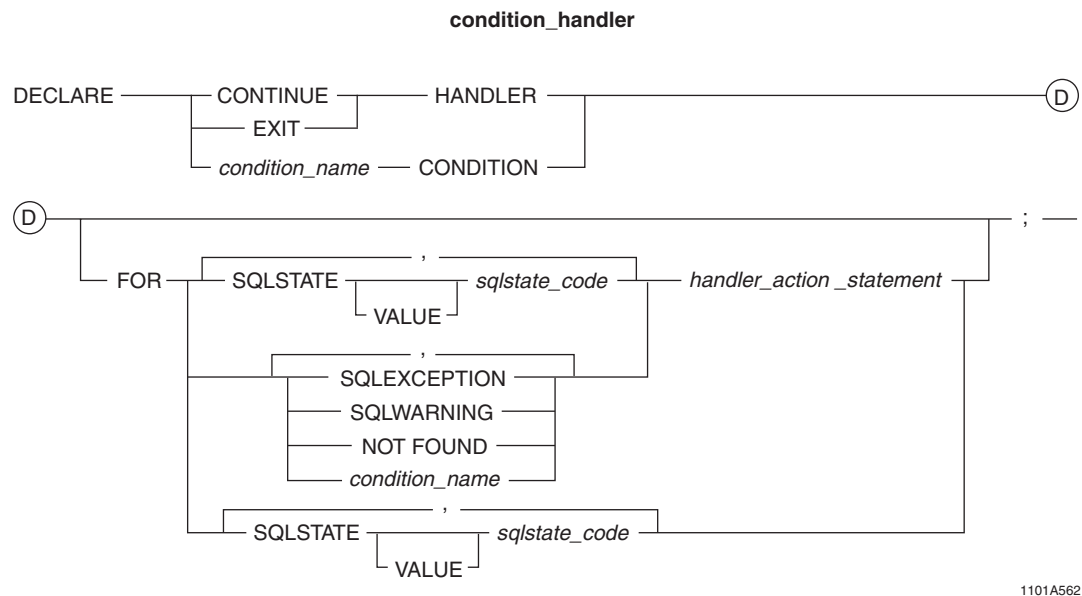




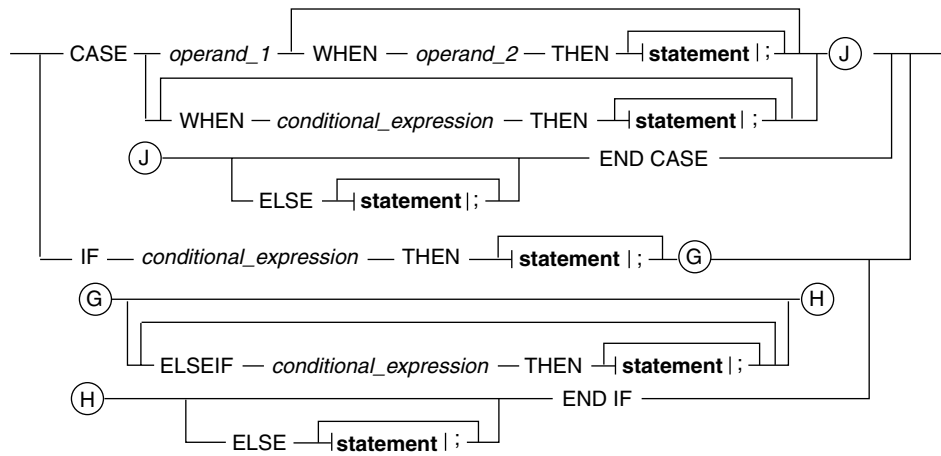
1101C448



1101B384

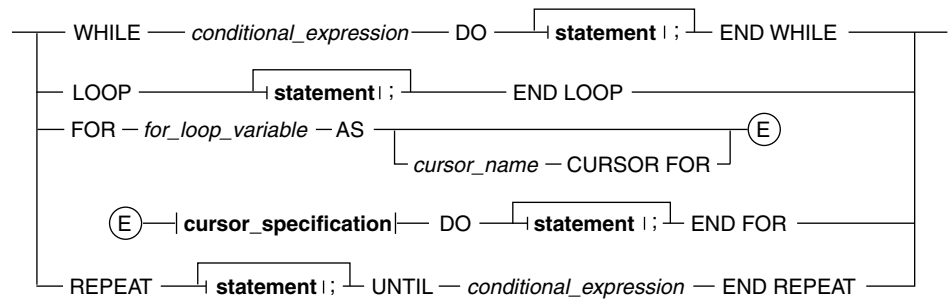


condition statement



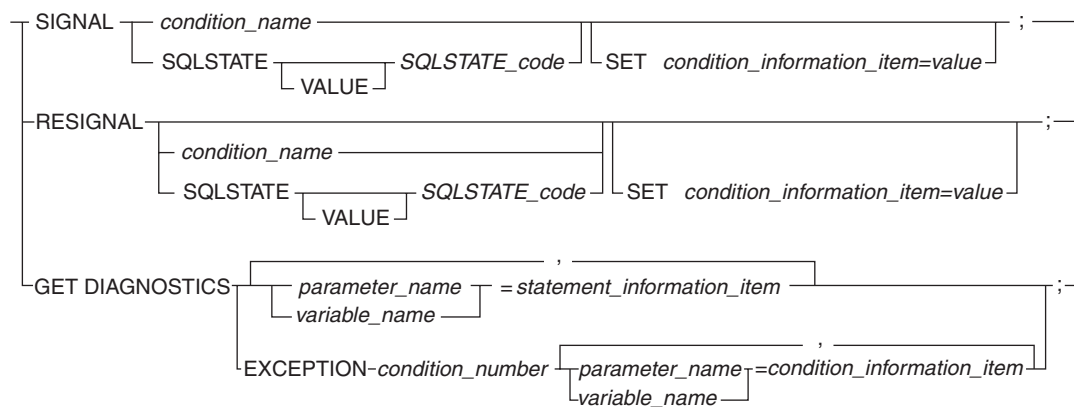
1101A381

iteration statement



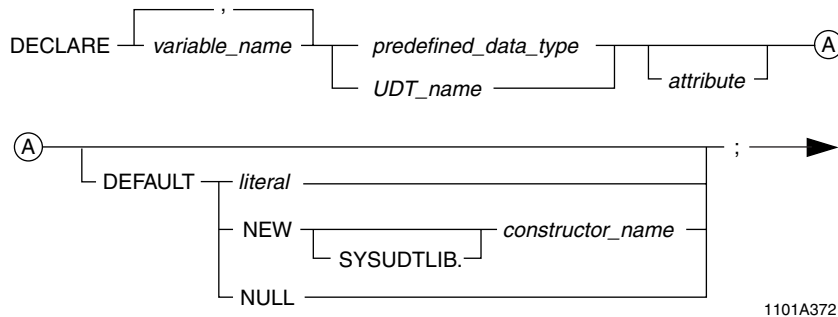
1101A382

diagnostic statement

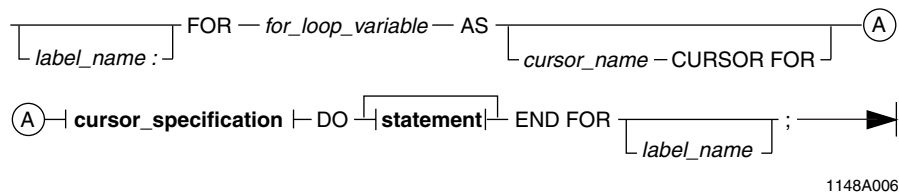


1101A616

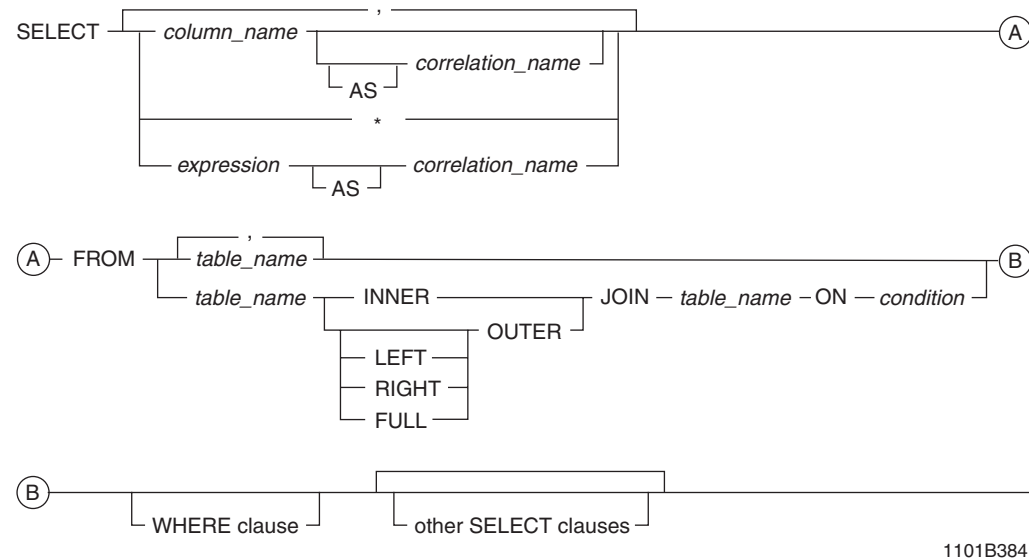
DECLARE



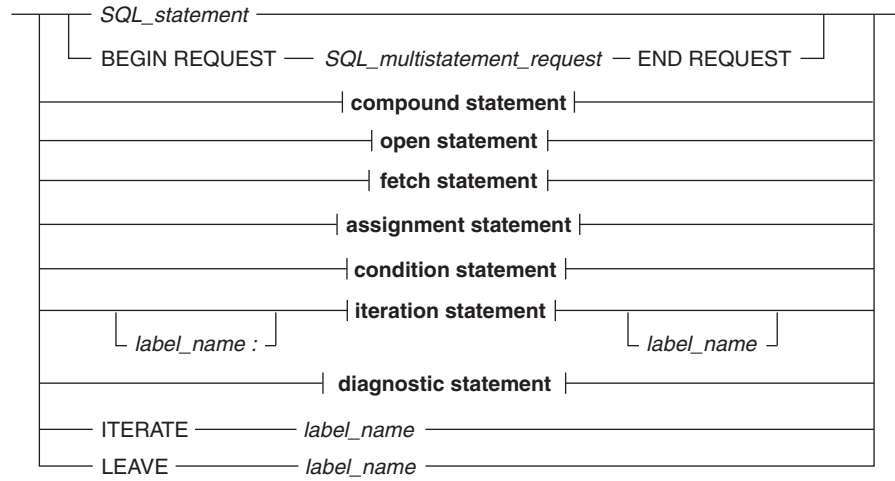
FOR



cursor_specification

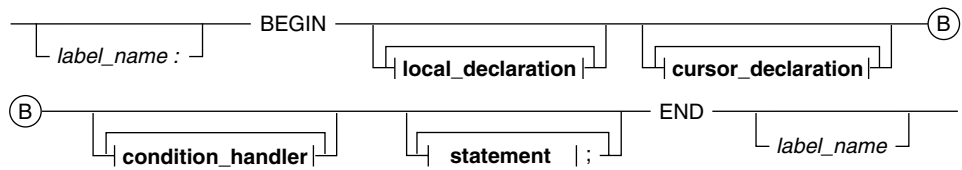


statement

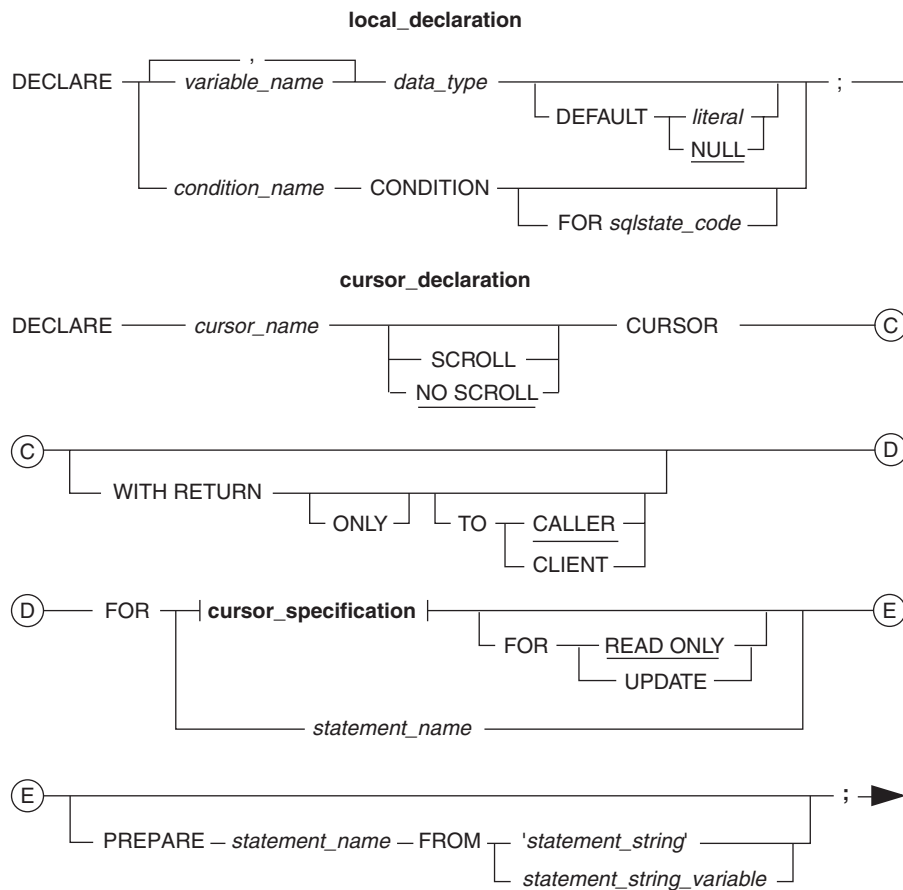


1101C234

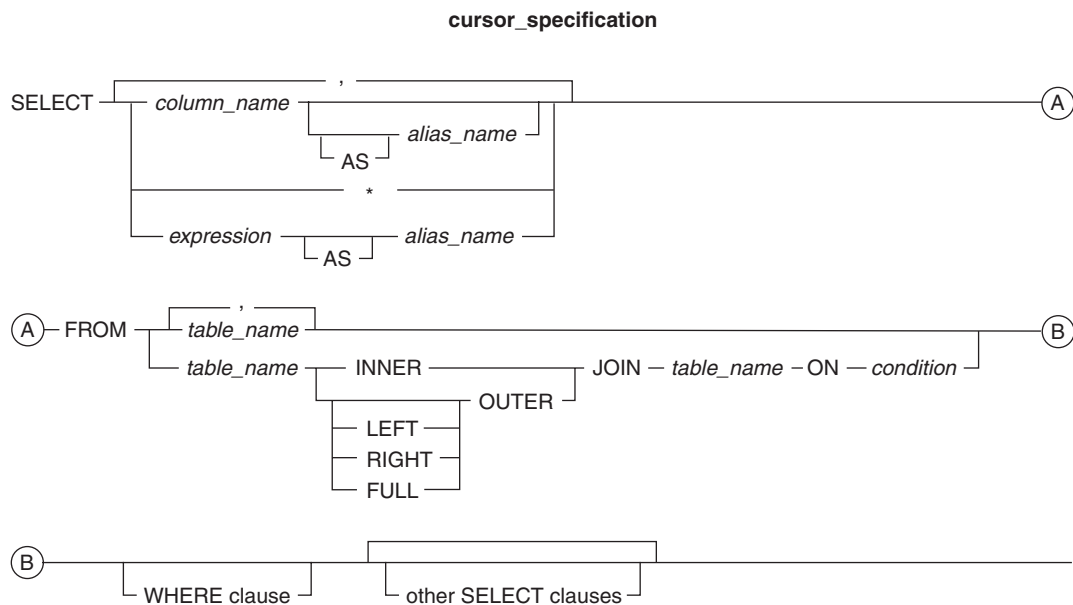
compound statement



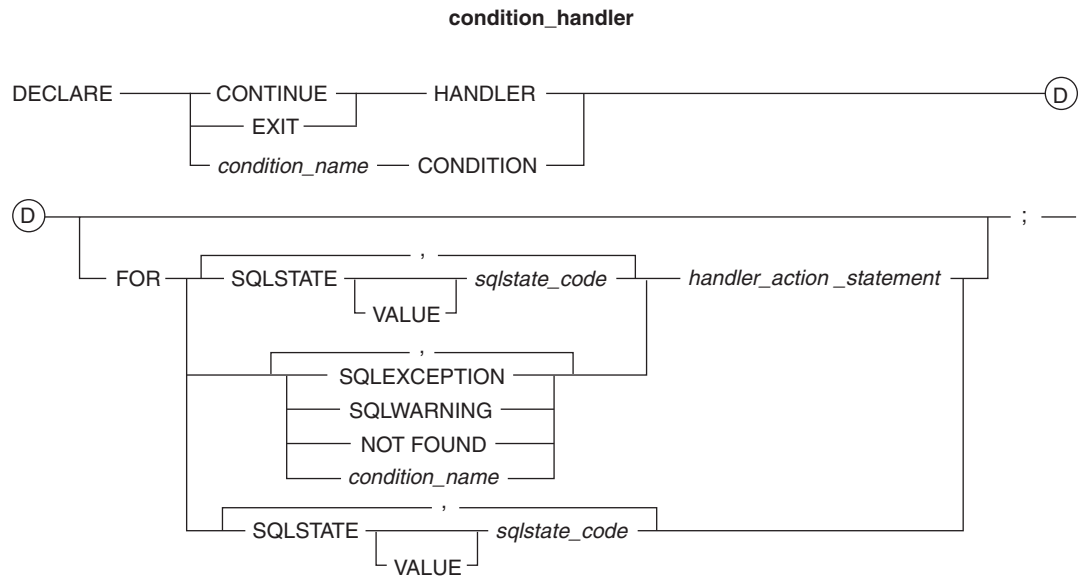
1101A383



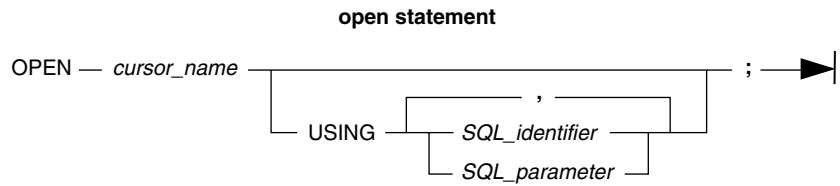
1101C448



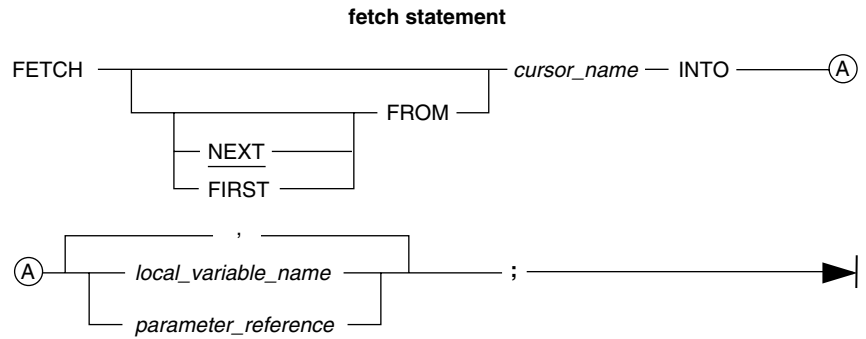
1101B384



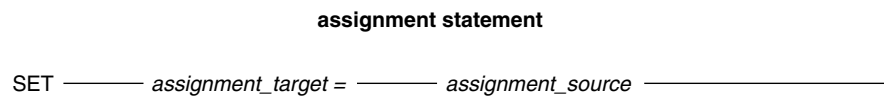
1101A562



1101A652

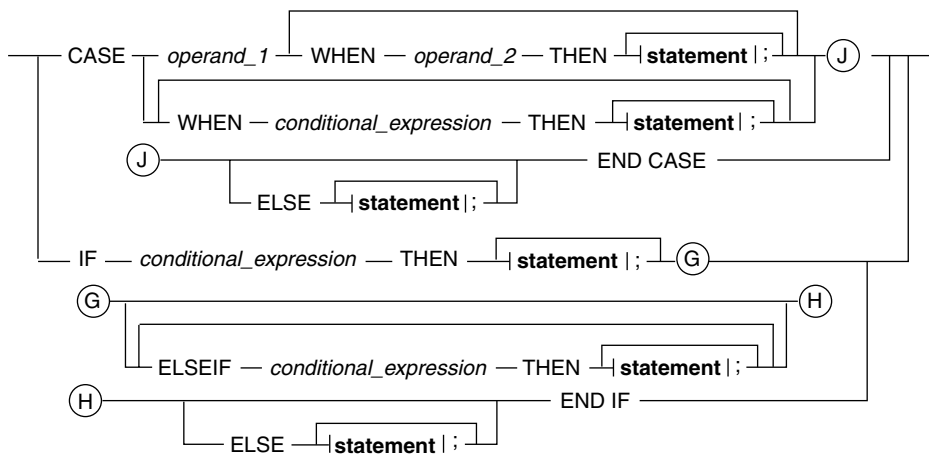


1101A653



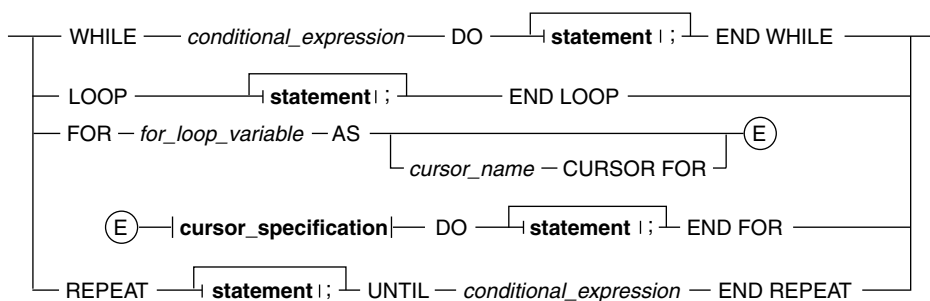
1101A380

condition statement



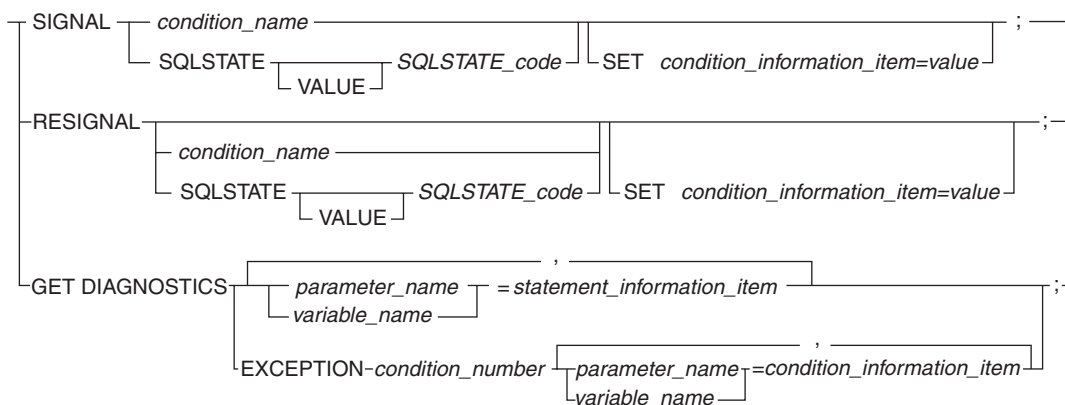
1101A381

iteration statement



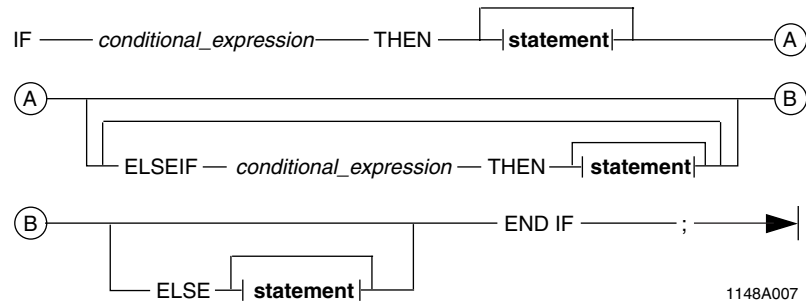
1101A382

diagnostic statement

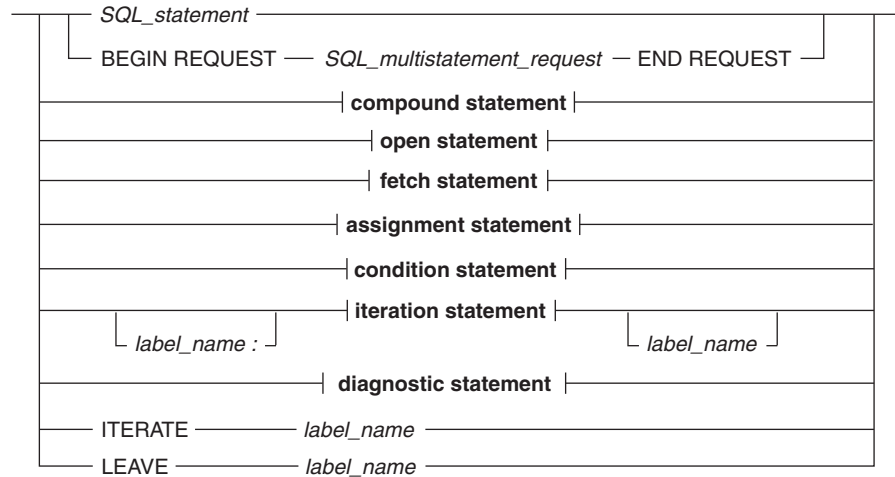


1101A616

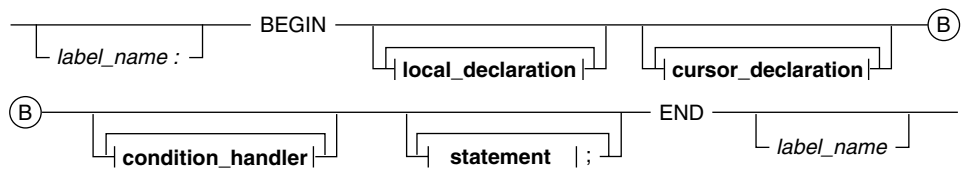
IF

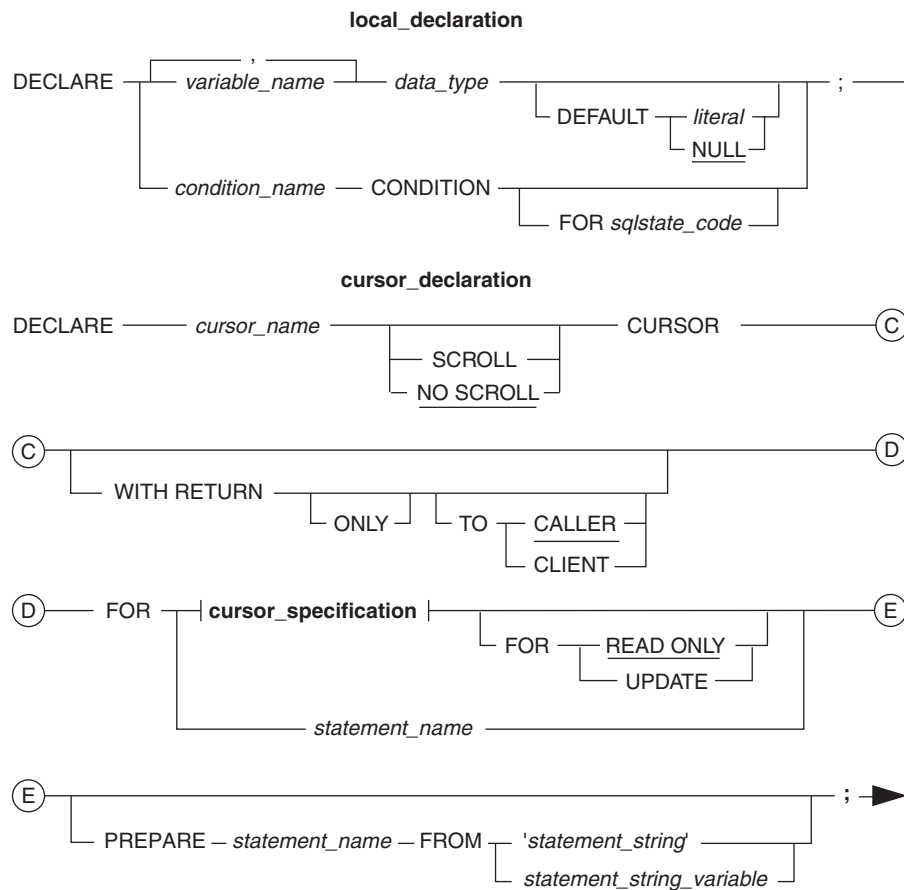


statement

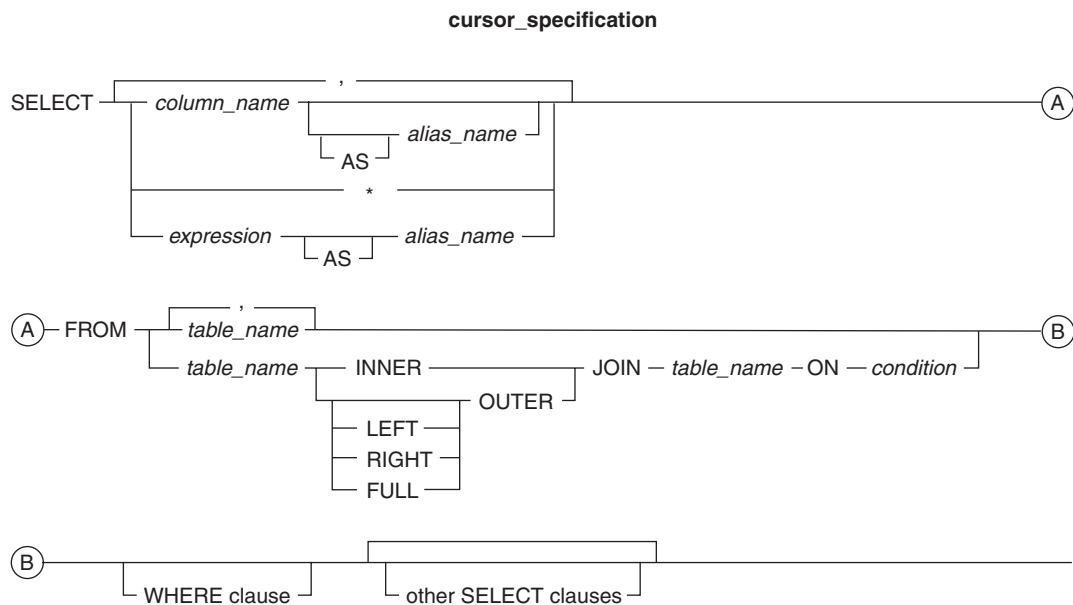


compound statement

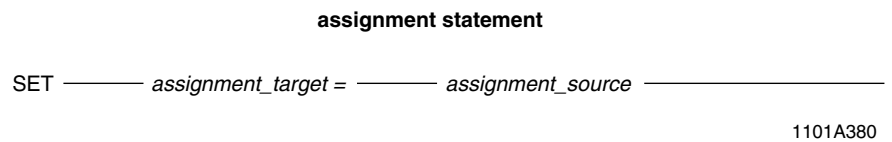
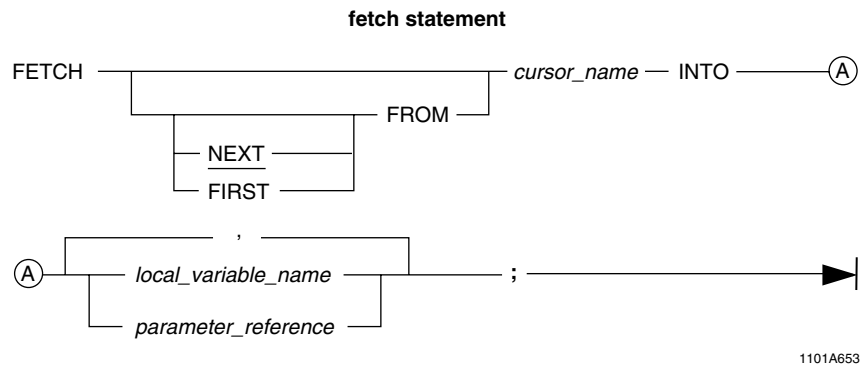
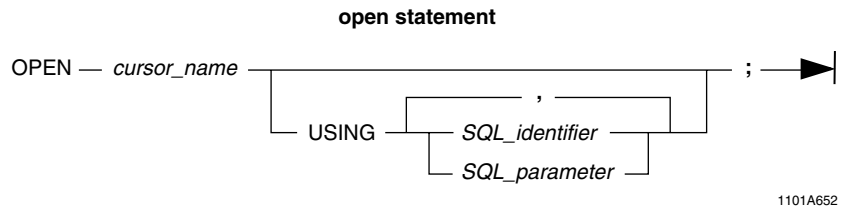
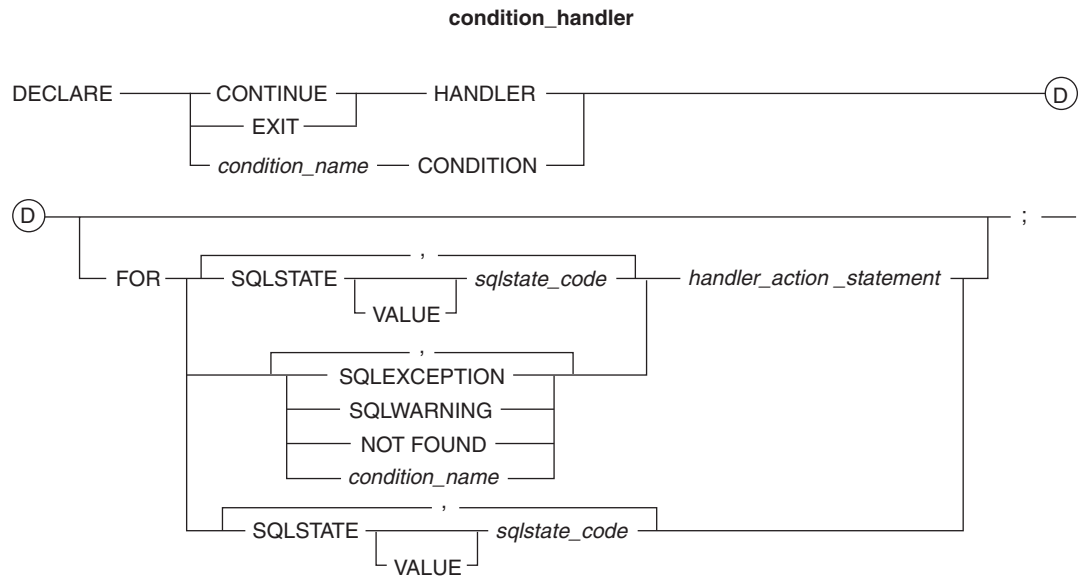




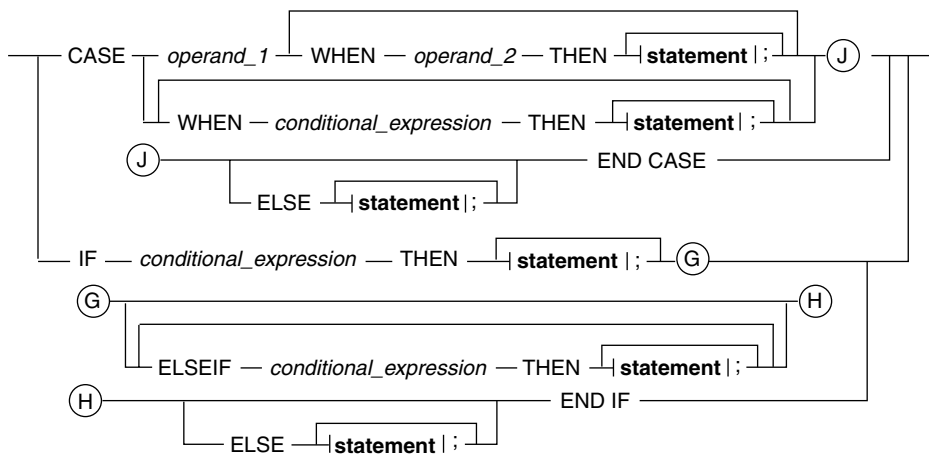
1101C448



1101B384

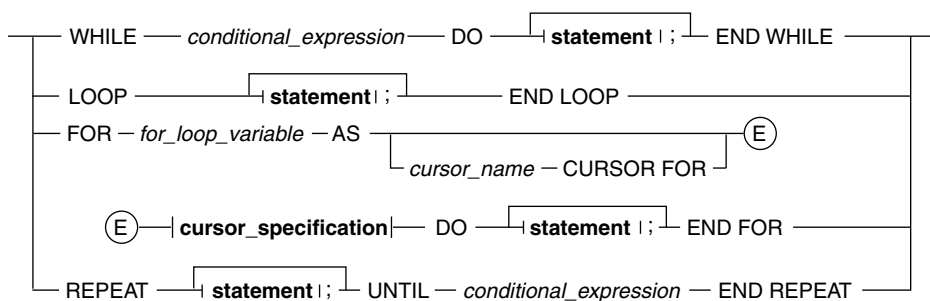


condition statement



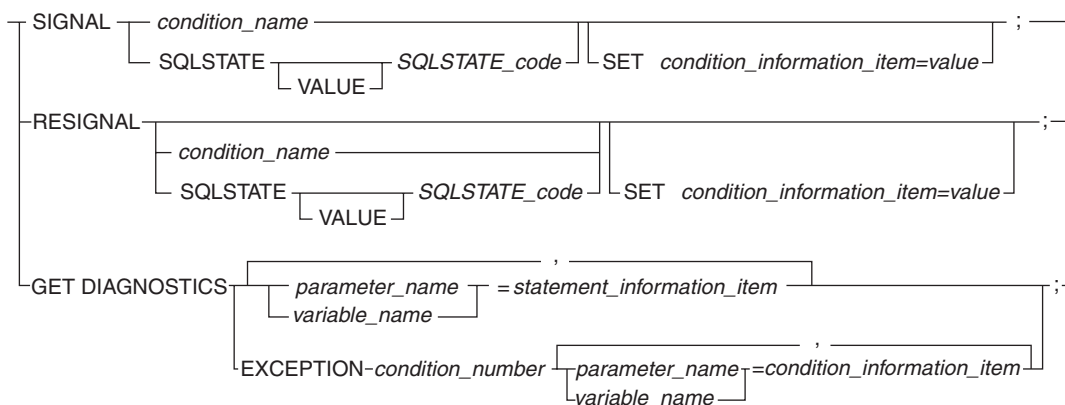
1101A381

iteration statement



1101A382

diagnostic statement



1101A616

ITERATE

ITERATE *label_name* ;

YS6ITER01

LEAVE

LEAVE *label_name* ;

YS6LEA01

LOOP

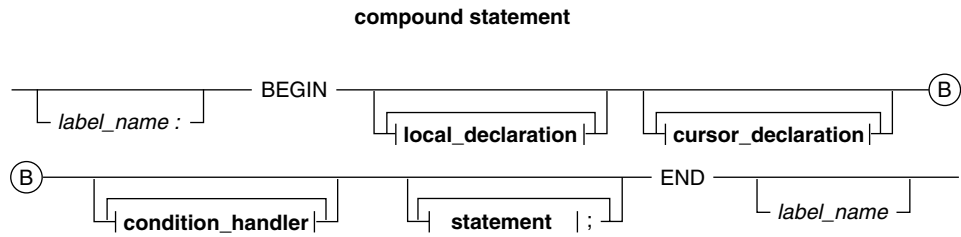
label_name : LOOP statement END LOOP *label_name* ;

1148A008

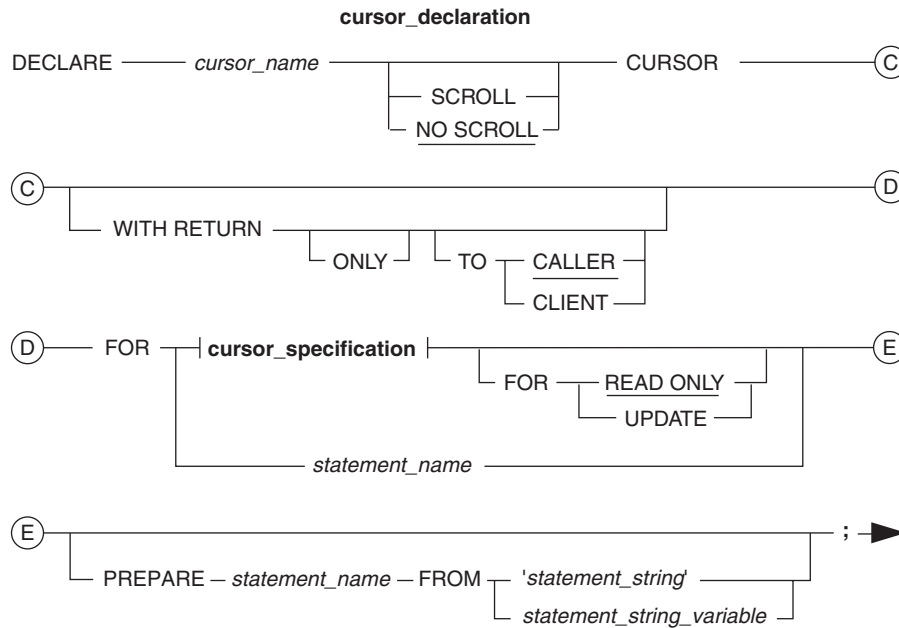
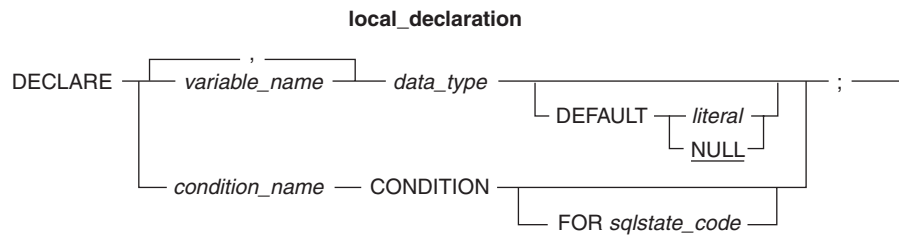
statement

<i>SQL_statement</i>	
BEGIN REQUEST	<i>SQL_multistatement_request</i> END REQUEST
compound statement	
open statement	
fetch statement	
assignment statement	
condition statement	
<i>label_name</i> :	iteration statement <i>label_name</i>
diagnostic statement	
ITERATE <i>label_name</i>	
LEAVE <i>label_name</i>	

1101C234

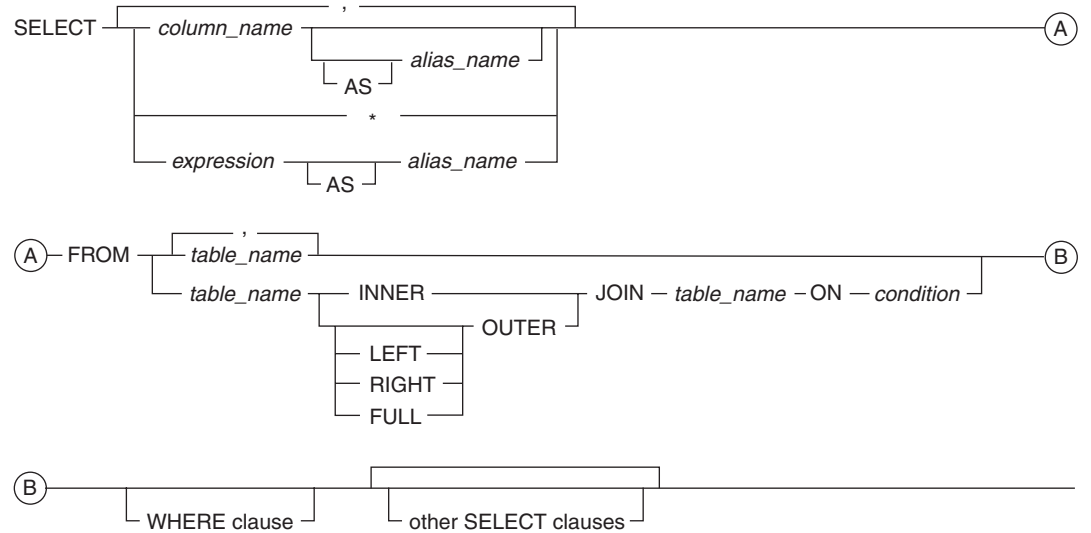


1101A383



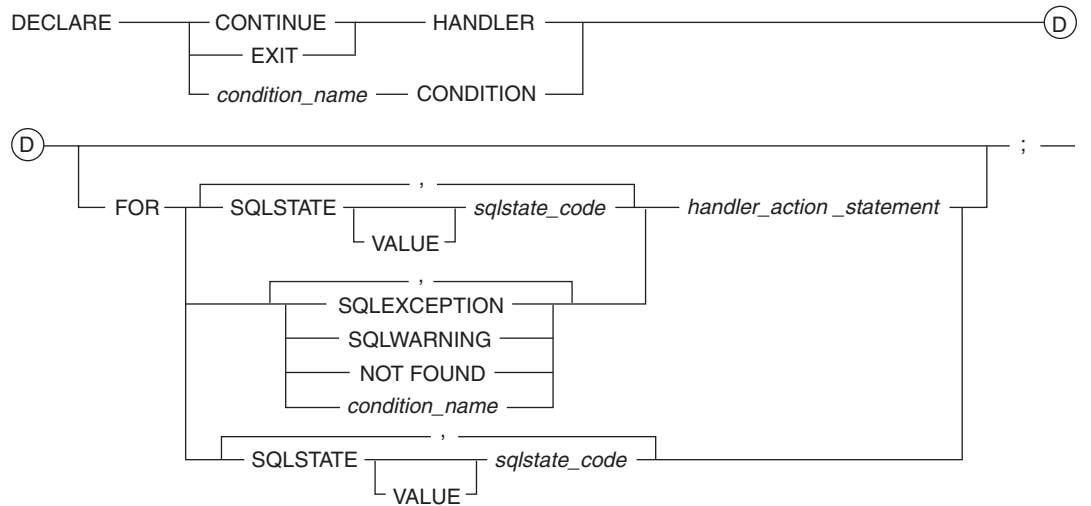
1101C448

cursor_specification



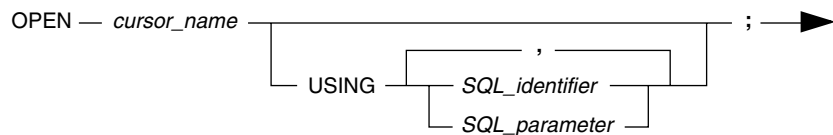
1101B384

condition_handler



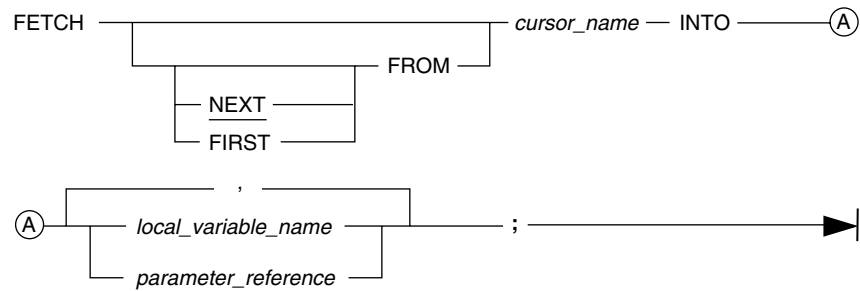
1101A562

open statement



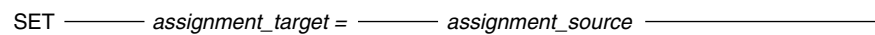
1101A652

fetch statement



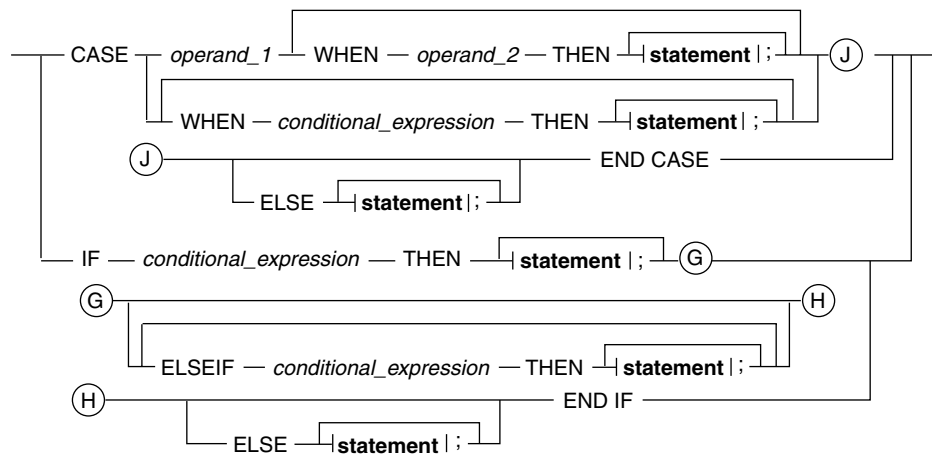
1101A653

assignment statement



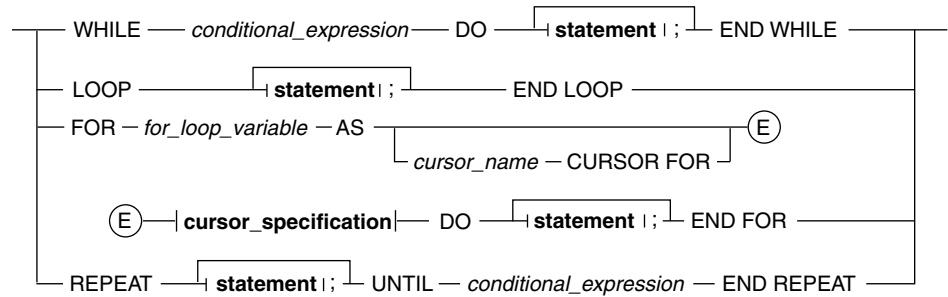
1101A380

condition statement



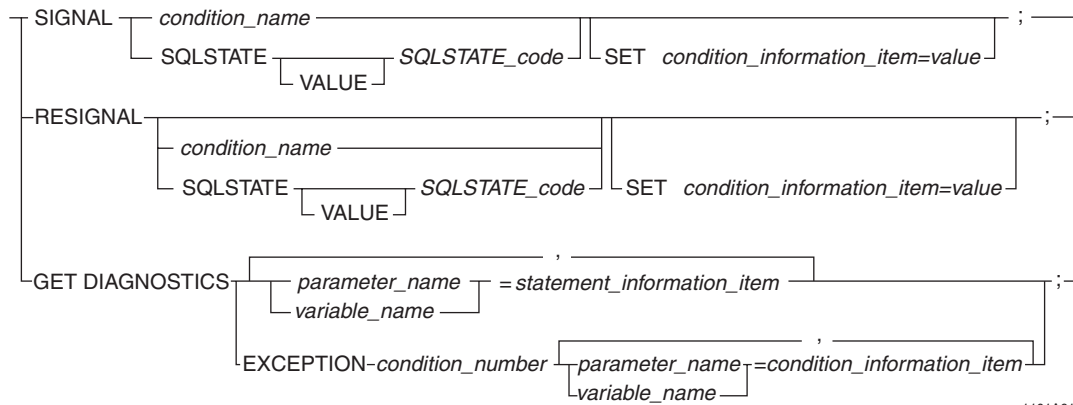
1101A381

iteration statement



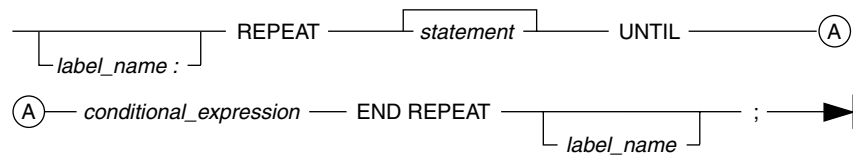
1101A382

diagnostic statement



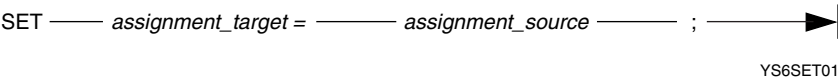
1101A616

REPEAT

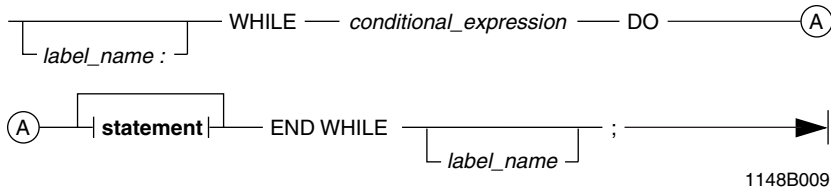


YS6RPT01

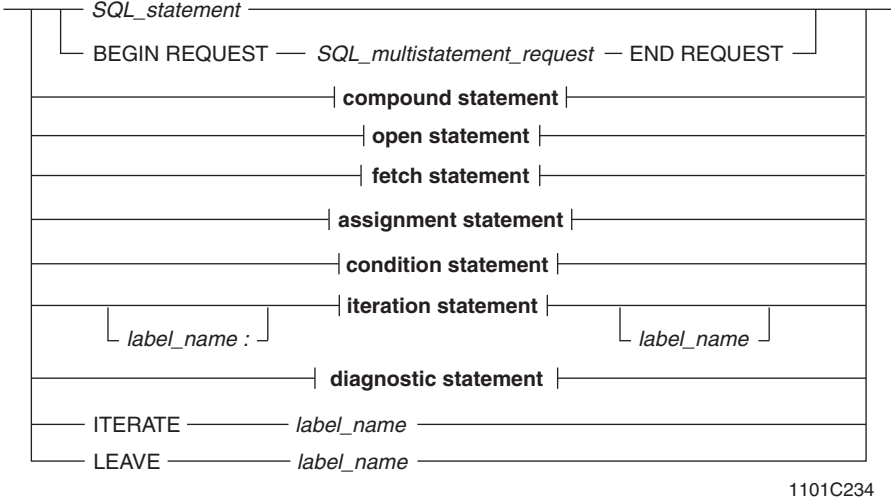
SET



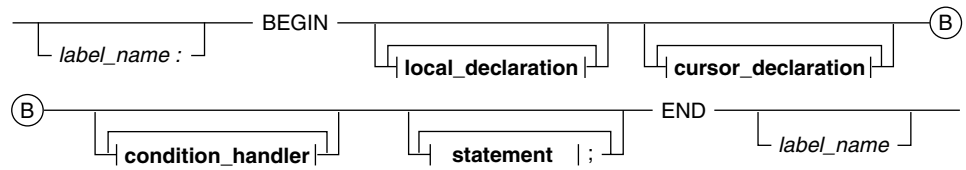
WHILE



statement

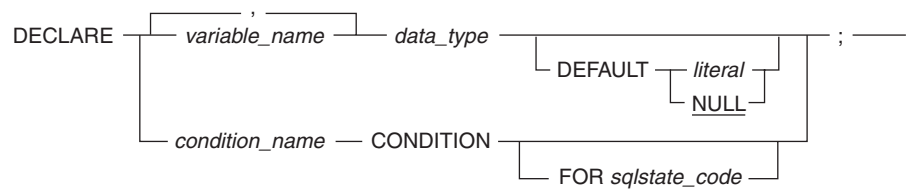


compound statement

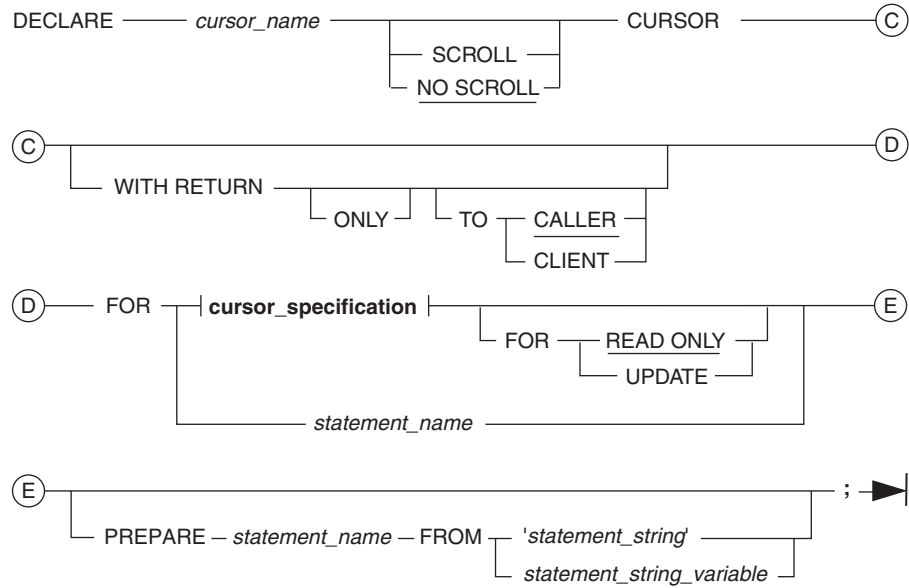


1101A383

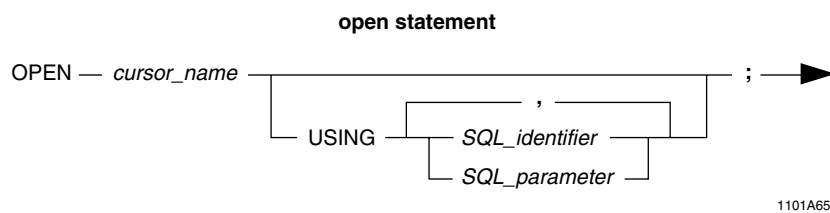
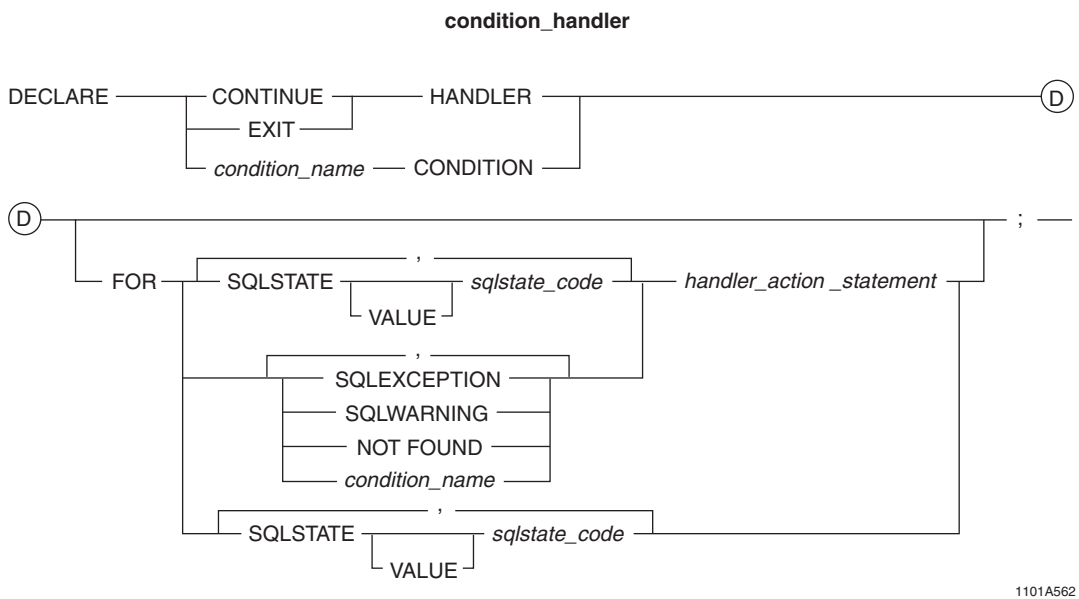
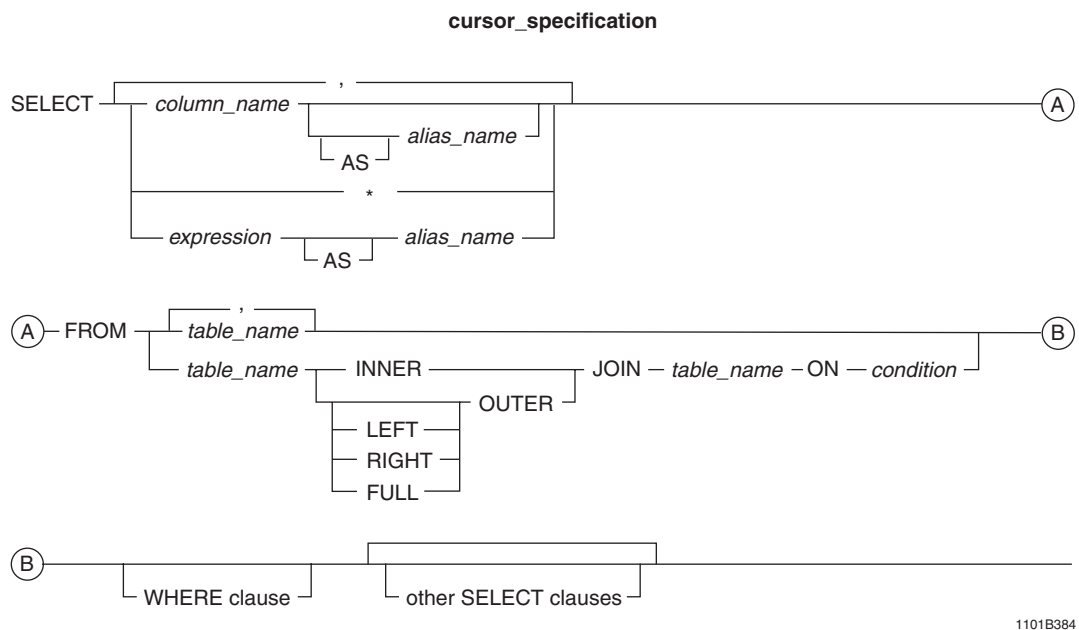
local_declaration



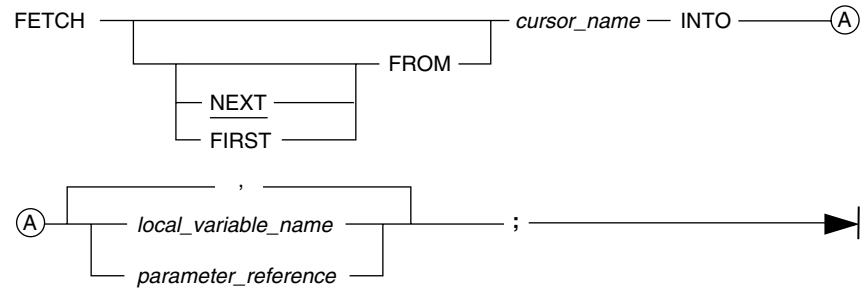
cursor_declaration



1101C448

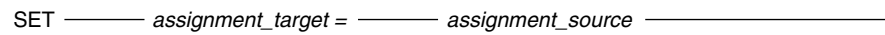


fetch statement



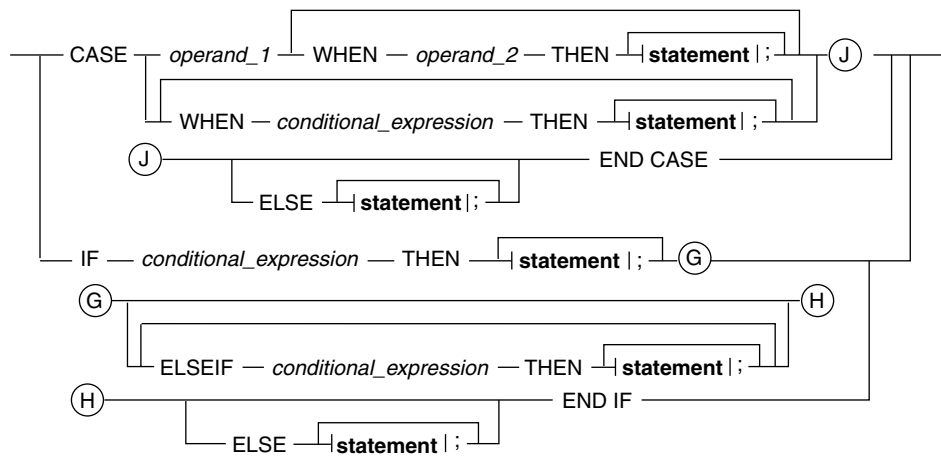
1101A653

assignment statement



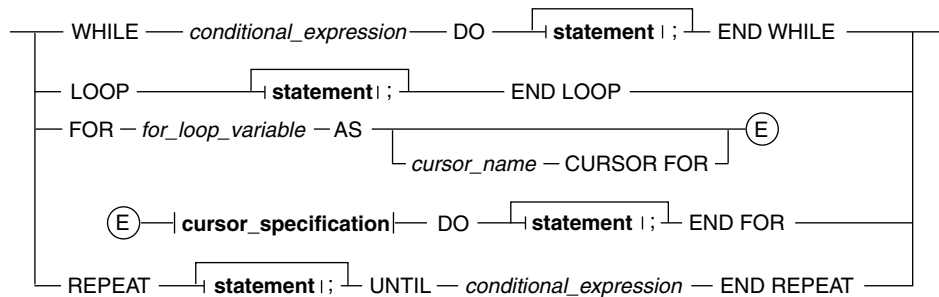
1101A380

condition statement



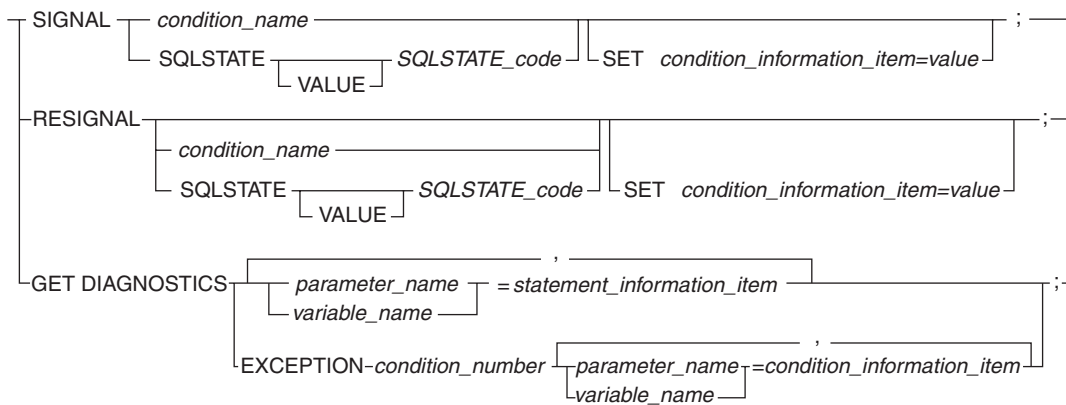
1101A381

iteration statement



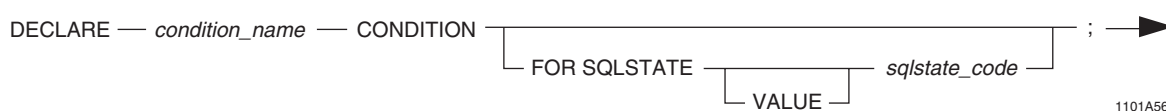
1101A382

diagnostic statement



1101A616

DECLARE CONDITION



1101A561

DECLARE HANDLER (Basic Syntax)



SIGNAL



RESIGNAL



GET DIAGNOSTICS

```
GET DIAGNOSTICS {  
    {  
        parameter_name  
        variable_name  
    } = statement_information_item  
    EXCEPTION-condition_number {  
        parameter_name  
        variable_name  
    } = condition_information_item  
};-
```

1101A636

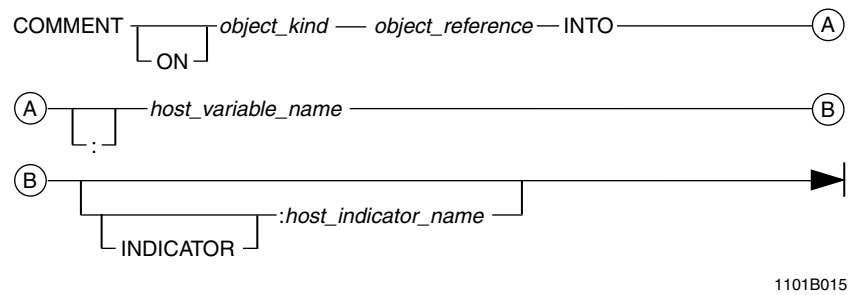
CHAPTER 8 Static Embedded SQL Statements

BEGIN DECLARE SECTION

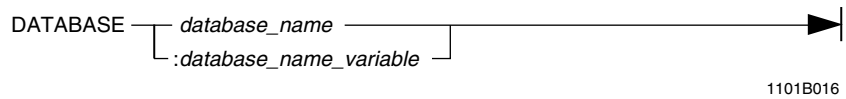


COMMENT

Returning Form



DATABASE



DECLARE STATEMENT

DECLARE — *statement_name* — STATEMENT —▶

GW01A013

DECLARE TABLE

DECLARE — *table_name* — TABLE — (A)
 — *view_name* —
(A) — (— *column_name* — *data_type* — *null_attribute* —) —▶

GW01R014

END DECLARE SECTION

END DECLARE SECTION —▶

GW01A016

END-EXEC Statement Terminator

— END-EXEC — [.] —▶

FF07D287

EXEC

EXEC — *macro_name* — (parameter_list) —▶

1101B043

EXEC SQL Statement Prefix

EXEC SQL — FOR : count_value — *embedded_sql_statement* — *sql_statement_terminator* —▶

1101A396

INCLUDE

INCLUDE — *include_file_name* —▶

1101B044

INCLUDE SQLCA

INCLUDE SQLCA —▶

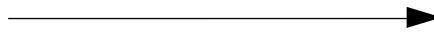

GW01A021

INCLUDE SQLDA

INCLUDE SQLDA  

GW01A022

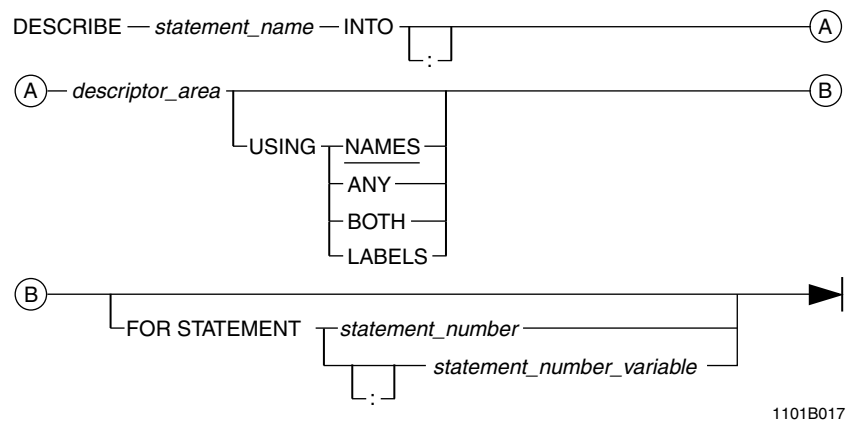
WHENEVER

WHENEVER — *condition* — *action*  

GW01R035

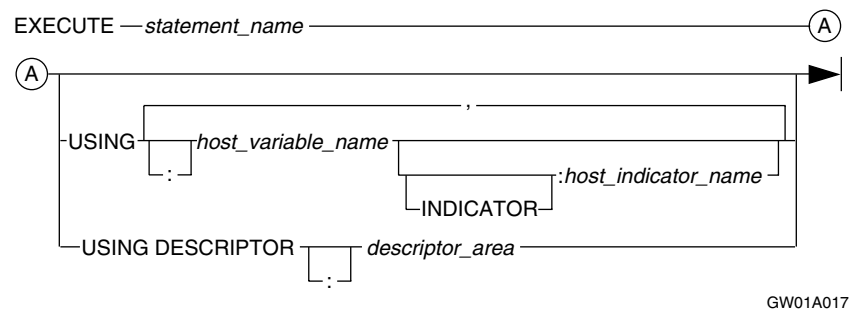
CHAPTER 9 Dynamic Embedded SQL Statements

DESCRIBE

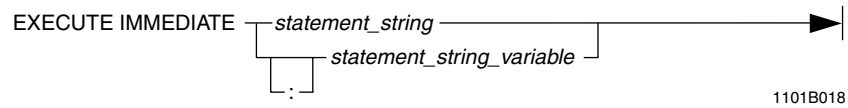


EXECUTE

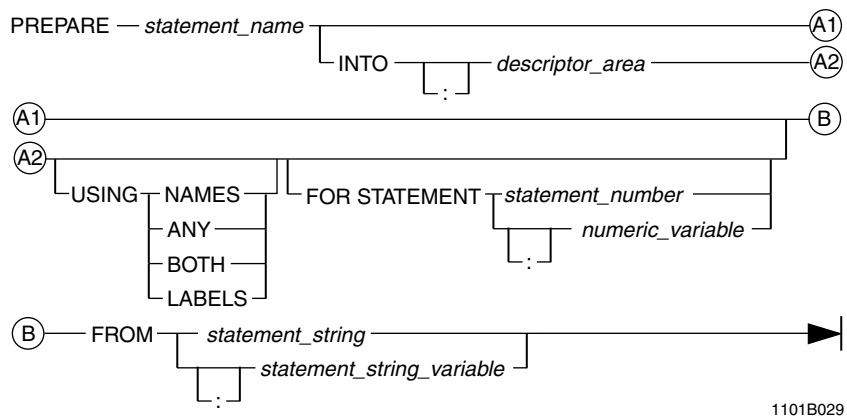
Dynamic SQL Form



EXECUTE IMMEDIATE

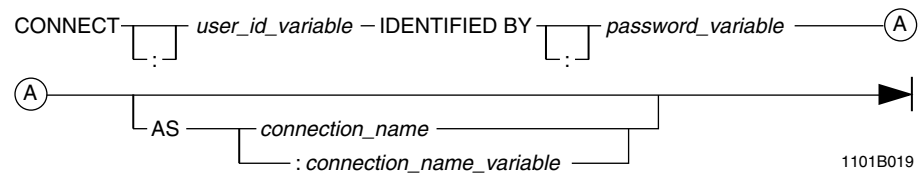


PREPARE

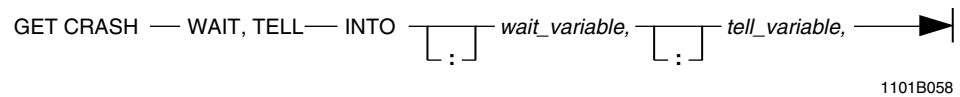


CHAPTER 10 SQL Client-Server Connectivity Statements

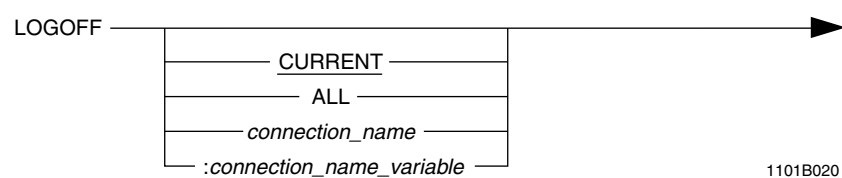
CONNECT



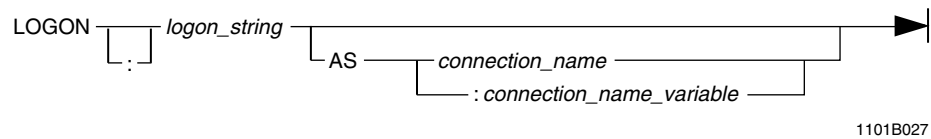
GET CRASH



LOGOFF



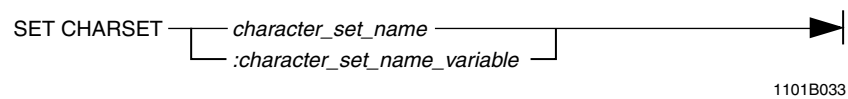
LOGON



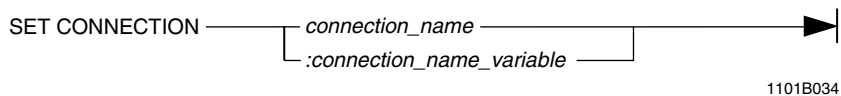
SET BUFFERSIZE



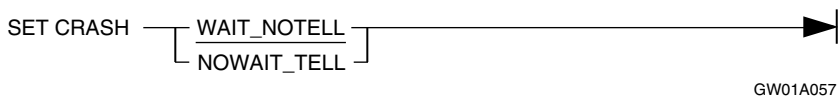
SET CHARSET



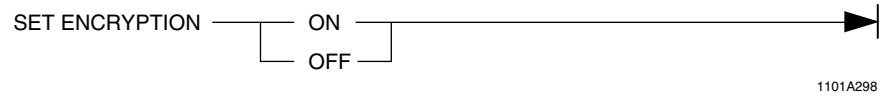
SET CONNECTION



SET CRASH



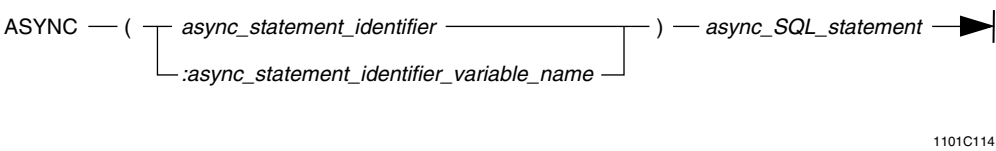
SET ENCRYPTION



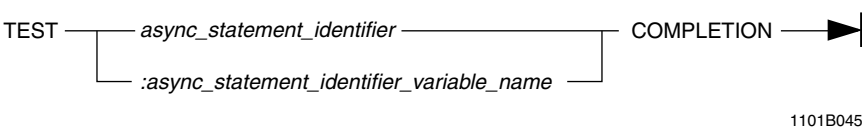
CHAPTER 11

Multisession Asynchronous Programming With Embedded SQL

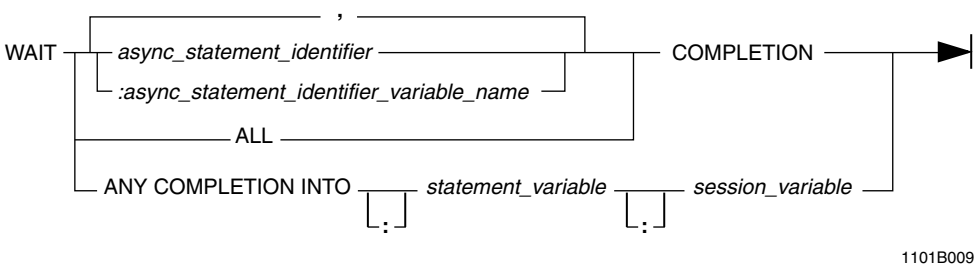
ASYNC Statement Modifier



TEST



WAIT



APPENDIX A How to Read Syntax Diagrams

This appendix describes the conventions that apply to reading the syntax diagrams used in this book.

Syntax Diagram Conventions

Notation Conventions

Item	Definition / Comments
Letter	An uppercase or lowercase alphabetic character ranging from A through Z.
Number	A digit ranging from 0 through 9. Do not use commas when typing a number with more than 3 digits.
Word	Keywords and variables. <ul style="list-style-type: none">UPPERCASE LETTERS represent a keyword. Syntax diagrams show all keywords in uppercase, unless operating system restrictions require them to be in lowercase.lowercase letters represent a keyword that you must type in lowercase, such as a UNIX command.<i>lowercase italic letters</i> represent a variable such as a column or table name. Substitute the variable with a proper value.lowercase bold letters represent an excerpt from the diagram. The excerpt is defined immediately following the diagram that contains it.<u>UNDERLINED LETTERS</u> represent the default value. This applies to both uppercase and lowercase words.
Spaces	Use one space between items such as keywords or variables.
Punctuation	Type all punctuation exactly as it appears in the diagram.

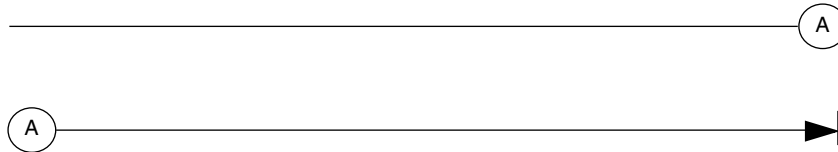
Paths

The main path along the syntax diagram begins at the left with a keyword, and proceeds, left to right, to the vertical bar, which marks the end of the diagram. Paths that do not have an arrow or a vertical bar only show portions of the syntax.

The only part of a path that reads from right to left is a loop.

Continuation Links

Paths that are too long for one line use continuation links. Continuation links are circled letters indicating the beginning and end of a link:



FE0CA002

When you see a circled letter in a syntax diagram, go to the corresponding circled letter and continue reading.

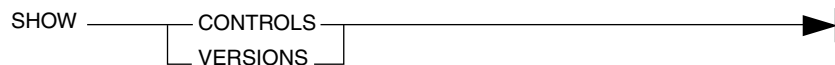
Required Entries

Required entries appear on the main path:



FE0CA003

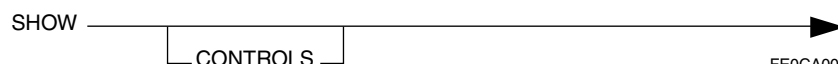
If you can choose from more than one entry, the choices appear vertically, in a stack. The first entry appears on the main path:



FE0CA005

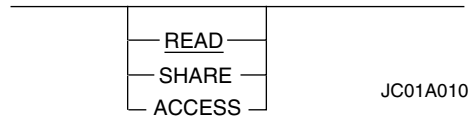
Optional Entries

You may choose to include or disregard optional entries. Optional entries appear below the main path:



FE0CA004

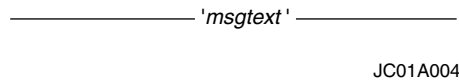
If you can optionally choose from more than one entry, all the choices appear below the main path:



Some commands and statements treat one of the optional choices as a default value. This value is UNDERLINED. It is presumed to be selected if you type the command or statement without specifying one of the options.

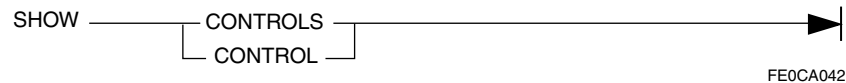
Strings

String literals appear in single quotes:



Abbreviations

If a keyword or a reserved word has a valid abbreviation, the unabbreviated form always appears on the main path. The shortest valid abbreviation appears beneath.

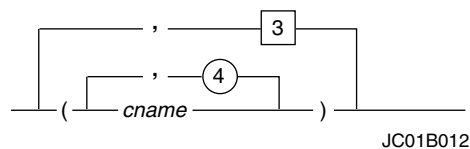


In the above syntax, the following formats are valid:

- SHOW CONTROLS
- SHOW CONTROL

Loops

A loop is an entry or a group of entries that you can repeat one or more times. Syntax diagrams show loops as a return path above the main path, over the item or items that you can repeat:



Read loops from right to left.

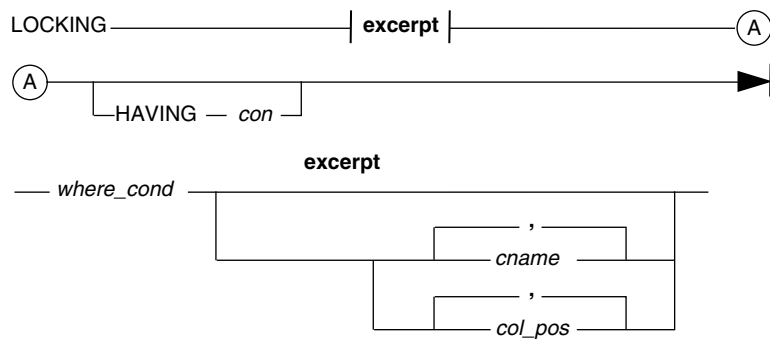
The following conventions apply to loops:

IF...	THEN...
there is a maximum number of entries allowed	the number appears in a circle on the return path. In the example, you may type <i>cname</i> a maximum of 4 times.
there is a minimum number of entries required	the number appears in a square on the return path. In the example, you must type at least three groups of column names.
a separator character is required between entries	the character appears on the return path. If the diagram does not show a separator character, use one blank space. In the example, the separator character is a comma.
a delimiter character is required around entries	the beginning and end characters appear outside the return path. Generally, a space is not needed between delimiter characters and entries. In the example, the delimiter characters are the left and right parentheses.

Excerpts

Sometimes a piece of a syntax phrase is too large to fit into the diagram. Such a phrase is indicated by a break in the path, marked by (|) terminators on each side of the break. The name for the excerpted piece appears between the terminators in boldface type.

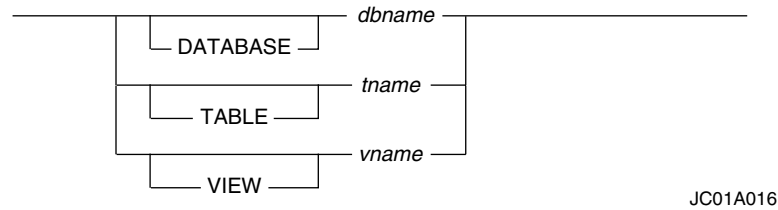
The boldface excerpt name and the excerpted phrase appears immediately after the main diagram. The excerpted phrase starts and ends with a plain horizontal line:



JC01A014

Multiple Legitimate Phrases

In a syntax diagram, it is possible for any number of phrases to be legitimate:



In this example, any of the following phrases are legitimate:

- *dbname*
- DATABASE *dbname*
- *tname*
- TABLE *tname*
- *vname*
- VIEW *vname*

Sample Syntax Diagram

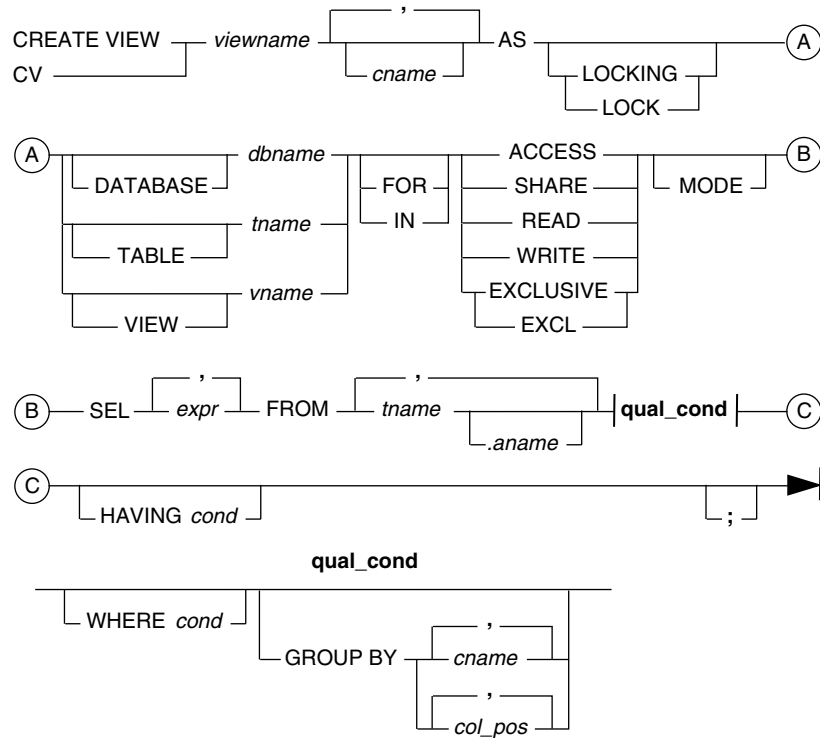


Diagram Identifier

The alphanumeric string that appears in the lower right corner of every diagram is an internal identifier used to catalog the diagram. The text never refers to this string.