TERADATA

Teradata Database

# SQL Quick Reference

Release 13.0
B035-1510-098B
April 2009

# 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<br>April 2009 | • Updated syntax diagrams throughout the book<br>• Added the following new syntax diagrams to Chapter 1:<br>  • Period Literals<br>  • PERIOD(DATE) Data Type<br>  • PERIOD(TIME) Data Type<br>  • PERIOD(TIME WITH TIME ZONE) Data Type<br>  • PERIOD(TIMESTAMP) Data Type |

| Date | Description |
|---|---|
| Teradata Database 13.0 (Continued) | • Added the following new syntax diagrams to Chapter 1:<br>  • PERIOD(TIMESTAMP WITH TIME ZONE) Data Type<br>  • VARIANT_TYPE UDT<br>  • Geospatial Data Types<br>• Added the following new syntax diagrams to Chapter 2:<br>  • CURRENT_ROLE<br>  • CURRENT_USER<br>  • CONTAINS<br>  • IS UNTIL_CHANGED/IS NOT UNTIL_CHANGED<br>  • MEETS<br>  • PRECEDES<br>  • SUCCEEDS<br>  • BEGIN<br>  • END<br>  • LAST<br>  • INTERVAL<br>  • PRIOR<br>  • NEXT<br>  • P_INTERSECT<br>  • LDIFF<br>  • RDIFF<br>  • P_NORMALIZE<br>  • Period Value Constructor<br>  • Arithmetic Operators<br>  • Scalar UDF Expression<br>  • Aggregate UDF Expression<br>  • NEW VARIANT_TYPE<br>• Added the following new syntax diagrams to Chapter 3:<br>  • COLLECT STATISTICS (Recollect Statistics)<br>  • CREATE GLOP SET<br>  • CREATE REPLICATION RULESET/REPLACE REPLICATION RULESET<br>  • DROP GLOP SET<br>  • DROP REPLICATION RULESET<br>  • SHOW QUERY LOGGING<br>• Added the following new syntax diagrams to Chapter 4:<br>  • GRANT CONNECT THROUGH<br>  • REVOKE CONNECT THROUGH<br>• Added the following new syntax diagrams to Chapter 5:<br>  • HASH BY Clause<br>  • LOCAL ORDER BY Clause |

| Date | Description |
|---|---|
| Teradata Database 13.0 (Continued) | • Added the following new syntax diagrams to Chapter 7:<br>  • SIGNAL<br>  • RESIGNAL<br>  • GET DIAGNOSTICS |
| Teradata Database 12.0<br>September 2007 | • Updated existing syntax diagrams in Chapters 2, 3, 4, 5, 6, 7, 8, 9 and 10<br>• Added the following new syntax diagrams to Chapter 1:<br>  • Hexadecimal Name Literals<br>  • Unicode Character String Literals<br>  • Unicode Delimited Identifier<br>• Added the following new syntax diagrams to Chapter 2:<br>  • DEGREES, RADIANS<br>  • STRING_CS<br>• Modified the syntax diagrams for Window Aggregate Functions in Chapter 2 to support the following aggregate functions:<br>  • CORR<br>  • COVAR_POP<br>  • COVAR_SAMP<br>  • REGR_AVGX<br>  • REGR_AVGY<br>  • REGR_COUNT<br>  • REGR_INTERCEPT<br>  • REGR_R2<br>  • REGR_SLOPE<br>  • REGR_SXX<br>  • REGR_SXY<br>  • REGR_SYY<br>  • STDDEV_POP<br>  • STDDEV_SAMP<br>  • VAR_POP<br>  • VAR_SAMP<br>• Added the following new syntax diagrams to Chapter 3:<br>  • CREATE ERROR TABLE<br>  • Java Simple and Object Mapped Data Types<br>  • DROP ERROR TABLE<br>  • LOGGING ONLINE ARCHIVE OFF<br>  • LOGGING ONLINE ARCHIVE ON<br>  • SET QUERY_BAND |

| Date | Description |
|---|---|
| Teradata Database 12.0 (Continued) | • Added the following new syntax diagrams to Chapter 5:<br>  • INITIATE PARTITION ANALYSIS<br>  • DIAGNOSTIC COSTPRINT<br>  • DIAGNOSTIC DUMP COSTS<br>  • DIAGNOSTIC HELP COSTS<br>  • DIAGNOSTIC SET COSTS<br>  • DIAGNOSTIC HELP PROFILE<br>  • DIAGNOSTIC SET PROFILE<br>• Added the following new syntax diagrams to Chapter 6:<br>  • HELP COLUMN - Syntaxes 6, 7 and 8<br>  • HELP ERROR TABLE |

# Additional Information

| URL | Description |
|---|---|
| http://www.info.teradata.com/ | Use the Teradata Information Products Publishing Library site to:<br>• View or download a manual:<br>  1 Under **Online Publications**, select **General Search**.<br>  2 Enter your search criteria and click **Search**.<br>• Download a documentation CD-ROM:<br>  1 Under **Online Publications**, select **General Search**.<br>  2 In the **Title or Keyword** field, enter *CD-ROM*, and click **Search**.<br>• Order printed manuals:<br>  Under **Print & CD Publications**, select **How to Order**. |
| http://www.teradata.com | The Teradata home page provides links to numerous sources of information about Teradata. Links include:<br>• Executive reports, case studies of customer experiences with Teradata, and thought leadership<br>• Technical information, solutions, and expert advice<br>• Press releases, mentions and media resources |
| http://teradatauniversitynetwork.com | Teradata University Network fosters education on data warehousing, business intelligence (BI) and database administration (DBA). |

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:

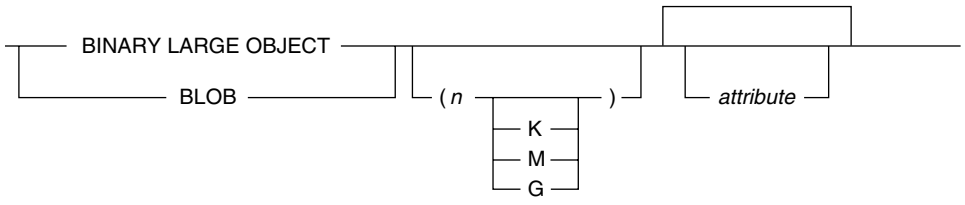
- “Windows” is Microsoft Windows Server 2003 64-bit.
- “Linux” is SUSE Linux Enterprise Server 9 and SUSE Linux Enterprise Server 10.

# Table of Contents

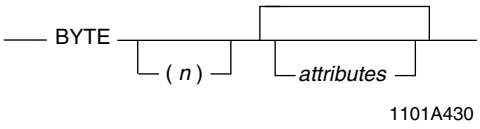
# CHAPTER 1 Data Types and Literals

## Byte and BLOB Data Types

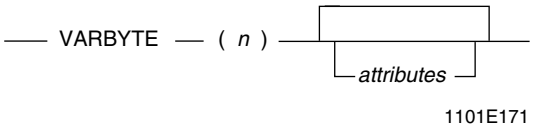
### BLOB Data Type




### BYTE Data Type

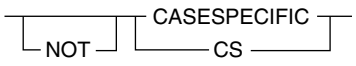



### VARBYTE Data Type



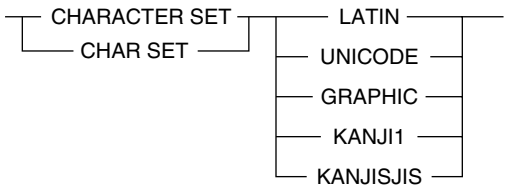
# Character and CLOB Data Types

## CASESPECIFIC Phrase



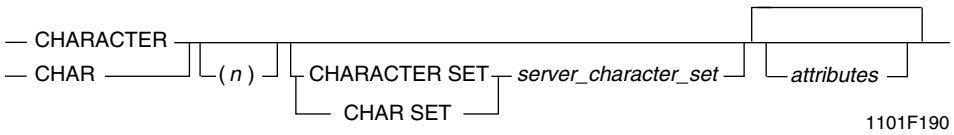

FF07D205

## CHARACTER SET Phrase

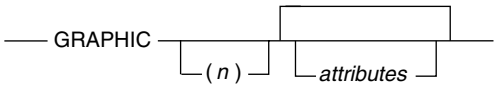



1101A409

## CHARACTER Data Type

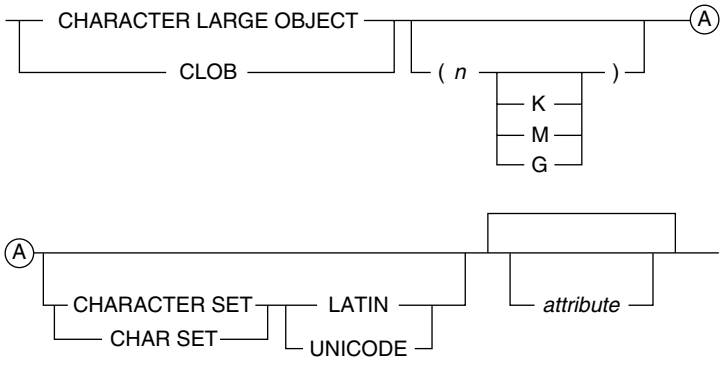



1101F190



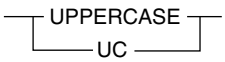

FF07D234

## CLOB Data Type



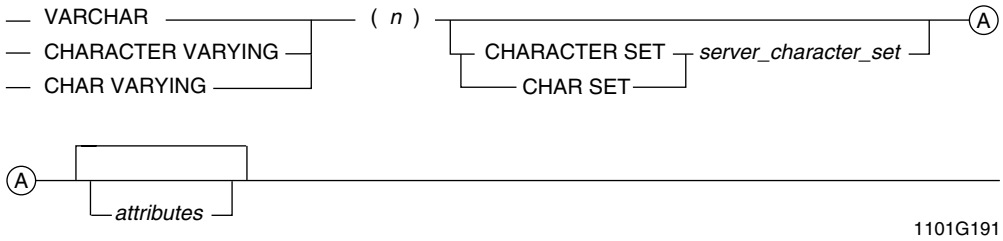

1101B093

## UPPERCASE Phrase

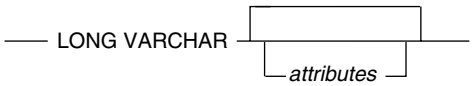



FF07D206

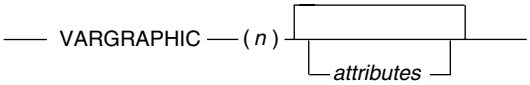
## VARCHAR Data Type



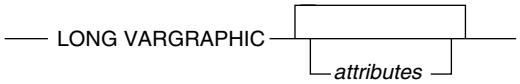

1101G191




FF07D192



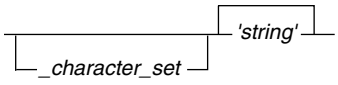

FF07D235




FF07D236

# Data Literals

## CHARACTER String Literals

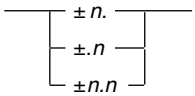



1101A009

## DATE Literals

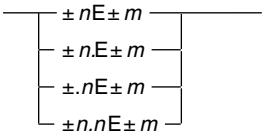

1101A036

## DECIMAL Literals




KR01A143

## FLOATING POINT Literals

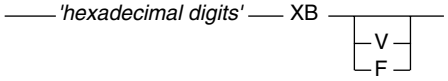



KR01A142

## GRAPHIC Literals

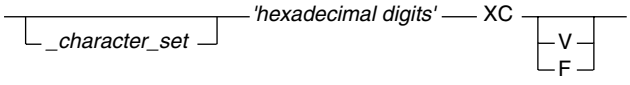

HH01A090

## Hexadecimal Byte Literals



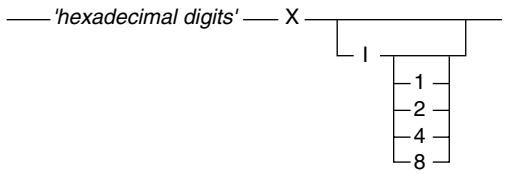

1101A391

## Hexadecimal Character Literals




1101A392

## Hexadecimal Integer Literals




1101A390

## Hexadecimal Name Literals


1101A494

## INTEGER Literals

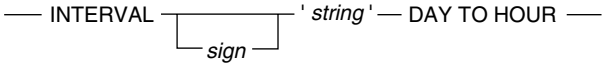



KR01A141

## INTERVAL DAY Literals

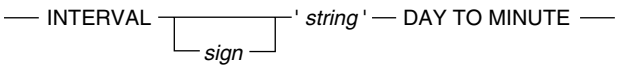

1101A026

## INTERVAL DAY TO HOUR Literals



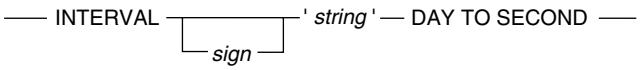

1101A027

## INTERVAL DAY TO MINUTE Literals



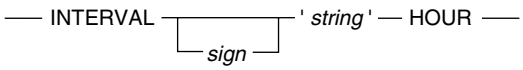

1101A028

## INTERVAL DAY TO SECOND Literals



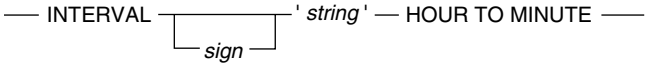

1101A029

## INTERVAL HOUR Literals



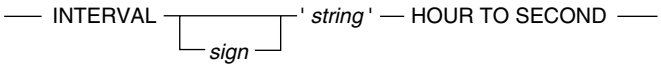

1101A030

## INTERVAL HOUR TO MINUTE Literals



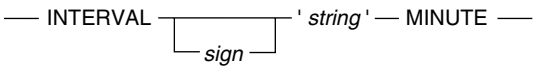

1101A037

## INTERVAL HOUR TO SECOND Literals



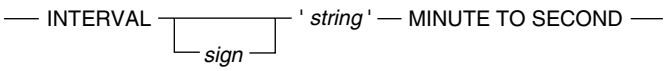

1101A038

## INTERVAL MINUTE Literals



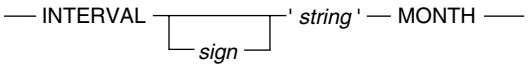

1101A031

## INTERVAL MINUTE TO SECOND Literals



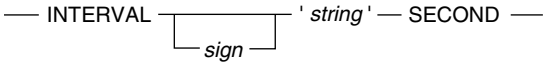

1101A032

## INTERVAL MONTH Literals




1101A025

## INTERVAL SECOND Literals

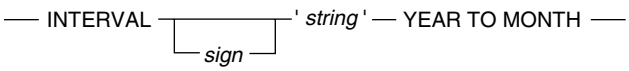



1101A033

## INTERVAL YEAR Literals

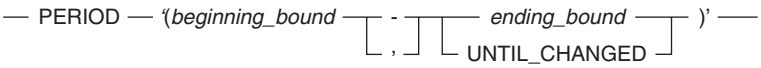

1101A023

## INTERVAL YEAR TO MONTH Literals



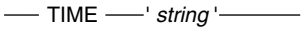

1101A024

## Period Literals



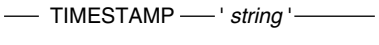

1101A591

## Time Literals



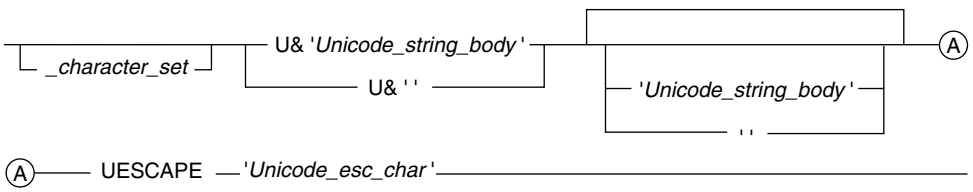

1101A021

## Timestamp Literals



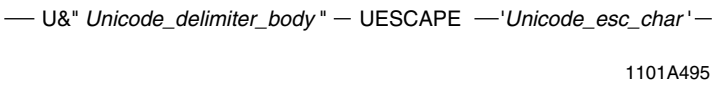

1101A022

## Unicode Character String Literals



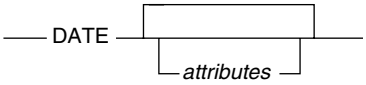

1101A493

### Unicode Delimited Identifier

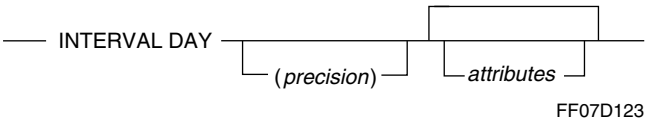



# DateTime and Interval Data Types

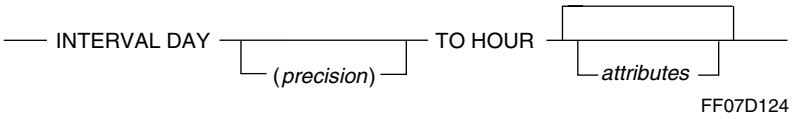
## DATE Data Type



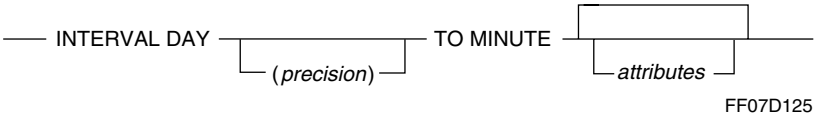

## INTERVAL DAY Data Type



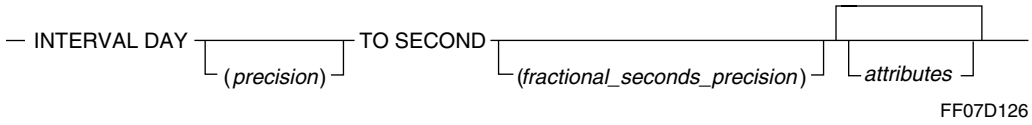

## INTERVAL DAY TO HOUR Data Type



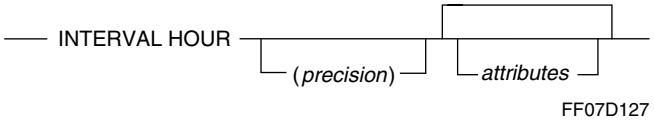

## INTERVAL DAY TO MINUTE Data Type



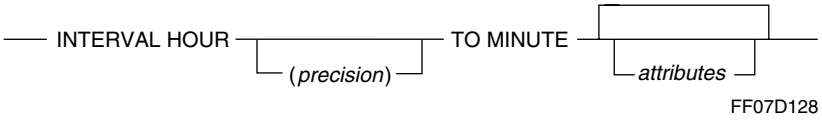
## INTERVAL DAY TO SECOND Data Type



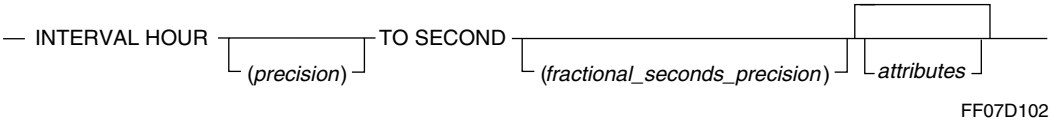

## INTERVAL HOUR Data Type



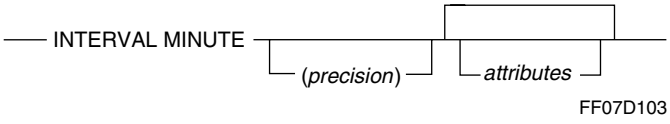

## INTERVAL HOUR TO MINUTE Data Type



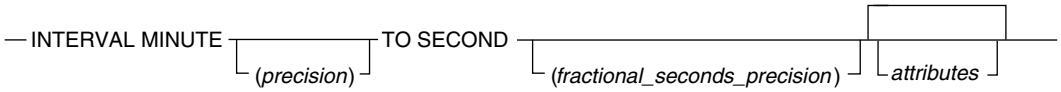

## INTERVAL HOUR TO SECOND Data Type




## INTERVAL MINUTE Data Type

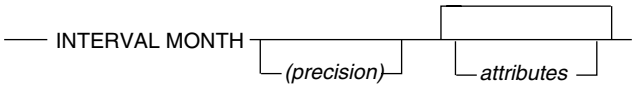


## INTERVAL MINUTE TO SECOND Data Type



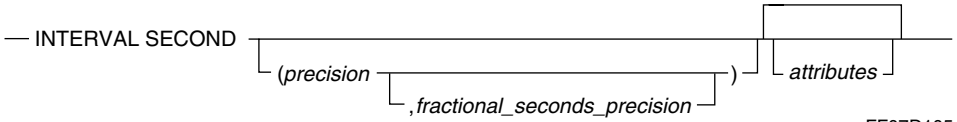

FF07D104

## INTERVAL MONTH Data Type



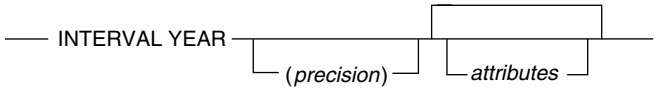

1101A013

## INTERVAL SECOND Data Type



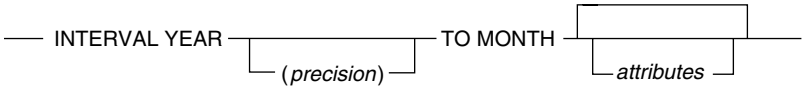

FF07D105

## INTERVAL YEAR Data Type



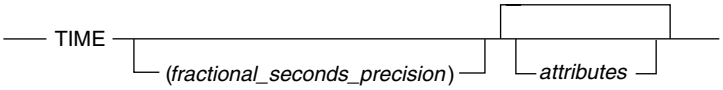

FF07D121

## INTERVAL YEAR TO MONTH Data Type



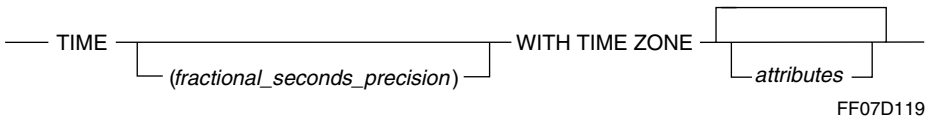

FF07D122

## TIME Data Type



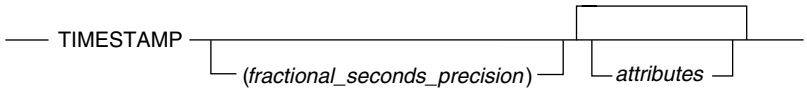

FF07D117

### TIME WITH TIME ZONE Data Type



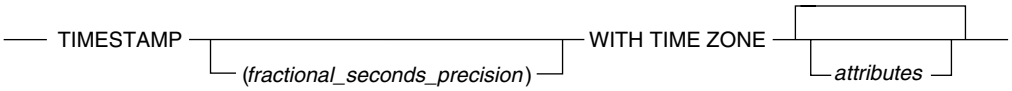

FF07D119

### TIMESTAMP Data Type




FF07D118

### TIMESTAMP WITH TIME ZONE Data Type



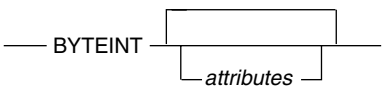

FF07D120

## Decimal/Numeric Data Types

### BIGINT Data Type

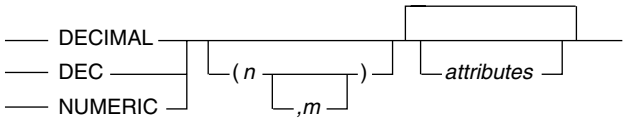

1101A393

### BYTEINT Data Type

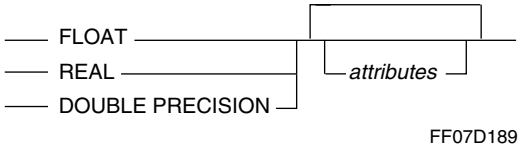



FF07D186

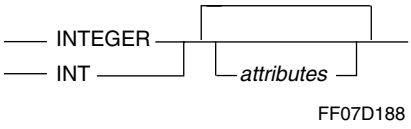
## DECIMAL/NUMERIC Data Types



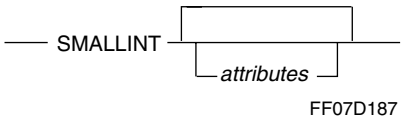

## FLOAT/REAL/DOUBLE PRECISION Data Types




## INTEGER Data Type

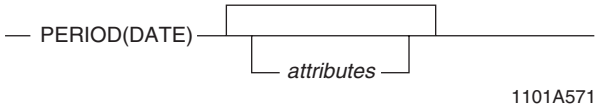



## SMALLINT Data Type

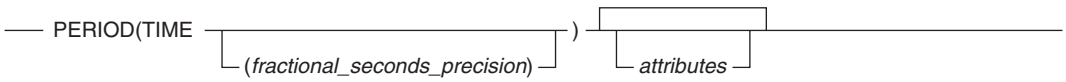



# PERIOD Data Types

## PERIOD(DATE) Data Type

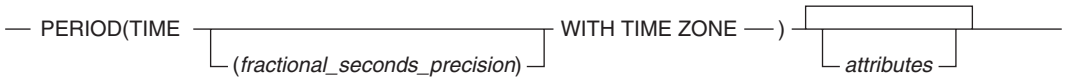


## PERIOD(TIME) Data Type



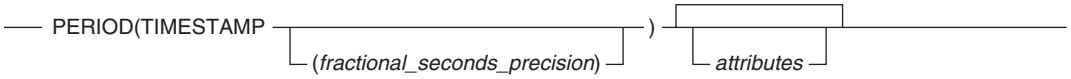

1101A617

## PERIOD(TIME WITH TIME ZONE) Data Type



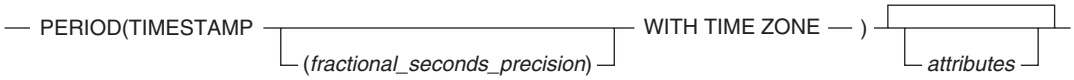

1101A618

## PERIOD(TIMESTAMP) Data Type




1101A619

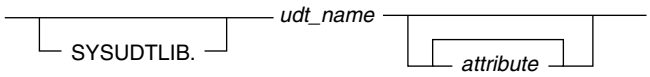
## PERIOD(TIMESTAMP WITH TIME ZONE) Data Type




1101A620

# UDT Data Types

## UDT Data Type



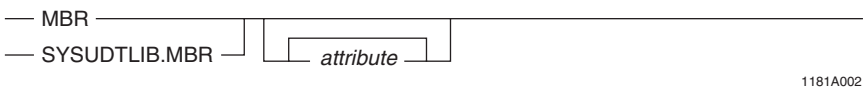

1101A366

## VARIANT_TYPE UDT

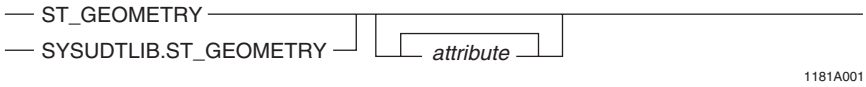

1101A575

# Geospatial Data Types

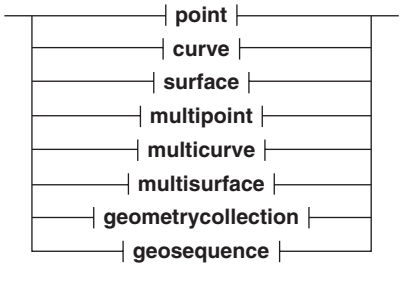
## MBR Type




## ST_Geometry Type

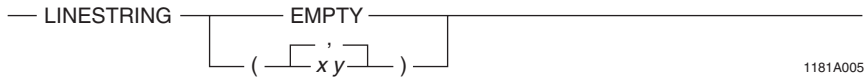



## Well-Known Text Representation

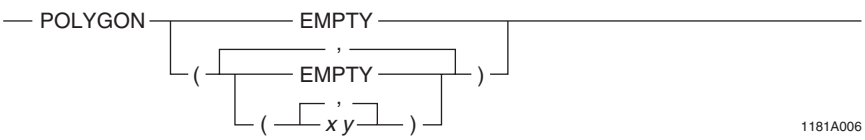



**point**

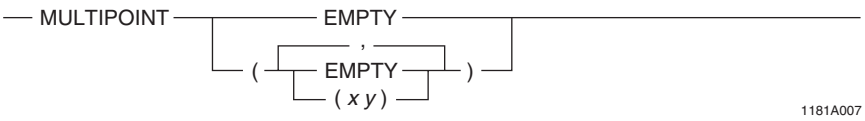

**curve**



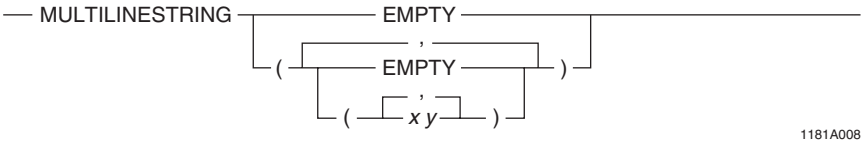
**surface**



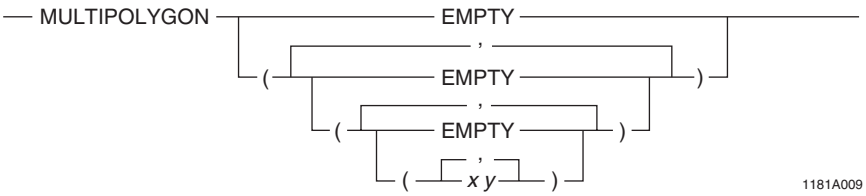

**multipoint**



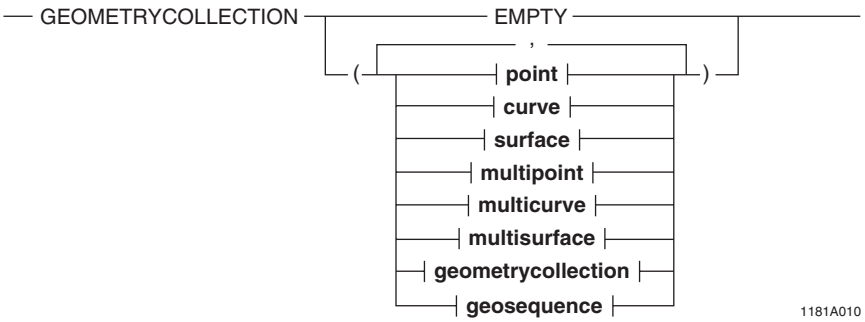

**multicurve**



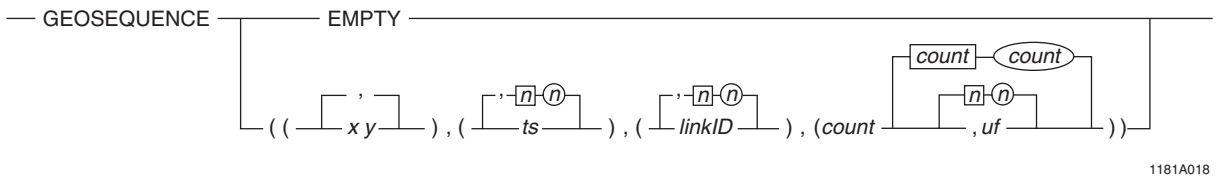

**multisurface**




**geometrycollection**

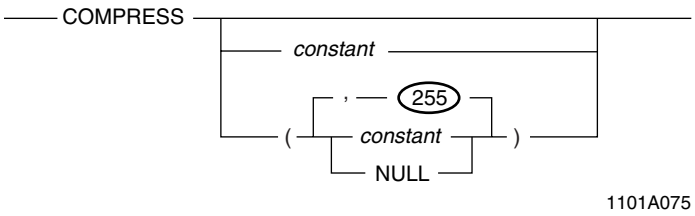


**geosequence**

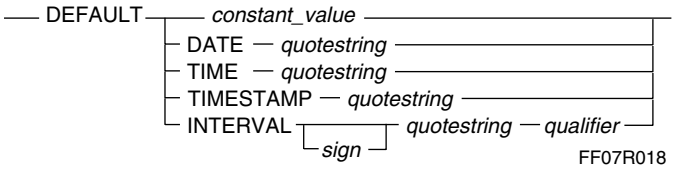



# Default Value Control Phrases

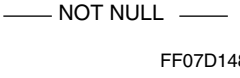
## COMPRESS Phrase



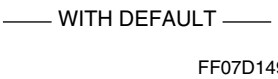

## DEFAULT Phrase




## NOT NULL Phrase

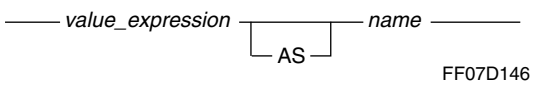



## WITH DEFAULT Phrase



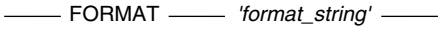
# Output Format Phrases

## AS




FF07D146

## FORMAT

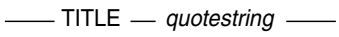



FF07D250

## NAMED


FF07D147

## TITLE

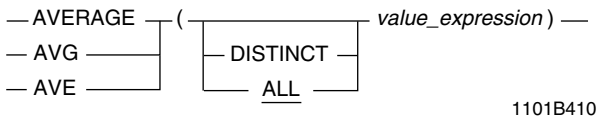



FF07D145

# CHAPTER 2 SQL Functions and Expressions

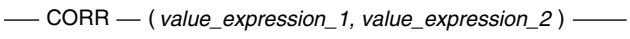
## Aggregate Functions

### AVG



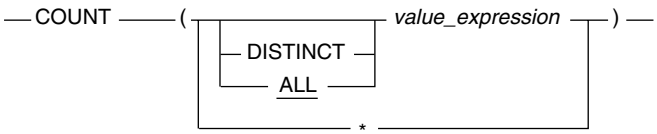

1101B410

### CORR




1101B217

### COUNT

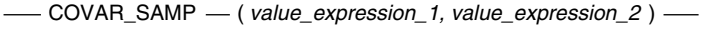



1101A411

### COVAR_POP

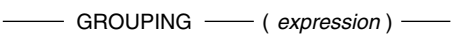

1101B216

### COVAR_SAMP



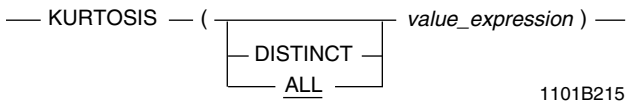

1101A456

### GROUPING

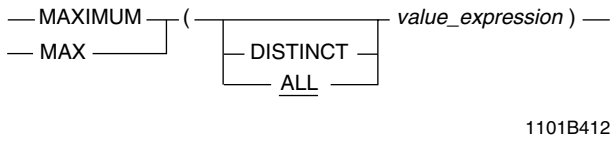



1101A461

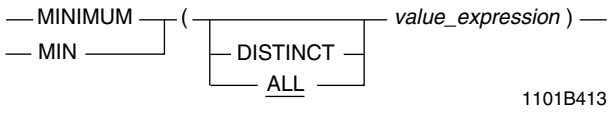
## KURTOSIS



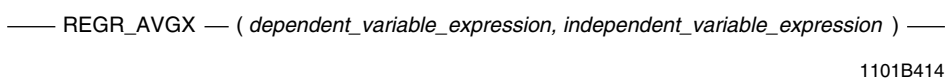

## MAX



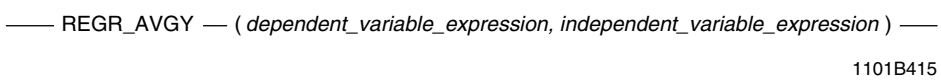

## MIN



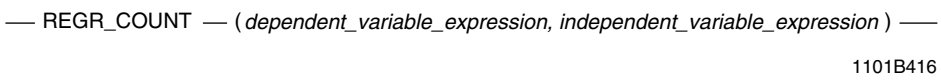

## REGR_AVGX



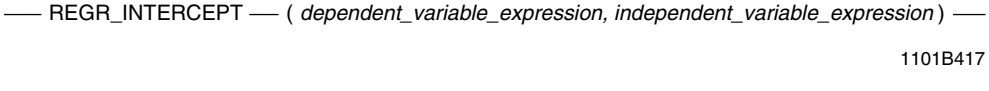

## REGR_AVGY



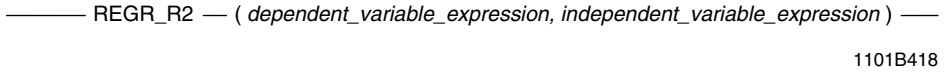

## REGR_COUNT



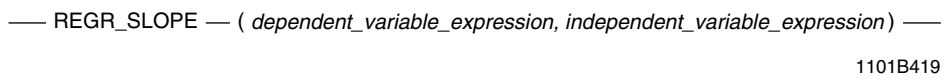

## REGR_INTERCEPT



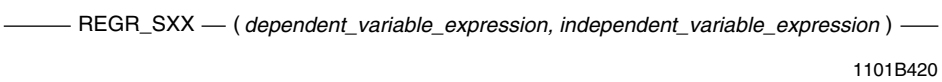

## REGR_R2



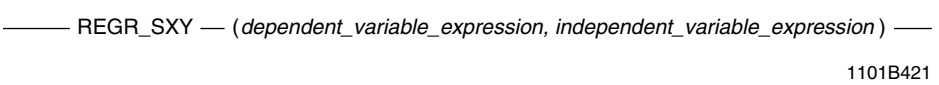
## REGR_SLOPE



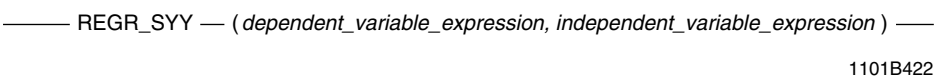

## REGR_SXX



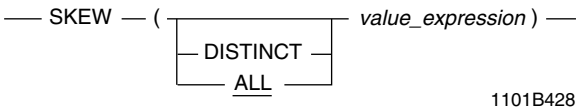

## REGR_SXY



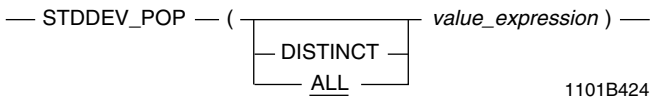

## REGR_SYY



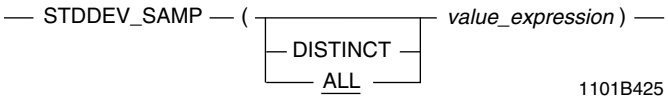

## SKEW



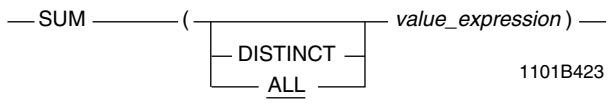

## STDDEV_POP



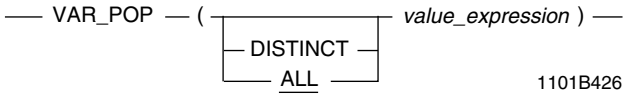

## STDDEV_SAMP



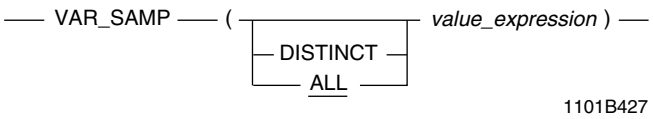

## SUM



## VAR_POP




## VAR_SAMP



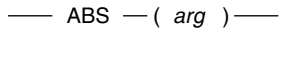

# Arithmetic Operators and Functions/ Trigonometric/Hyperbolic Functions

## Arithmetic Operators

Teradata Database supports the following arithmetic operators:

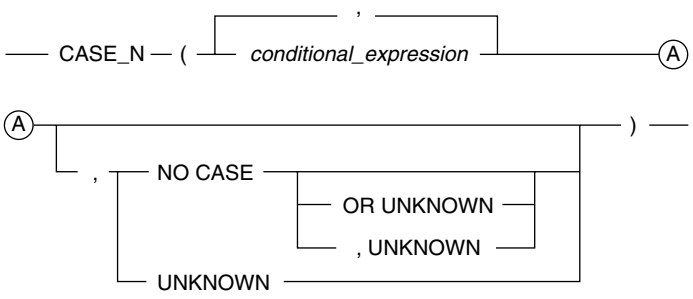
| Operator | Function |
|---|---|
| ** | Exponentiate<br>This is a Teradata extension to the ANSI SQL-99 standard. |
| * | Multiply |
| / | Divide |
| MOD | Modulo (remainder).<br>MOD calculates the remainder in a division operation.<br>For example, 60 MOD 7 = 4: 60 divided by 7 equals 8, with a remainder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

## ABS



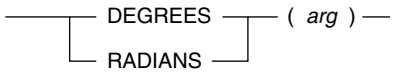

1101A480

## CASE_N




1101A069

## DEGREES/ RADIANS

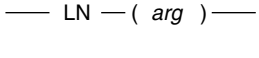



1101A481

## EXP

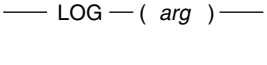

1101A484

## LN



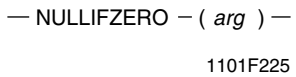

1101A485

## LOG

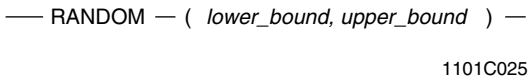



1101A486

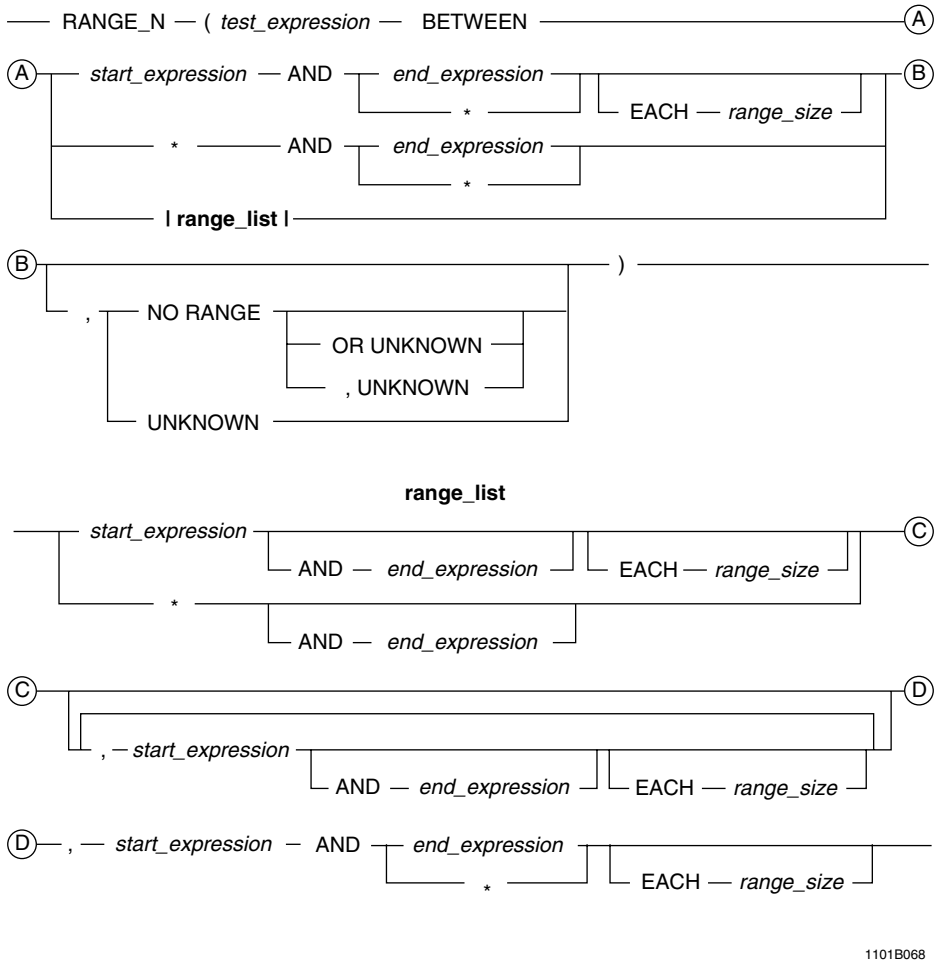
## NULLIFZERO



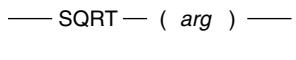

## RANDOM



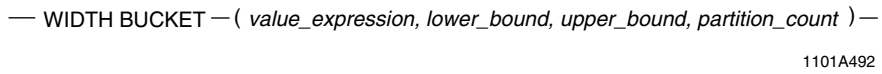

## RANGE_N



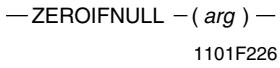

## SQRT



### WIDTH_BUCKET

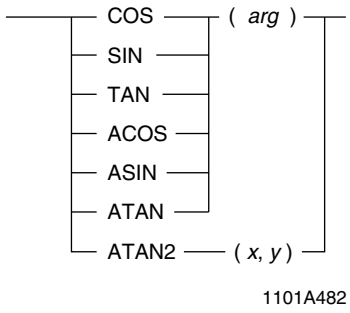



### ZEROIFNULL



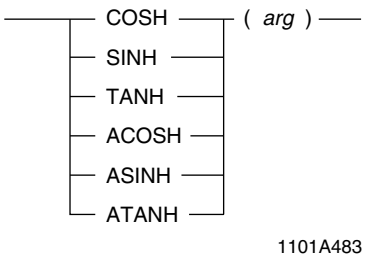

# Trigonometric Functions

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



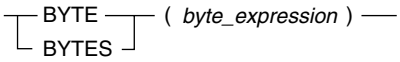

# Hyperbolic Functions

### COSH, SINH, TANH, ACOSH, ASINH, ATANH

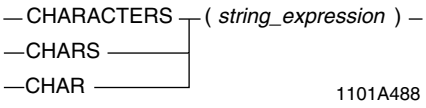


# Attribute Functions

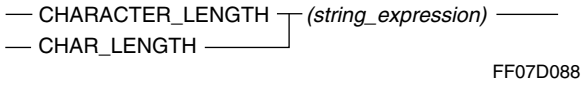
## BYTES



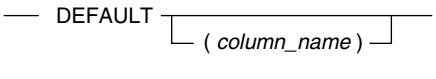

## CHARACTERS



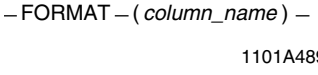

## CHARACTER_LENGTH



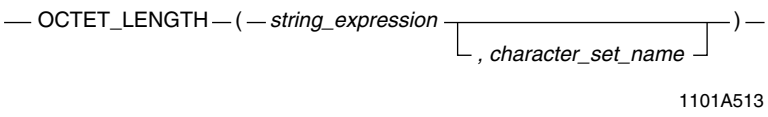

## DEFAULT



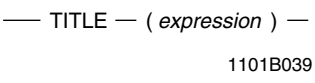

## FORMAT



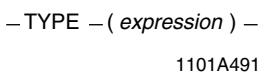

## OCTET_LENGTH




## TITLE

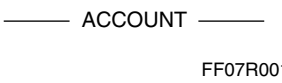


### TYPE

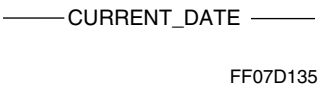



# Built-In Functions

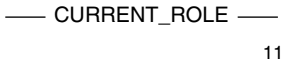
## ACCOUNT



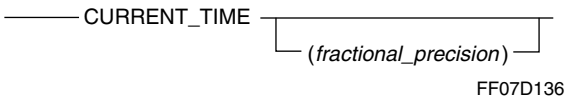

## CURRENT_DATE



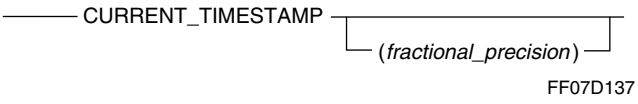

## CURRENT_ROLE



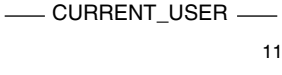

## CURRENT_TIME



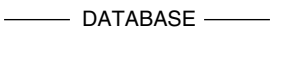

## CURRENT_TIMESTAMP




## CURRENT_USER

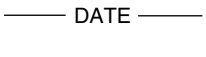


## DATABASE



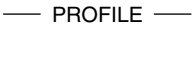

FF07R002

## DATE



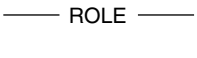

FF07D134

## PROFILE



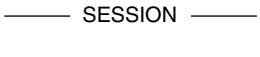

KZ01A006

## ROLE



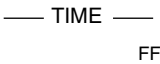

KZ01A007

## SESSION



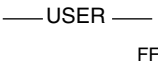

FF07R003

## TIME




FF07D271

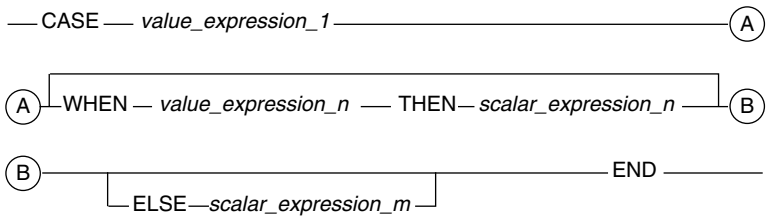
## USER




FF07D272

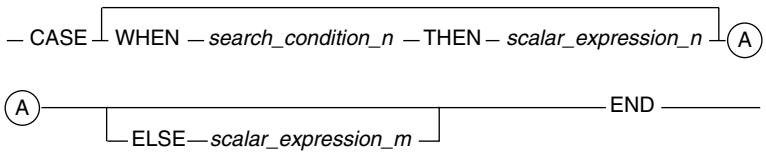
# CASE Expressions

## Valued CASE Expression



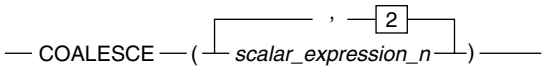

1101A012

## Searched CASE Expression



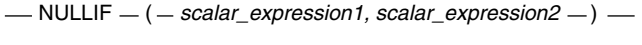

FF07D224

## COALESCE Expression




1101E227

## NULLIF Expression



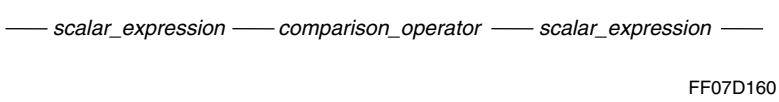

HH01B094

# Comparison Operators

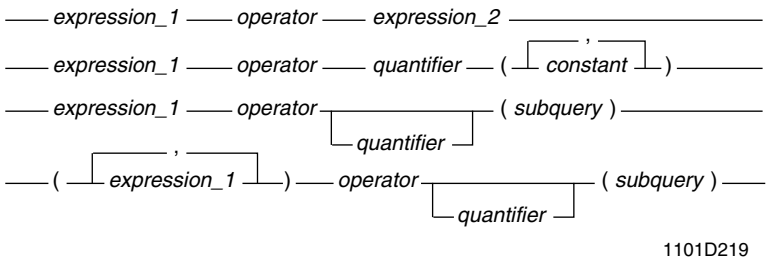
Teradata Database supports the following comparison operators:

| ANSI Operator | Teradata Extensions | Function |
|---|---|---|
| = | EQ | Tests for equality. |
| <> | ^=<br>NE<br>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

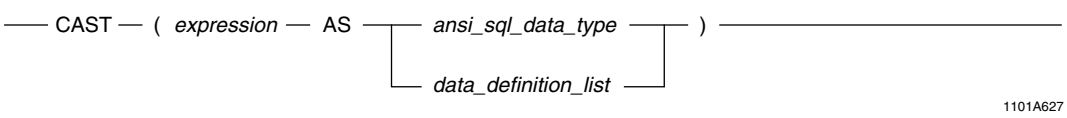



## Comparison Operators in Logical Expressions

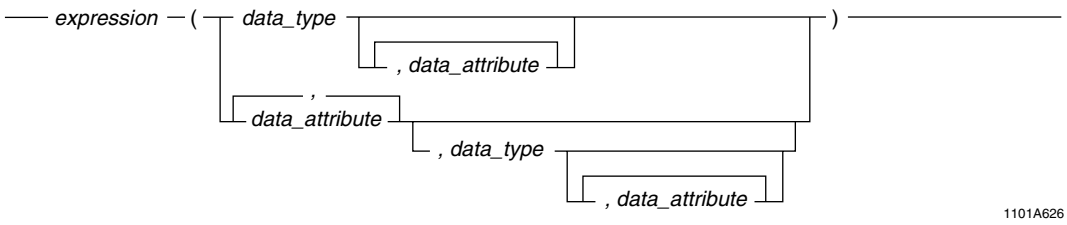


# Data Type Conversions

## CAST

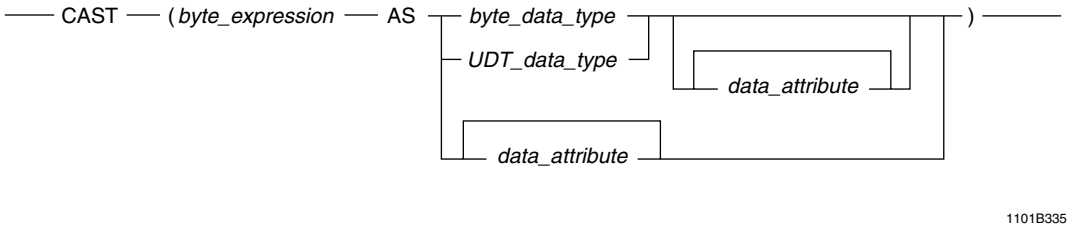



## Teradata Conversion Syntax

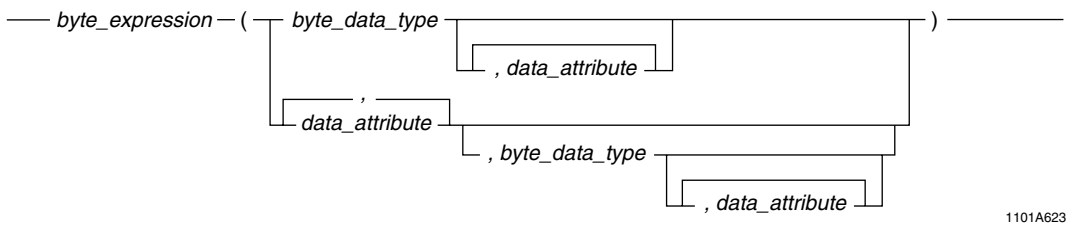



# Byte Conversion

## CAST

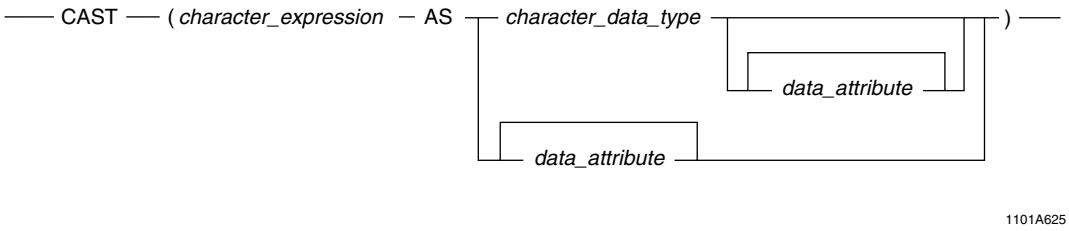


### Teradata Conversion

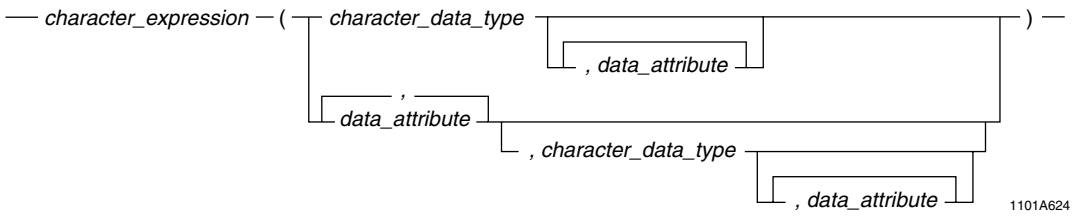



# Character-to-Character Conversion

## CAST

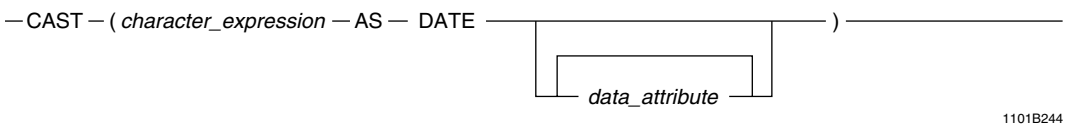



### Teradata Conversion

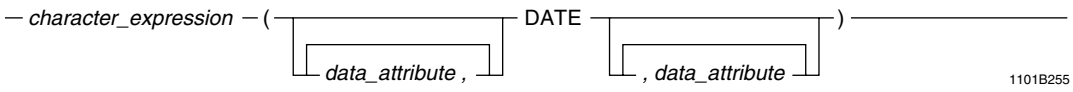


# Character-to-DATE Conversion

## CAST

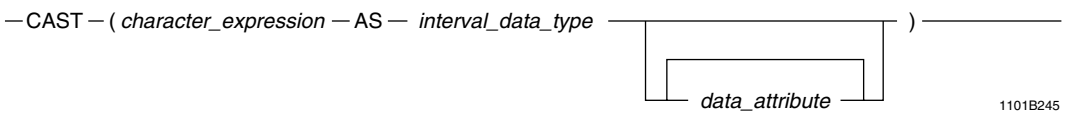



## Teradata Conversion

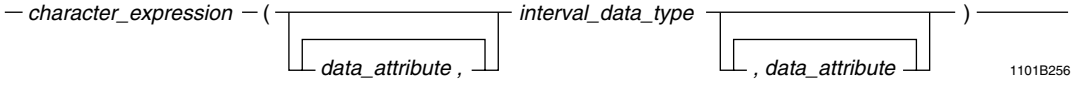



# Character-to-INTERVAL Conversion

## CAST

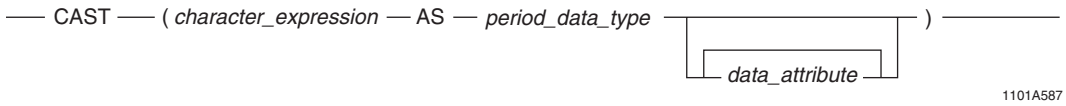



## Teradata Conversion



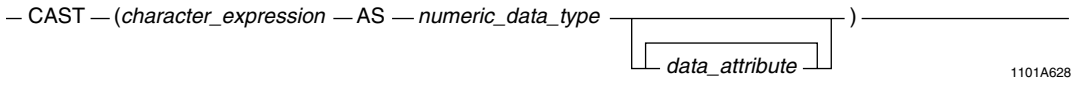
# Character-to-Period Conversion

## CAST

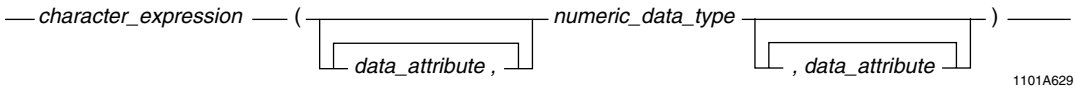



# Character-to-Numeric Conversion

## CAST

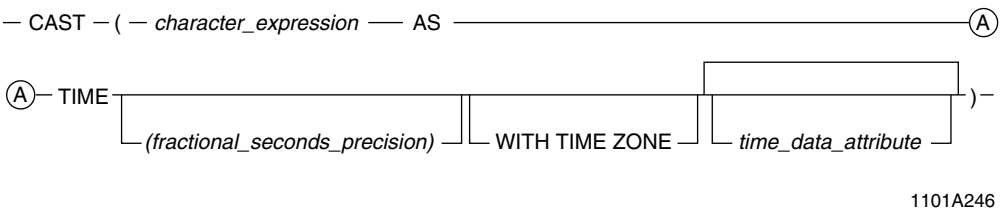



## Teradata

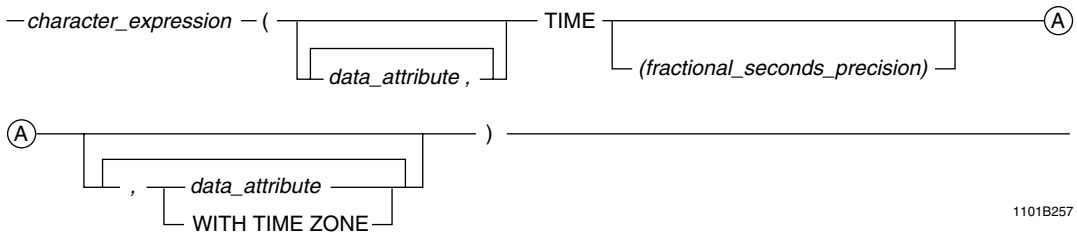



# Character-to-TIME Conversion

## CAST

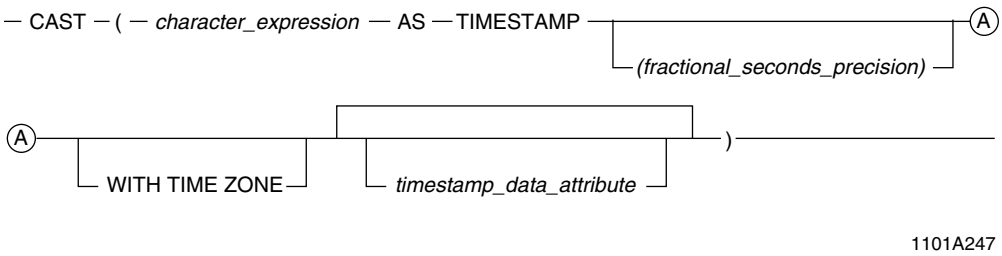


### Teradata Conversion

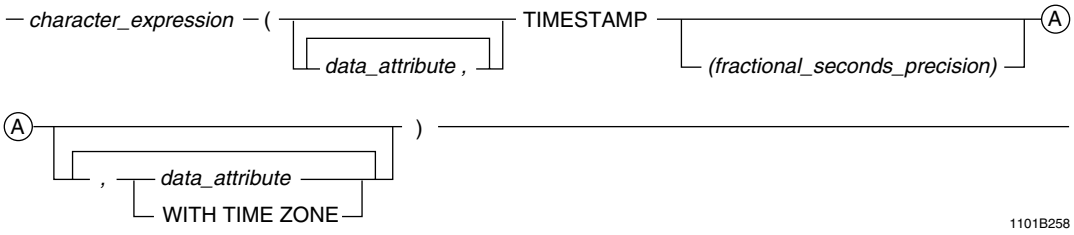



# Character-to-TIMESTAMP Conversion

## CAST

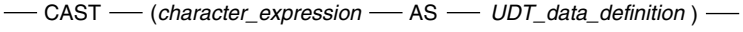



## Teradata Conversion



# Character-to-UDT Conversion

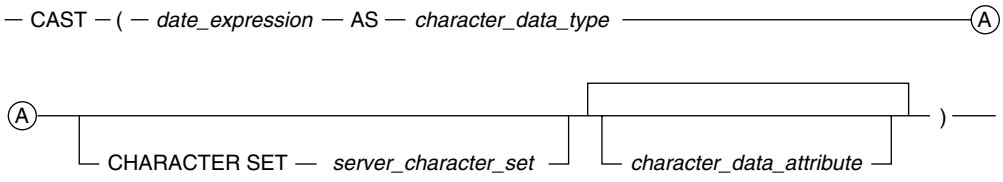
## CAST




1101A336

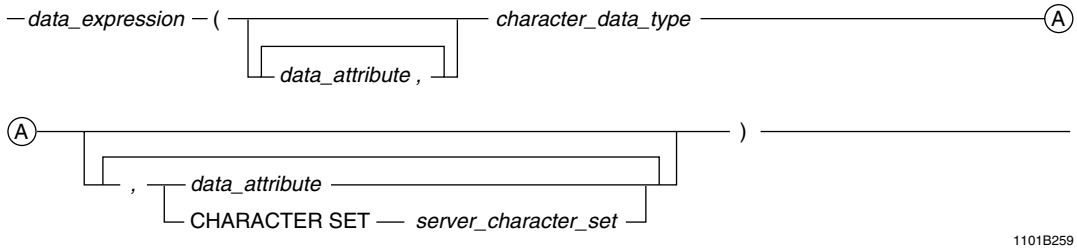
# DATE-to-Character Conversion

## CAST




1101A248

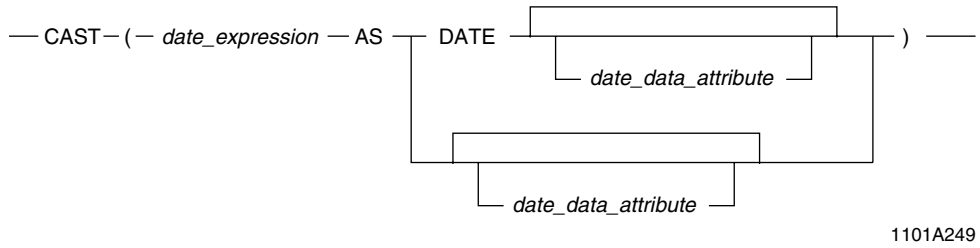
## Teradata Conversion



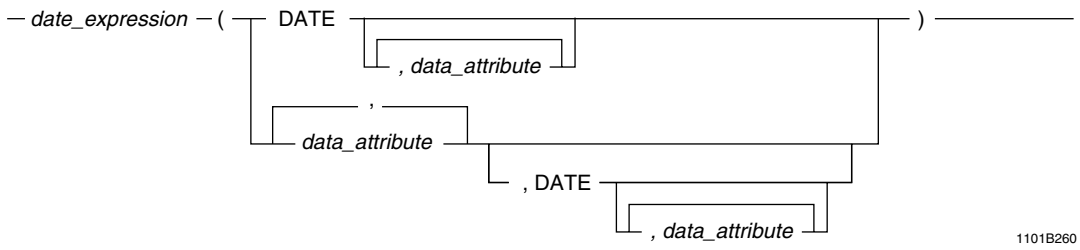

1101B259

# DATE-to-DATE Conversion

## CAST

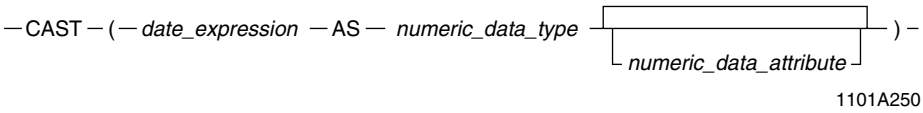



## Teradata Conversion

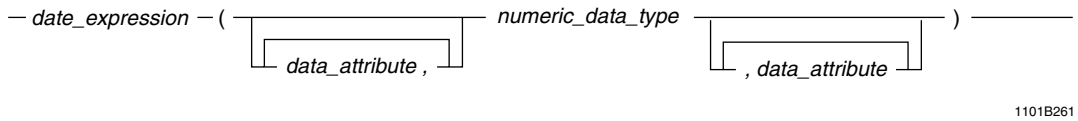



# DATE-to-Numeric Conversion

## CAST

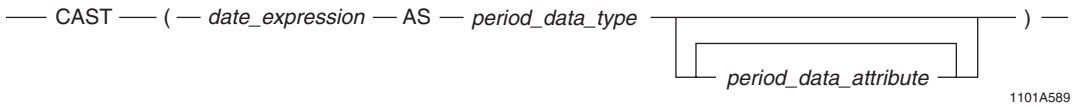


### Teradata Conversion



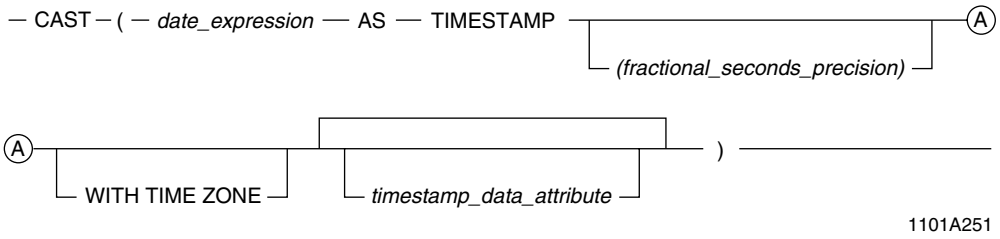

# DATE-to-Period Conversion

### CAST

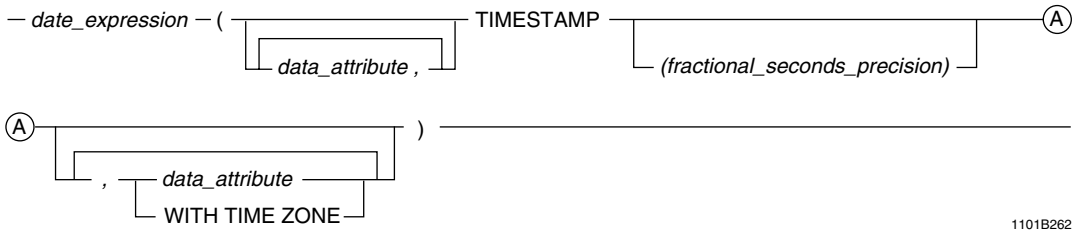



# DATE-to-TIMESTAMP Conversion

### CAST

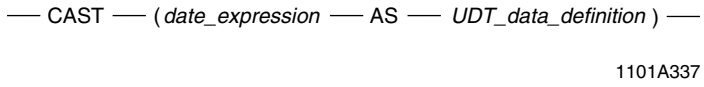


### Teradata Conversion



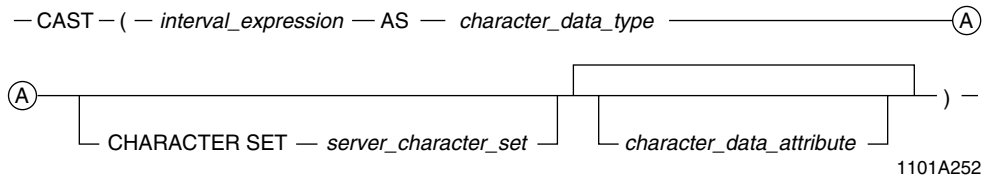

# DATE-to-UDT Conversion

### CAST

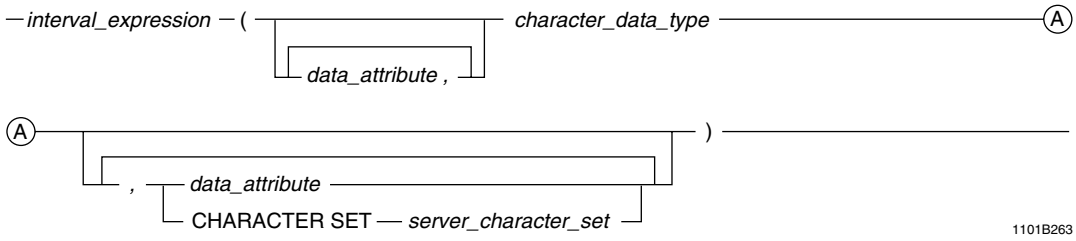



# INTERVAL-to-Character Conversion

### CAST

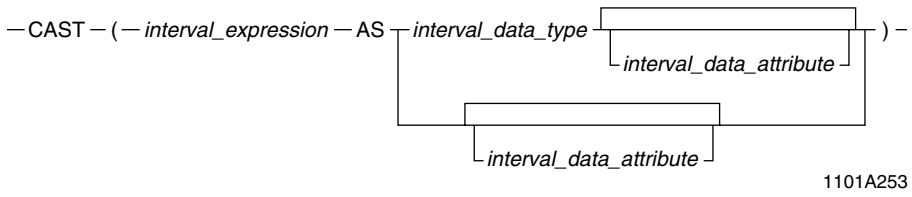


## Teradata Conversion

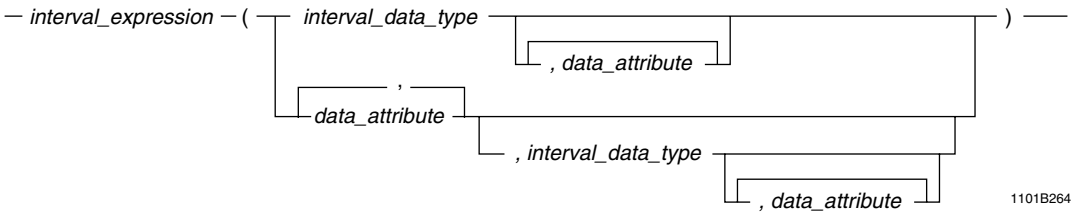



# INTERVAL-to-INTERVAL Conversion

## CAST

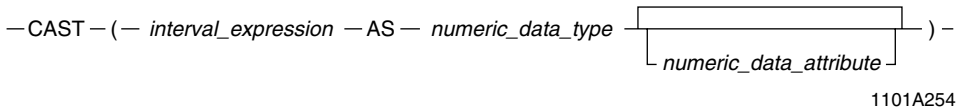



## Teradata Conversion

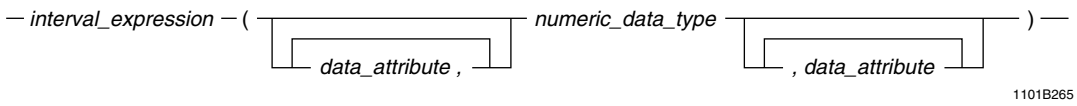


# INTERVAL-to-Numeric Conversion

## CAST

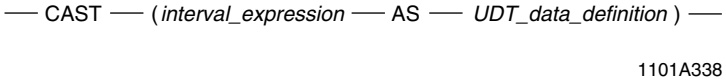



## Teradata Conversion



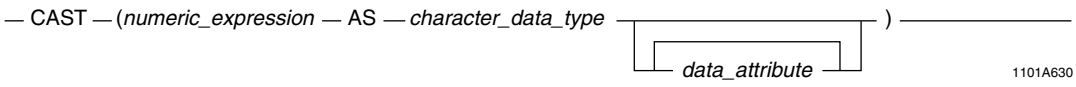

# INTERVAL-to-UDT Conversion

## CAST

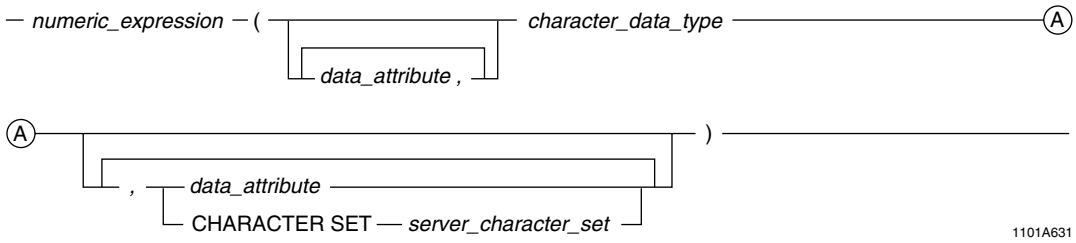



# Numeric-to-Character Conversion

## CAST

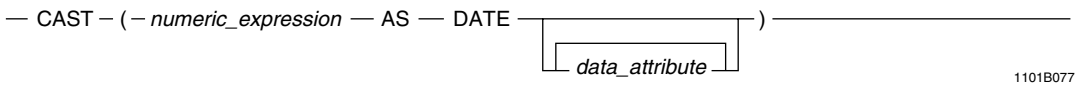


### Teradata Conversion

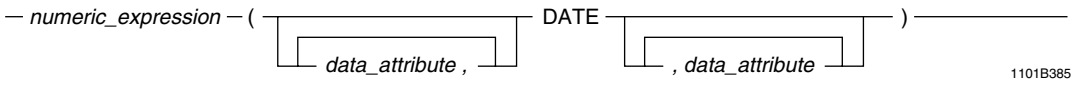



# Numeric-to-DATE Conversion

## CAST

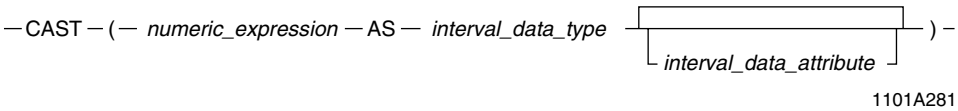



## Teradata Conversion

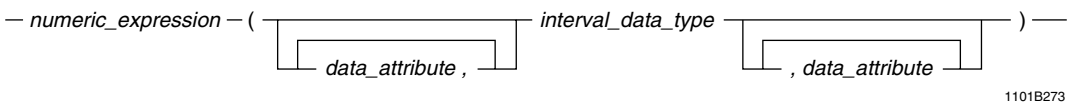



# Numeric-to-INTERVAL Conversion

## CAST

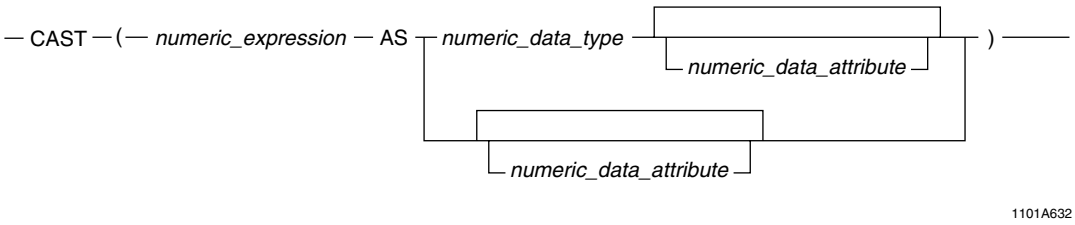


### Teradata Conversion

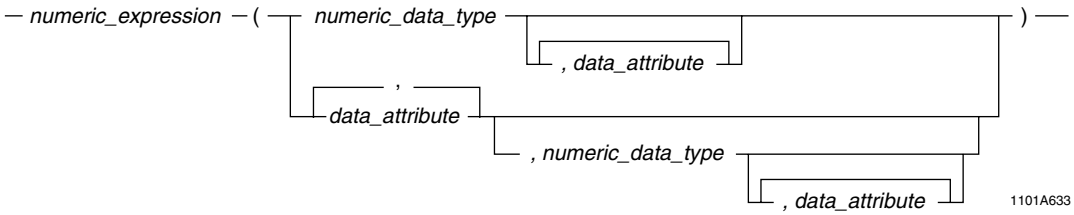



# Numeric-to-Numeric Conversion

### CAST

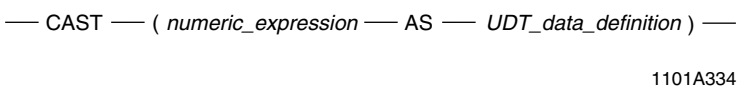



### Teradata Conversion



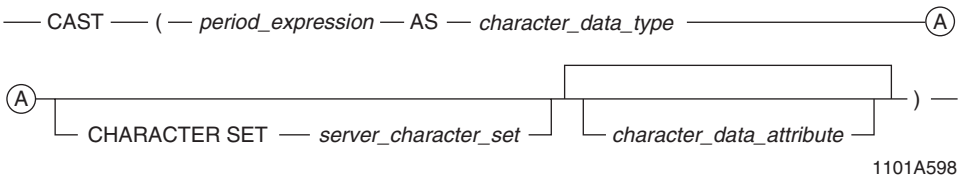

# Numeric-to-UDT Conversion

### CAST



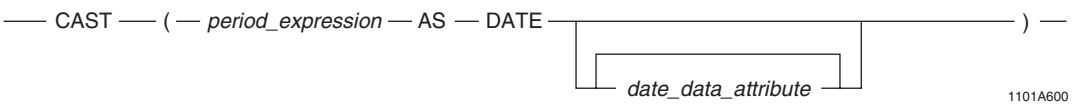
# Period-to-Character Conversion

## CAST



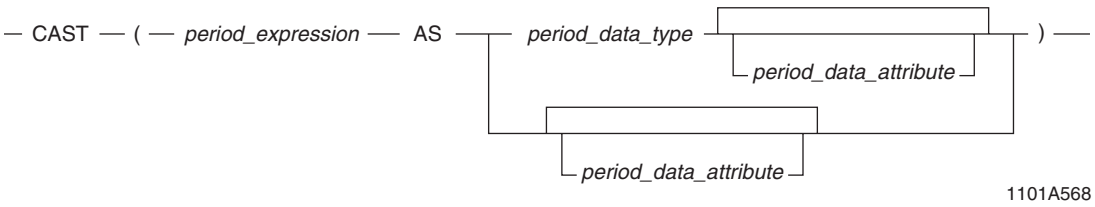

# Period-to-DATE Conversion

## CAST



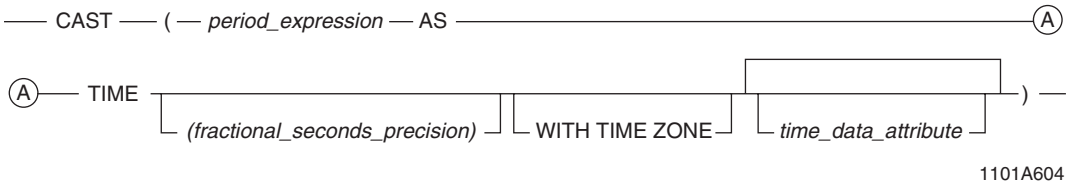

# Period-to-Period Conversion

## CAST



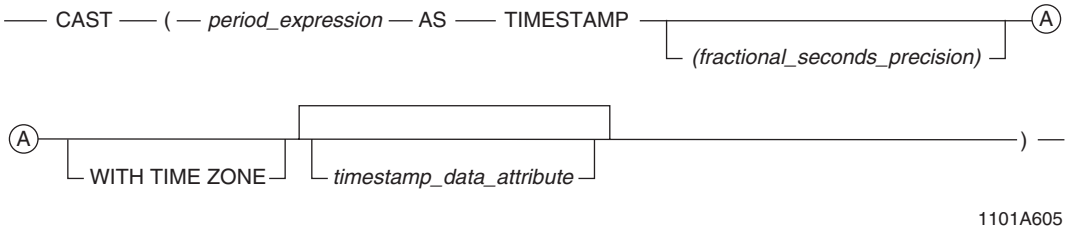
# Period-to-TIME Conversion

## CAST



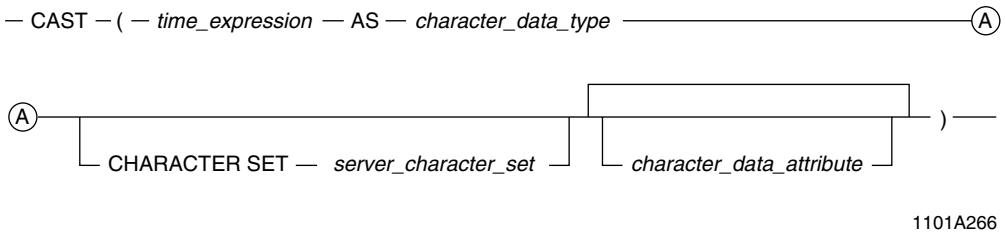

# Period-to-TIMESTAMP Conversion

## CAST

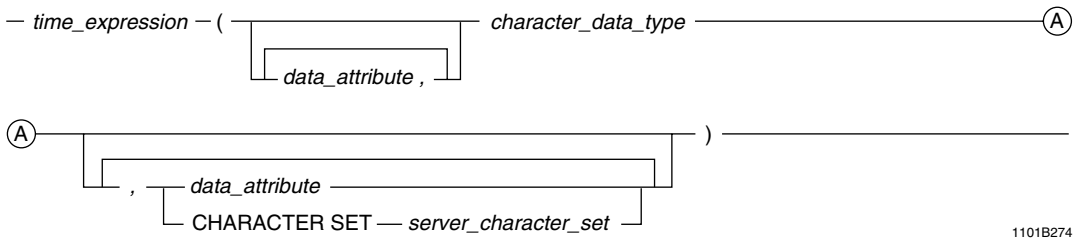



# TIME-to-Character Conversion

## CAST

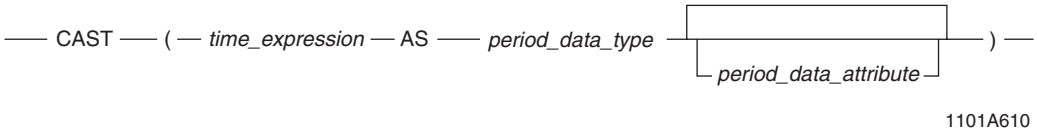


### Teradata Conversion



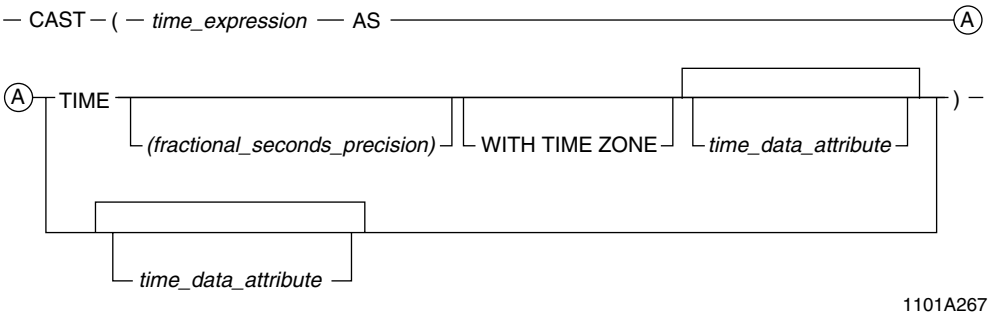

# TIME-to-Period Conversion

## CAST

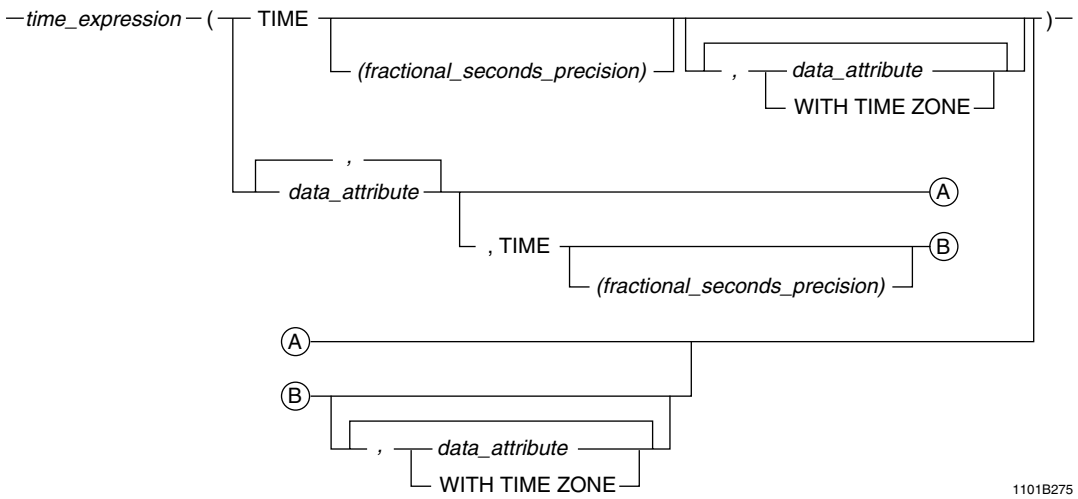



# TIME-to-TIME Conversion

## CAST

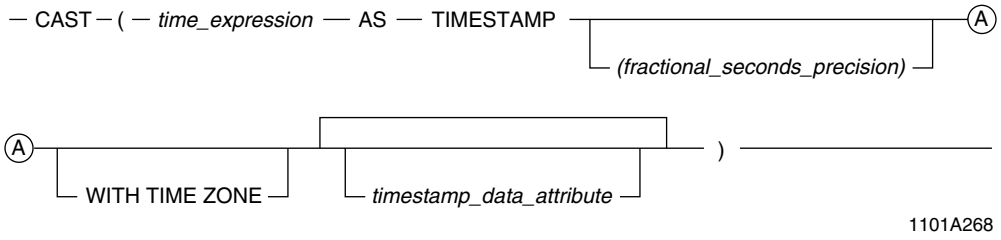


## Teradata Conversion

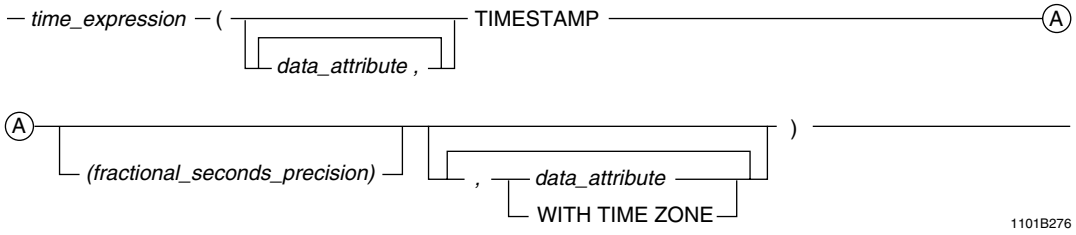



# TIME-to-TIMESTAMP Conversion

## CAST

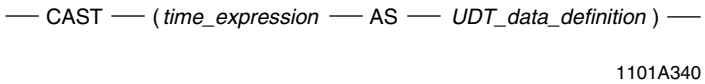



## Teradata Conversion



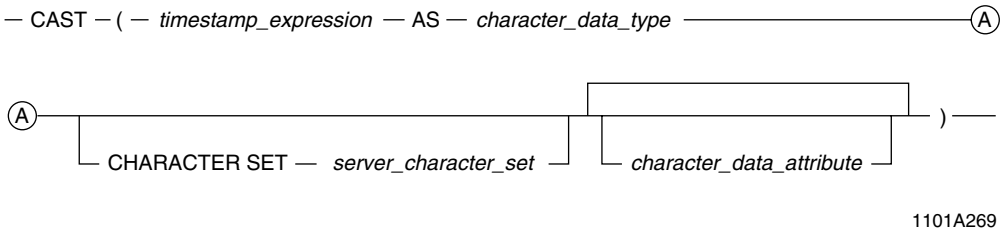
# TIME-to-UDT Conversion

## CAST

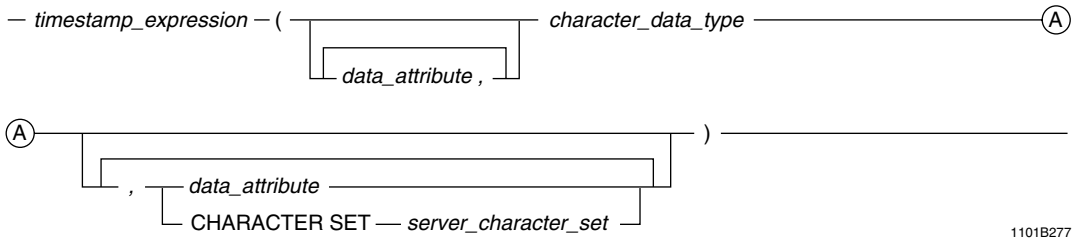



# TIMESTAMP-to-Character Conversion

## CAST

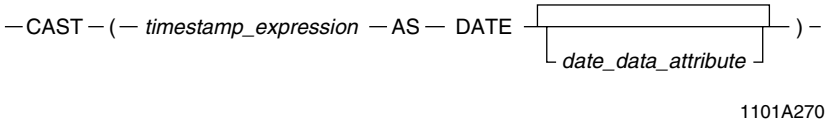



## Teradata Conversion



# TIMESTAMP-to-DATE Conversion

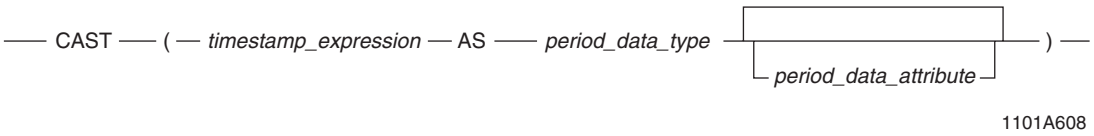
## CAST




## Teradata Conversion

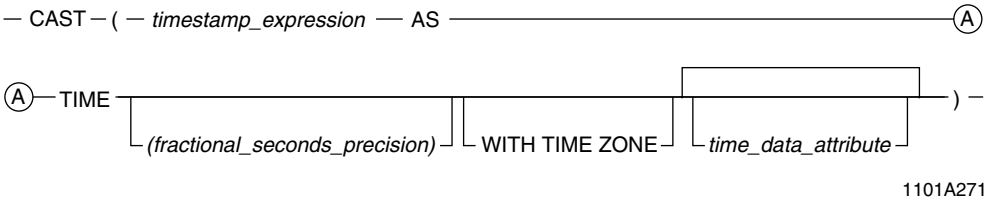

# TIMESTAMP-to-Period Conversion

## CAST

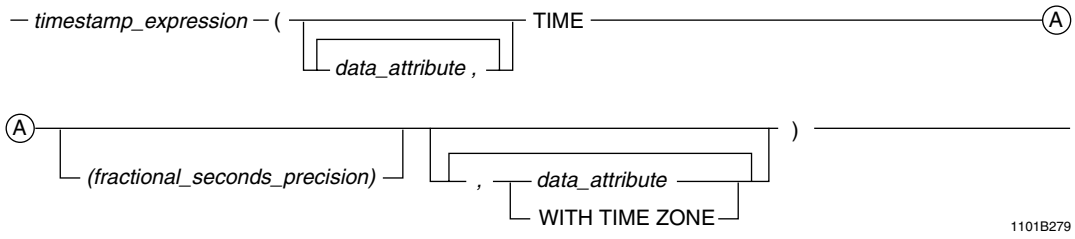



# TIMESTAMP-to-TIME Conversion

## CAST

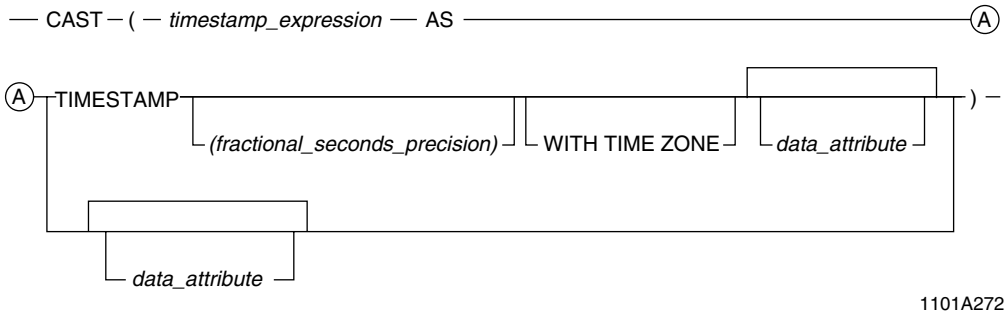


### Teradata Conversion

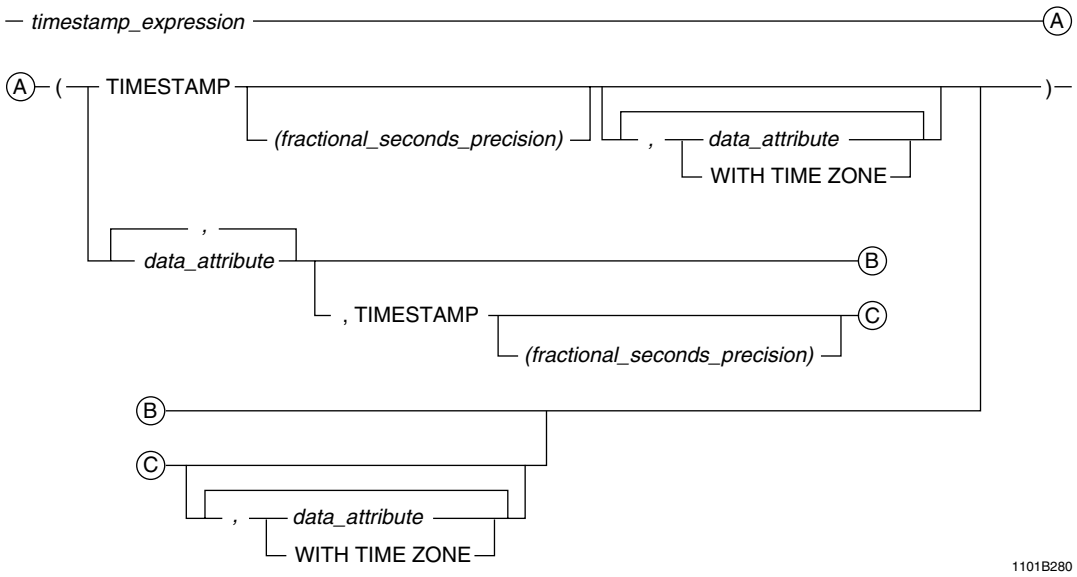



# TIMESTAMP-to-TIMESTAMP Conversion

### CAST

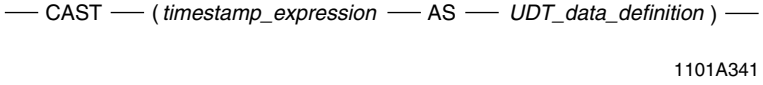


### Teradata Conversion



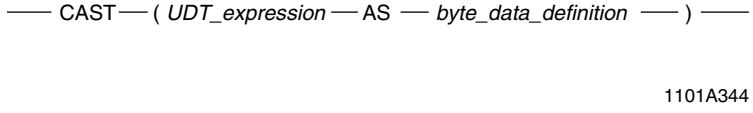

# TIMESTAMP-to-UDT Conversion

### CAST

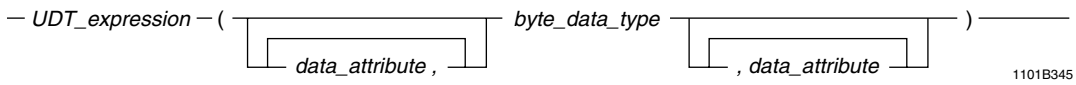



# UDT-to-Byte Conversion

### CAST

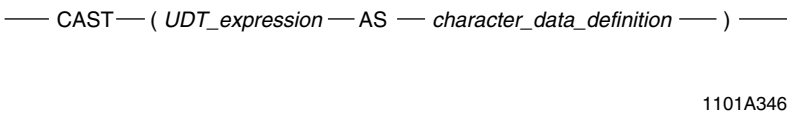


### Teradata Conversion

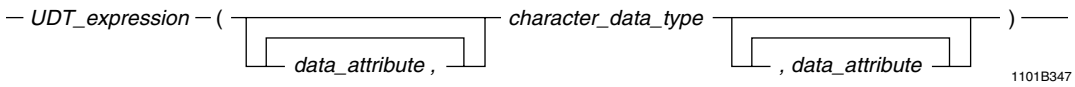



# UDT-to-Character Conversion

### CAST

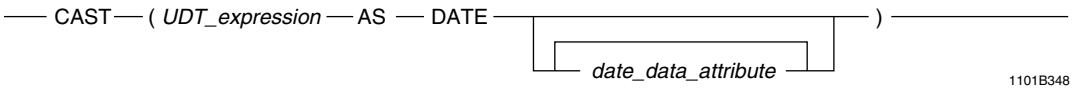



### Teradata Conversion

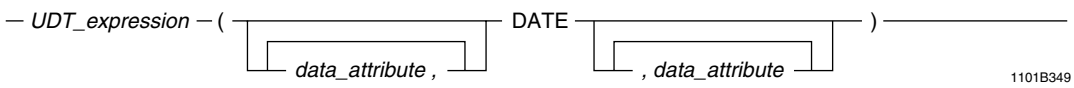



# UDT-to-DATE Conversion

### CAST

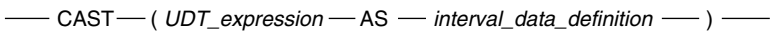


### Teradata Conversion



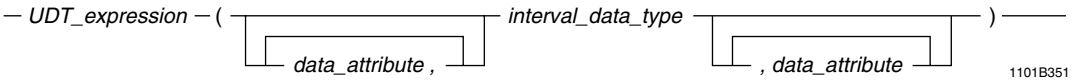

# UDT-to-INTERVAL Conversion

### CAST



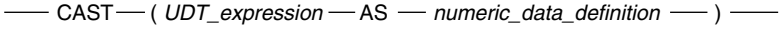

1101A350

### Teradata Conversion



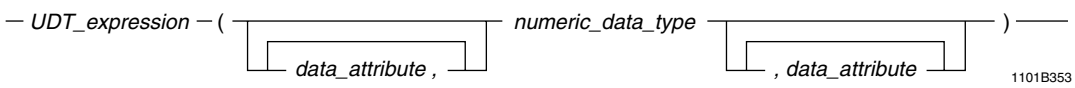

# UDT-to-Numeric Conversion

### CAST



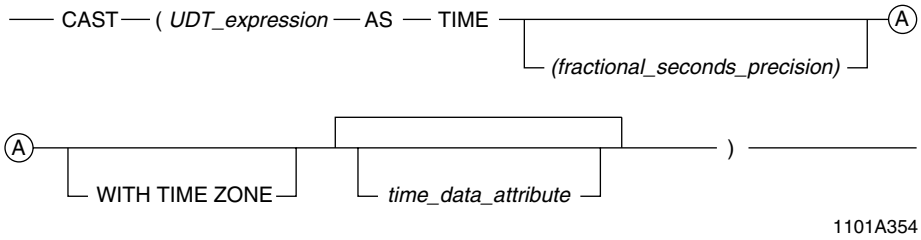

1101A352

### Teradata Conversion

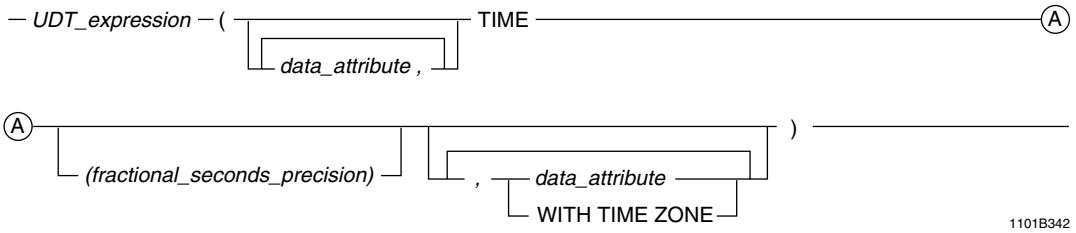



# UDT-to-TIME Conversion

## CAST

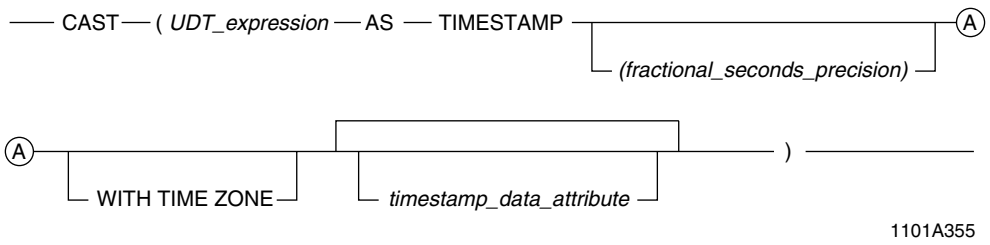



## Teradata Conversion

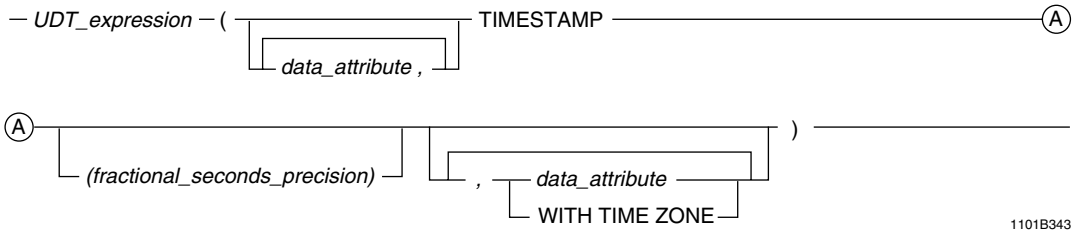


# UDT-to-TIMESTAMP Conversion

## CAST

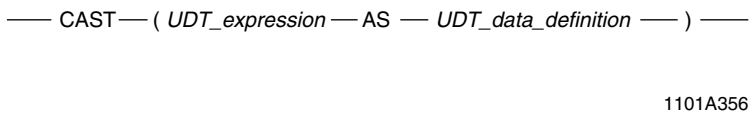



## Teradata Conversion




# UDT-to-UDT Conversion

## CAST




# DateTime and Interval Functions and Expressions

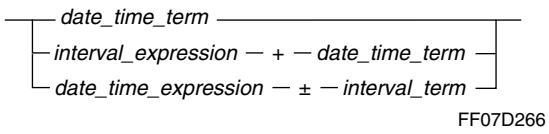
## Arithmetic Operators and Result Types

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

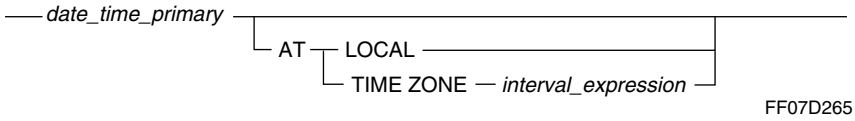
| First Value Type | Operator | Second Value Type | Result Type |
|---|---|---|---|
| DateTime | - | DateTime | Interval |
| DateTime | + | Interval | DateTime |
| DateTime | - | Interval | DateTime |
| Interval | + | DateTime | DateTime |
| Interval | + | Interval | Interval |
| Interval | - | Interval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

## ANSI DateTime Expressions

*date_time_expression* **Syntax**

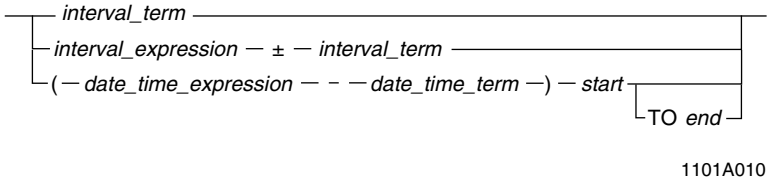



*date_time_term* **Syntax**

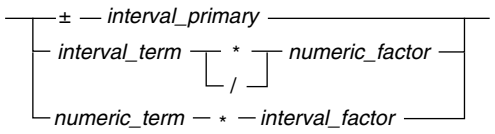



## ANSI Interval Expressions

*interval_expression* **Syntax**

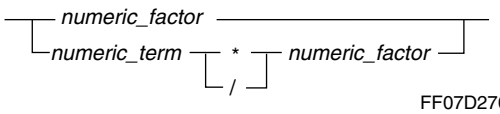


*interval_term* **Syntax**



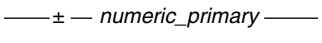

FF07D268

*numeric_term* **Syntax**



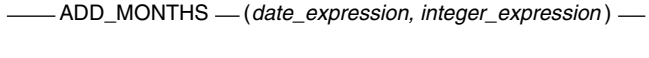

FF07D270

*numeric_factor* **Syntax**



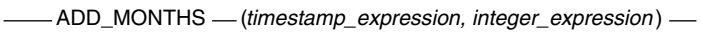

FF07D269

## ADD_MONTHS (DATE Syntax)



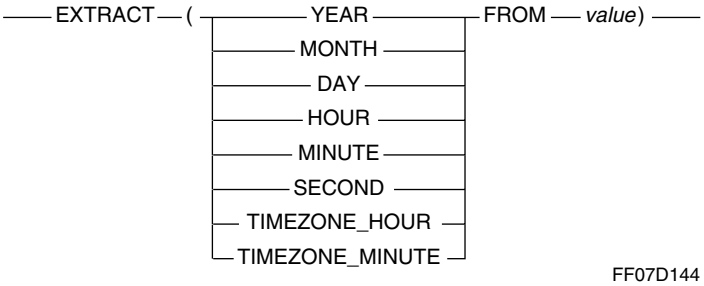

FF07D202

## ADD_MONTHS (TIMESTAMP Syntax)




FF07D208

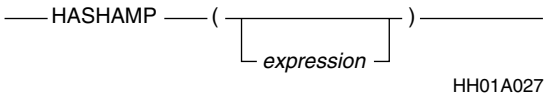
## EXTRACT



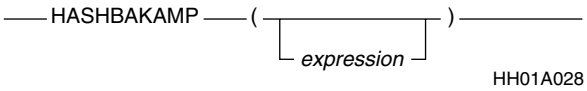

FF07D144

# Hash-Related Functions

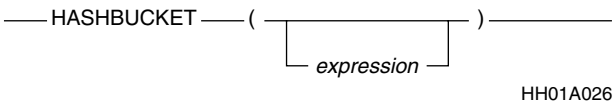
## HASHAMP



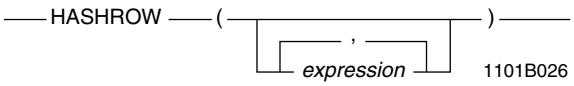

## HASHBAKAMP




## HASHBUCKET

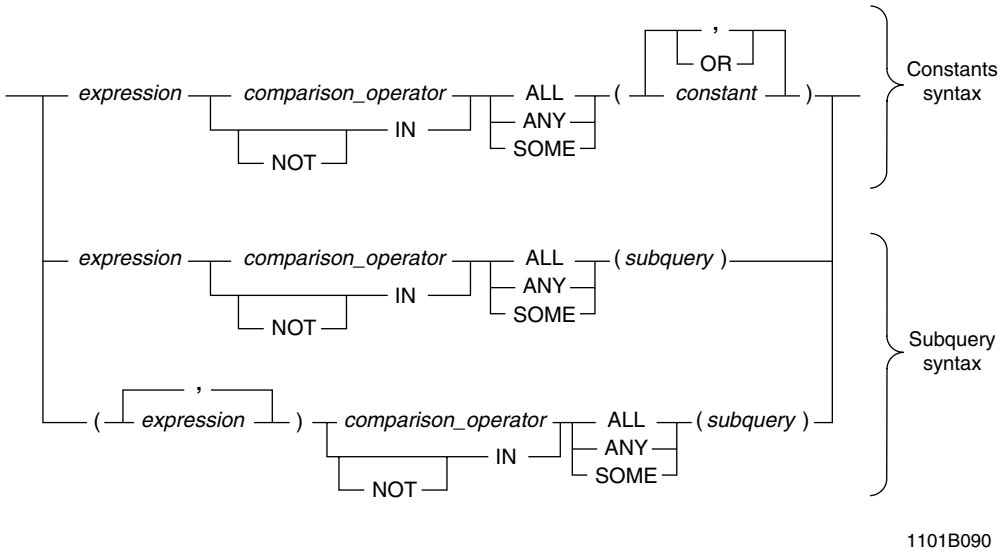



## HASHROW

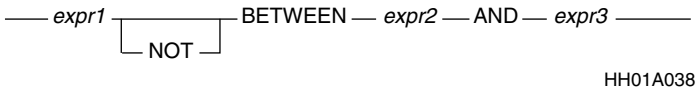


# Logical Predicates

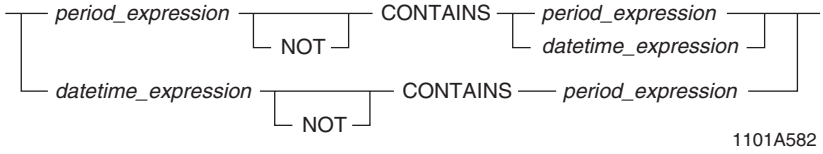
## ANY/ALL/SOME Quantifiers



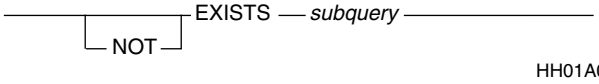

## BETWEEN/NOT BETWEEN



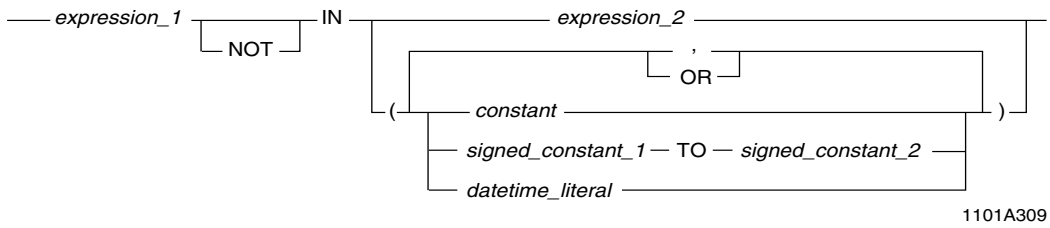

## CONTAINS



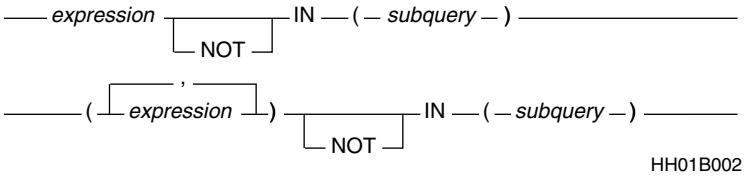

## EXISTS/NOT EXISTS



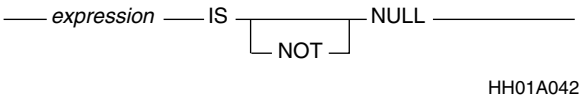
## IN/NOT IN (Syntax 1)




## IN/NOT IN (Syntax 2)

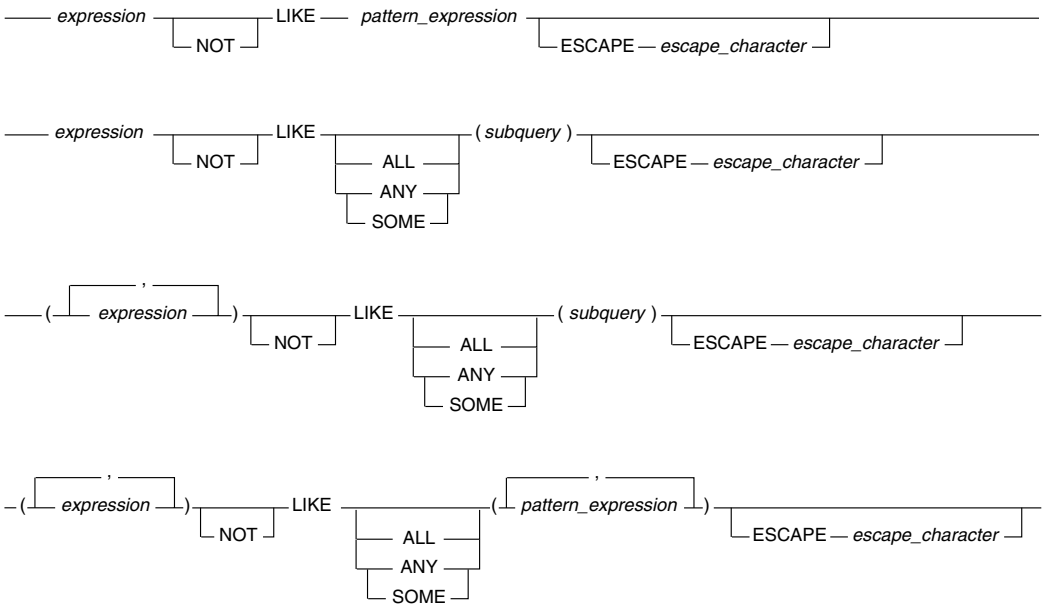



## IS NULL/IS NOT NULL



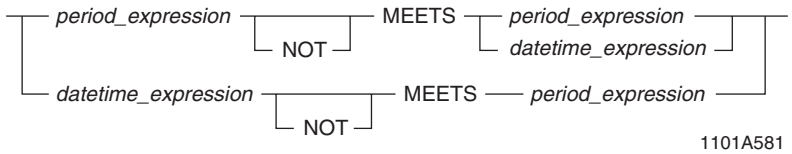

## IS UNTIL_CHANGED/IS NOT UNTIL_CHANGED

## LIKE



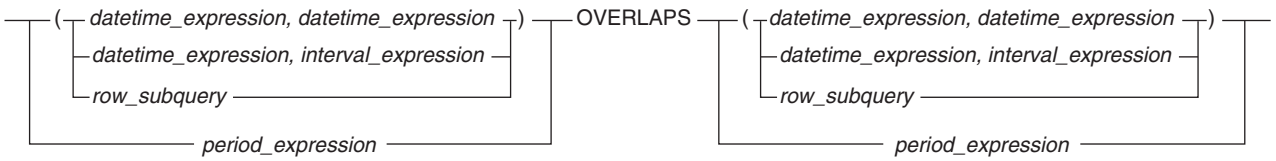

FF07D196

## MEETS



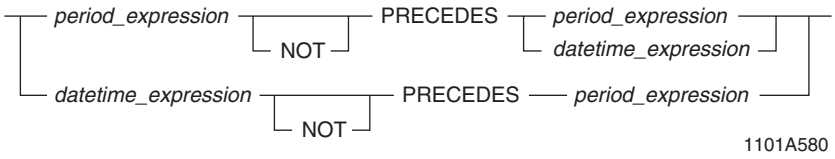

1101A581

## OVERLAPS

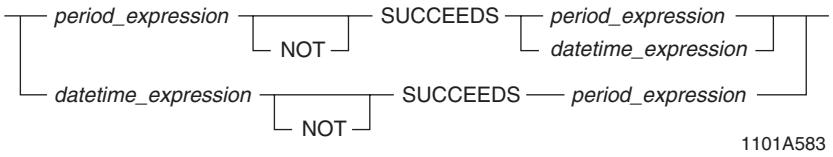



1101A612

## PRECEDES



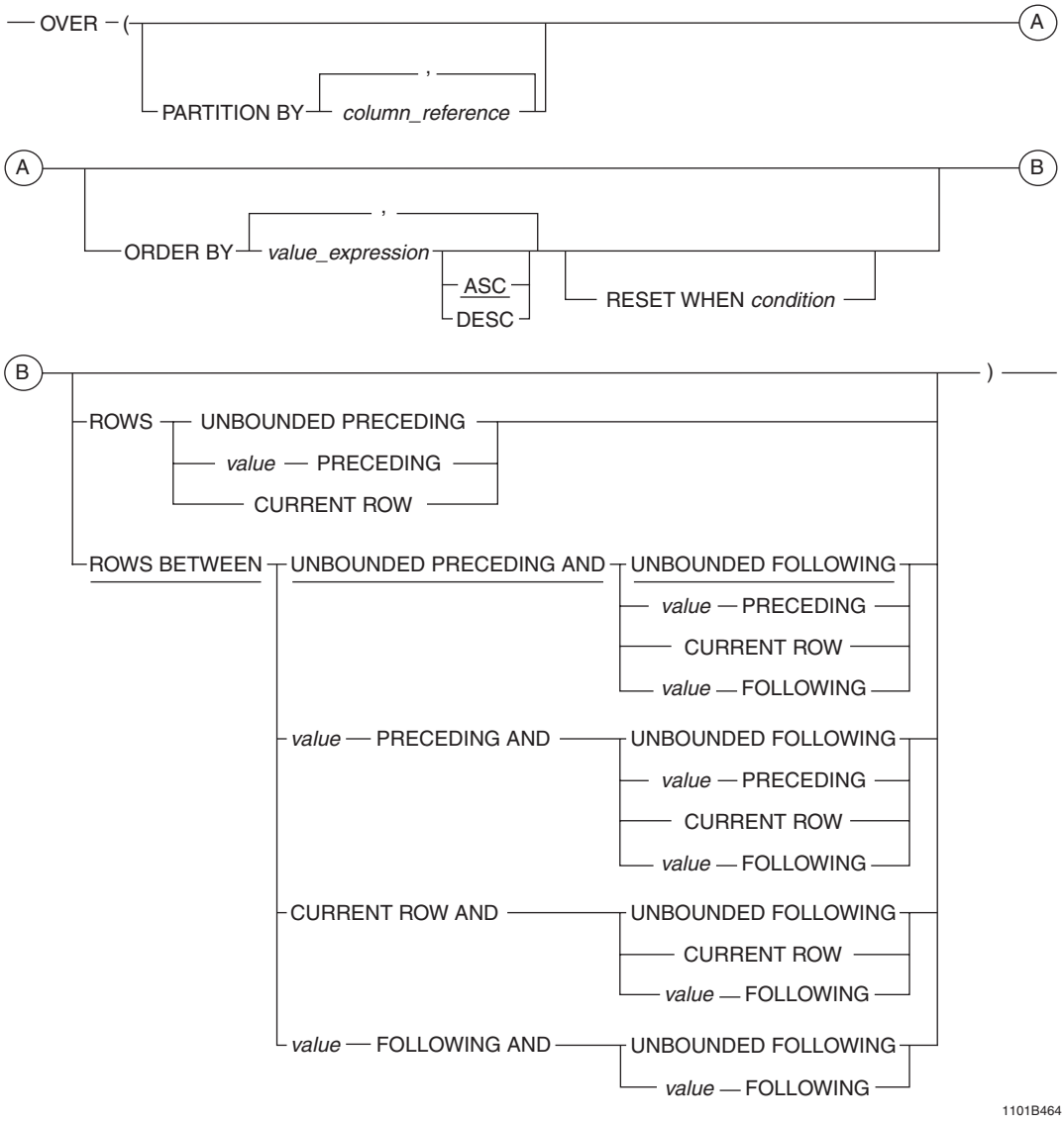

## SUCCEEDS



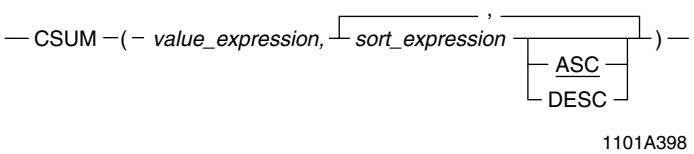
# Ordered Analytical Functions

## 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)

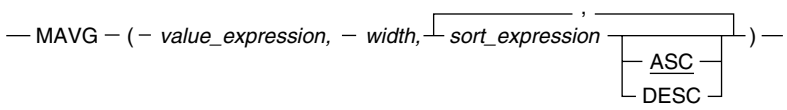
**window**


OVER (
PARTITION BY
,
column_reference
A
ORDER BY
value_expression
ASC
DESC
RESET WHEN condition
B
)
ROWS
UNBOUNDED PRECEDING
value PRECEDING
CURRENT ROW
ROWS BETWEEN
UNBOUNDED PRECEDING AND
UNBOUNDED FOLLOWING
value PRECEDING
CURRENT ROW
value FOLLOWING
value PRECEDING AND
CURRENT ROW AND
value FOLLOWING AND
1101B464

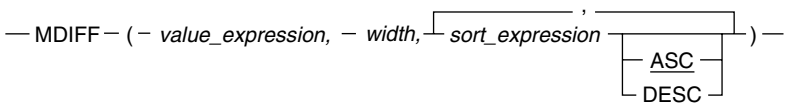
## CSUM


CSUM (
value_expression,
sort_expression
,
ASC
DESC
)
1101A398

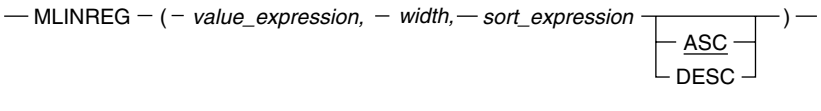
## MAVG



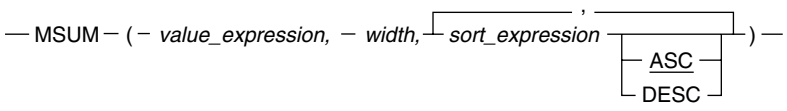

## MDIFF



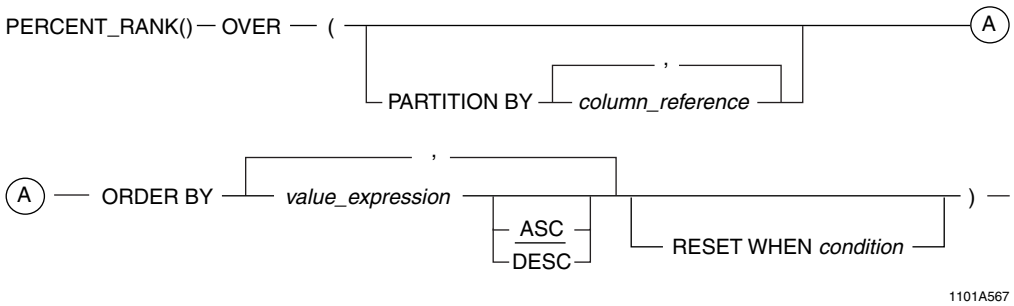

## MLINREG



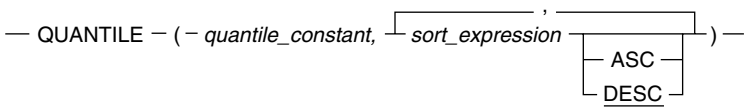

## MSUM



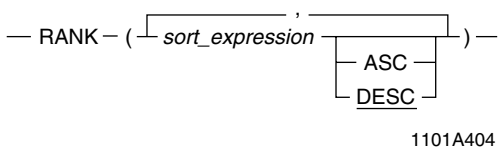

## PERCENT_RANK



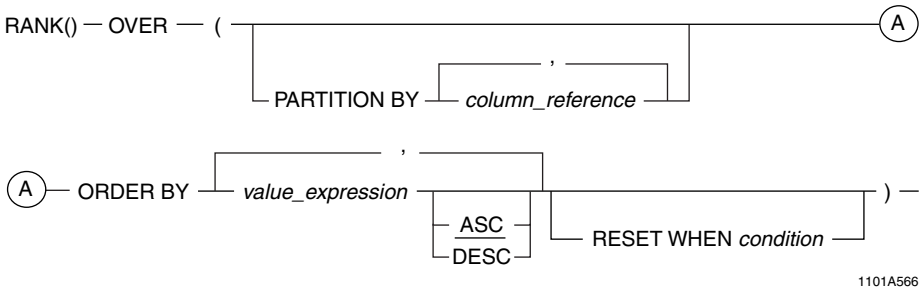
## QUANTILE



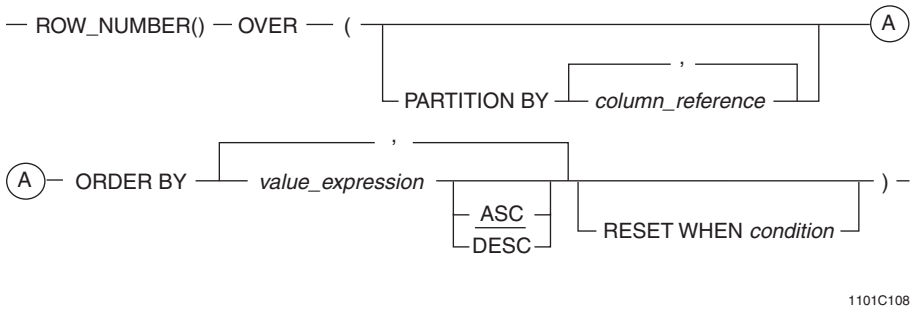

## RANK (Teradata-Specific Function)




## RANK (SQL:2008 Window Function)

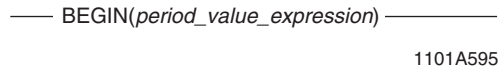



## ROW_NUMBER

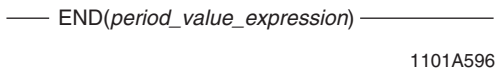


# Period Functions and Operators

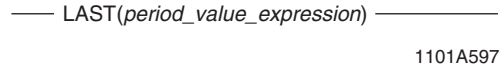
## BEGIN



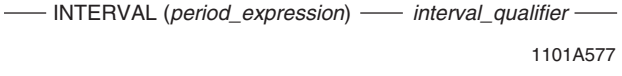

## END



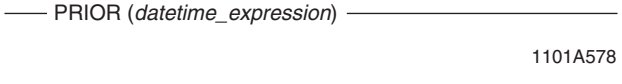

## LAST




## INTERVAL

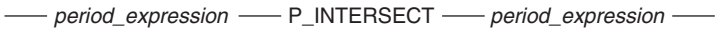



## PRIOR




## NEXT

## P_INTERSECT

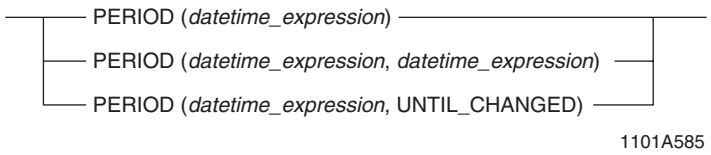



## LDIFF

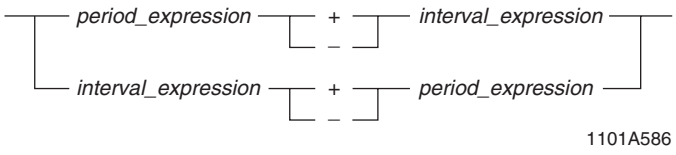

## RDIFF


## P_NORMALIZE


## Period Value Constructor

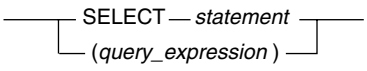



## Arithmetic Operators



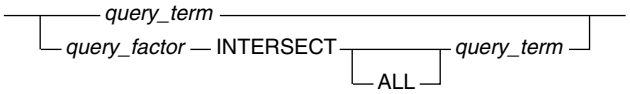
# Set Operators

## Syntax for *query_term*



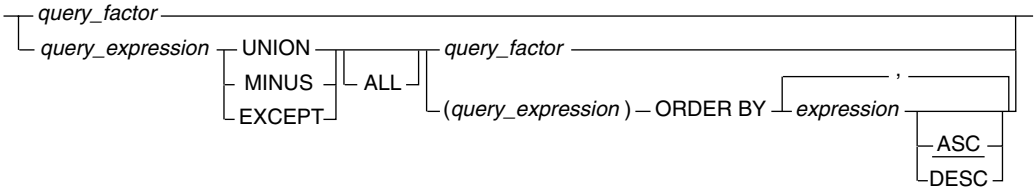

FF07D178

## Syntax for *query_factor*



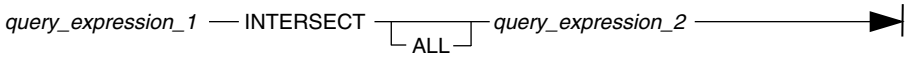

HH01A061

## Syntax for *query_expression*



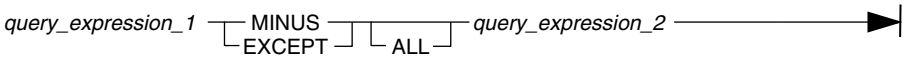

FF07D179

## INTERSECT Operator



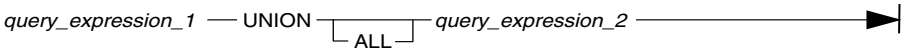

FF07D176

## MINUS/EXCEPT Operator




FF07D177

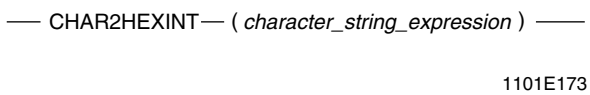
## UNION Operator




FF07D175

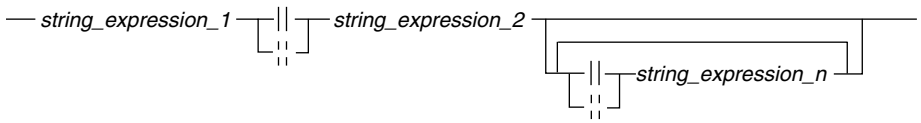
# String Operator and Functions

## CHAR2HEXINT



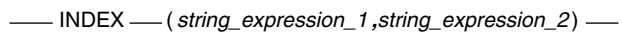

1101E173

## Concatenation Operator



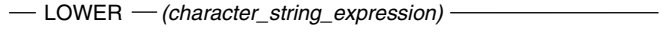

FF07D195

## INDEX



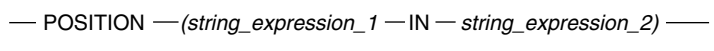

FF07D253

## LOWER



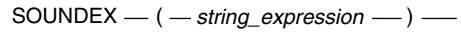

FF07D091

## POSITION



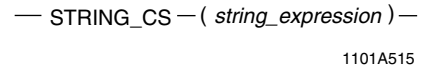

FF07D090

## SOUNDEX



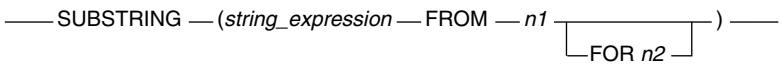

KO01A060

## STRING_CS



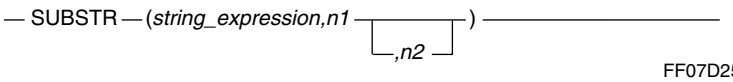

1101A515

## SUBSTRING/SUBSTR (ANSI Syntax)



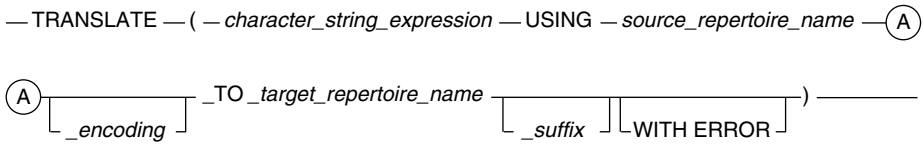

FF07D256

## SUBSTRING/SUBSTR (Teradata Syntax)



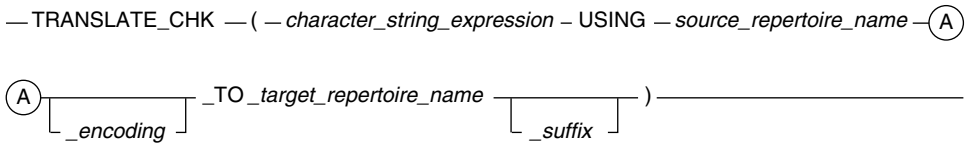

FF07D257

## TRANSLATE




1101E198

## TRANSLATE_CHK

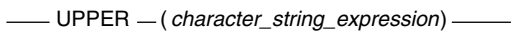



1101E199

## TRIM

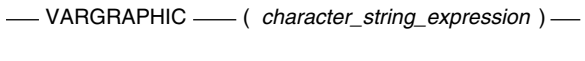

1101F200

## UPPER




FF07D258

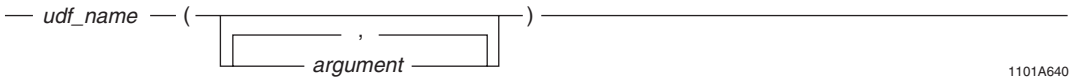
### VARGRAPHIC




1101E197

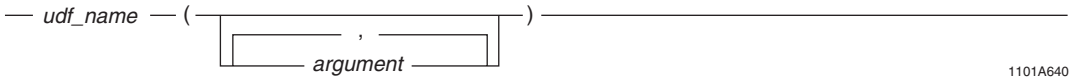
# UDF Expressions

## Scalar UDF Expression




1101A640

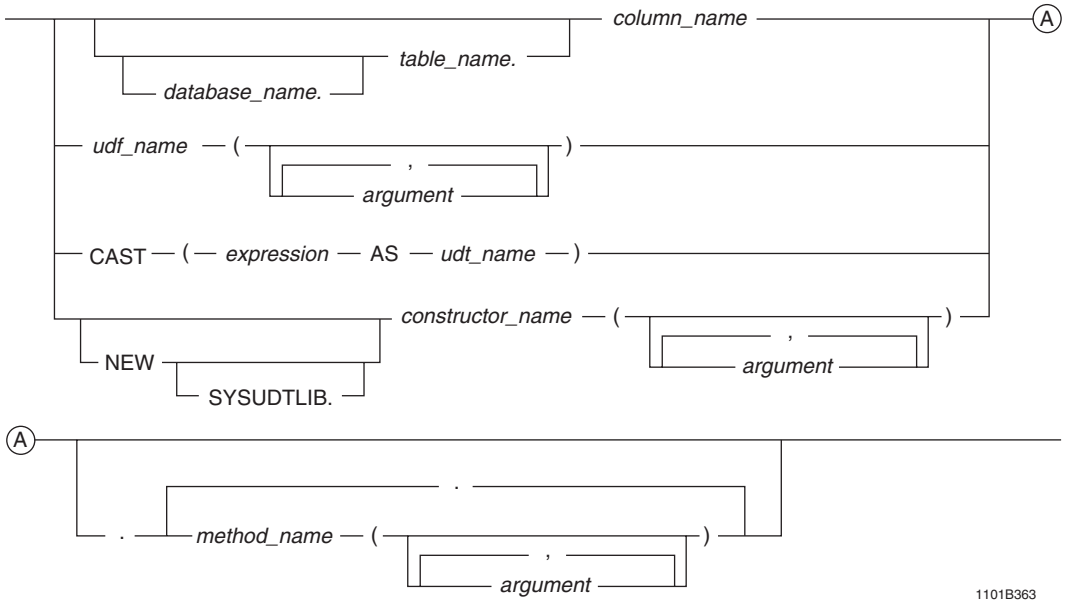
## Aggregate UDF Expression



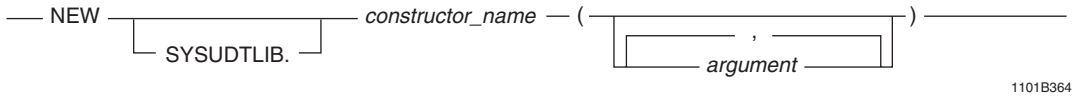

1101A640

# UDT Expressions and Methods

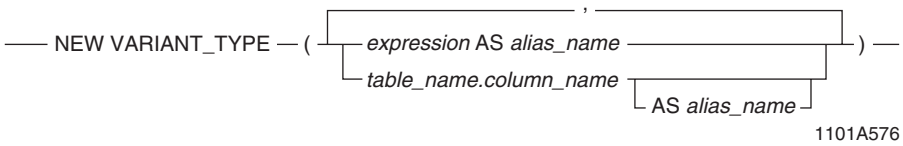
## UDT Expression



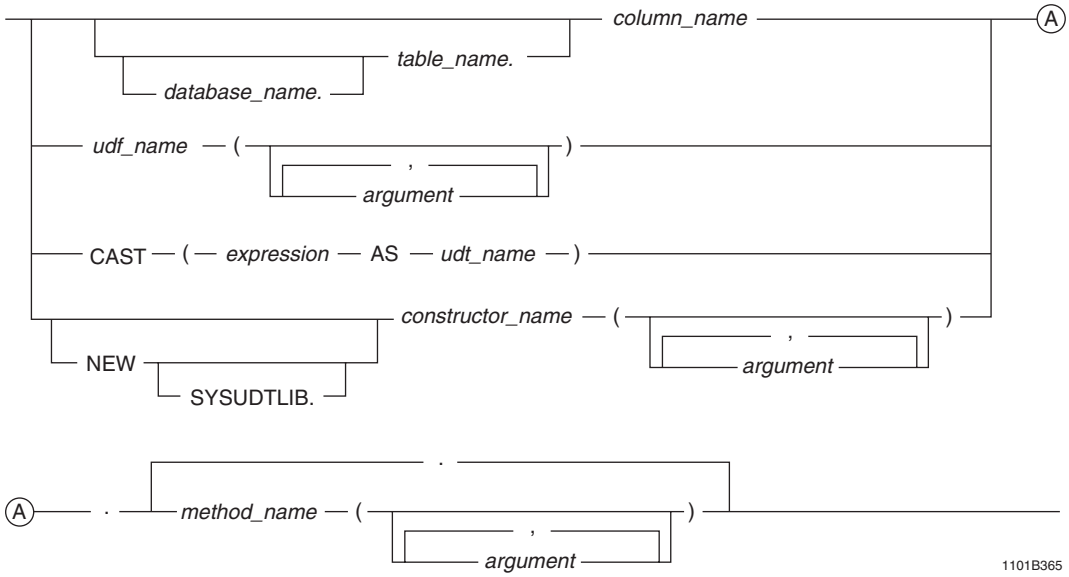

## NEW




## NEW VARIANT_TYPE

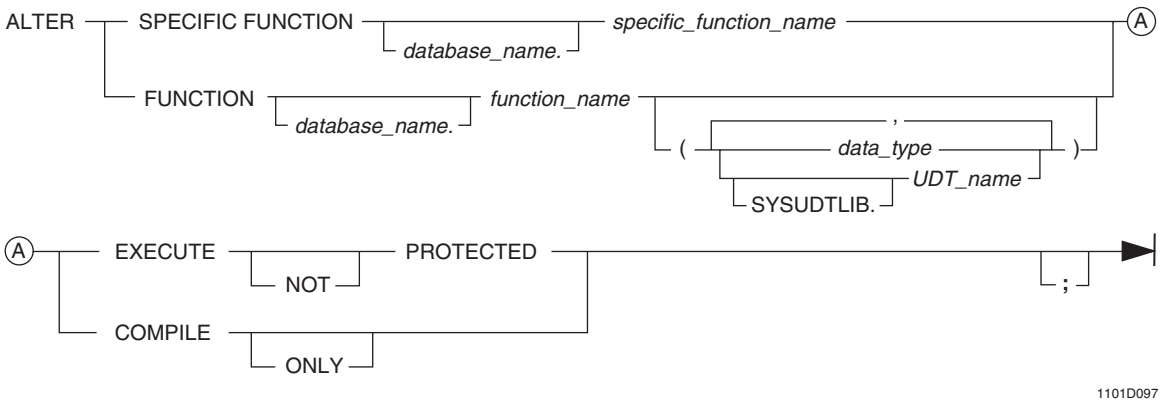


## Method Invocation

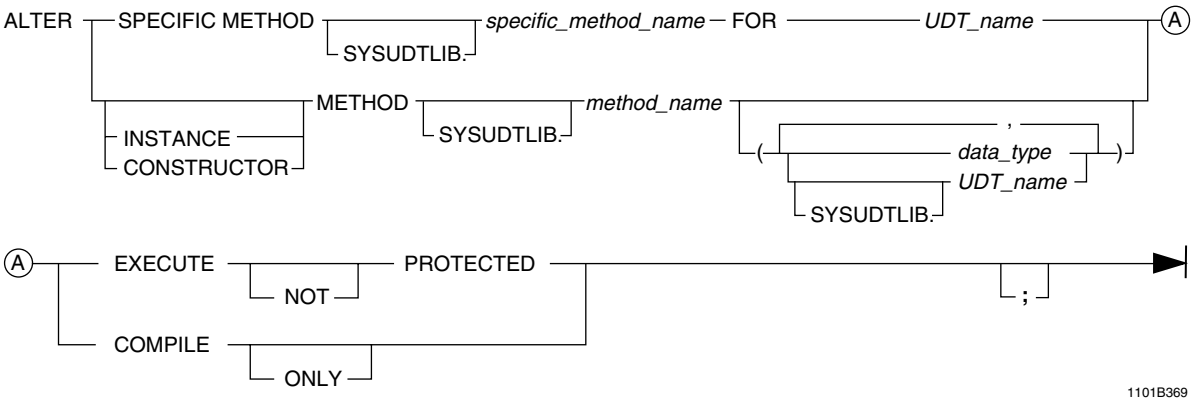


# CHAPTER 3 SQL Data Definition Language

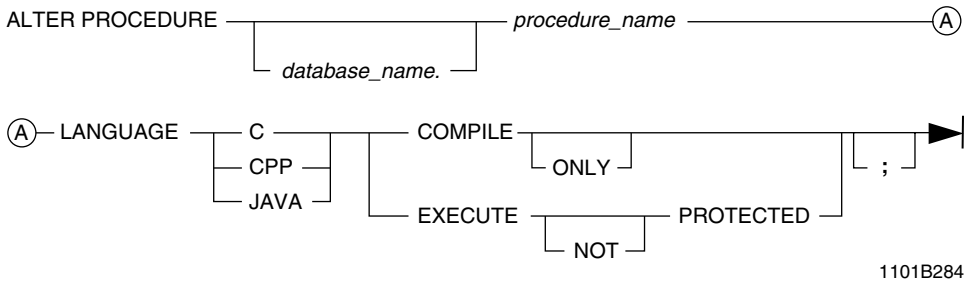
## ALTER FUNCTION



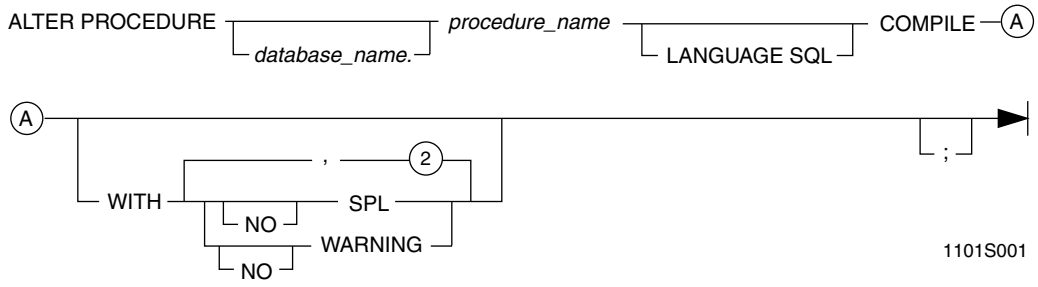

## ALTER METHOD



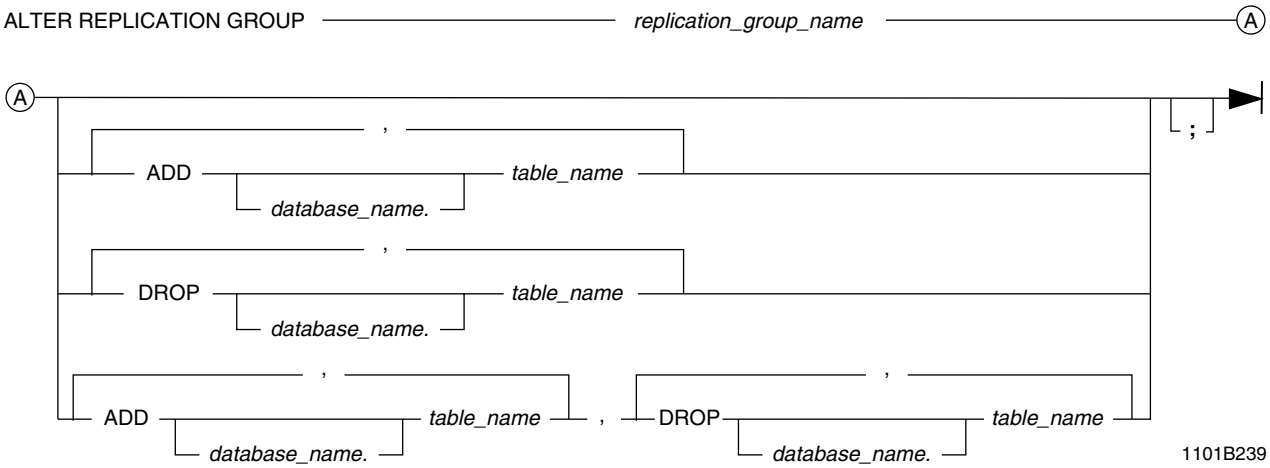
# ALTER PROCEDURE (External Form)




# ALTER PROCEDURE (SQL Form)

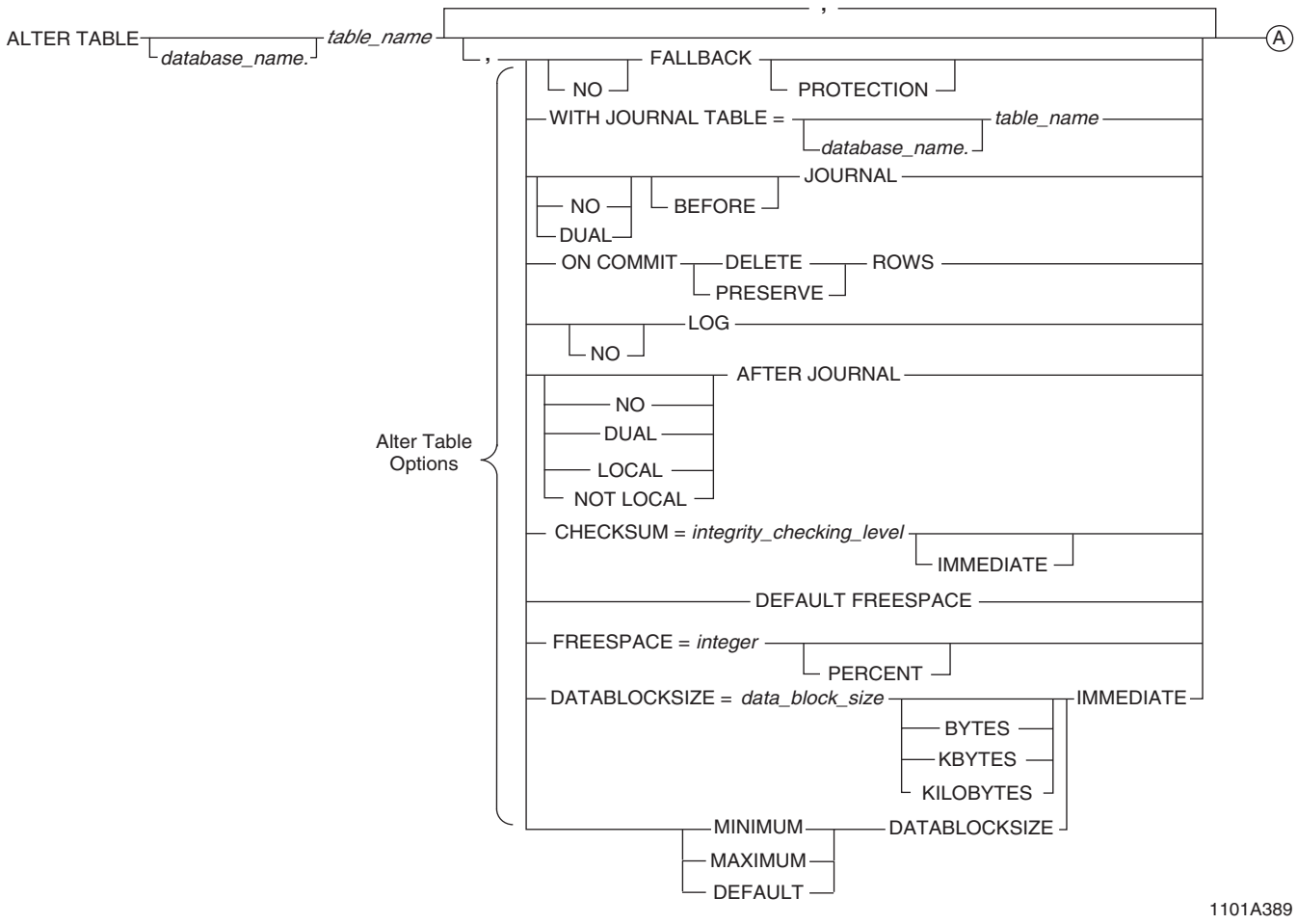


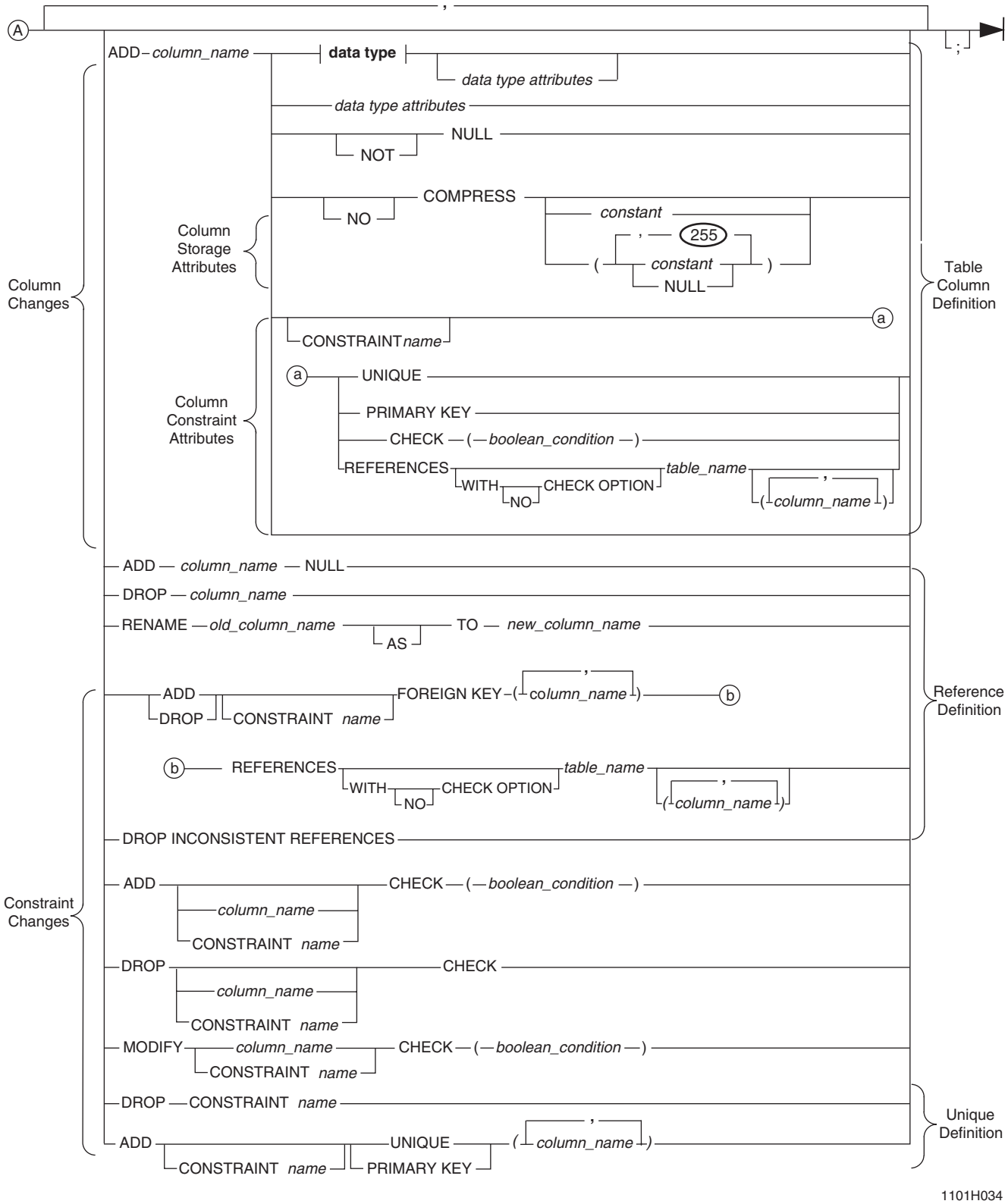

# ALTER REPLICATION GROUP



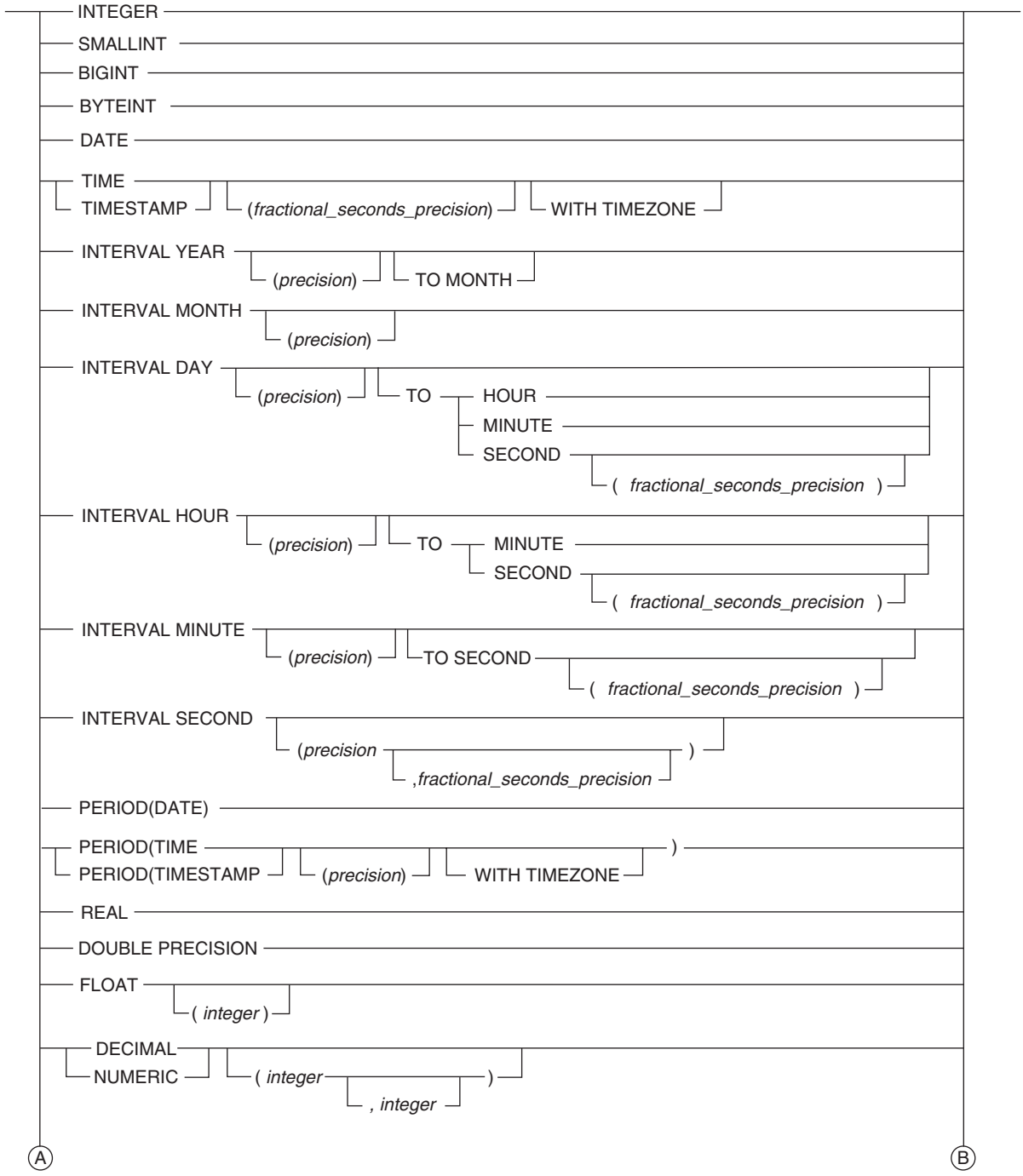
# ALTER TABLE

## Basic Table Parameters Modification Syntax

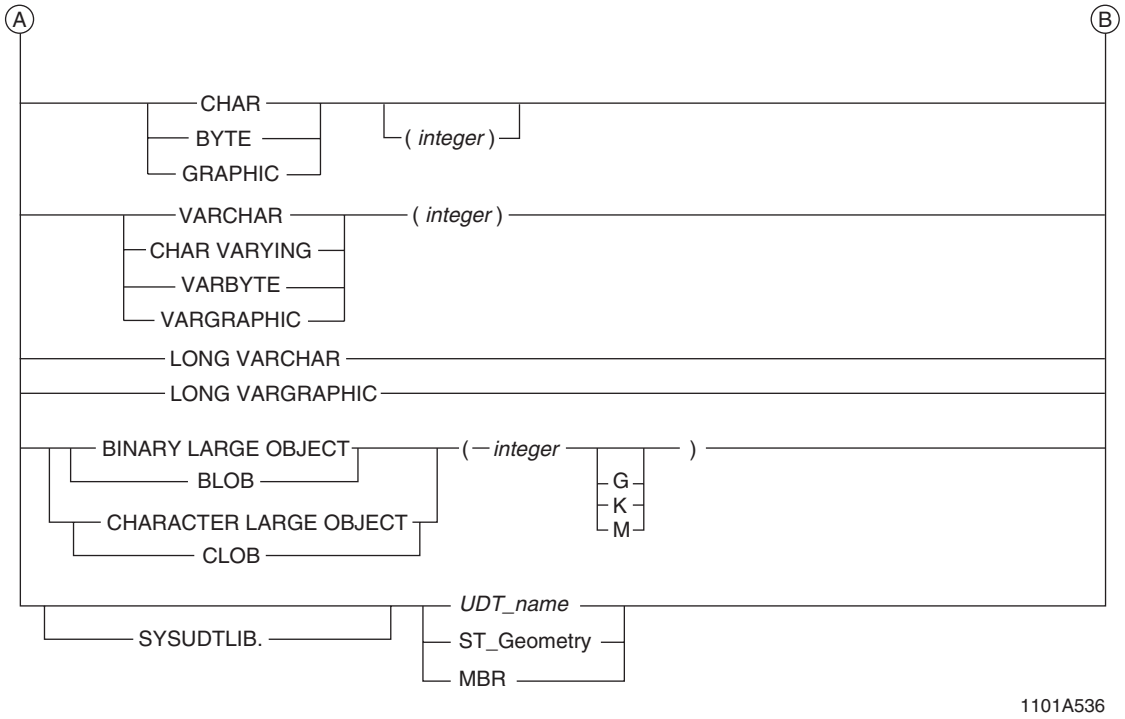



A
,
;
ADD column_name
data type
data type attributes
data type attributes
NULL
NOT
COMPRESS
NO
constant
255
constant
NULL
Column Storage Attributes
Column Changes
Table Column Definition
CONSTRAINT name
a
UNIQUE
PRIMARY KEY
CHECK ( boolean_condition )
REFERENCES
WITH
NO
CHECK OPTION
table_name
column_name
Column Constraint Attributes
ADD column_name NULL
DROP column_name
RENAME old_column_name
AS
TO new_column_name
ADD
DROP
CONSTRAINT name
FOREIGN KEY
column_name
b
REFERENCES
WITH
NO
CHECK OPTION
table_name
column_name
Reference Definition
DROP INCONSISTENT REFERENCES
ADD
column_name
CONSTRAINT name
CHECK ( boolean_condition )
DROP
column_name
CONSTRAINT name
CHECK
MODIFY
column_name
CONSTRAINT name
CHECK ( boolean_condition )
DROP CONSTRAINT name
ADD
CONSTRAINT name
UNIQUE
PRIMARY KEY
column_name
Unique Definition
Constraint Changes
1101H034

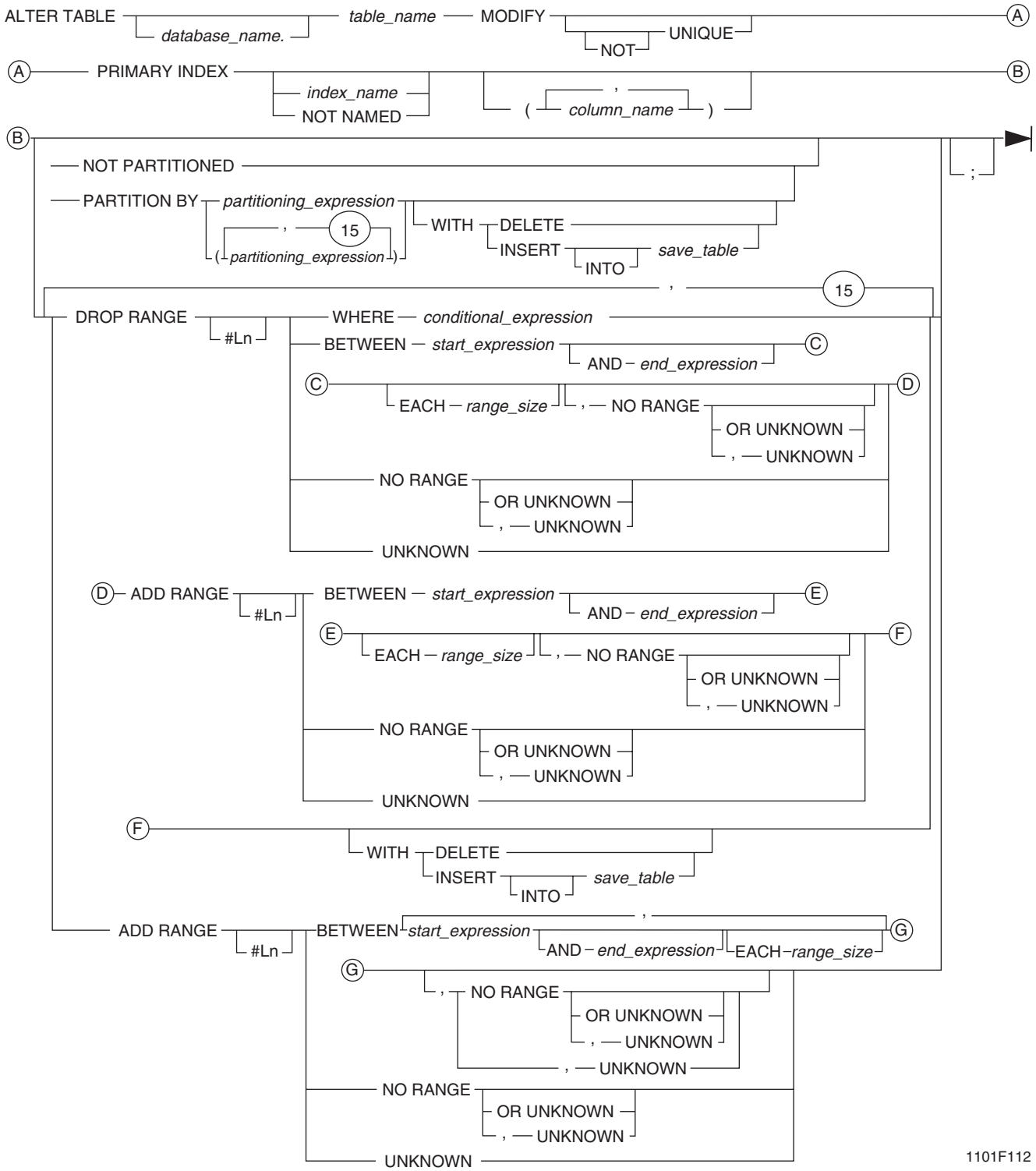
**Data Type**



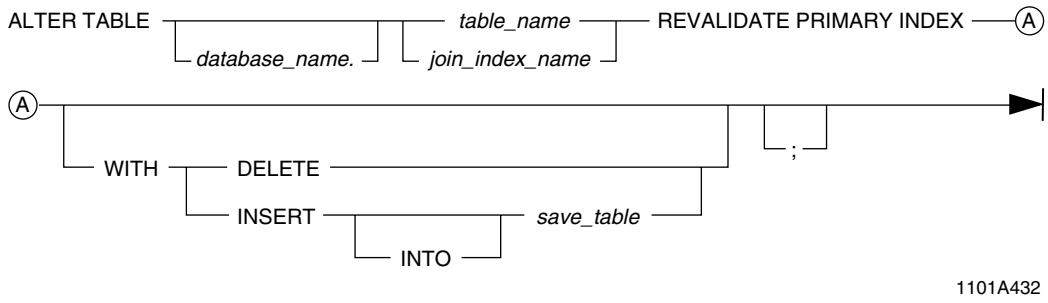

1101A535


A
B
CHAR
BYTE
GRAPHIC
( integer )
VARCHAR
CHAR VARYING
VARBYTE
VARGRAPHIC
( integer )
LONG VARCHAR
LONG VARGRAPHIC
BINARY LARGE OBJECT
BLOB
CHARACTER LARGE OBJECT
CLOB
( integer
G
K
M
)
SYSUDTLIB.
UDT_name
ST_Geometry
MBR
1101A536

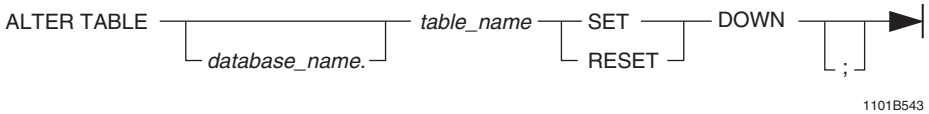
## Primary Index Modification Syntax



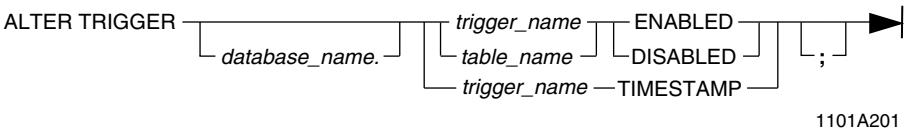
## Partitioned Primary Index Revalidation Syntax



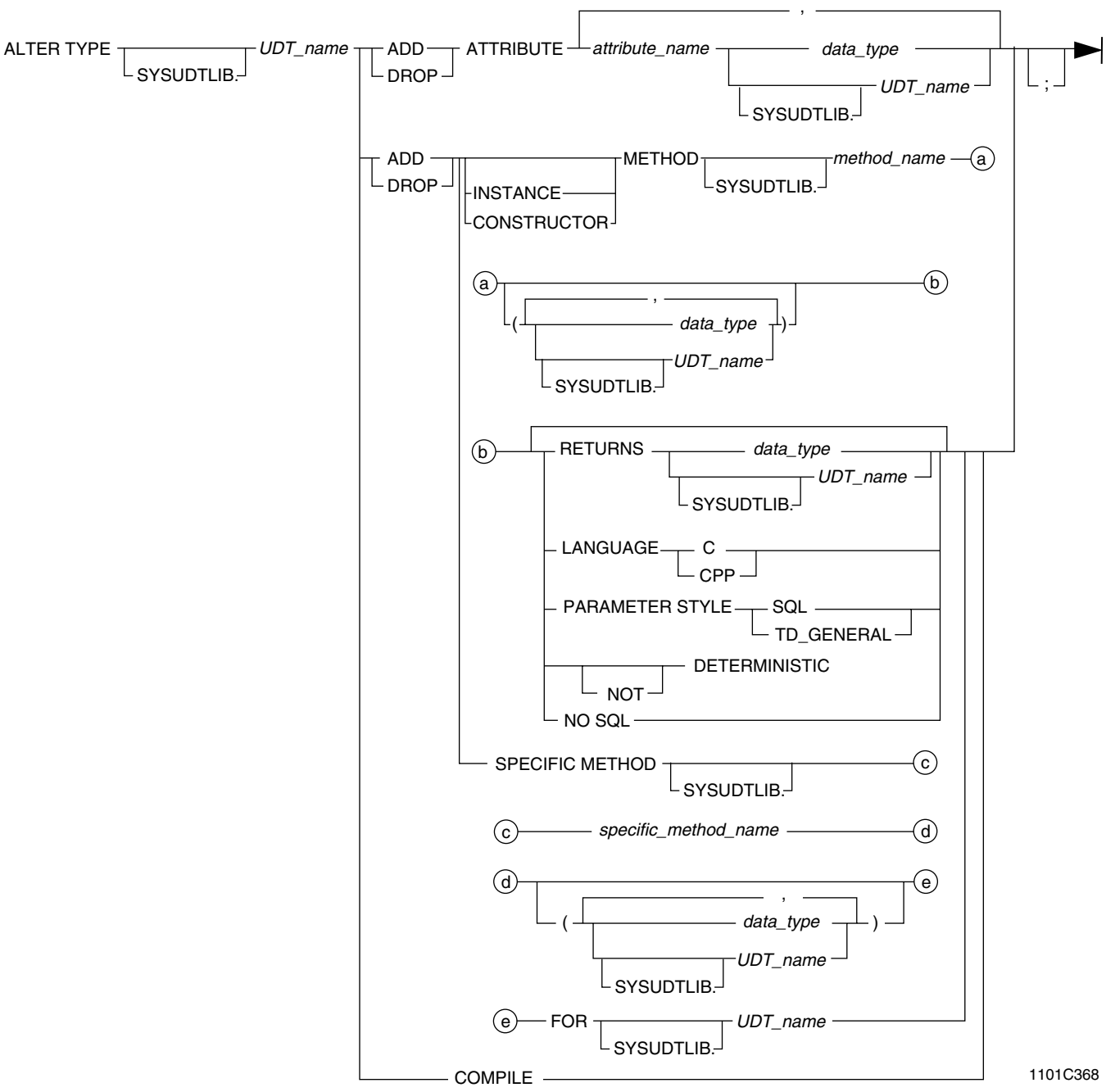

## Set Down/Reset Down Syntax



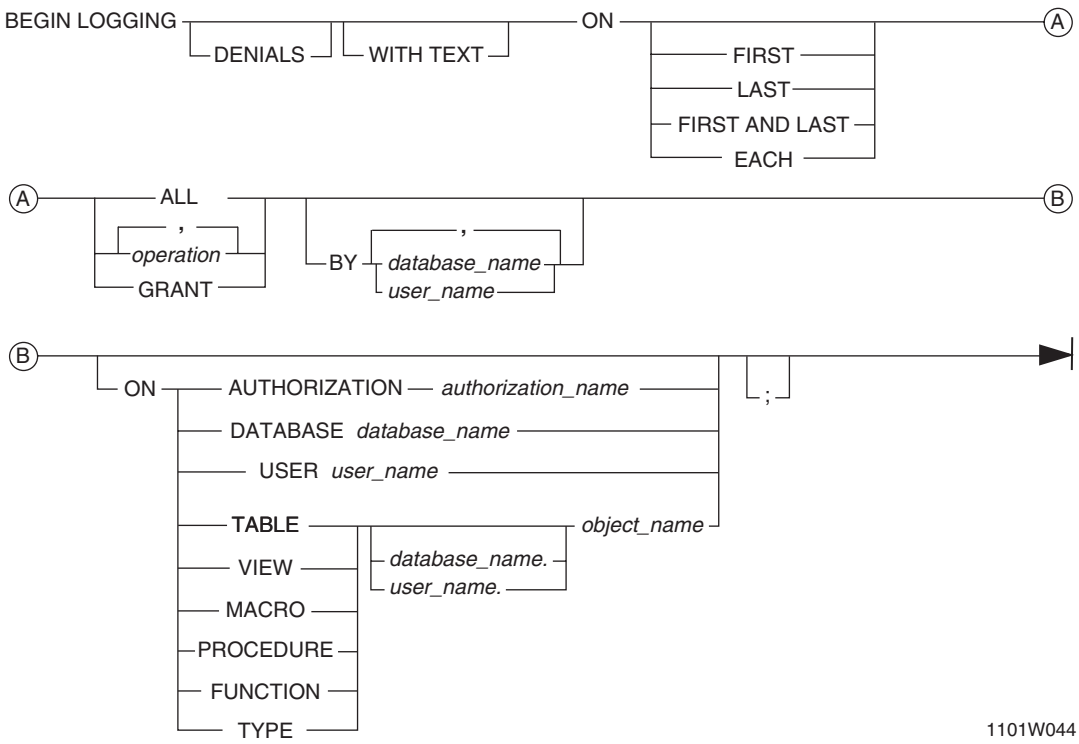

# ALTER TRIGGER



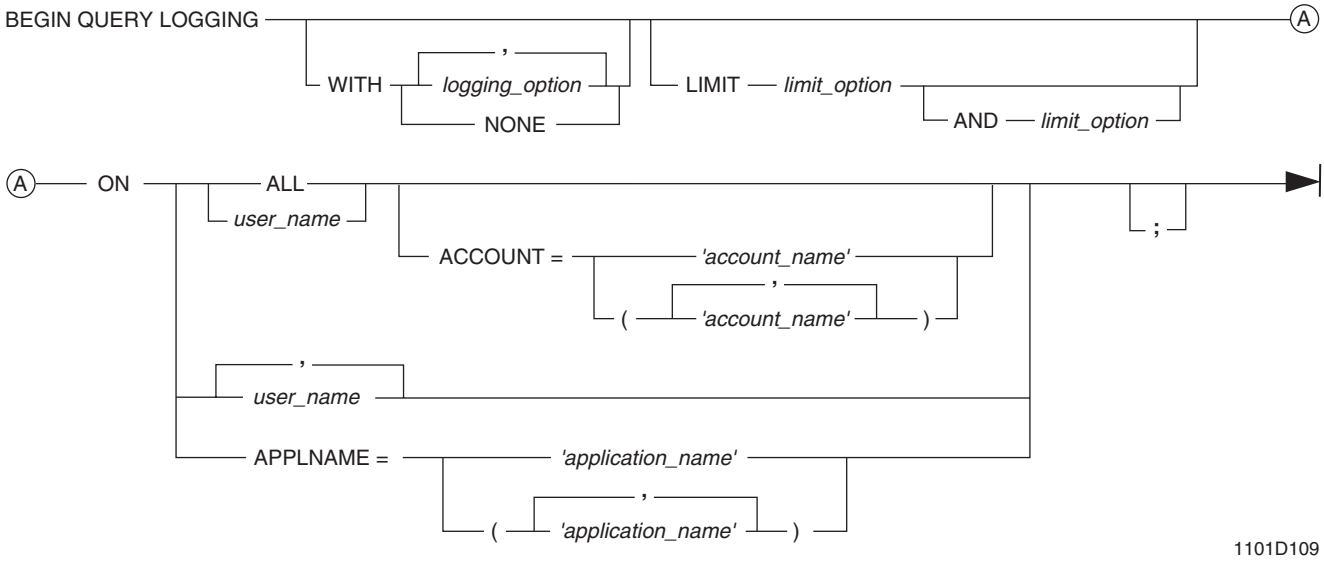
# ALTER TYPE


ALTER TYPE
SYSUDTLIB.
UDT_name
ADD
DROP
ATTRIBUTE
,
attribute_name
data_type
UDT_name
SYSUDTLIB.
;
ADD
DROP
INSTANCE
CONSTRUCTOR
METHOD
SYSUDTLIB.
method_name
a
b
(
)
RETURNS
LANGUAGE
C
CPP
PARAMETER STYLE
SQL
TD_GENERAL
DETERMINISTIC
NOT
NO SQL
SPECIFIC METHOD
c
d
e
FOR
COMPILE
1101C368

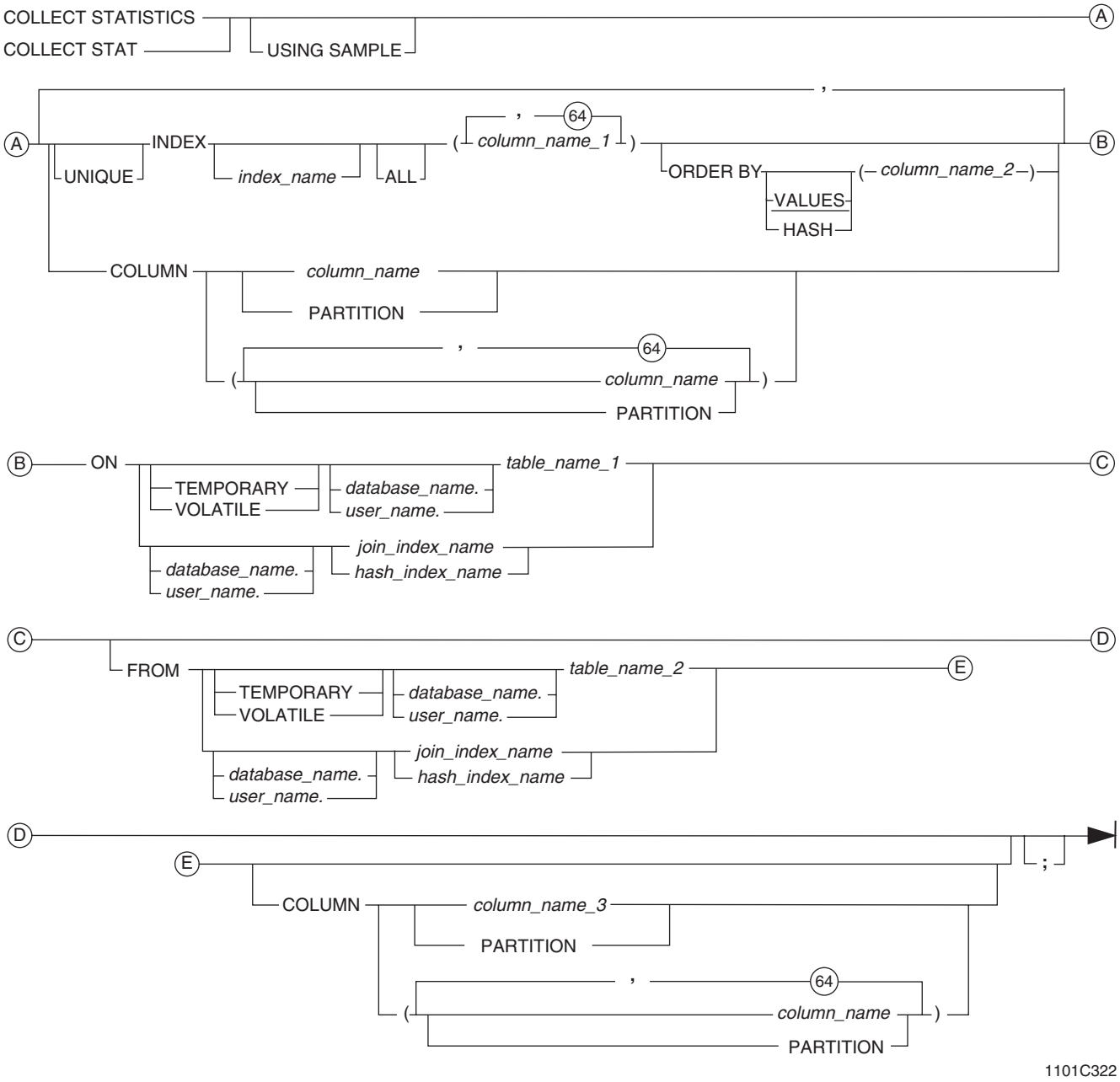
# BEGIN LOGGING



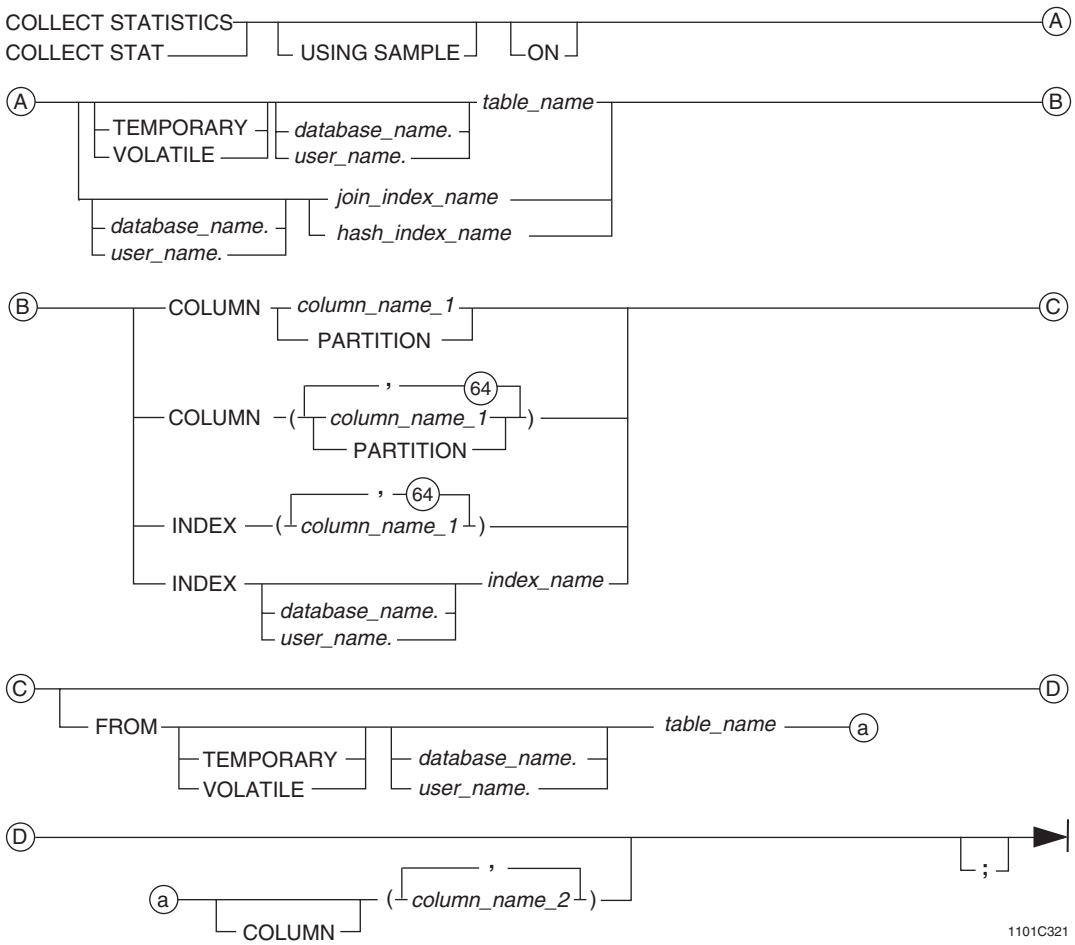
# BEGIN QUERY LOGGING



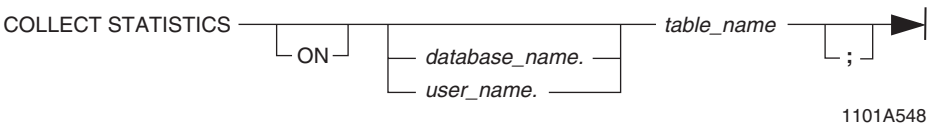
# COLLECT STATISTICS (Optimizer Form)



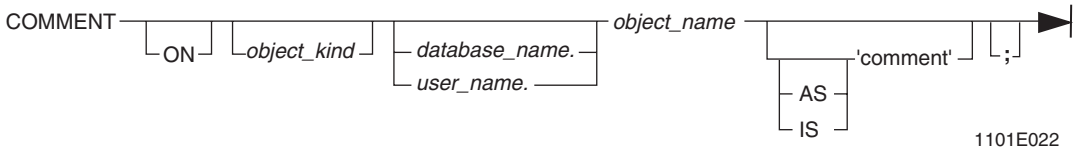
# COLLECT STATISTICS (Alternate Optimizer Form)



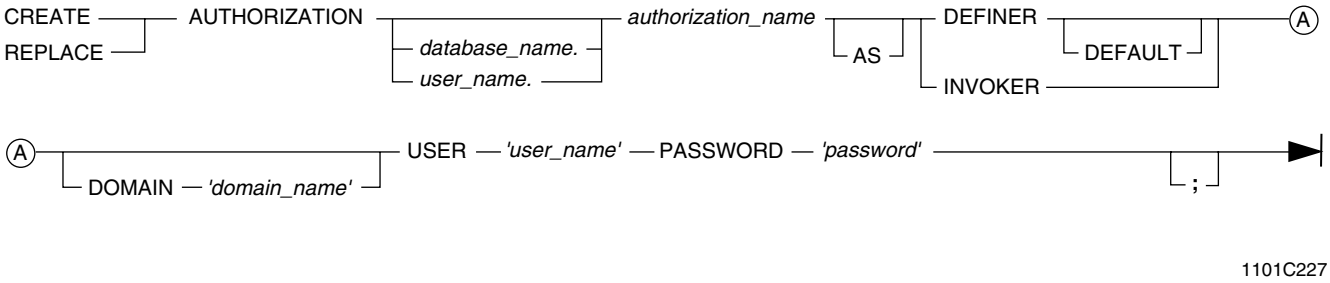

# COLLECT STATISTICS (Recollect Statistics)



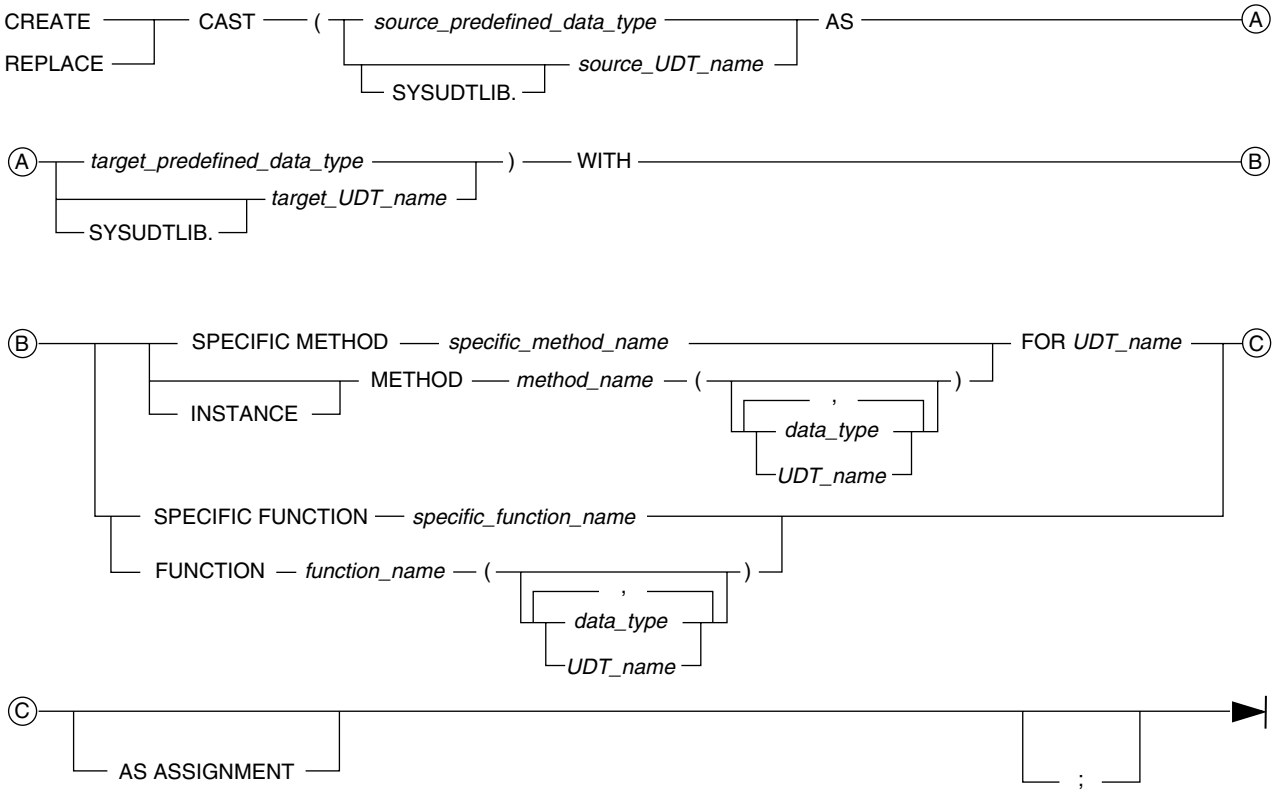
# COMMENT (Comment Placing Form)




# CREATE AUTHORIZATION/ REPLACE AUTHORIZATION

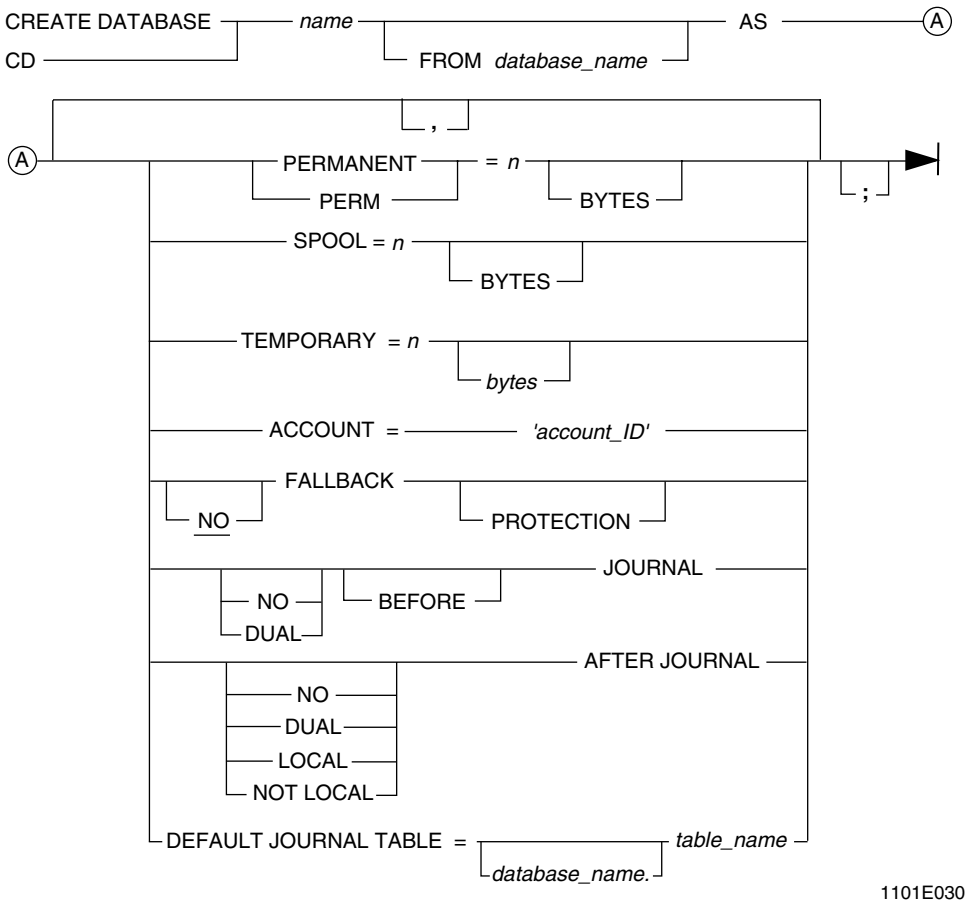


# CREATE CAST/ REPLACE CAST

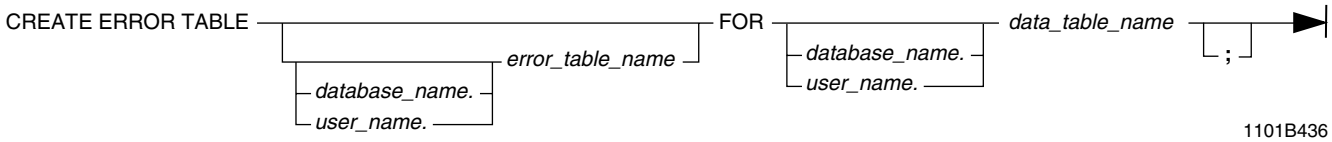



1101A358

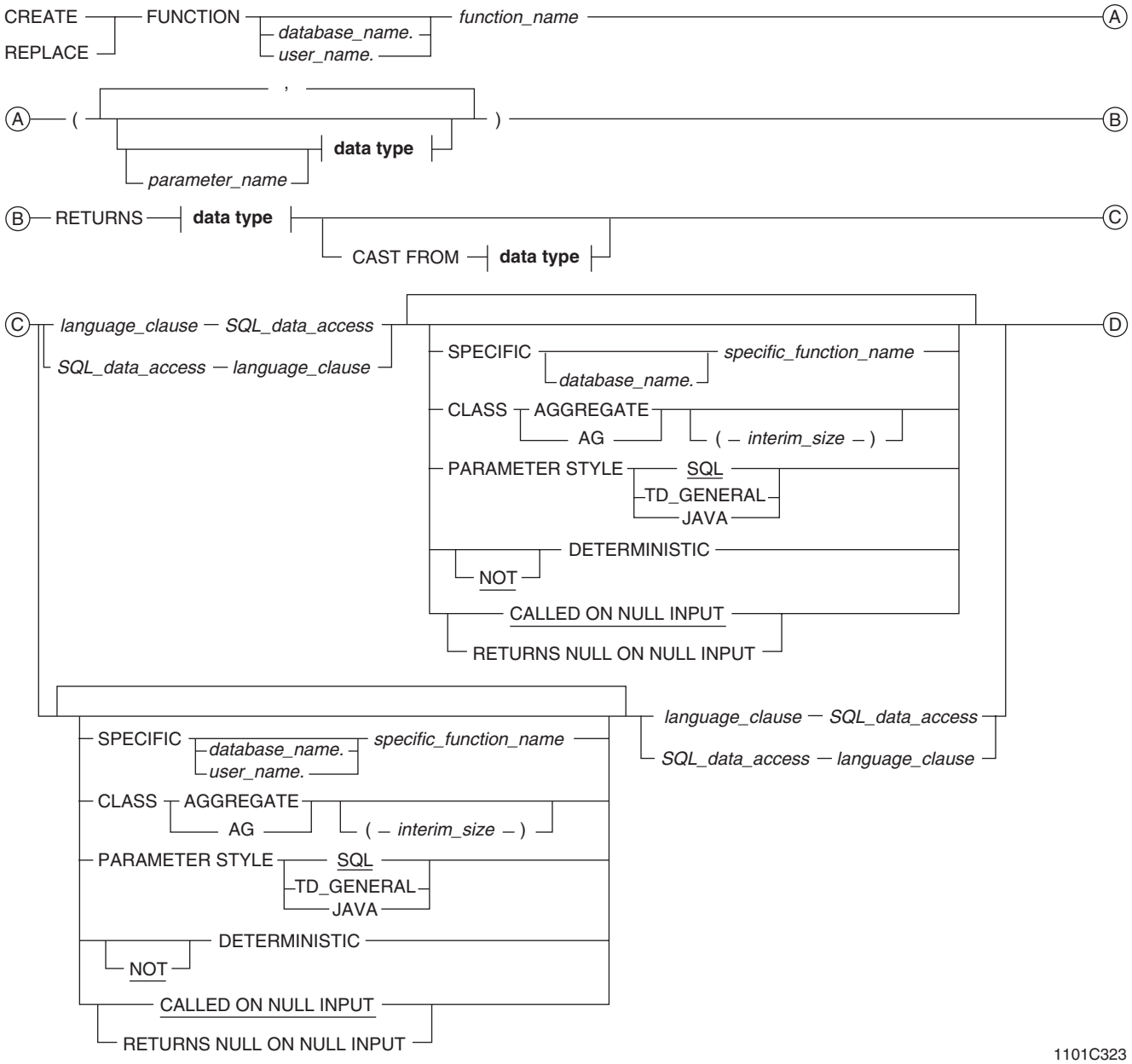
# CREATE DATABASE

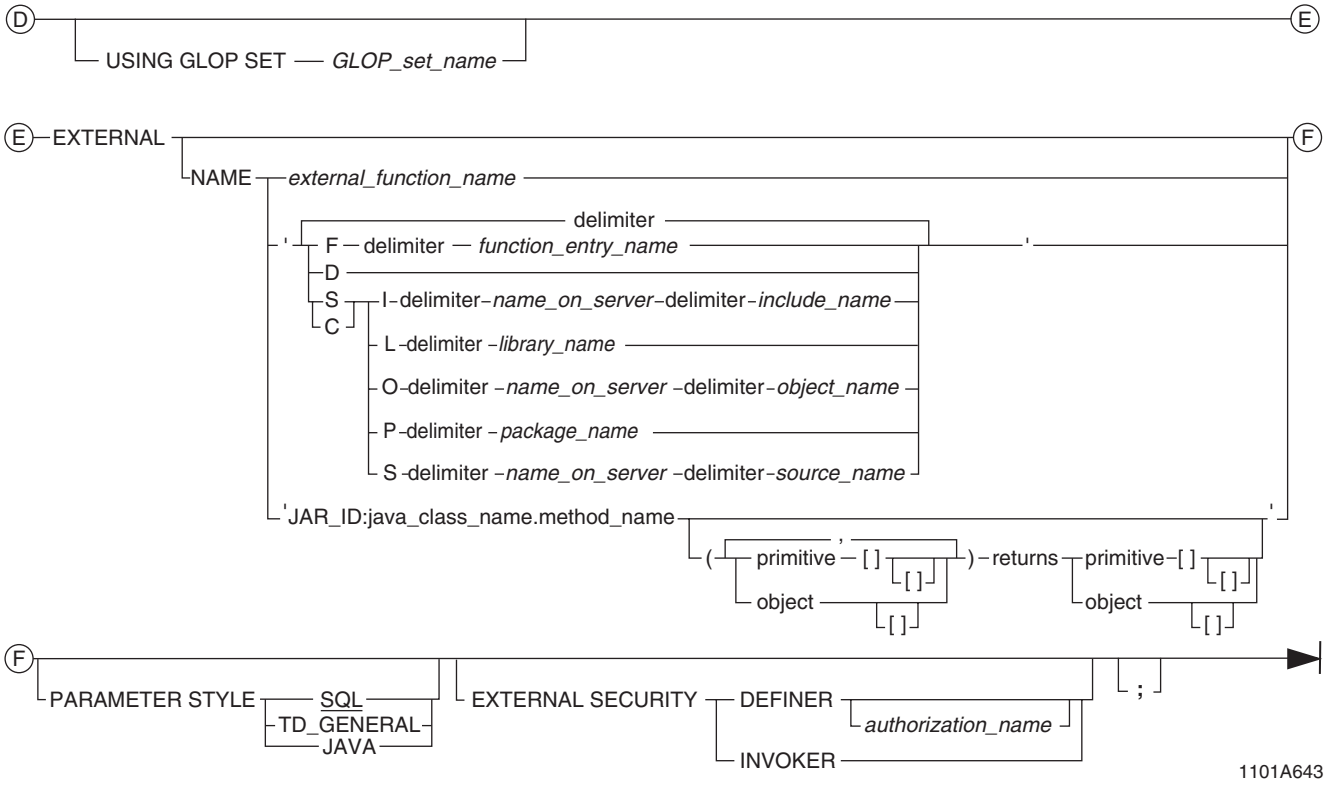



# CREATE ERROR TABLE

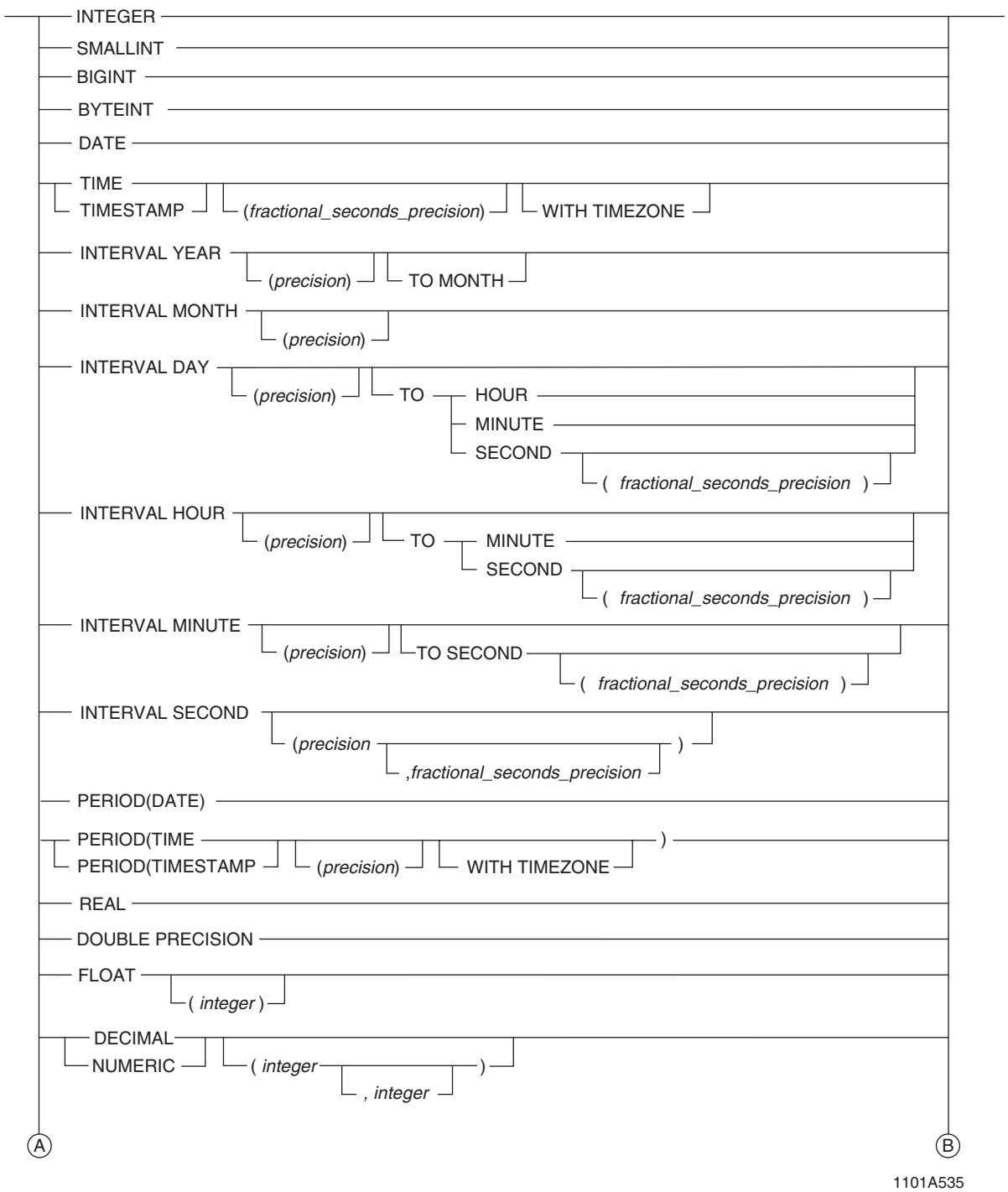


# CREATE FUNCTION/ REPLACE FUNCTION

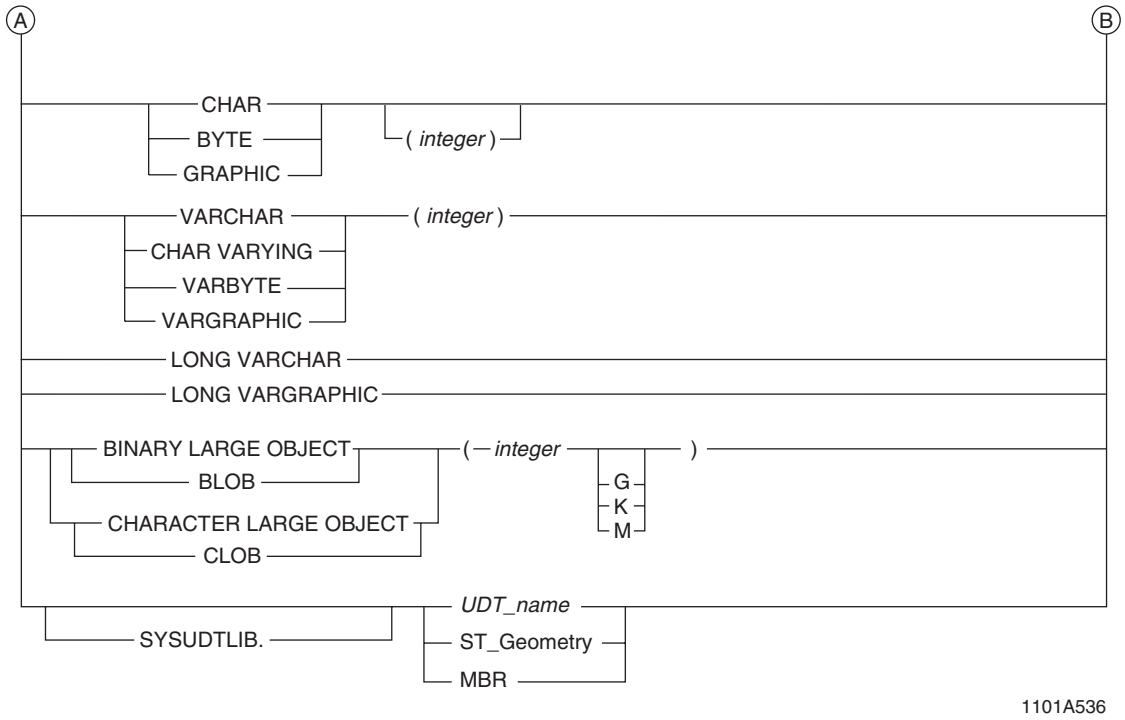



D
USING GLOP SET
GLOP_set_name
E
E
EXTERNAL
NAME
external_function_name
delimiter
'
F
delimiter
function_entry_name
D
S
C
I
delimiter
name_on_server
delimiter
include_name
L
delimiter
library_name
O
delimiter
name_on_server
delimiter
object_name
P
delimiter
package_name
S
delimiter
name_on_server
delimiter
source_name
'
'JAR_ID:java_class_name.method_name'
(
primitive
[ ]
,
object
)
returns
primitive
object
F
PARAMETER STYLE
SQL
TD_GENERAL
JAVA
EXTERNAL SECURITY
DEFINER
authorization_name
INVOKER
;
1101A643

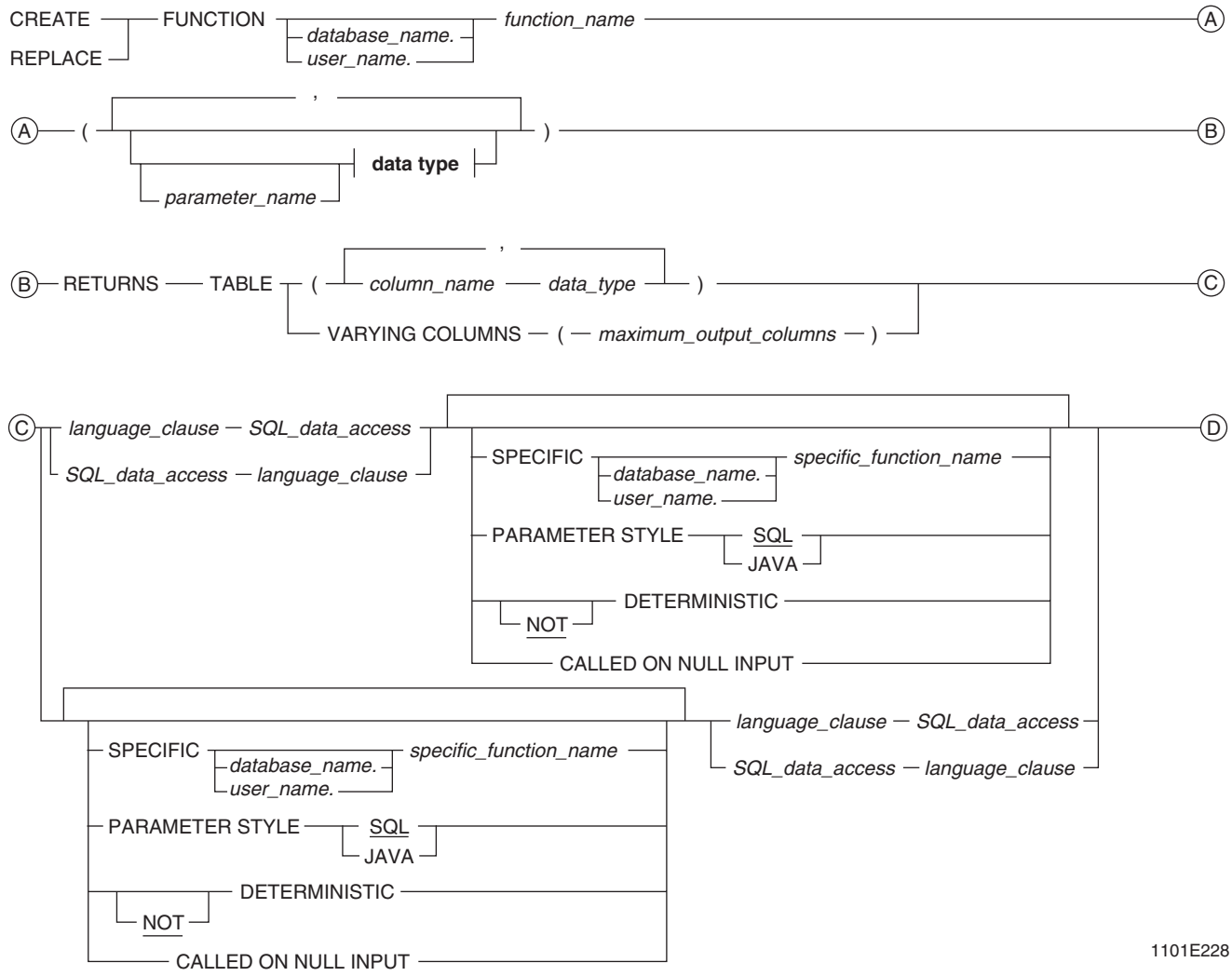
**Data Type**

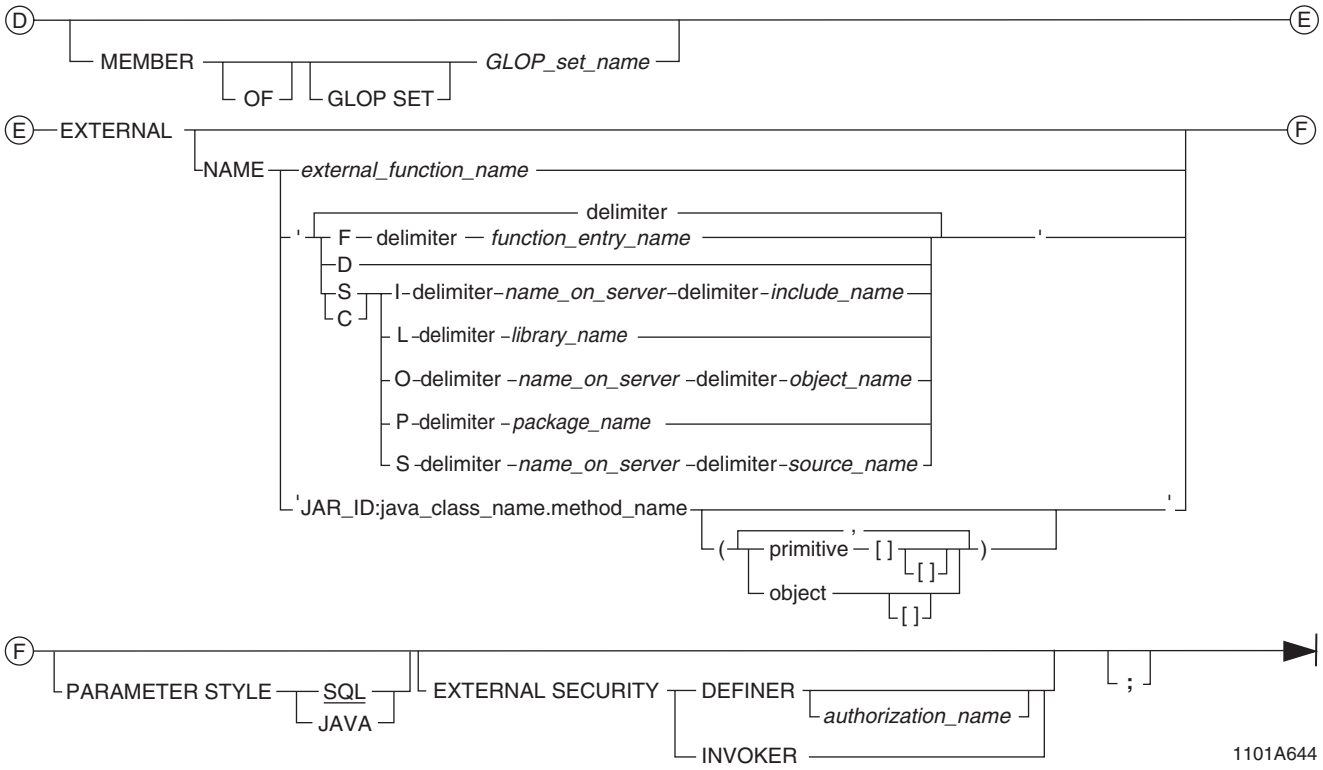



1101A535

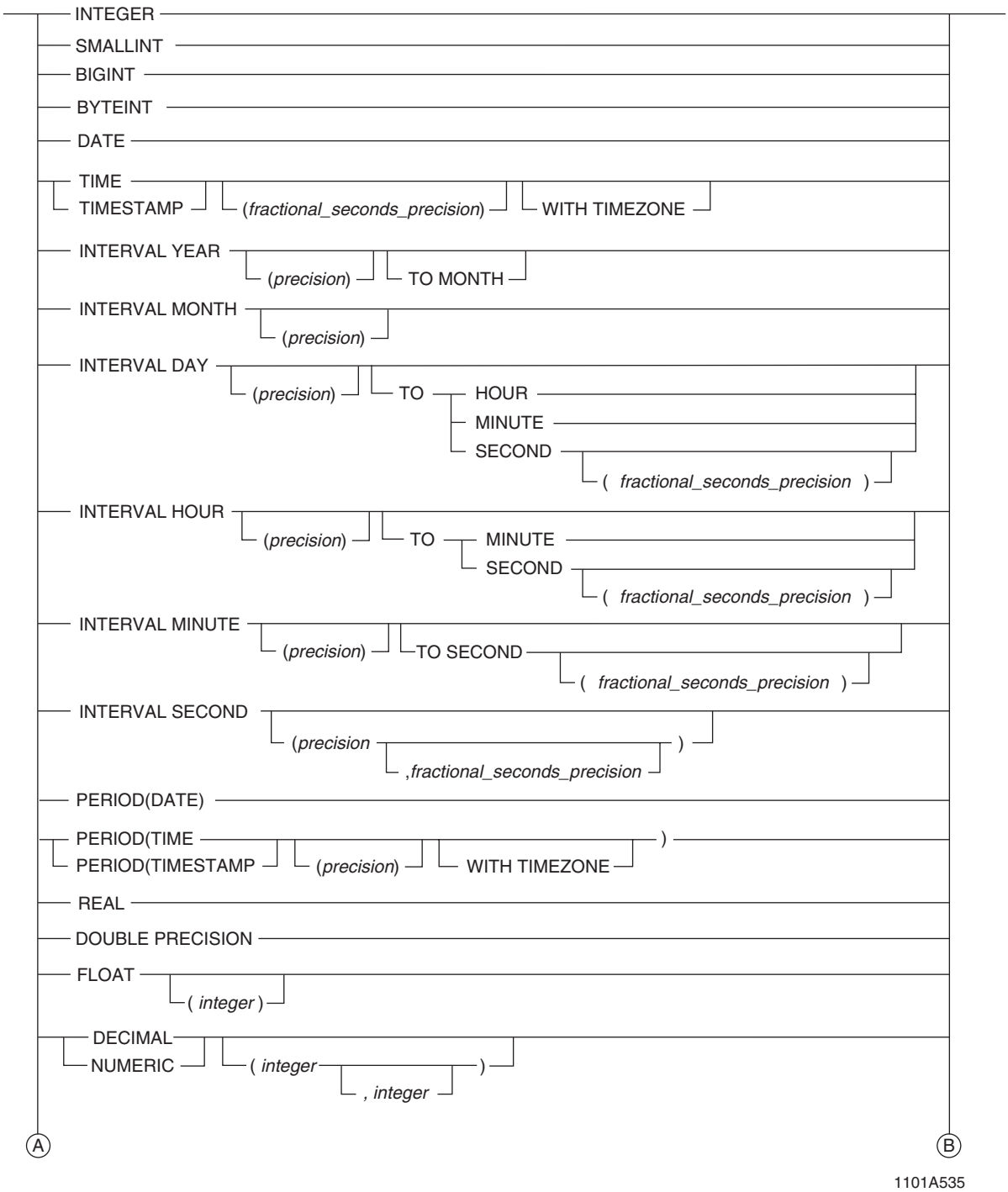

A
B
CHAR
BYTE
GRAPHIC
( integer )
VARCHAR
CHAR VARYING
VARBYTE
VARGRAPHIC
( integer )
LONG VARCHAR
LONG VARGRAPHIC
BINARY LARGE OBJECT
BLOB
CHARACTER LARGE OBJECT
CLOB
( integer
G
K
M
)
SYSUDTLIB.
UDT_name
ST_Geometry
MBR
1101A536

# CREATE FUNCTION (Table Form)

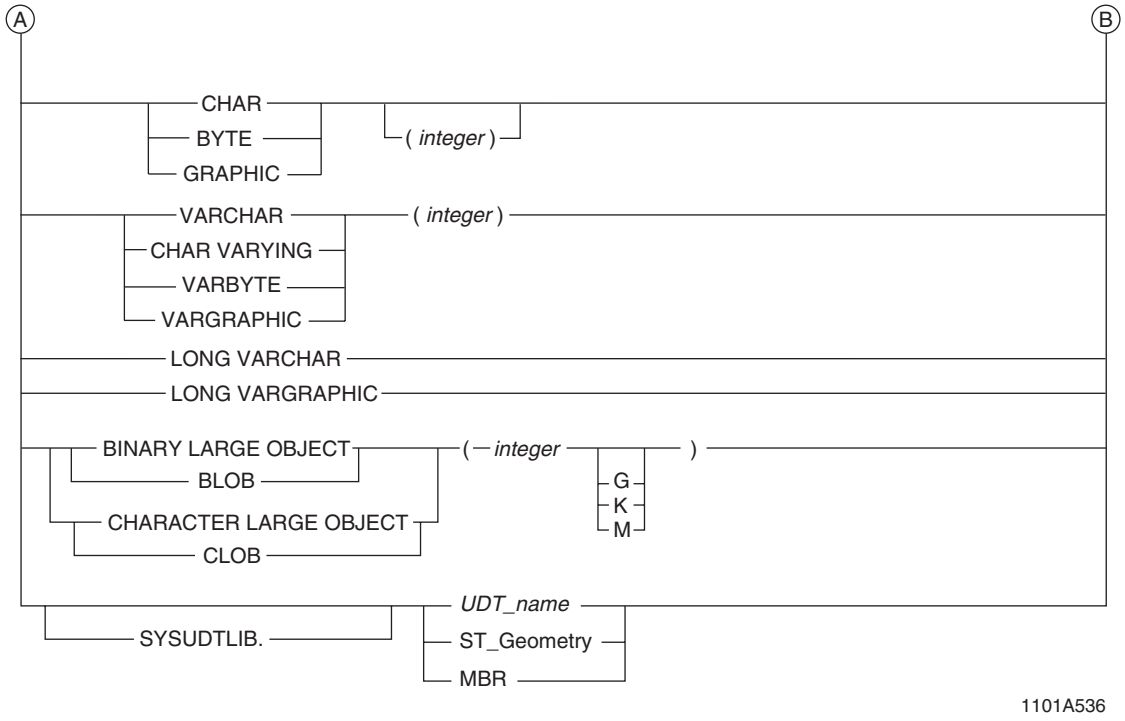



D
E
MEMBER
OF
GLOP SET
GLOP_set_name
E
EXTERNAL
F
NAME
external_function_name
delimiter
'
F
delimiter
function_entry_name
'
D
S
C
I
delimiter
name_on_server
delimiter
include_name
L
delimiter
library_name
O
delimiter
name_on_server
delimiter
object_name
P
delimiter
package_name
S
delimiter
name_on_server
delimiter
source_name
'JAR_ID:java_class_name.method_name'
(
,
primitive
[ ]
[ ]
)
object
[ ]
F
PARAMETER STYLE
SQL
JAVA
EXTERNAL SECURITY
DEFINER
authorization_name
INVOKER
;
1101A644

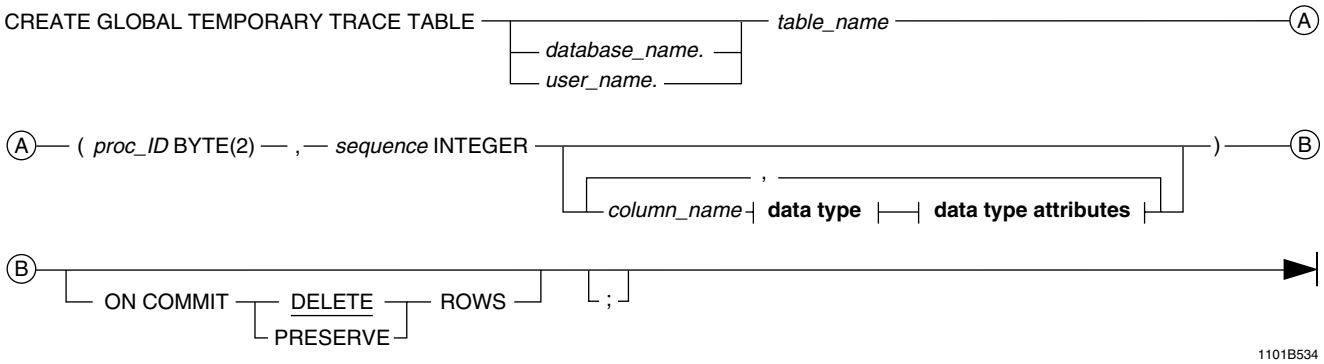
**Data Type**



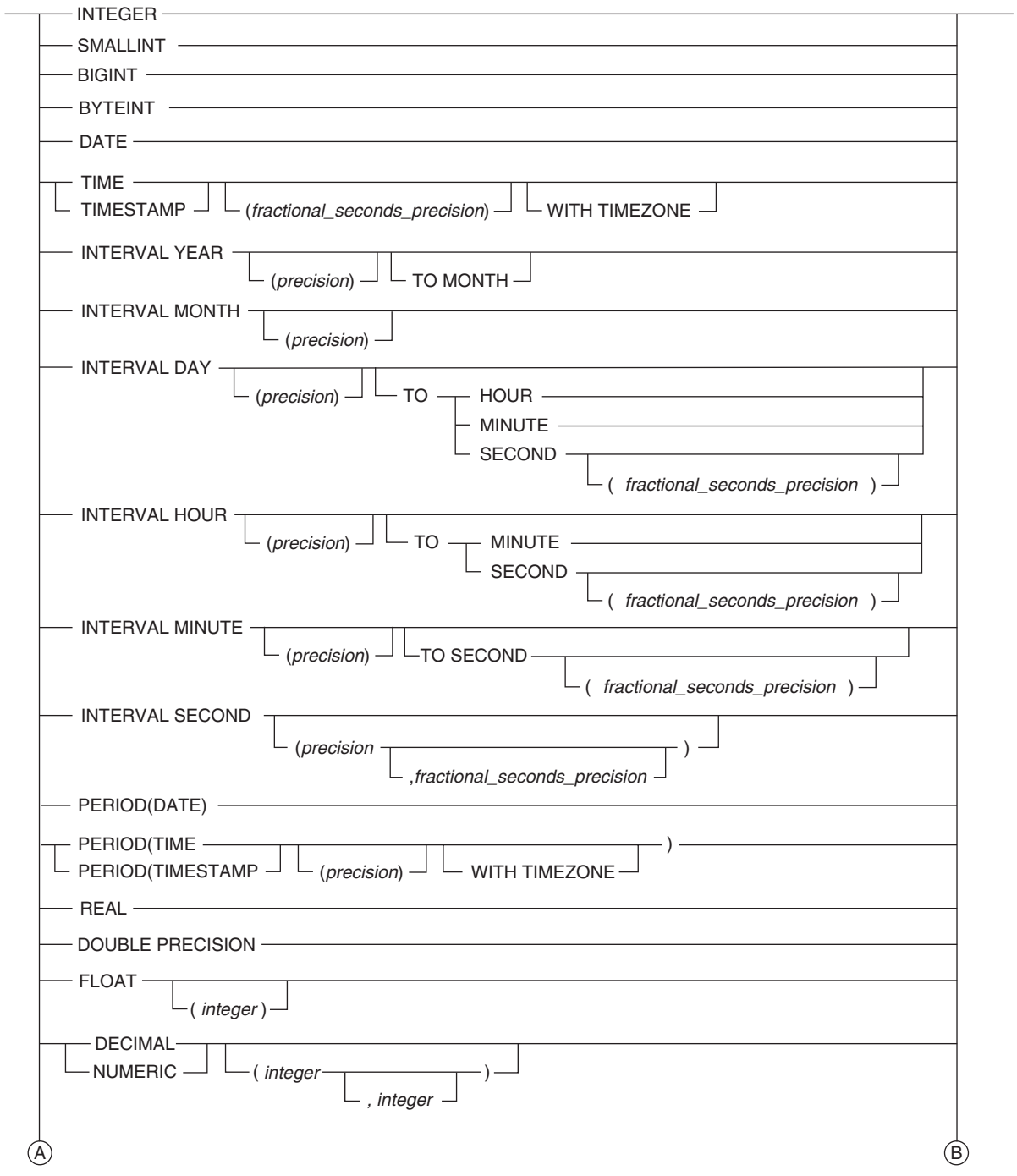

1101A535



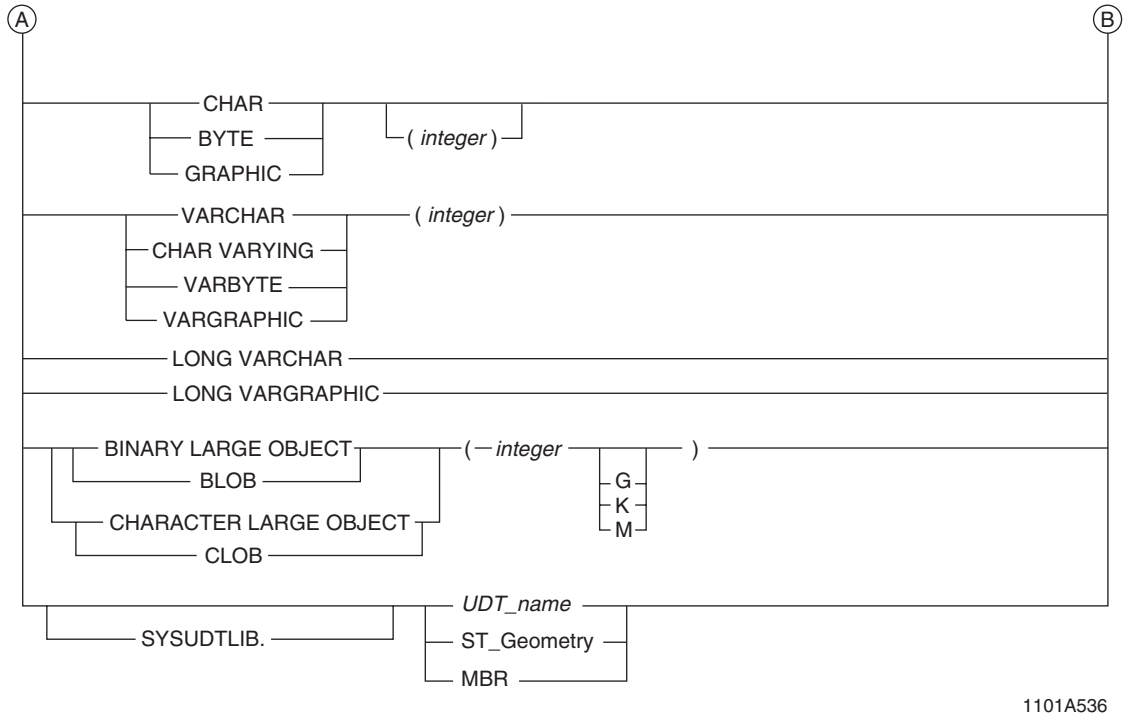

# CREATE GLOBAL TEMPORARY TRACE TABLE



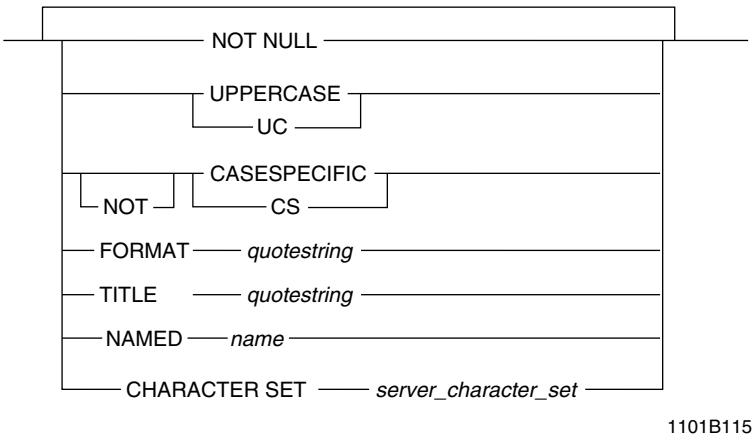
**Data Type**



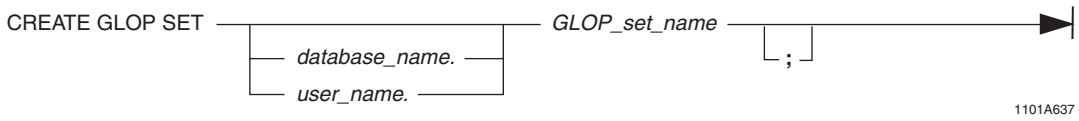

1101A535



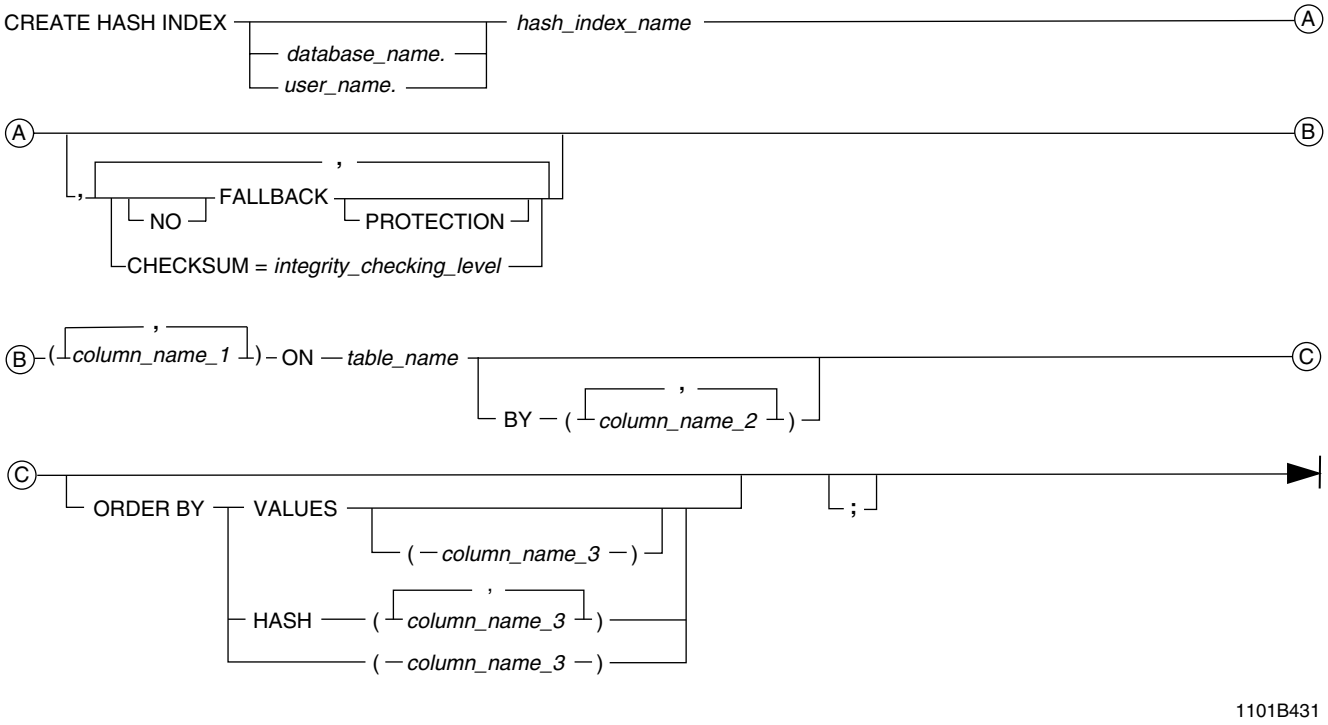

**Data Type Attributes**



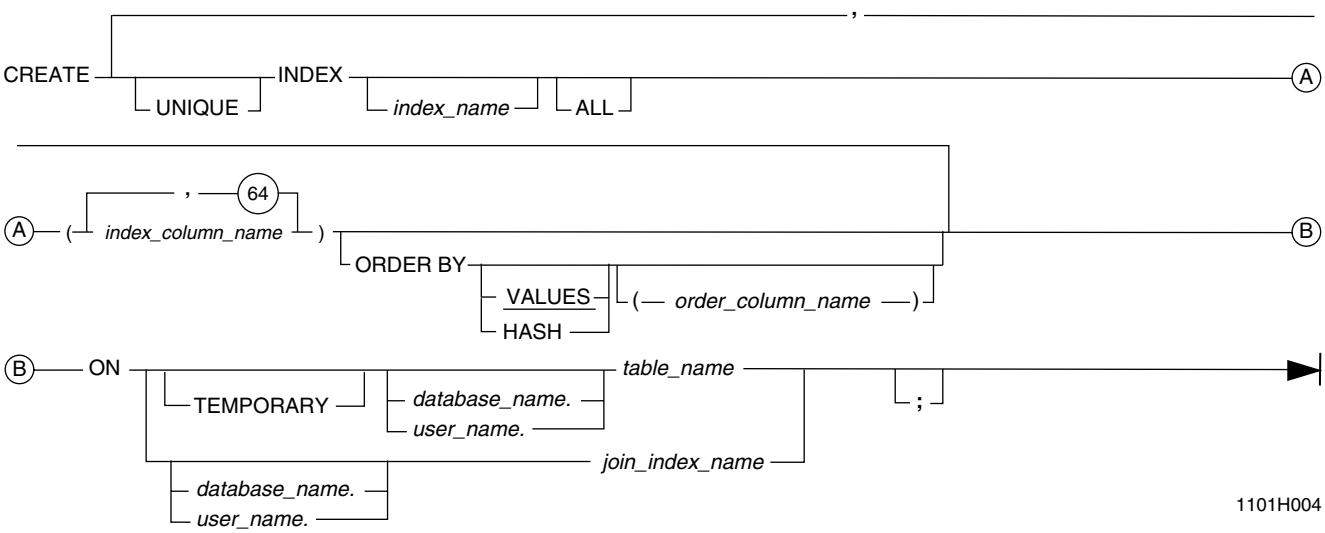
# CREATE GLOP SET



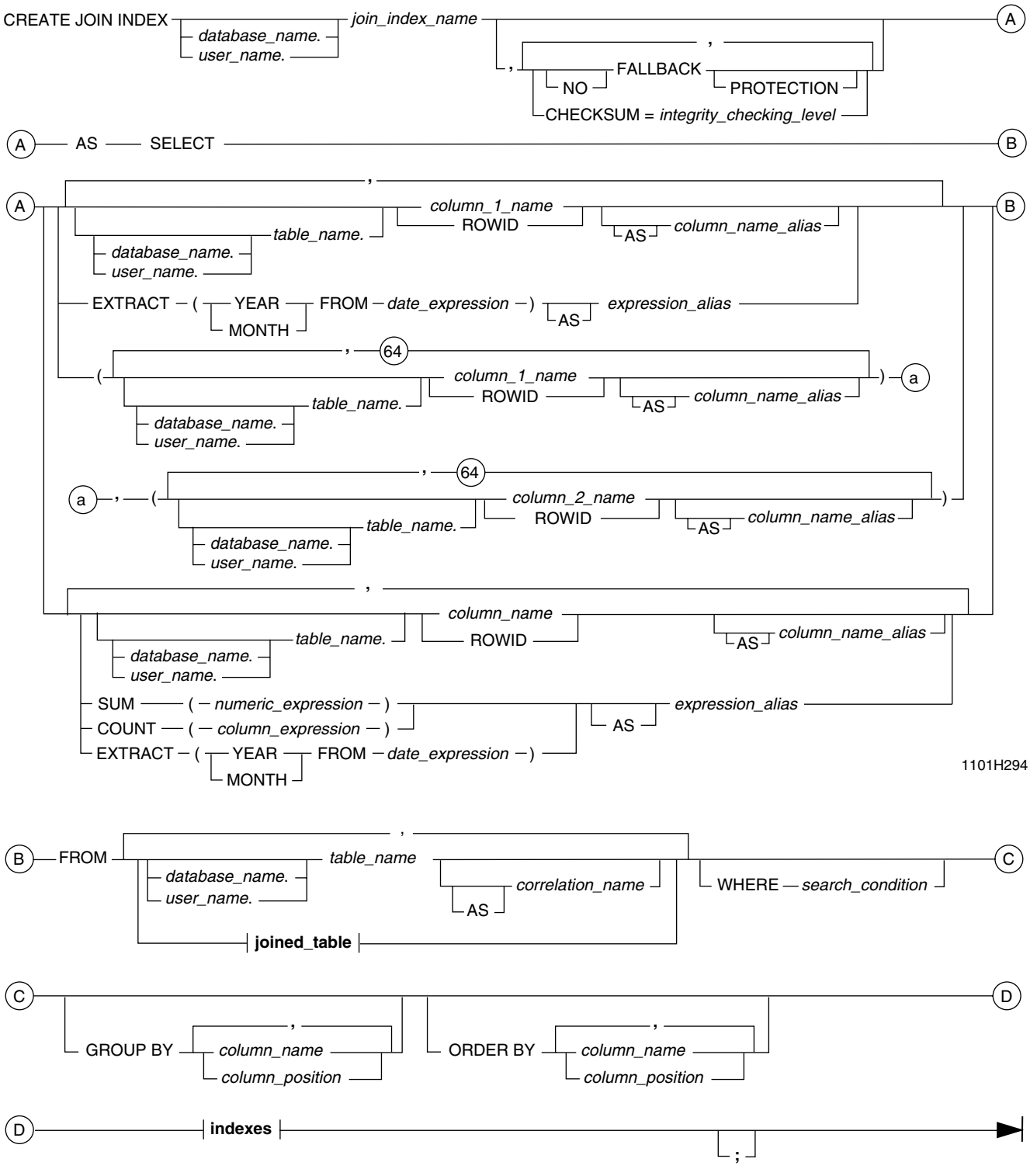

# CREATE HASH INDEX



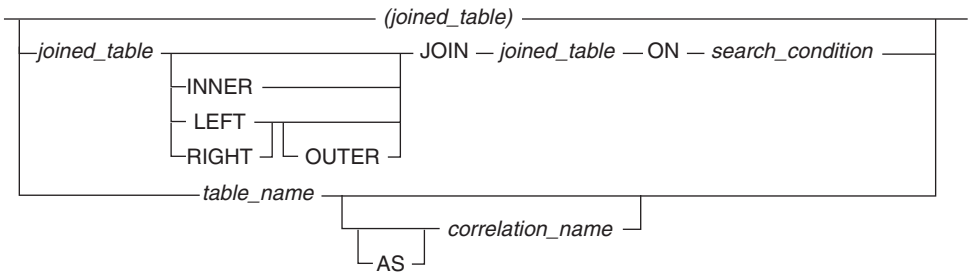
# CREATE INDEX



# CREATE JOIN INDEX

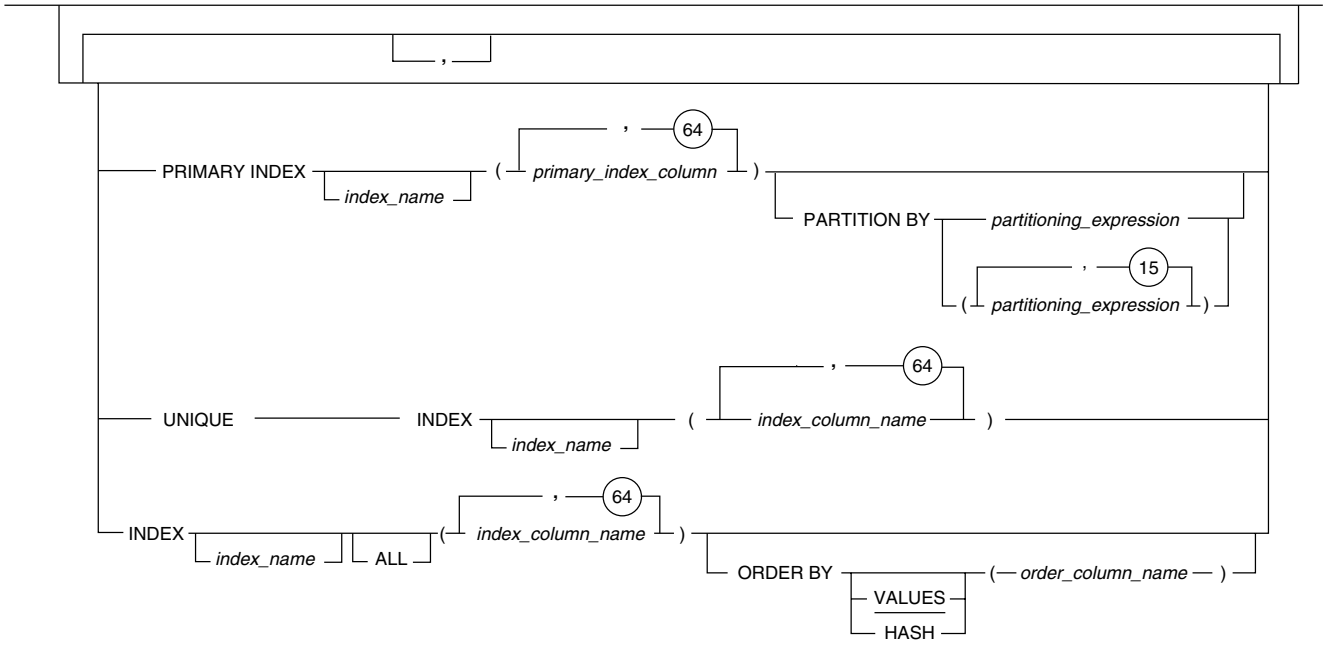


**joined_table**



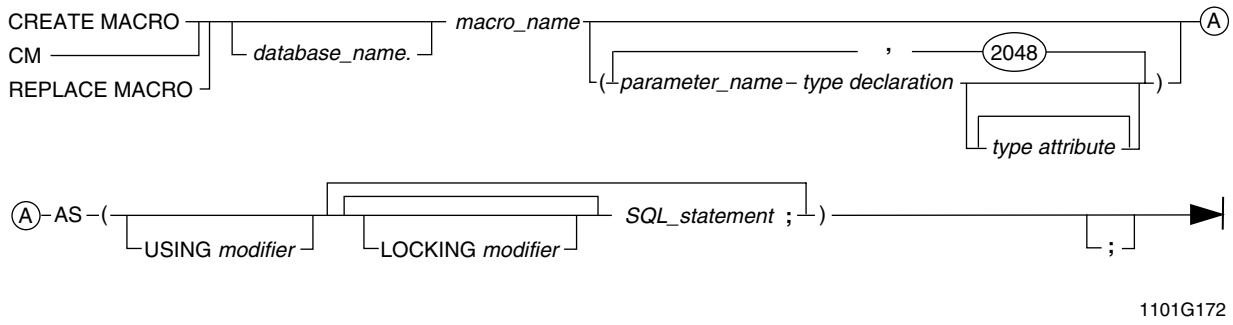

1101E050

**indexes**

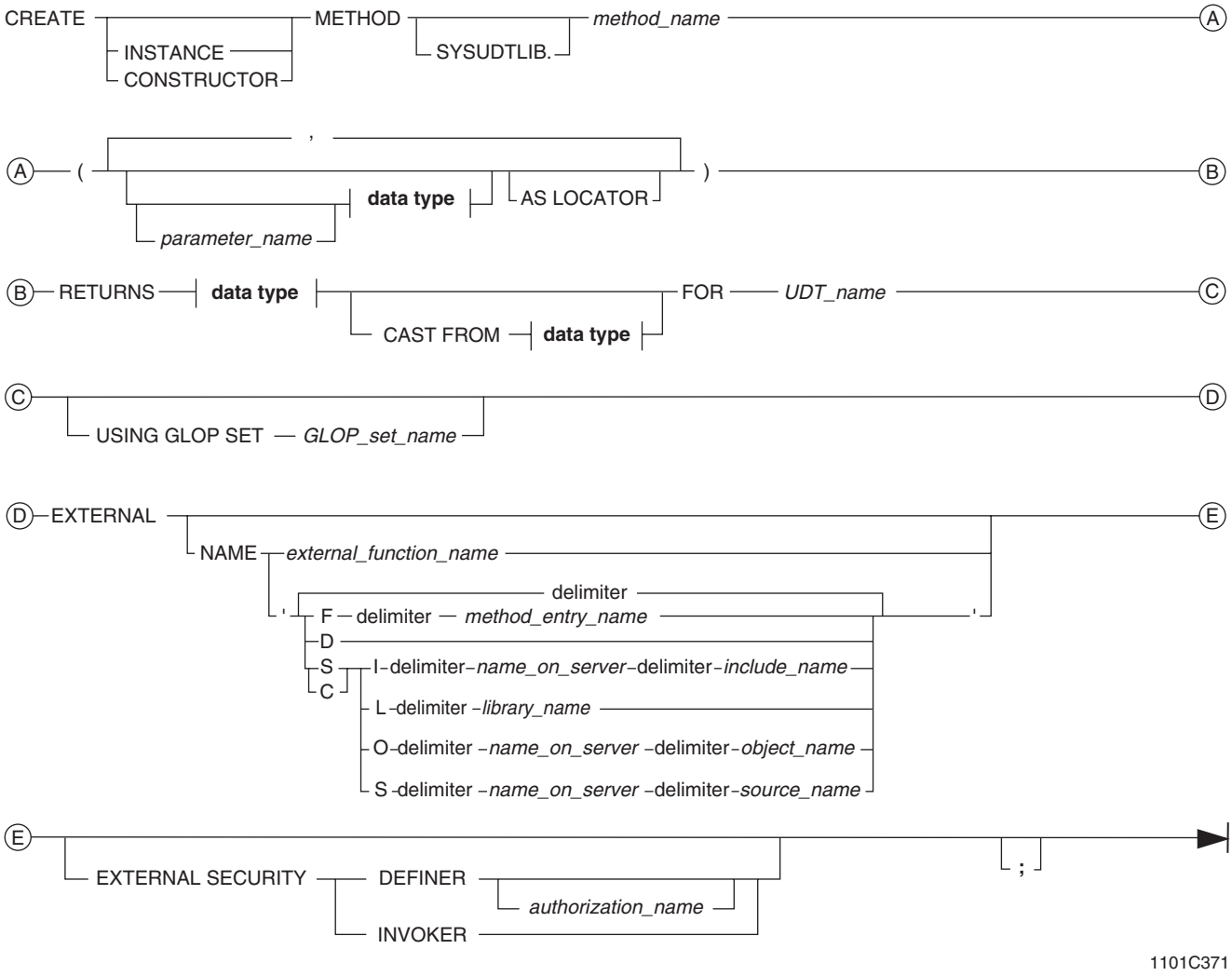



1101I051

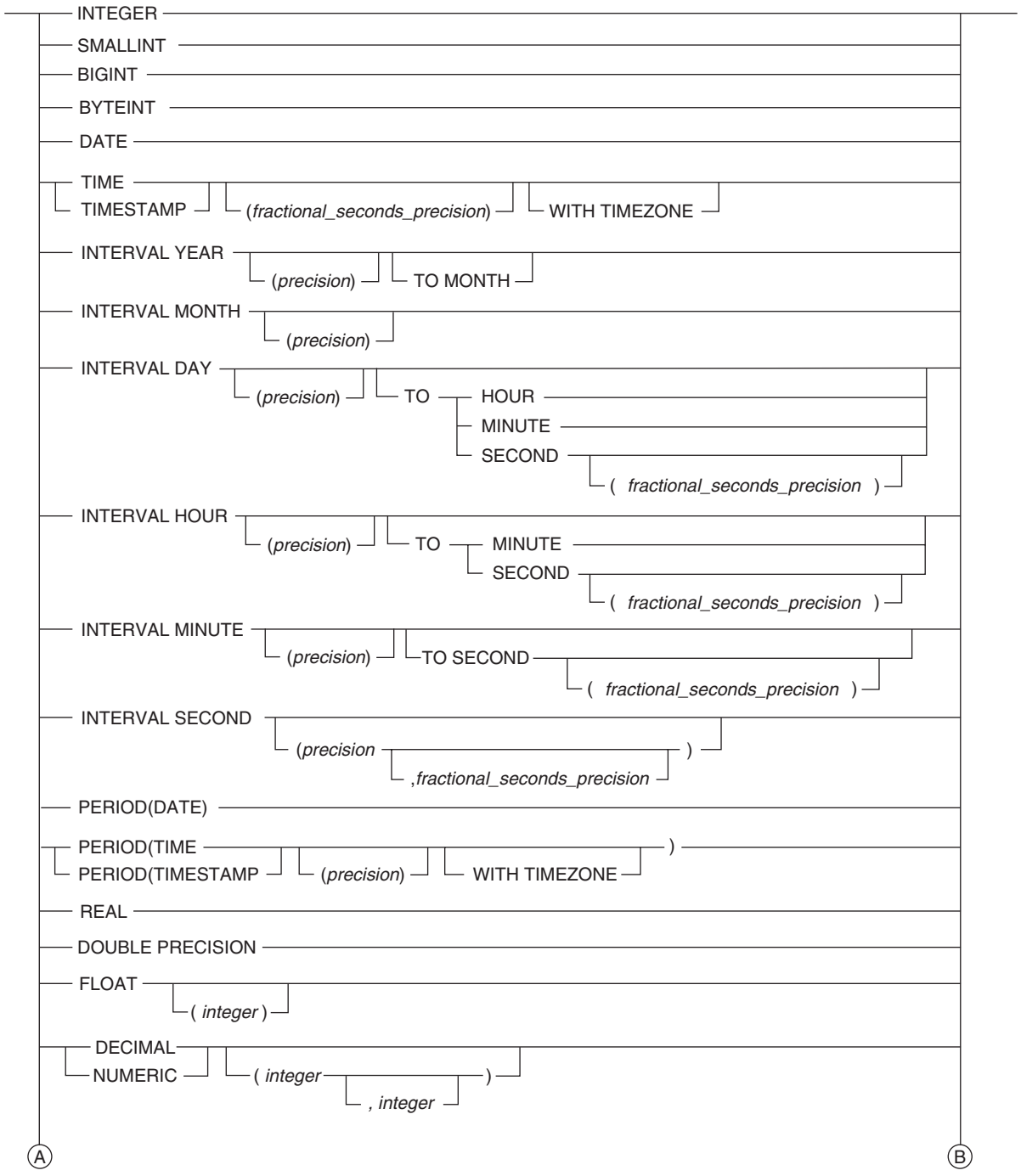
# CREATE MACRO/ REPLACE MACRO



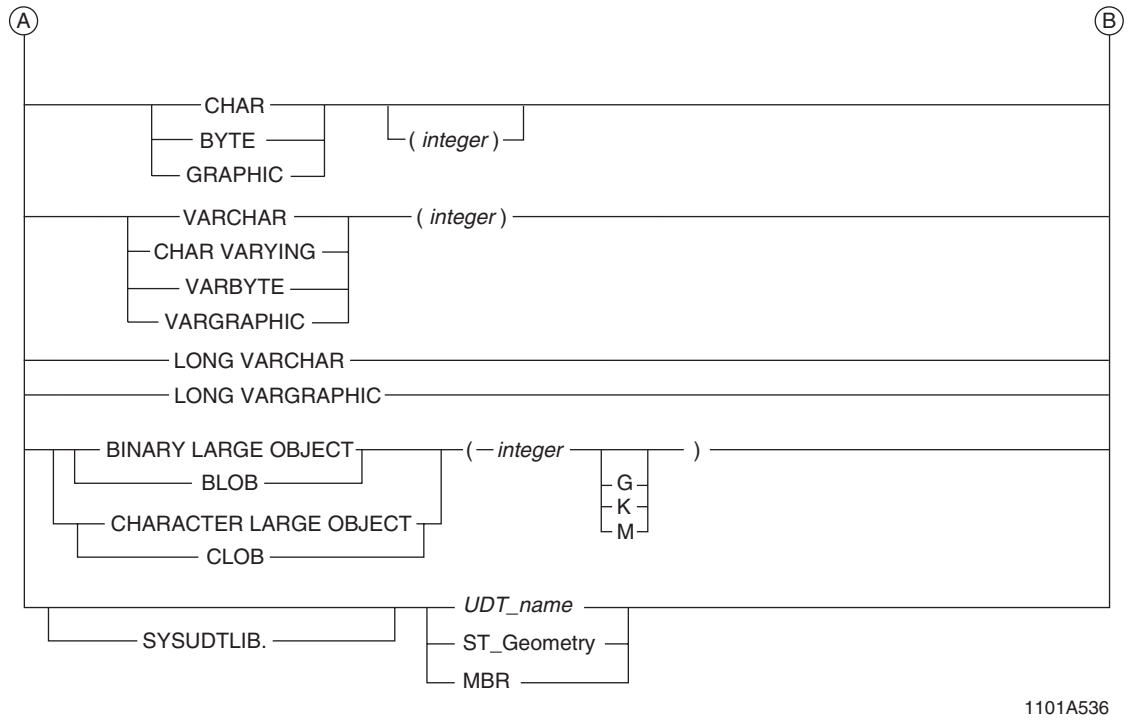
# CREATE METHOD



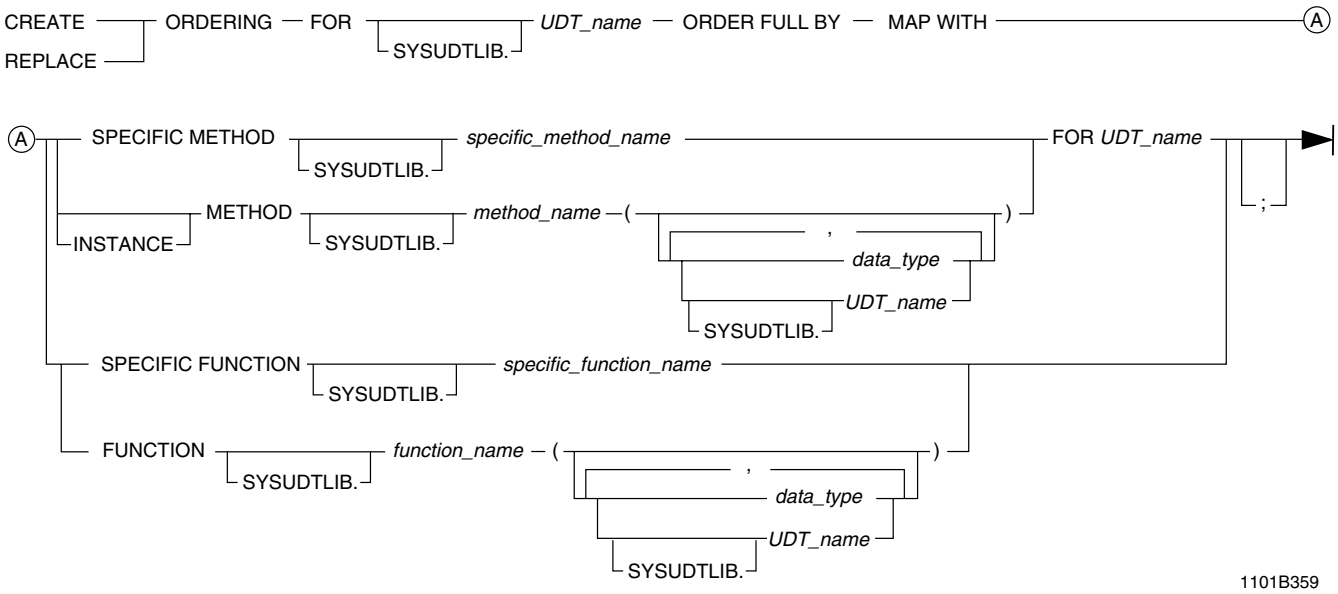
**Data Type**



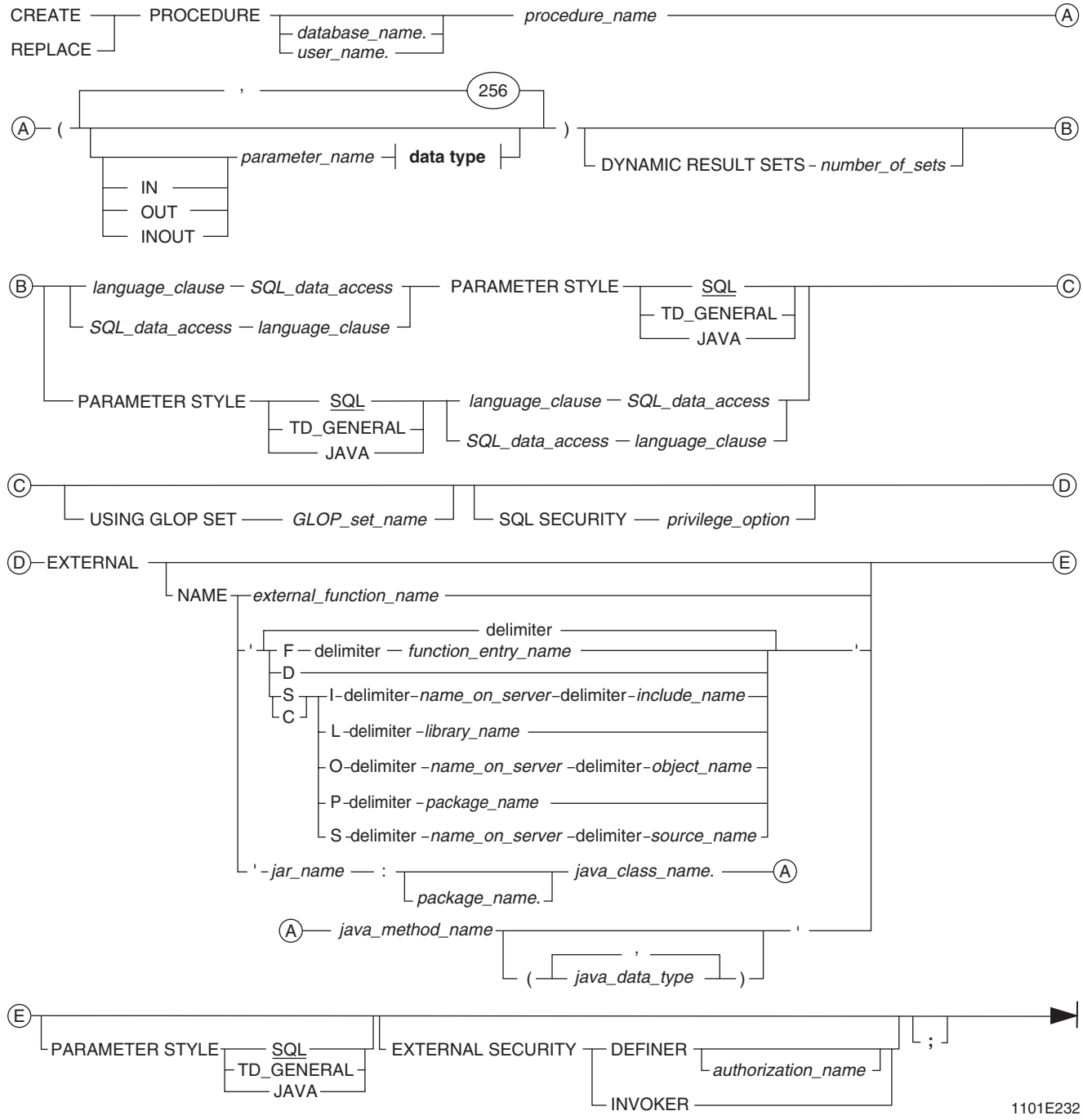

1101A535



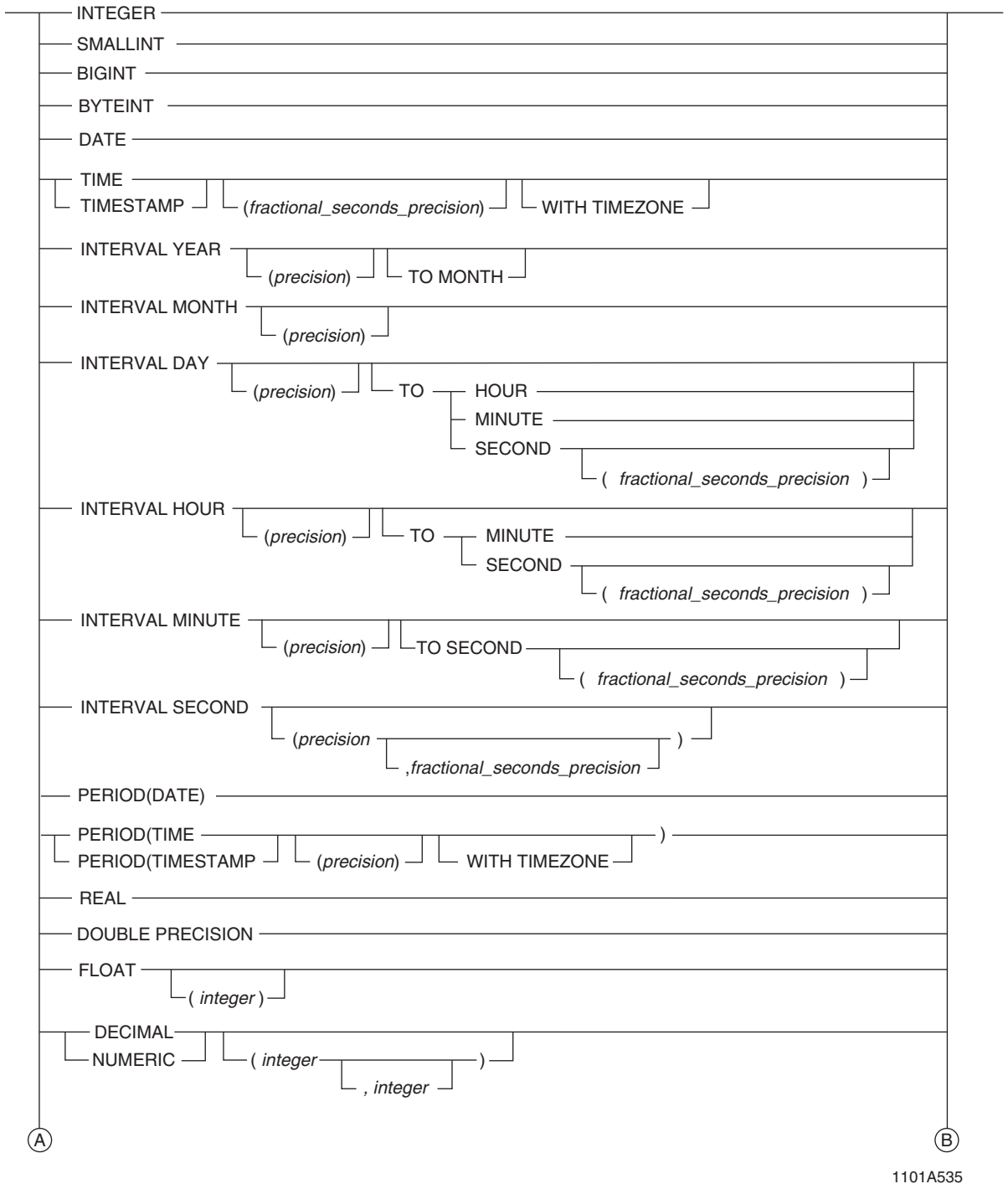

# CREATE ORDERING/ REPLACE ORDERING



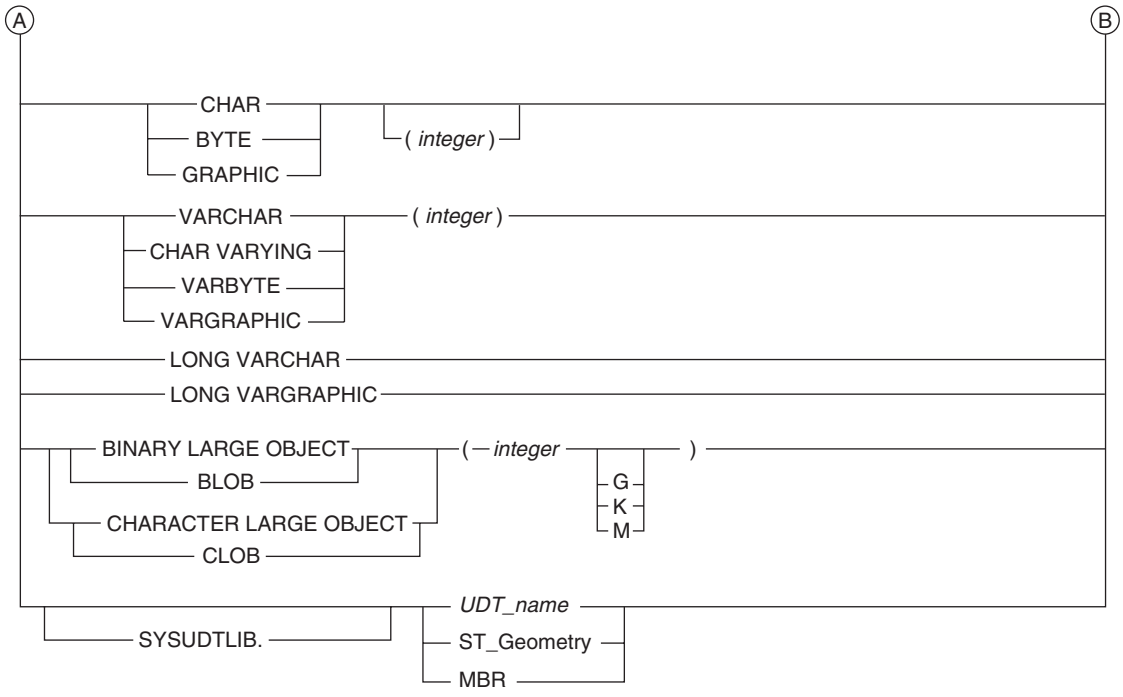
# CREATE PROCEDURE (External Form)/ REPLACE PROCEDURE (External Form)



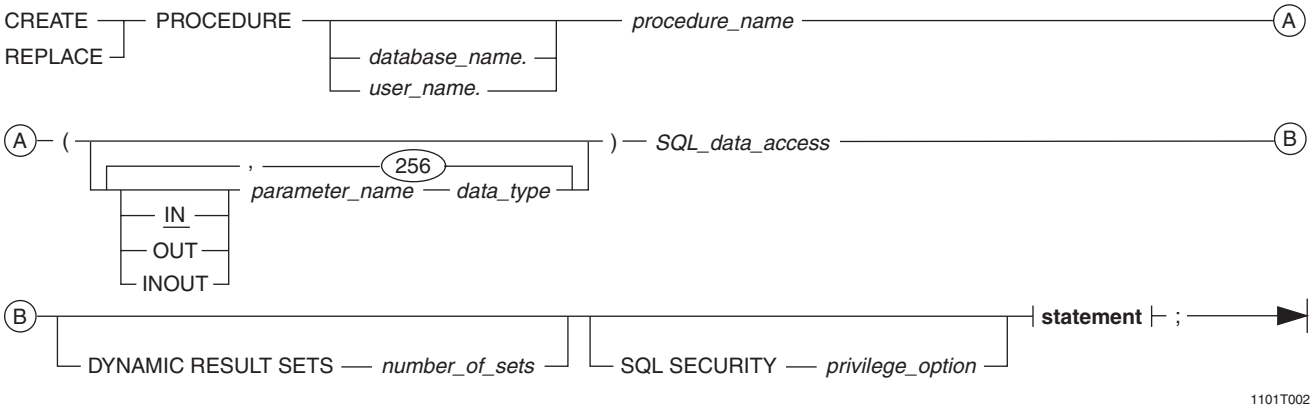
**Data Type**



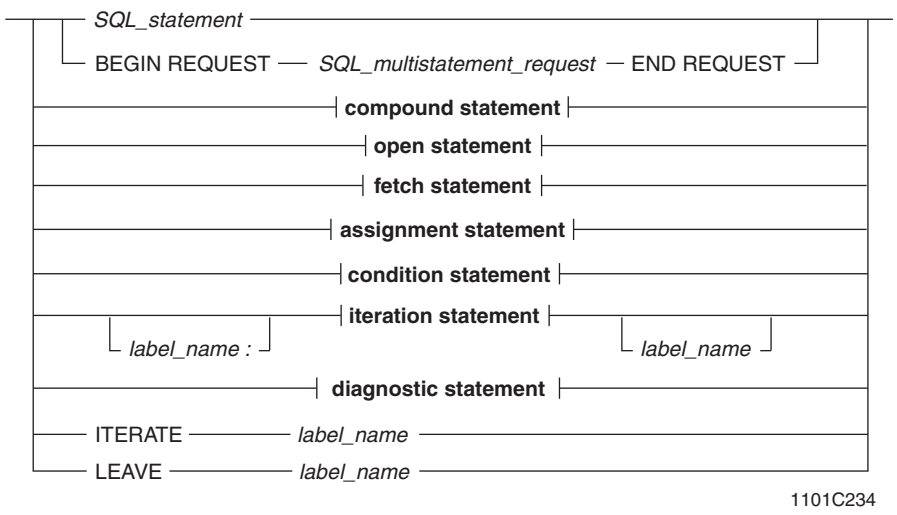

1101A535



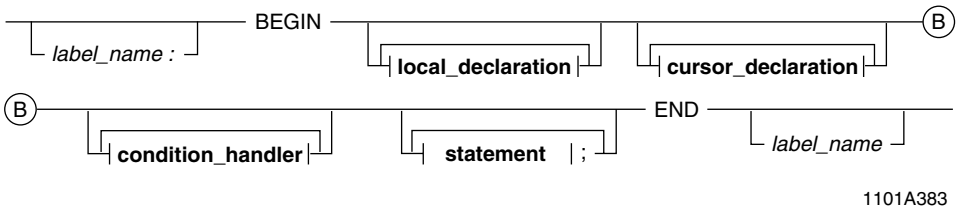

# CREATE PROCEDURE (SQL Form)/ REPLACE PROCEDURE



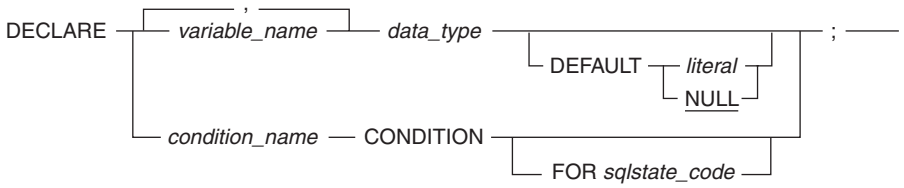
**statement**



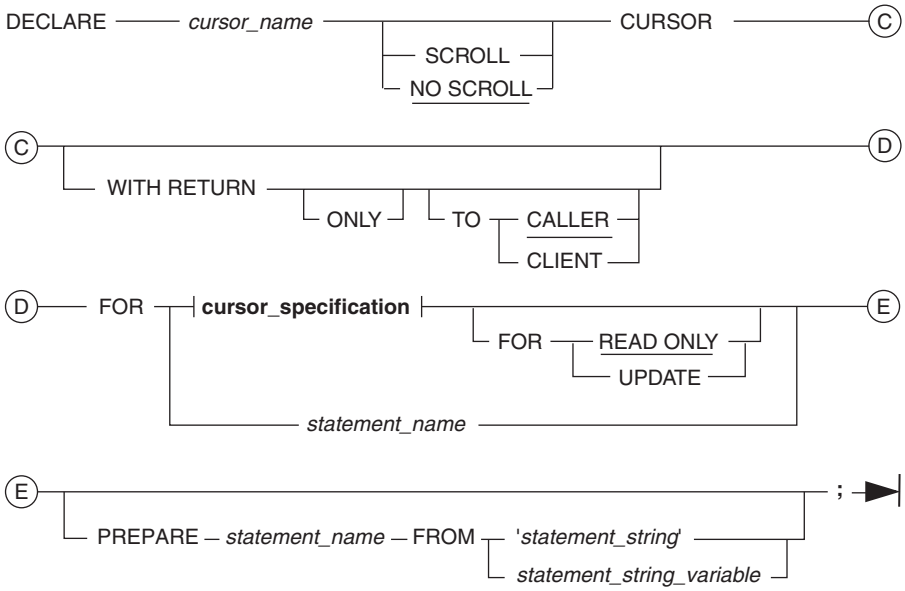

**compound statement**



**local_declaration**

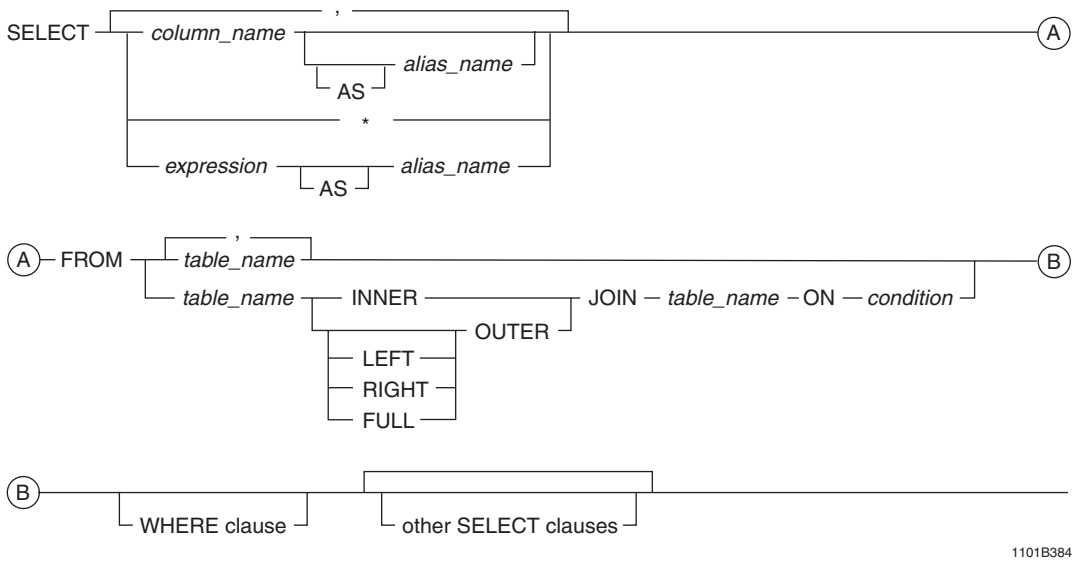



**cursor_declaration**



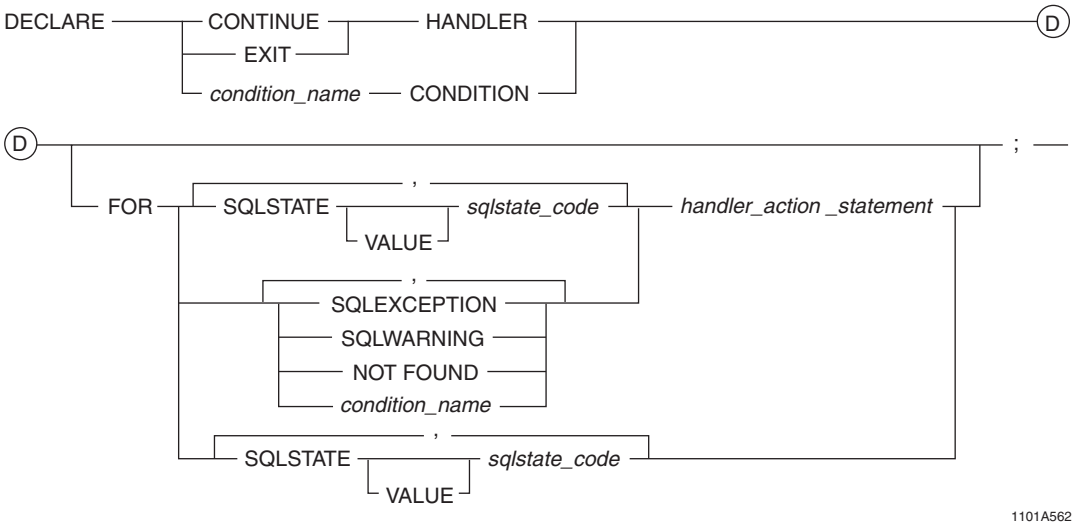

1101C448

**cursor_specification**

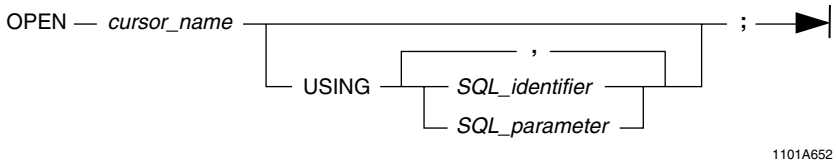



1101B384

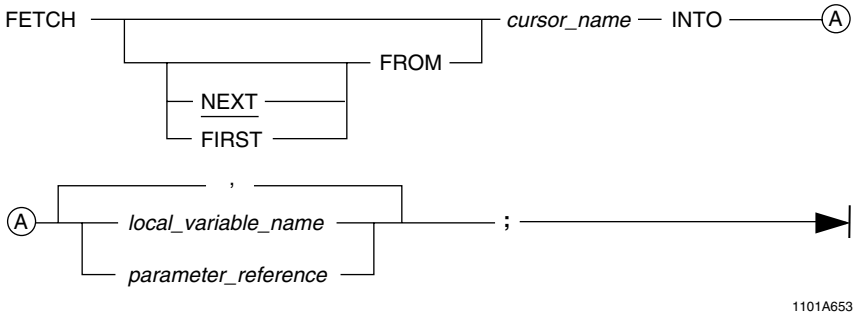
**condition_handler**



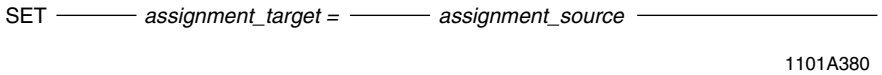

**open statement**



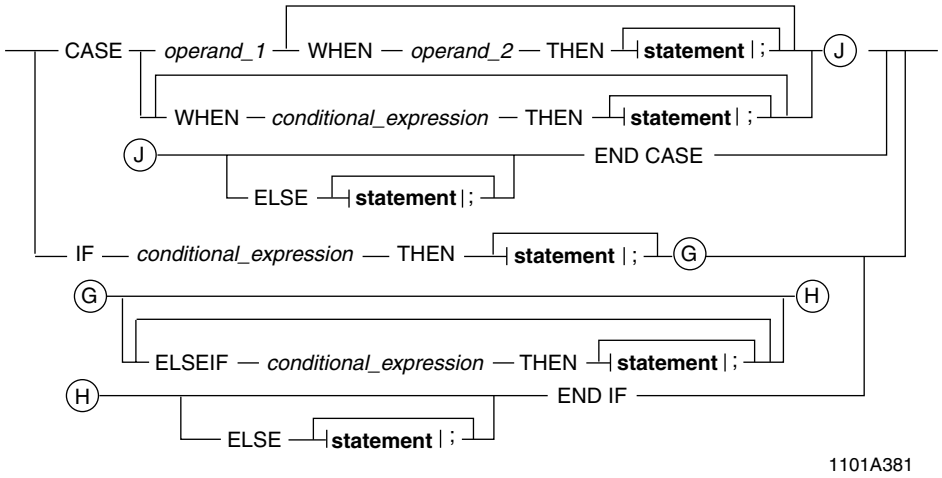

**fetch statement**




**assignment statement**

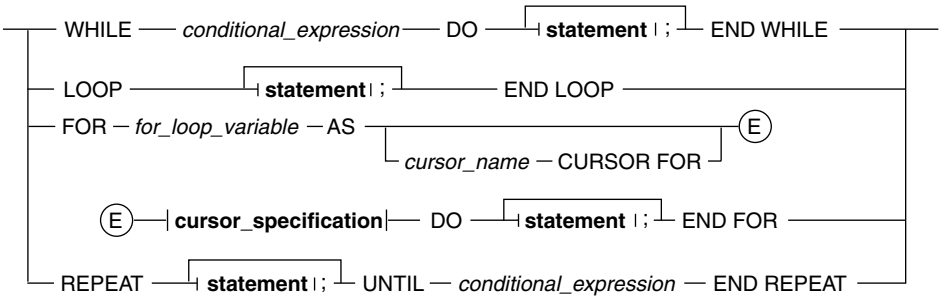


**condition statement**



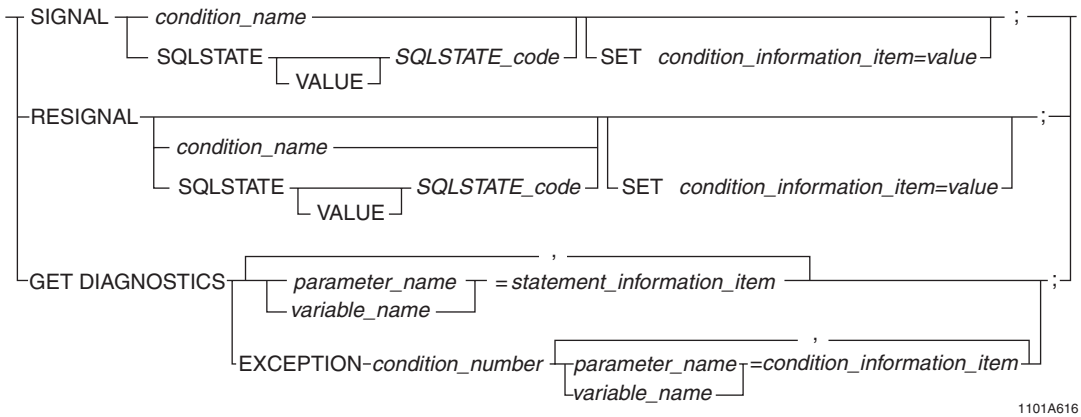

1101A381

**iteration statement**



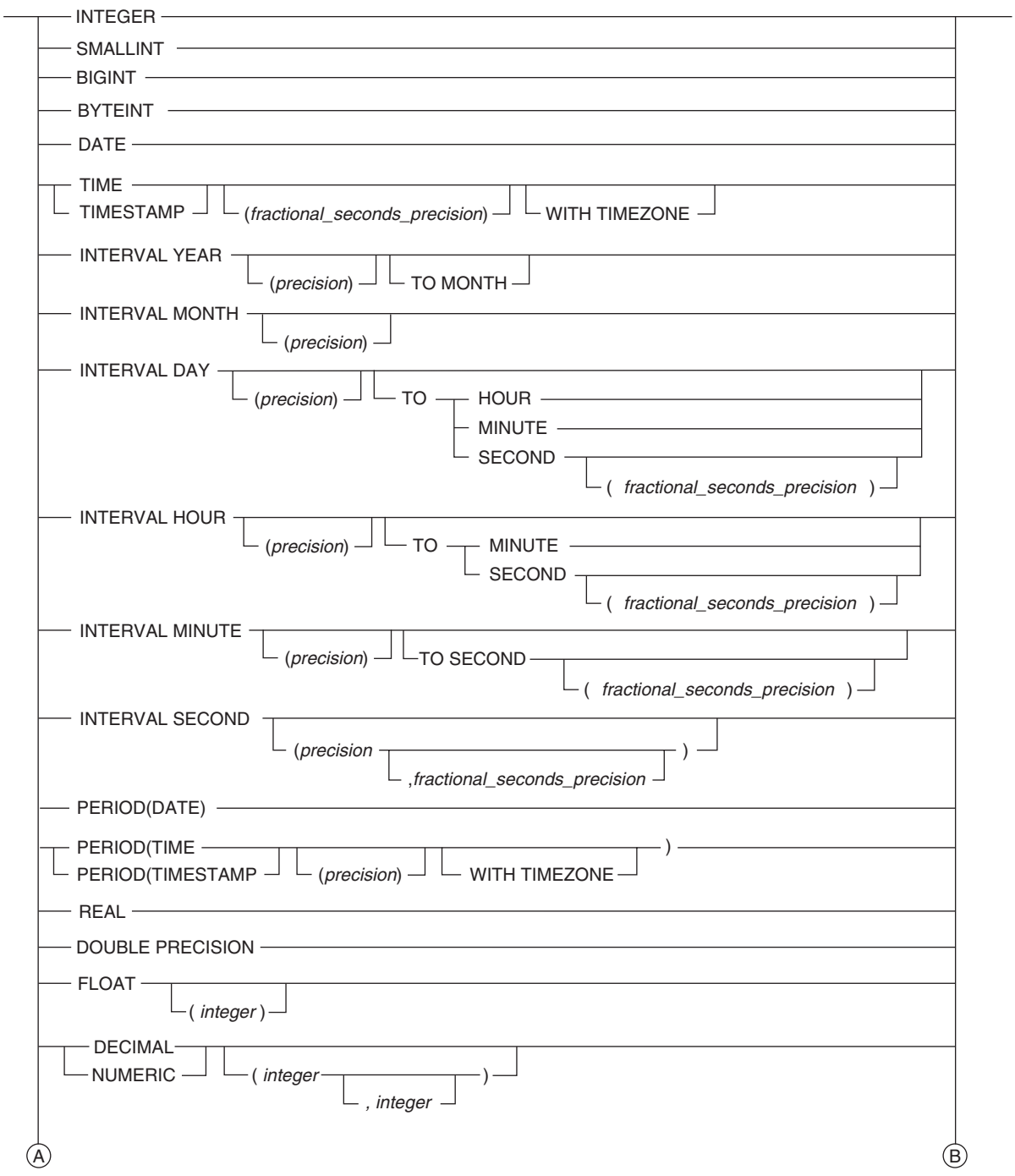

1101A382

**diagnostic statement**

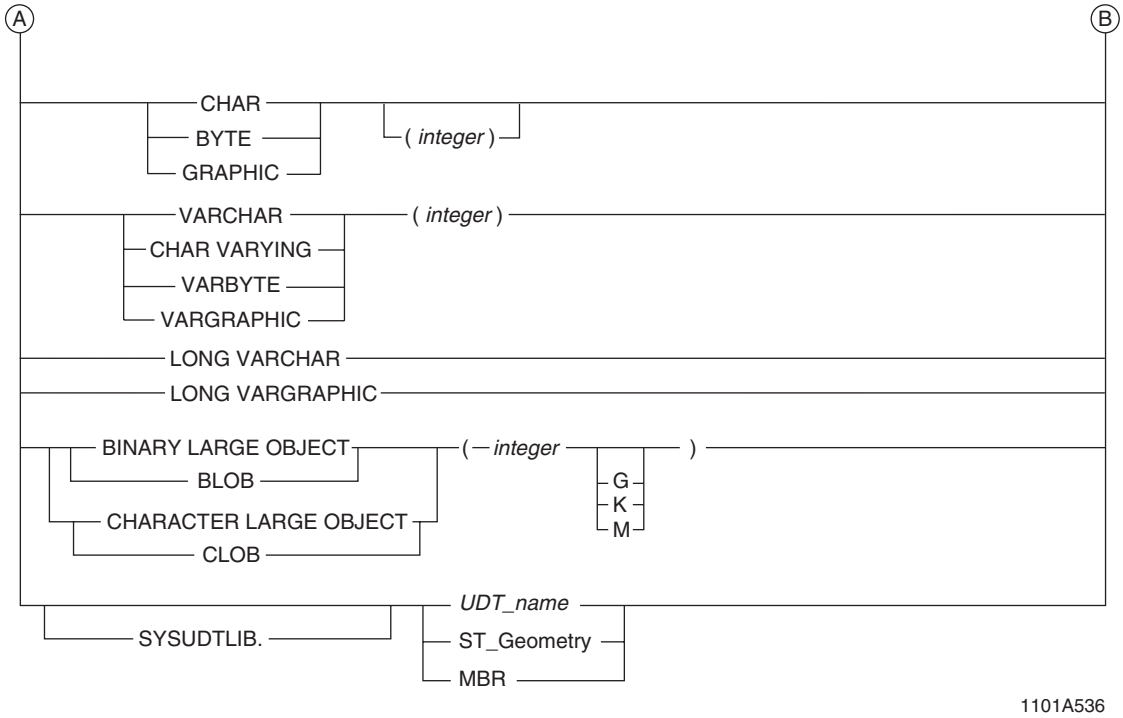



1101A616

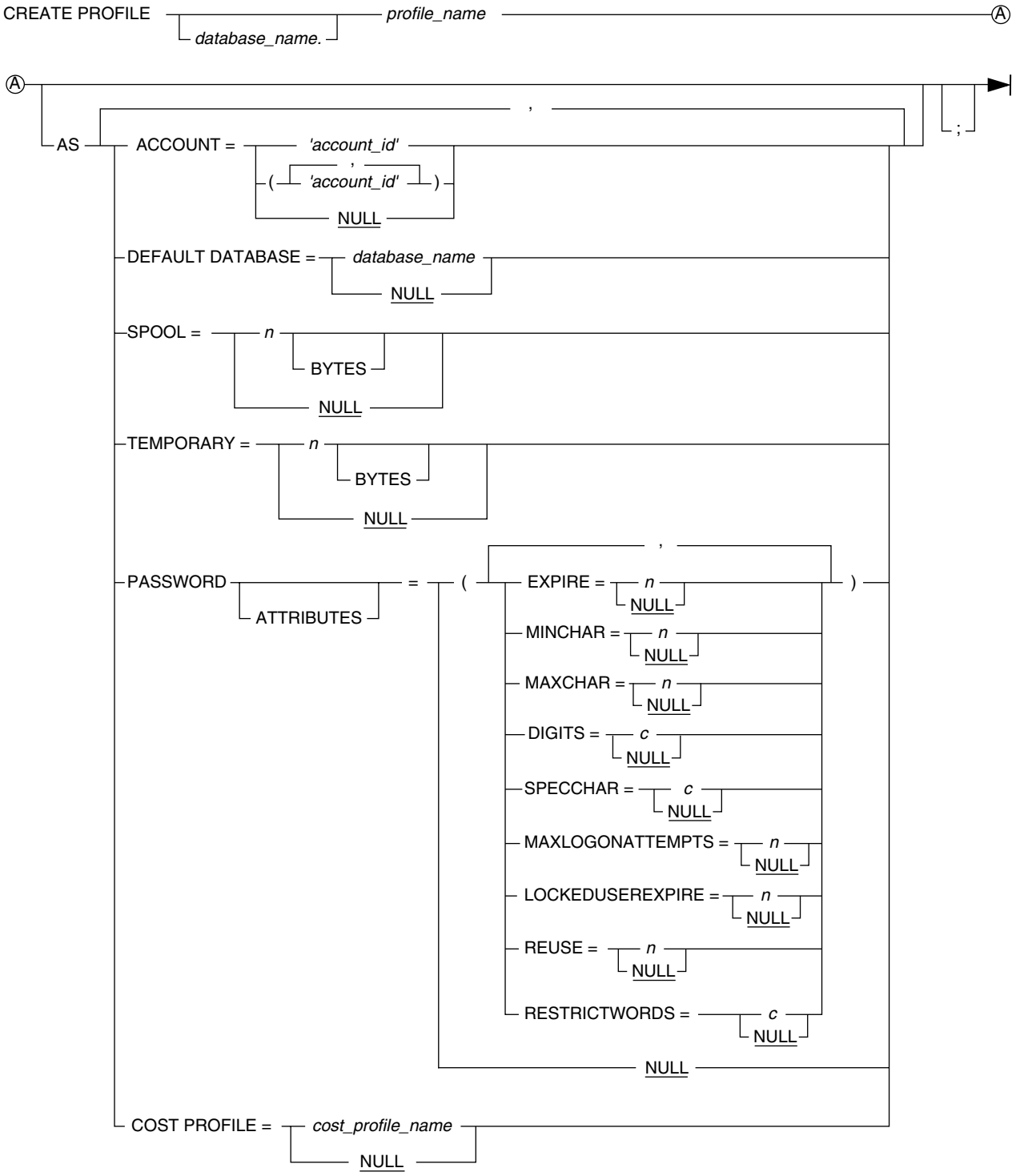
**Data Type**



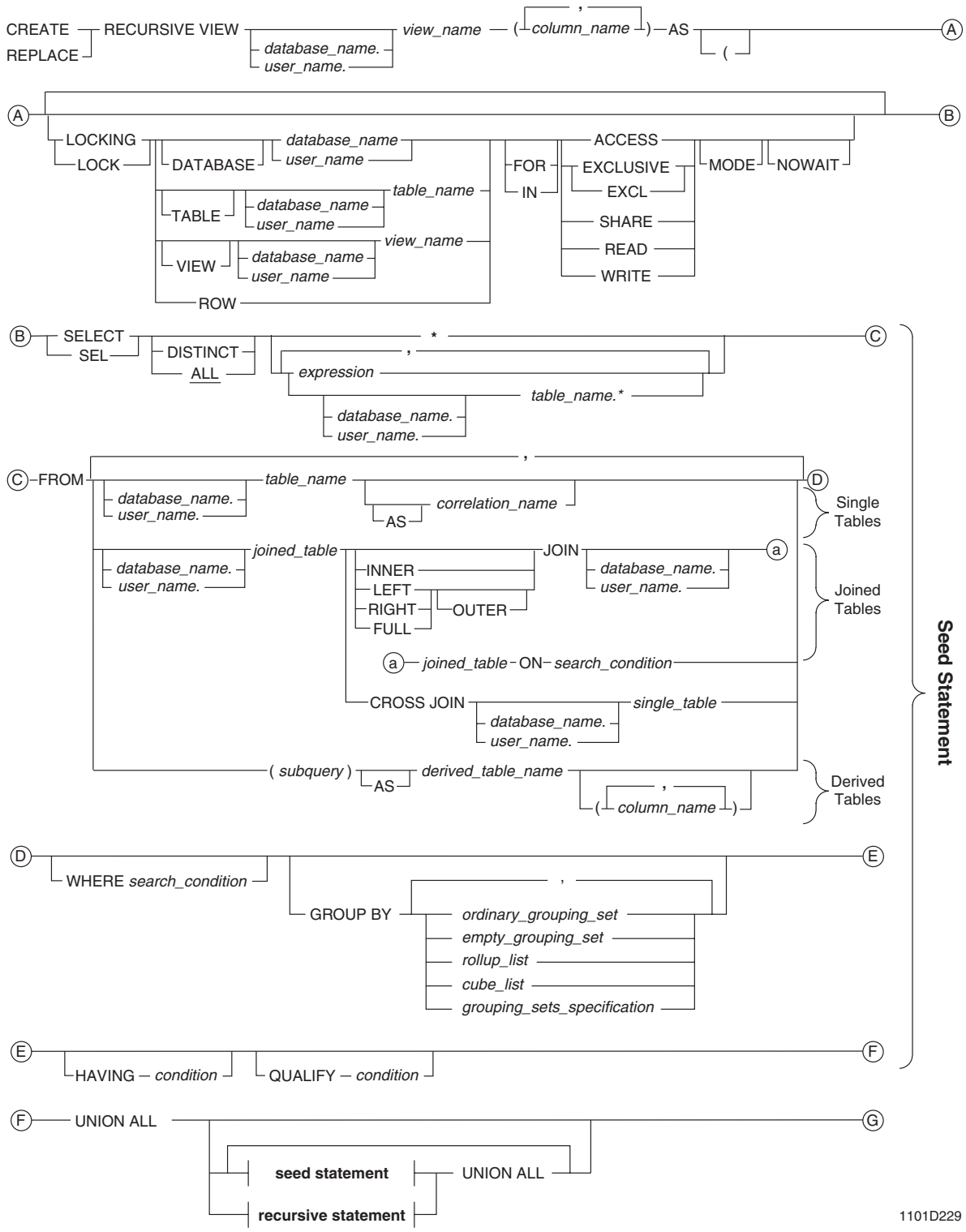

1101A535

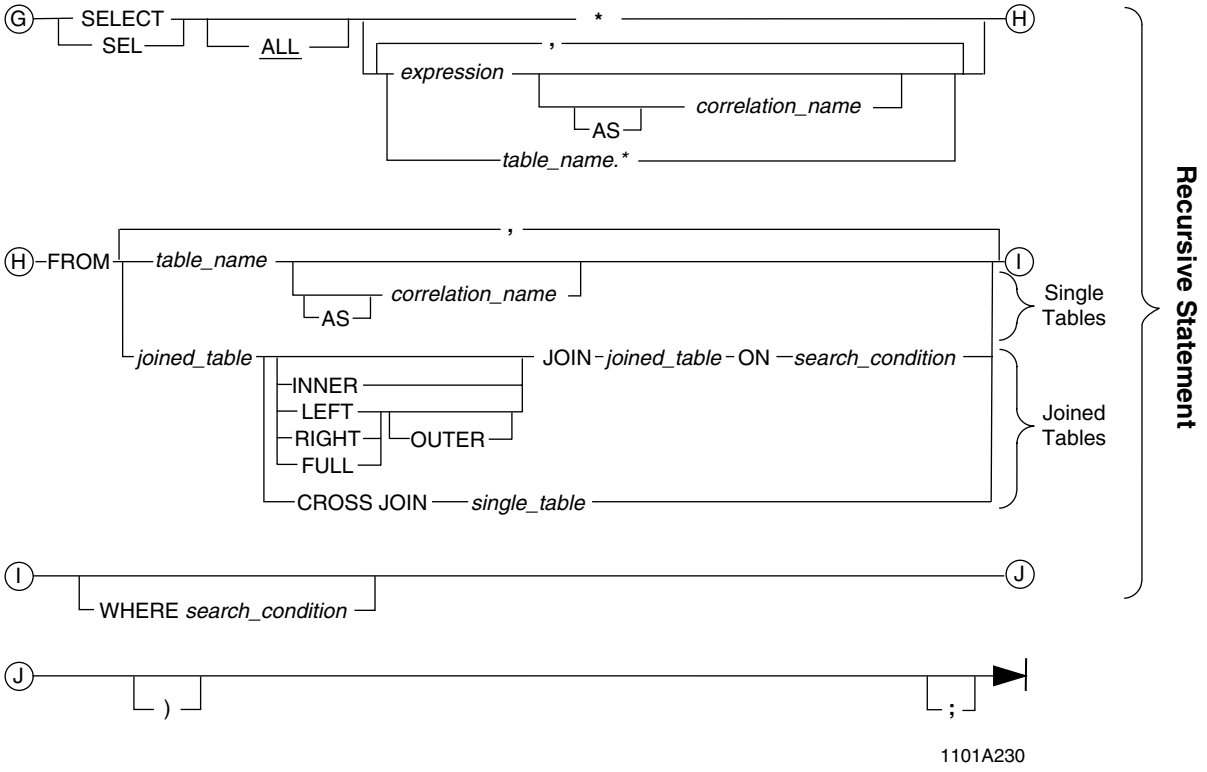

A
B
CHAR
BYTE
GRAPHIC
( integer )
VARCHAR
CHAR VARYING
VARBYTE
VARGRAPHIC
( integer )
LONG VARCHAR
LONG VARGRAPHIC
BINARY LARGE OBJECT
BLOB
CHARACTER LARGE OBJECT
CLOB
( integer
G
K
M
)
SYSUDTLIB.
UDT_name
ST_Geometry
MBR
1101A536

# CREATE PROFILE

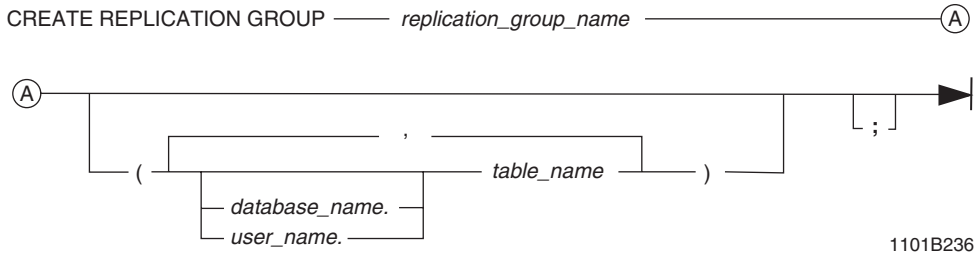


# CREATE RECURSIVE VIEW/ REPLACE RECURSIVE VIEW

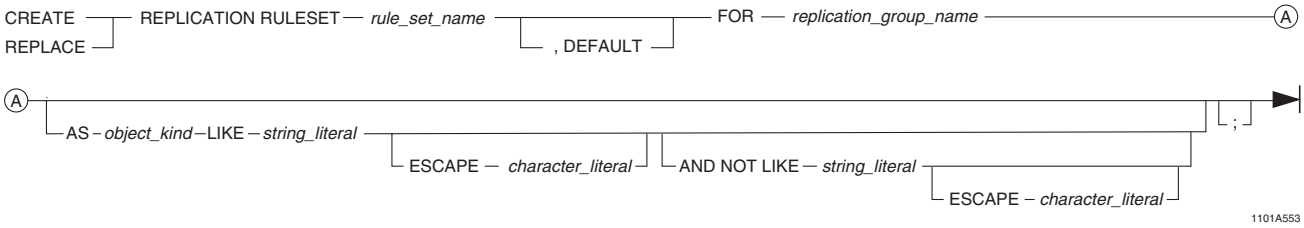




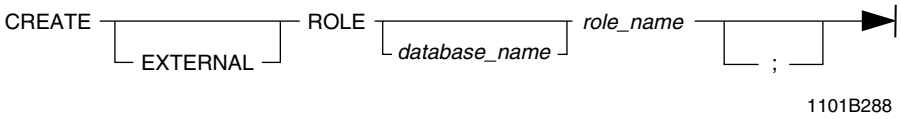

# CREATE REPLICATION GROUP



# CREATE REPLICATION RULESET/ REPLACE REPLICATION RULESET

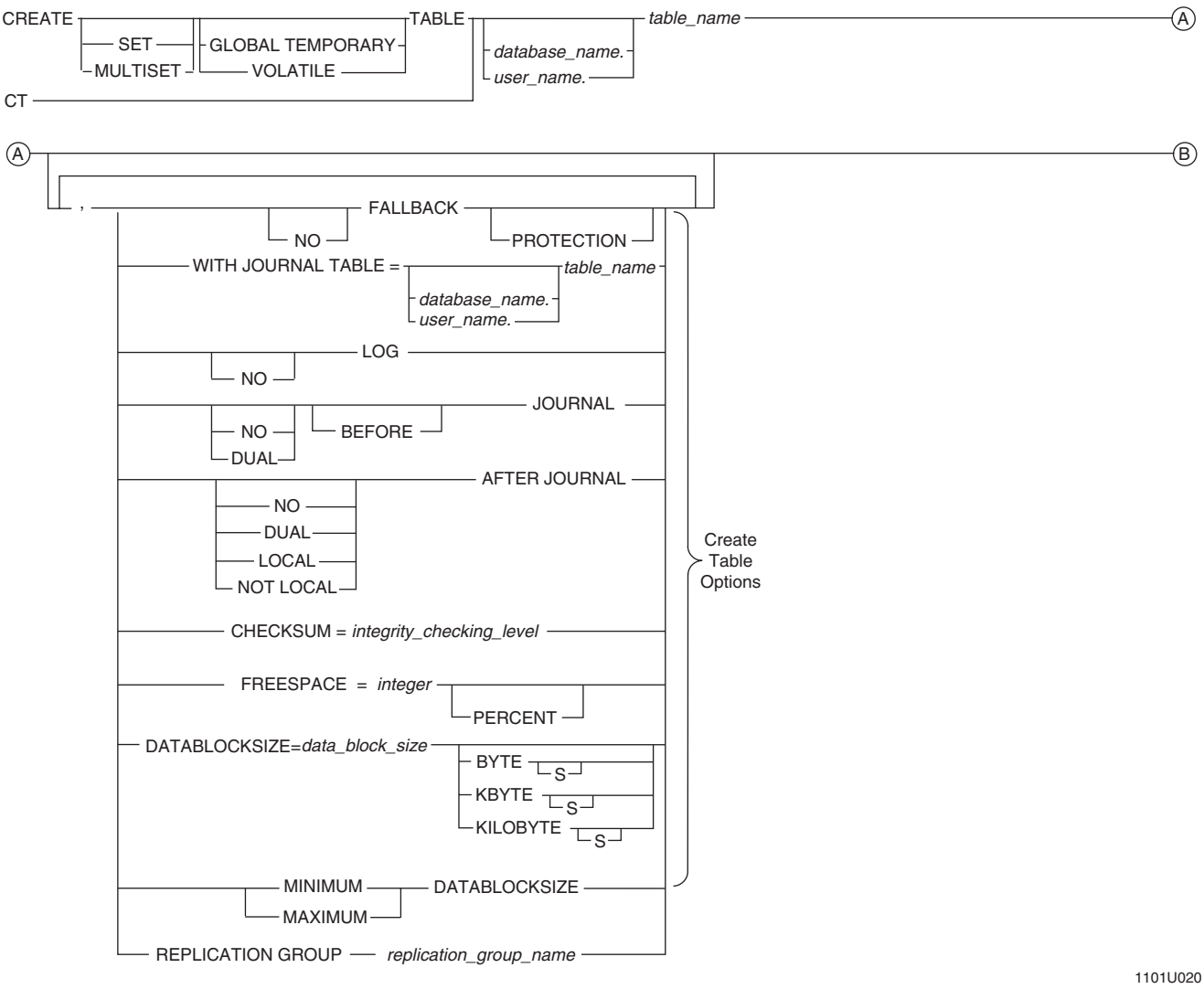


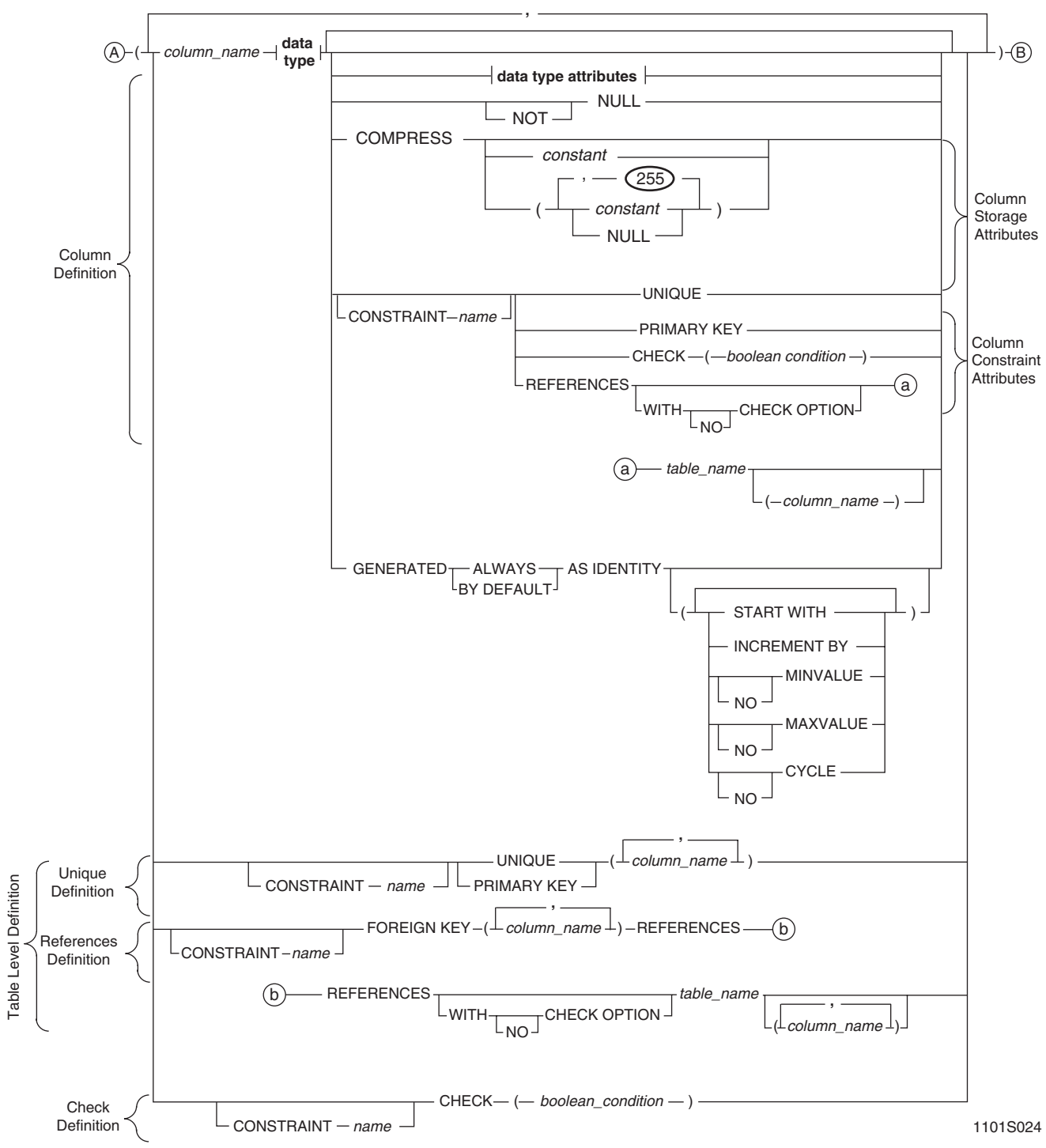

# CREATE ROLE

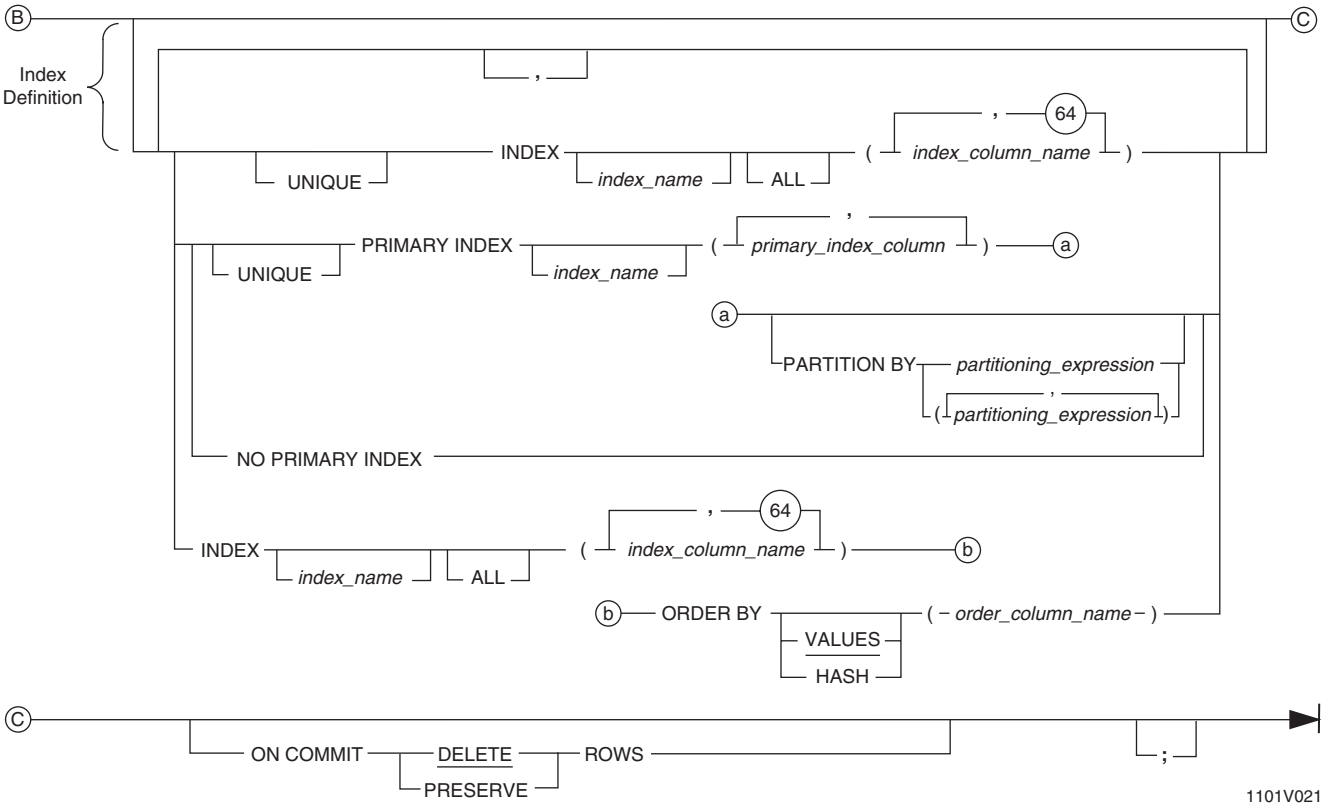


# CREATE TABLE

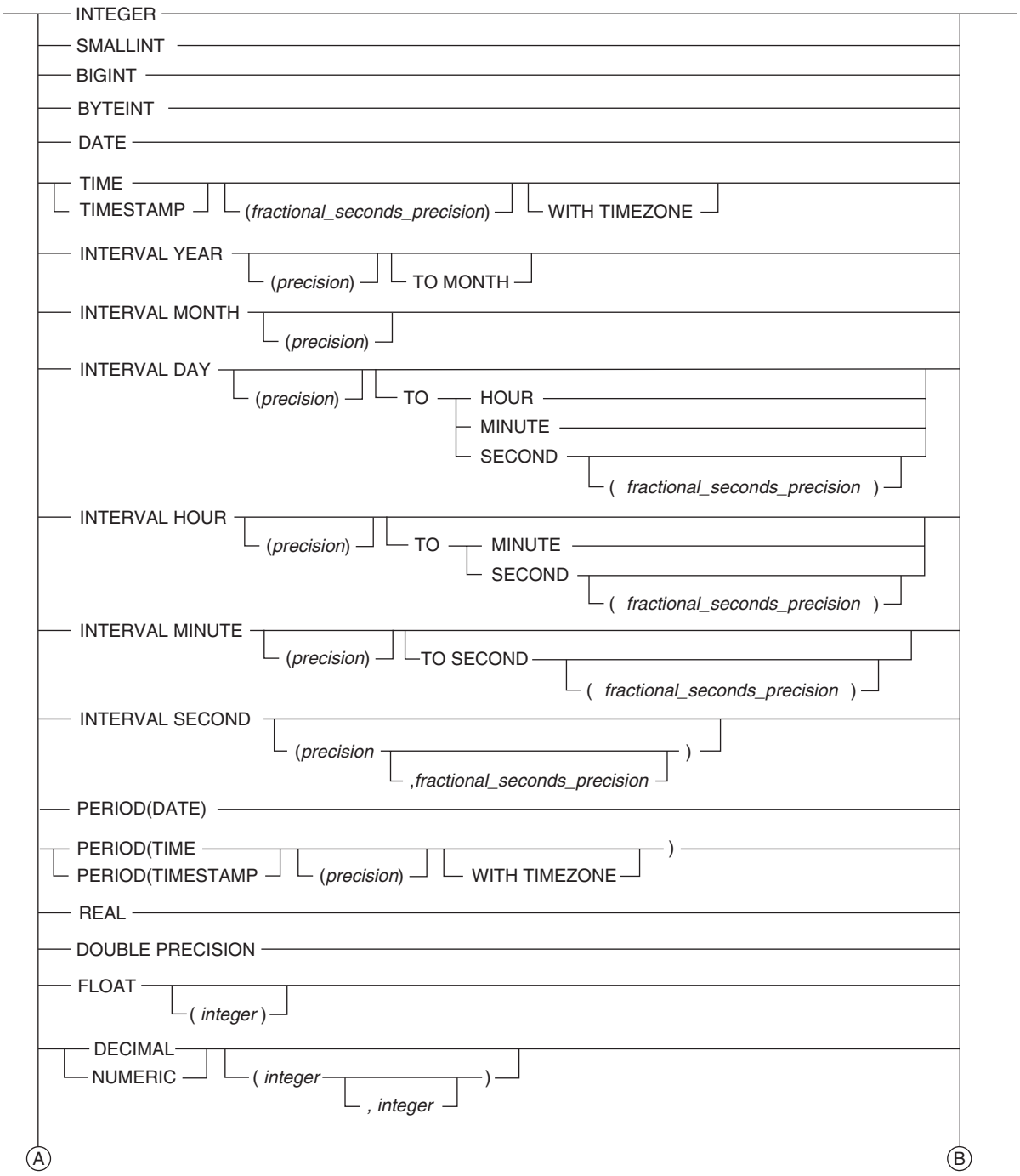
## Create Table Syntax



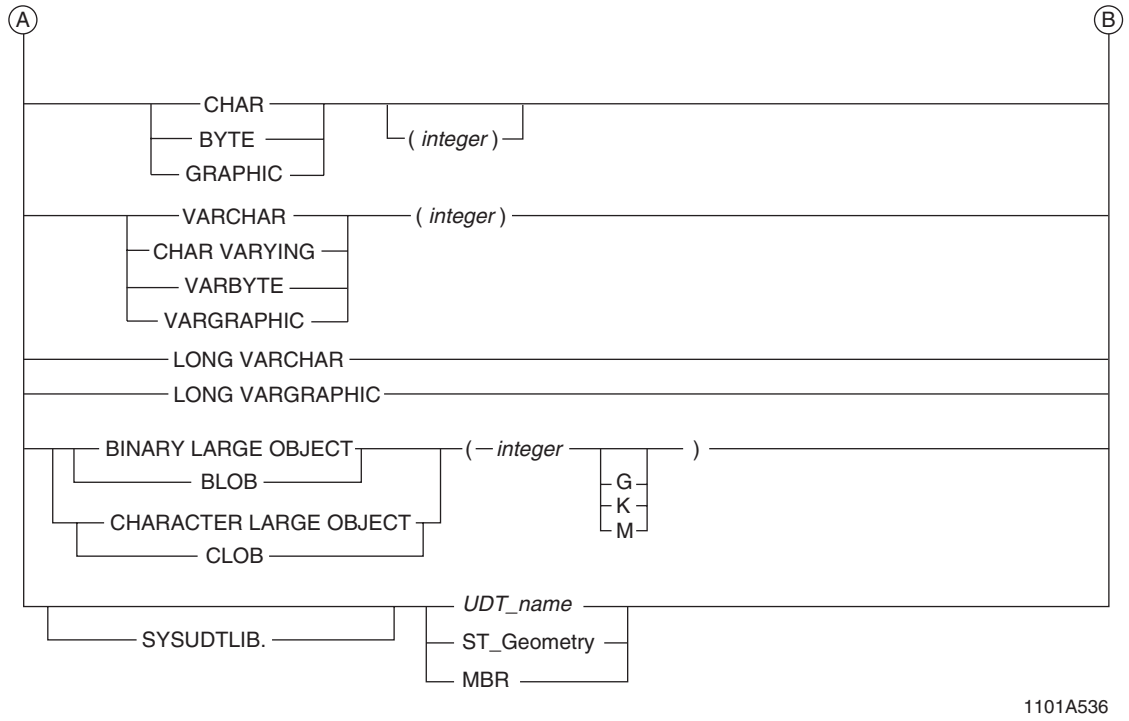

A
B
column_name
data type
data type attributes
NOT
NULL
COMPRESS
constant
255
NULL
Column Definition
Column Storage Attributes
UNIQUE
CONSTRAINT—name
PRIMARY KEY
CHECK—(—boolean condition—)
REFERENCES
WITH
NO
CHECK OPTION
a
table_name
column_name
GENERATED
ALWAYS
BY DEFAULT
AS IDENTITY
START WITH
INCREMENT BY
MINVALUE
MAXVALUE
CYCLE
Column Constraint Attributes
Table Level Definition
Unique Definition
References Definition
Check Definition
FOREIGN KEY
b
CHECK
boolean_condition
1101S024


B
C
Index Definition
,
UNIQUE
INDEX
index_name
ALL
(
index_column_name
,
64
)
UNIQUE
PRIMARY INDEX
index_name
(
primary_index_column
,
)
a
a
PARTITION BY
partitioning_expression
(
partitioning_expression
,
)
NO PRIMARY INDEX
INDEX
index_name
ALL
(
index_column_name
,
64
)
b
b
ORDER BY
(
order_column_name
)
VALUES
HASH
C
ON COMMIT
DELETE
PRESERVE
ROWS
;
1101V021

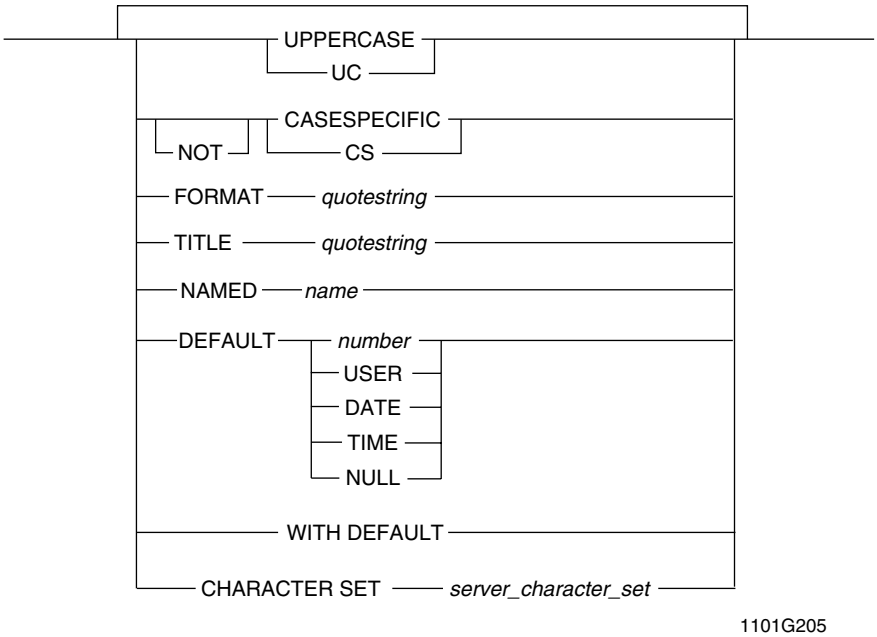
**Data Type**



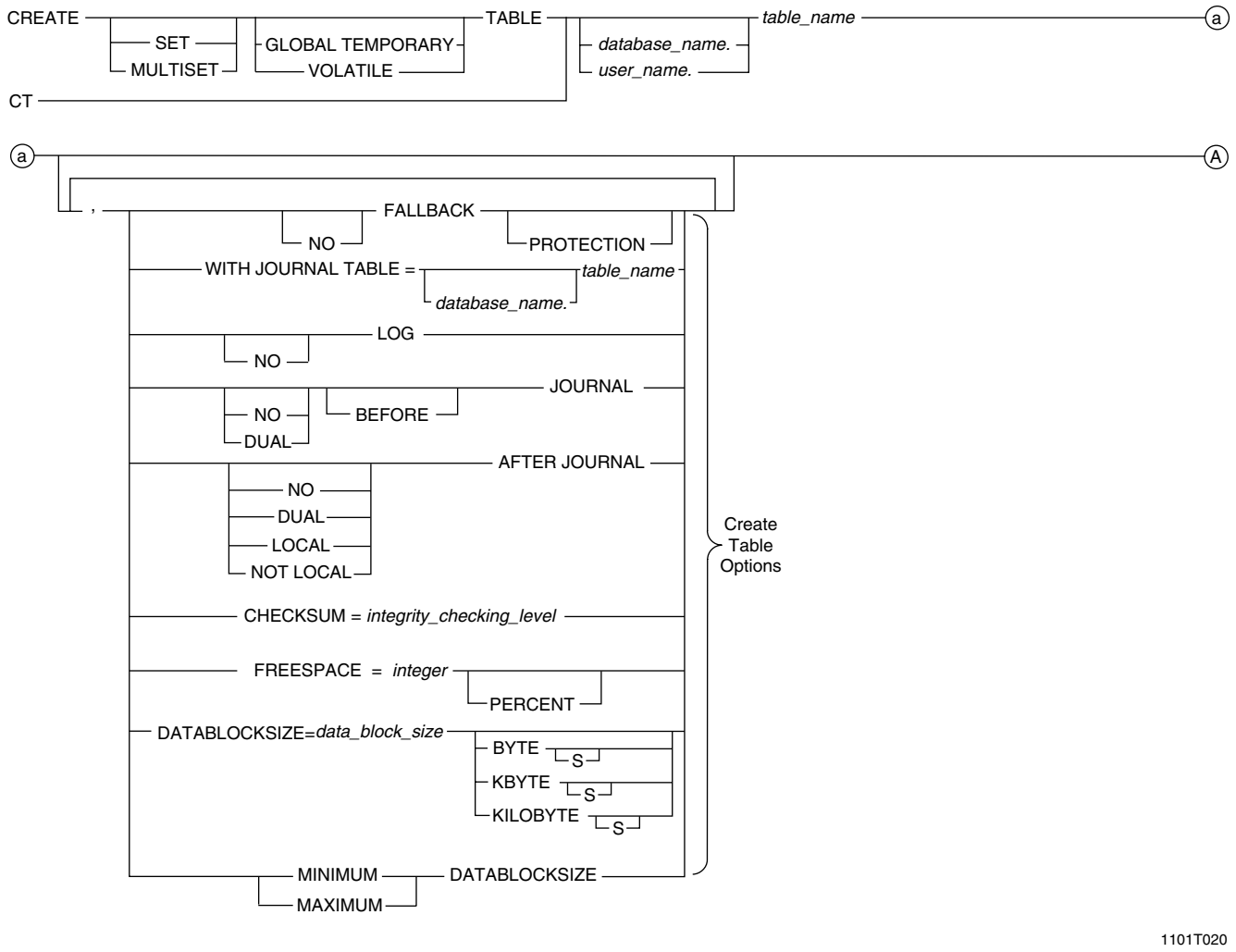

1101A535

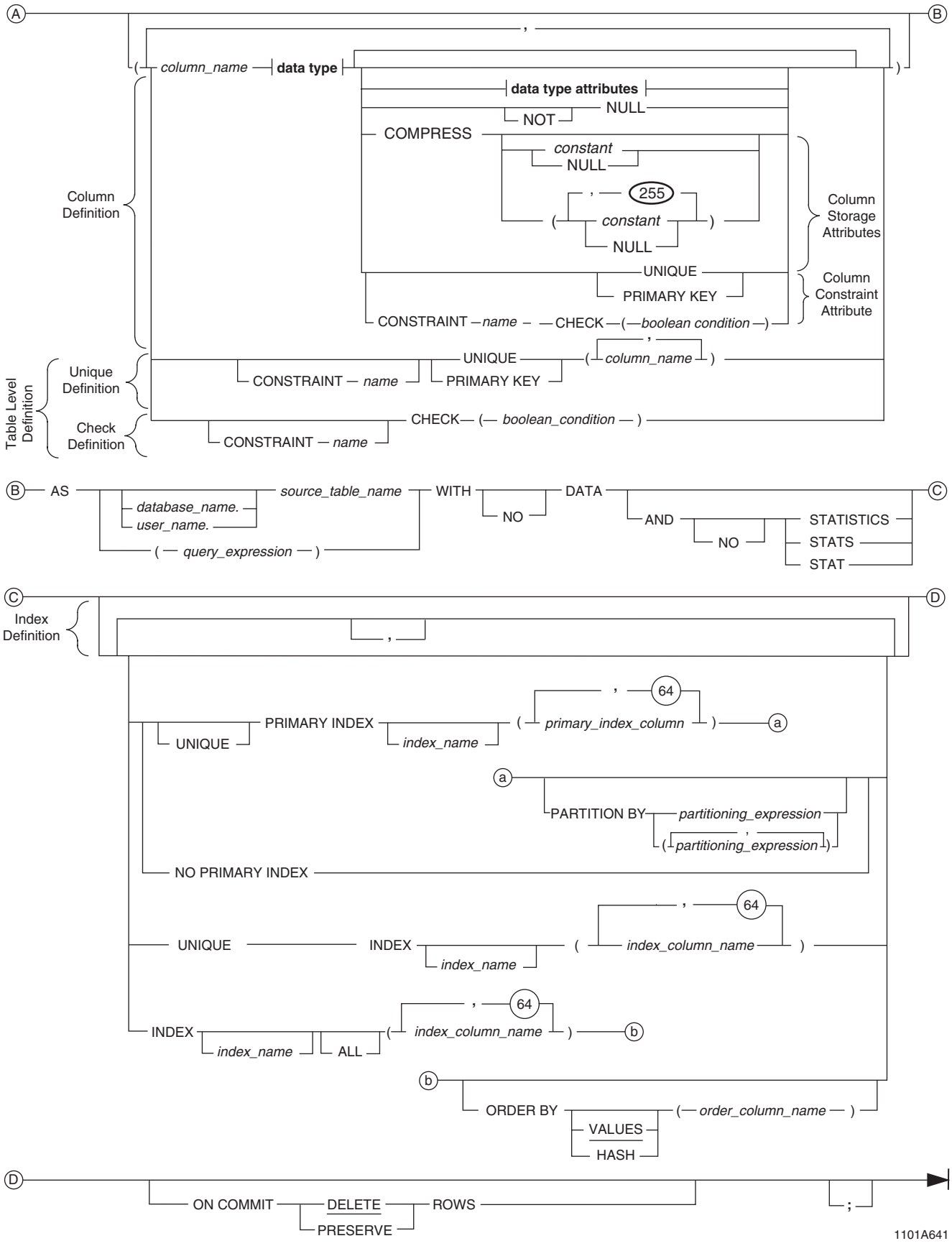



**Data Type Attributes**

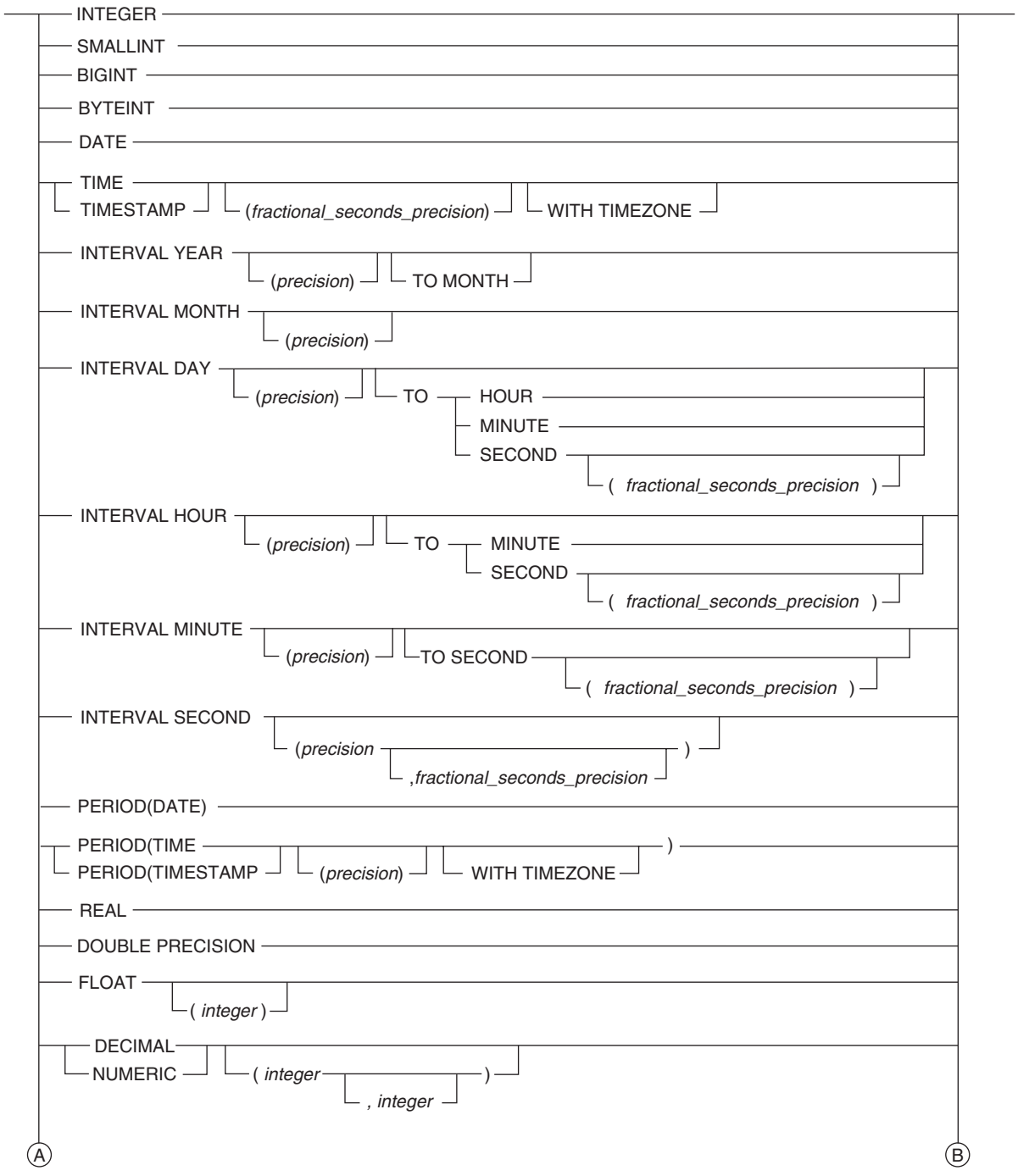


## Copy Table Syntax

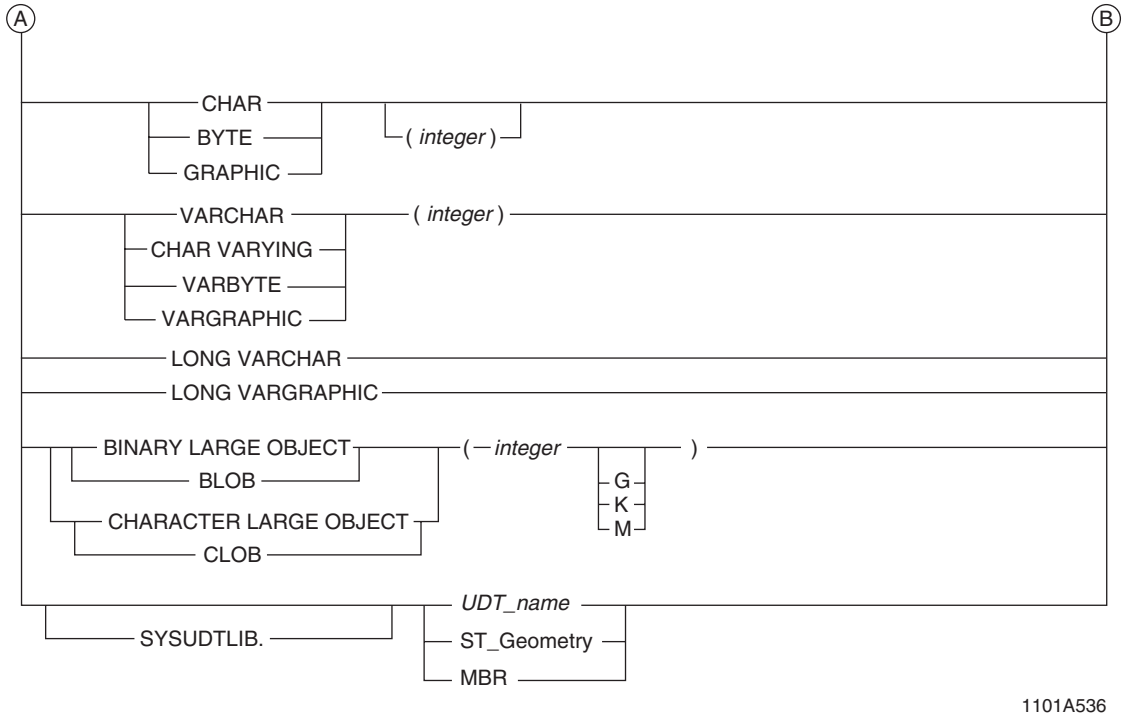



A
B
column_name
data type
data type attributes
NOT
NULL
COMPRESS
constant
NULL
255
Column Definition
Column Storage Attributes
UNIQUE
PRIMARY KEY
Column Constraint Attribute
CONSTRAINT name
CHECK
boolean condition
Table Level Definition
Unique Definition
Check Definition
CONSTRAINT name
PRIMARY KEY
CHECK
boolean_condition
B
AS
database_name.
user_name.
source_table_name
query_expression
WITH
NO
DATA
AND
STATISTICS
STATS
STAT
C
D
Index Definition
PRIMARY INDEX
index_name
primary_index_column
64
a
PARTITION BY
partitioning_expression
NO PRIMARY INDEX
INDEX
index_column_name
ALL
b
ORDER BY
order_column_name
VALUES
HASH
ON COMMIT
DELETE
PRESERVE
ROWS
;
1101A641

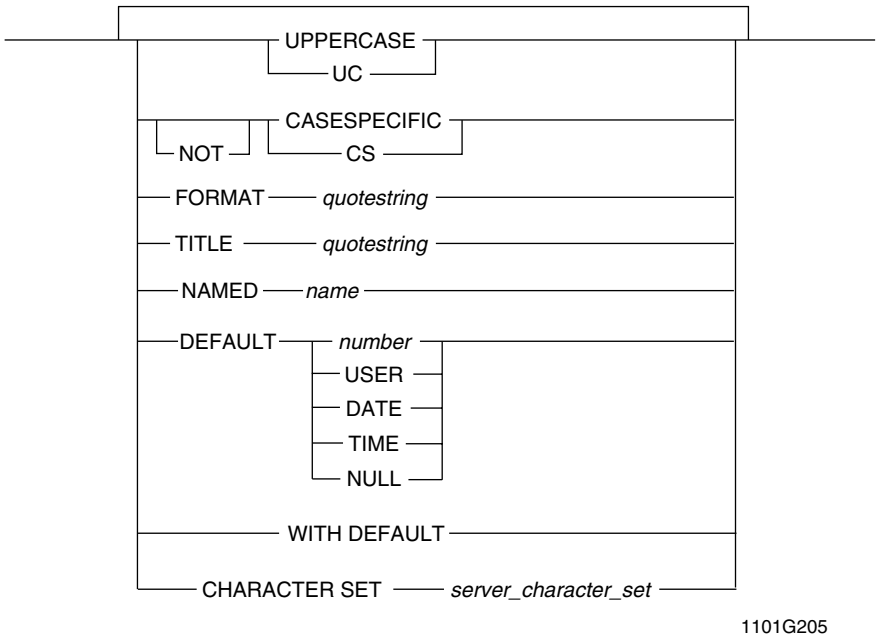
**Data Type**



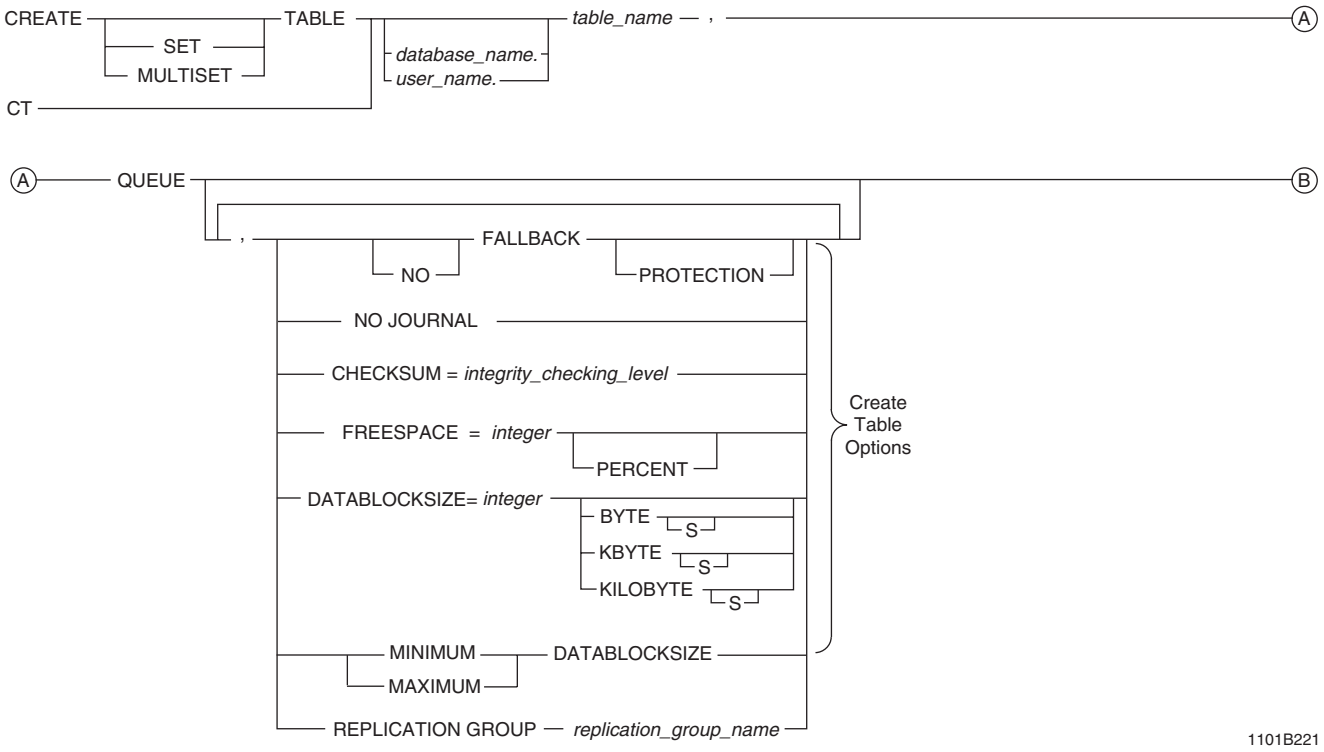

1101A535

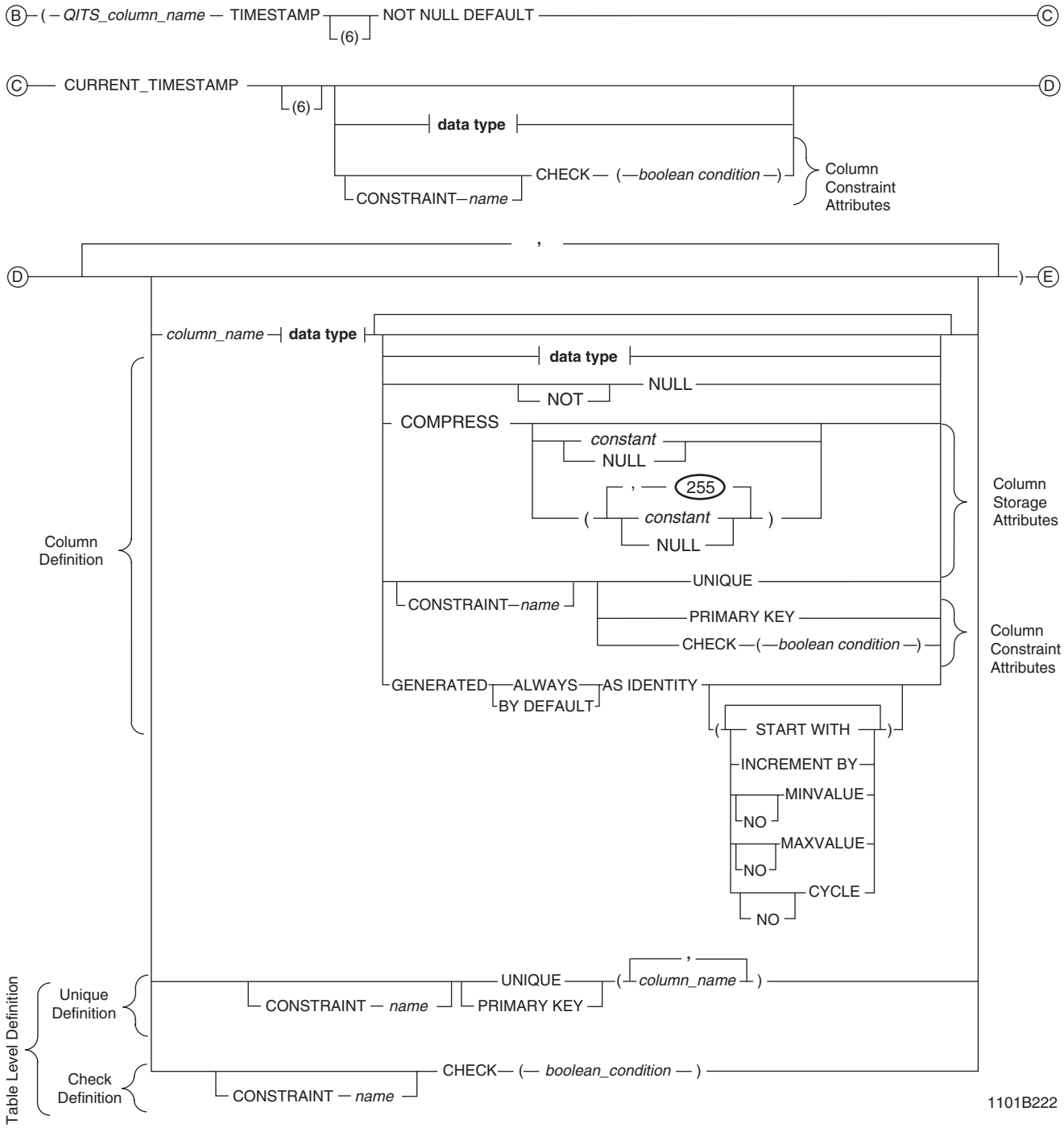


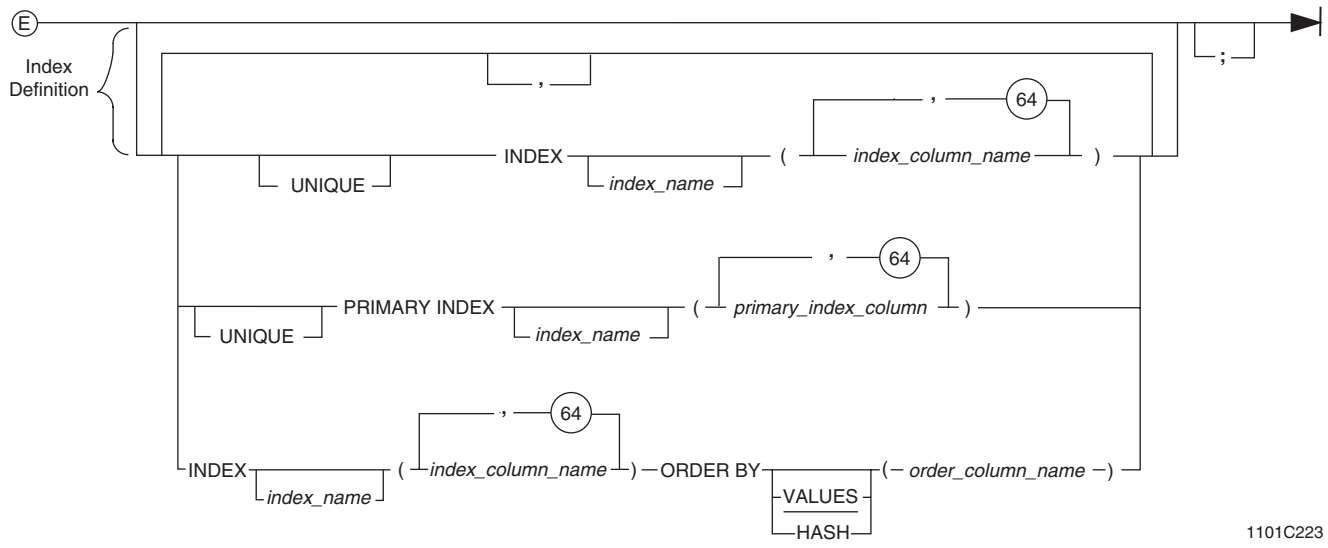

**Data Type Attributes**



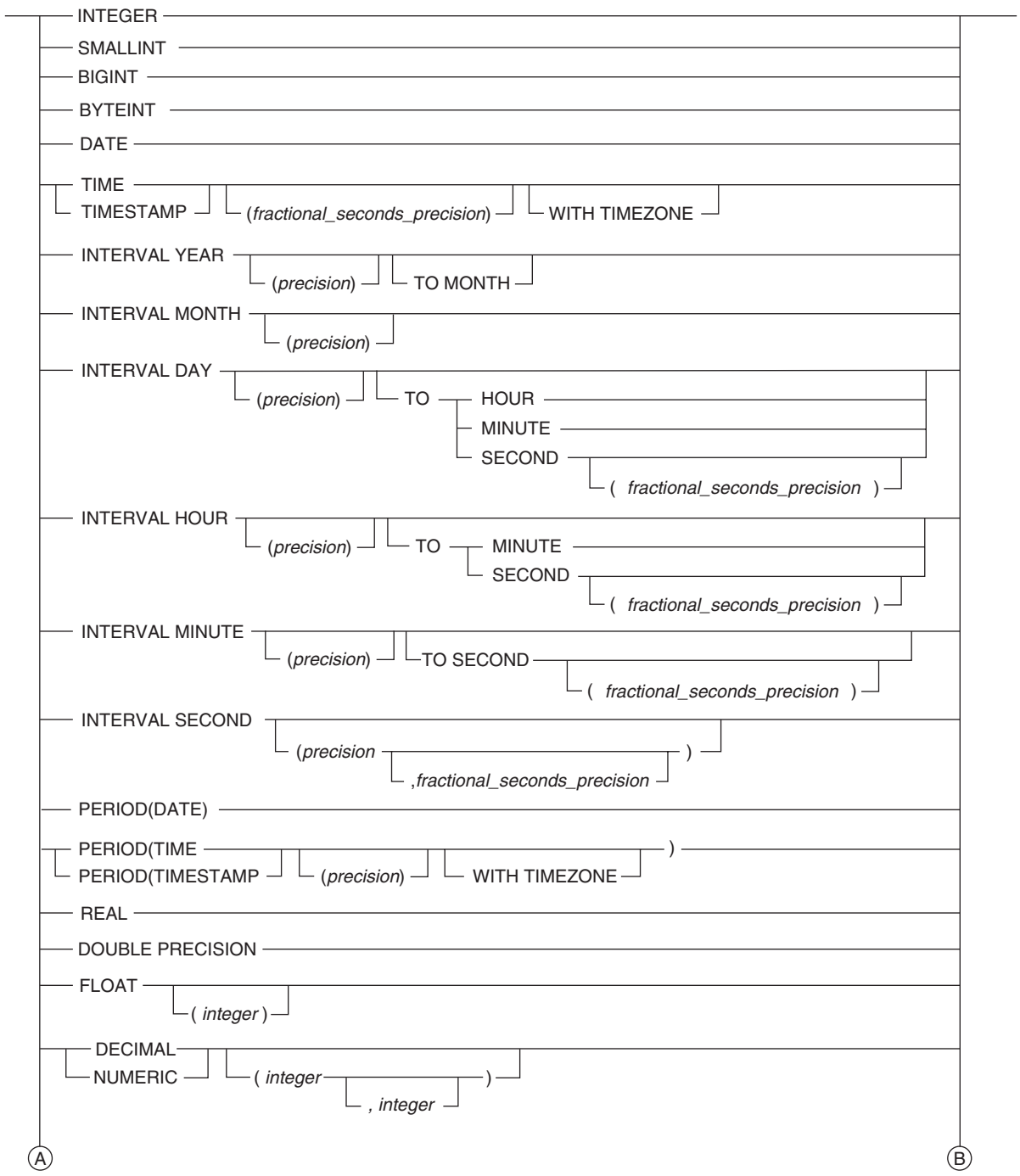
# CREATE TABLE (Queue Table Form)



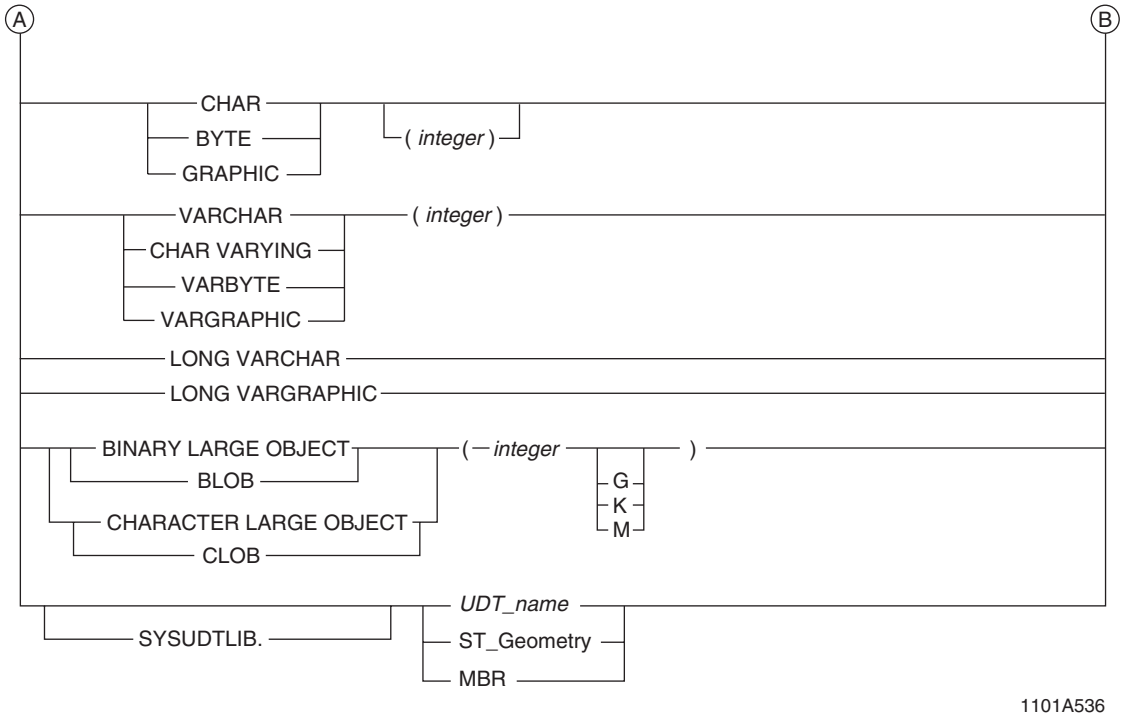

B
( – QITS_column_name — TIMESTAMP
(6)
NOT NULL DEFAULT
C
C
CURRENT_TIMESTAMP
(6)
D
data type
CHECK — (—boolean condition —)
CONSTRAINT—name
Column Constraint Attributes
,
D
)
E
column_name
data type
data type
NULL
NOT
COMPRESS
constant
NULL
,
255
(
constant
)
NULL
Column Storage Attributes
UNIQUE
CONSTRAINT—name
PRIMARY KEY
CHECK —(—boolean condition —)
Column Constraint Attributes
GENERATED
ALWAYS
BY DEFAULT
AS IDENTITY
(
START WITH
)
INCREMENT BY
MINVALUE
NO
MAXVALUE
NO
CYCLE
NO
Column Definition
,
UNIQUE
(
column_name
)
CONSTRAINT — name
PRIMARY KEY
Unique Definition
CHECK— (— boolean_condition — )
CONSTRAINT — name
Check Definition
Table Level Definition
1101B222


E
Index Definition
,
,
64
UNIQUE
INDEX
index_name
(
index_column_name
)
;
UNIQUE
PRIMARY INDEX
index_name
(
,
64
primary_index_column
)
INDEX
index_name
(
,
64
index_column_name
)
ORDER BY
VALUES
HASH
(
order_column_name
)
1101C223

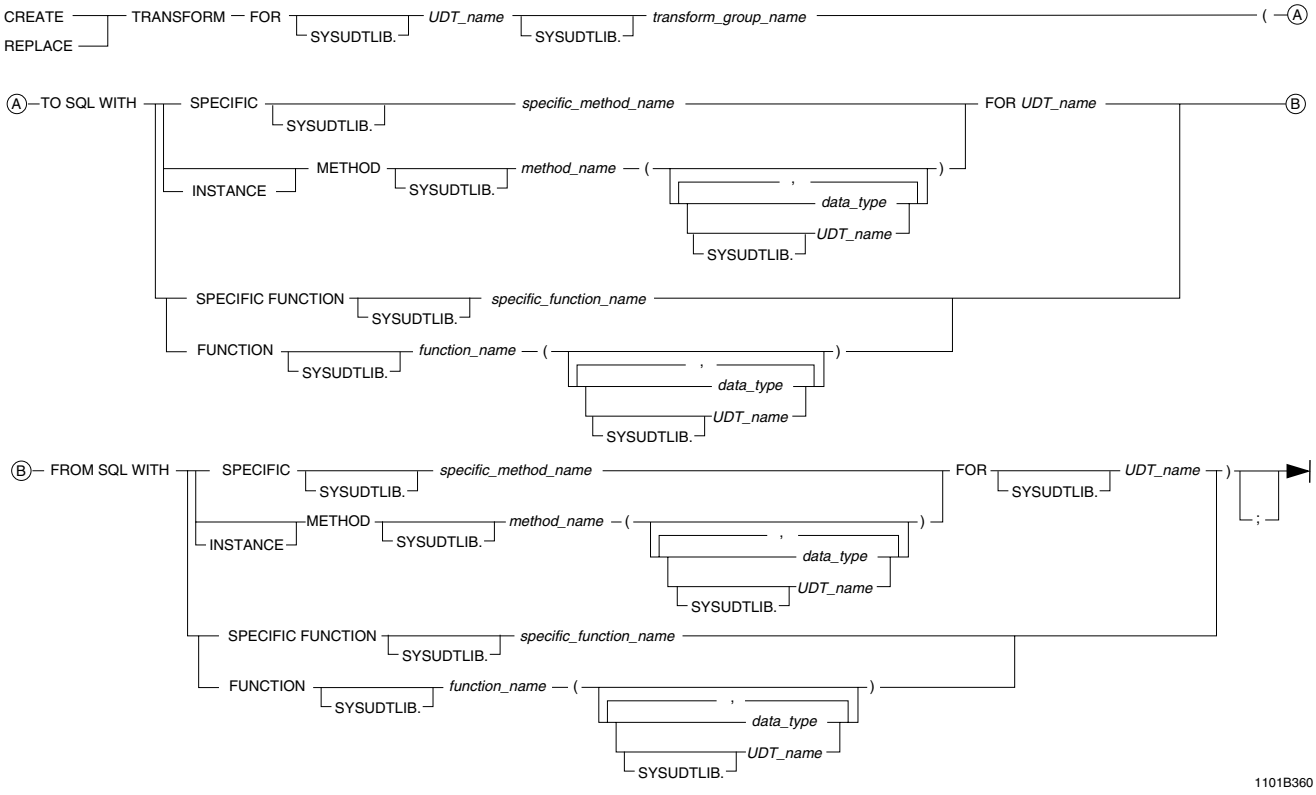
**Data Type**



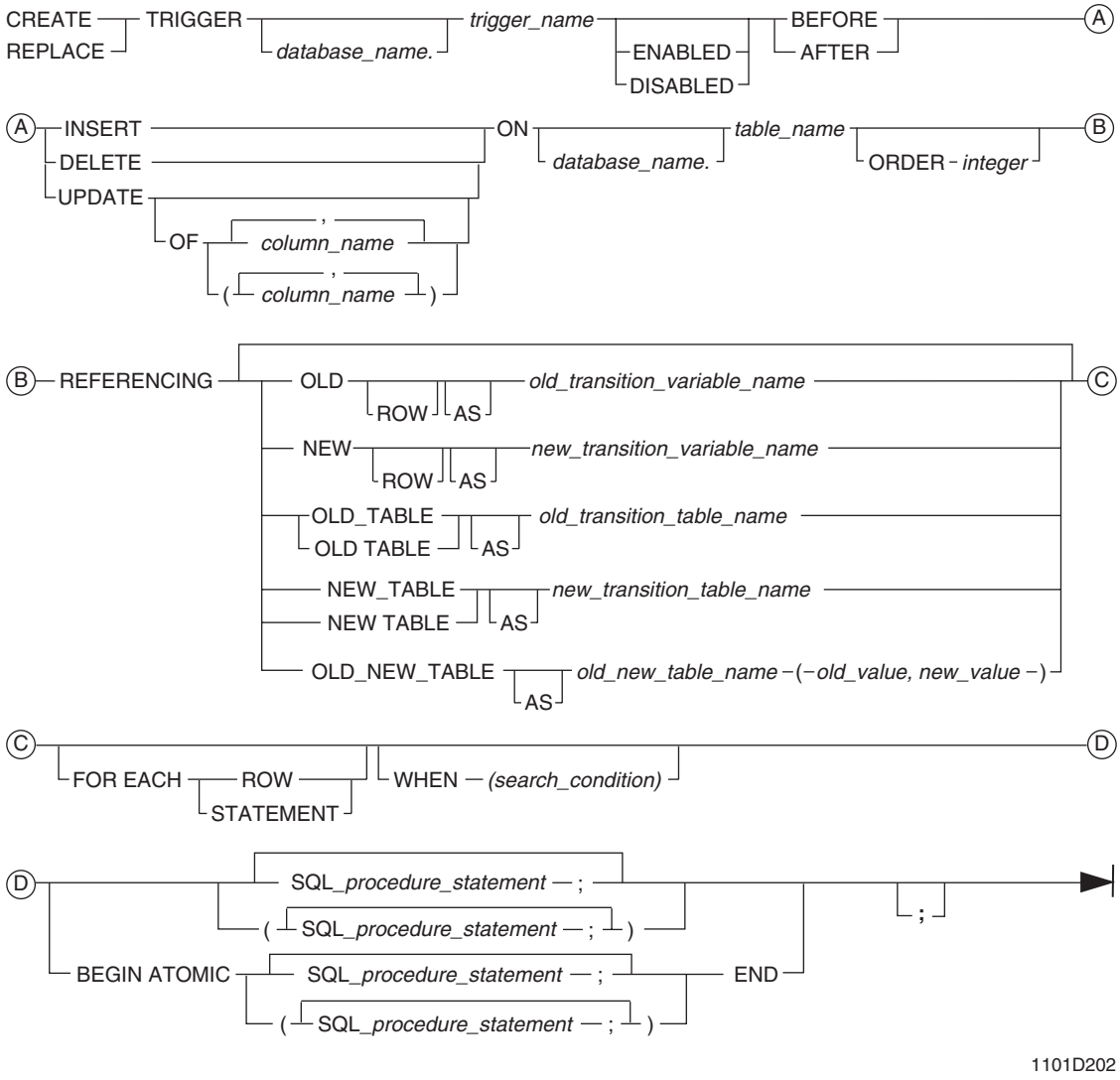

1101A535


A
B
CHAR
BYTE
GRAPHIC
( integer )
VARCHAR
CHAR VARYING
VARBYTE
VARGRAPHIC
( integer )
LONG VARCHAR
LONG VARGRAPHIC
BINARY LARGE OBJECT
BLOB
CHARACTER LARGE OBJECT
CLOB
( integer
G
K
M
)
SYSUDTLIB.
UDT_name
ST_Geometry
MBR
1101A536

# CREATE TRANSFORM/ REPLACE TRANSFORM

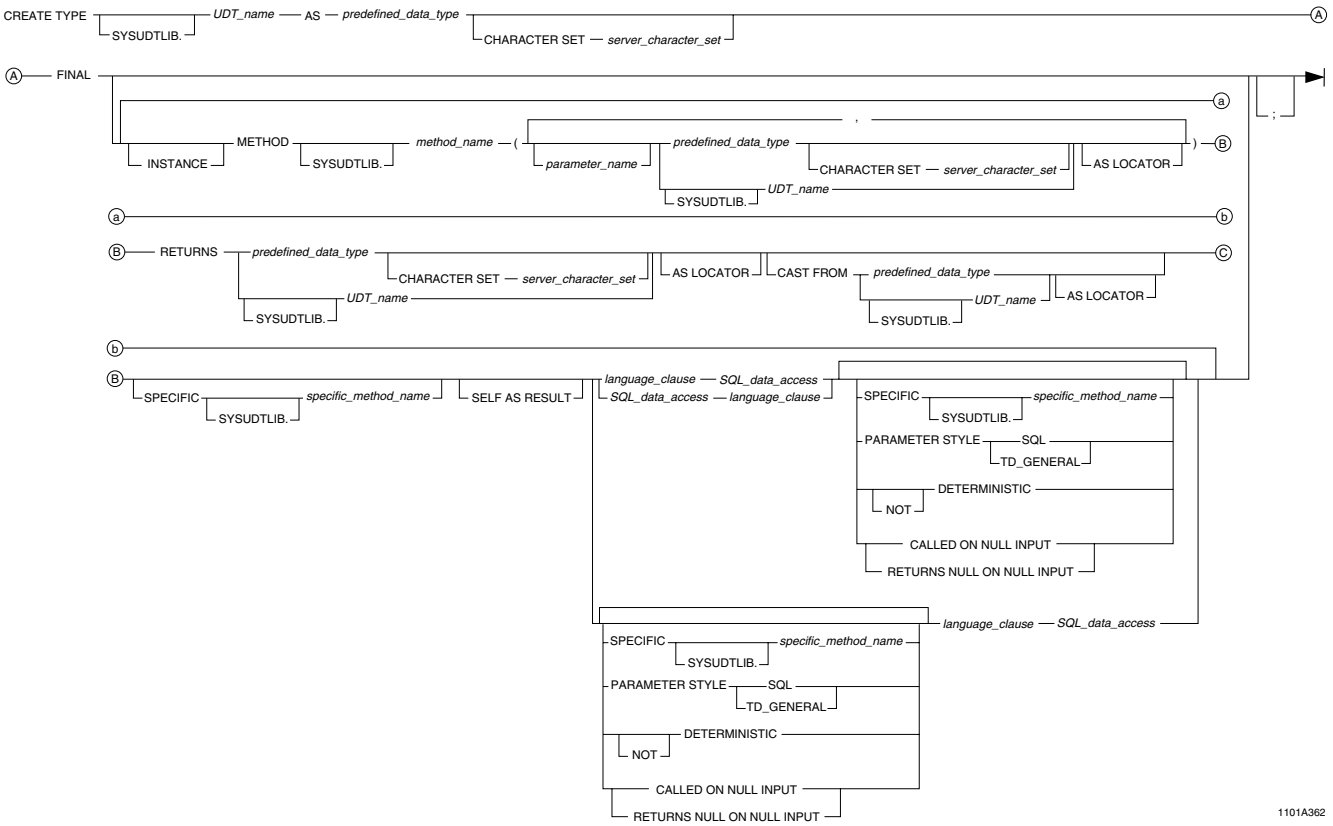


# CREATE TRIGGER/ REPLACE TRIGGER

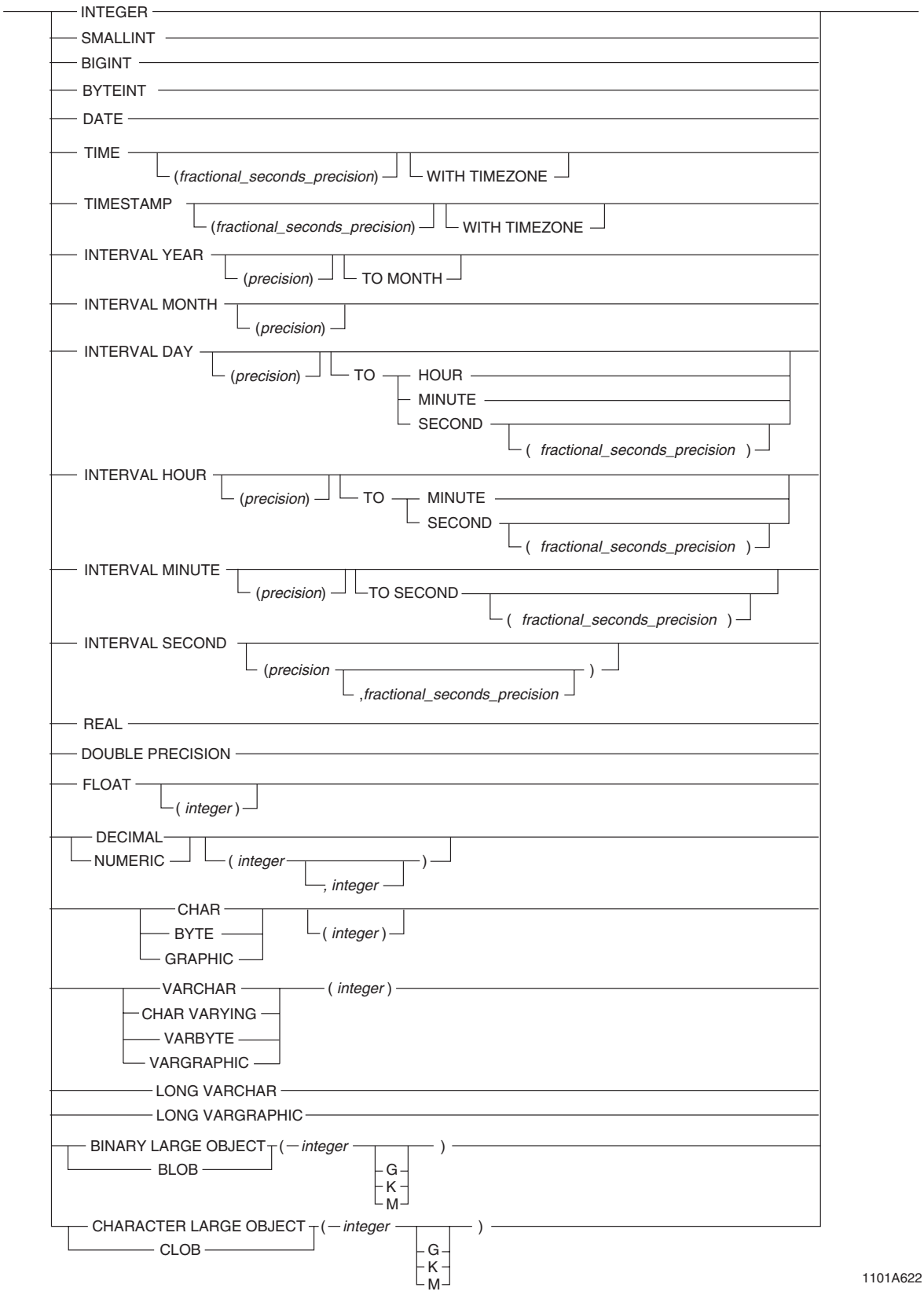



1101D202

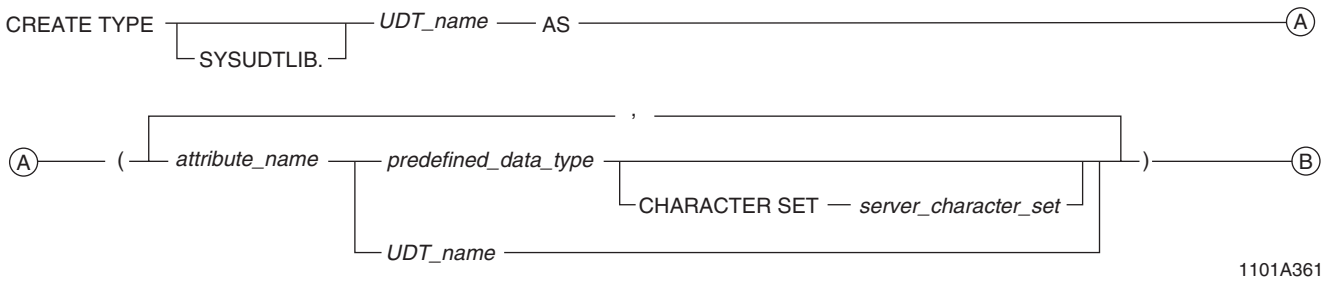
# CREATE TYPE (Distinct Form)

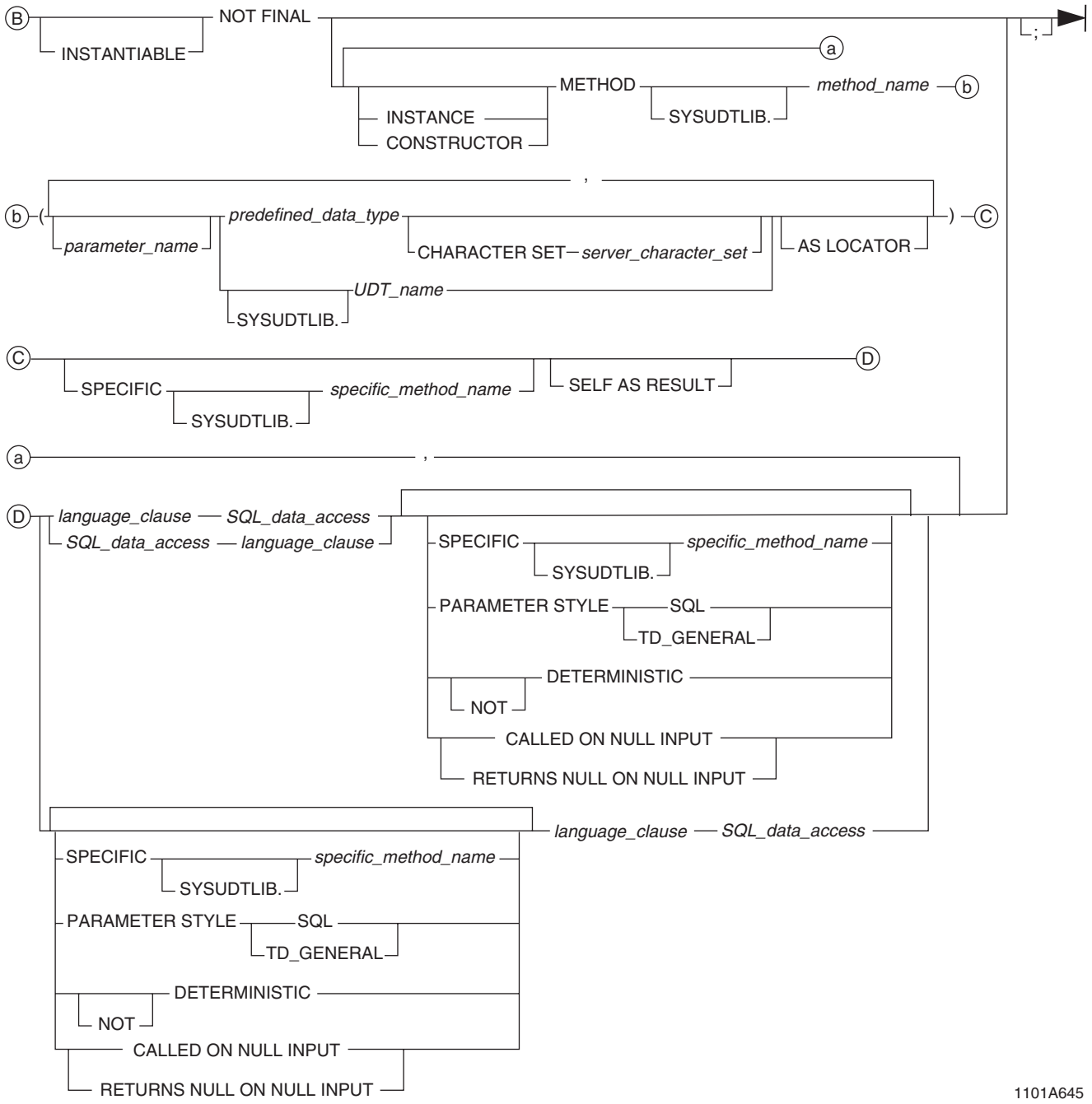


**Data Type Declaration**

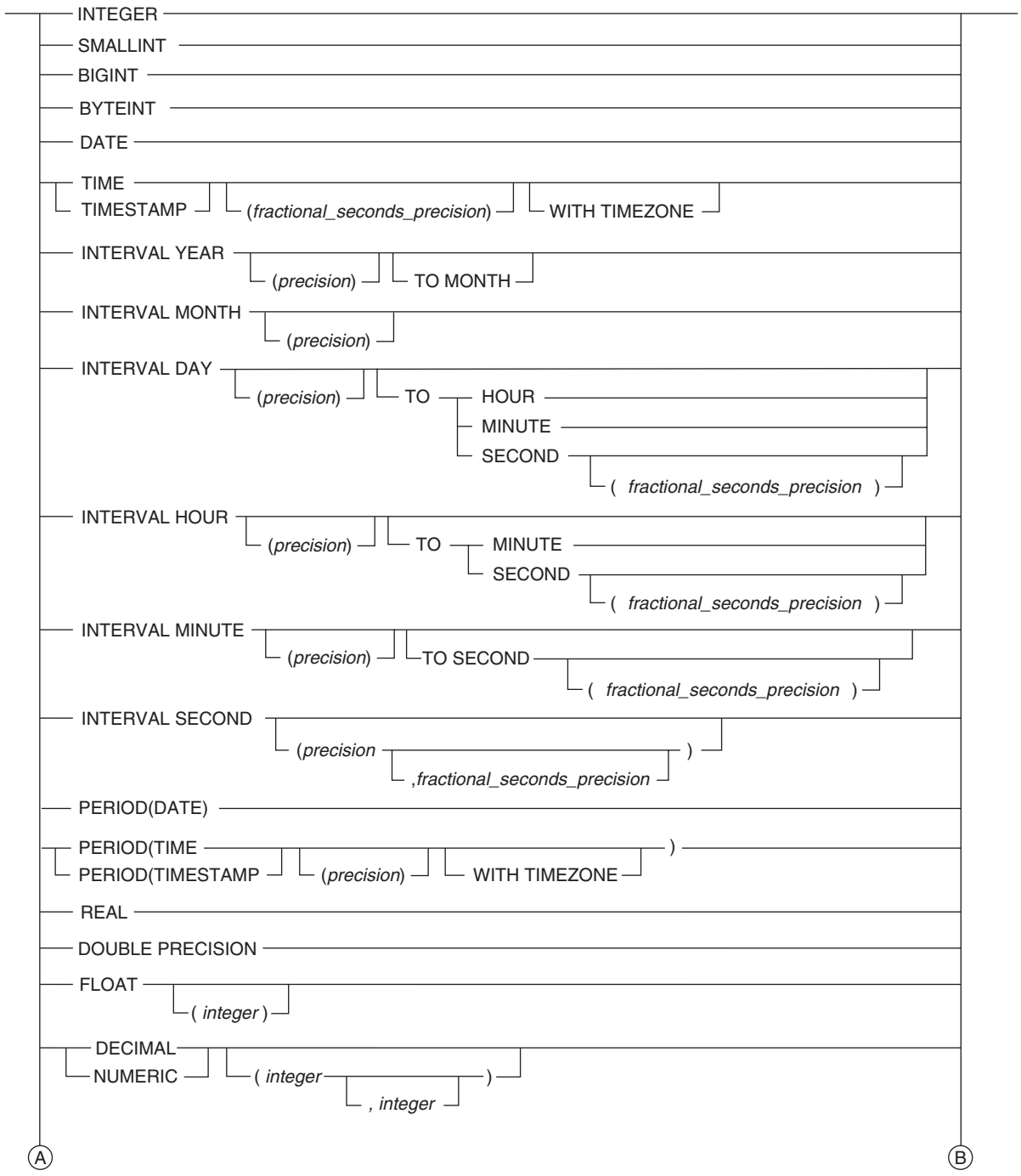


# CREATE TYPE (Structured Form)

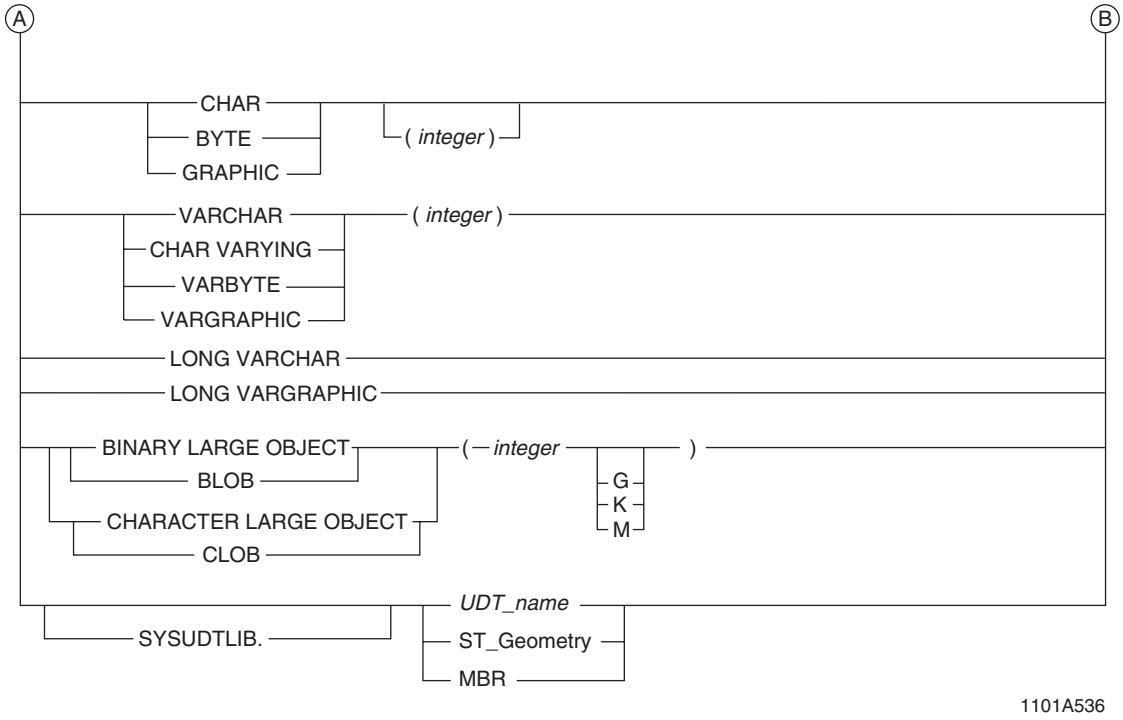



B
INSTANTIABLE
NOT FINAL
;
a
METHOD
method_name
b
INSTANCE
CONSTRUCTOR
SYSUDTLIB.
,
b
(
parameter_name
predefined_data_type
CHARACTER SET
server_character_set
AS LOCATOR
)
C
UDT_name
SYSUDTLIB.
C
SPECIFIC
specific_method_name
SYSUDTLIB.
SELF AS RESULT
D
a
,
D
language_clause
SQL_data_access
SQL_data_access
language_clause
SPECIFIC
specific_method_name
SYSUDTLIB.
PARAMETER STYLE
SQL
TD_GENERAL
DETERMINISTIC
NOT
CALLED ON NULL INPUT
RETURNS NULL ON NULL INPUT
language_clause
SQL_data_access
SPECIFIC
specific_method_name
SYSUDTLIB.
PARAMETER STYLE
SQL
TD_GENERAL
DETERMINISTIC
NOT
CALLED ON NULL INPUT
RETURNS NULL ON NULL INPUT
1101A645

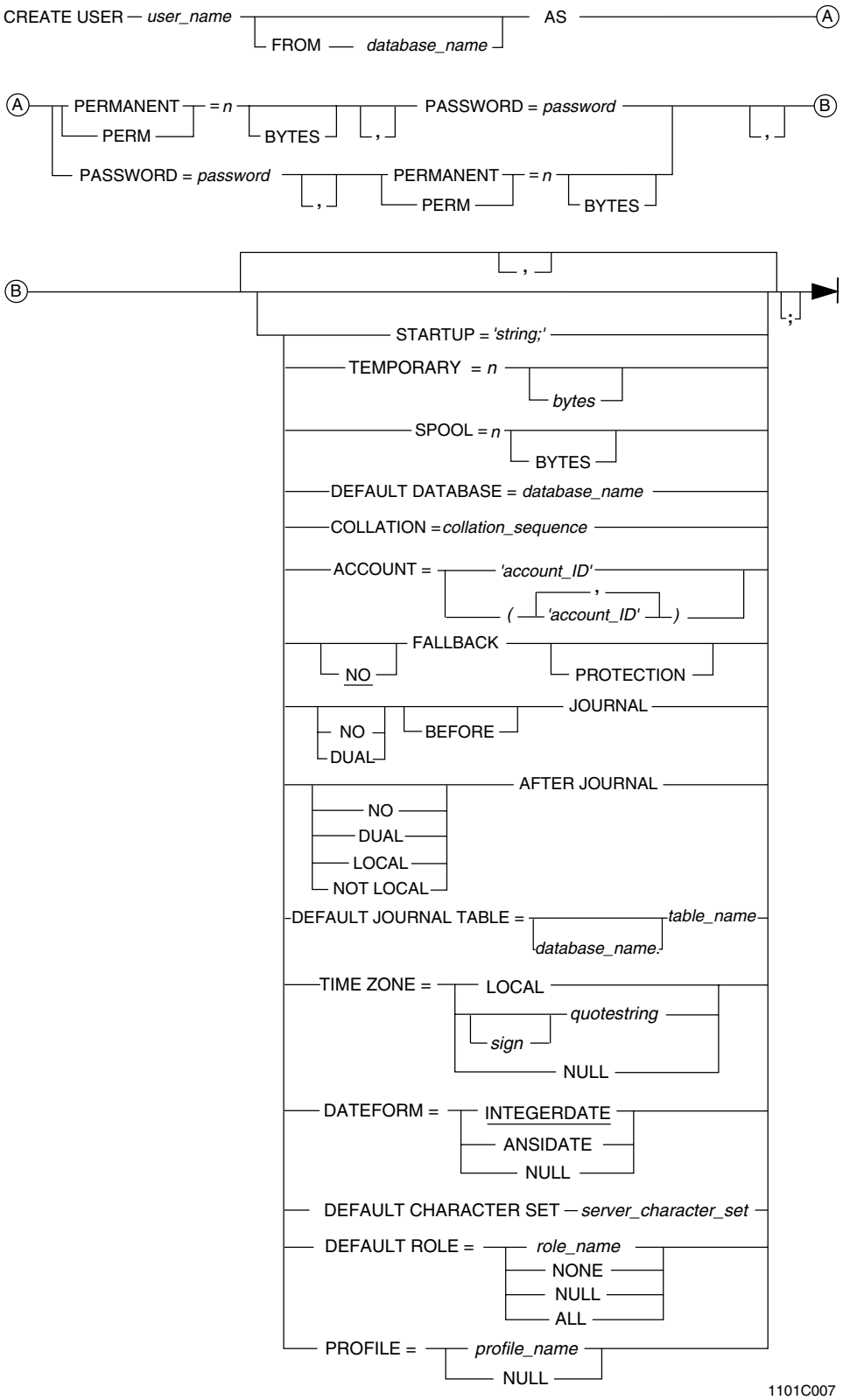
**Data Type**



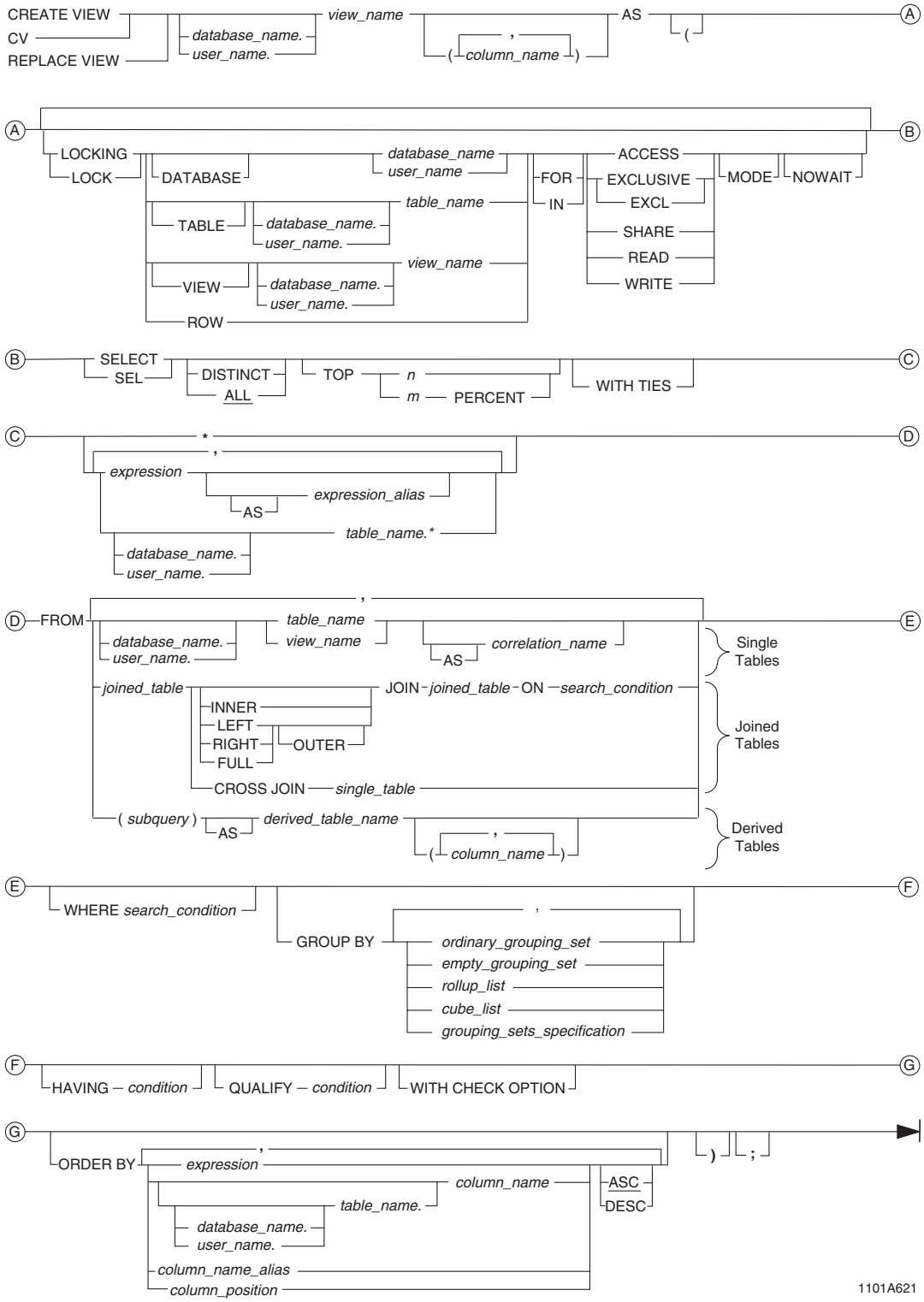

1101A535


A
B
CHAR
BYTE
GRAPHIC
( integer )
VARCHAR
CHAR VARYING
VARBYTE
VARGRAPHIC
( integer )
LONG VARCHAR
LONG VARGRAPHIC
BINARY LARGE OBJECT
BLOB
CHARACTER LARGE OBJECT
CLOB
( integer
G
K
M
)
SYSUDTLIB.
UDT_name
ST_Geometry
MBR
1101A536

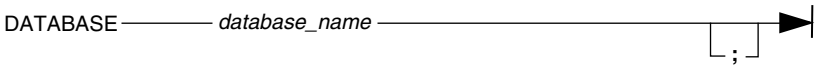
# CREATE USER


CREATE USER
user_name
FROM
database_name
AS
A
PERMANENT
PERM
= n
BYTES
,
PASSWORD = password
B
PASSWORD = password
,
PERMANENT
PERM
= n
BYTES
;
STARTUP = 'string;'
TEMPORARY = n
bytes
SPOOL = n
BYTES
DEFAULT DATABASE = database_name
COLLATION = collation_sequence
ACCOUNT =
'account_ID'
( 'account_ID' )
FALLBACK
NO
PROTECTION
JOURNAL
NO
DUAL
BEFORE
AFTER JOURNAL
NO
DUAL
LOCAL
NOT LOCAL
DEFAULT JOURNAL TABLE =
table_name
database_name.
TIME ZONE =
LOCAL
quotestring
sign
NULL
DATEFORM =
INTEGERDATE
ANSIDATE
NULL
DEFAULT CHARACTER SET
server_character_set
DEFAULT ROLE =
role_name
NONE
NULL
ALL
PROFILE =
profile_name
NULL
1101C007

# CREATE VIEW/ REPLACE VIEW

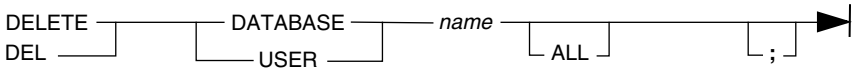


# DATABASE



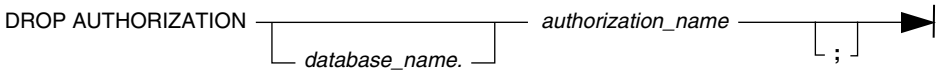

FF07A016

# DELETE DATABASE
# DELETE USER



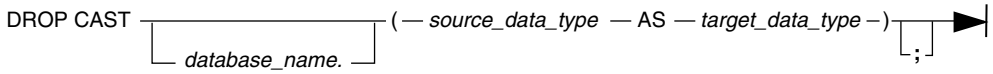

FF07D026

# DROP AUTHORIZATION



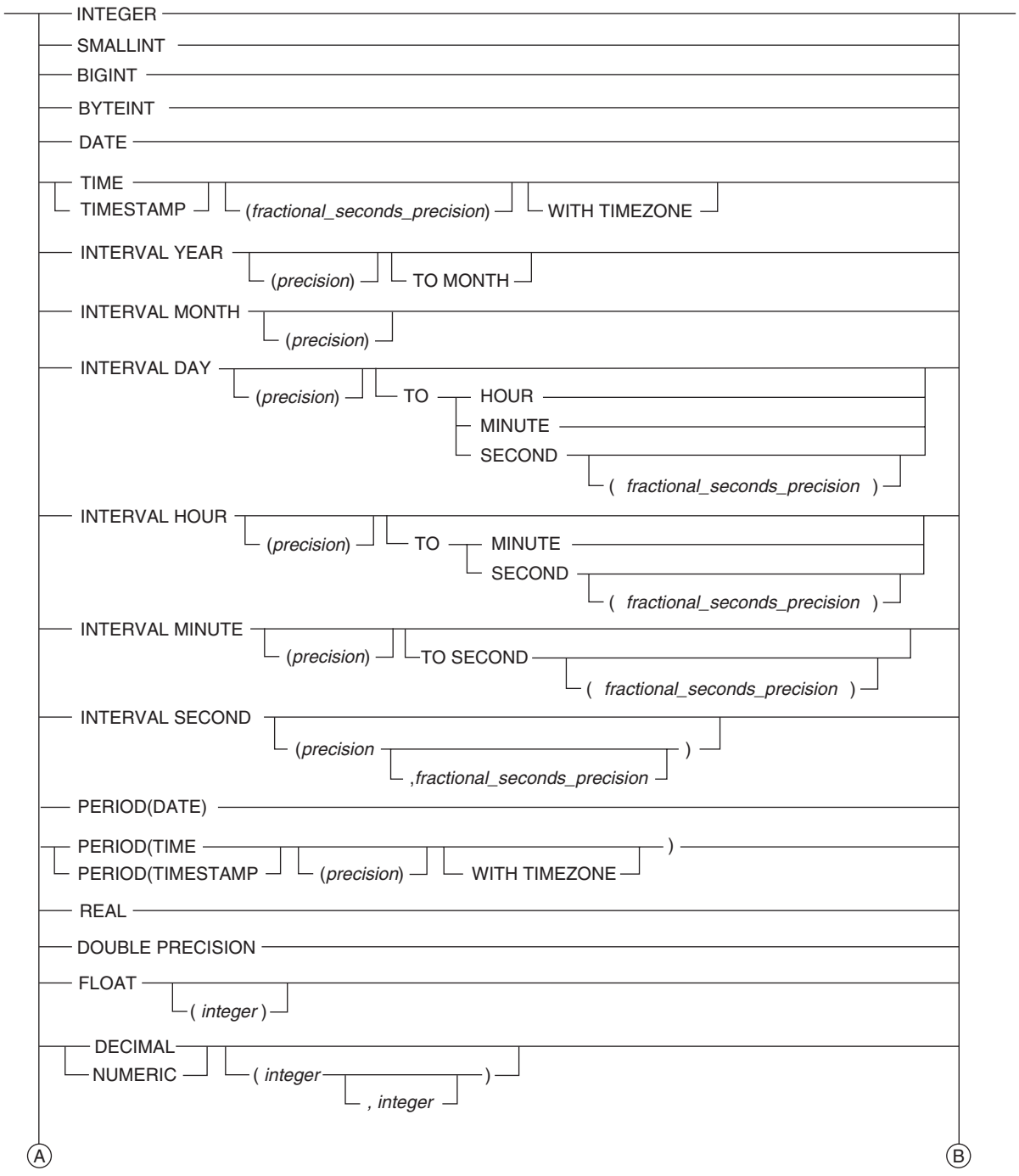

1101A324

# DROP CAST

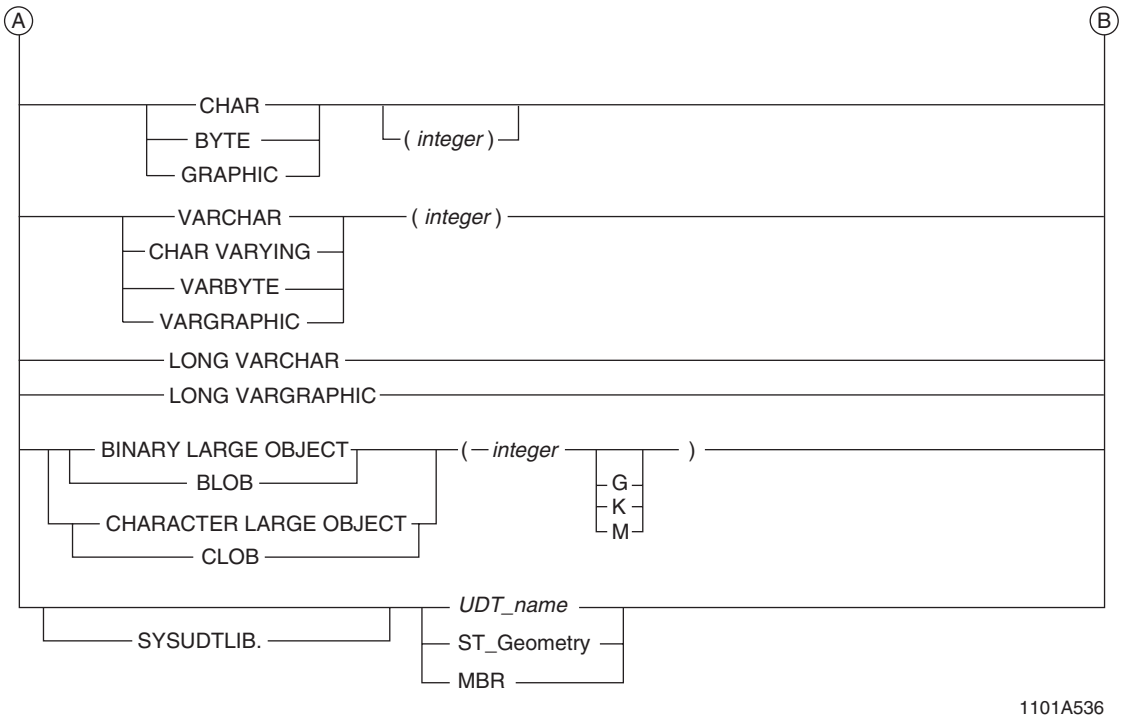



1101A325

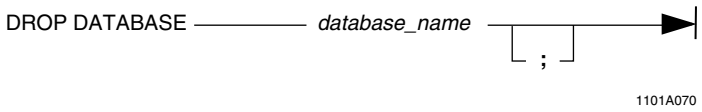
**Data Type**



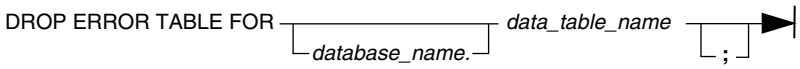

1101A535

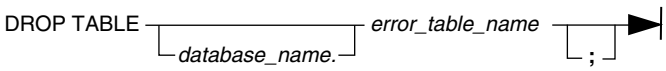



# DROP DATABASE

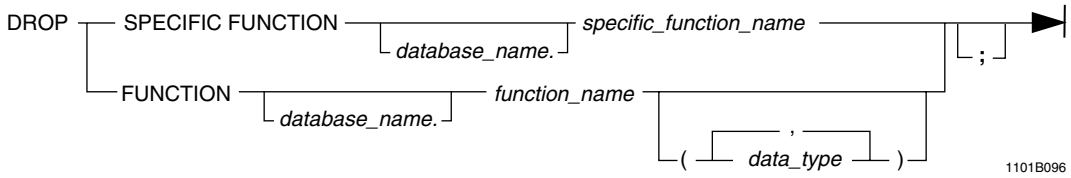



# DROP ERROR TABLE

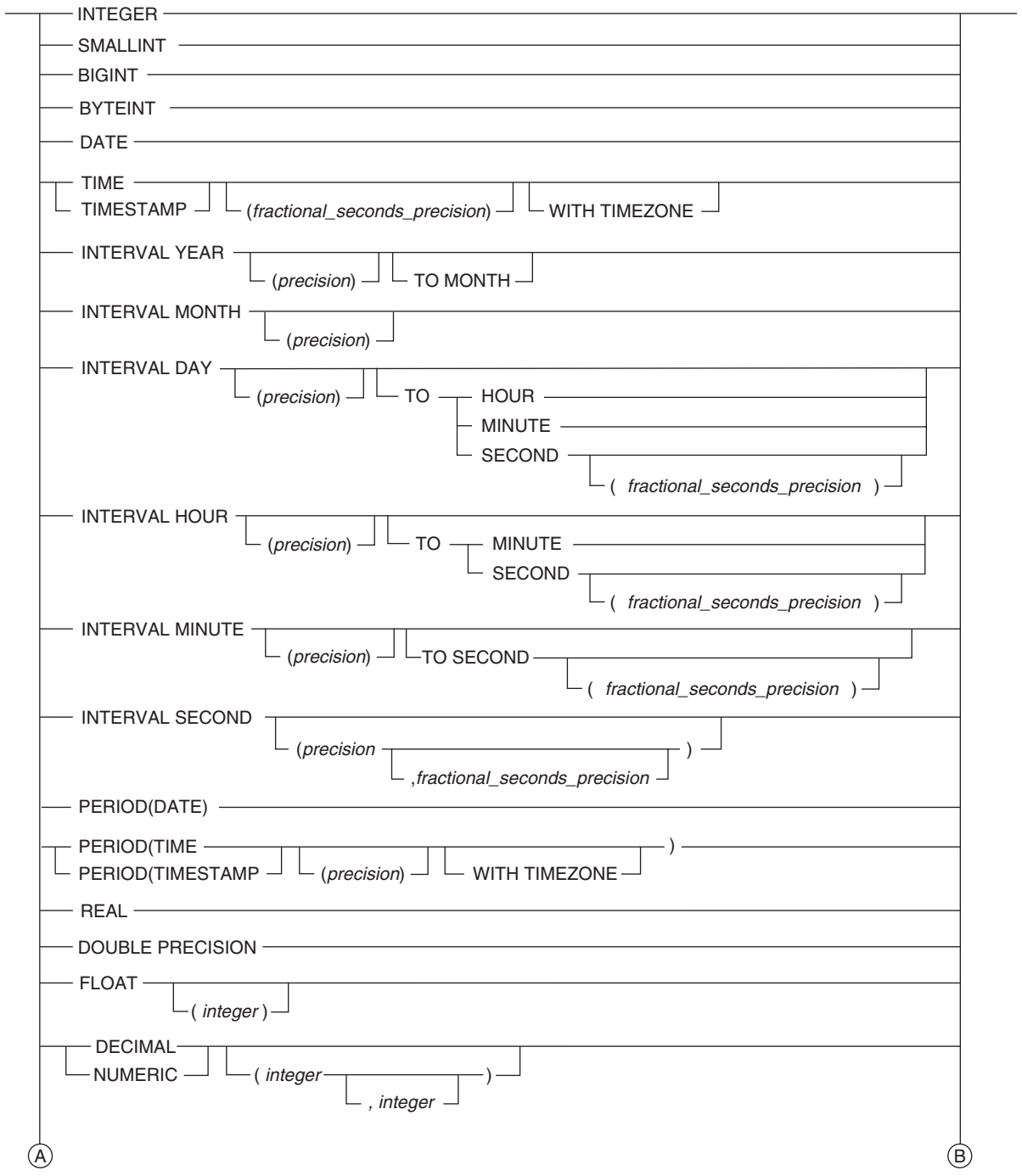


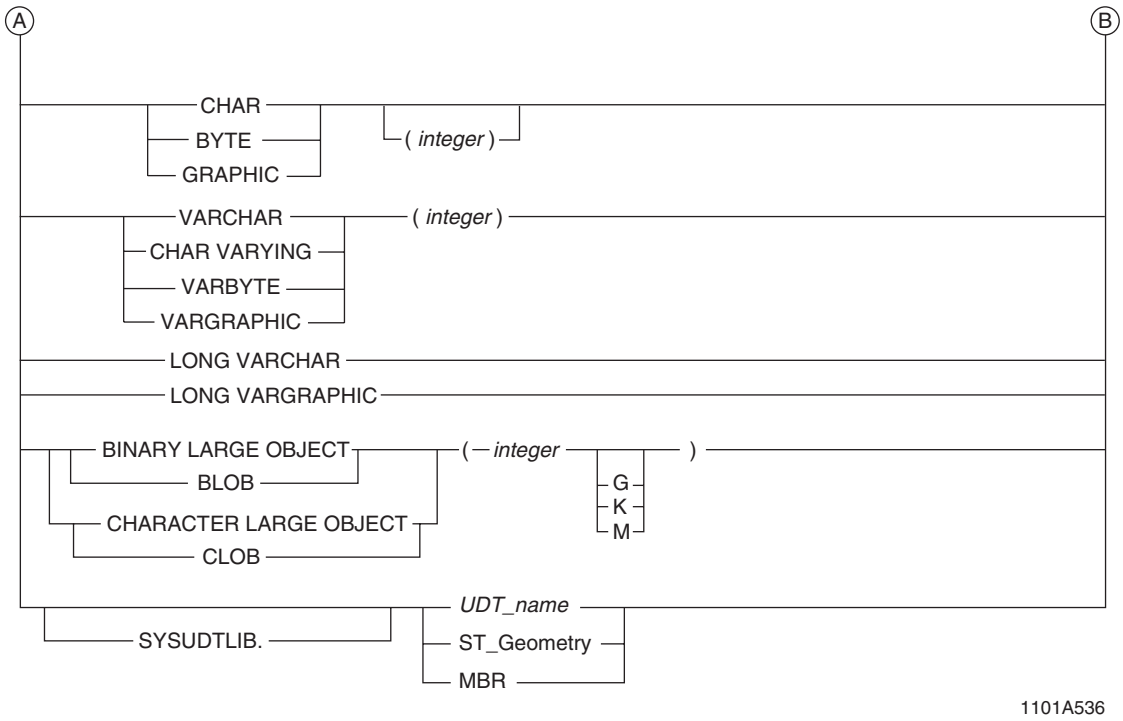
# DROP FUNCTION



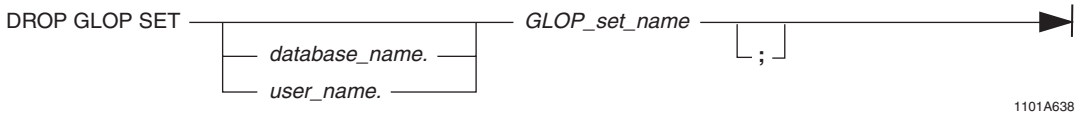
**Data Type**



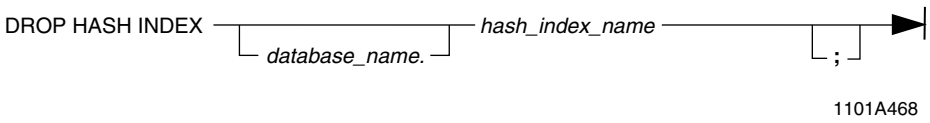

1101A535




# DROP GLOP SET

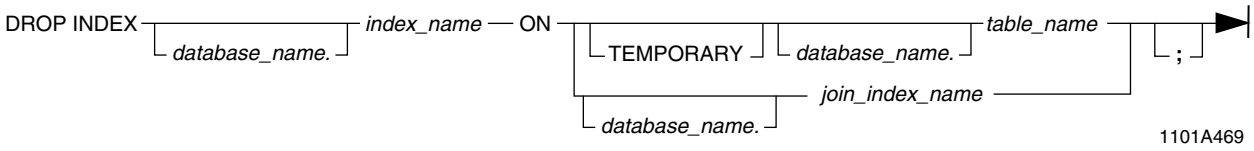



# DROP HASH INDEX

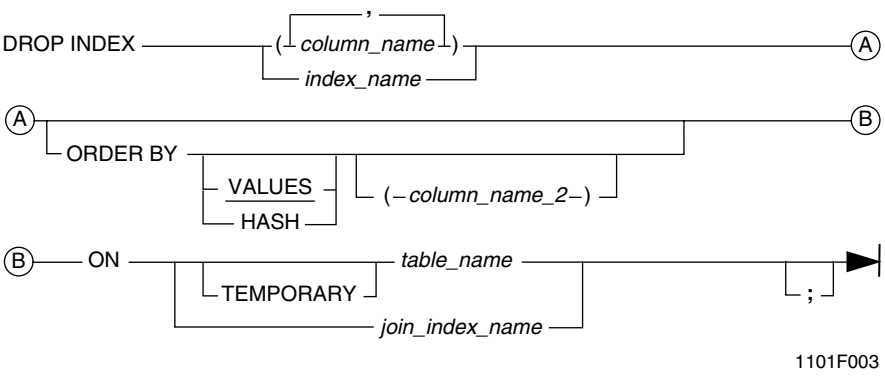


# DROP INDEX

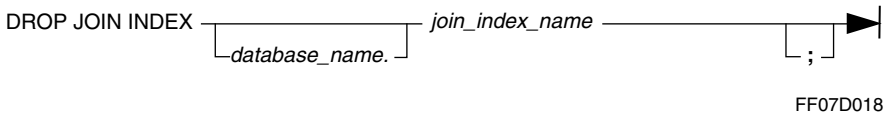
## DROP *index_name* Syntax




## DROP *index_definition* Syntax

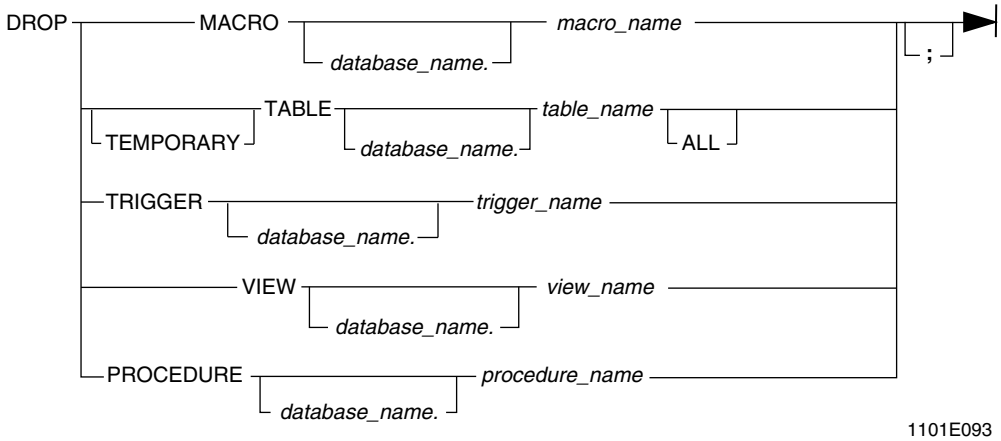



# DROP JOIN INDEX

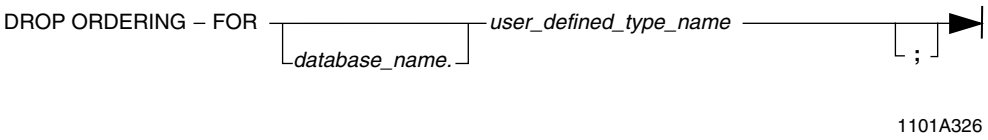



# DROP MACRO/ DROP PROCEDURE/ DROP TABLE/

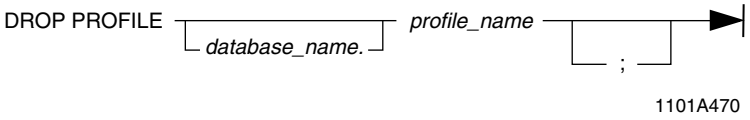
# DROP TRIGGER/ DROP VIEW



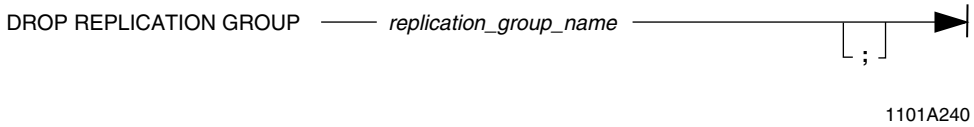

# DROP ORDERING



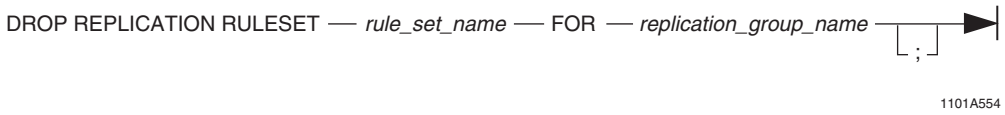

# DROP PROFILE



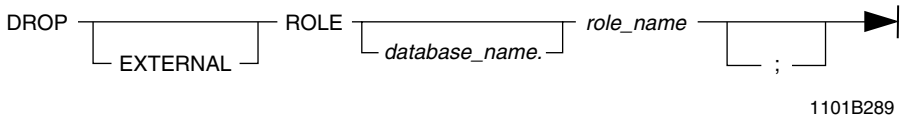
# DROP REPLICATION GROUP



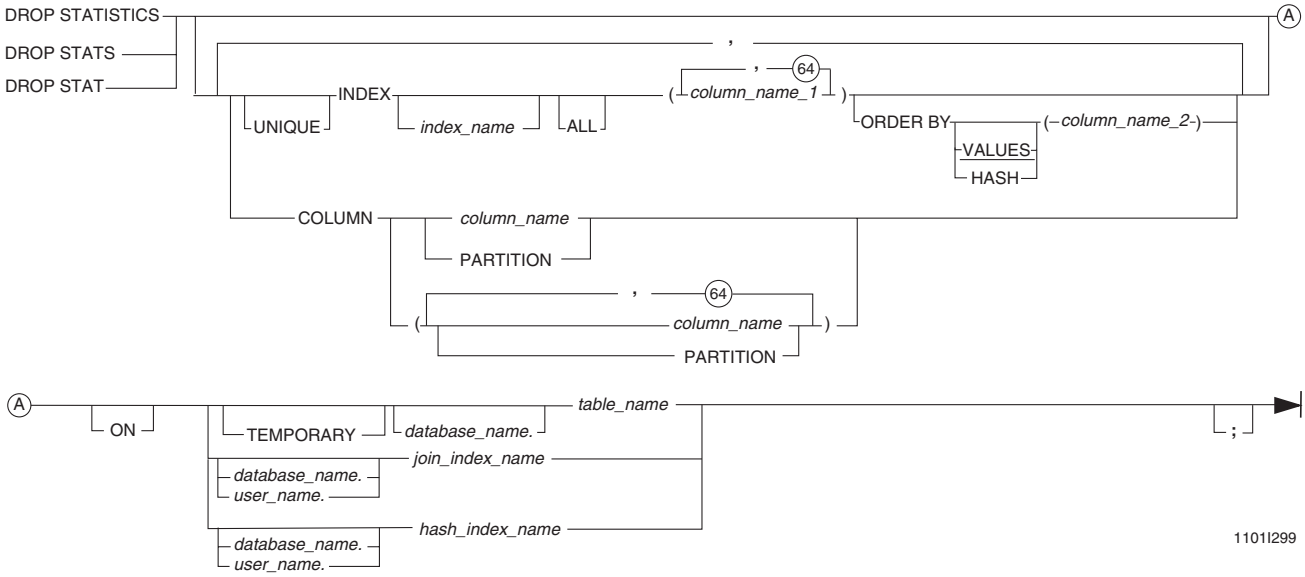

# DROP REPLICATION RULESET



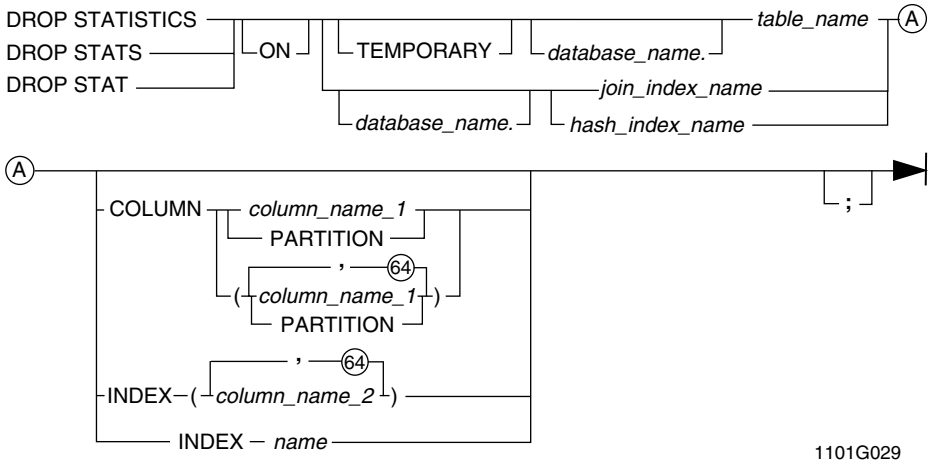

# DROP ROLE



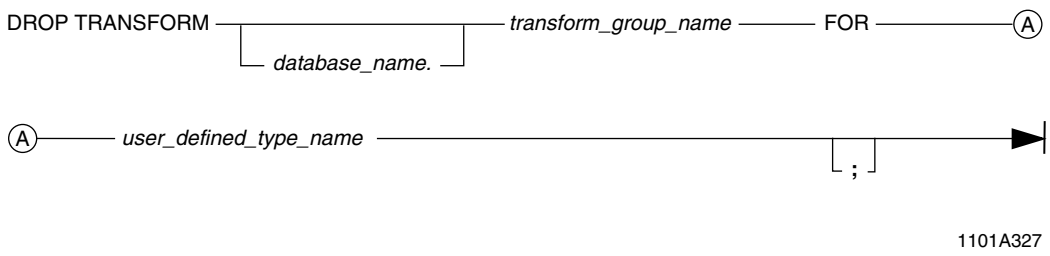
# DROP STATISTICS (Optimizer Form)



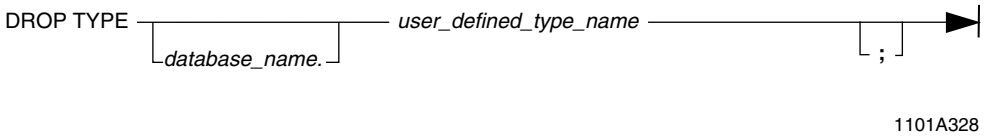

## Syntax (Alternate)



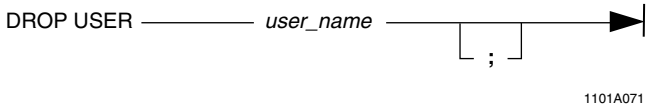
# DROP TRANSFORM



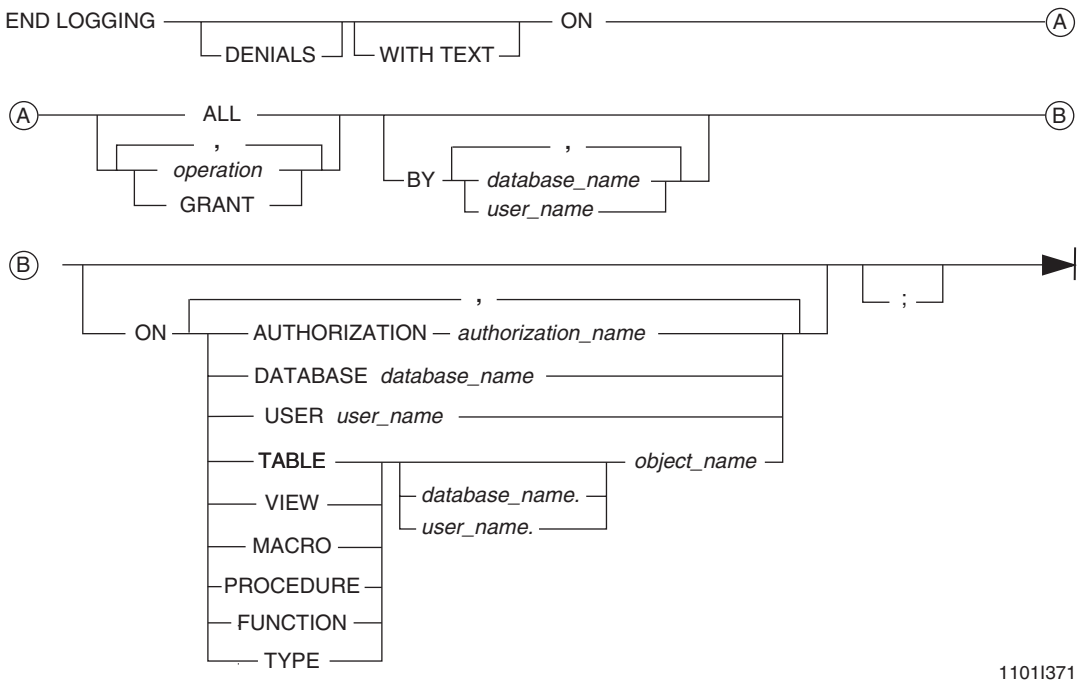

# DROP TYPE



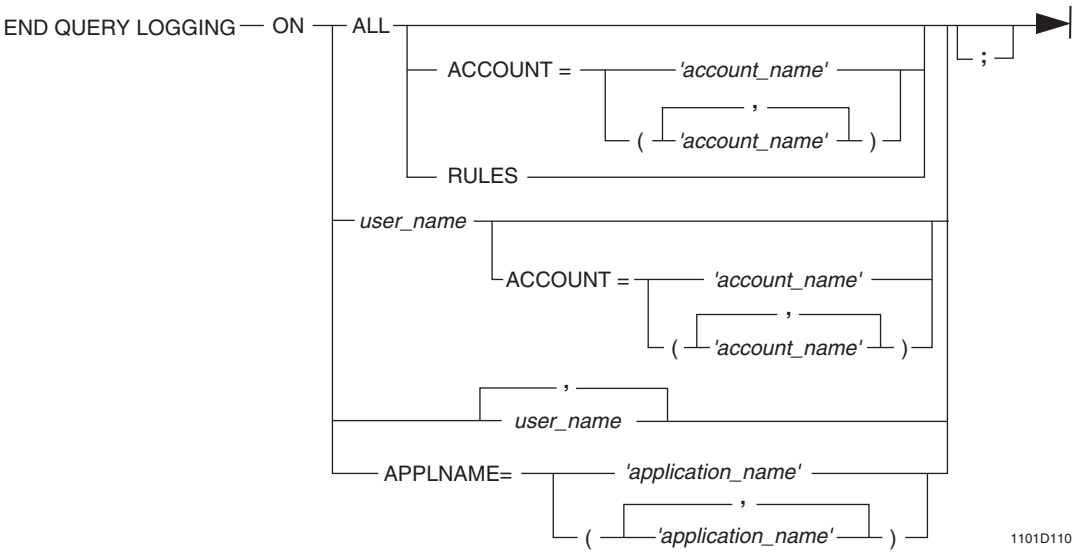

# DROP USER



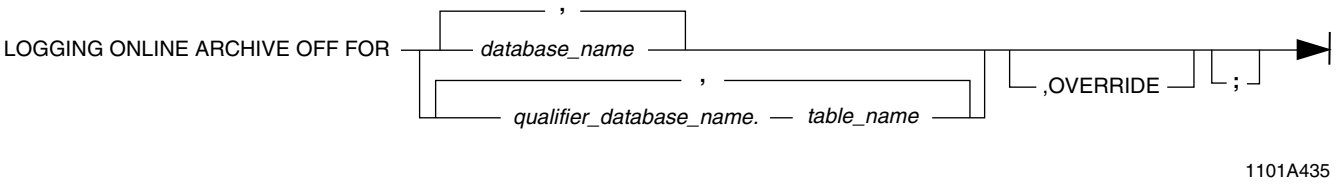
# END LOGGING


END LOGGING
DENIALS
WITH TEXT
ON
A
ALL
,
operation
GRANT
BY
,
database_name
user_name
B
;
ON
,
AUTHORIZATION
authorization_name
DATABASE
database_name
USER
user_name
TABLE
VIEW
MACRO
PROCEDURE
FUNCTION
TYPE
database_name.
user_name.
object_name
1101I371

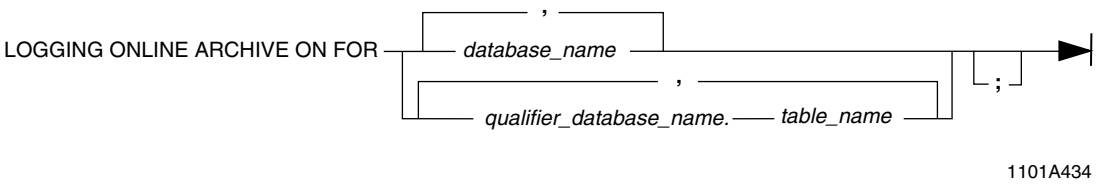
# END QUERY LOGGING



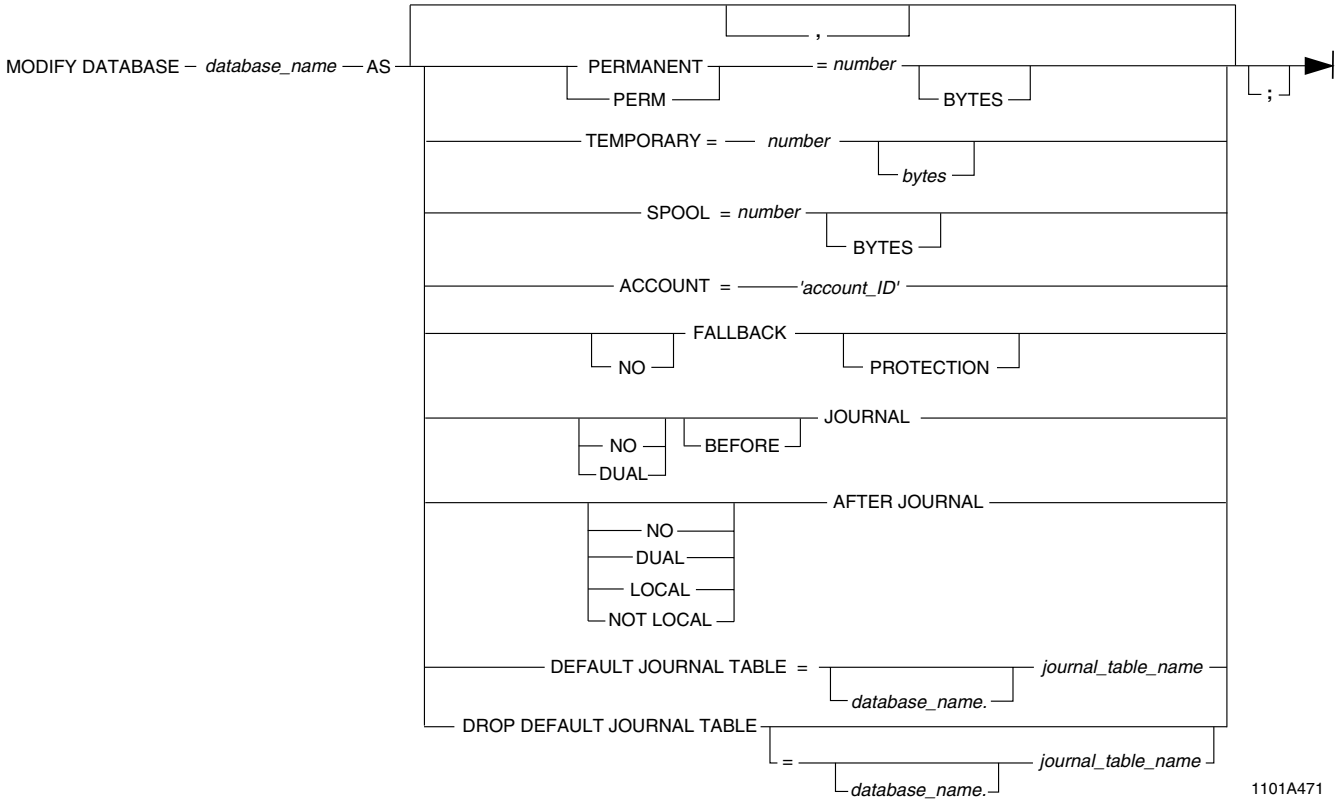

# LOGGING ONLINE ARCHIVE OFF



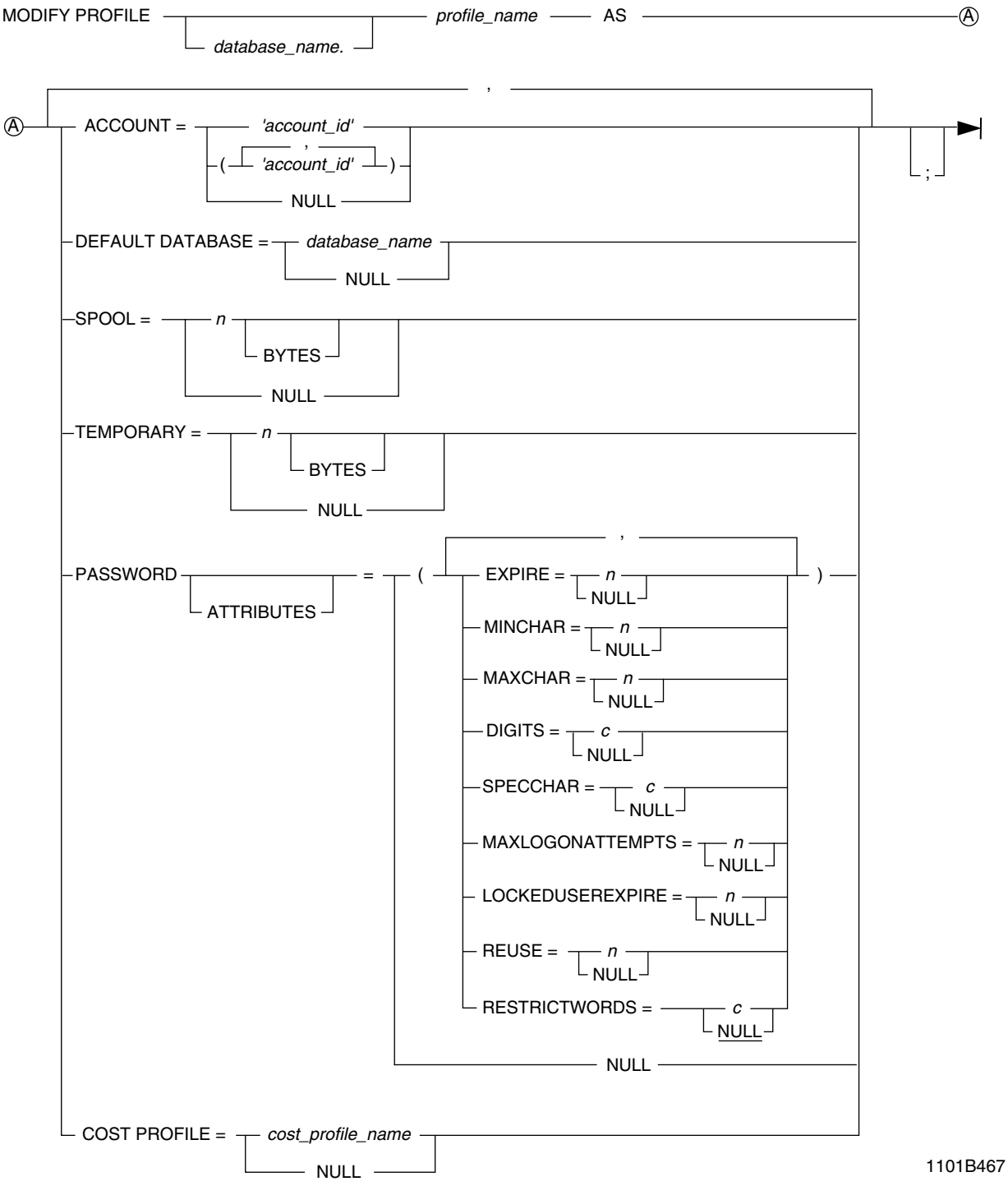

# LOGGING ONLINE ARCHIVE ON



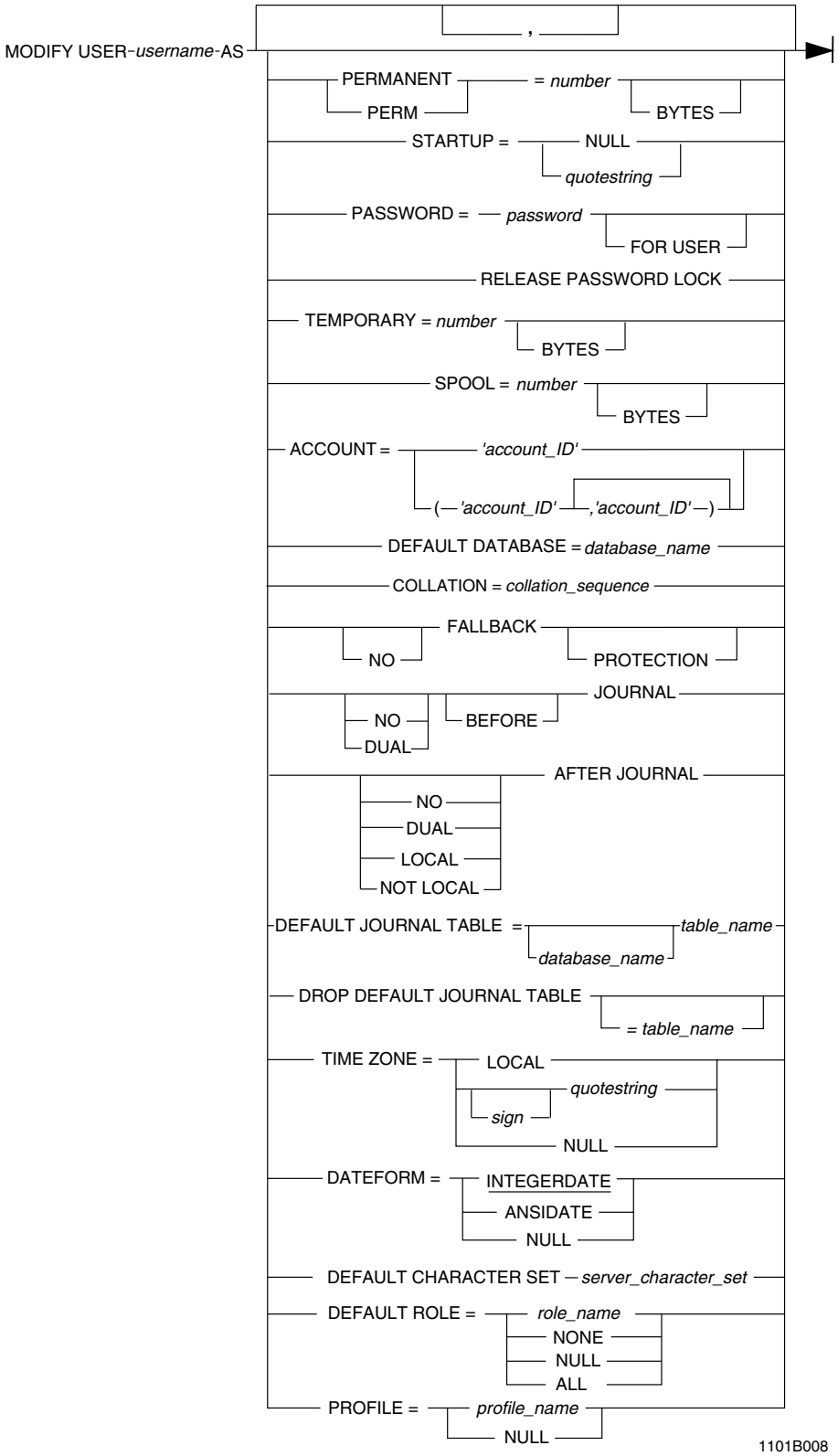
# MODIFY DATABASE



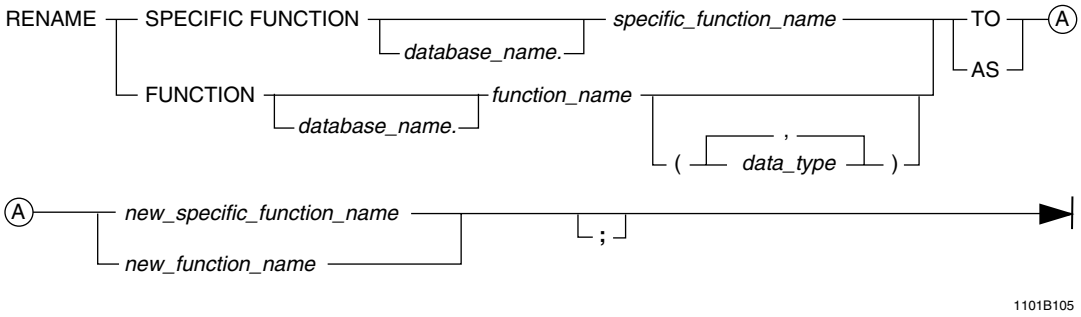
# MODIFY PROFILE



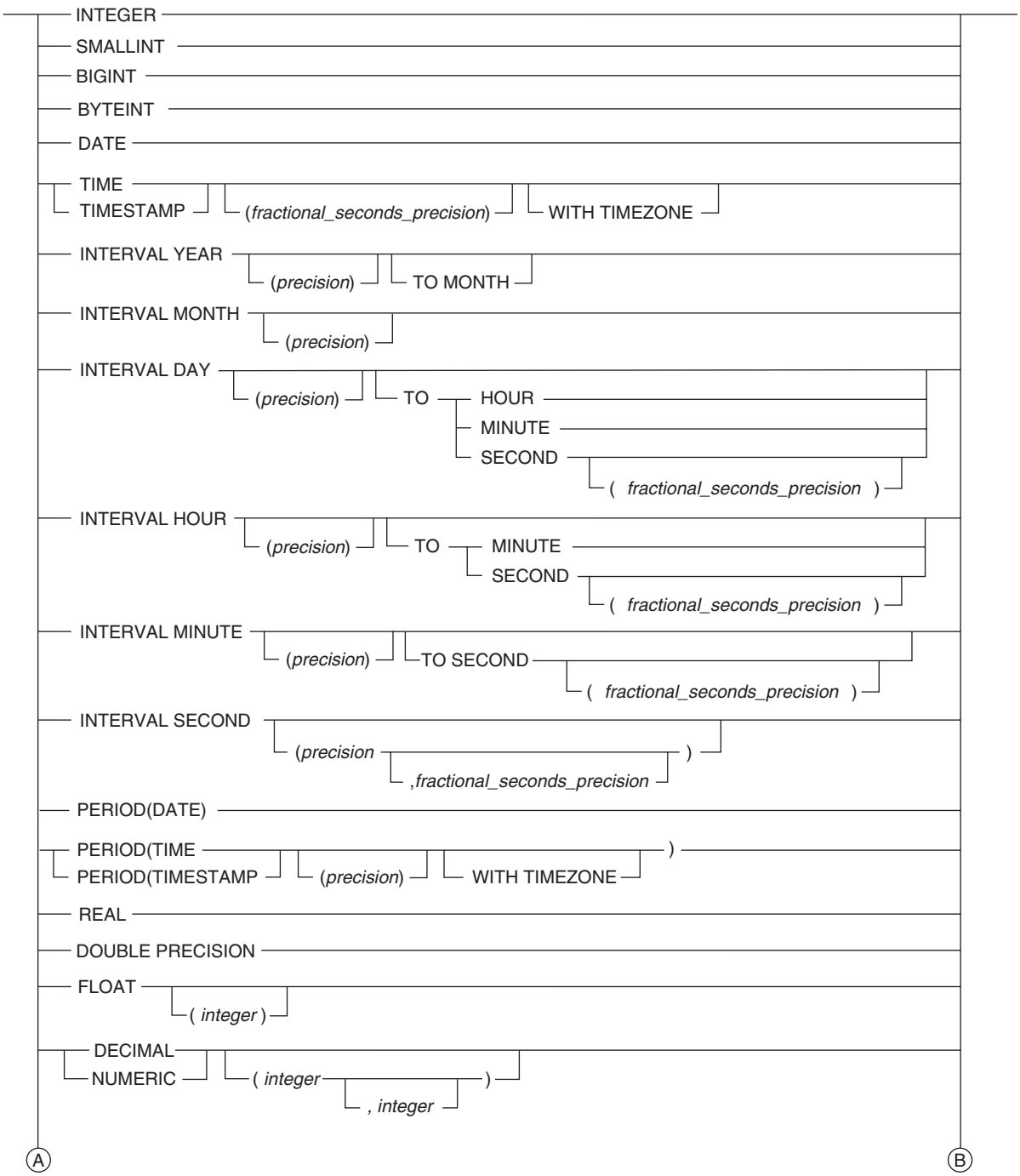
# MODIFY USER


MODIFY USER username AS
,
PERMANENT
PERM
= number
BYTES
STARTUP =
NULL
quotestring
PASSWORD =
password
FOR USER
RELEASE PASSWORD LOCK
TEMPORARY = number
BYTES
SPOOL = number
BYTES
ACCOUNT =
'account_ID'
('account_ID' ,'account_ID' )
DEFAULT DATABASE = database_name
COLLATION = collation_sequence
FALLBACK
NO
PROTECTION
JOURNAL
NO
DUAL
BEFORE
AFTER JOURNAL
NO
DUAL
LOCAL
NOT LOCAL
DEFAULT JOURNAL TABLE =
table_name
database_name
DROP DEFAULT JOURNAL TABLE
= table_name
TIME ZONE =
LOCAL
quotestring
sign
NULL
DATEFORM =
INTEGERDATE
ANSIDATE
NULL
DEFAULT CHARACTER SET server_character_set
DEFAULT ROLE =
role_name
NONE
NULL
ALL
PROFILE =
profile_name
NULL
1101B008

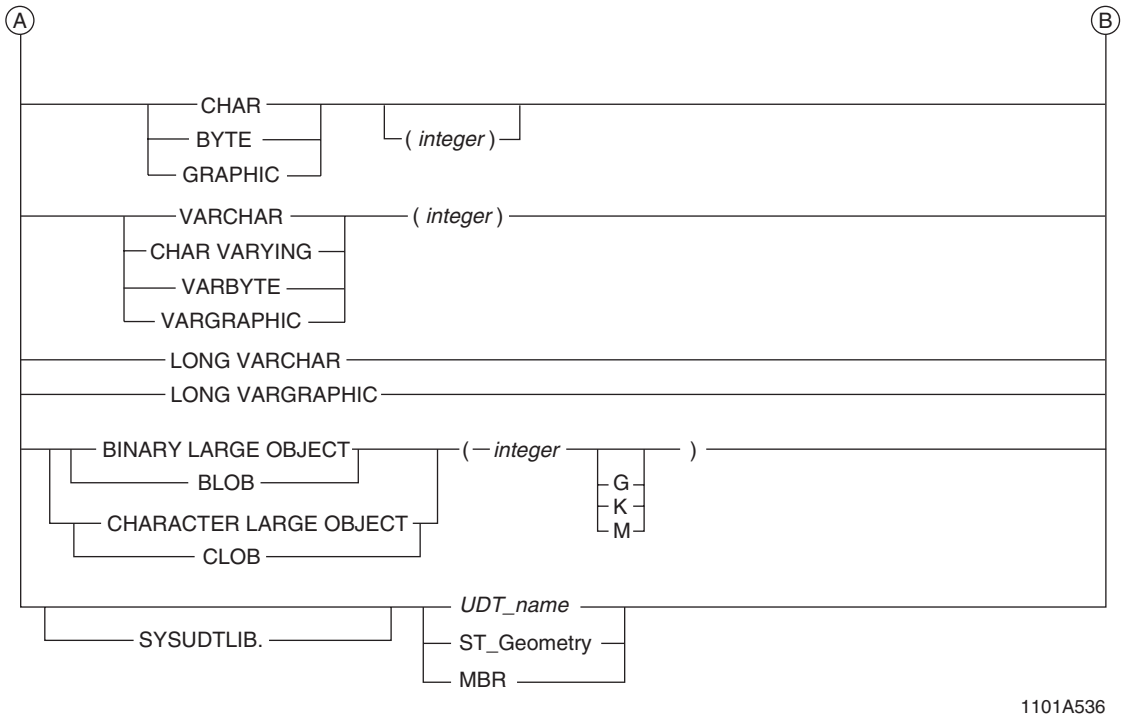
# RENAME FUNCTION



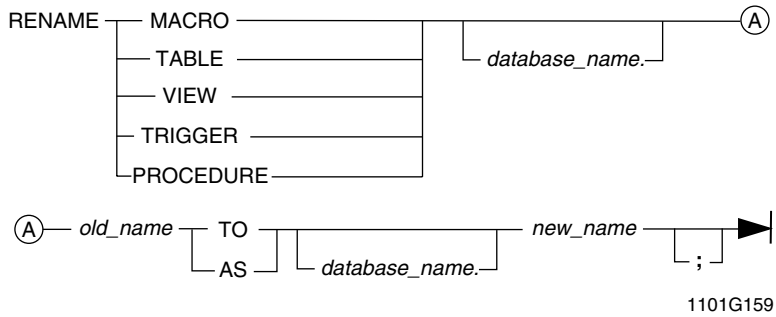
**Data Type**



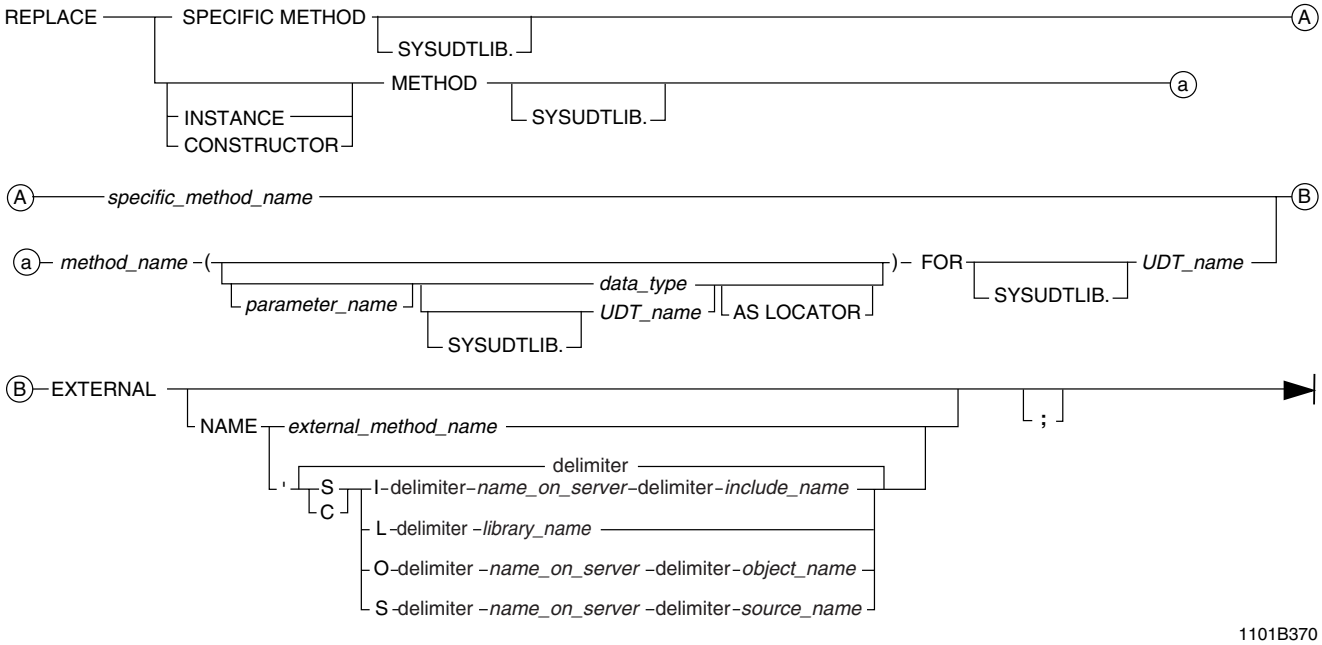

1101A535

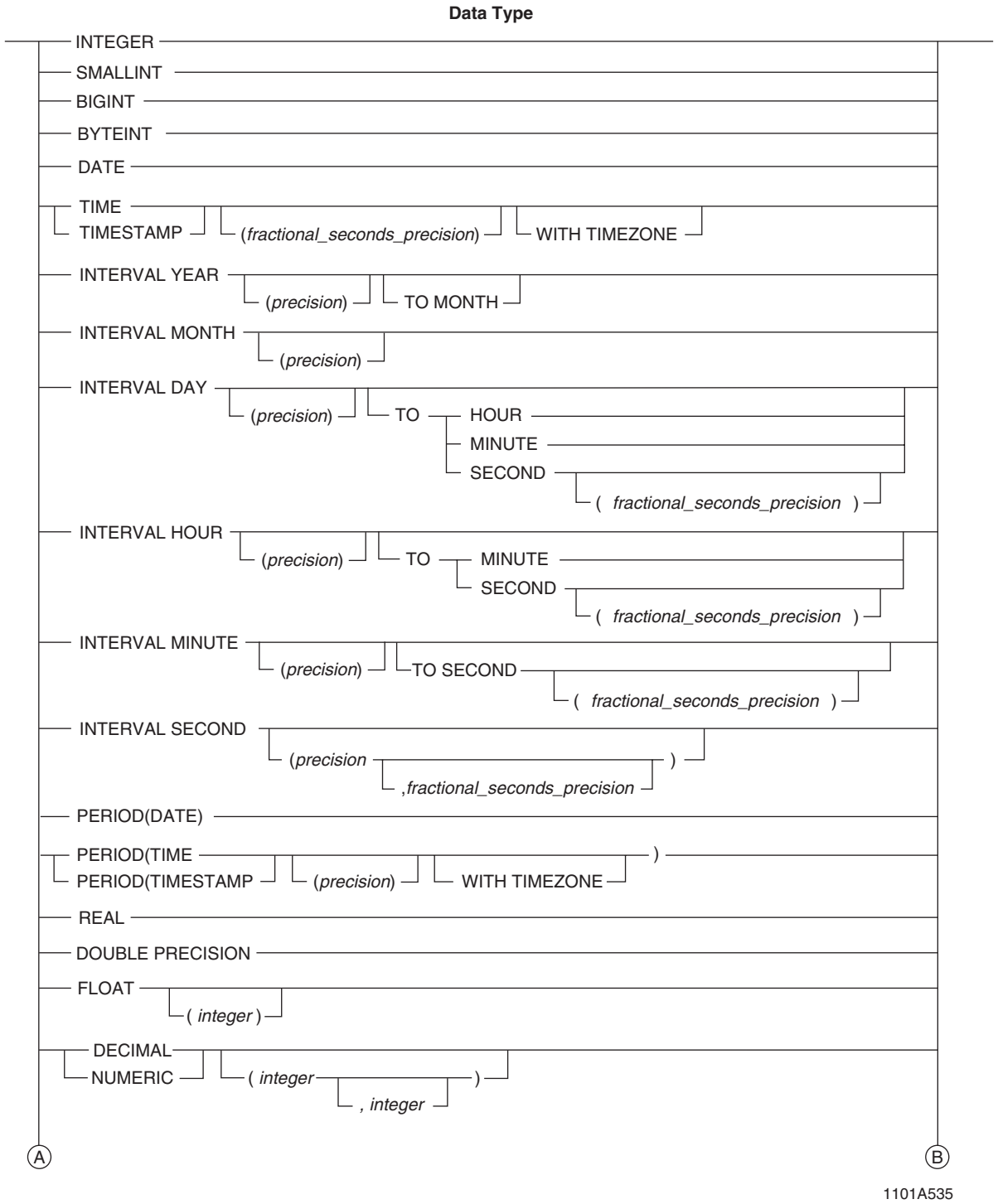


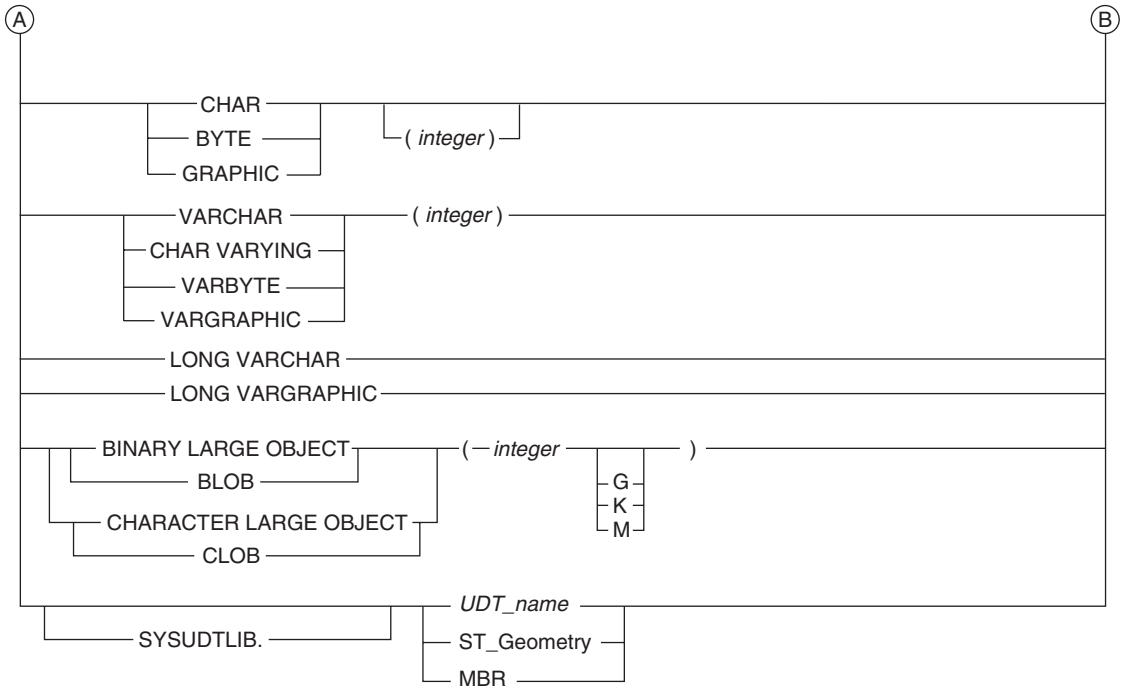

# RENAME MACRO/ RENAME PROCEDURE/ RENAME TABLE/ RENAME TRIGGER/ RENAME VIEW



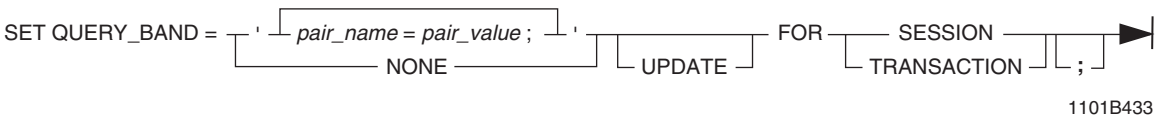
# REPLACE METHOD



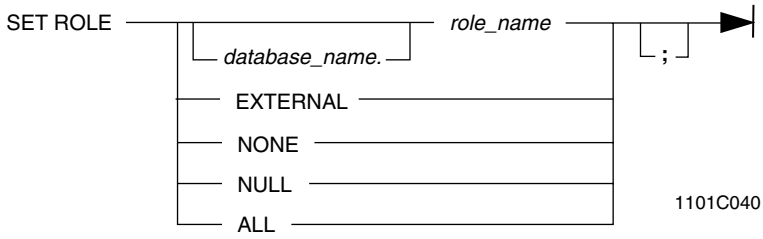

Data Type
INTEGER
SMALLINT
BIGINT
BYTEINT
DATE
TIME
TIMESTAMP
(fractional_seconds_precision)
WITH TIMEZONE
INTERVAL YEAR
(precision)
TO MONTH
INTERVAL MONTH
(precision)
INTERVAL DAY
(precision)
TO
HOUR
MINUTE
SECOND
( fractional_seconds_precision )
INTERVAL HOUR
(precision)
TO
MINUTE
SECOND
( fractional_seconds_precision )
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



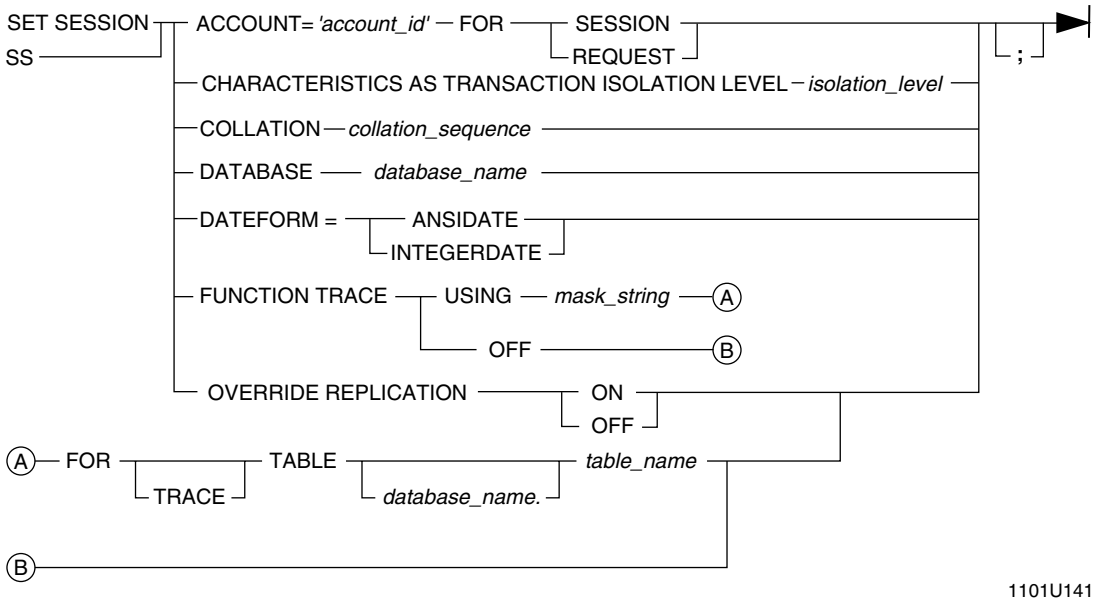

# SET QUERY_BAND



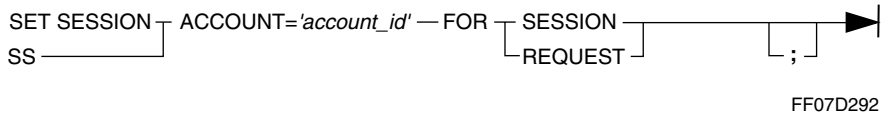
# SET ROLE



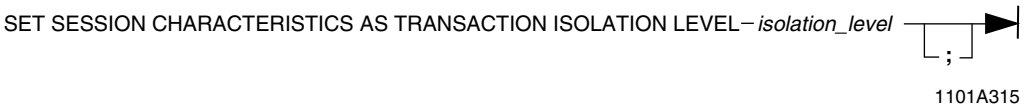

# SET SESSION




# SET SESSION ACCOUNT

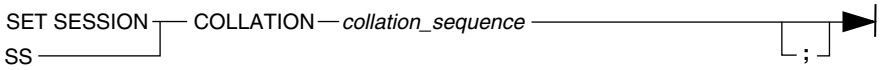


# SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL



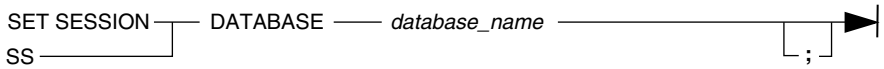

1101A315

# SET SESSION COLLATION



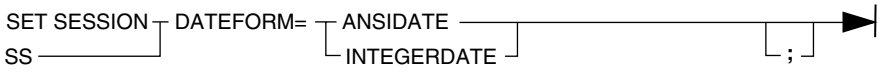

FF07D291

# SET SESSION DATABASE



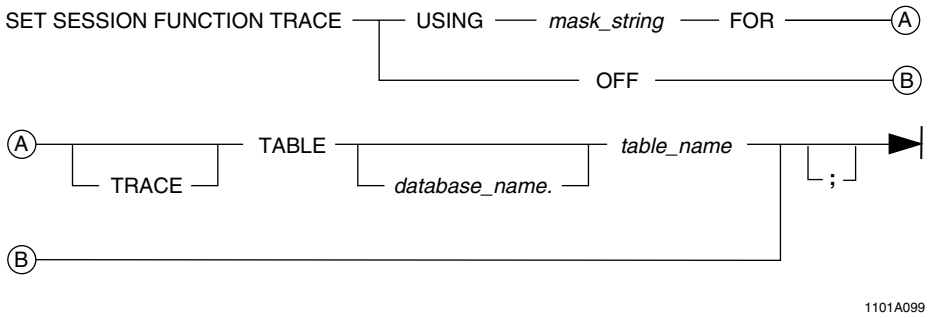

1101A076

# SET SESSION DATEFORM

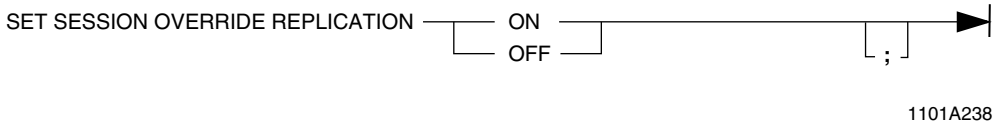



FF07D293

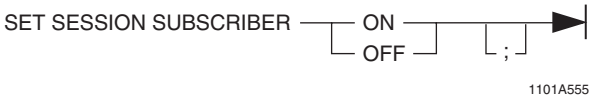
# SET SESSION FUNCTION TRACE



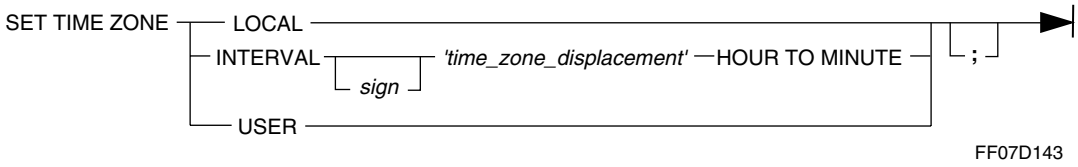

# SET SESSION OVERRIDE REPLICATION



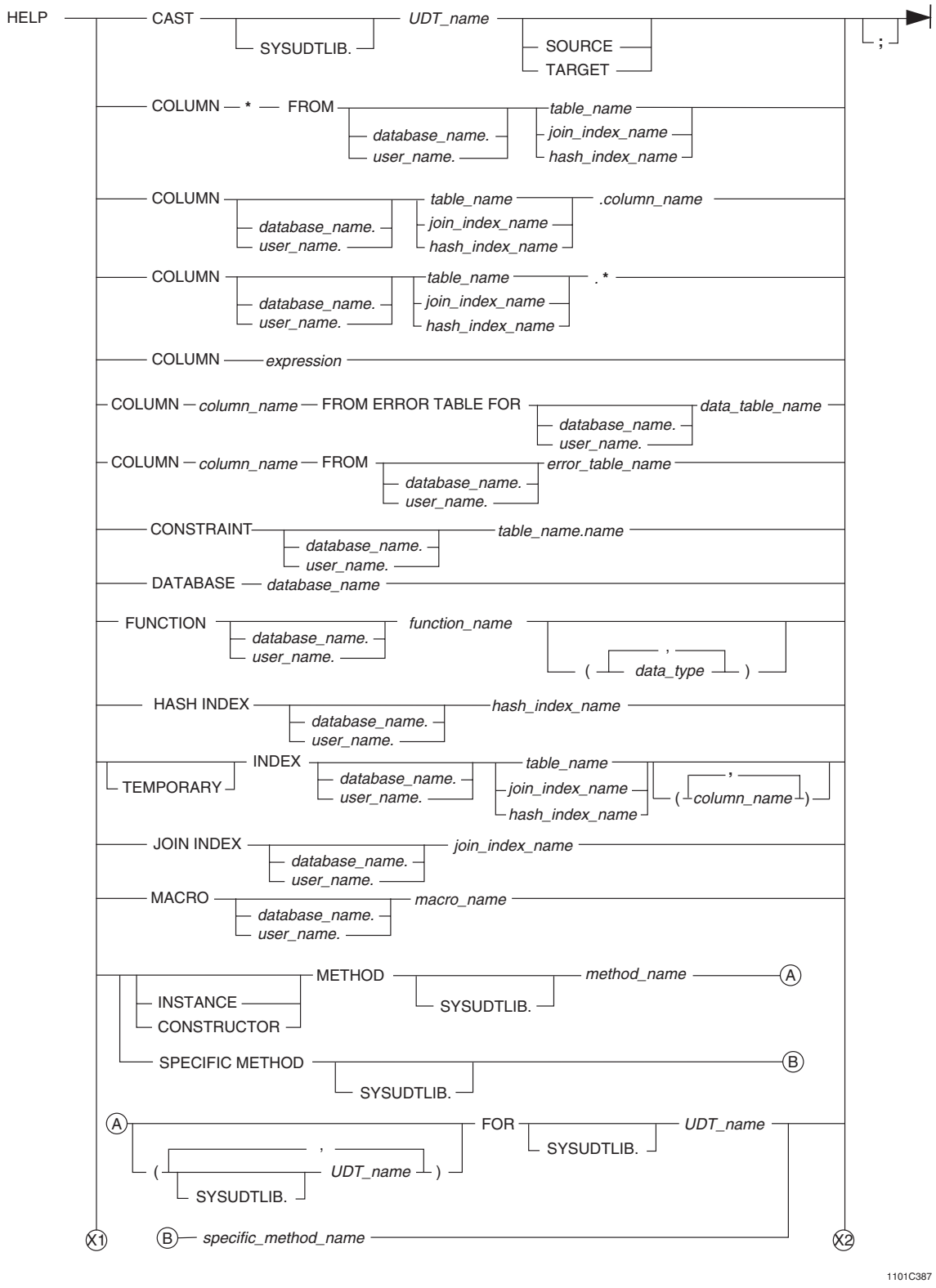

# SET SESSION SUBSCRIBER

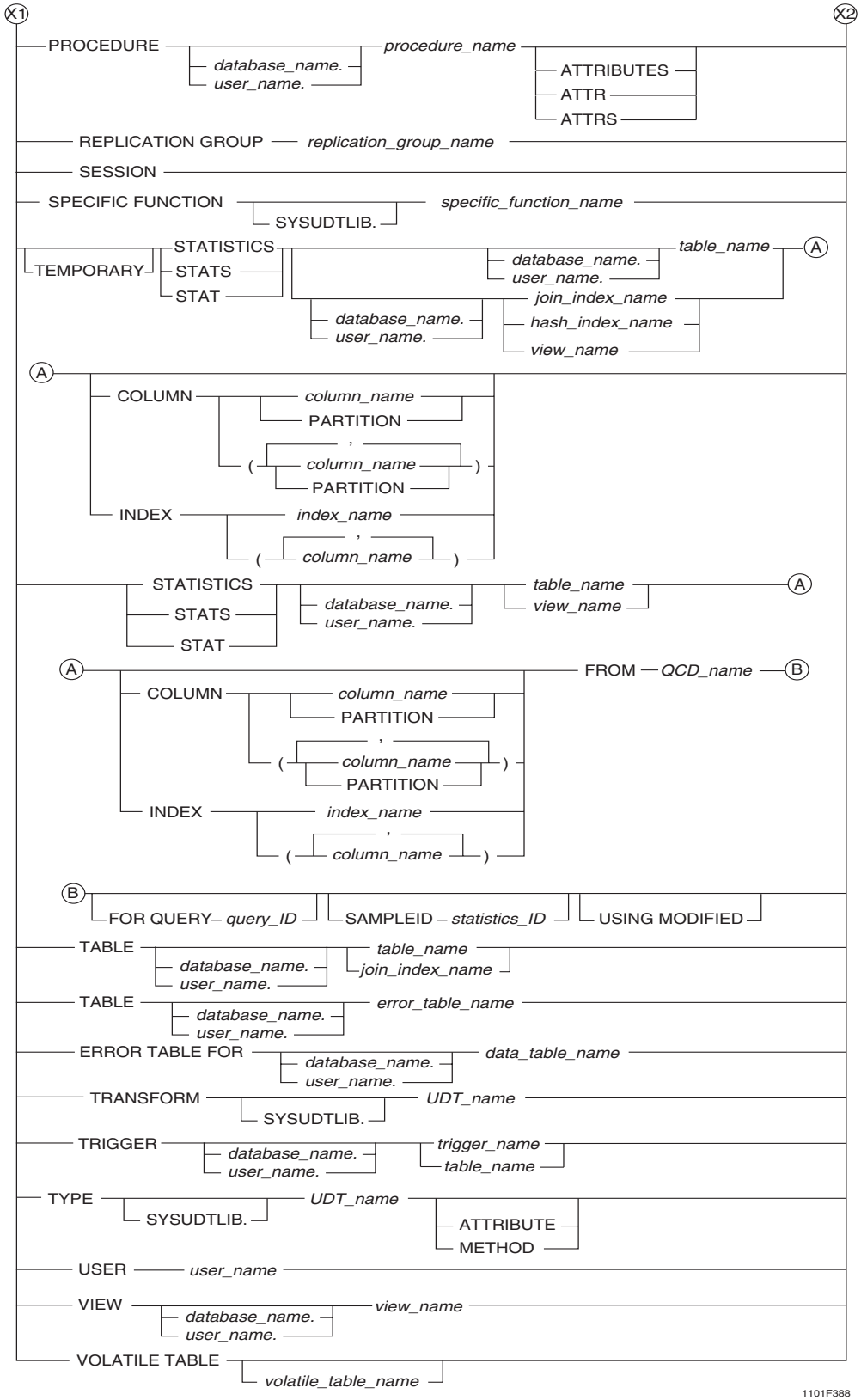



# SET TIME ZONE

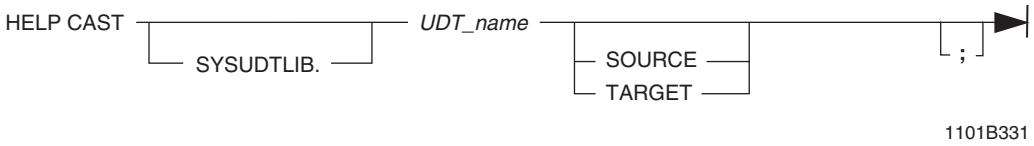


# HELP


HELP
CAST
SYSUDTLIB.
UDT_name
SOURCE
TARGET
;
COLUMN * FROM
database_name.
user_name.
table_name
join_index_name
hash_index_name
COLUMN
database_name.
user_name.
table_name
join_index_name
hash_index_name
.column_name
COLUMN
database_name.
user_name.
table_name
join_index_name
hash_index_name
.*
COLUMN
expression
COLUMN column_name FROM ERROR TABLE FOR
database_name.
user_name.
data_table_name
COLUMN column_name FROM
database_name.
user_name.
error_table_name
CONSTRAINT
database_name.
user_name.
table_name.name
DATABASE database_name
FUNCTION
database_name.
user_name.
function_name
( data_type )
HASH INDEX
database_name.
user_name.
hash_index_name
TEMPORARY
INDEX
database_name.
user_name.
table_name
join_index_name
hash_index_name
( column_name )
JOIN INDEX
database_name.
user_name.
join_index_name
MACRO
database_name.
user_name.
macro_name
METHOD
INSTANCE
CONSTRUCTOR
SYSUDTLIB.
method_name
A
SPECIFIC METHOD
SYSUDTLIB.
B
A
( SYSUDTLIB. UDT_name )
FOR
SYSUDTLIB.
UDT_name
X1
B
specific_method_name
X2
1101C387

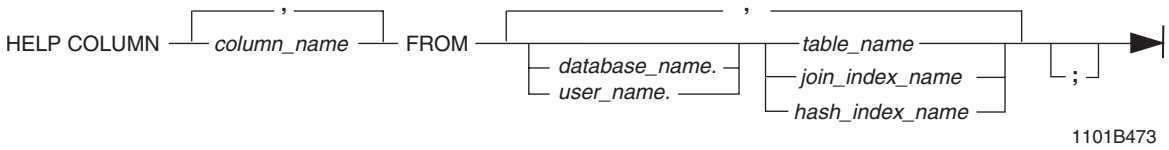

X1
X2
PROCEDURE
database_name.
user_name.
procedure_name
ATTRIBUTES
ATTR
ATTRS
REPLICATION GROUP
replication_group_name
SESSION
SPECIFIC FUNCTION
SYSUDTLIB.
specific_function_name
TEMPORARY
STATISTICS
STATS
STAT
database_name.
user_name.
table_name
A
join_index_name
hash_index_name
view_name
COLUMN
column_name
PARTITION
INDEX
index_name
STATISTICS
STATS
STAT
database_name.
user_name.
table_name
view_name
FROM
QCD_name
B
FOR QUERY
query_ID
SAMPLEID
statistics_ID
USING MODIFIED
TABLE
database_name.
user_name.
table_name
join_index_name
error_table_name
ERROR TABLE FOR
data_table_name
TRANSFORM
SYSUDTLIB.
UDT_name
TRIGGER
trigger_name
TYPE
ATTRIBUTE
METHOD
USER
user_name
VIEW
view_name
VOLATILE TABLE
volatile_table_name
1101F388

# HELP CAST

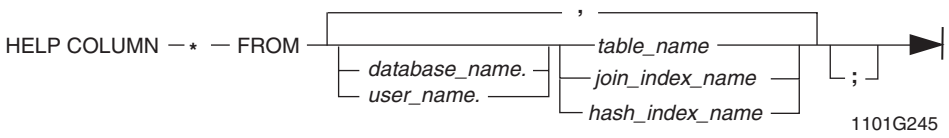



# HELP COLUMN

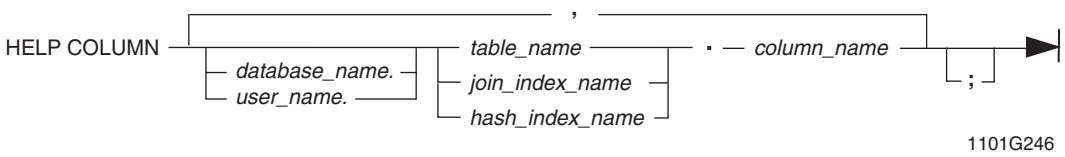
## Syntax 1



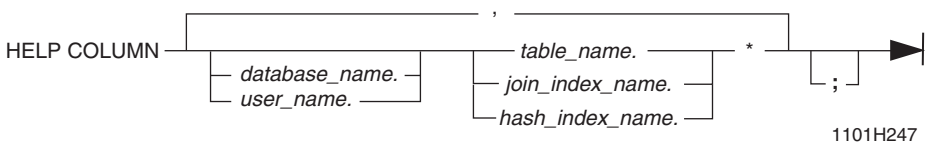

## Syntax 2



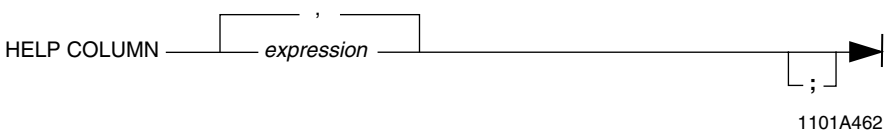

## Syntax 3



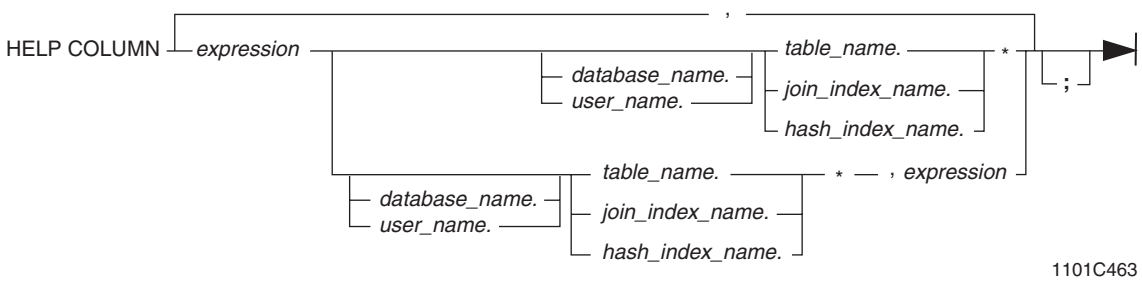

## Syntax 4



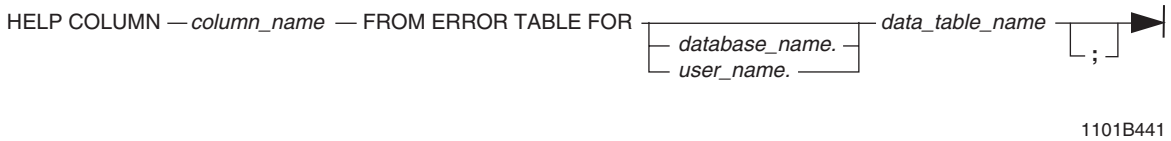
## Syntax 5



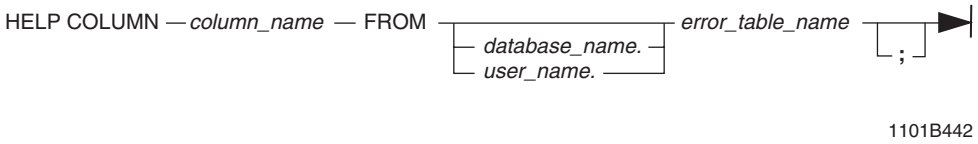

## Syntax 6



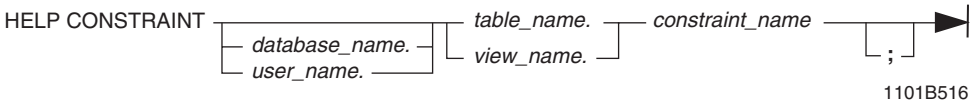

## Syntax 7



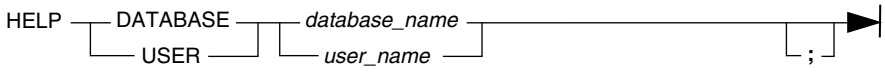

## Syntax 8




# HELP CONSTRAINT



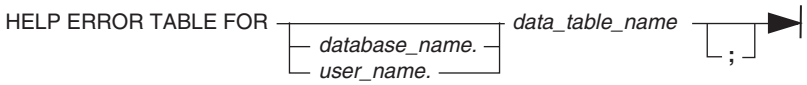
# HELP DATABASE/ HELP USER




FF07D240

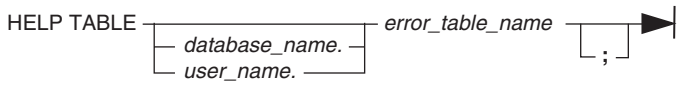
# HELP ERROR TABLE

## Syntax 1



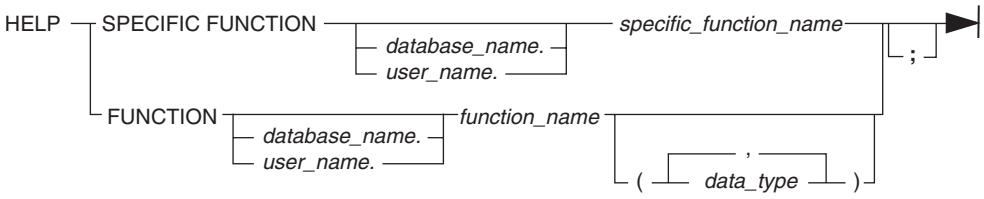

1101B439

## Syntax 2



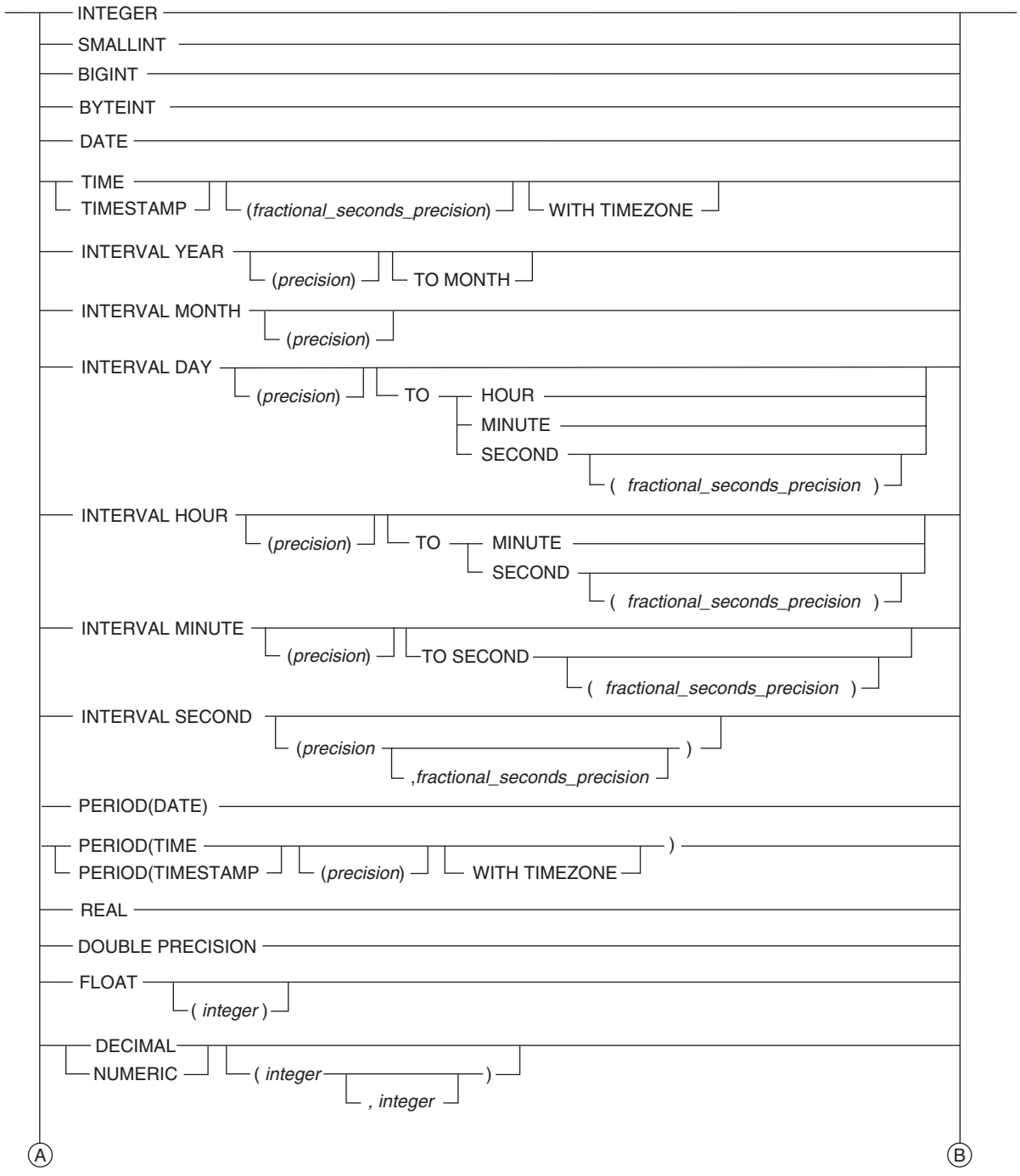

1101B440

# HELP FUNCTION

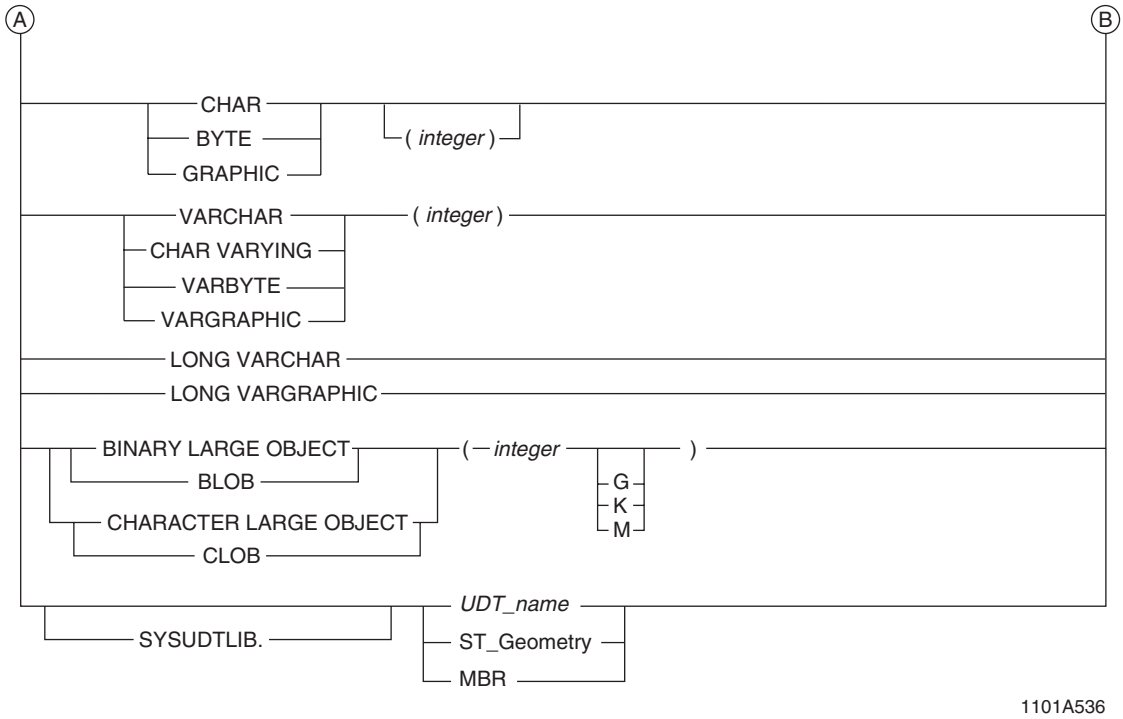



1101C098

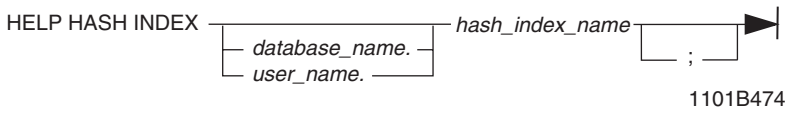
**Data Type**



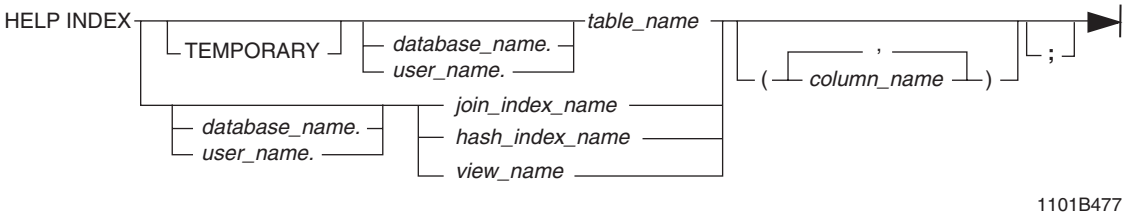

1101A535



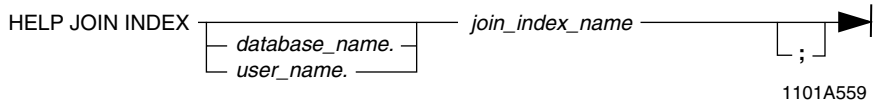

# HELP HASH INDEX



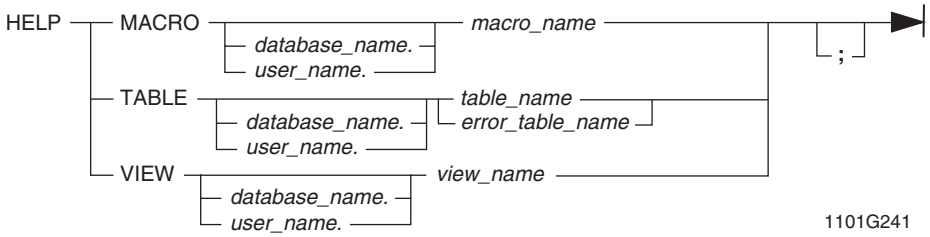

# HELP INDEX



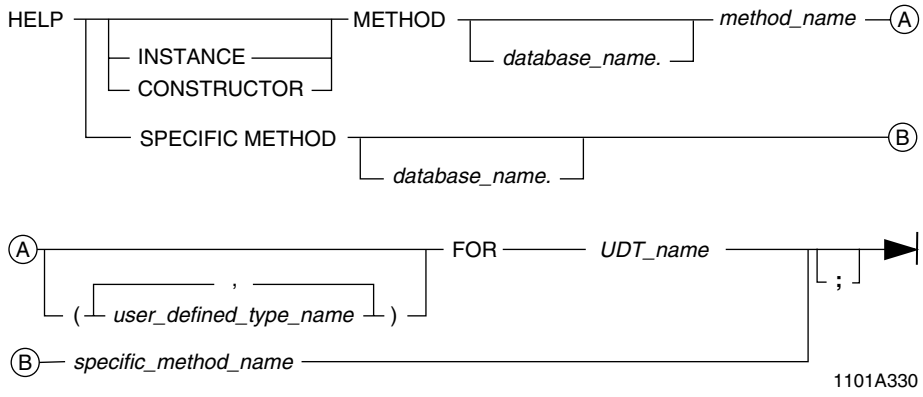
# HELP JOIN INDEX



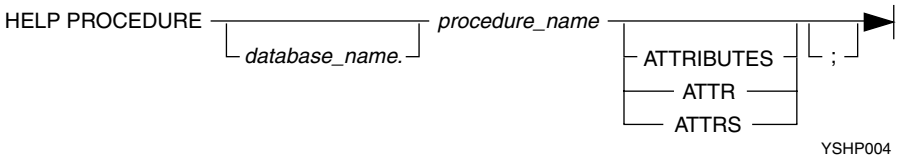

# HELP MACRO/ HELP TABLE/ HELP VIEW



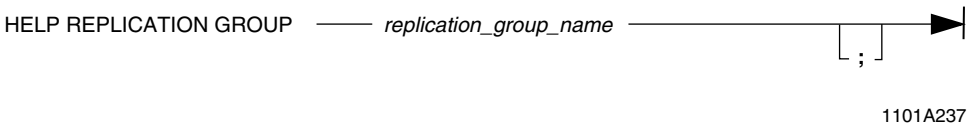

# HELP METHOD



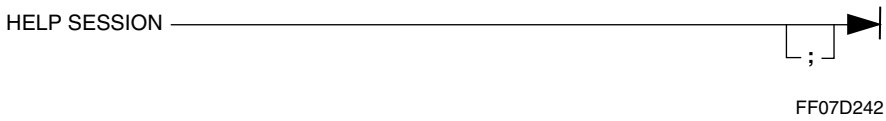
# HELP PROCEDURE



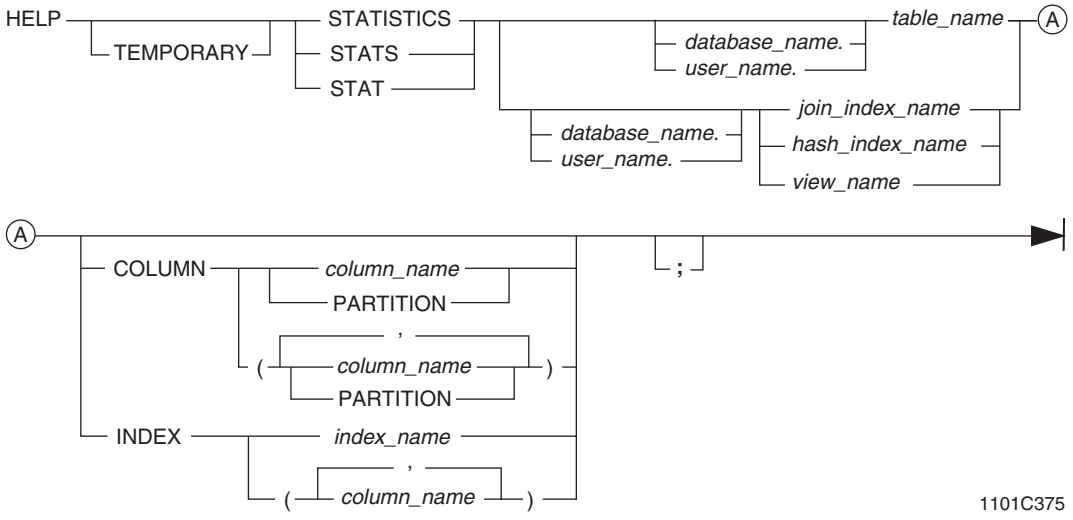

# HELP REPLICATION GROUP



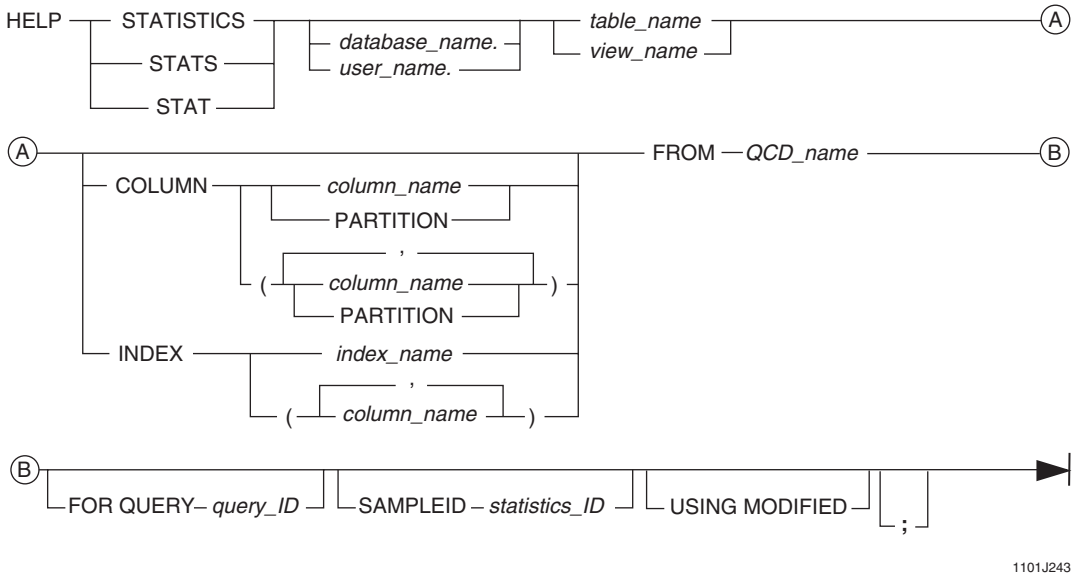

# HELP SESSION



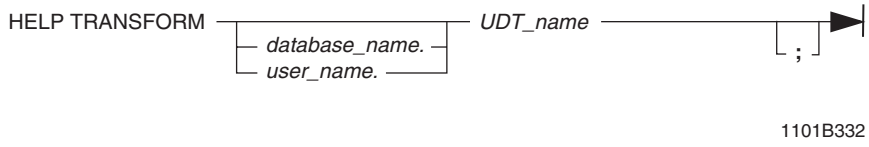

# HELP STATISTICS (Optimizer Form)



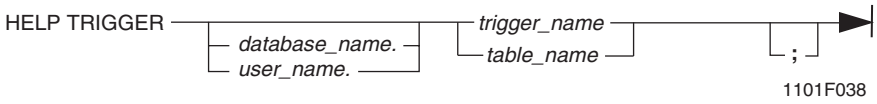
# HELP STATISTICS (QCD Form)



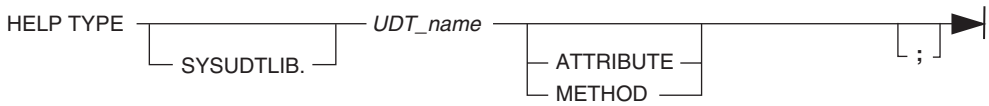

# HELP TRANSFORM



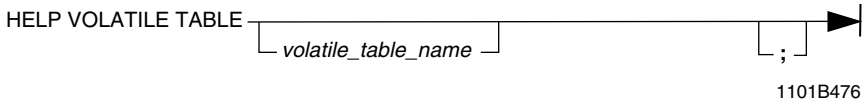

# HELP TRIGGER



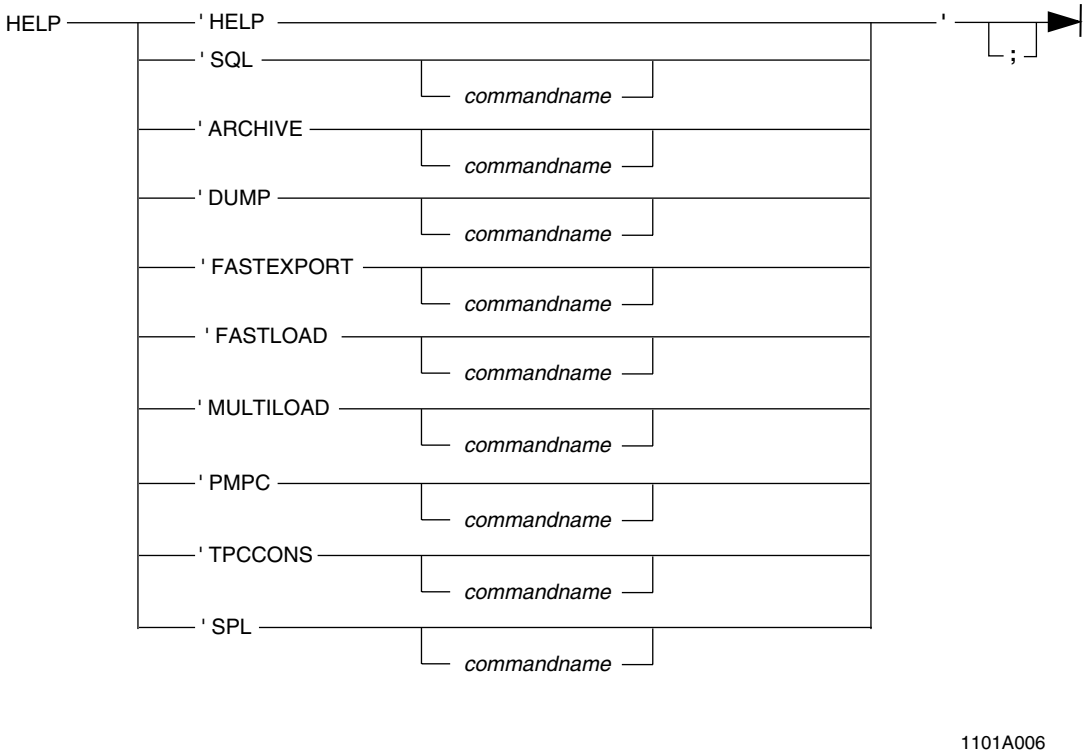
# HELP TYPE



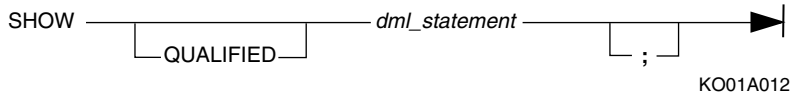

# HELP VOLATILE TABLE




# HELP (Online Form)

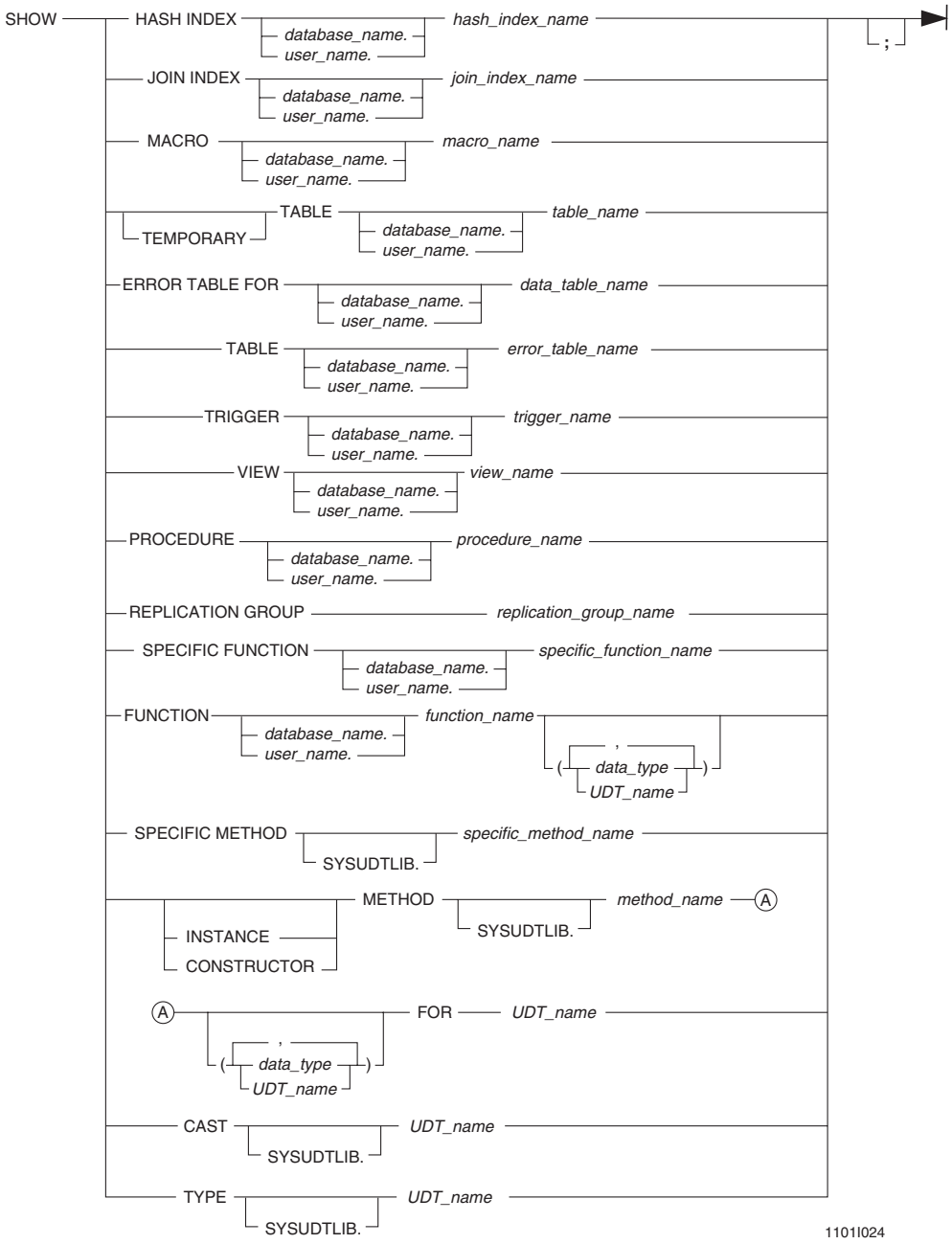


# SHOW

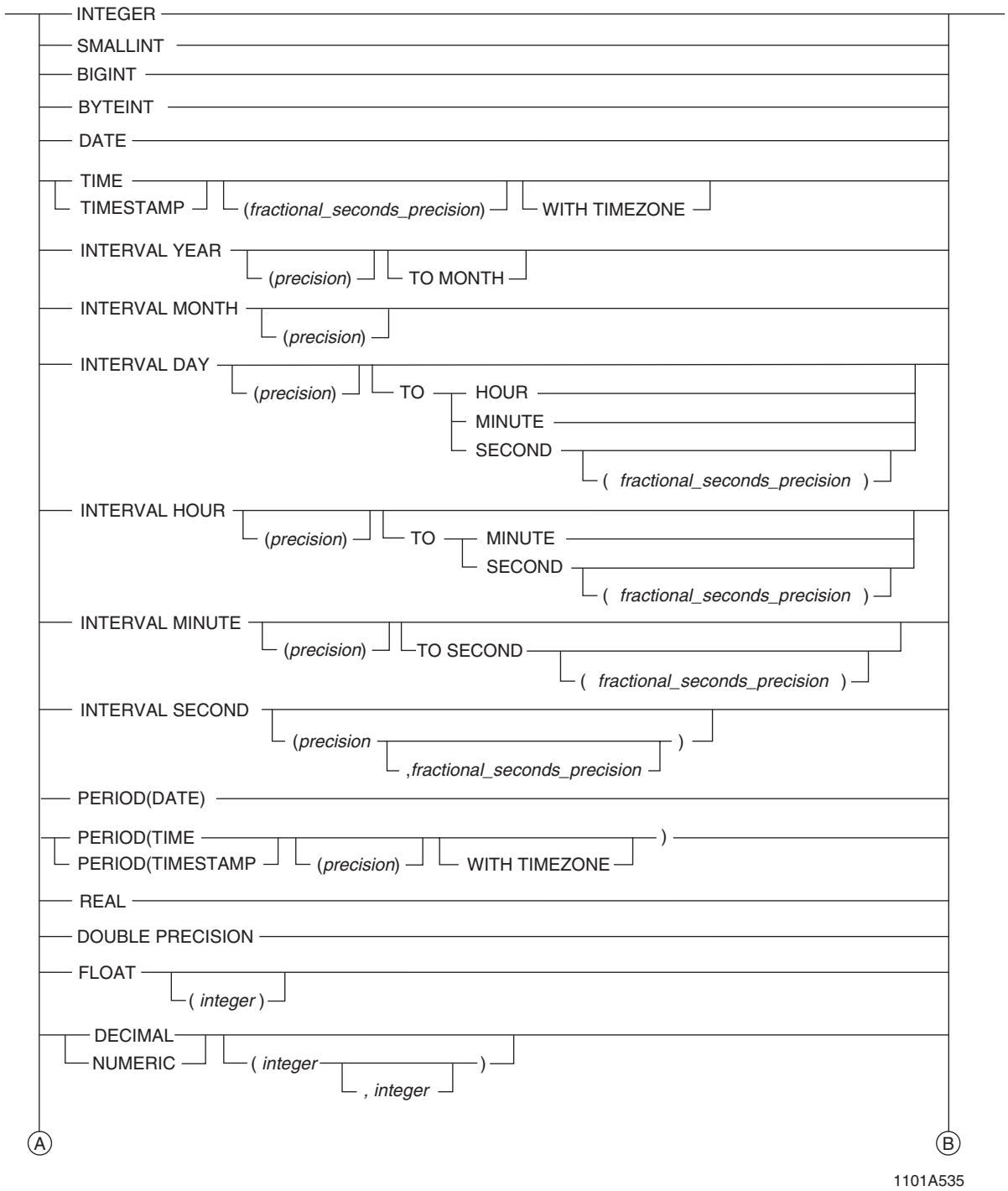



## SHOW CAST/
## SHOW ERROR TABLE/
## SHOW FUNCTION/
## SHOW HASH INDEX/
## SHOW JOIN INDEX/
## SHOW MACRO/
## SHOW METHOD/
## SHOW PROCEDURE/
## SHOW REPLICATION GROUP/
## SHOW TABLE/
## SHOW TRIGGER/
## SHOW TYPE/
## SHOW VIEW

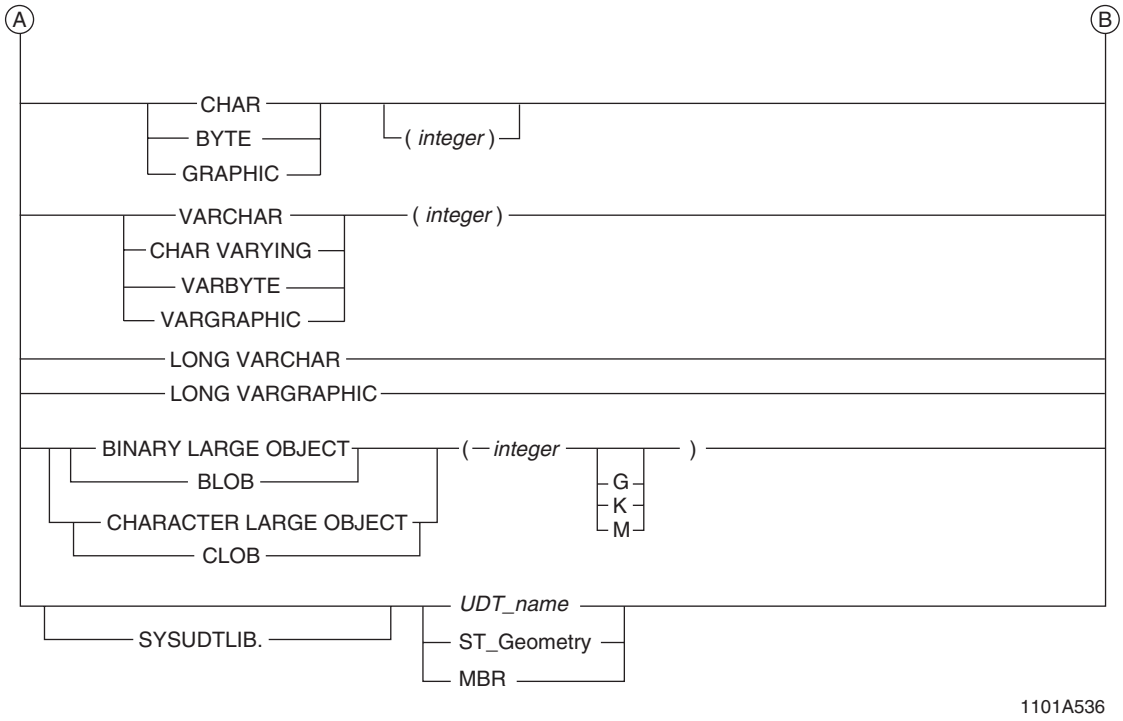
## General Syntax



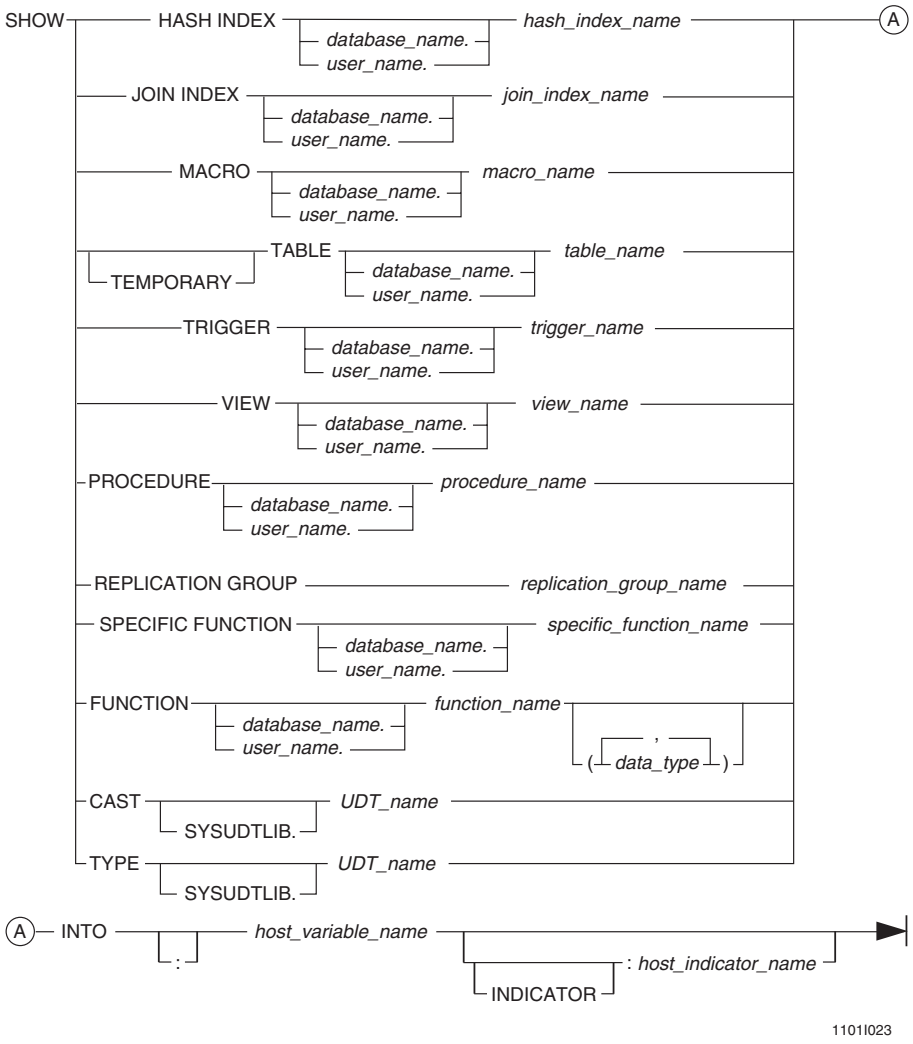
**Data Type**



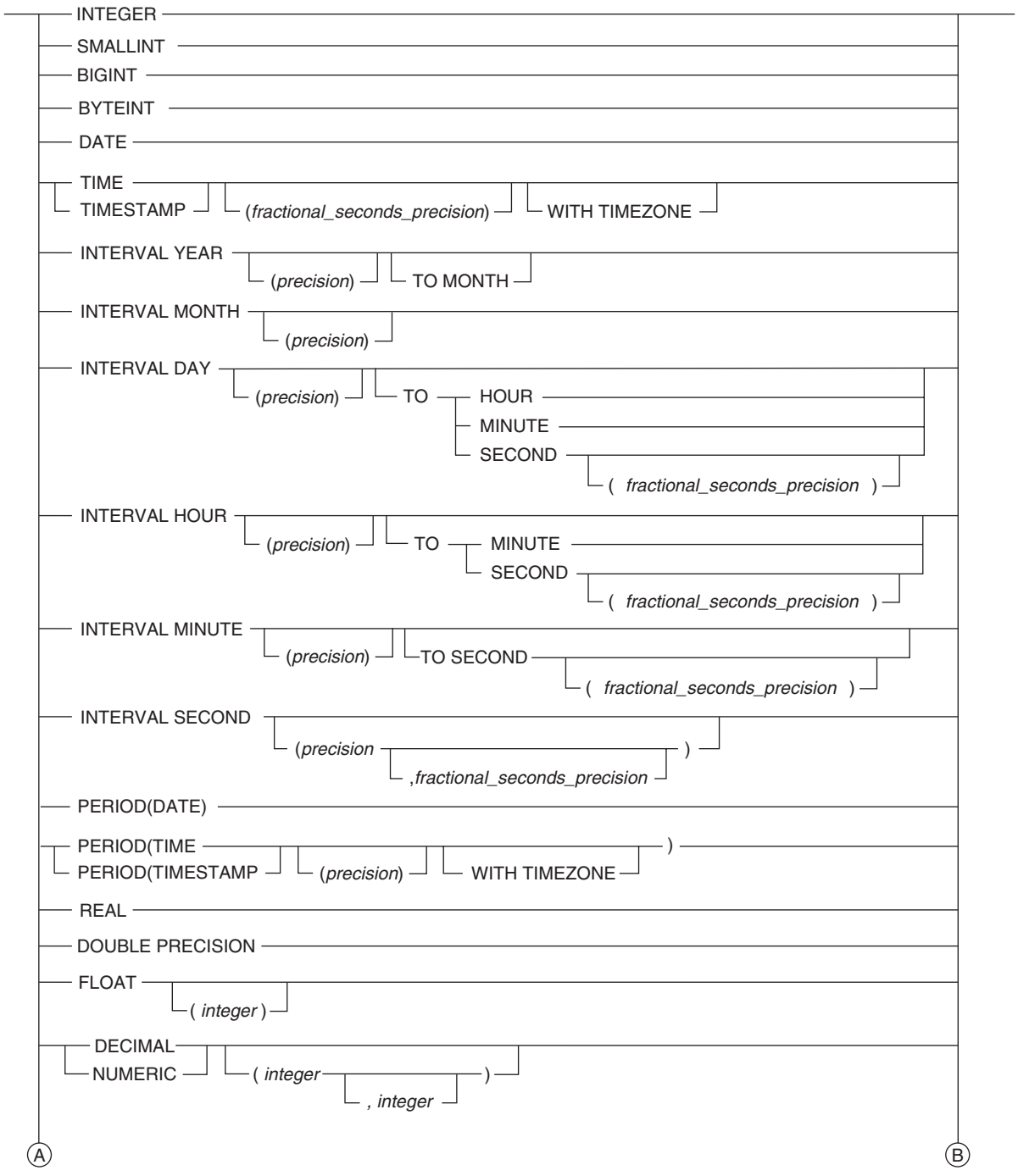

1101A535


A
B
CHAR
BYTE
GRAPHIC
( integer )
VARCHAR
CHAR VARYING
VARBYTE
VARGRAPHIC
( integer )
LONG VARCHAR
LONG VARGRAPHIC
BINARY LARGE OBJECT
BLOB
CHARACTER LARGE OBJECT
CLOB
( integer
G
K
M
)
SYSUDTLIB.
UDT_name
ST_Geometry
MBR
1101A536

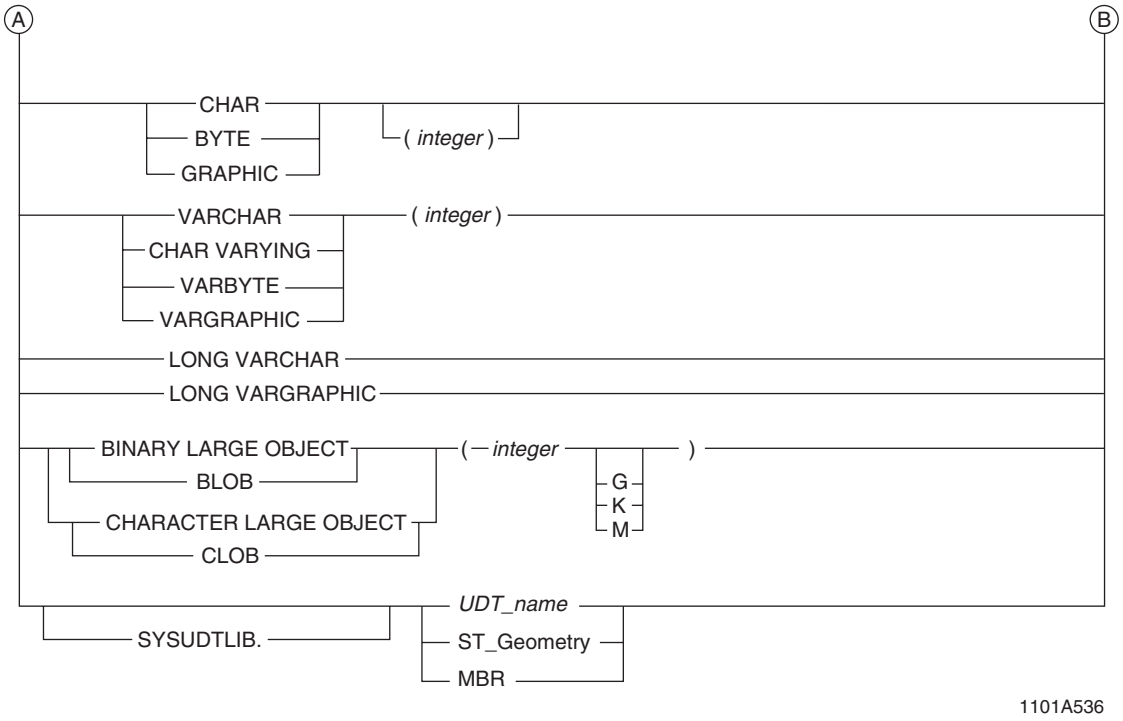
## Embedded SQL Syntax



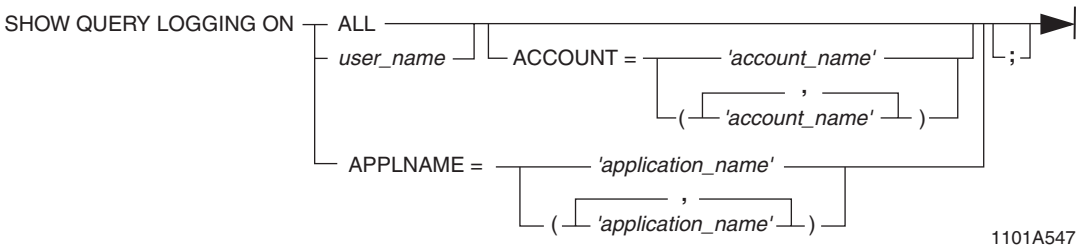
**Data Type**




1101A535

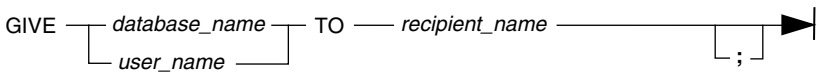



# SHOW QUERY LOGGING



# CHAPTER 4 SQL Data Control Language

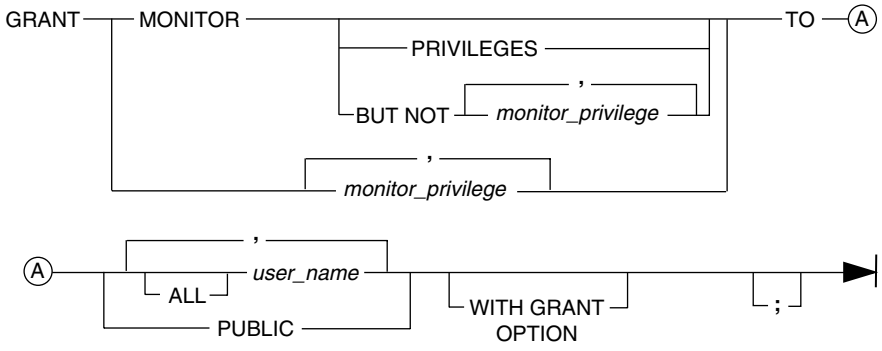
## GIVE




FF07A025

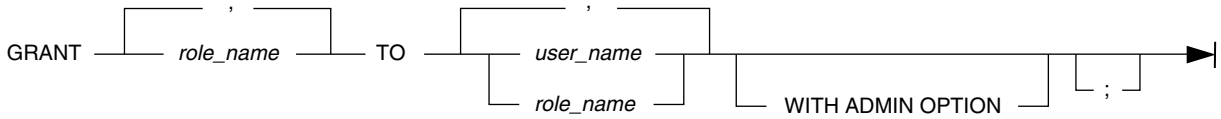
## GRANT

### Monitor Form



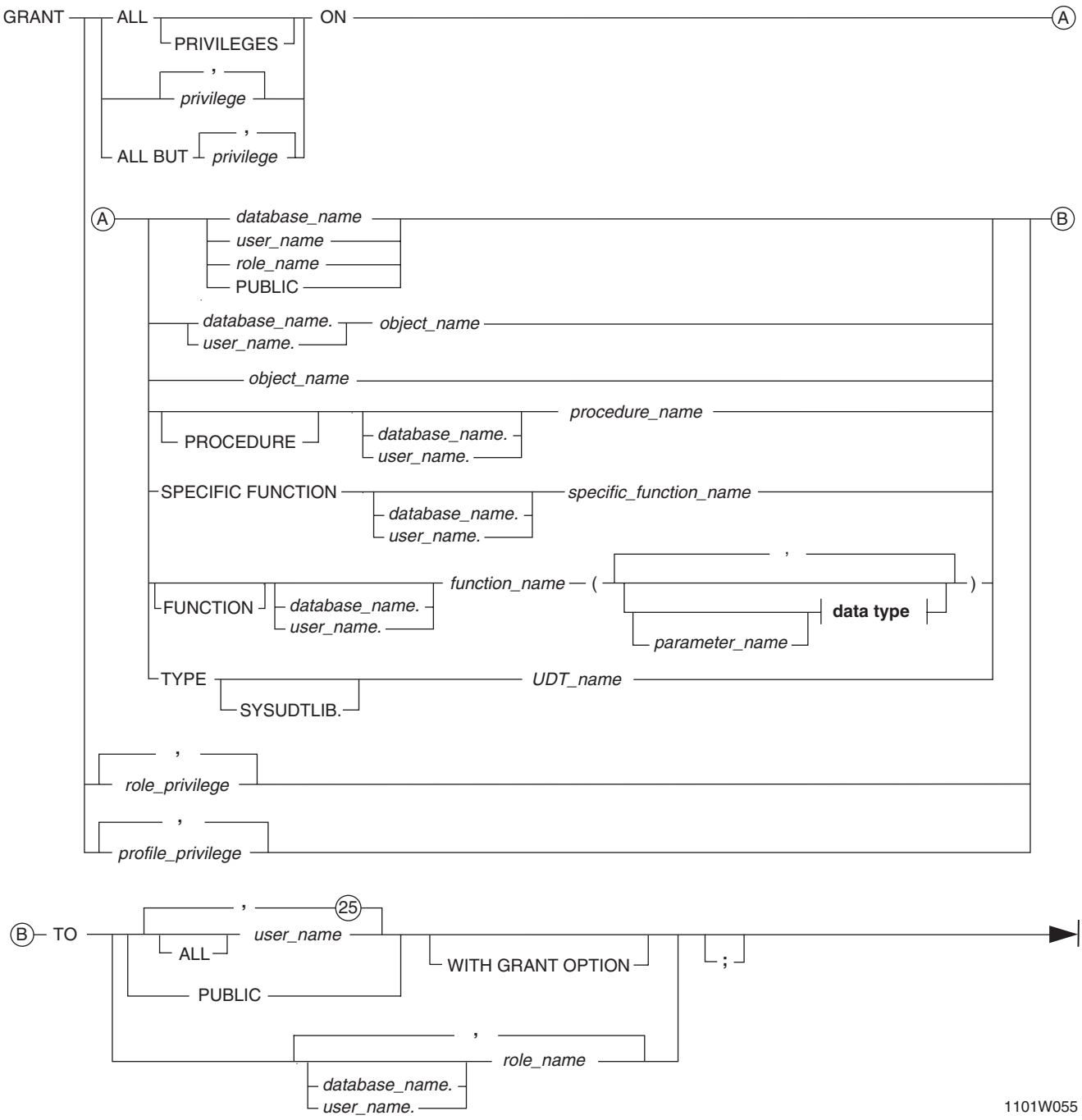

FF07A056

### Role Form

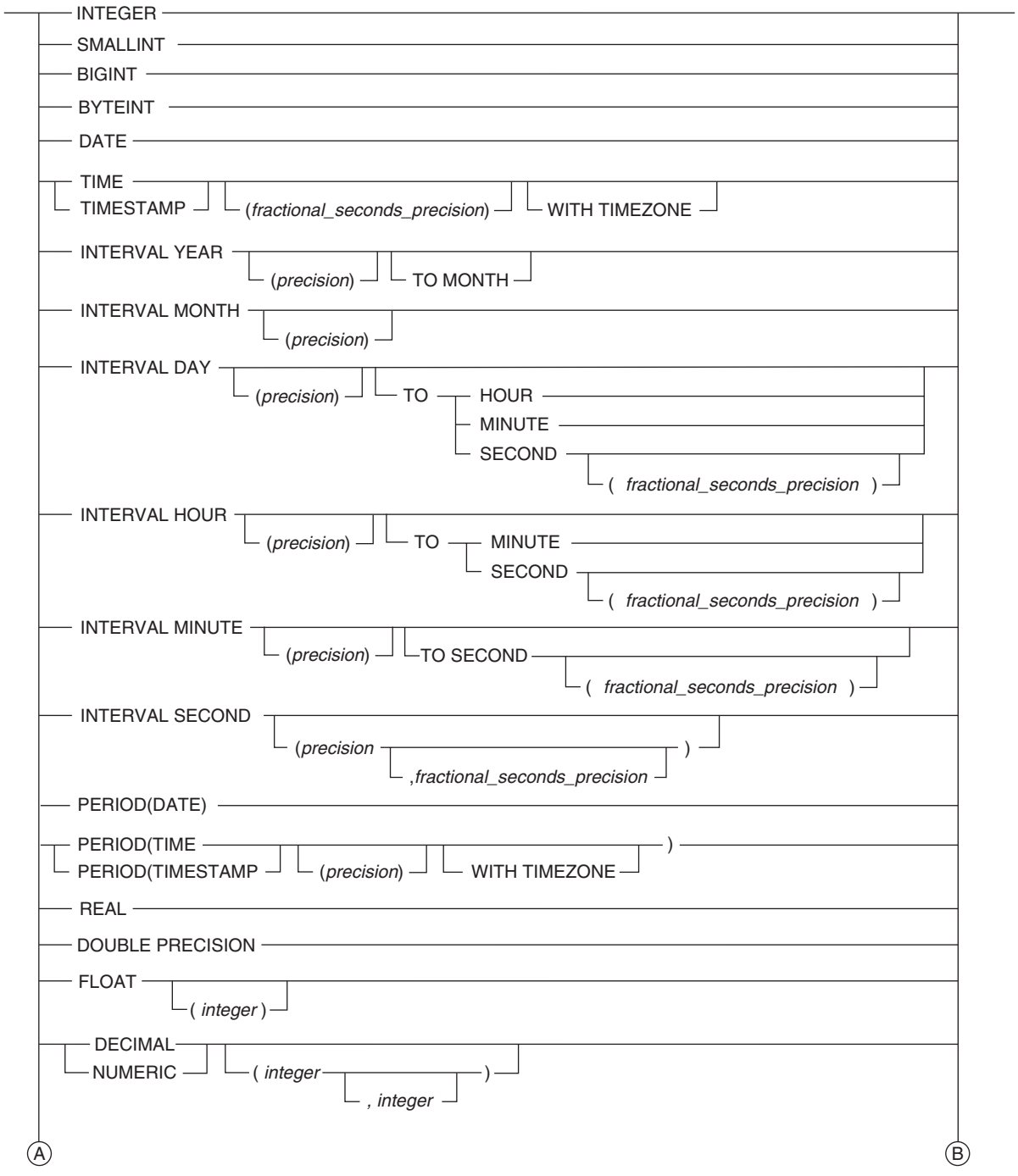



KZ01a008

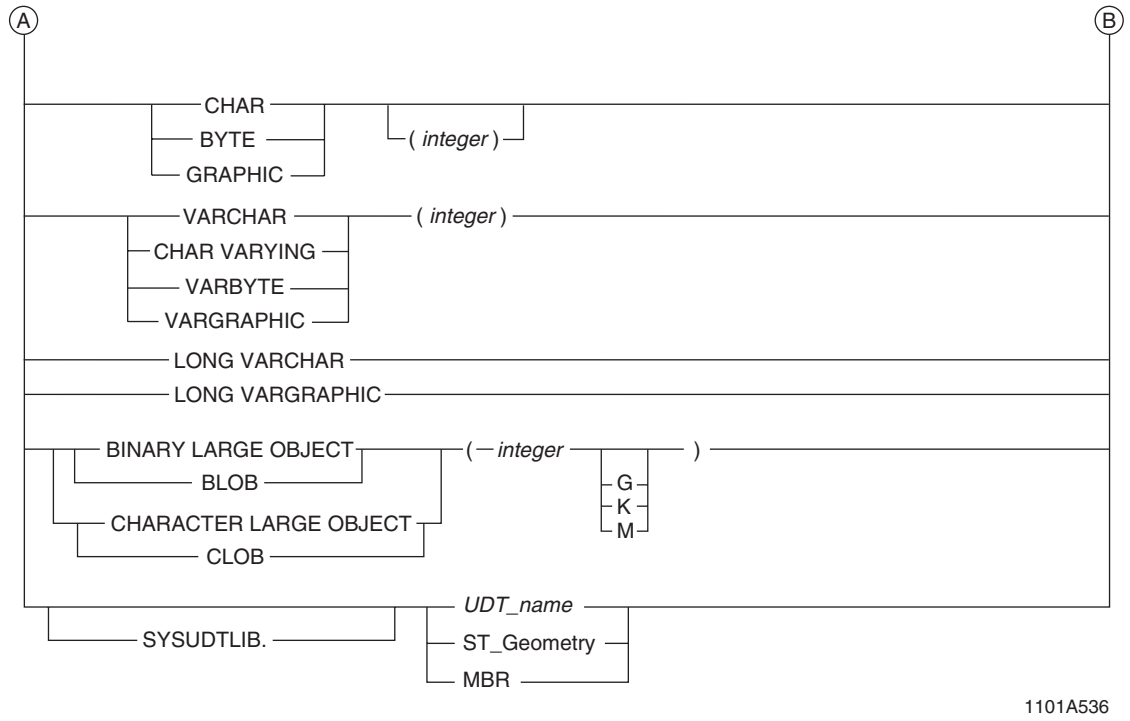
## SQL Form


GRANT
ALL
PRIVILEGES
privilege
ALL BUT
privilege
ON
A
B
database_name
user_name
role_name
PUBLIC
database_name.
user_name.
object_name
PROCEDURE
procedure_name
SPECIFIC FUNCTION
specific_function_name
FUNCTION
function_name
data type
parameter_name
TYPE
SYSUDTLIB.
UDT_name
role_privilege
profile_privilege
TO
25
ALL
PUBLIC
WITH GRANT OPTION
role_name
1101W055

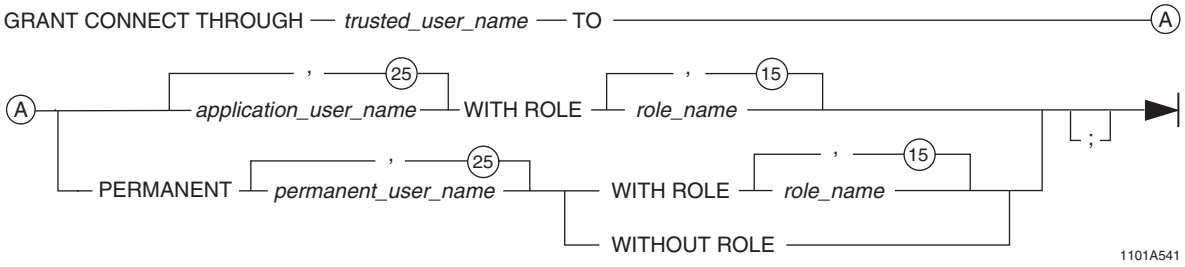
**Data Type**



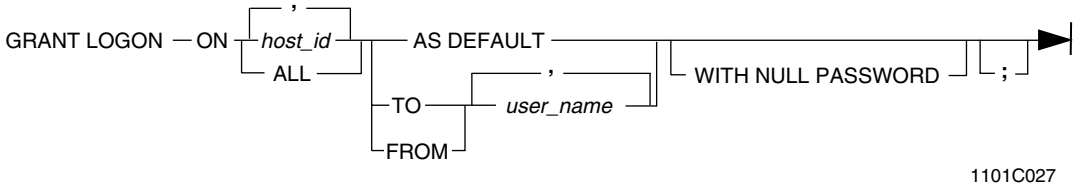

1101A535




# GRANT CONNECT THROUGH

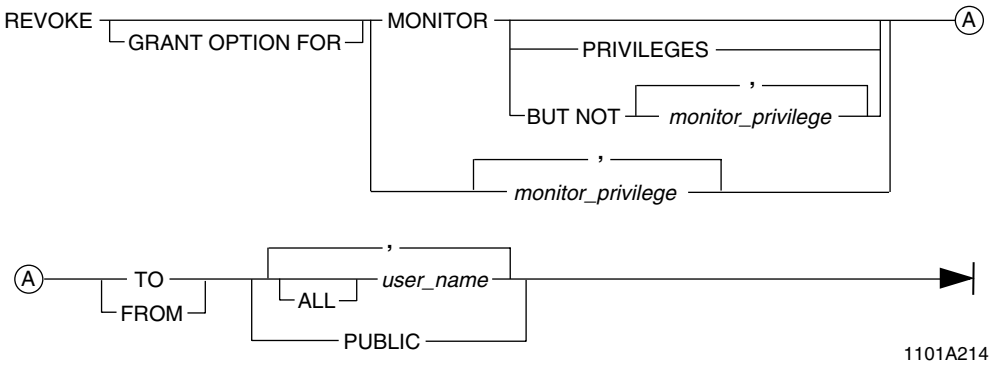


# GRANT LOGON

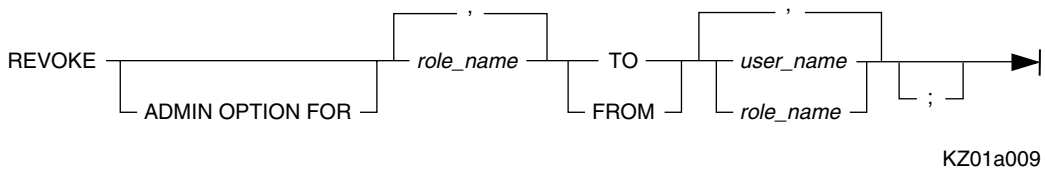



# REVOKE

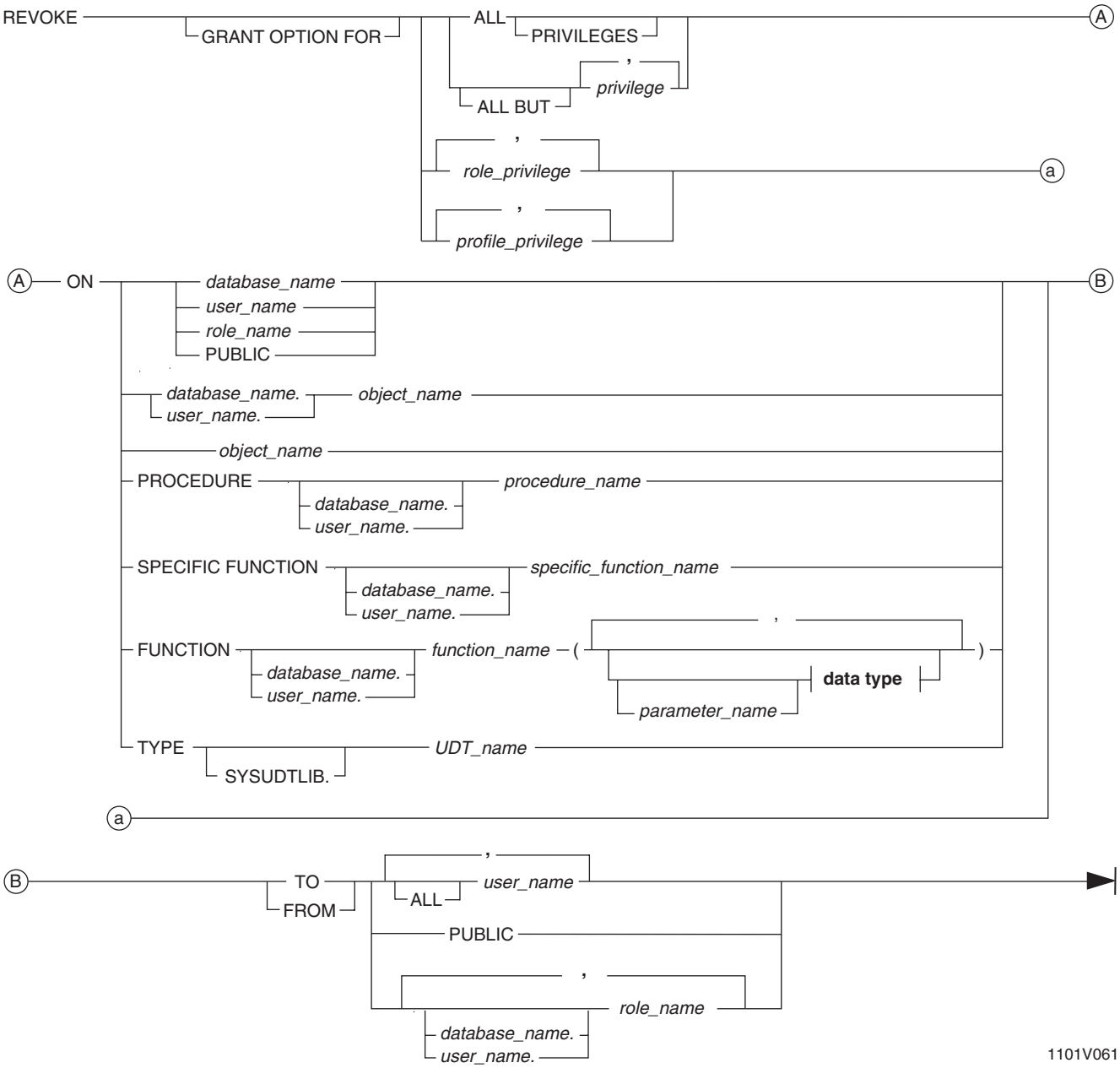
## Monitor Form



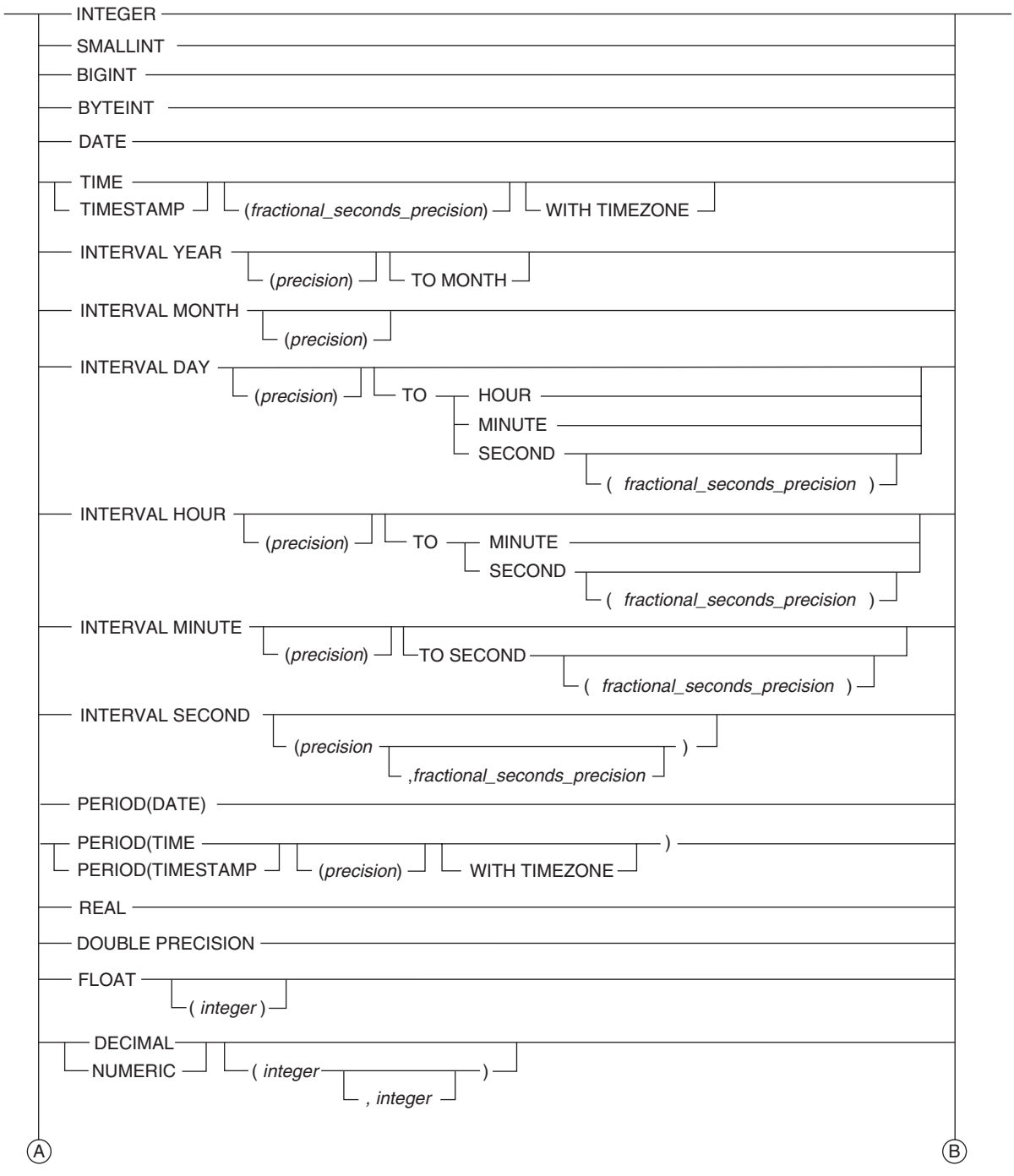

## Role Form



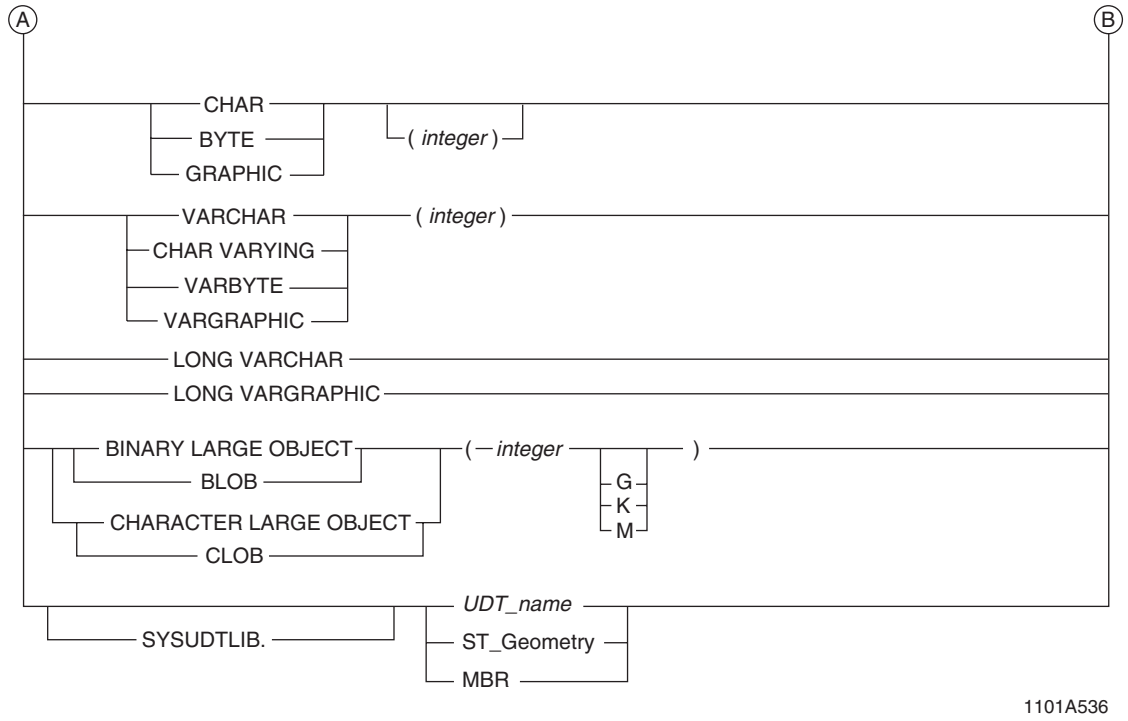
## SQL Form


REVOKE
GRANT OPTION FOR
ALL
PRIVILEGES
ALL BUT
privilege
role_privilege
profile_privilege
ON
database_name
user_name
role_name
PUBLIC
database_name.
user_name.
object_name
PROCEDURE
procedure_name
SPECIFIC FUNCTION
specific_function_name
FUNCTION
function_name
data type
parameter_name
TYPE
UDT_name
SYSUDTLIB.
TO
FROM
ALL
user_name
PUBLIC
role_name
1101V061

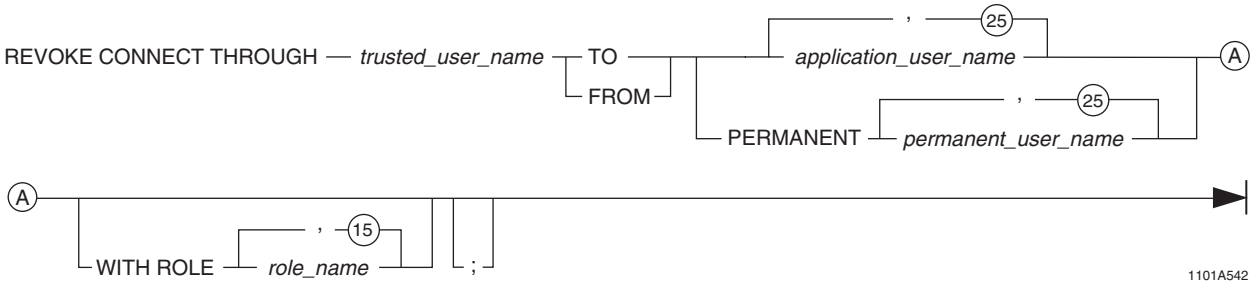
**Data Type**



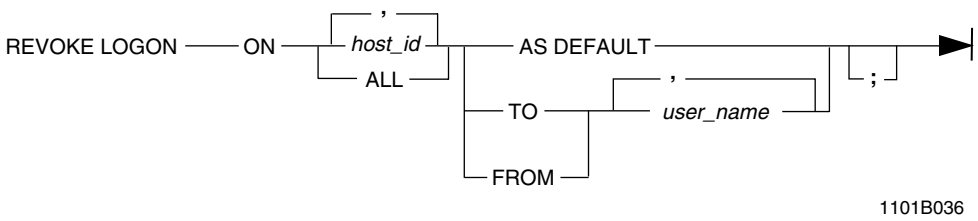

1101A535




# REVOKE CONNECT THROUGH

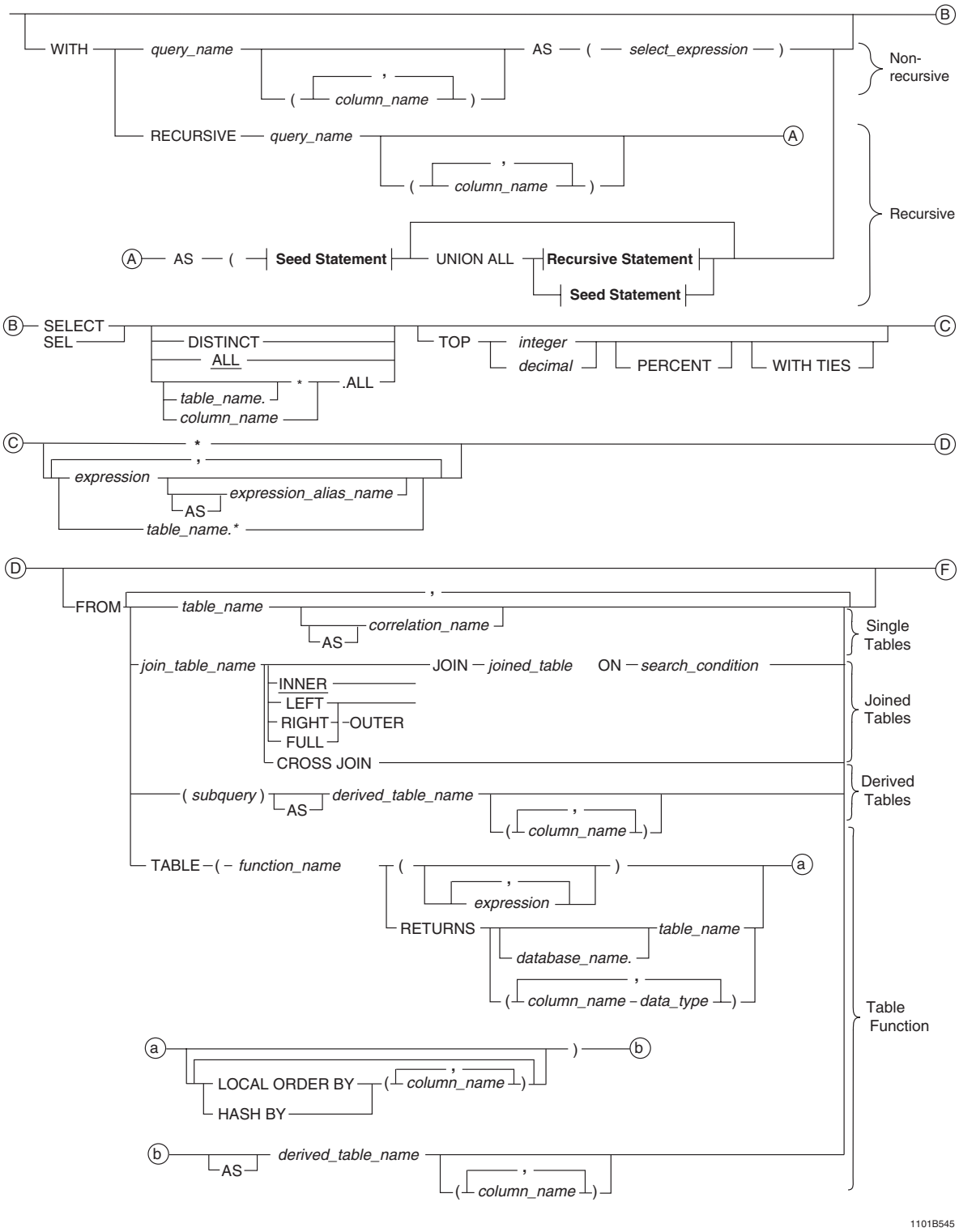


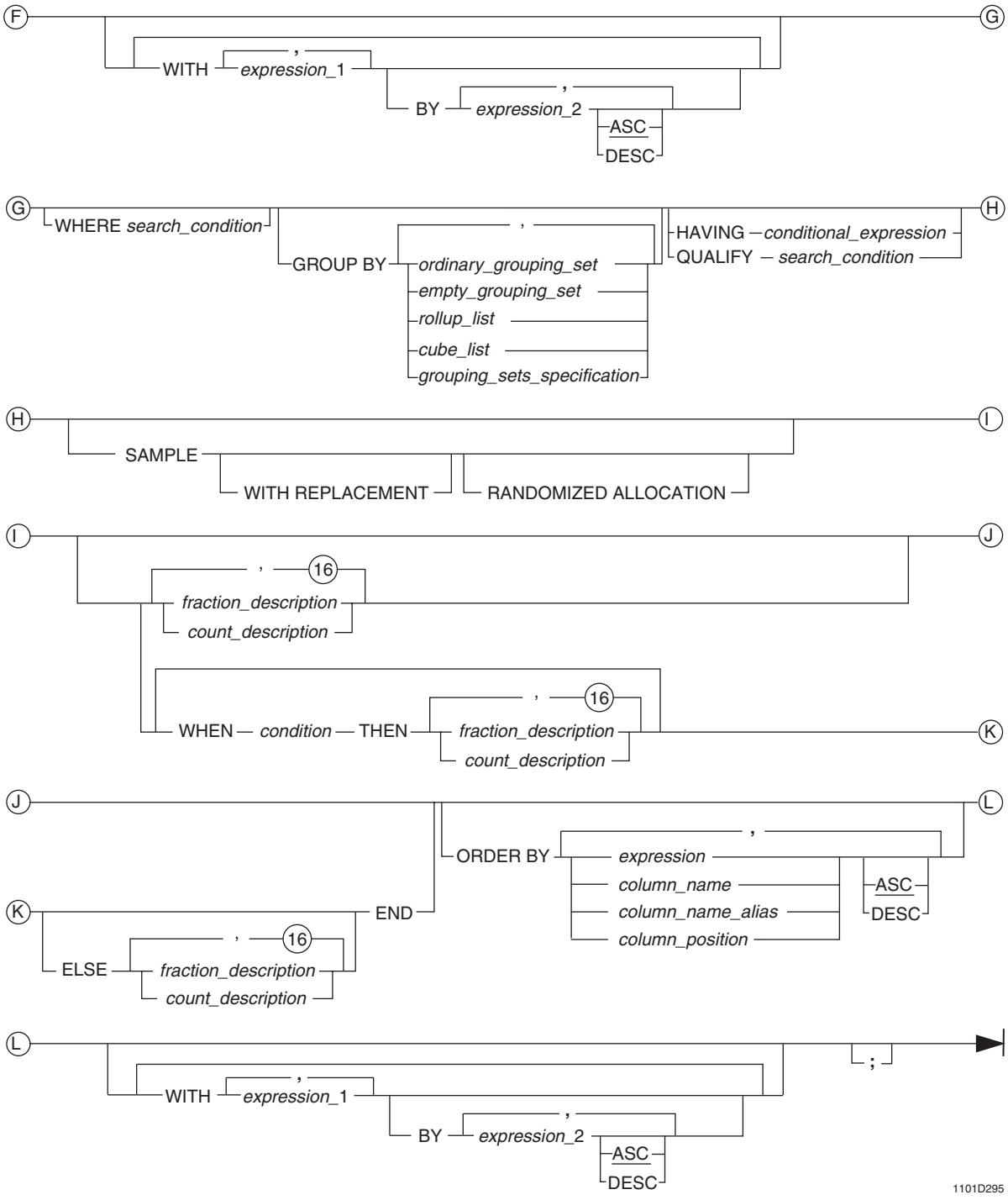
# REVOKE LOGON



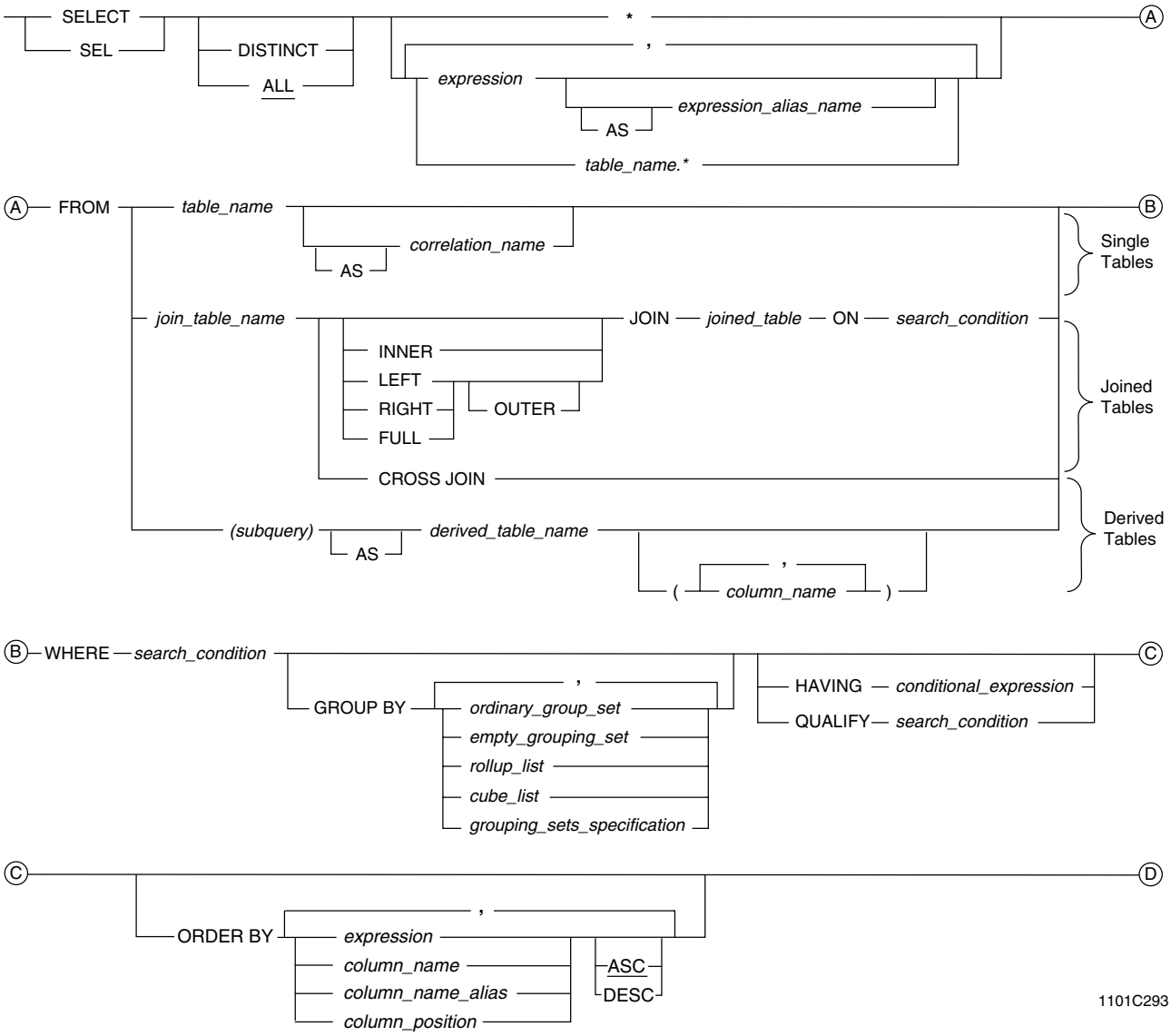
# CHAPTER 5 SQL Data Manipulation Language

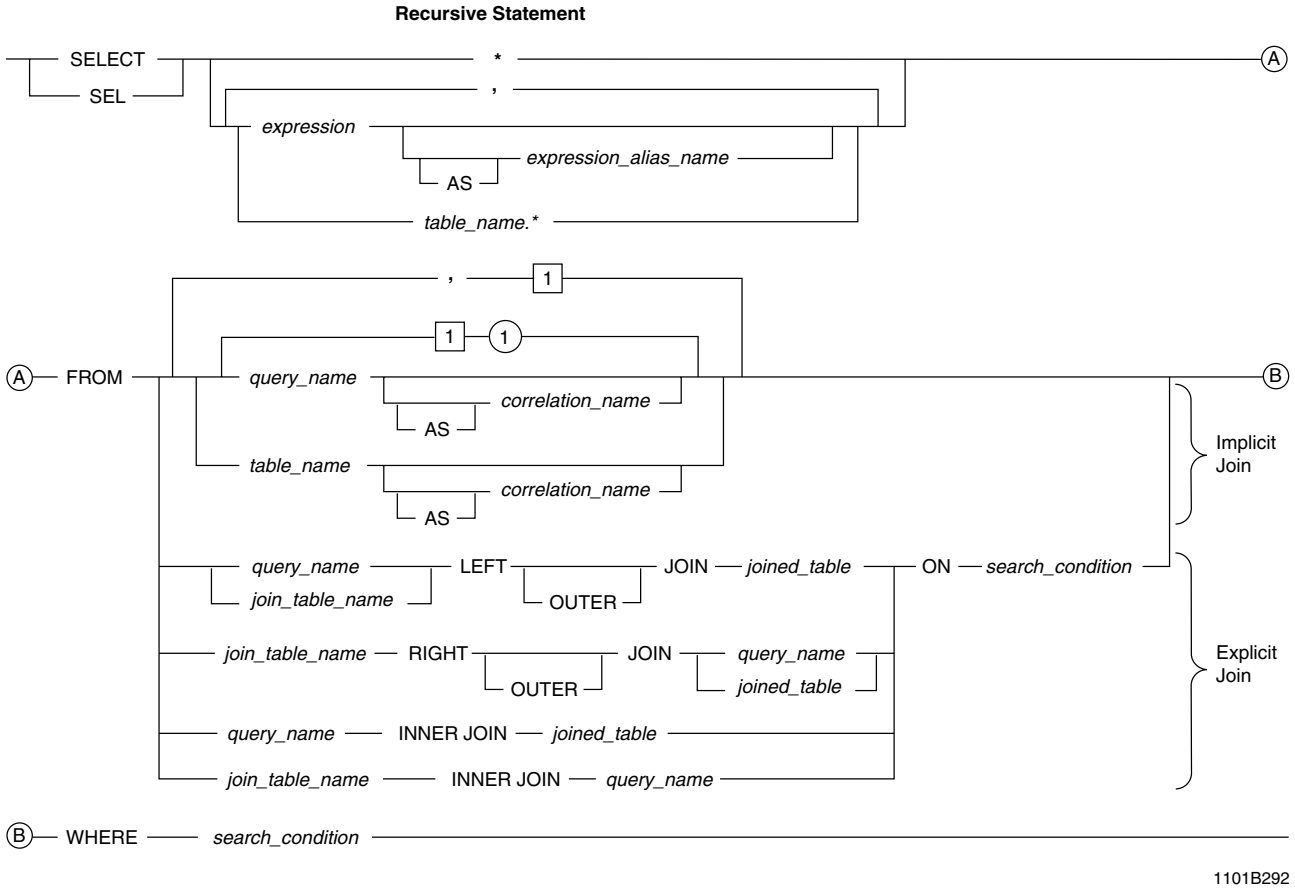
# SELECT


WITH
query_name
AS
select_expression
column_name
Non-recursive
RECURSIVE
query_name
column_name
Recursive
AS
Seed Statement
UNION ALL
Recursive Statement
Seed Statement
SELECT
SEL
DISTINCT
ALL
table_name.
column_name
.ALL
TOP
integer
decimal
PERCENT
WITH TIES
expression
expression_alias_name
AS
table_name.*
FROM
table_name
correlation_name
AS
Single Tables
join_table_name
JOIN
joined_table
ON
search_condition
INNER
LEFT
RIGHT
FULL
OUTER
CROSS JOIN
Joined Tables
subquery
AS
derived_table_name
column_name
Derived Tables
TABLE
function_name
expression
RETURNS
table_name
database_name.
column_name
data_type
LOCAL ORDER BY
HASH BY
column_name
Table Function
1101B545

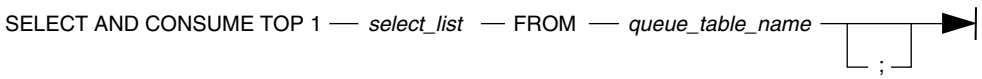

F
G
WITH
,
expression_1
BY
expression_2
ASC
DESC
G
WHERE search_condition
GROUP BY
ordinary_grouping_set
empty_grouping_set
rollup_list
cube_list
grouping_sets_specification
HAVING conditional_expression
QUALIFY search_condition
H
H
I
SAMPLE
WITH REPLACEMENT
RANDOMIZED ALLOCATION
I
J
16
fraction_description
count_description
WHEN condition THEN
K
J
L
ORDER BY
expression
column_name
column_name_alias
column_position
ASC
DESC
K
END
ELSE
L
;
1101D295

**Seed Statement**

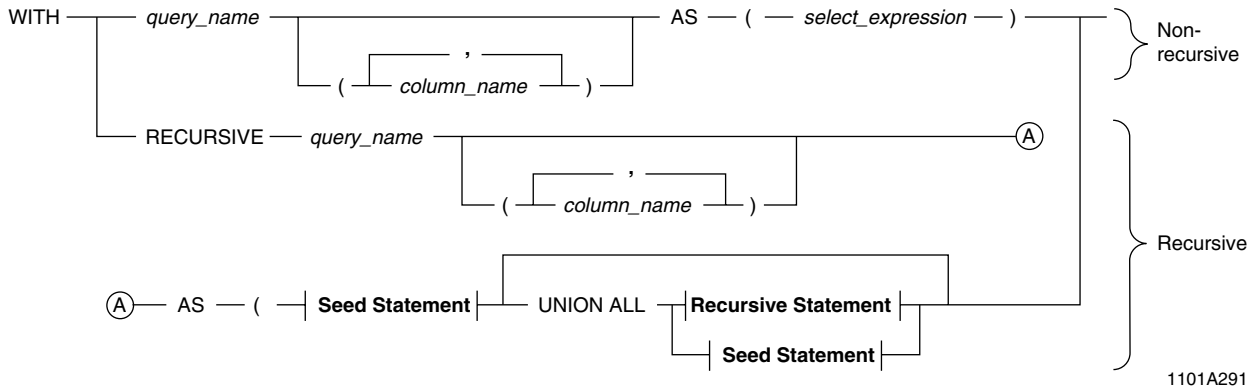




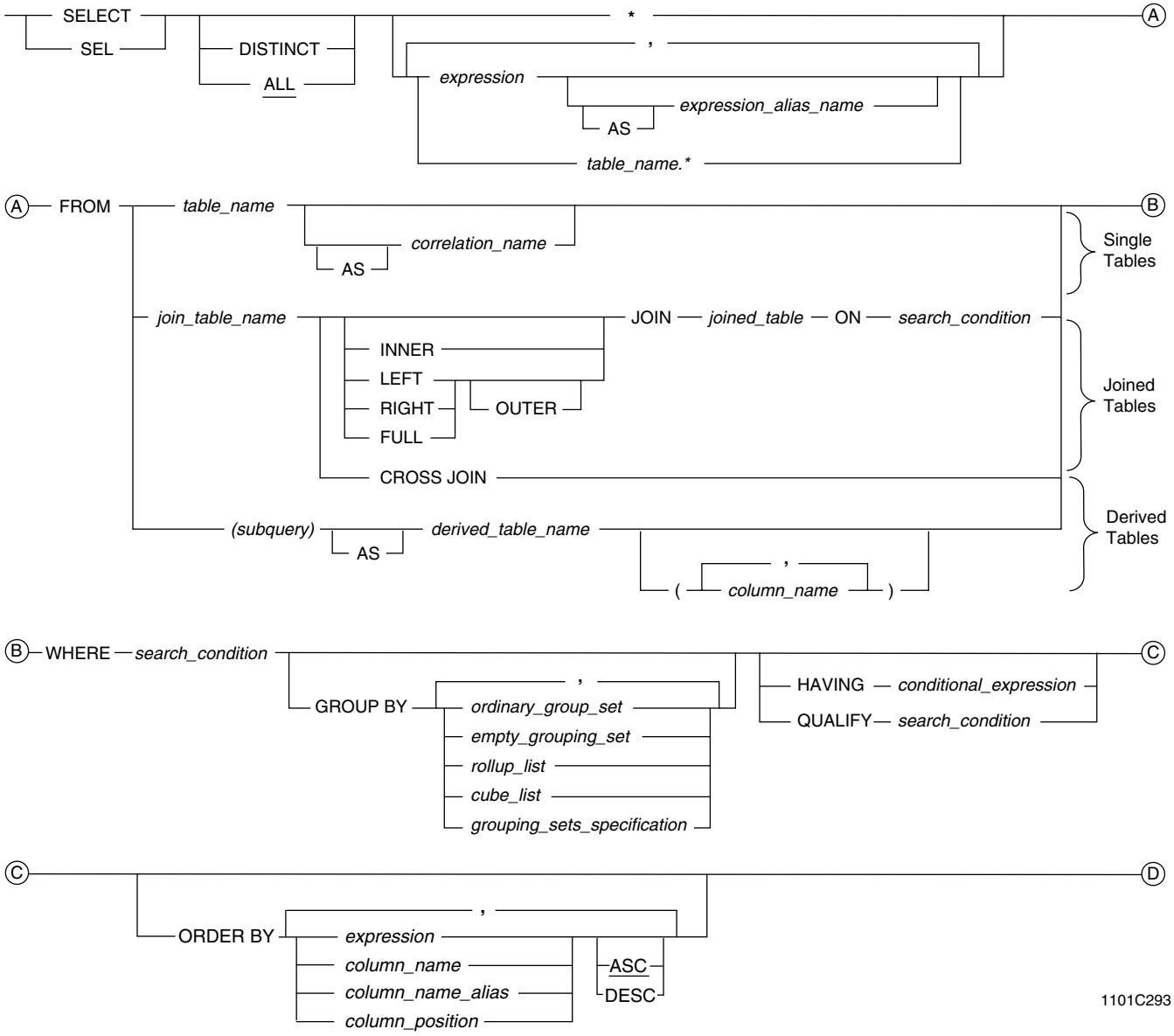

# SELECT AND CONSUME

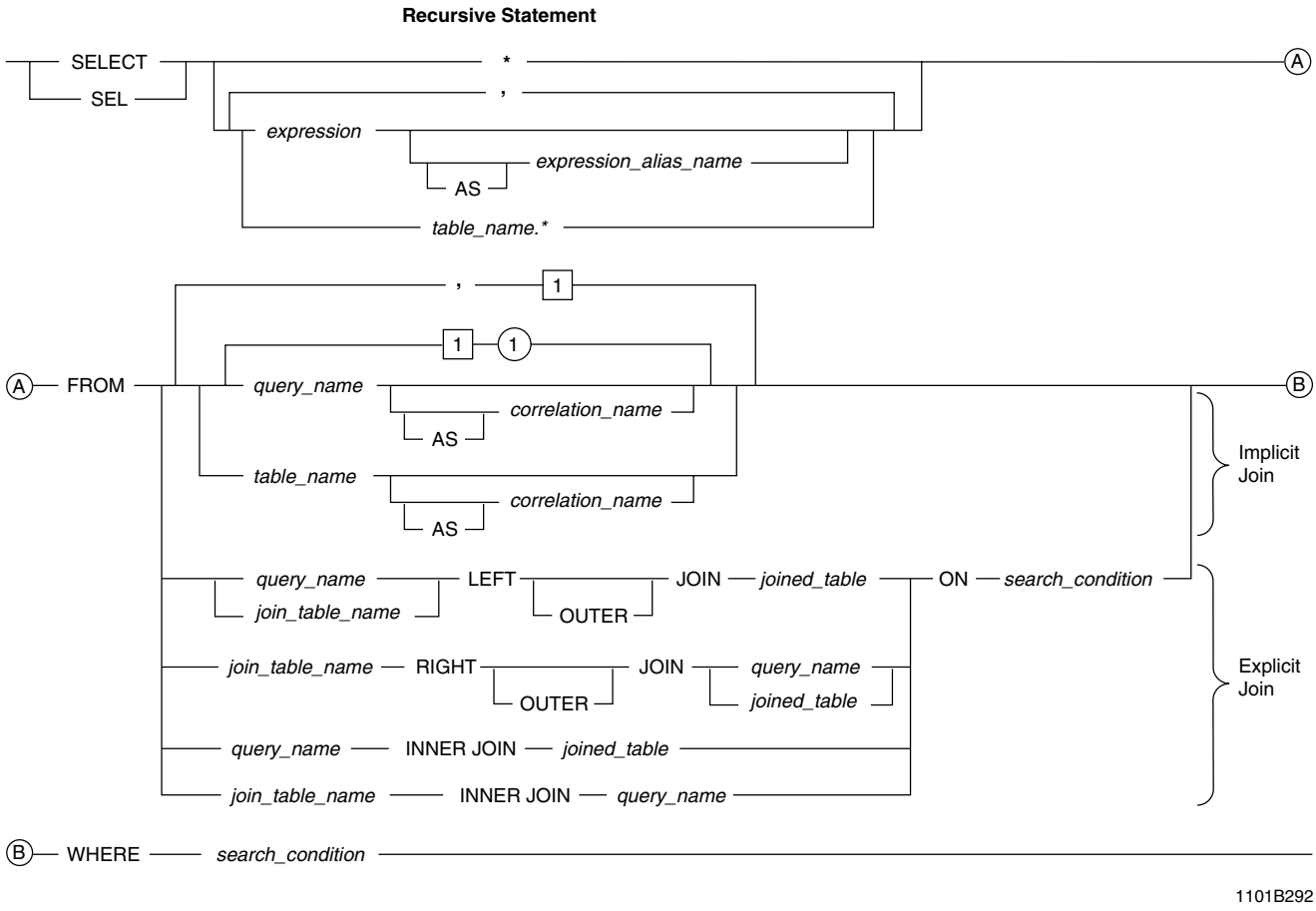


# WITH [RECURSIVE] Request Modifier

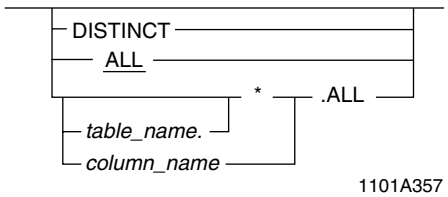


**Seed Statement**

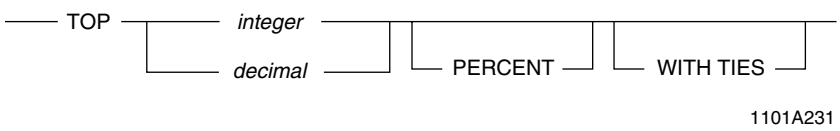




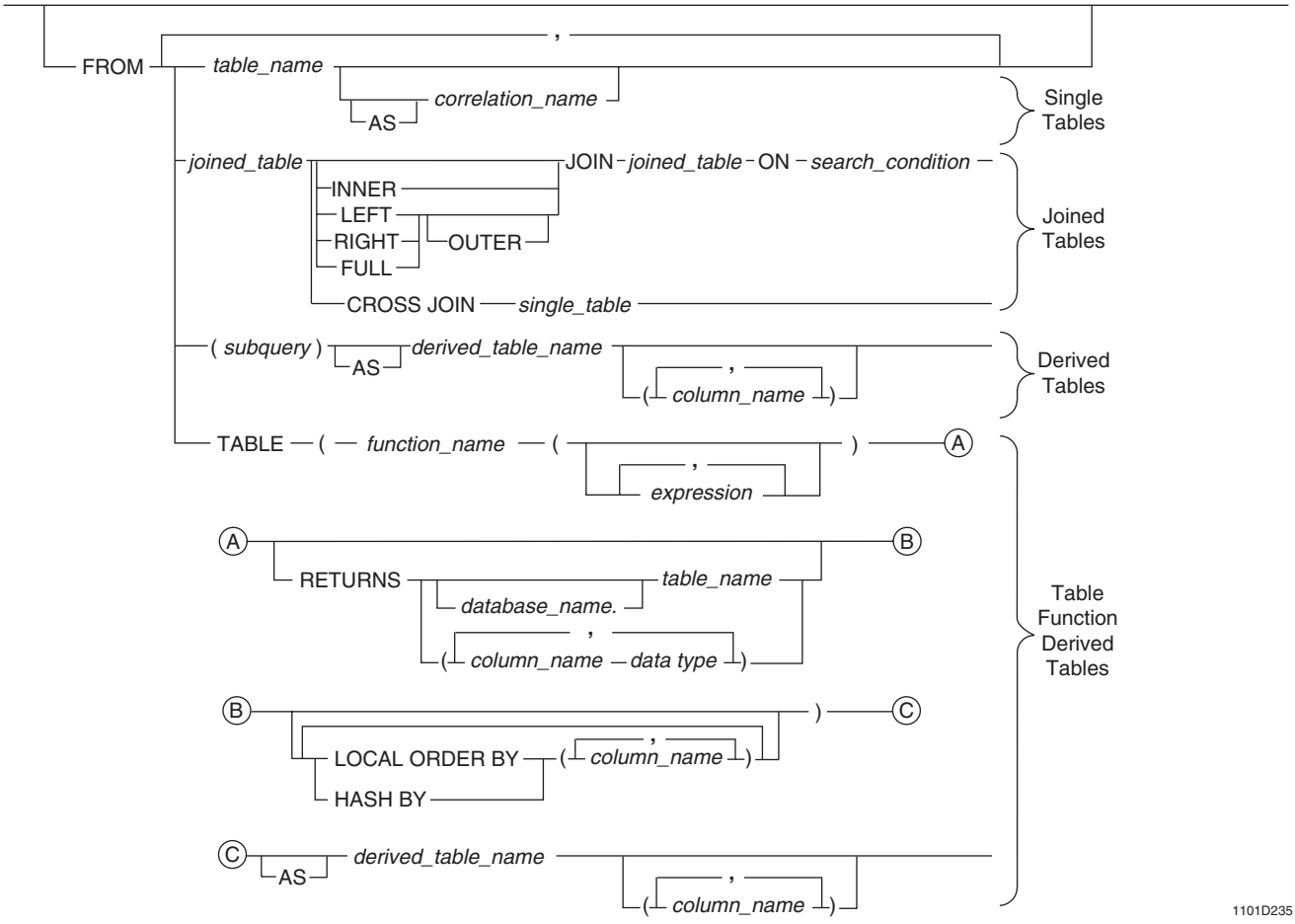

# DISTINCT, ALL, and .ALL Options



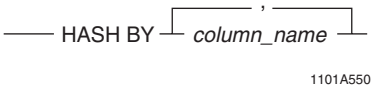
# TOP *n* Operator



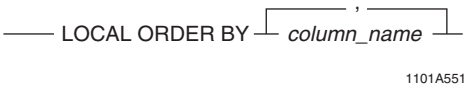

# FROM Clause



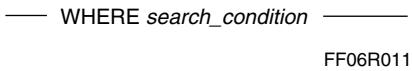
# HASH BY Clause




# LOCAL ORDER BY Clause

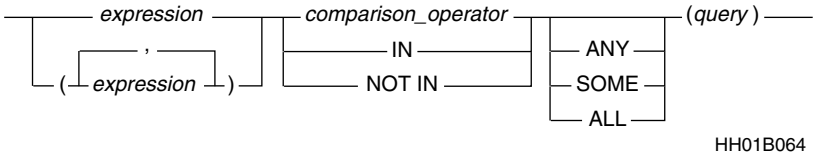



# WHERE Clause

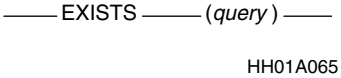



# Subqueries in Search Conditions

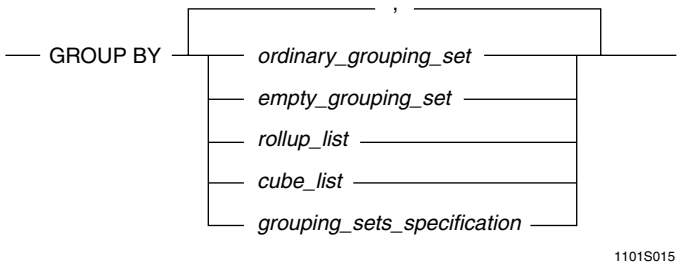
## Syntax 1



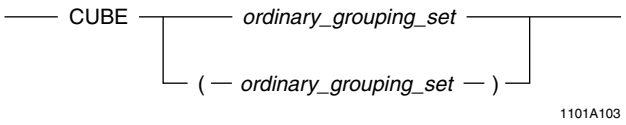

## Syntax 2: Logical Expressions



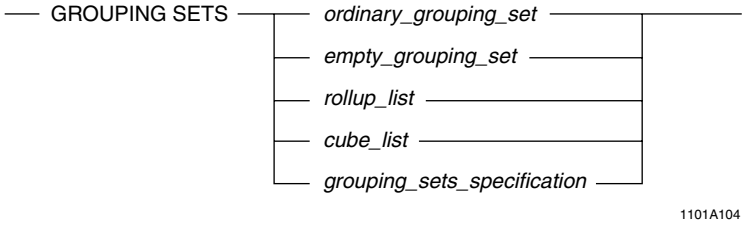
# GROUP BY Clause



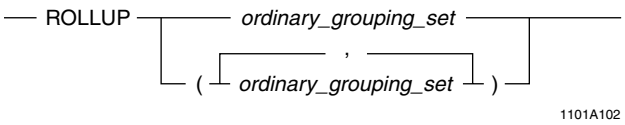

## CUBE Option



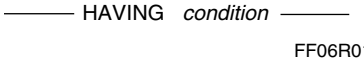

## GROUPING SETS Option



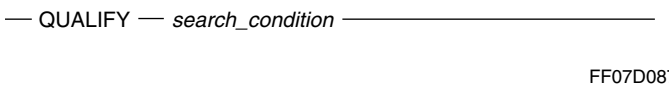

## ROLLUP Option




# HAVING Clause

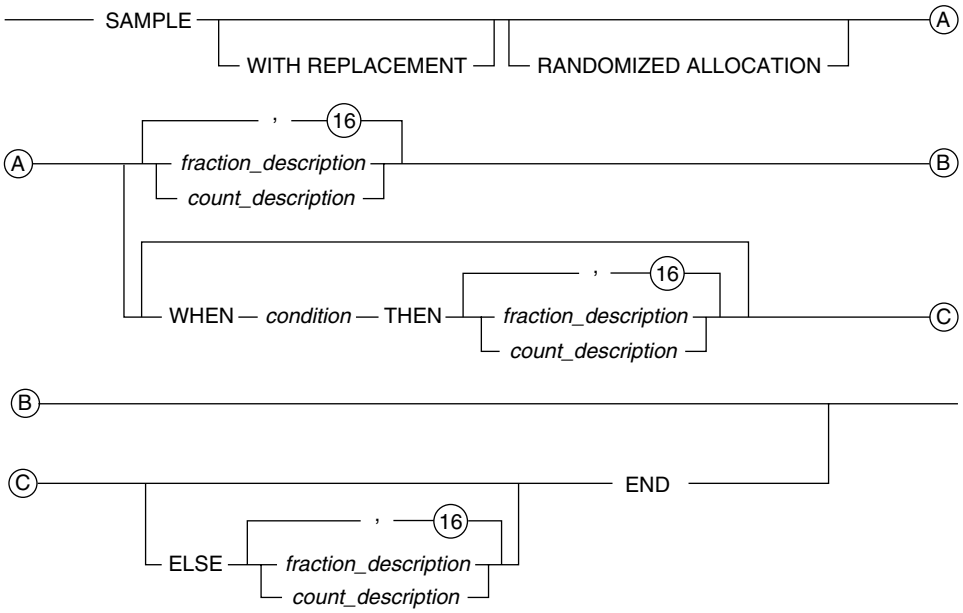


# QUALIFY Clause




FF07D087

# SAMPLE Clause

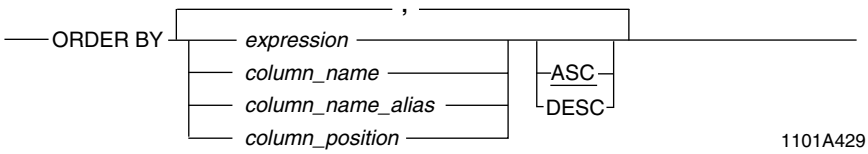



1101B065

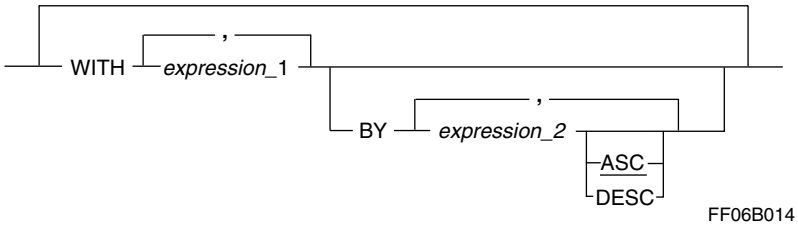
# SAMPLEID Expression


FF07D180

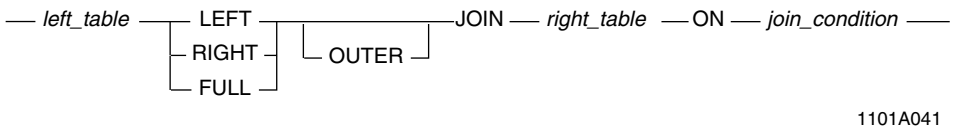
# ORDER BY Clause



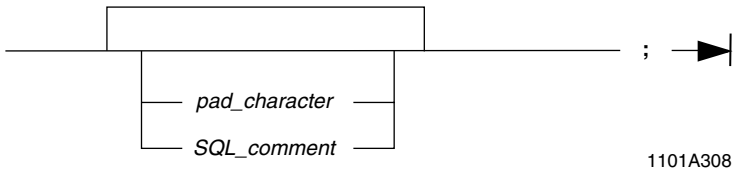

# WITH Clause



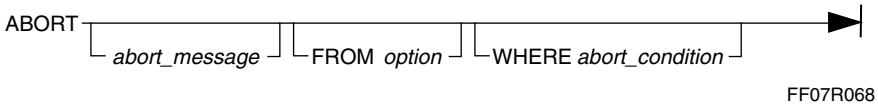

# Outer Join




# Null

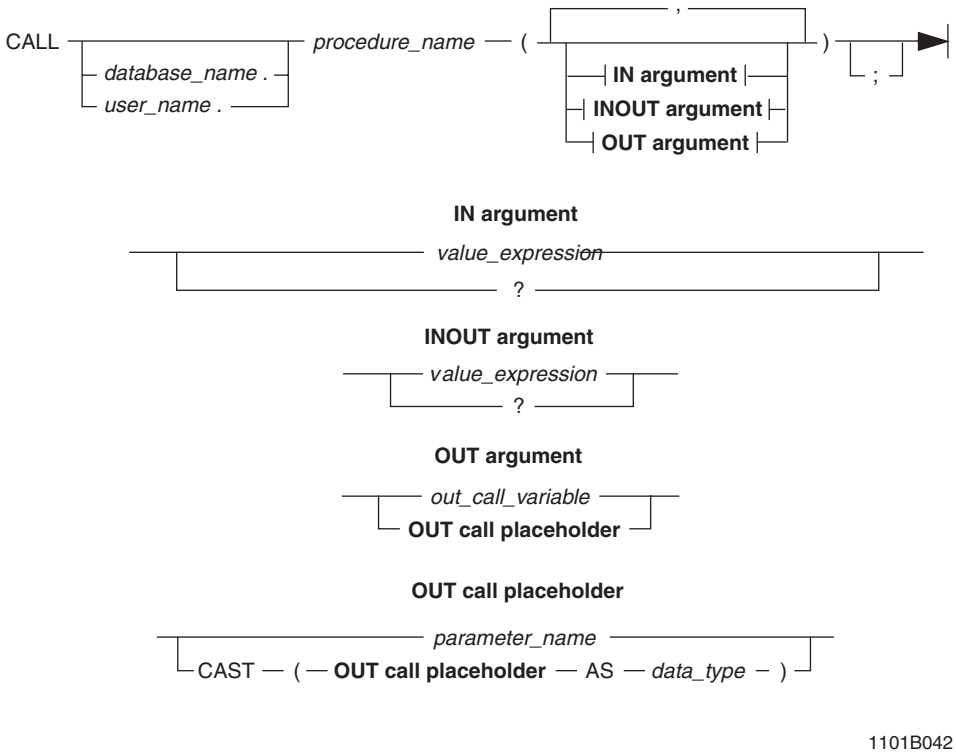


# ABORT



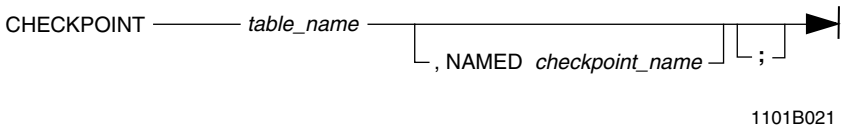

# BEGIN TRANSACTION


# CALL

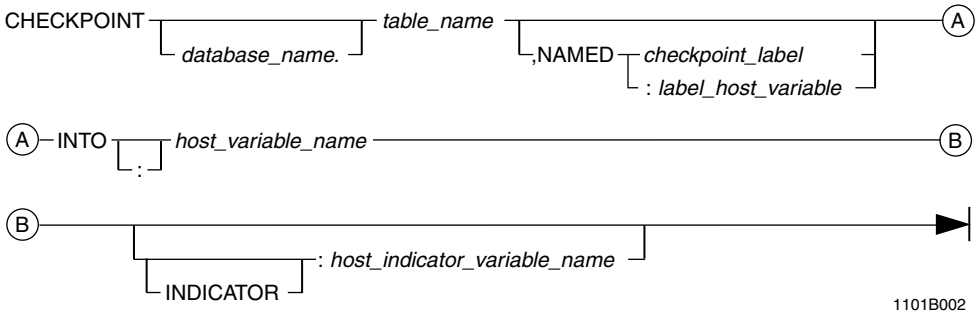


# CHECKPOINT

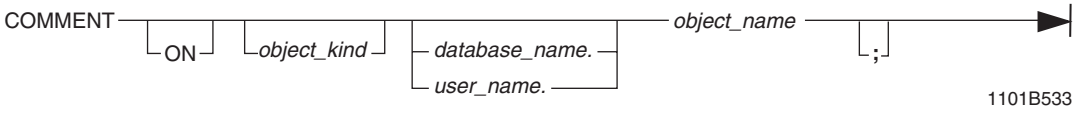
## Interactive Syntax



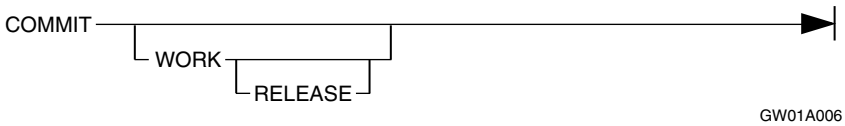

## Embedded SQL and Stored Procedure Syntax




# COMMENT (Comment-Retrieving Form)

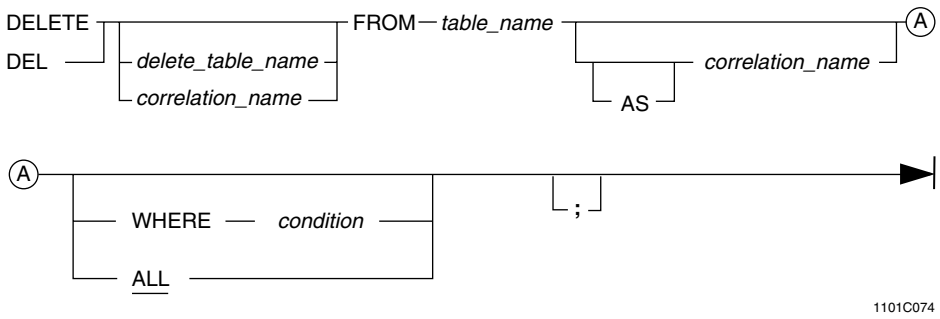



# COMMIT

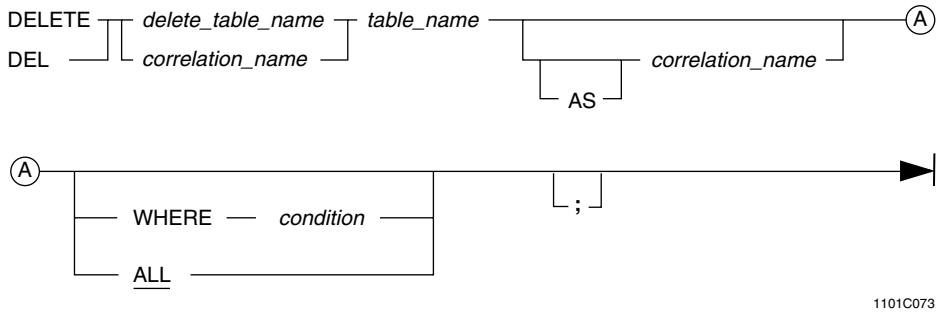


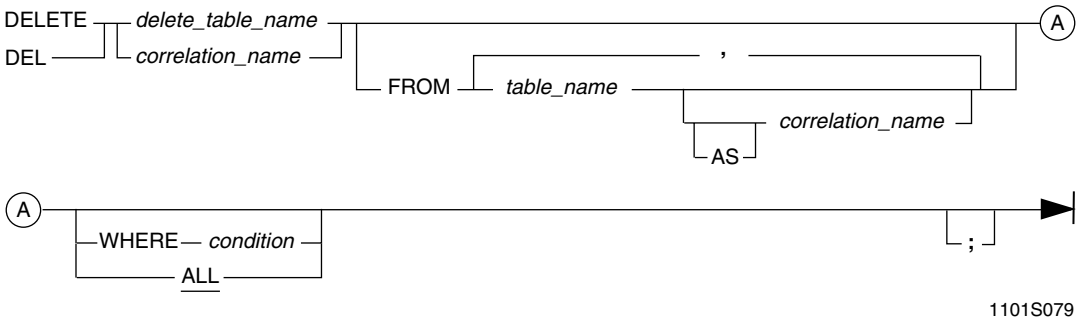
# DELETE

## Basic/Searched Form




## Join Condition Form



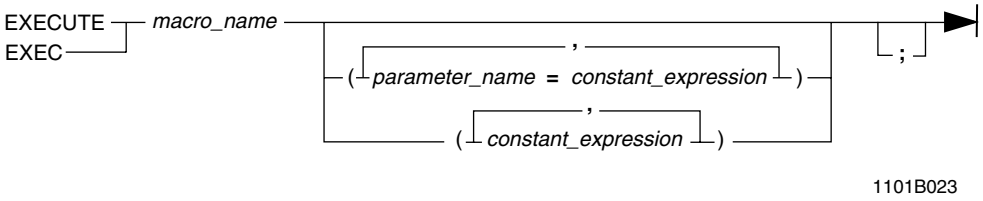



# ECHO

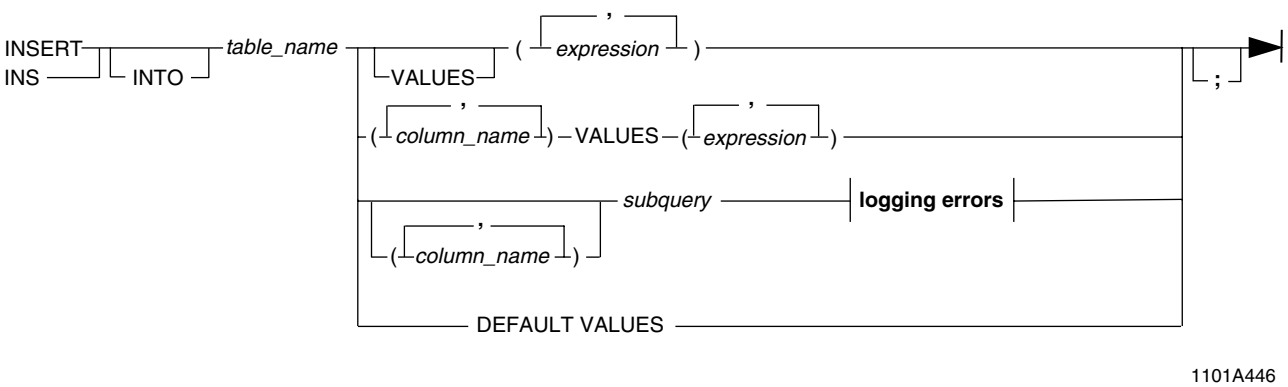

# END TRANSACTION


# EXECUTE

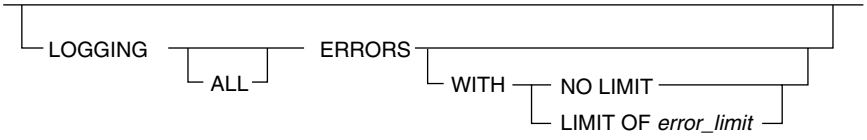
## Macro Form




# INSERT/INSERT . . . SELECT

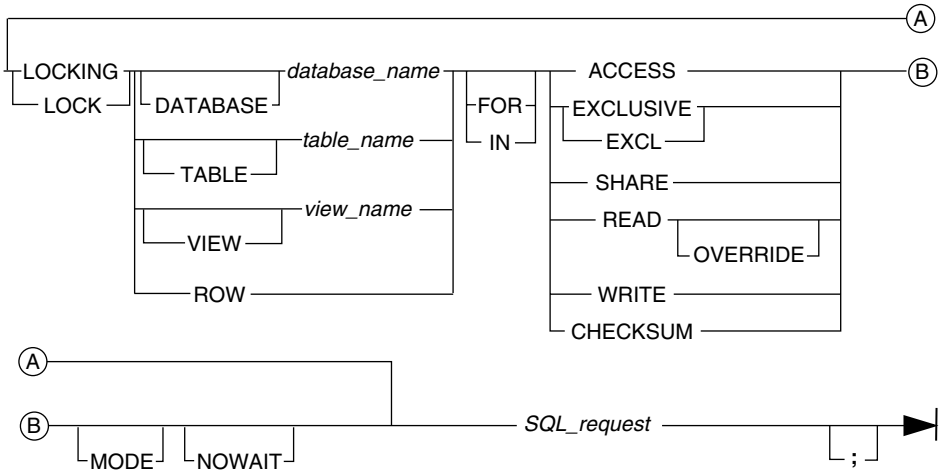


**logging errors**



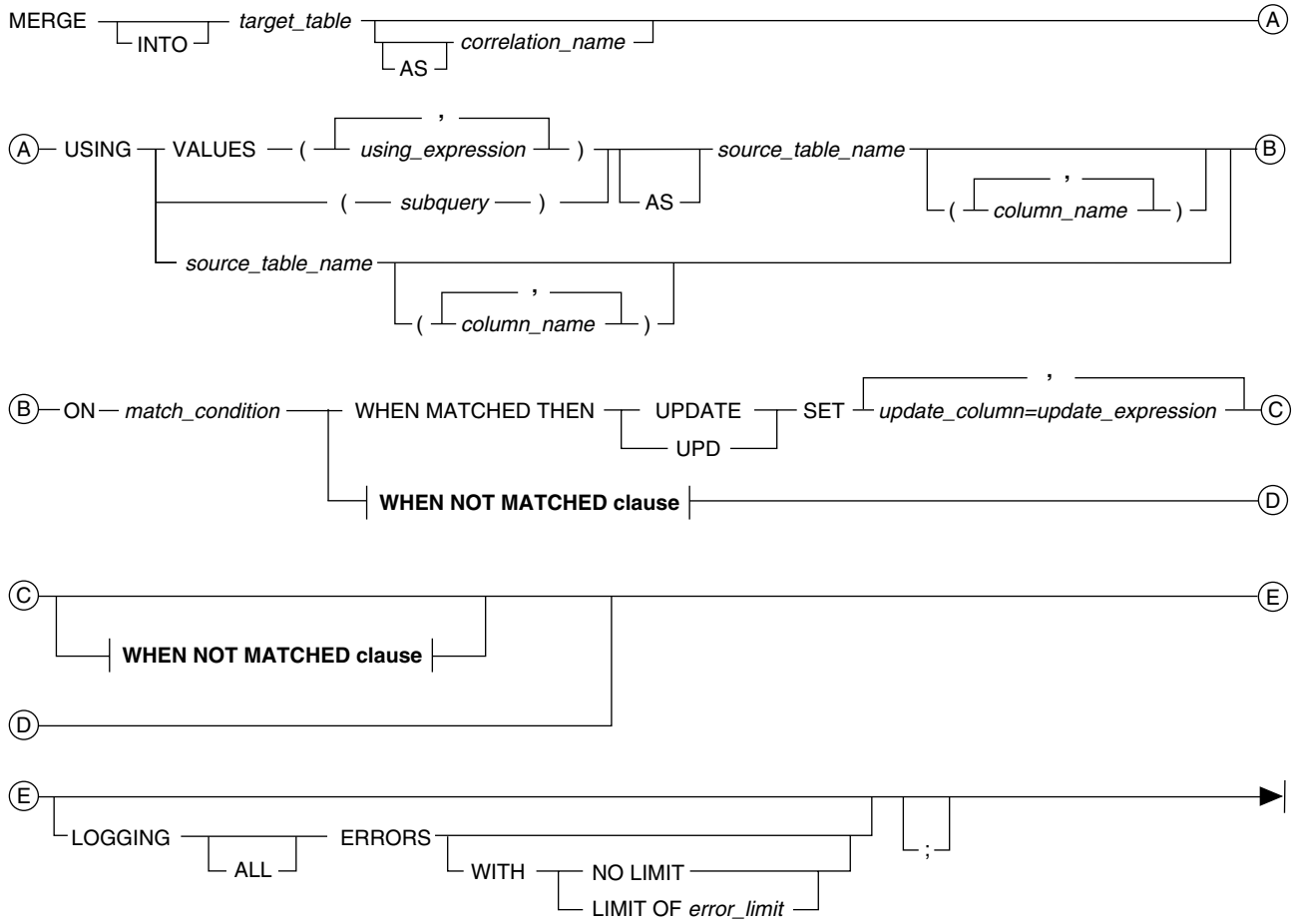

1101A447

# LOCKING Request Modifier

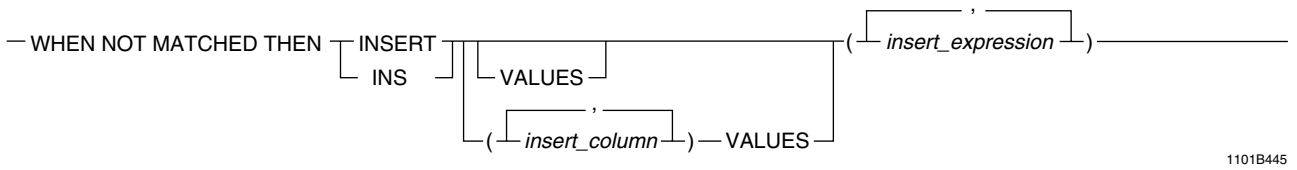



1101U031

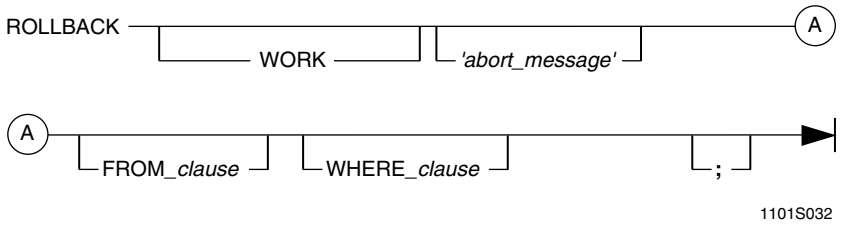
# MERGE




**WHEN NOT MATCHED clause**

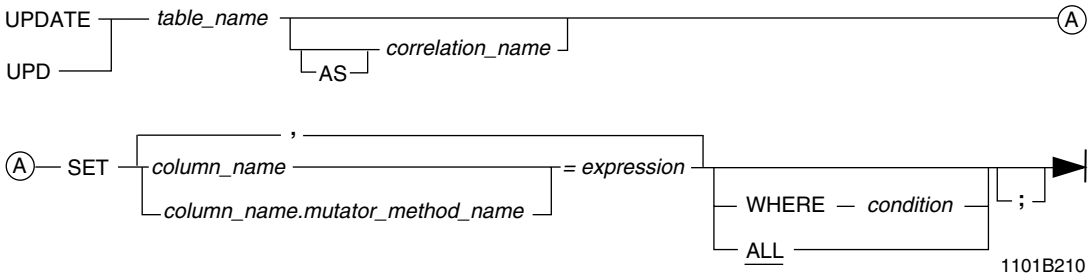


# ROLLBACK

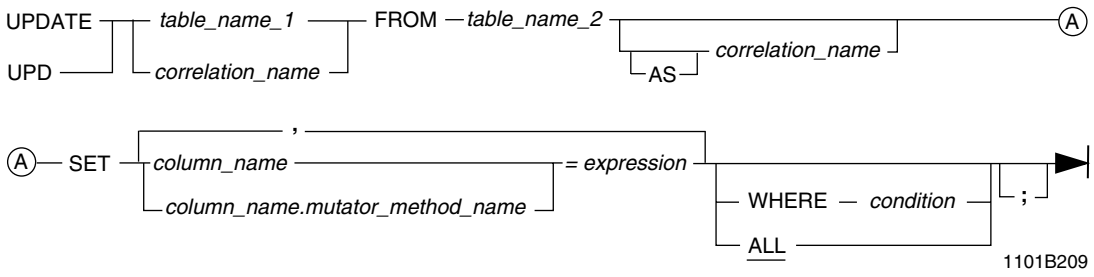



# UPDATE

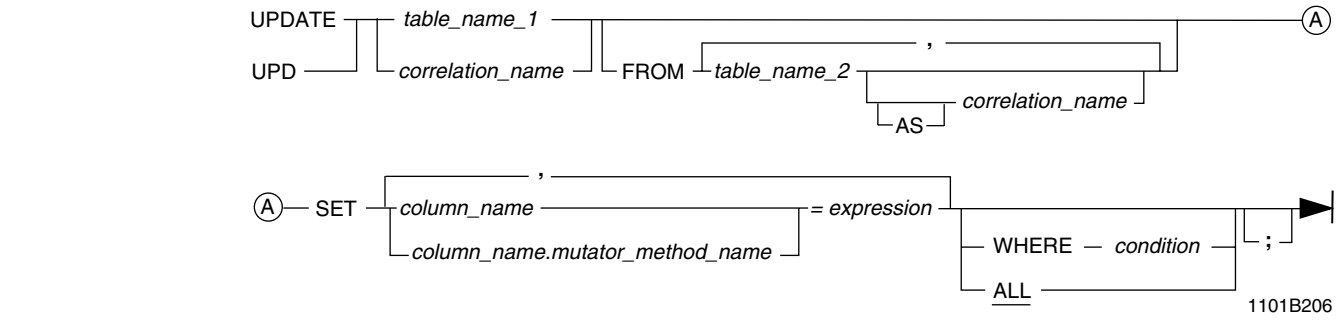
## Basic Form, No FROM Clause Syntax



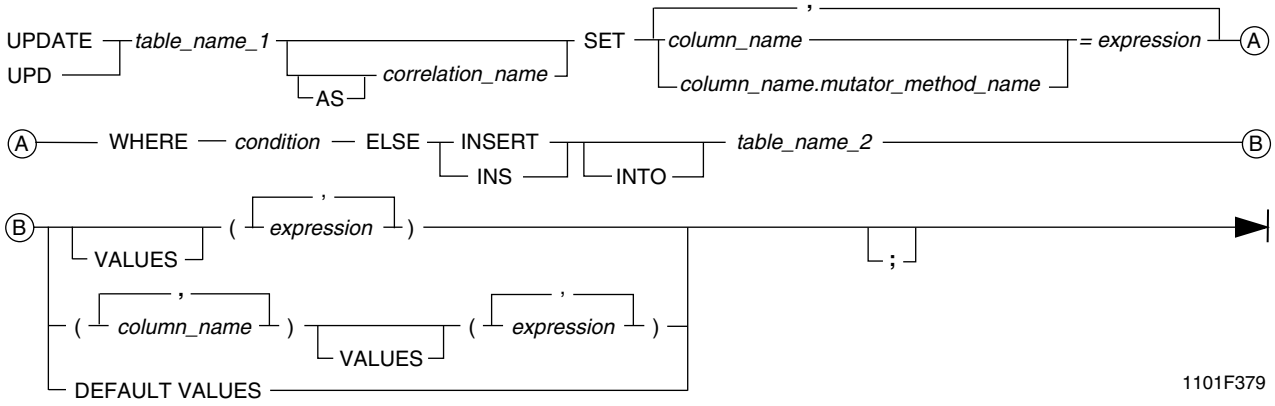

## Basic Form, FROM Clause Syntax



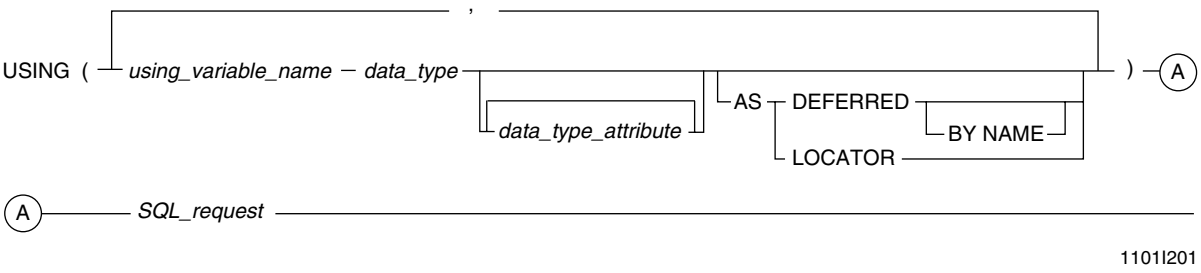
## Joined Tables Syntax



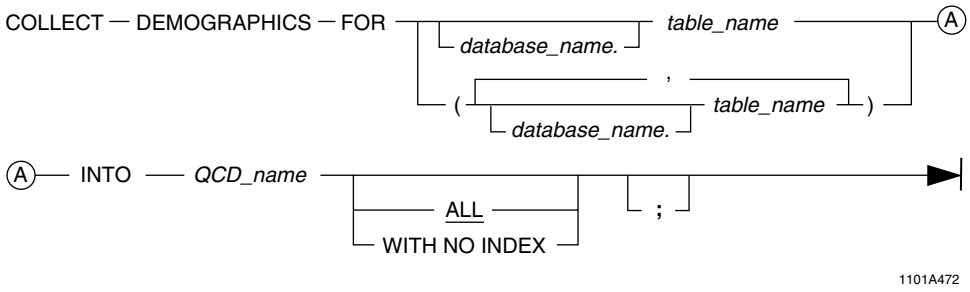

## Upsert Form



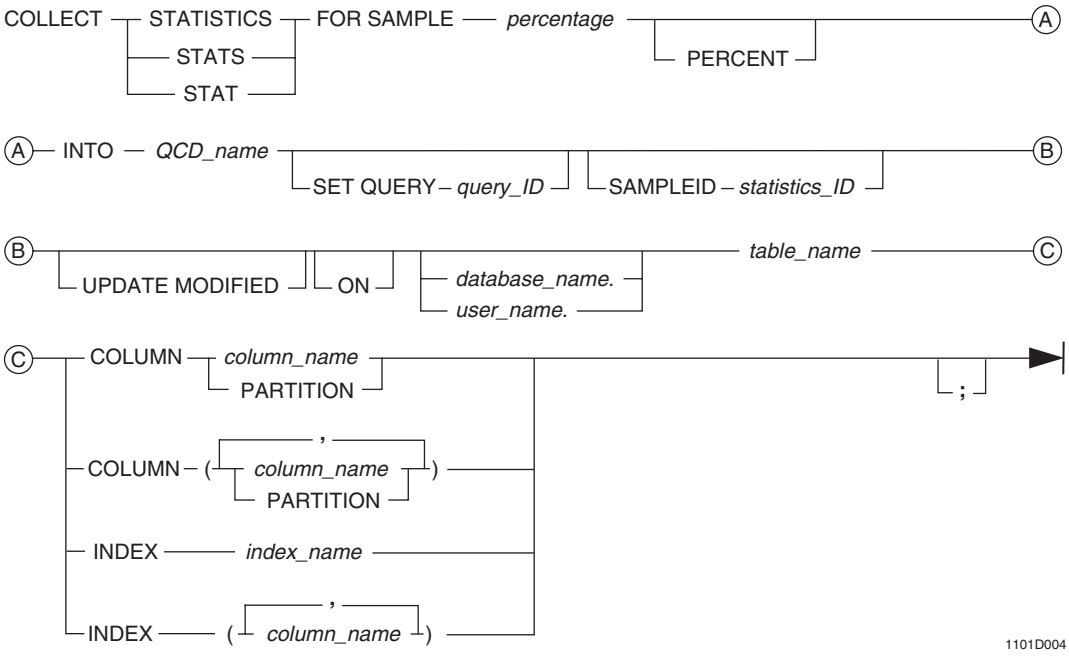

# USING Request Modifier



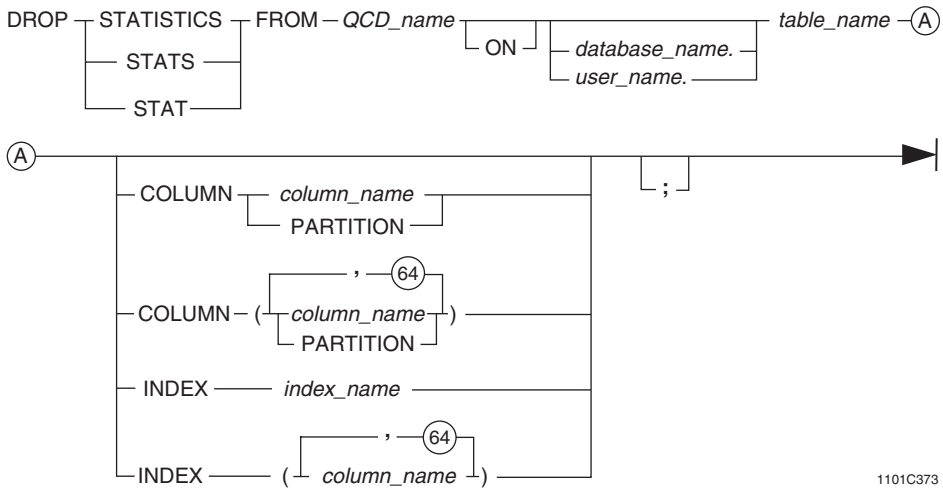
# COLLECT DEMOGRAPHICS



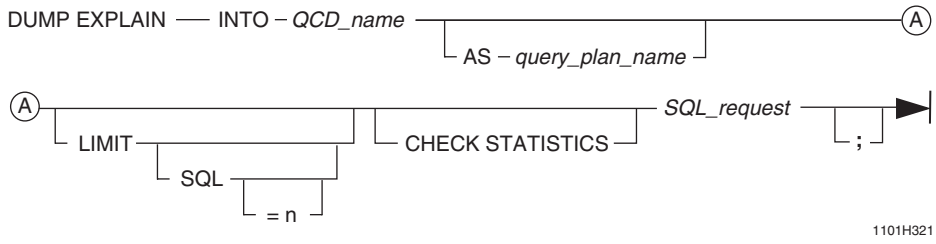

# COLLECT STATISTICS (QCD Form)



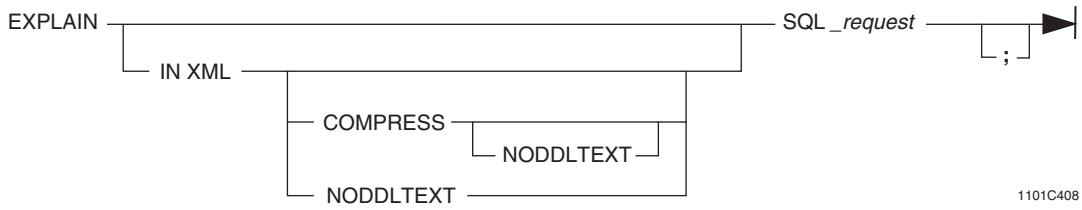
# DROP STATISTICS (QCD Form)



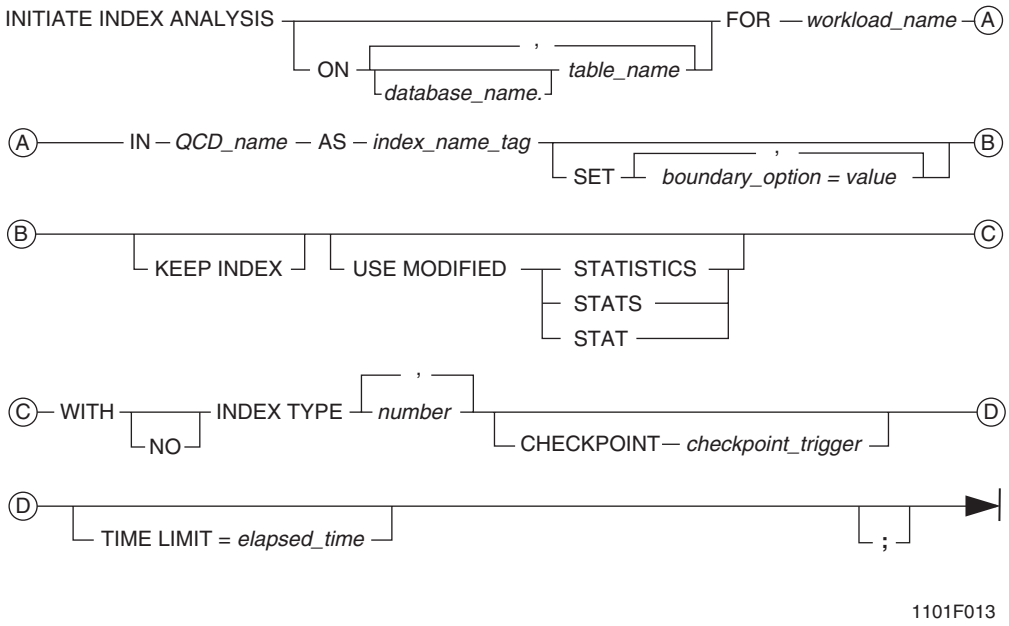

# DUMP EXPLAIN




# EXPLAIN Request Modifier

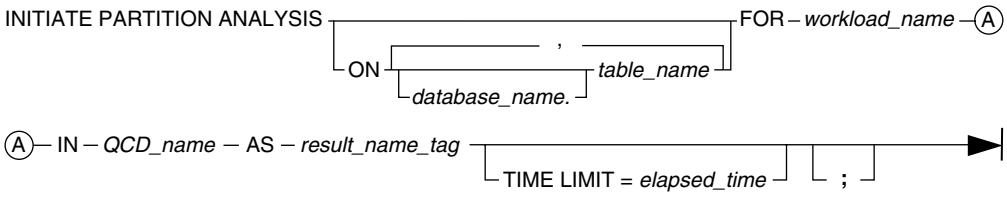


# INITIATE INDEX ANALYSIS



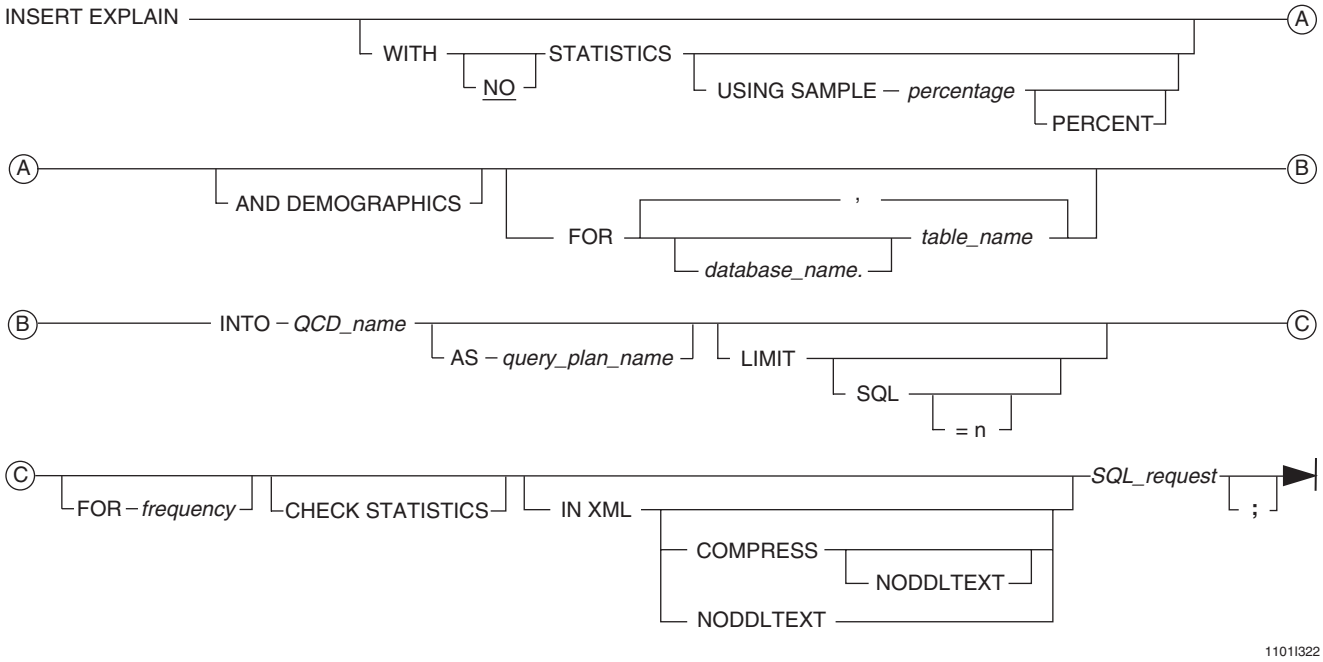

1101F013

# INITIATE PARTITION ANALYSIS

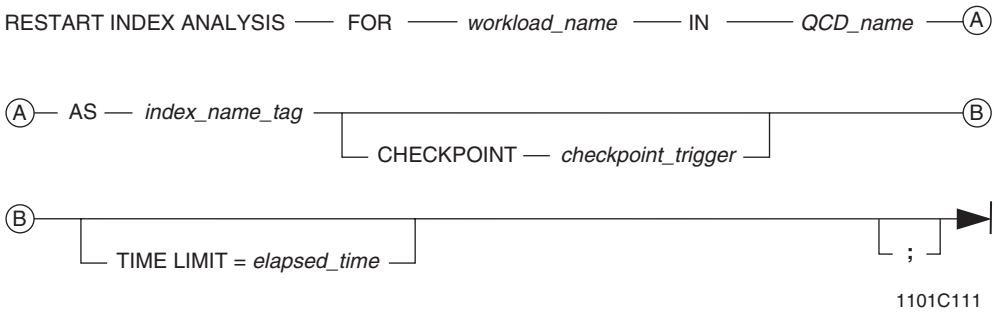



1101B443

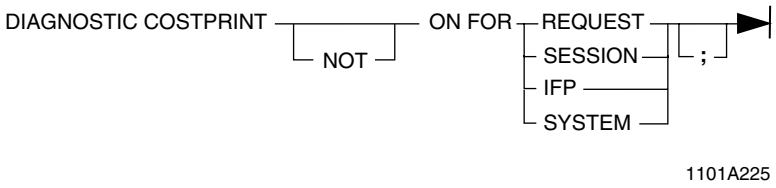
# INSERT EXPLAIN



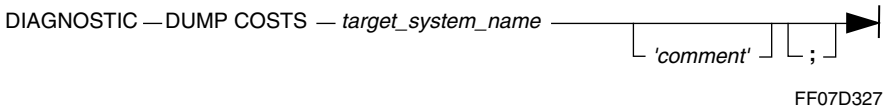

# RESTART INDEX ANALYSIS



# DIAGNOSTIC COSTPRINT



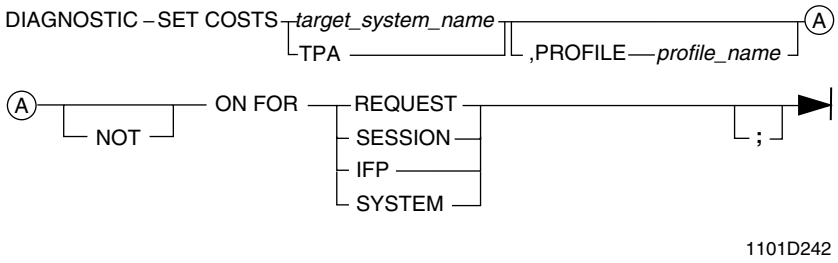

# DIAGNOSTIC DUMP COSTS



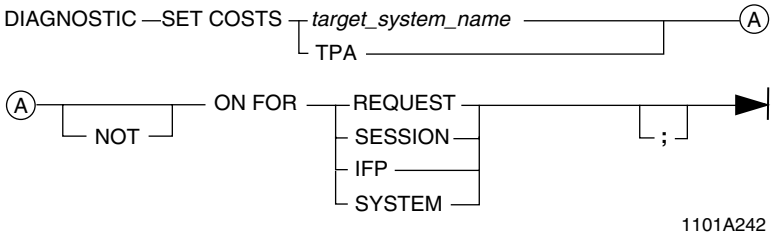

# DIAGNOSTIC HELP COSTS

# DIAGNOSTIC SET COSTS

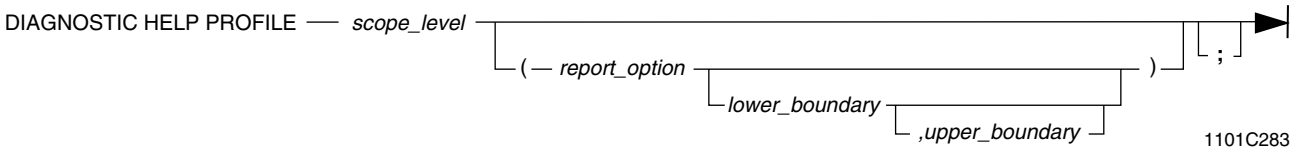
## Syntax (Full)



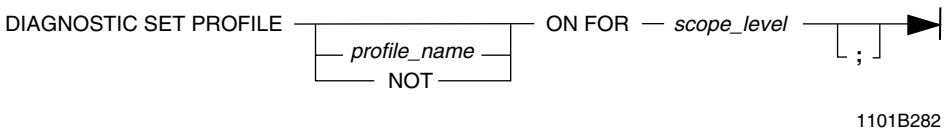

## Syntax (Restricted)



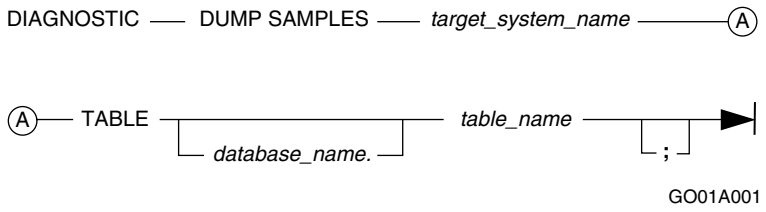

# DIAGNOSTIC HELP PROFILE




# DIAGNOSTIC SET PROFILE



# DIAGNOSTIC DUMP SAMPLES



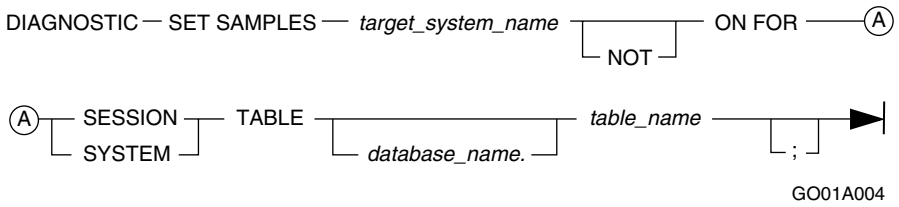

GO01A001

# DIAGNOSTIC HELP SAMPLES


GO01A002

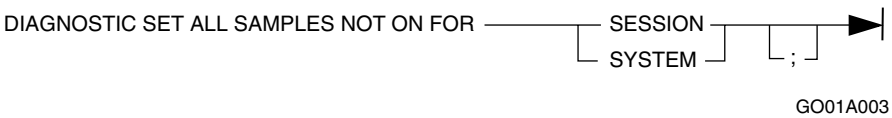
# DIAGNOSTIC SET SAMPLES

## General Syntax



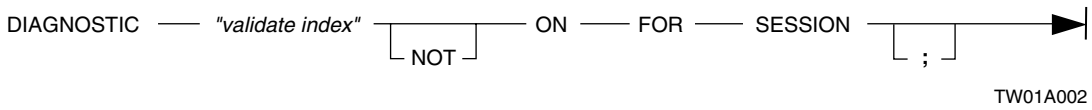

GO01A004

## Disable All Samples Syntax




GO01A003

# DIAGNOSTIC “Validate Index”




TW01A002

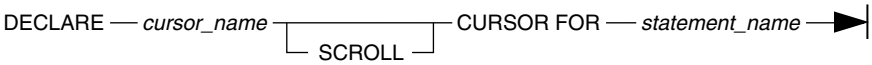
# CHAPTER 6 SQL Cursor Control

## CLOSE


GW01A003 3

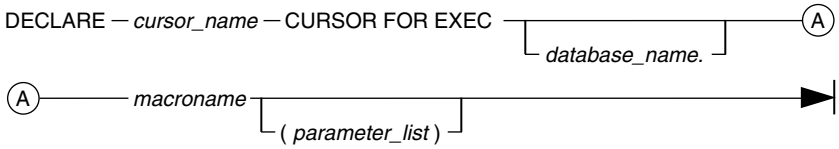
## DECLARE CURSOR

### Dynamic SQL Form




1101A307

### Macro Form

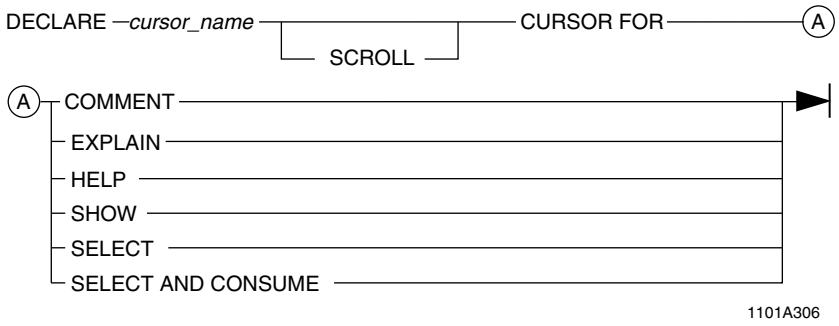



1101B011

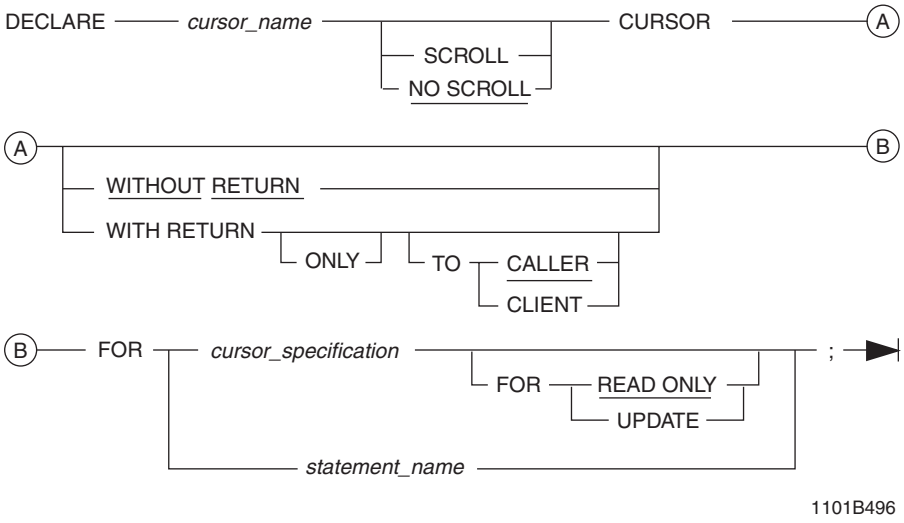
### Request Form


1101B301

## Selection Form

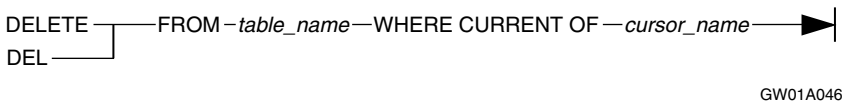



## Stored Procedures Form



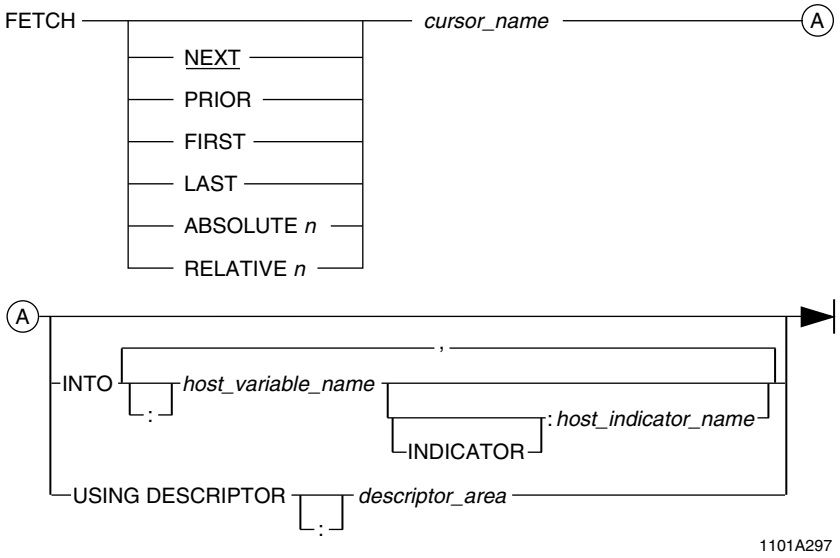

# DELETE

## Positioned Form

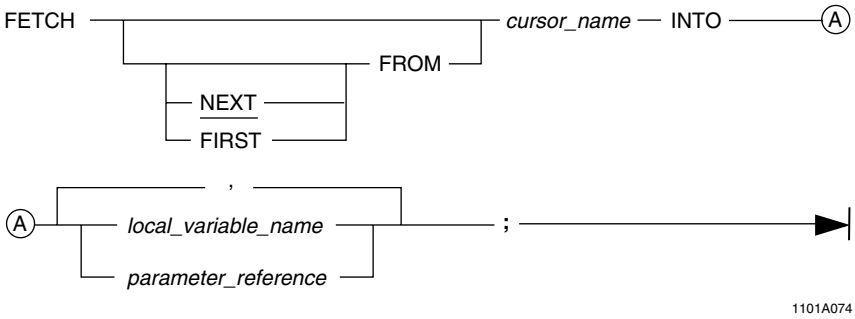


# FETCH

## Embedded SQL Form

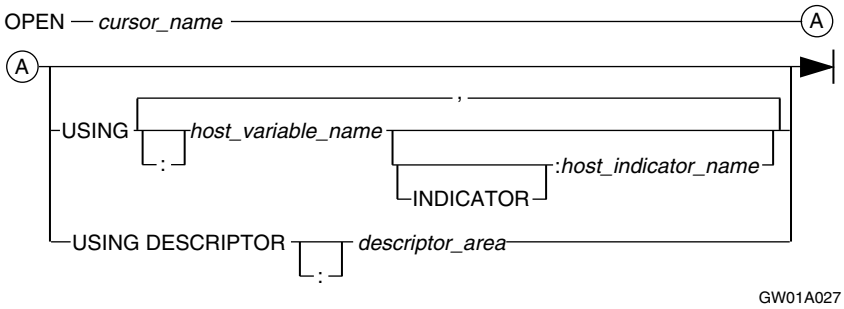



## Stored Procedures Form

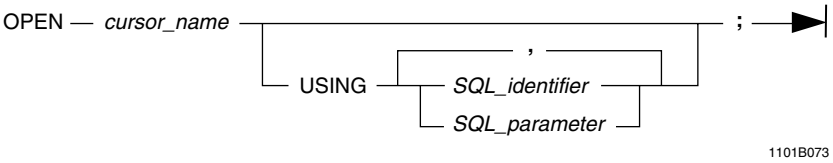


# OPEN

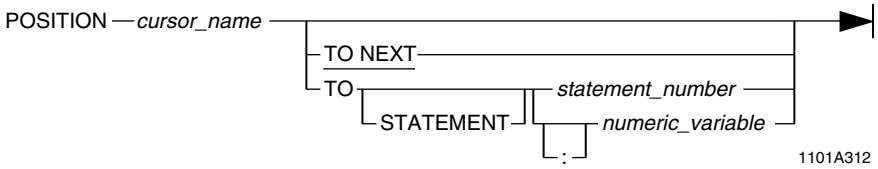
## Embedded SQL Form



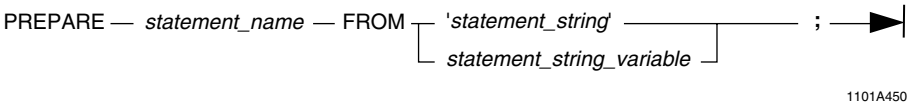

## Stored Procedures Form



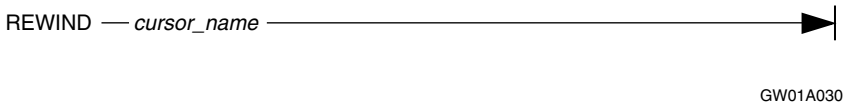

# POSITION




# PREPARE

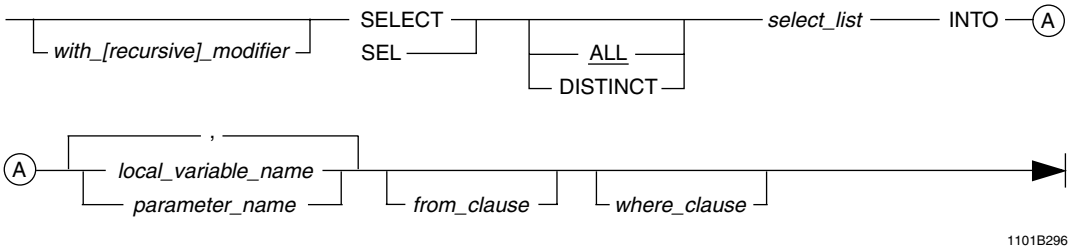


# REWIND

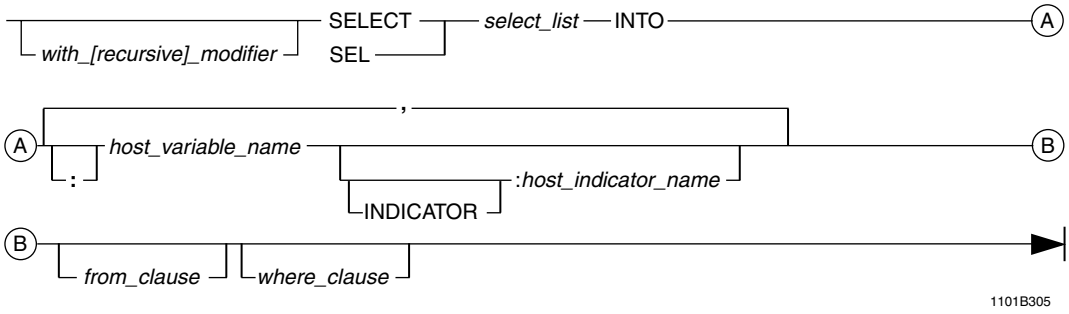



# SELECT ... INTO

## Stored Procedures Only

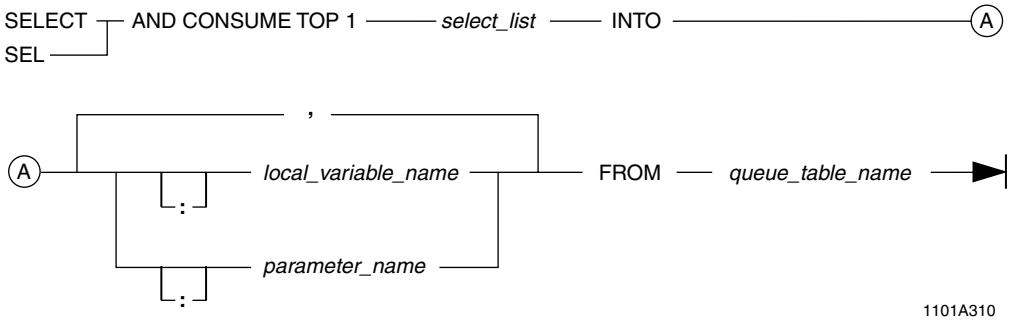



## Embedded SQL Only

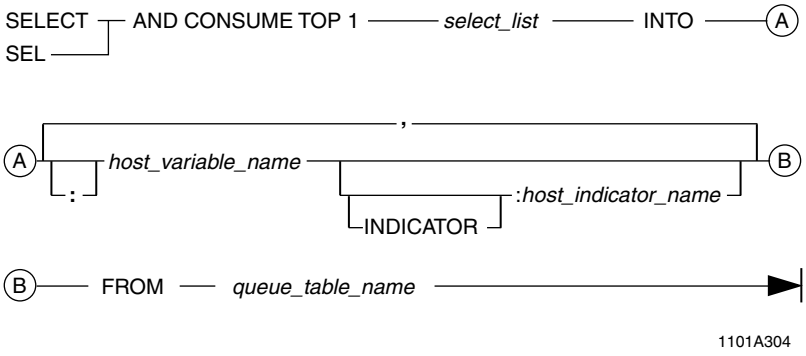


# SELECT AND CONSUME ... INTO

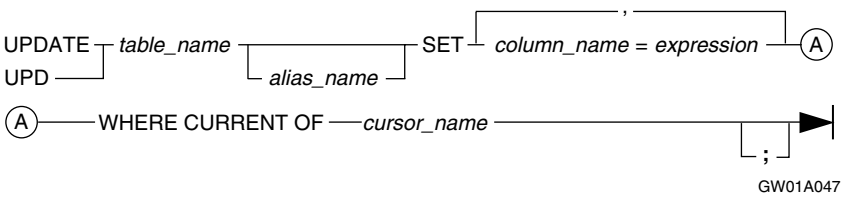
## Stored Procedures Only




## Embedded SQL Only

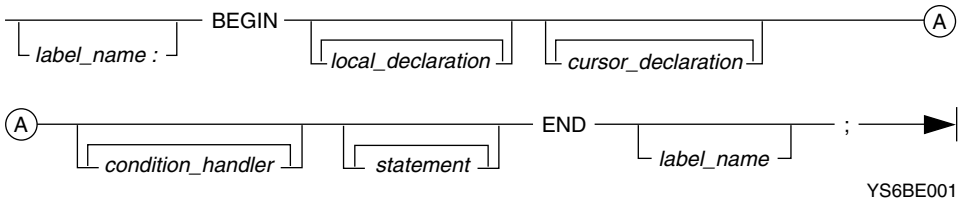



# UPDATE (Positioned Form)



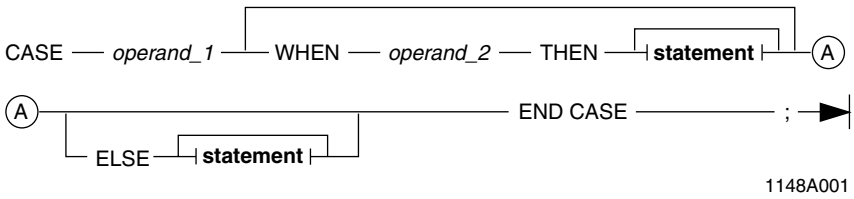
# CHAPTER 7 SQL Stored Procedures: Control Statements and Condition Handling

## BEGIN - END Statement

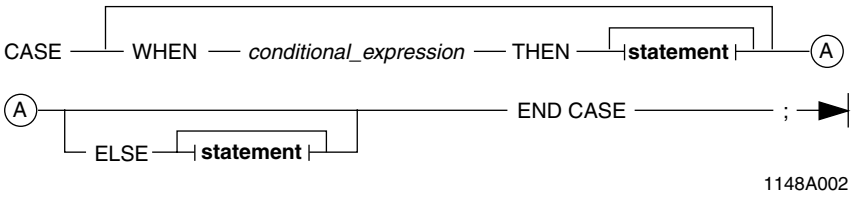



## CASE

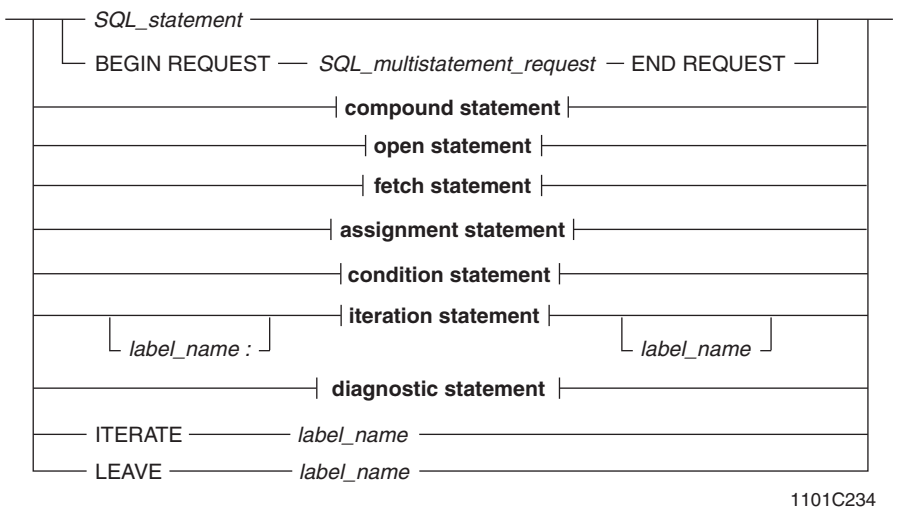
### Syntax 1



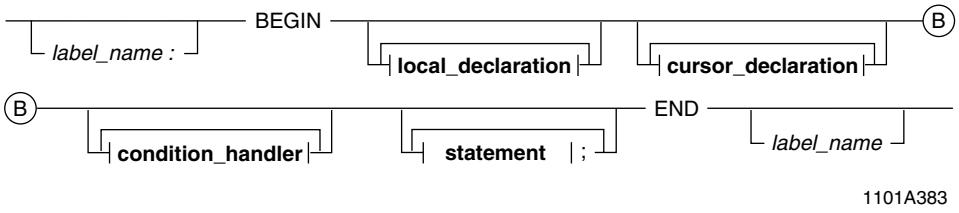

### Syntax 2



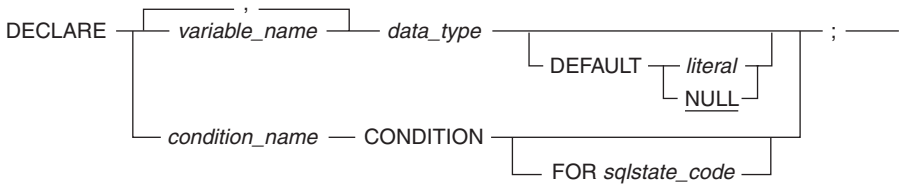
**statement**



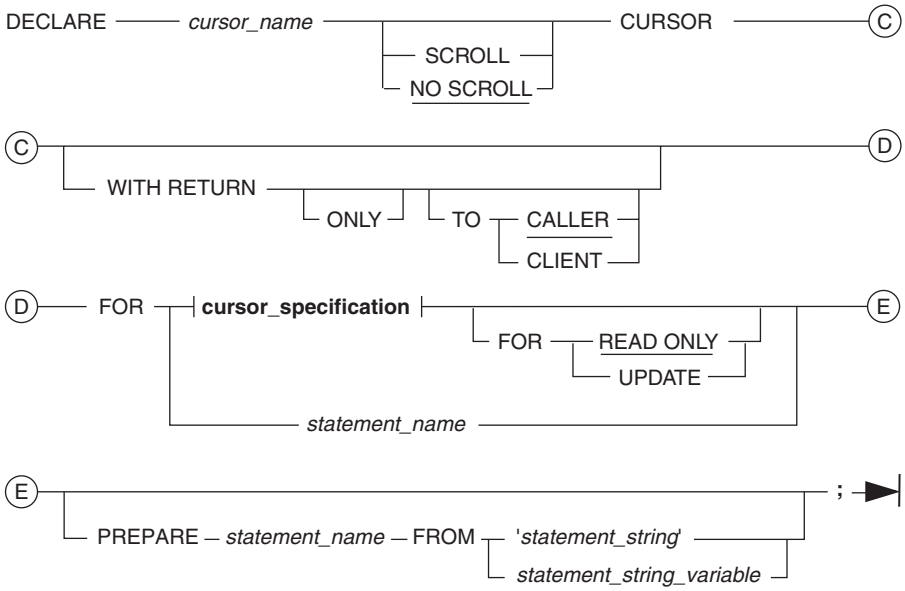

**compound statement**



**local_declaration**

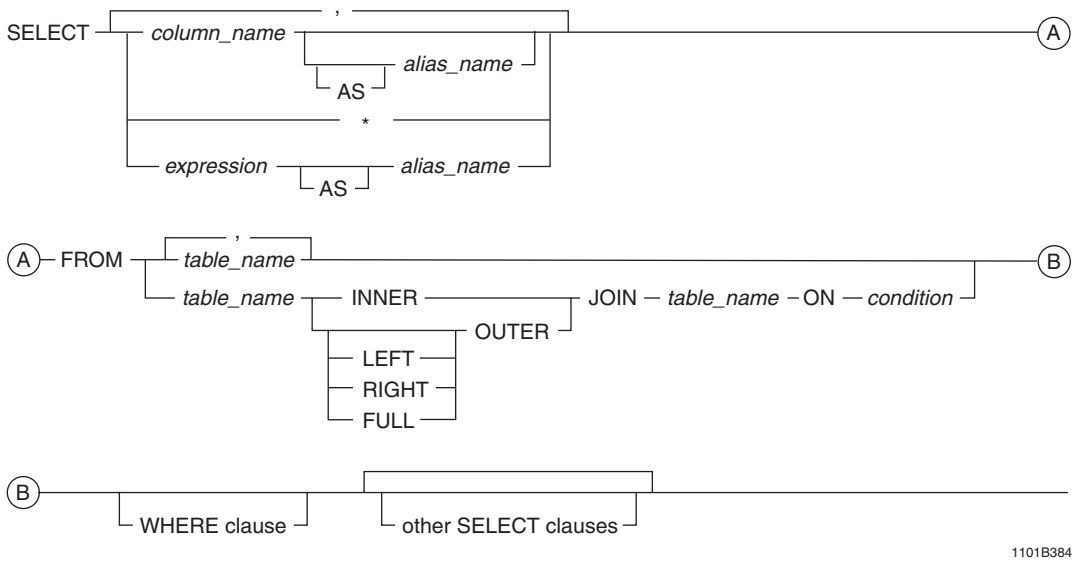



**cursor_declaration**



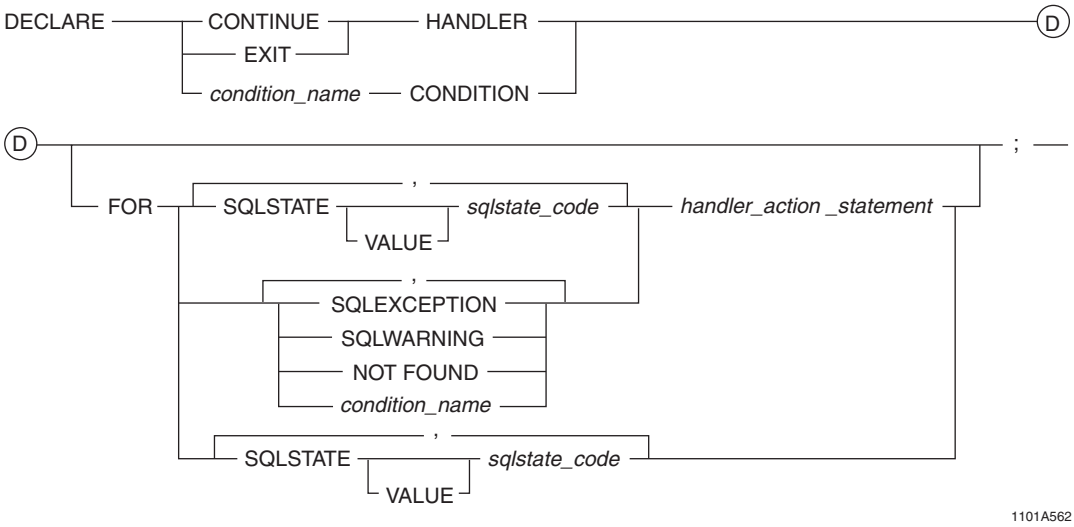

1101C448

**cursor_specification**

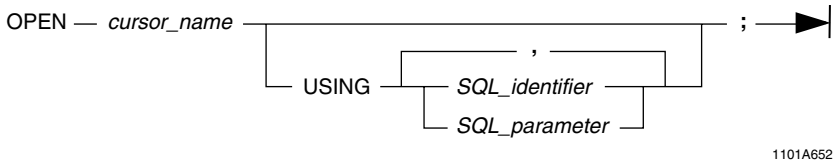



1101B384

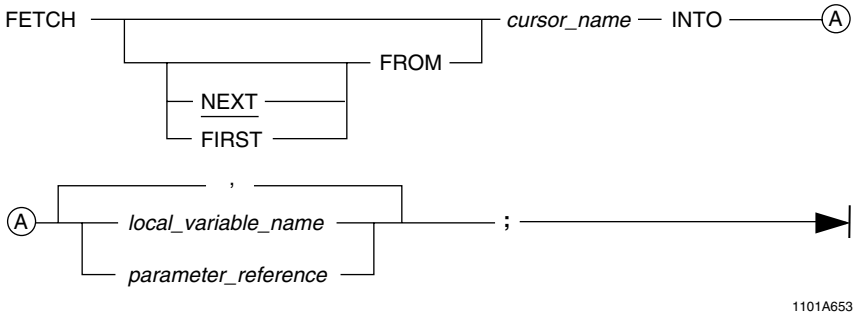
**condition_handler**



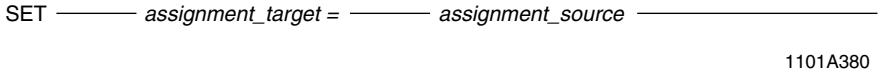

**open statement**



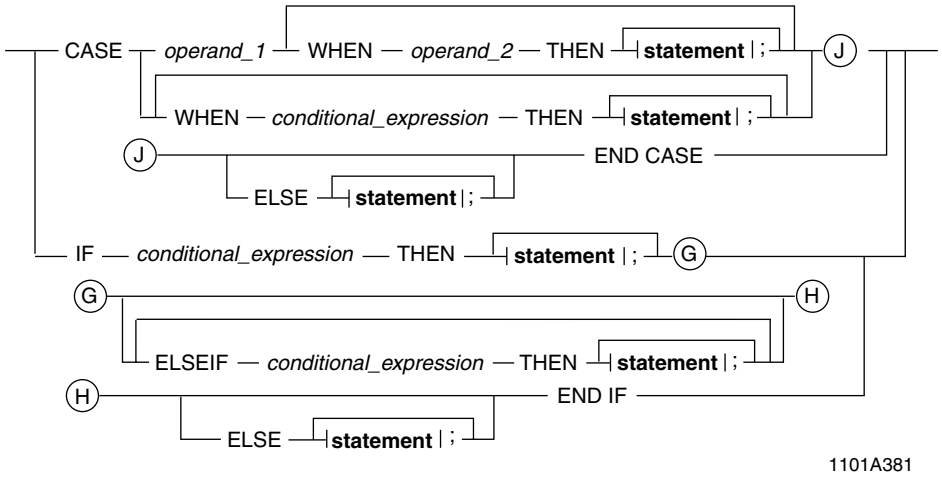

**fetch statement**




**assignment statement**

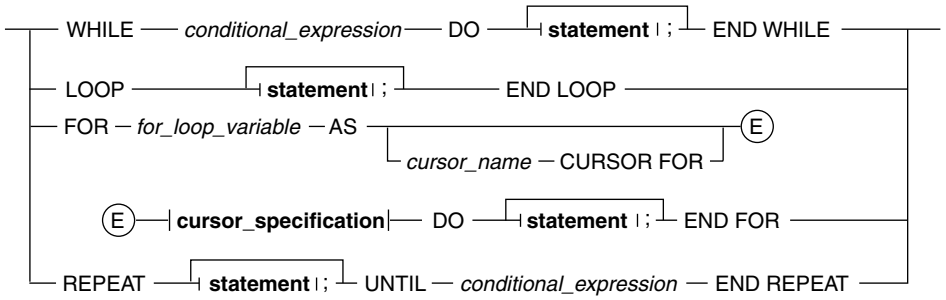


**condition statement**



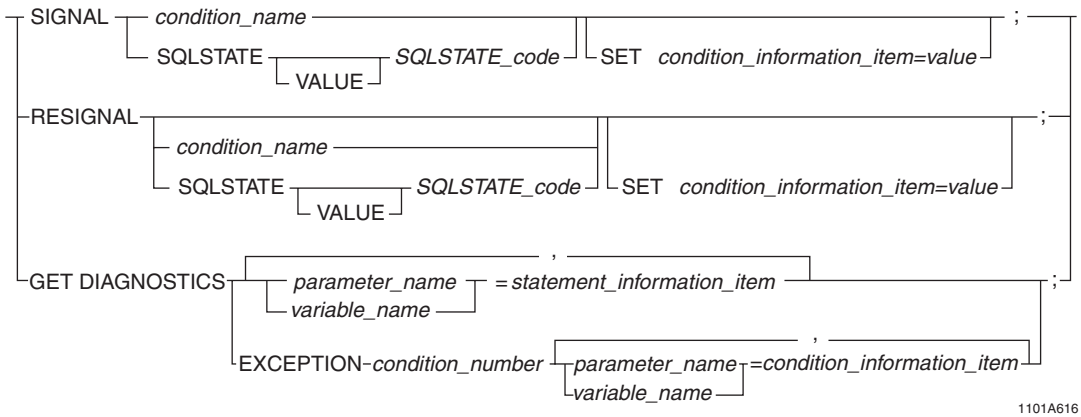

1101A381

**iteration statement**



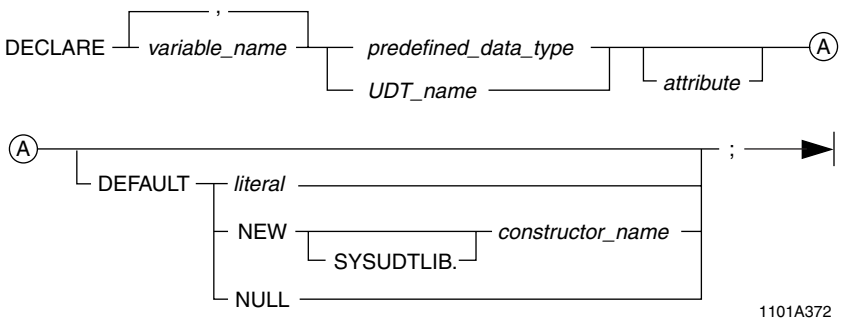

1101A382

**diagnostic statement**

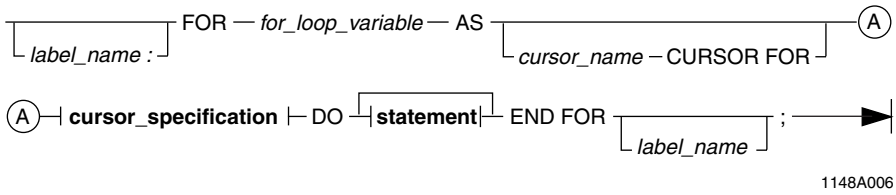



1101A616

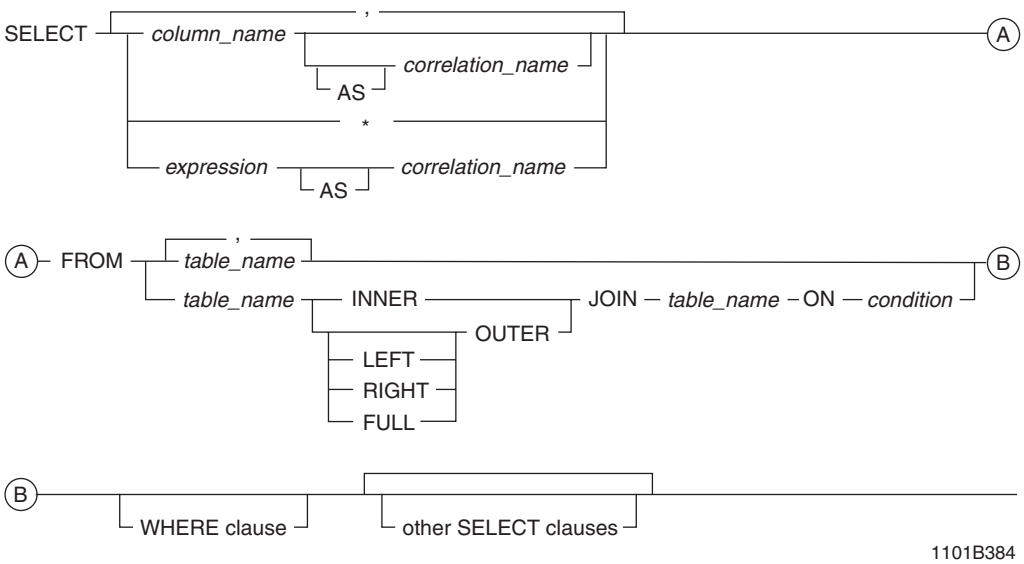
# DECLARE



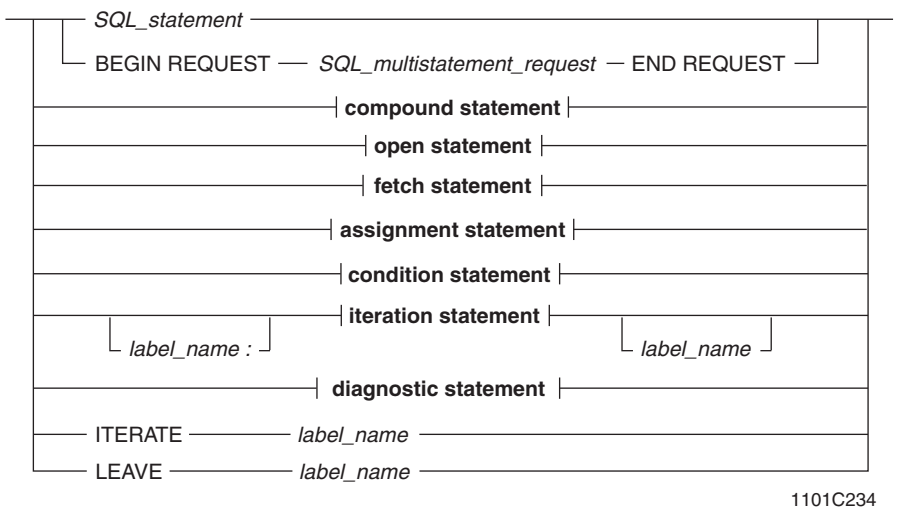

# FOR



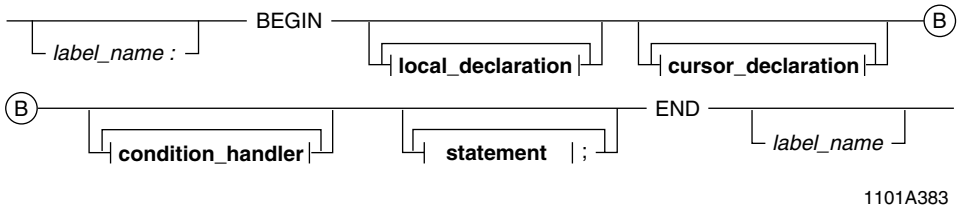

**cursor_specification**



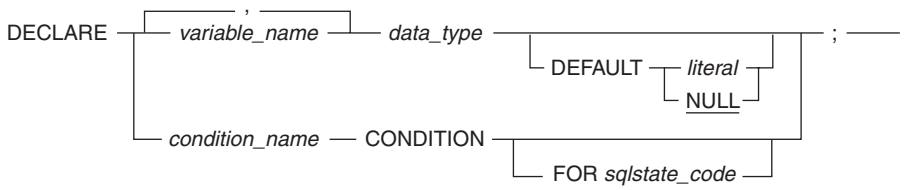
**statement**



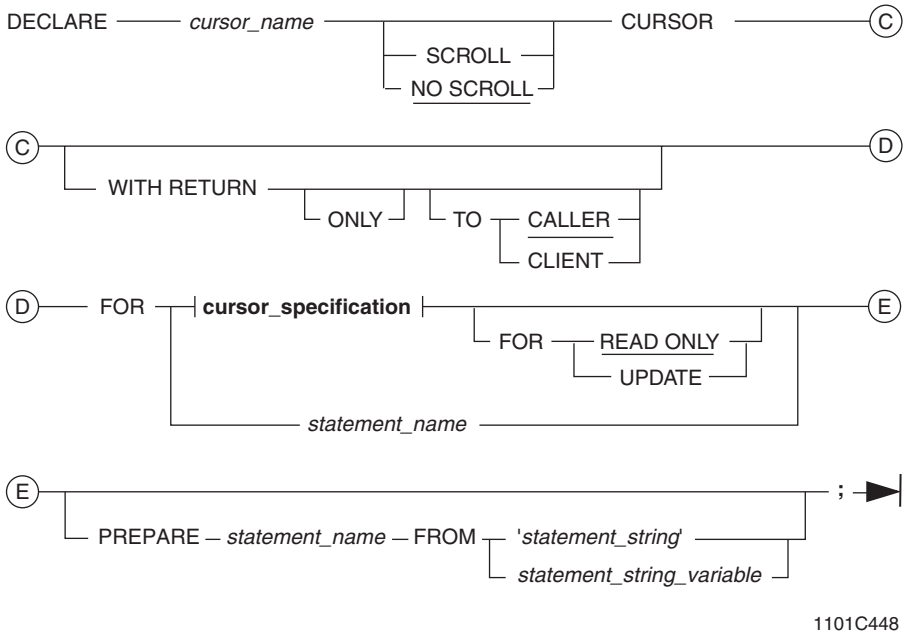

**compound statement**



**local_declaration**

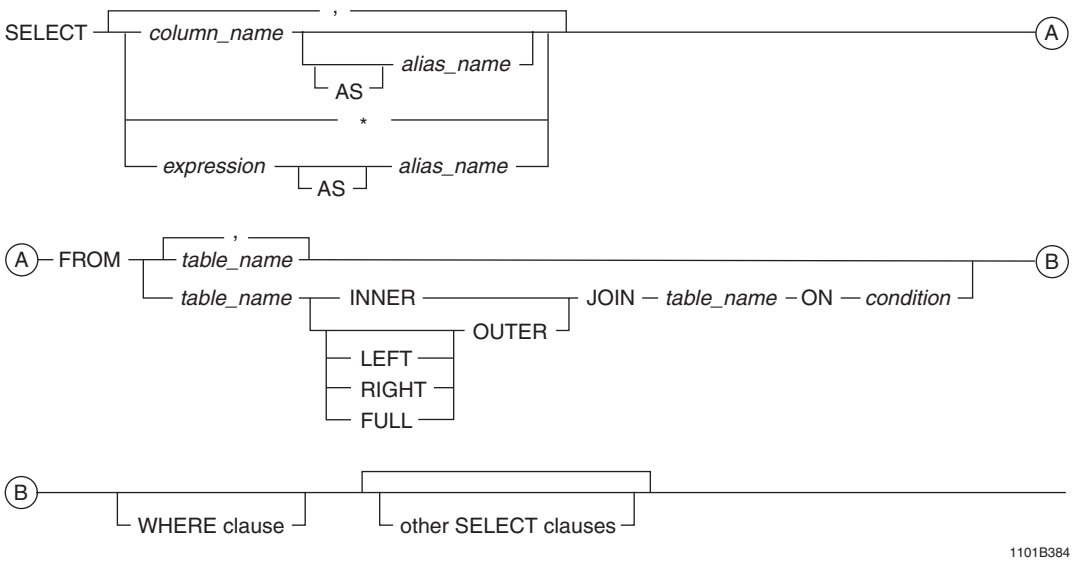



**cursor_declaration**



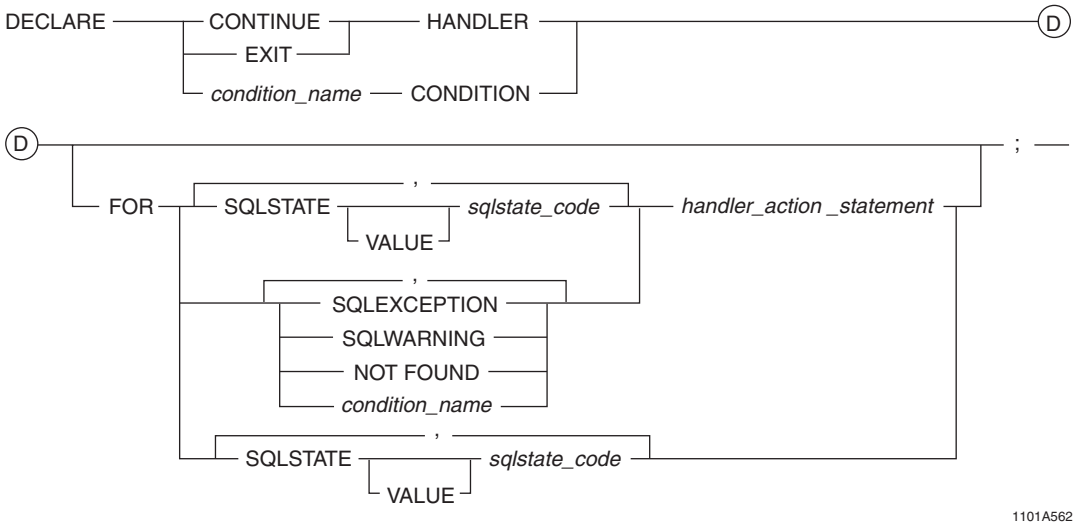

1101C448

**cursor_specification**

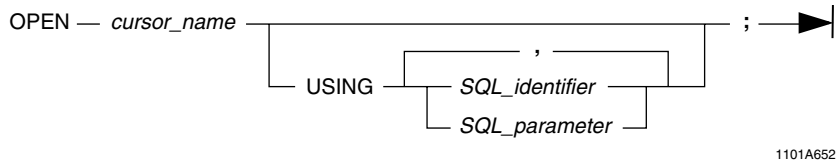



1101B384

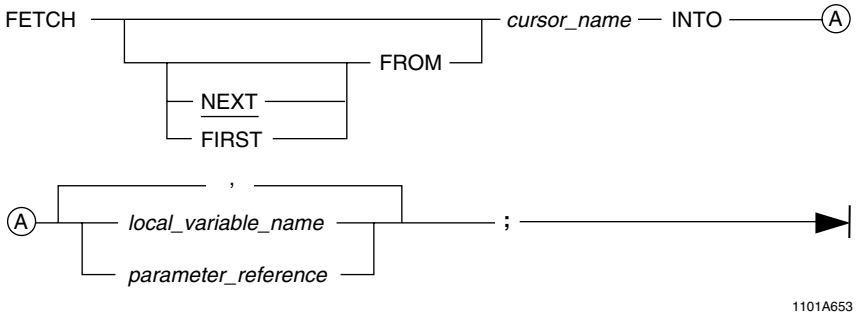
**condition_handler**



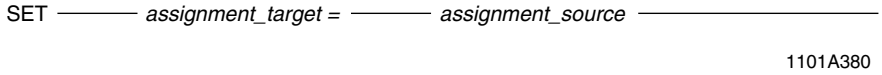

**open statement**



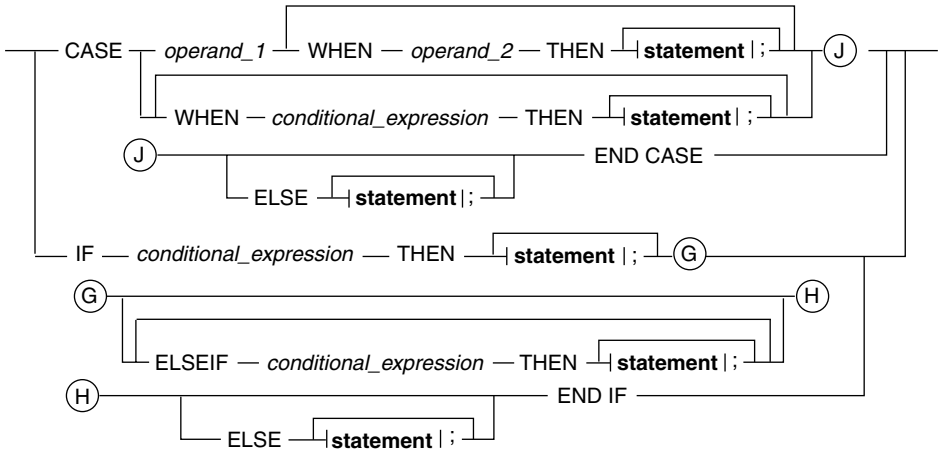

**fetch statement**




**assignment statement**

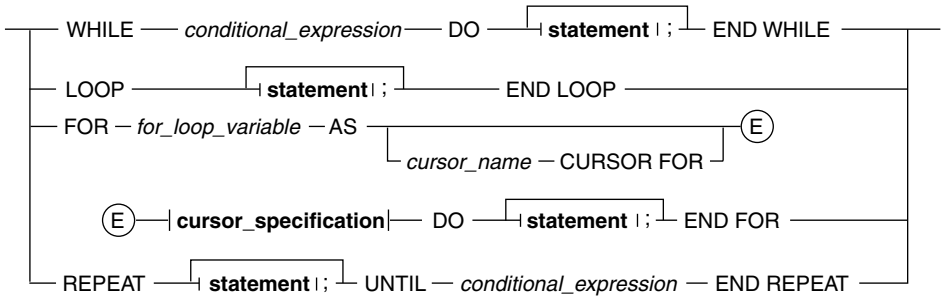


**condition statement**



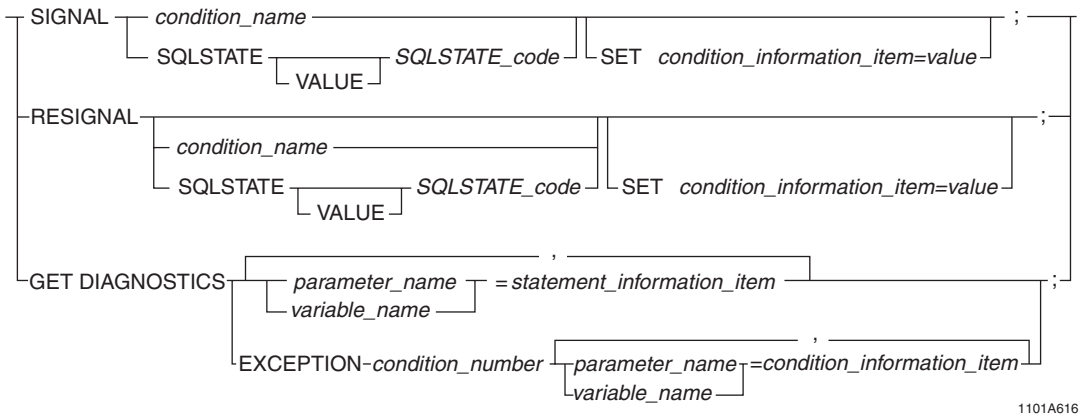

1101A381

**iteration statement**



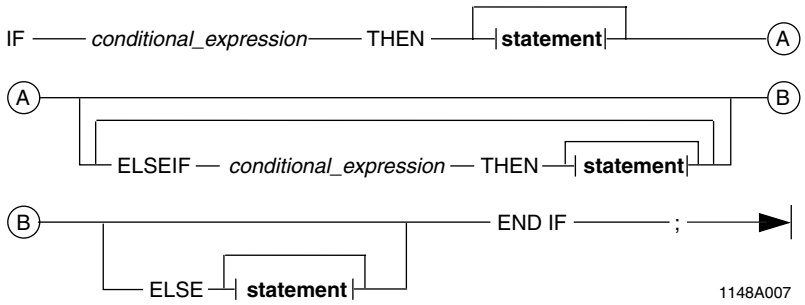

1101A382

**diagnostic statement**

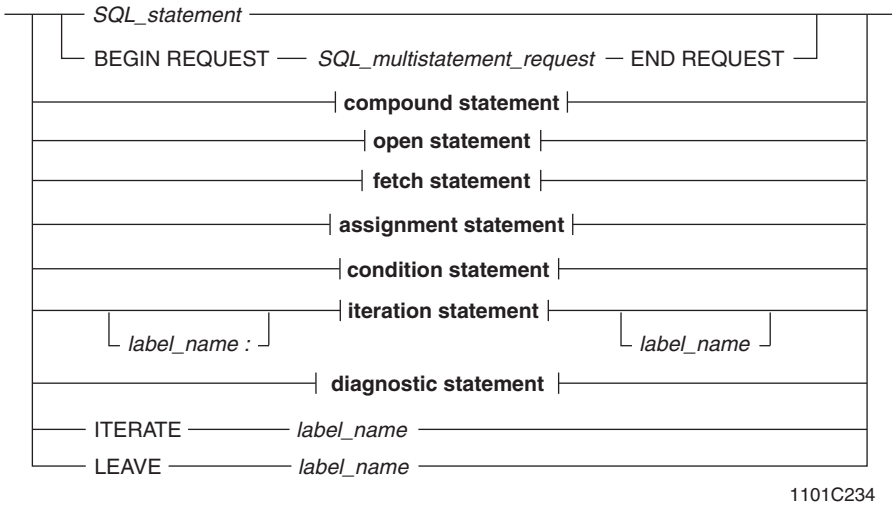



1101A616

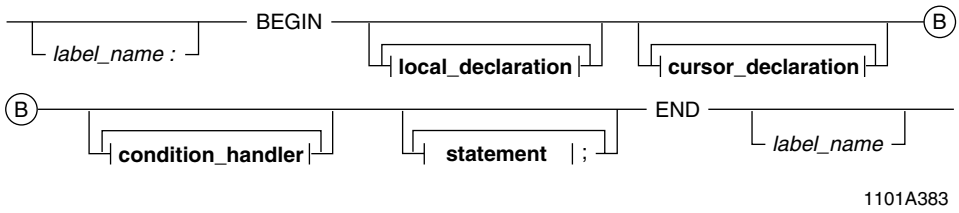
# IF



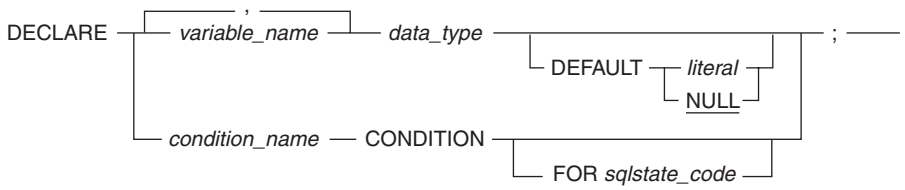

**statement**



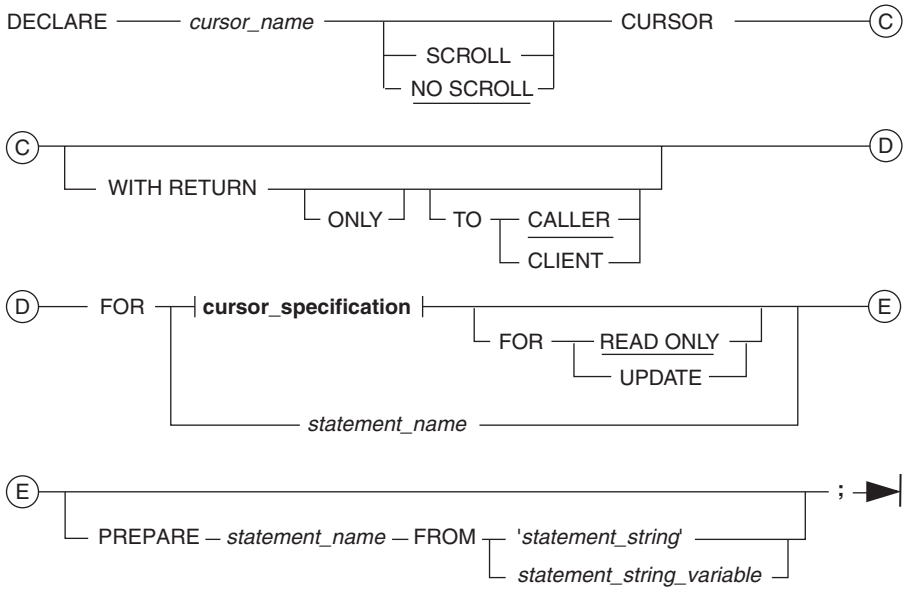

**compound statement**



**local_declaration**

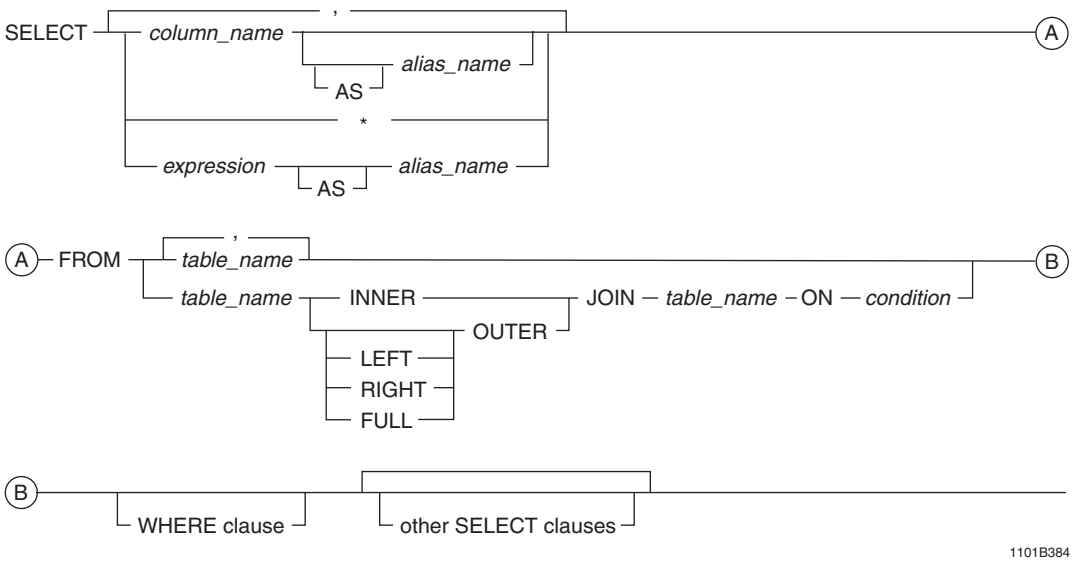



**cursor_declaration**



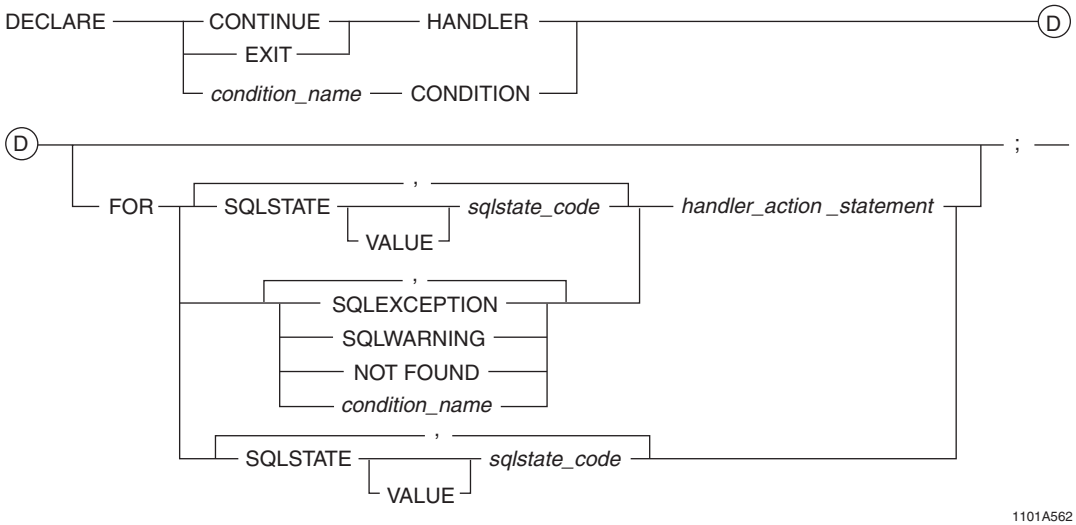

1101C448

**cursor_specification**

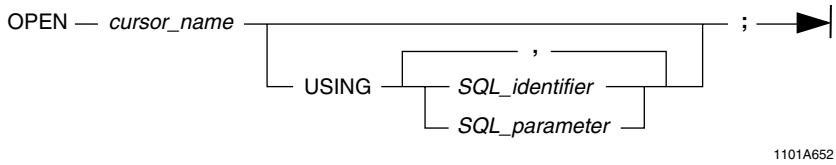



1101B384

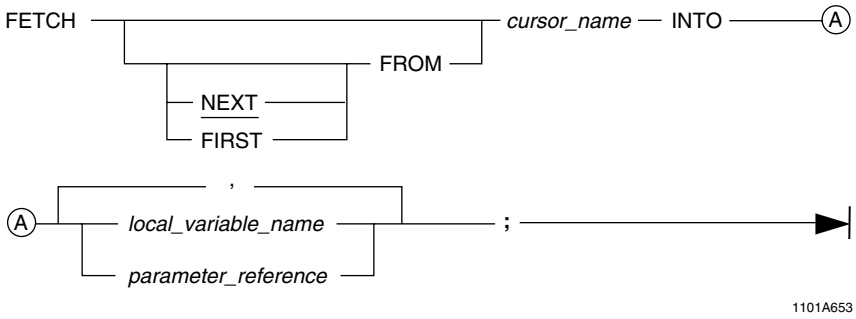
**condition_handler**



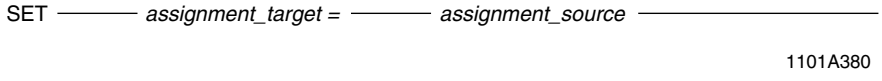

**open statement**



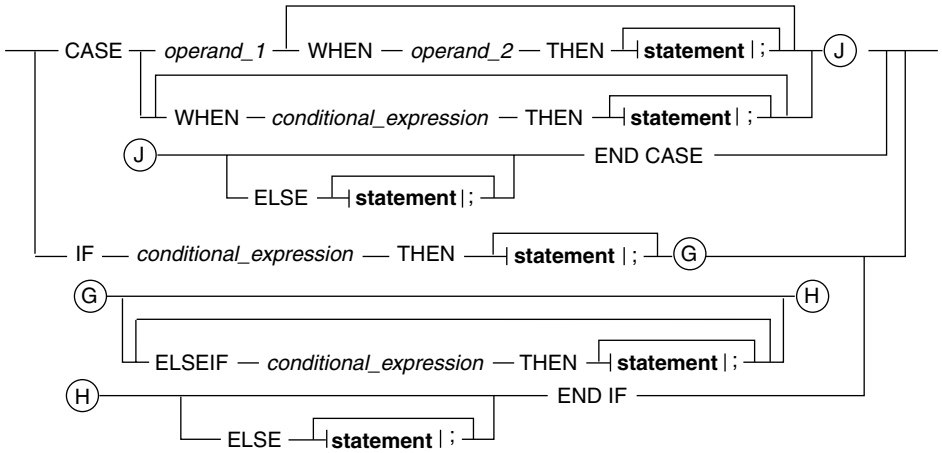

**fetch statement**




**assignment statement**

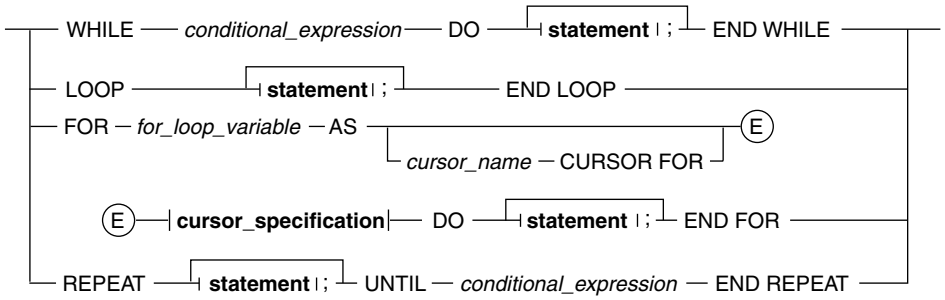


**condition statement**



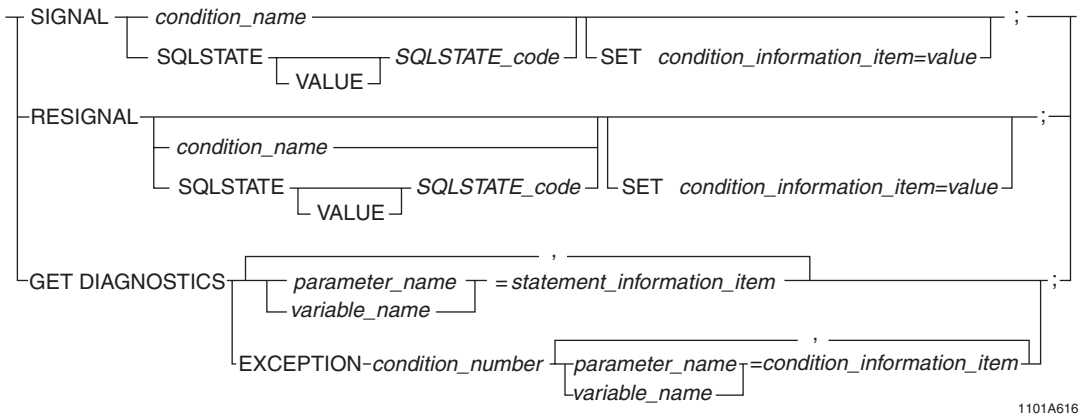

1101A381

**iteration statement**



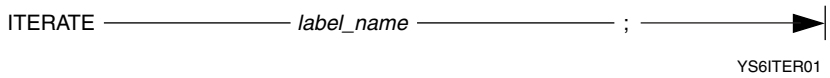

1101A382

**diagnostic statement**



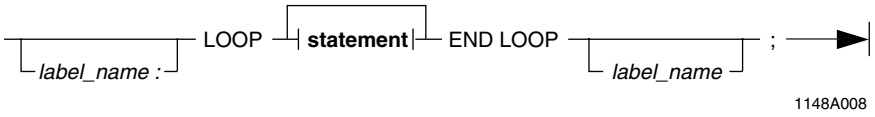

1101A616

# ITERATE

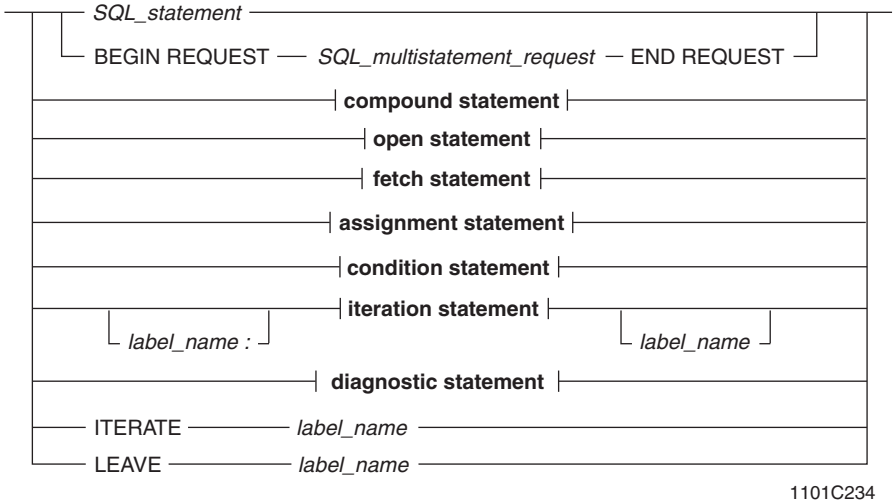



# LEAVE

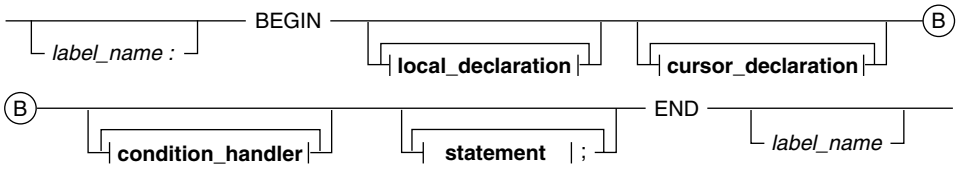

# LOOP



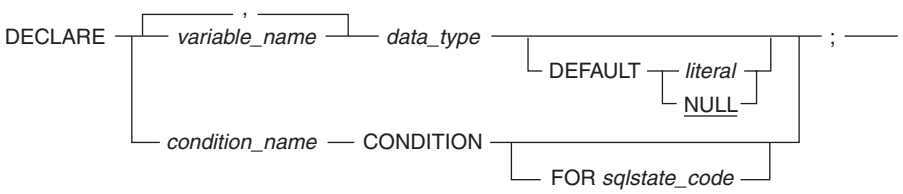

**statement**



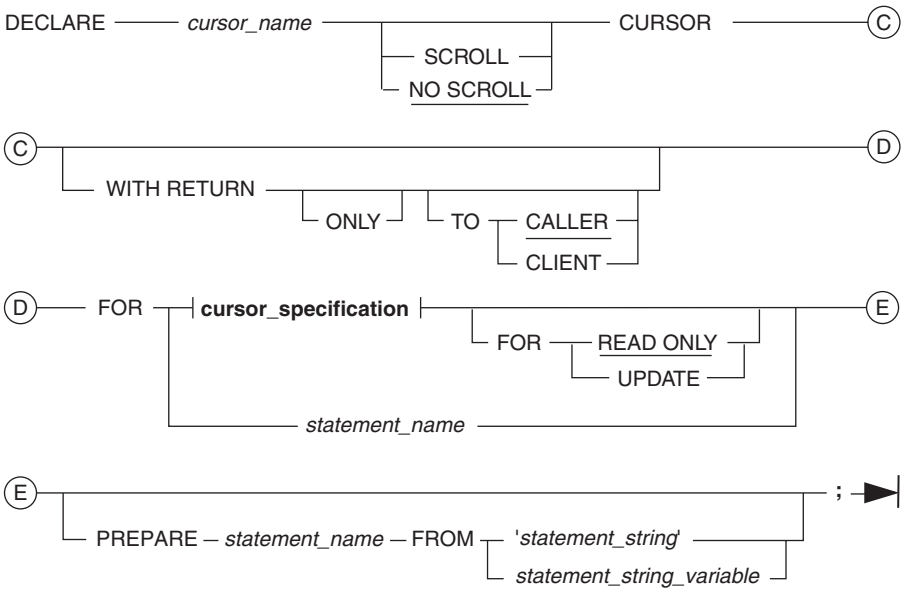
**compound statement**



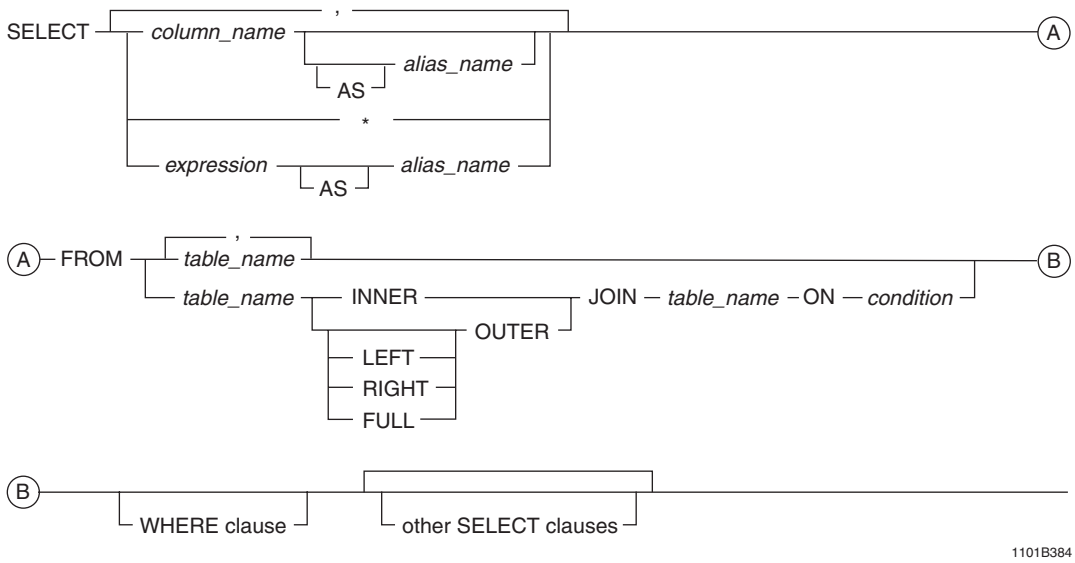

**local_declaration**



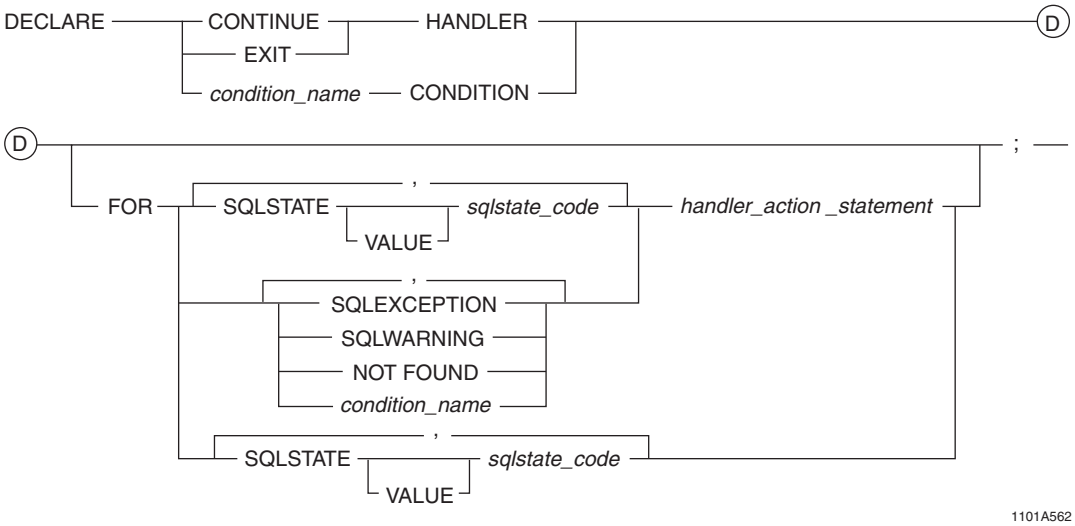

**cursor_declaration**



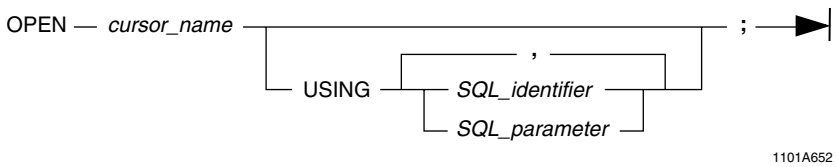
**cursor_specification**



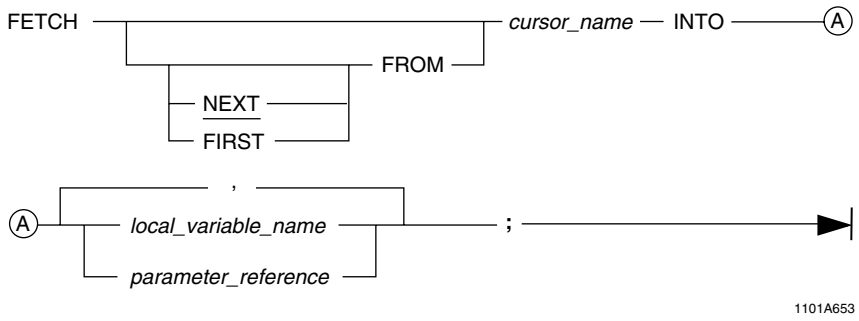

**condition_handler**



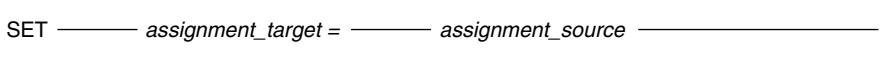

**open statement**



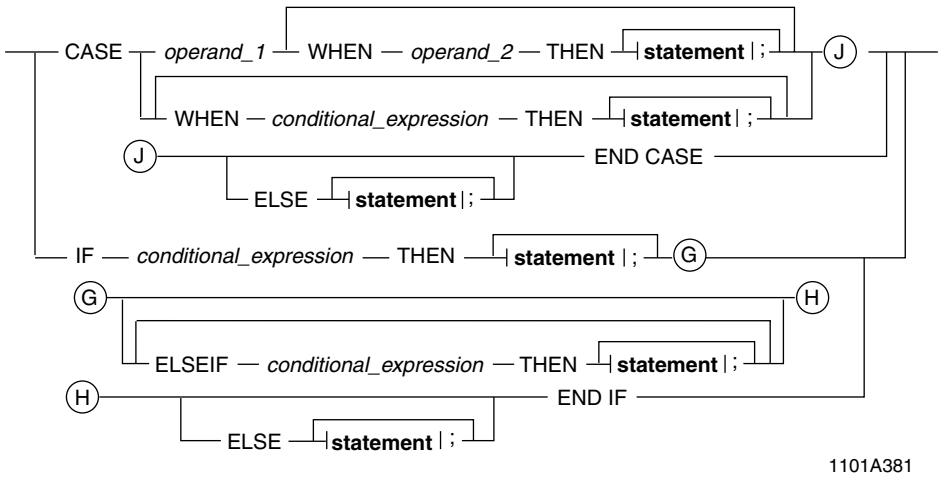
**fetch statement**



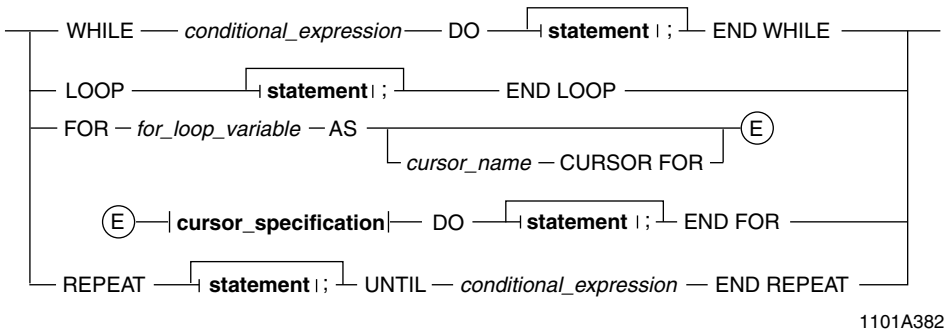

**assignment statement**



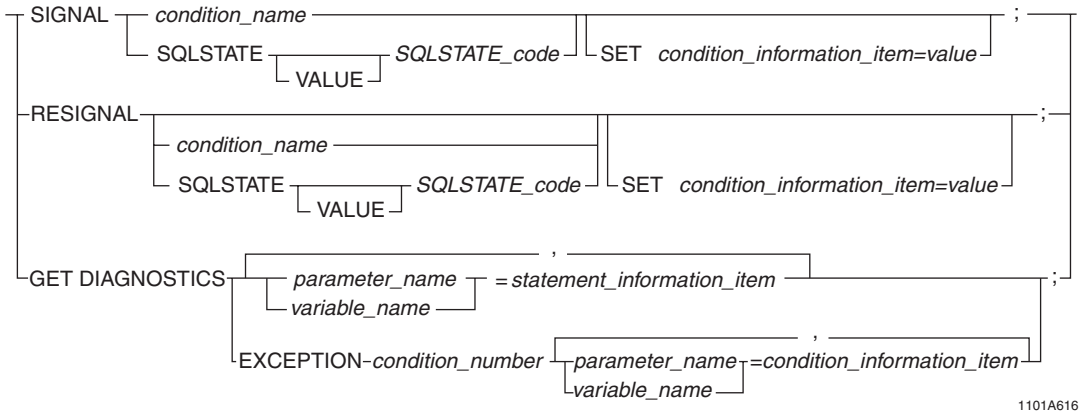

**condition statement**



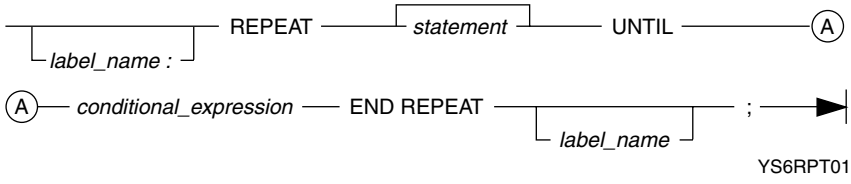
**iteration statement**



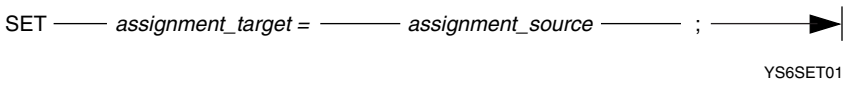

**diagnostic statement**




# REPEAT

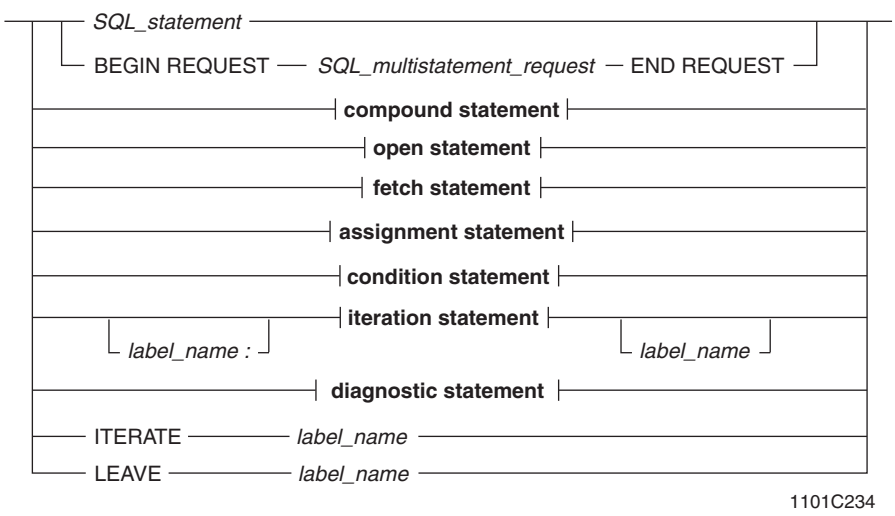


# SET

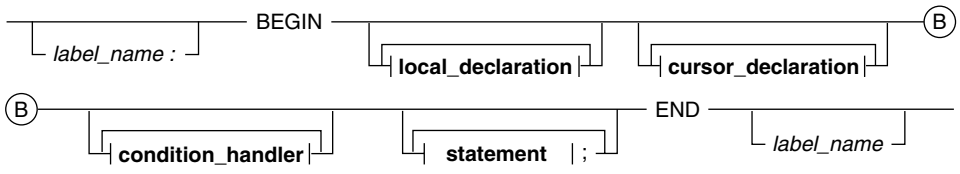



# WHILE


**statement**

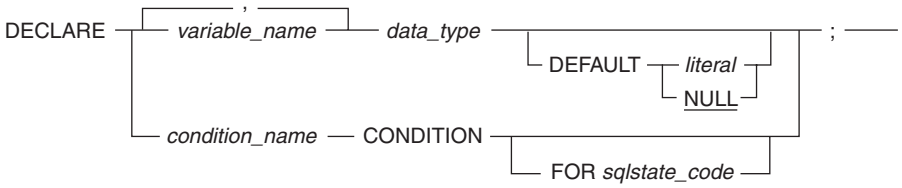


**compound statement**

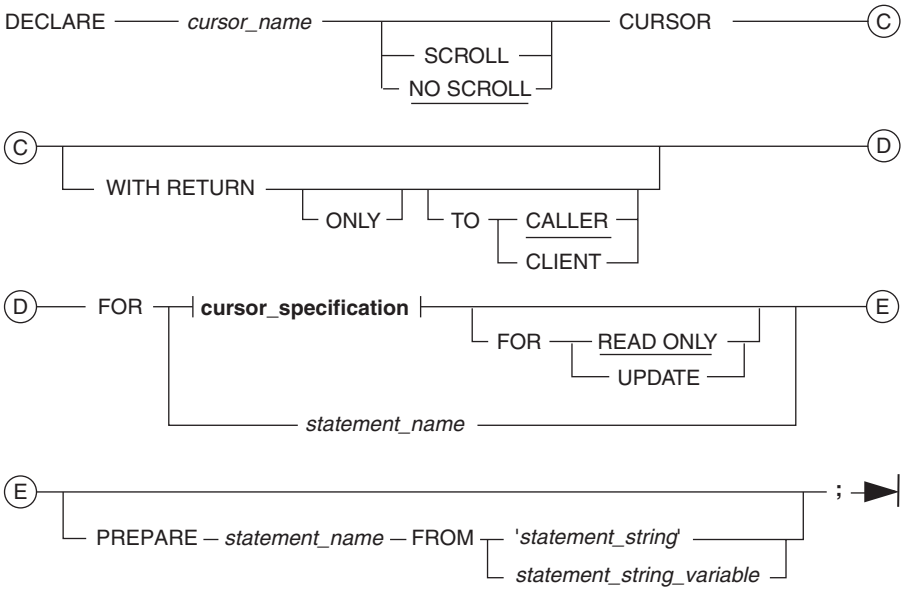



1101A383

**local_declaration**

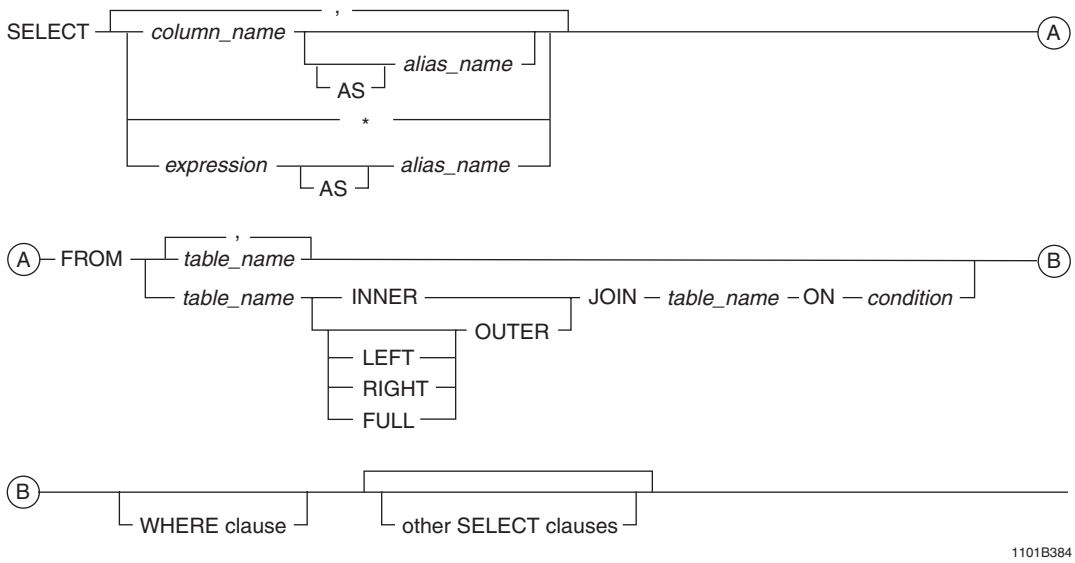



**cursor_declaration**

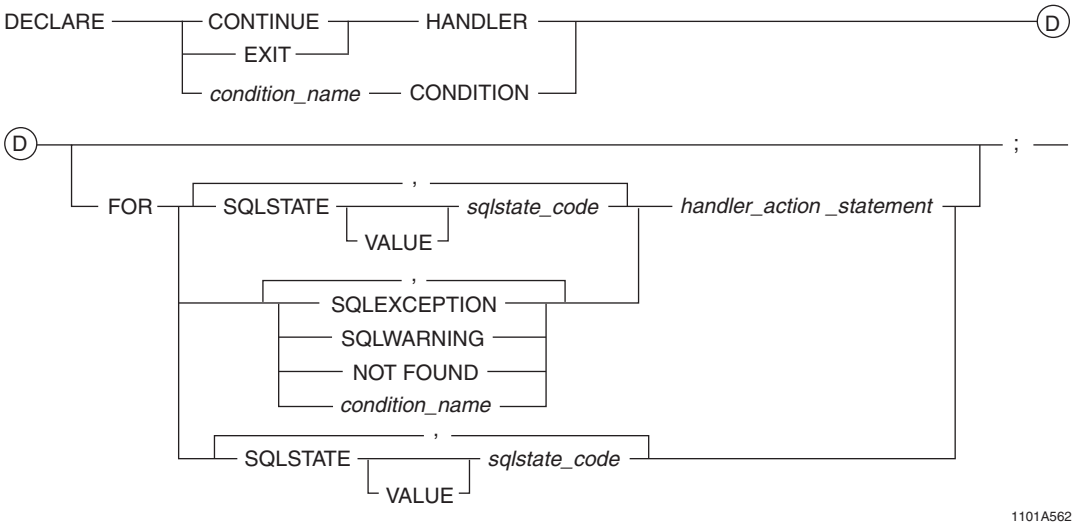



1101C448

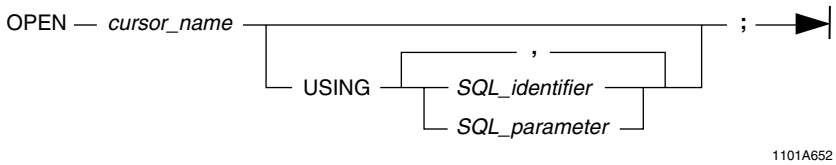
**cursor_specification**



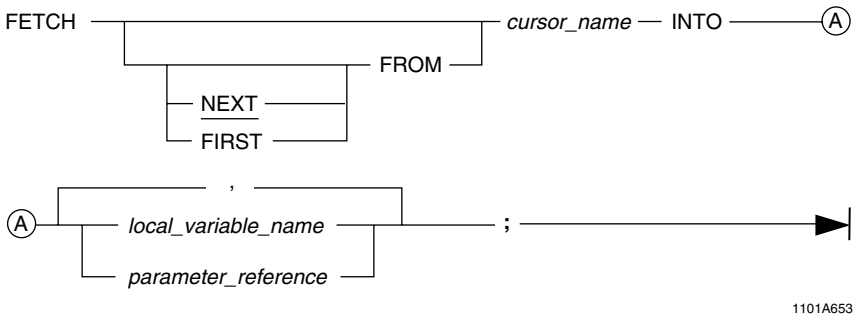

**condition_handler**



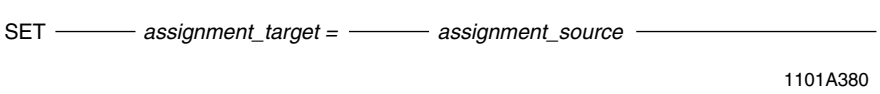

**open statement**



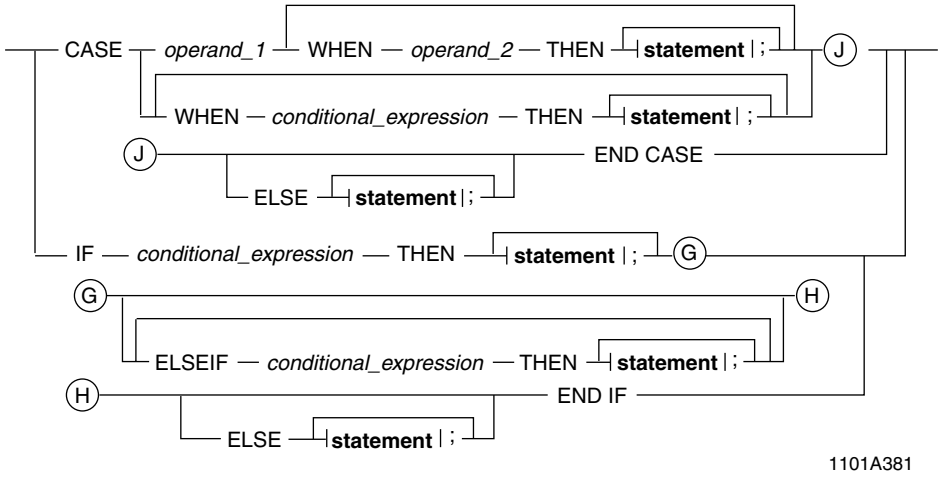
**fetch statement**



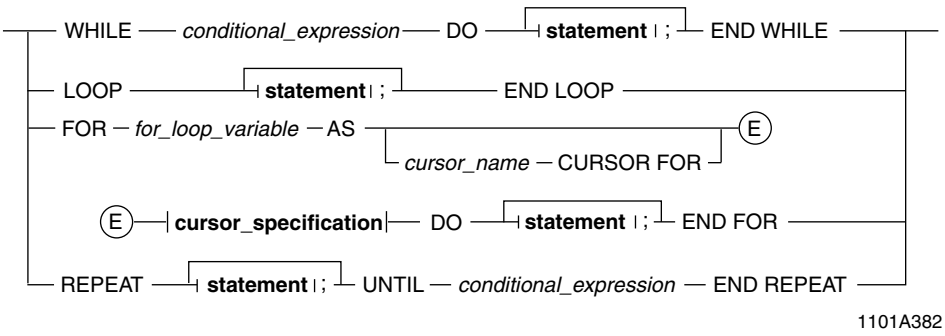

**assignment statement**



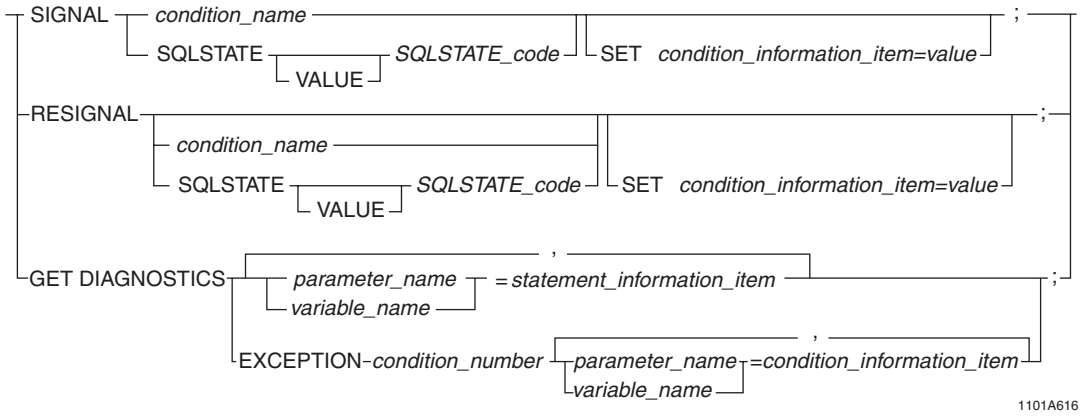

**condition statement**



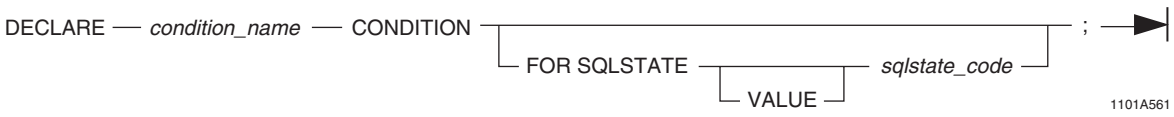
**iteration statement**




**diagnostic statement**

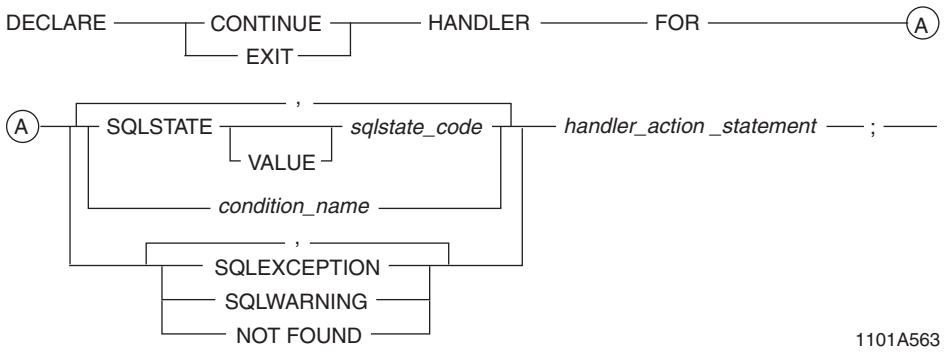



# DECLARE CONDITION

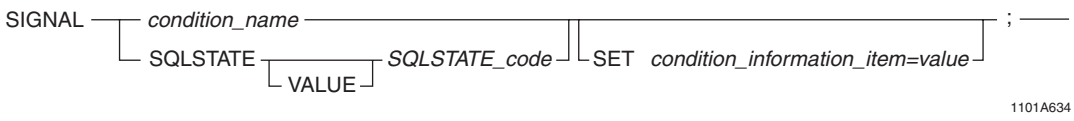


# DECLARE HANDLER (Basic Syntax)

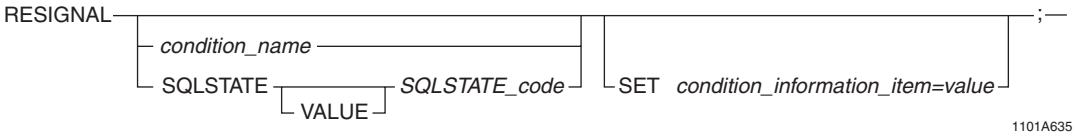
**condition_handler**



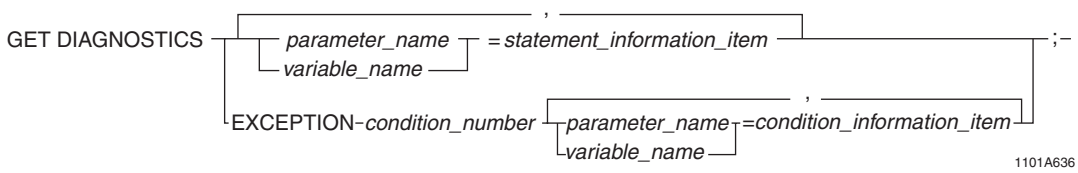

# SIGNAL




# RESIGNAL



# GET DIAGNOSTICS



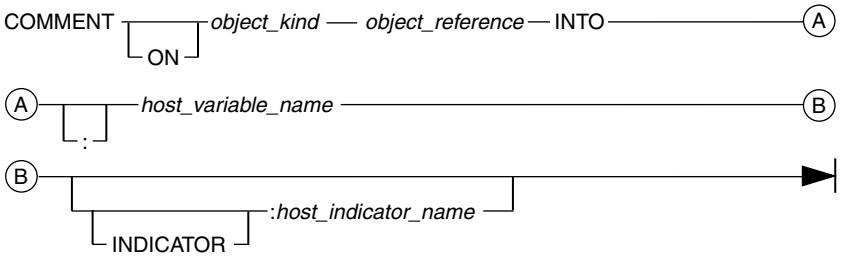
# CHAPTER 8 Static Embedded SQL Statements

## BEGIN DECLARE SECTION


GW01A001

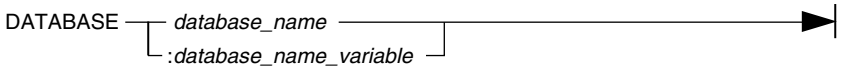
## COMMENT

### Returning Form




1101B015

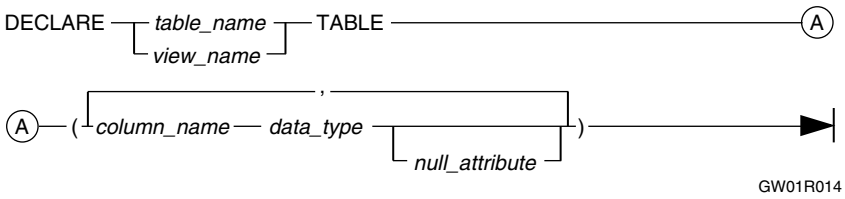
## DATABASE




1101B016

# DECLARE STATEMENT

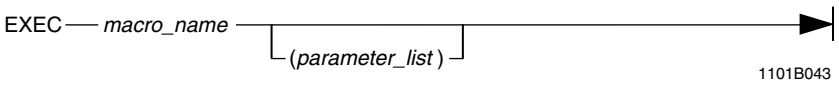

# DECLARE TABLE



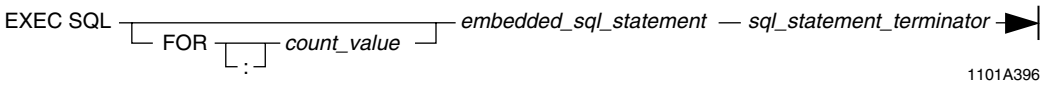

# END DECLARE SECTION


# END-EXEC Statement Terminator

# EXEC



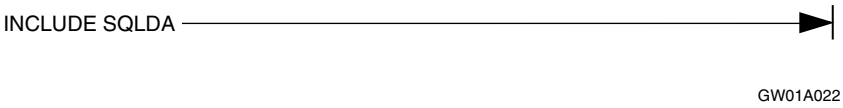

# EXEC SQL Statement Prefix



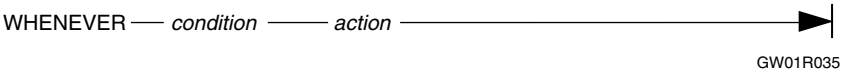

# INCLUDE


# INCLUDE SQLCA

# INCLUDE SQLDA

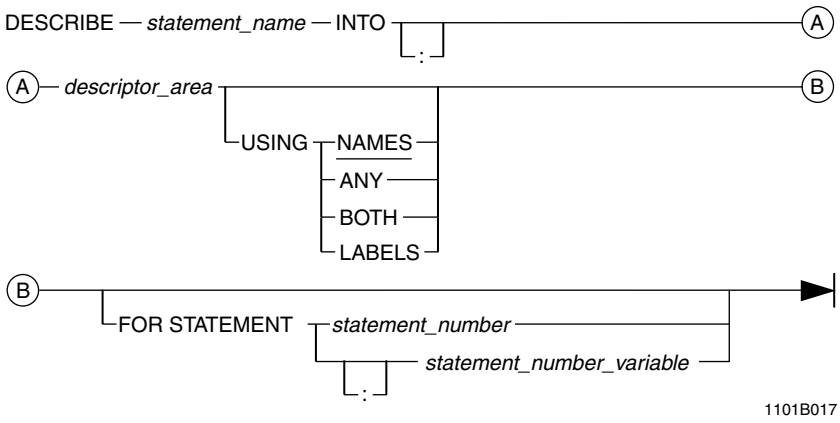



# WHENEVER



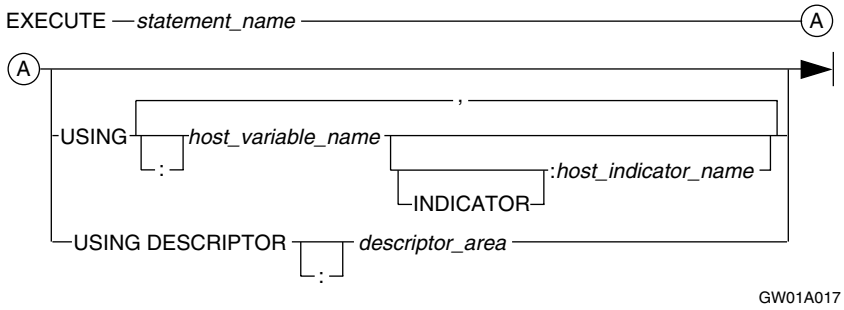
# CHAPTER 9 Dynamic Embedded SQL Statements

## DESCRIBE

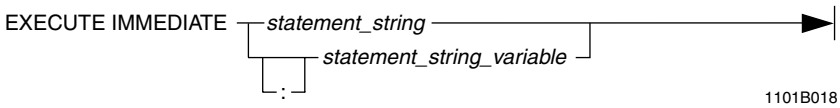



## EXECUTE

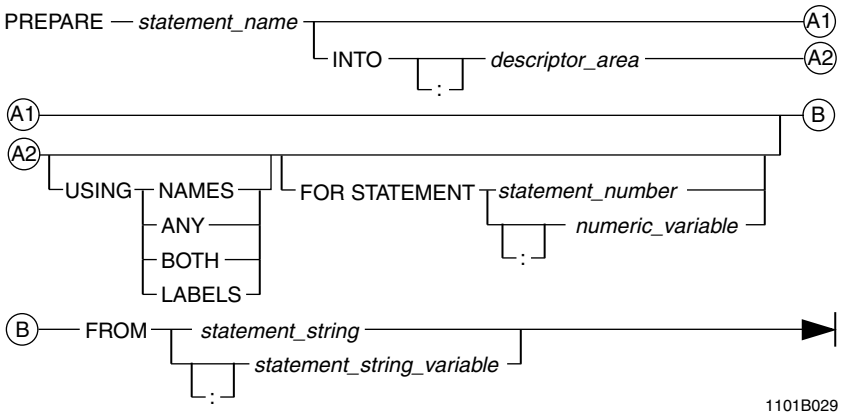
### Dynamic SQL Form



# EXECUTE IMMEDIATE

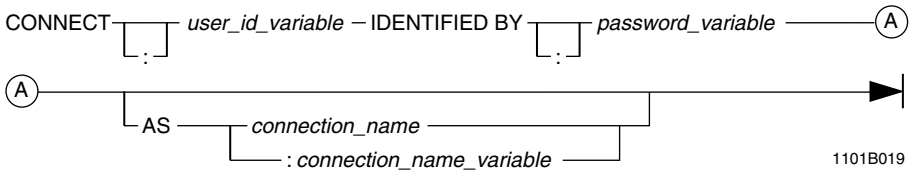



# PREPARE

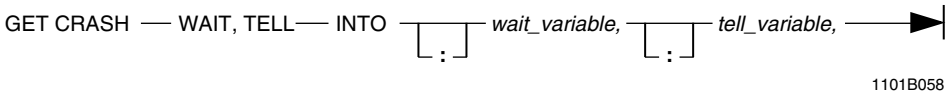


# CHAPTER 10 SQL Client-Server Connectivity Statements

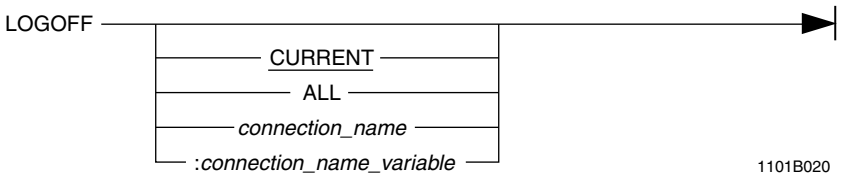
## CONNECT



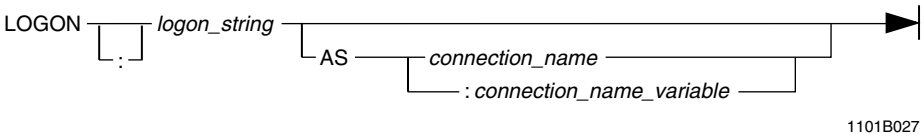

## GET CRASH




## LOGOFF

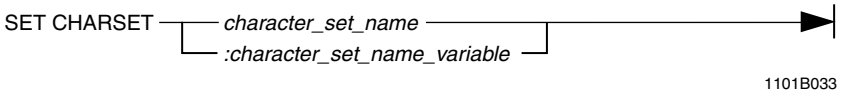


# LOGON

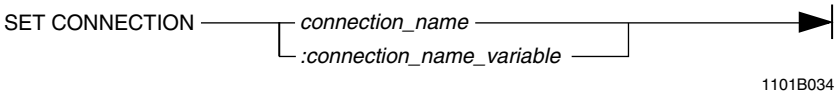



# SET BUFFERSIZE


# SET CHARSET




# SET CONNECTION

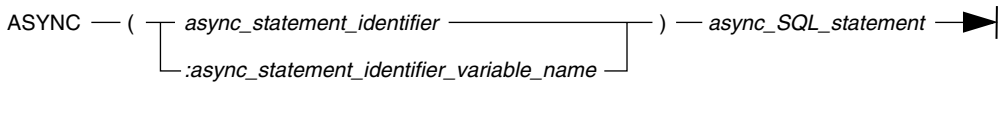



# SET CRASH

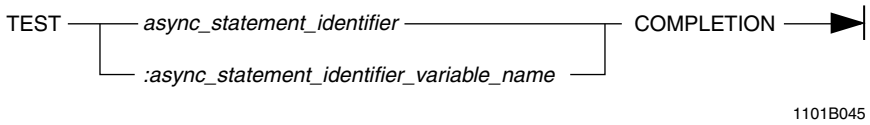
# SET ENCRYPTION

# CHAPTER 11 Multisession Asynchronous Programming With Embedded SQL

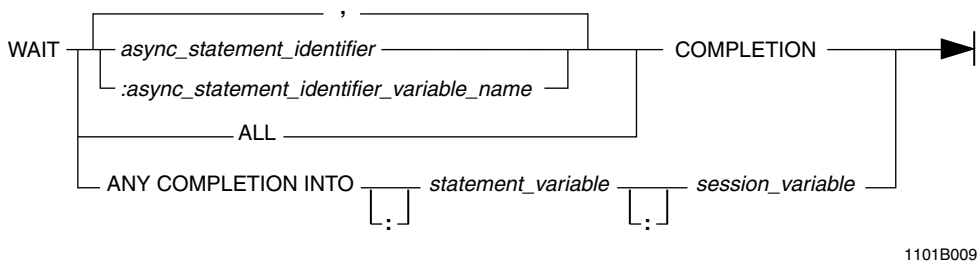
## ASYNC Statement Modifier




## 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.<br>Do not use commas when typing a number with more than 3 digits. |
| Word | Keywords and variables.<br>• UPPERCASE LETTERS represent a keyword.<br>Syntax diagrams show all keywords in uppercase, unless operating system restrictions require them to be in lowercase.<br>• lowercase letters represent a keyword that you must type in lowercase, such as a UNIX command.<br>• *lowercase italic letters* represent a variable such as a column or table name.<br>Substitute the variable with a proper value.<br>• **lowercase bold letters** represent an excerpt from the diagram. The excerpt is defined immediately following the diagram that contains it.<br>• UNDERLINED LETTERS represent the default value.<br>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. |

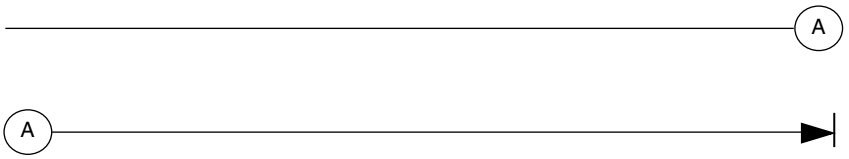
### Paths

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:

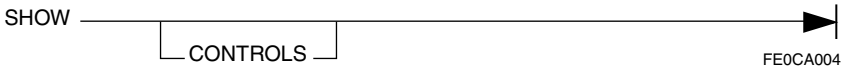

FE0CA003

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

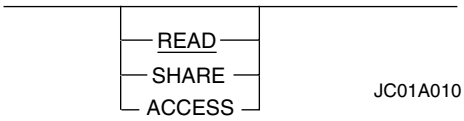
## Optional Entries

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




FE0CA004

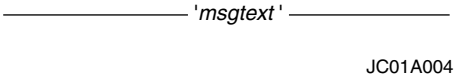
If you can optionally choose from more than one entry, all the choices appear below the main path:




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

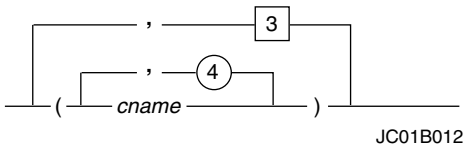
If a keyword or a reserved word has a valid abbreviation, the unabbreviated form always appears on the main path. The shortest valid abbreviation appears beneath.


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.

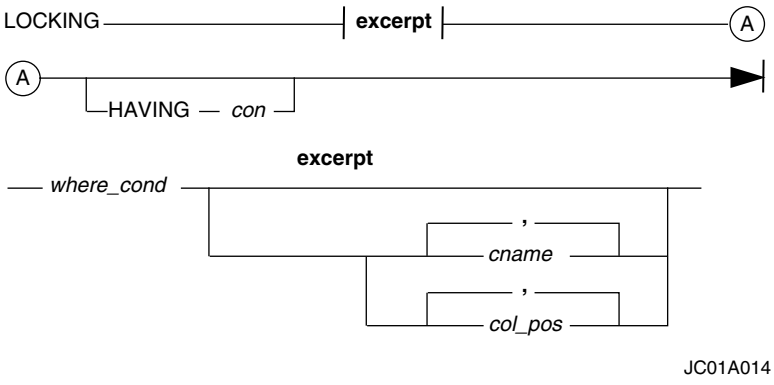
The following conventions apply to loops:

| IF... | THEN... |
|---|---|
| there is a maximum number of entries allowed | the number appears in a circle on the return path.<br>In the example, you may type *cname* a maximum of 4 times. |
| there is a minimum number of entries required | the number appears in a square on the return path.<br>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.<br>If the diagram does not show a separator character, use one blank space.<br>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.<br>Generally, a space is not needed between delimiter characters and entries.<br>In the example, the delimiter characters are the left and right parentheses. |

## Excerpts

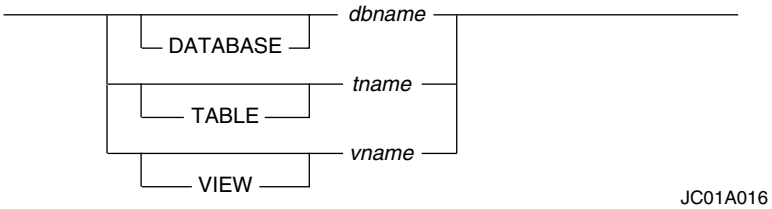
Sometimes a piece of a syntax phrase is too large to fit into the diagram. Such a phrase is indicated by a break in the path, marked by (|) terminators on each side of the break. The name for the excerpted piece appears between the terminators in boldface type.

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:




## Multiple Legitimate Phrases

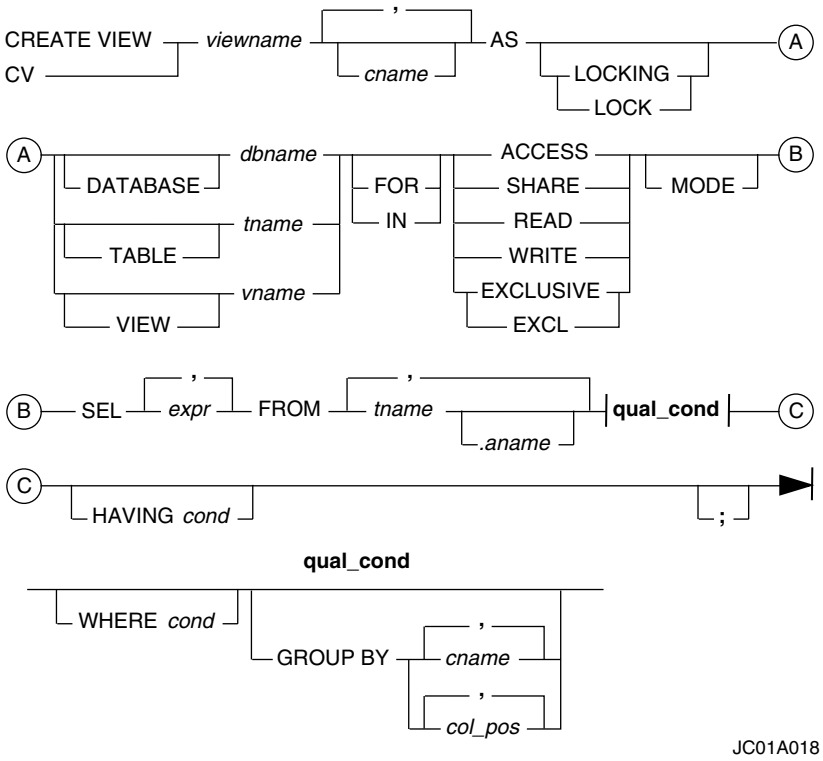
In a syntax diagram, it is possible for any number of phrases to be legitimate:




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.