DB2® Server for VSE & VM

Quick Reference

Version 6 Release 1

SC09-2670-00

DB2® Server for VSE & VM

Quick Reference

Version 6 Release 1

SC09-2670-00

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page vii.

This book is also provided as an online book that can be viewed with the IBM® BookManager® READ and IBM Library Reader™ licensed programs.

First Edition (December 1998)

This edition, SC09-2670, applies to Version 6 Release 1 of the IBM DATABASE 2™ Server for VSE & VM Program 5648-A70, and to all subsequent releases of this product until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to: IBM Canada Ltd. Laboratory, Information Development, 2G/345/1150/TOR 1150 Eglinton Ave East North York, Ontario, Canada. M3C 1H7

You can also send your comments by facsimile to (416) 448-6161 addressed to the attention of the RCF Coordinator. If you have access to the Internet, you can send your comments electronically to **torrcf@ca.ibm.com;** IBMLink™, to **toribm(torrcf);** IBM/PROFS®, to **torolab4(torrcf)**; IBMMAIL, to **ibmmail(caibmwt9)**; or through our home page at **http://www.software.ibm.com/data/db2/vse-vm**

If you choose to respond through the Internet, please include either your entire network address, or a postal address.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM
| Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Canada Ltd., Department 071, 1150 Eglinton Avenue East, North York, Ontario Canada M3C 1H7. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

This publication may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Programming Interface Information

This book documents intended Programming Interfaces that allow the customer to write programs to obtain services of DB2 Server for VSE & VM.

Trademarks

| The following terms are trademarks of International Business
| Machines Corporation in the United States and/or other countries:

IBM
CICS
CICS/VSE
DB2
DATABASE 2 Server
PROFS
IBMLink
BookManager
Library Reader

| Other company, product, and service names may be trademarks or
| service marks of others.

About This Manual

This reference pictorially summarizes Structured Query Language statements used by:

- DB2 Server for VM and DB2 Server for VSE
- Interactive SQL Facility (ISQL) commands
- Database Services Utility (DBS Utility) commands.

It also contains information about the following:

- SQL language elements
- Functions
- Queries
- Preprocessing application programs
- ISQL program function keys
- Operator commands
- Catalog tables
- Application server support for remote applications
- Multiple application server support for DB2 Server for VSE
- SQL communication area (SQLCA) and SQL descriptor area (SQLDA)
- SQL reserved words
- Database Services Utility reserved words.

Who Should Use This Manual

This manual is intended as a quick reference for application developers, system programmers, and database administrators who write application programs using SQL, or use ISQL, or the Database Services Utility in a DB2 Server for VM or DB2 Server for VSE environment. It contains syntax diagrams for SQL statements, ISQL commands, operator commands, and DBS Utility commands.

It is assumed that the VM user has some knowledge of VM (CMS, CP), a programming language, and structured query language (SQL). It is assumed that the VSE user has some knowledge of a VSE system, a CICS/VSE® system or batch as applicable, a programming language, and structured query language (SQL).

Both the VSE and VM user should be familiar with the information in the *DB2 Server for VSE & VM Overview*. For further information on the required environment, refer to either the *DB2 Server for VM Program Directory* or the *DB2 Server for VSE Installation* for your database manager.

Related Publications

For more information about the DB2 Server for VM and DB2 Server for VSE database managers, ISQL, and the DBS Utility, refer to the following IBM publications for DB2 Server for VM or DB2 Server for VSE as appropriate:

- *DB2 Server for VSE & VM Overview*
- *DB2 Server for VSE & VM SQL Reference*
- *DB2 Server for VSE & VM Interactive SQL Guide and Reference*
- *DB2 Server for VSE & VM Database Services Utility*
- *DB2 Server for VSE & VM Operation*.

Syntax Notation Conventions

Throughout this manual, syntax is described using the structure defined below.

- Read the syntax diagrams from left to right and from top to bottom, following the path of the line.

  The >>—— symbol indicates the beginning of a statement or command.

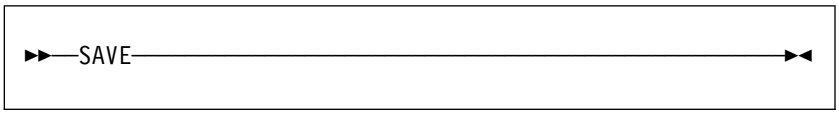
  The ——> symbol indicates that the statement syntax is continued on the next line.

  The >—— symbol indicates that a statement is continued from the previous line.

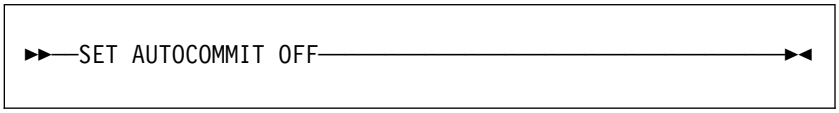
  The ——>< symbol indicates the end of a statement.

  Diagrams of syntactical units that are not complete statements start with the >—— symbol and end with the ——> symbol.

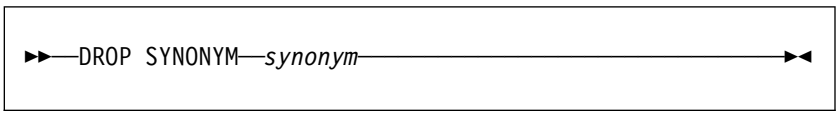
- Some SQL statements, Interactive SQL (ISQL) commands, or database services utility (DBS Utility) commands can stand alone. For example:



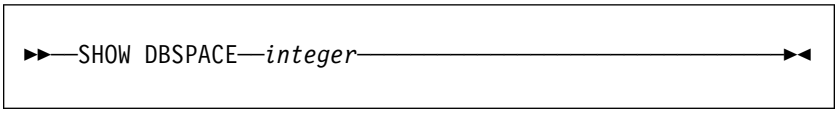
Others must be followed by one or more keywords or variables. For example:



- Keywords may have parameters associated with them which represent user-supplied names or values. These names or values can be specified as either constants or as user-defined variables called *host_variables* (*host_variables* can only be used in programs).



- Keywords appear in either uppercase (for example, SAVE) or mixed case (for example, CHARacter). All uppercase characters in keywords must be present; you can omit those in lowercase.
- Parameters appear in lowercase and in italics (for example, *synonym*).
- If such symbols as punctuation marks, parentheses, or arithmetic operators are shown, you must use them as indicated by the syntax diagram.
- All items (parameters and keywords) must be separated by one or more blanks.
- Required items appear on the same horizontal line (the main path). For example, the parameter *integer* is a required item in the following command:



This command might appear as:

```
SHOW DBSPACE 1
```

- Optional items appear below the main path. For example:

This statement could appear as either:

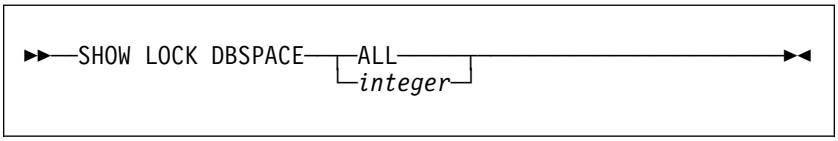
```
CREATE INDEX
```

or

```
CREATE UNIQUE INDEX
```

- If you can choose from two or more items, they appear vertically in a stack.

  If you must choose one of the items, one item appears on the main path. For example:



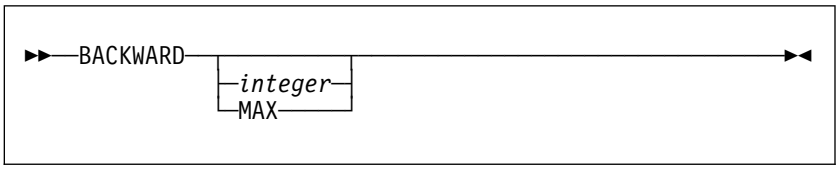
Here, the command could be either:

```
SHOW LOCK DBSPACE ALL
```

or

```
SHOW LOCK DBSPACE 1
```

If choosing one of the items is optional, the entire stack appears below the main path. For example:



Here, the command could be:

```
BACKWARD
```

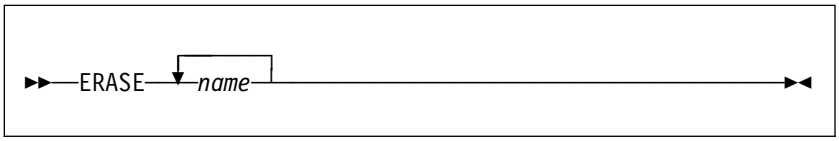
or

```
BACKWARD 2
```

or

```
BACKWARD MAX
```

- The repeat symbol indicates that an item can be repeated. For example:



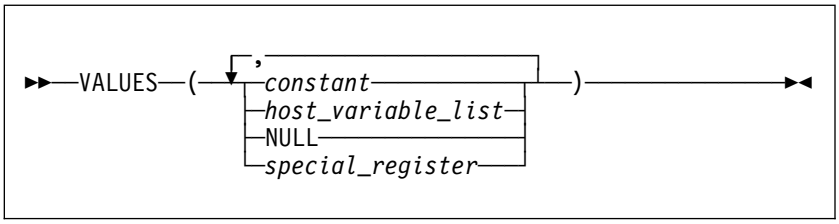
This statement could appear as:

```
ERASE NAME1
```

or

```
ERASE NAME1 NAME2
```

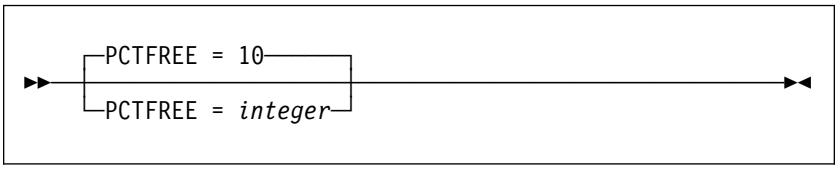
A repeat symbol above a stack indicates that you can make more than one choice from the stacked items, or repeat a choice. For example:



- If an item is above the main line, it represents a default, which means that it will be used if no other item is specified. In the following example, the ASC keyword appears above the line in a stack with DESC. If neither of these values is specified, the command would be processed with option ASC.

- When an optional keyword is followed on the same path by an optional default parameter, the default parameter is assumed if the keyword is not entered. However, if this keyword is entered, one of its associated optional parameters must also be specified.

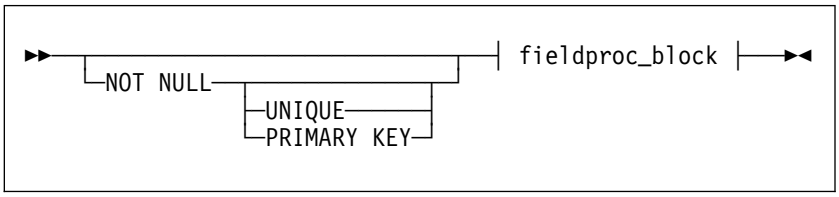
  In the following example, if you enter the optional keyword PCTFREE =, you also have to specify one of its associated optional parameters. If you do not enter PCTFREE =, the database manager will set it to the default value of 10.

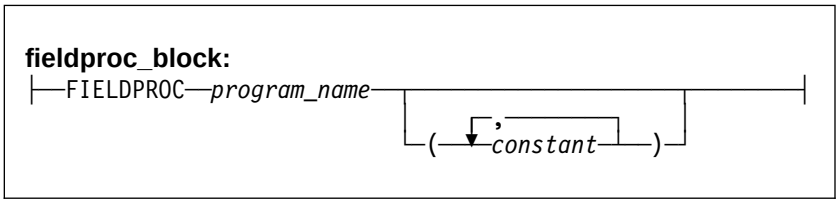


- Words that are only used for readability and have no effect on the execution of the statement are shown as a single uppercase default. For example:

  Here, specifying either REVOKE ALL or REVOKE ALL PRIVILEGES means the same thing.

- Sometimes a single parameter represents a fragment of syntax that is expanded below. In the following example, **fieldproc_block** is such a fragment and it is expanded following the syntax diagram containing it.





Conventions for Representing Mixed Data Values

When mixed data values are shown in examples, the following conventions are used:

Convention	Meaning
<	Represents the mixed shift-out character (X'0E').
>	Represents the mixed shift-in character (X'0F').
x	Represents an SBCS character (x can be any lowercase letter).
XX	Represents a DBCS character (**XX** can be any double-byte uppercase letter).

Short Forms Used in Syntax Diagrams

Some words have been shortened in the syntax diagrams in this book. The words are:

Full Word	Short Form
duration	dur
expressions	exp
string	str

DB2 Language Elements

Primitive Elements

character	A letter, digit, space, or special-character
letter	The letters a to z, A to Z, or national language extender (# @ $), or as specified in SYSCHARSETS
digit	The digits 0 to 9
space	The space character
special-character	Any element in a character set other than a letter, digit, or space
hexadecimal-character	A pair of characters in the range 00 to FF
double-byte-character	A character that occupies 2 bytes.

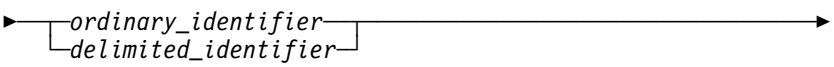
SQL Comments

An SQL comment is all text following two consecutive hyphens (--) on the same line of a static SQL statement in an application program or the command portion of a DBS Utility command.

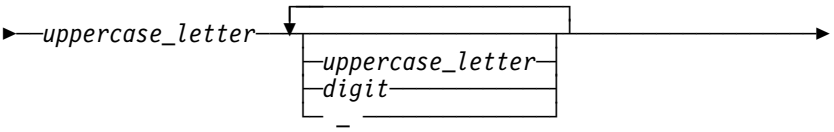
Comments are allowed wherever a separator (space character) is valid.

Identifiers

identifier

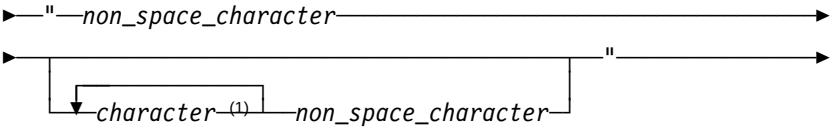


ordinary_identifier



Note:
A reserved word cannot be used as an ordinary identifier. For a list of SQL reserved words, see “SQL Reserved Words” on page 104.

delimited_identifier



Note:
1 With the exception of ".

long_identifier An identifier with a maximum length of 18 characters (not including any quotation marks).

short_identifier An identifier with a maximum length of 8 characters (not including any quotation marks).

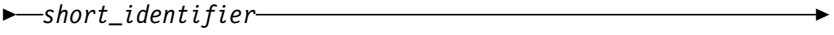
host_identifier As defined by the host language, has a maximum length imposed by the host language.

Names and Other Metavariables

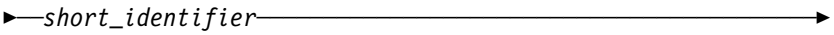
A metavariable (or parameter) is a lowercase character or group of characters used in syntax diagrams to represent a group of variables.

authorization_name

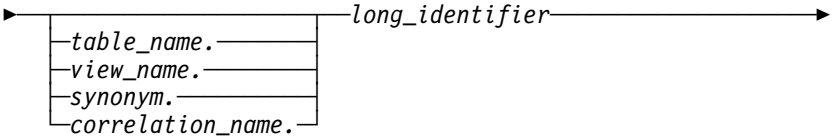
With a VSE system, authorization names and passwords are limited to 8 characters and cannot have embedded blanks.



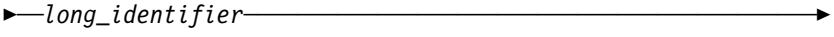
collection_id



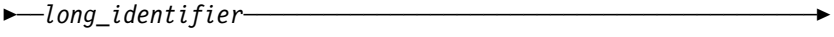
column_name



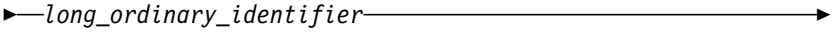
constraint_name



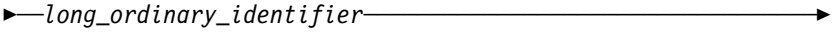
correlation_name



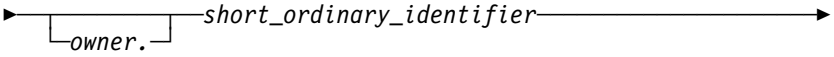
cursor_name



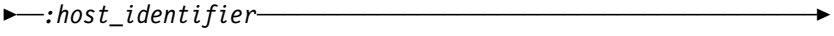
cursor_variable



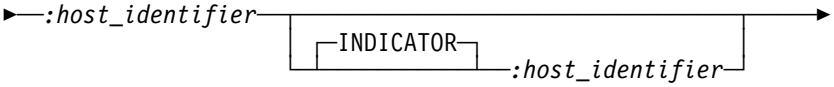
dbspace_name



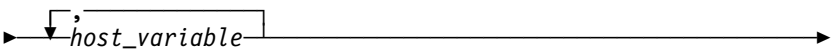
descriptor_name



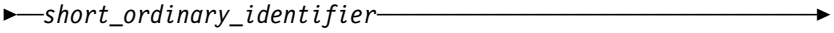
host_variable



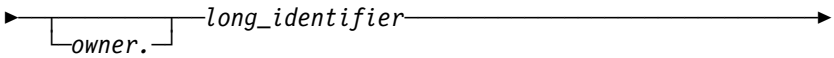
host_variable_list



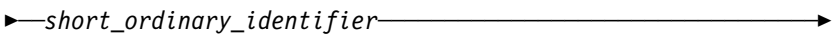
index_id



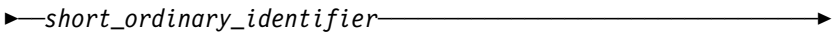
index_name



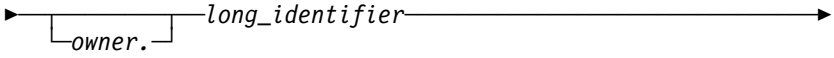
owner_name



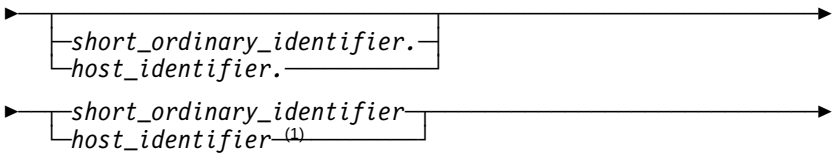
package_id



package_name

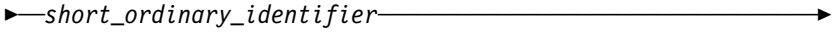


package_spec

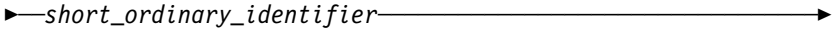


Note:
[1] Cannot be a qualified subfield name.

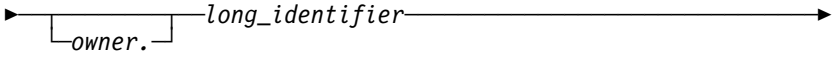
password



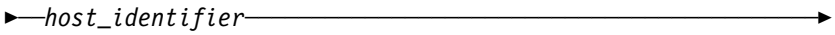
program_name



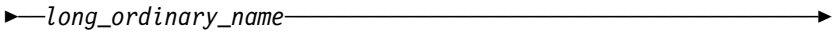
routine_name



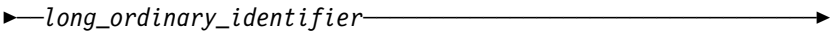
section_variable



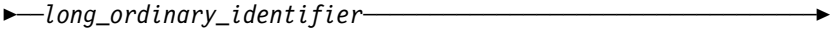
server_name



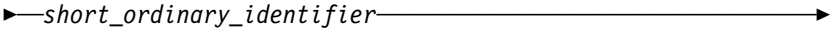
statement_name



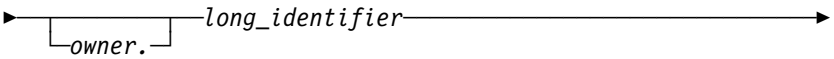
statement_variable



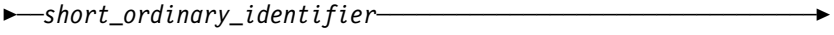
subsystemid



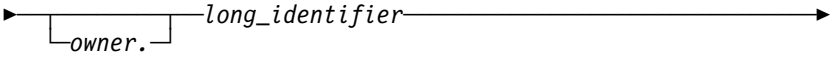
synonym



table_id

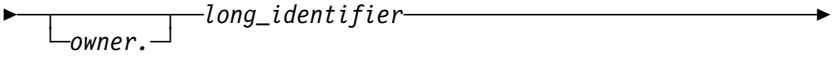


table_name



view_id

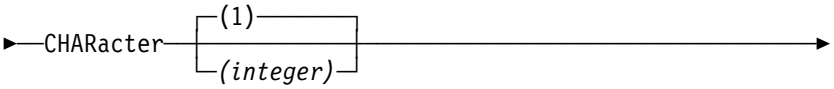
view_name



Data Types

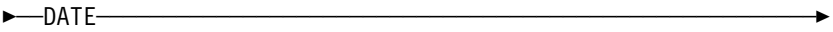
CHARacter

For character data that has a fixed number of characters (*integers*). The maximum number of characters is 254.



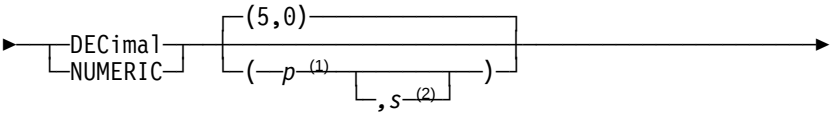
DATE

A three-part value that designates a point in time according to the Gregorian calendar. Internally represented as 4-byte packed decimal. The three parts are the year, month, and day. The date can be formatted in several ways. The range of year is 0001 to 9999. The range of month is 1 to 12. The range of day is 1 to n where n depends on the month.



DECimal

For decimal data. The *p* identifies the total number of decimal digits a number can have. The *s* identifies the number of digits to the right of the decimal point. For example, DECIMAL(5,2) creates a decimal column consisting of five digits, two of which are to the right of the decimal point. The NUMERIC parameter is a synonym for DECIMAL.



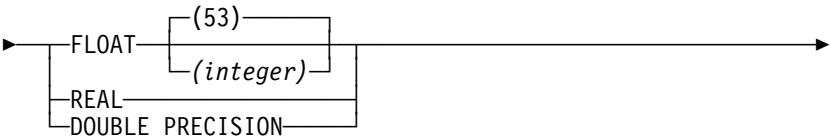
Notes:

1 The ***p*** is an integer value that defines the precision of the number.

2 The ***s*** is an integer value that defines the scale of the number.

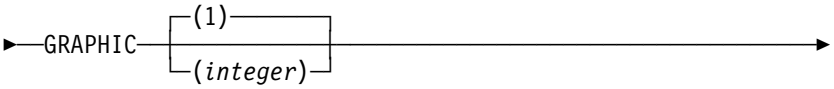
FLOAT

For floating-point numbers. Floating-point numbers range from 5.4E–79 to 7.2E+75. When *integer* is between 1 and 21, it is a single-precision floating-point number; REAL is a synonym for FLOAT in this situation. When *integer* is between 22 and 53, it is a double-precision floating-point number; DOUBLE PRECISION is a synonym for FLOAT in this situation.



GRAPHIC

For double-byte character set (DBCS) data that has a fixed number of DBCS characters *(integer)*. The maximum number of DBCS characters is 127.



INTeger

For large positive or negative whole numbers. The largest number that can be accommodated is 2147483647; the smallest number is –2147483648.

LONG VARCHAR

For character data that varies in length up to 32,767 characters.

Note:
[1] ISQL does not support INSERT, UPDATE, or SELECT of tables or views with LONG VARCHAR columns.

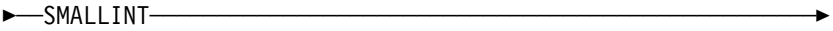
LONG VARGRAPHIC

For double-byte character set (DBCS) data that varies in length. A LONG VARGRAPHIC can be up to a maximum of 16,383 DBCS characters.

Note:
[1] ISQL does not support INSERT, UPDATE, or SELECT of tables or views with LONG VARGRAPHIC columns.

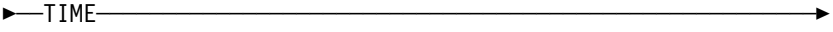
SMALLINT

For small positive or negative whole numbers. The largest number that can be accommodated is 32767; the smallest is –32768.



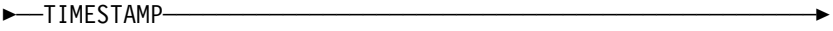
TIME

A three-part value in a number of formats that designates a time of day according to a 24-hour clock. Internally represented as 3-byte packed decimal. The three parts are the hour, minute, and second. The range of hour is 0 to 24, and the range of minute and second is 0 to 59.



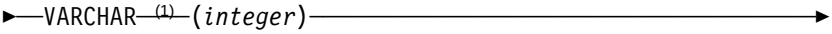
TIMESTAMP

A seven-part value that designates a date and time, including a fractional part. Internally represented as 10-byte packed decimal. The seven parts are year, month, day, hour, minute, second, and microsecond.



VARCHAR

For character data that varies in length. The *integer* refers to the maximum number of characters for any entry and can be a value up to 32767. When the value is greater than 254, the data type is considered a long string.

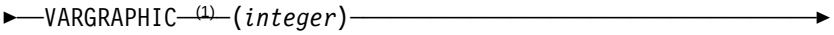


Note:

[1] ISQL does not support INSERT, UPDATE, or SELECT of tables or views with VARCHAR>254.

VARGRAPHIC

For double-byte character set (DBCS) data that varies in length. The *integer* is the number of DBCS characters for any entry; the maximum is 16383. When *integer* is greater than 127, the data type is considered a long string.



Note:

1 ISQL does not support INSERT, UPDATE, or SELECT of tables or views with VARGRAPHIC>127.

String Representations of Dates and Times

Date Strings

A string representation of a date is a character string that starts with a digit and has a length of at least 8 characters.

Format Name	Abbrev.	Date Format	Example
International Standards Organization	ISO	yyyy-mm-dd	1993-12-12
IBM USA standard	USA	mm/dd/yyyy	12/12/1993
IBM European standard	EUR	dd.mm.yyyy	12.12.1993
Japanese Industrial Standard Christian Era	JIS	yyyy-mm-dd	1993-12-12
Site-defined	LOCAL	Any site-defined form	—

Time Strings

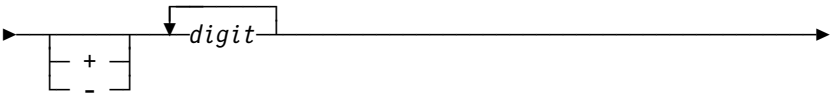
A string representation of a time is a character string that starts with a digit and has a length of at least 4 characters.

Format Name	Abbrev.	Time Format	Example
International Standards Organization	ISO	hh.mm.ss	13.30.05
IBM USA standard	USA	hh:mm AM or PM	1:30 PM
IBM European standard	EUR	hh.mm.ss	13.30.05
Japanese Industrial Standard Christian Era	JIS	hh:mm:ss	13:30:05

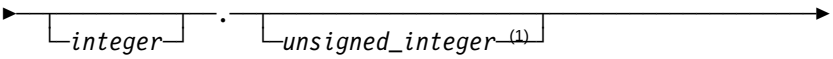
Format Name	Abbrev.	Time Format	Example
Site-defined	LOCAL	Any site-defined form	—

Constants

Integer Constant



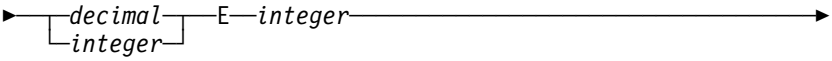
Decimal Constant



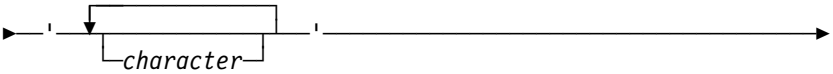
Note:

1 At least one number is needed with the decimal point.

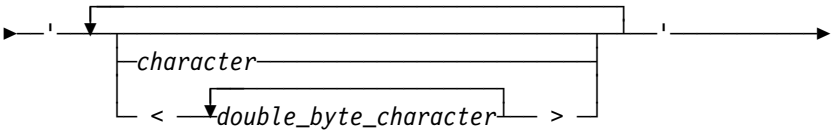
Floating-Point Constant



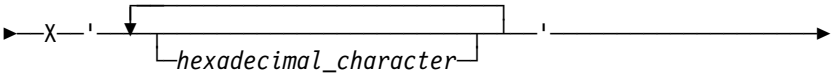
Character Constant - SBCS



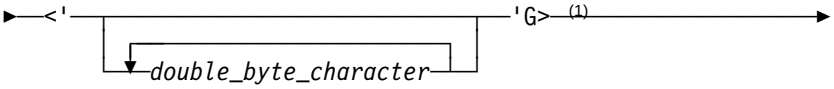
Character Constant - MIXED



Character Constant - Hexadecimal

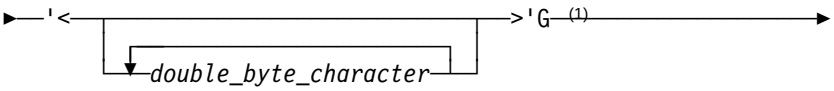


Graphic Constant - in PL/I Programs



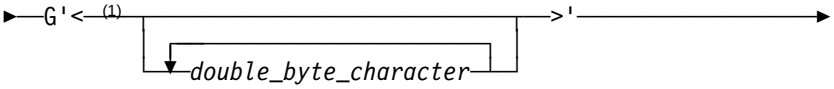
Note:
[1] N is a synonym for G.

or



Note:
[1] N is a synonym for G.

Graphic Constant - In All Other Contexts



Note:
[1] N is a synonym for G.

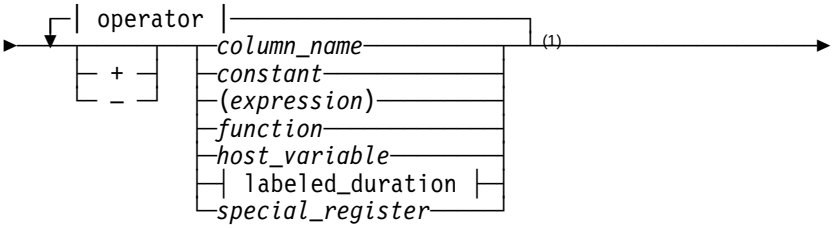
Special Registers

The following special registers are supported by the database manager.

Special Registers	Description
USER	The runtime authorization ID
CURRENT DATE	The current date in the local time zone
CURRENT SERVER	The current application server
CURRENT TIME	The current time in the local time zone
CURRENT TIMESTAMP	The current timestamp in the local time zone
CURRENT TIMEZONE	A signed time duration as a DECIMAL(6,0) number containing the local time-zone value.

Expressions

An expression specifies a value. The form of an expression is as follows:



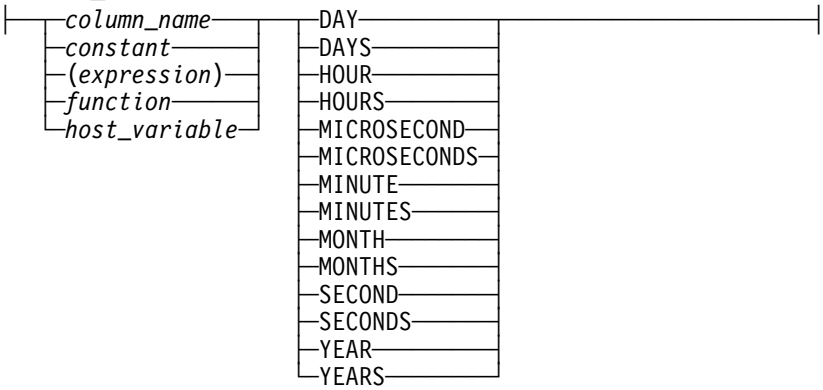
operator:

Notes:

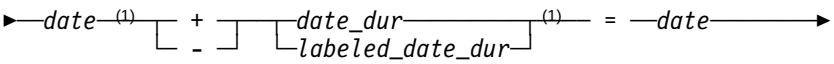
1 Not all combinations of operands and operations are supported.

2 Either || or !! can be used as a synonym for CONCAT.

labeled_duration:

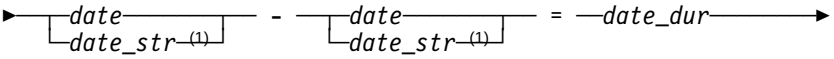


Date Arithmetic



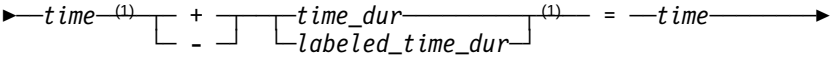
Note:

1 These operands can be specified in either order.

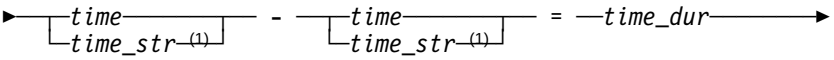


Note:
[1] Only one of these two operands can be a string.

Time Arithmetic

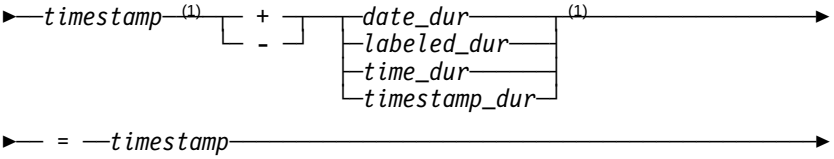


Note:
[1] These operands can be specified in either order.

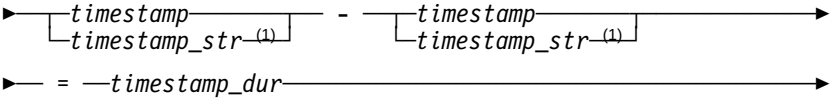


Note:
[1] Only one of these two operands can be a string.

Timestamp Arithmetic



Note:
[1] These operands can be specified in either order.

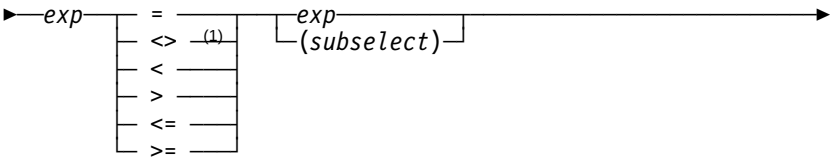


Note:
[1] Only one of these two operands can be a string.

Predicate

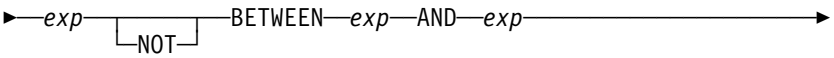
Specifies a condition that is true, false, or unknown about a row or group.

Basic Predicate

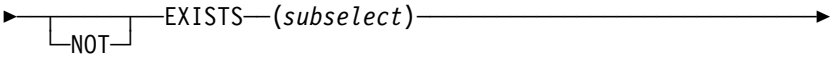


Note:
[1] Either ¬= or ^= can be used as an alternative to the <> operator.

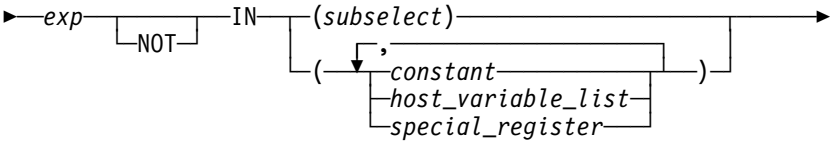
BETWEEN Predicate



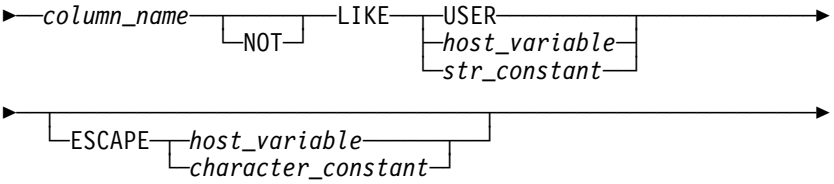
EXISTS Predicate



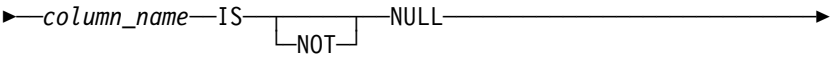
IN Predicate



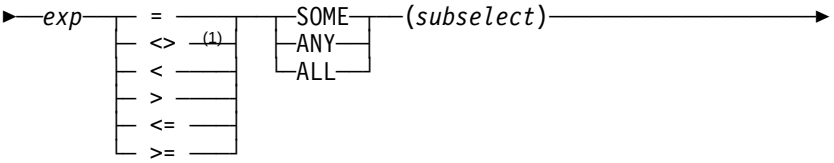
LIKE Predicate



NULL Predicate



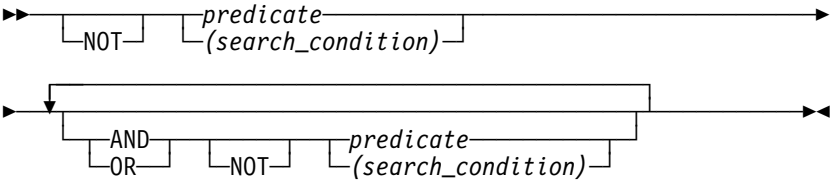
Quantified Predicate



Note:
[1] Either ¬= or ^= can be used as an alternative to the <> operator.

Search Conditions

Specifies a condition that is true, false, or unknown about a row or group. The common form of a search condition is column_name operator value. Refer to “Predicate” on page 15 for additional functions available for search conditions.



Functions

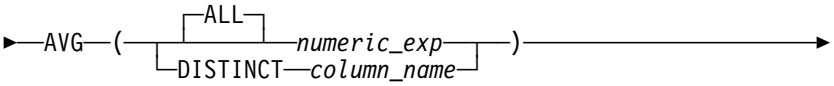
The two kinds of functions are column and scalar.

Column Functions

Produce a value from an argument having a collection of values that are derived from one or more columns.

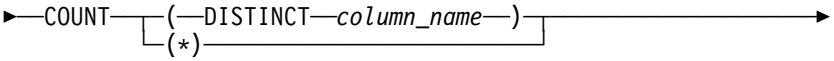
AVG

Calculates the average of a group of column values.



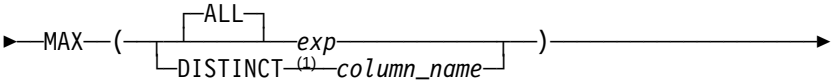
COUNT

Obtains the number of rows or distinct values in a collection of rows or column values.



MAX

Obtains the maximum value in a collection of column values.

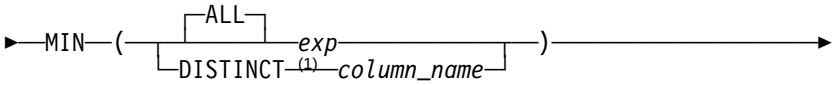


Note:

[1] Although the keyword DISTINCT is allowed, it does not affect the result of the function.

MIN

Obtains the minimum value in a set of column values.

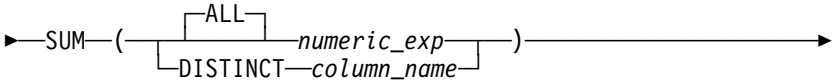


Note:

1 Although the keyword DISTINCT is allowed, it does not affect the result of the function.

SUM

Obtains the total of all values in a group.

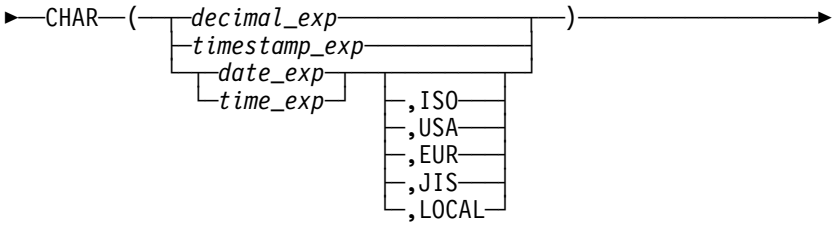


Scalar Functions

Produce a single value from an argument having a single value.

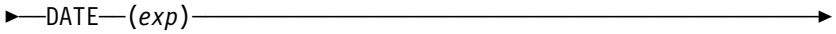
CHAR

Creates a character representation of certain noncharacter data types.



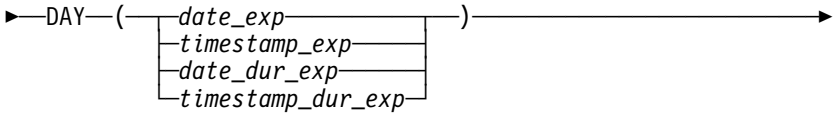
DATE

Creates a date from an expression or timestamp.



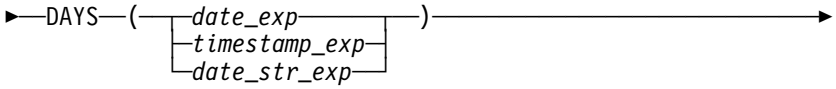
DAY

Extracts the day part of a value.



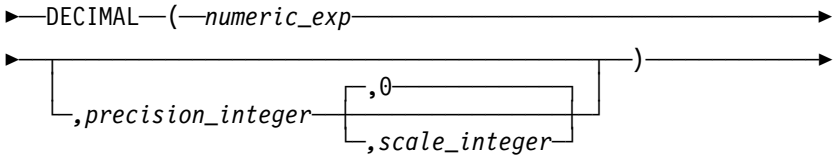
DAYS

Extracts an integer representation of a date. The result is one more than the number of days from January 1, 0001, to the date.



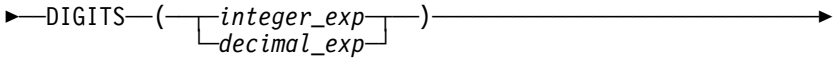
DECIMAL

Returns a decimal representation of a numeric value.



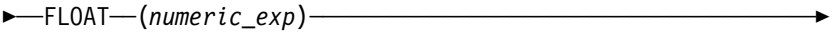
DIGITS

Returns a character string representation of a number without a sign or decimal point.



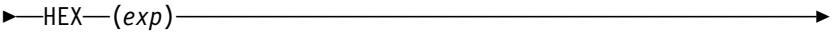
FLOAT

Returns a floating-point representation of a number.



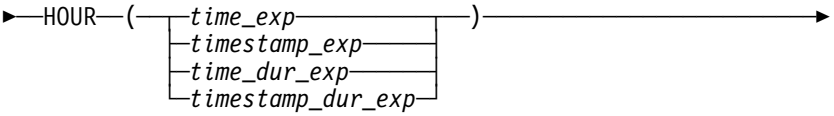
HEX

Returns a hexadecimal representation of a value.



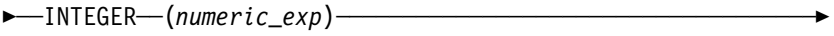
HOUR

Extracts the hour part of a value.



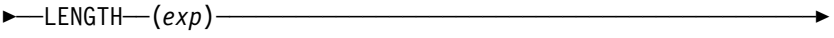
INTEGER

Returns an integer representation of a number.



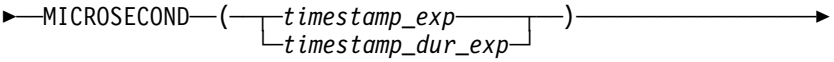
LENGTH

Returns the length of a value.



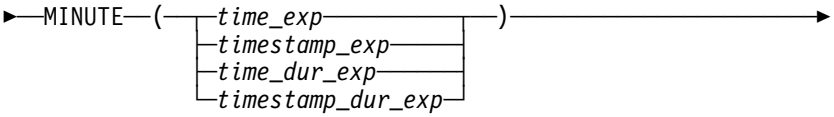
MICROSECOND

Extracts the microsecond part of a value.



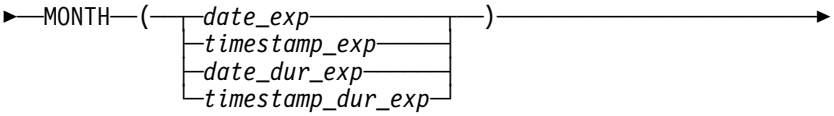
MINUTE

Extracts the minute part of a value.



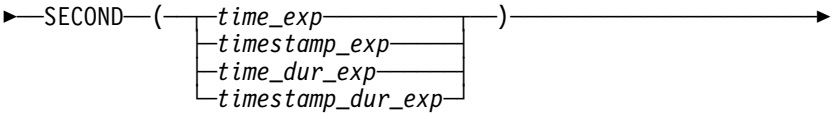
MONTH

Extracts the month part of a value.



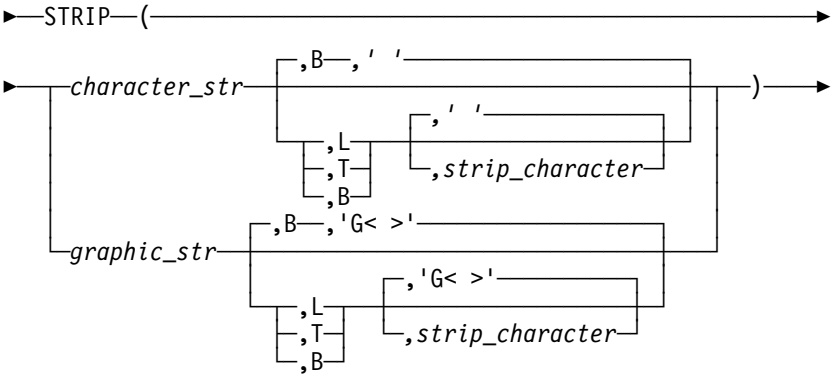
SECOND

Extracts the seconds part of a value.



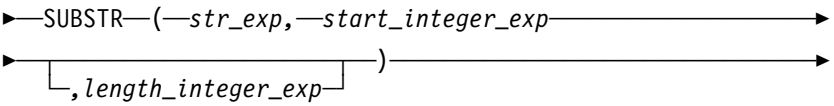
STRIP

Removes blanks or another specified character from the end or the beginning of a string.



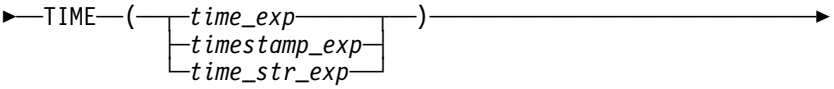
SUBSTR

Returns a part (substring) of string as indicated by the length and starting position.



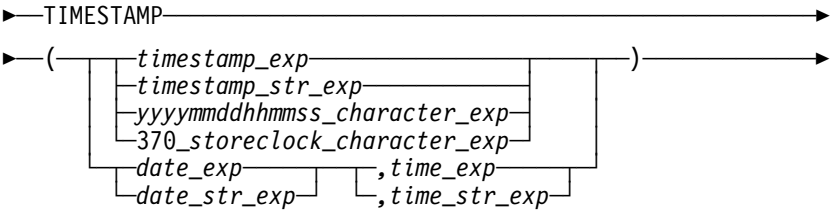
TIME

Creates a time from a value.



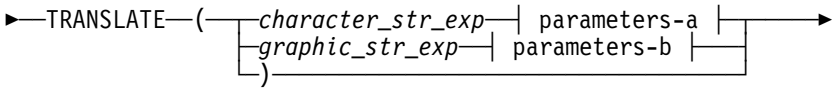
TIMESTAMP

Creates a timestamp from a value or a pair of values that represent a date and time.

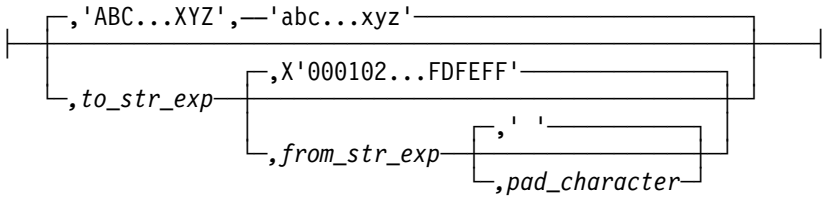


TRANSLATE

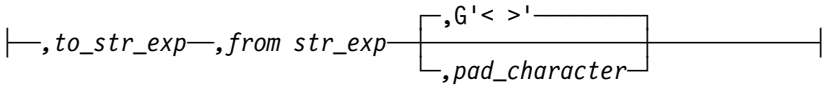
Changes one or more characters in a string expression into other characters. For example, it can be used to reorder characters in a string to uppercase.



parameters-a:

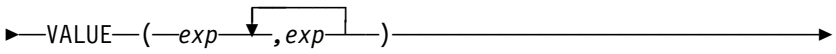


parameters-b:



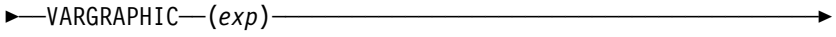
VALUE

Returns the first nonnull result in a series of SQL expressions.



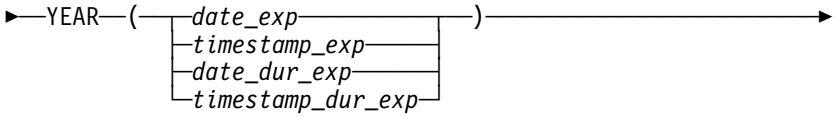
VARGRAPHIC

Returns a graphic string representation of a character string.



YEAR

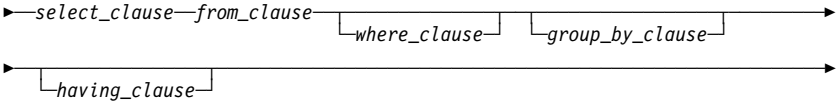
Extracts the year part of a value.



Queries

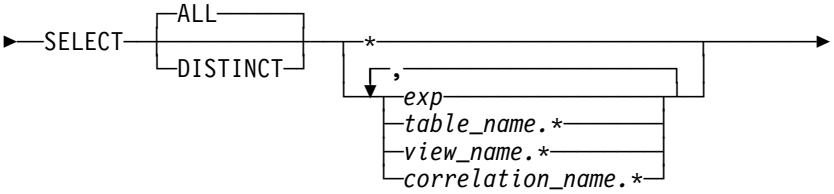
subselect

Specifies a result table derived from the tables or views identified in the FROM clause. Subselect is a component of the fullselect statement, the CREATE VIEW statement, the INSERT statement, and certain predicates.



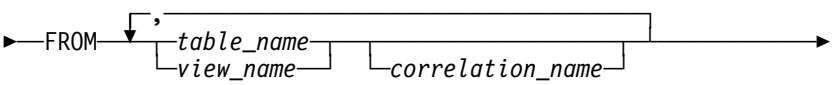
select-clause

Produces a final result table by selecting only the columns indicated by the *select list* from R, where R is the result of the previous operation. For example, if the group-by-clause and having-clause are not specified, R is the result of the where-clause.



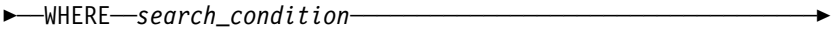
from-clause

Names a single table or view, or produces an intermediate result table. The intermediate result table contains all possible combinations of the rows of the named tables or views.



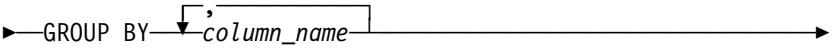
where-clause

Produces an intermediate result table by applying *search-condition* to each row of R, where R is the result of the FROM clause. The result table contains the rows of R for which the *search-condition* is true.



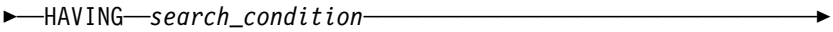
group-by-clause

Produces an intermediate result table by grouping the rows of R, where R is the result of the previous clause.



having-clause

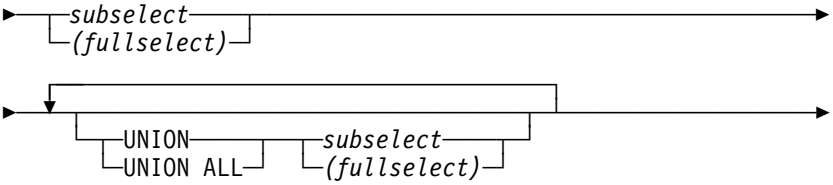
Produces an intermediate result table by applying *search-condition* to each group of R, where R is the result of the previous clause.



---

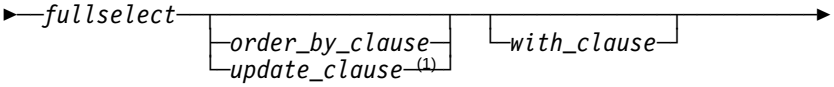
fullselect

Specifies a result table. If UNION is not used, the result of the fullselect is the result of the specified subselect.



select-statement

Is the form of a query that can be specified or referenced in a DECLARE CURSOR statement and in the interactive select statement.

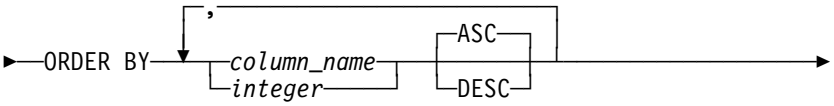


Note:

1 The interactive select-statement does not incorporate the update-clause. That clause cannot be issued in ISQL or in the DBS Utility.

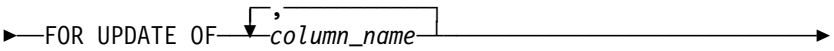
order-by-clause

Orders the rows of the result table by the values of the identified columns.



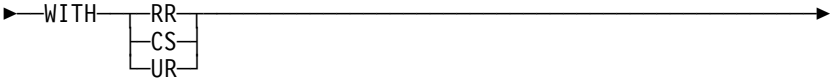
update-clause

Refers to the cursor in a positioned UPDATE statement. The UPDATE statement can update only columns in the *column-name* list. This update-clause cannot be used interactively.



with-clause

Specifies the isolation level at which the statement is executed.



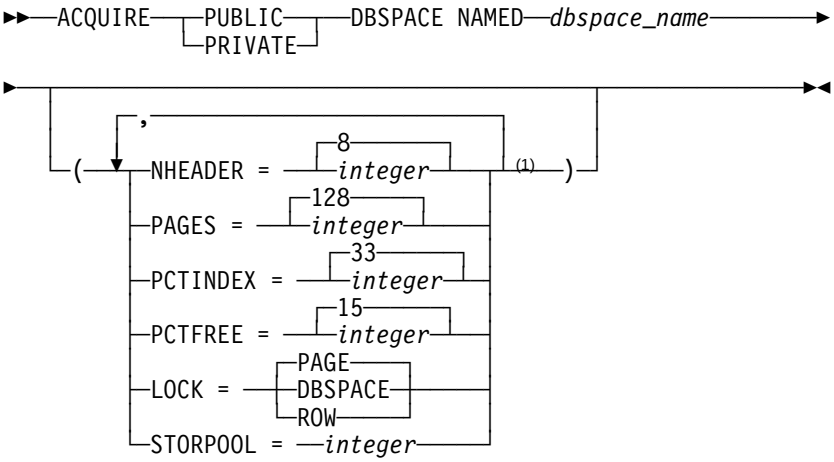
SQL Statements

Invocation

The letters **I** and **P**, printed to the right of the statement, indicate where each statement can be used. The **I** indicates the statement can be issued interactively, and the **P** indicates the statement can be embedded in an application program.

ACQUIRE DBSPACE (I,P)

Finds and names an available dbspace.

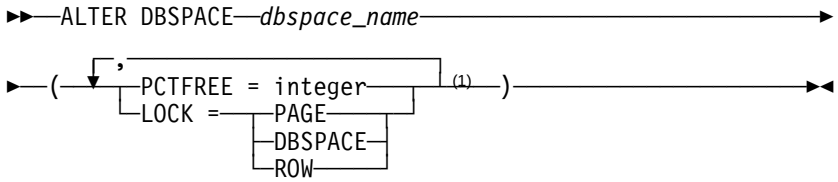


Note:

1 If any of these clauses is specified more than once, the value with the first specification is used.

ALTER DBSPACE (I,P)

Changes the percentage of free space and the type of locking of a PUBLIC dbspace.

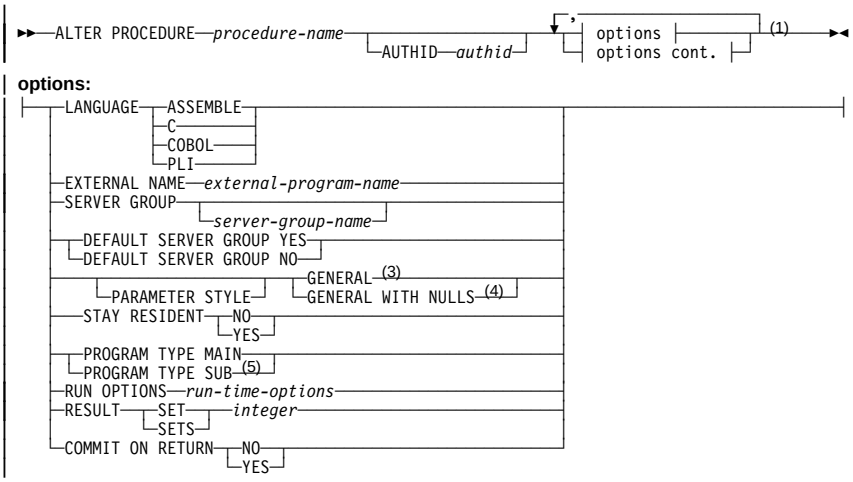


Note:

1 If either of these clauses is specified more than once, the value with the first specification is used.

ALTER PROCEDURE (I,P)

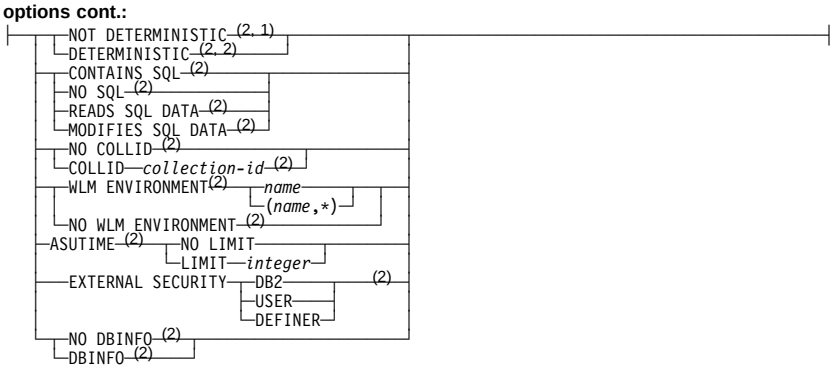
Alters the definition of a stored procedure.



Notes:

1. One or more clauses may be specified, however each clause may be specified at most once.
2. This parameter is included for compatibility with the DB2 family. If specified, it is ignored.
3. SIMPLE CALL may be used as an alternative to GENERAL. This is for compatibility within the DB2 family.
4. SIMPLE CALL WITH NULLS may be used as an alternative to GENERAL WITH NULLS. This is for compatibility within the DB2 family.
5. Currently, DB2 Server for VSE & VM supports stored procedures written as main programs only.

options cont.:



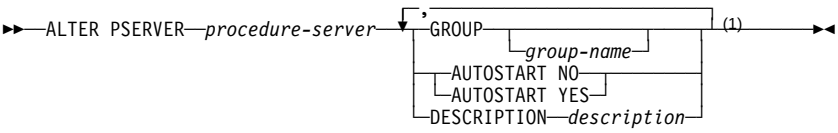
Notes:

1 VARIANT may be specified as an alternative to NOT DETERMINISTIC. This is for compatibility within the DB2 family.

2 NOT VARIANT may be specified as an alternative to DETERMINISTIC. This is for compatibility within the DB2 family.

ALTER PSERVER (I,P)

Alters the definition of a stored procedure server.

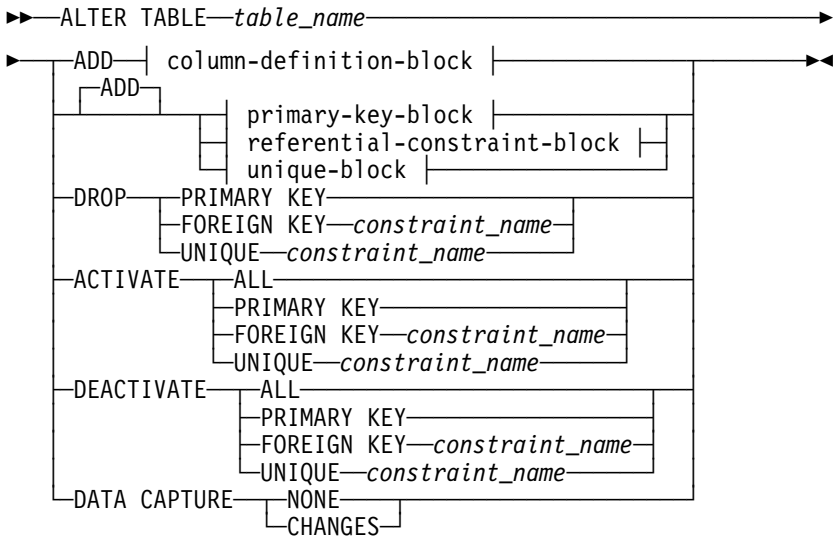


Note:

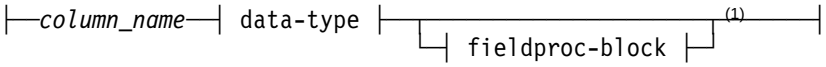
1 One or more clauses may be specified, however each clause may be specified at most once.

ALTER TABLE (I,P)

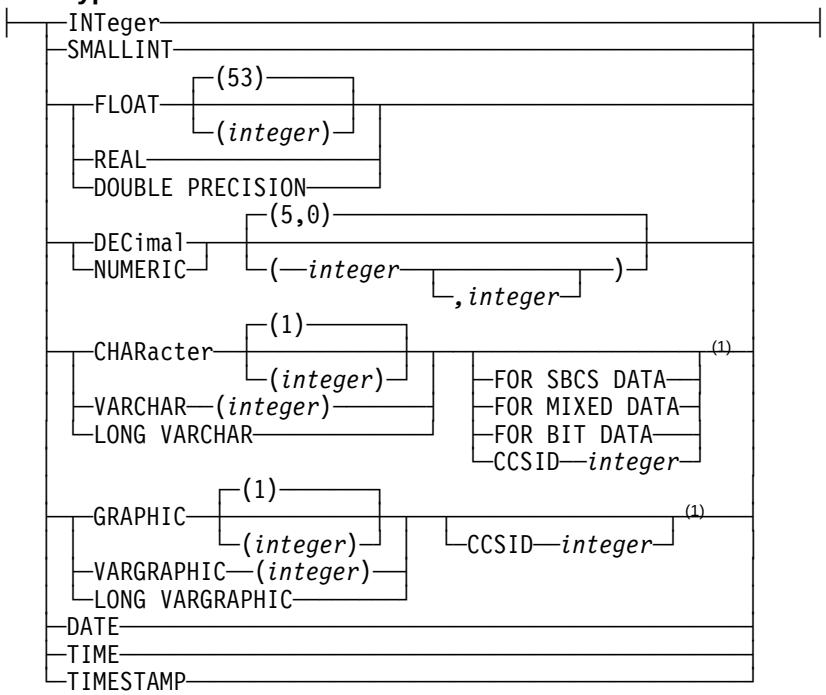
Adds a new column, or adds, drops, activates, or deactivates a primary key, foreign key, or unique constraint on a specified table.


ALTER TABLE
table_name
ADD
column-definition-block
ADD
primary-key-block
referential-constraint-block
unique-block
DROP
PRIMARY KEY
FOREIGN KEY
constraint_name
UNIQUE
constraint_name
ACTIVATE
ALL
PRIMARY KEY
FOREIGN KEY
constraint_name
UNIQUE
constraint_name
DEACTIVATE
ALL
PRIMARY KEY
FOREIGN KEY
constraint_name
UNIQUE
constraint_name
DATA CAPTURE
NONE
CHANGES

column-definition-block:



data-type:

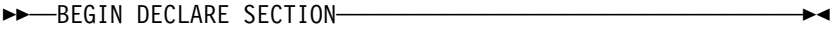


Note:

[1] These clauses may be specified in any order.

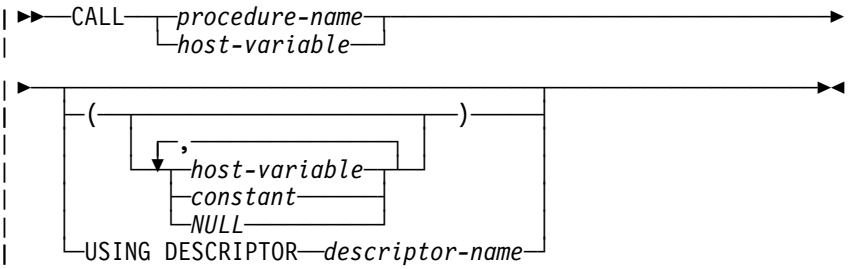
BEGIN DECLARE SECTION (P)

Marks the beginning of a host variable declare section, including host structures.



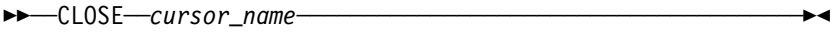
CALL (P)

Invokes a stored procedure with a list of input/output parameters.



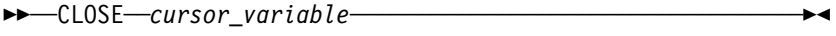
CLOSE (P)

Closes the cursor identified by *cursor-name*.



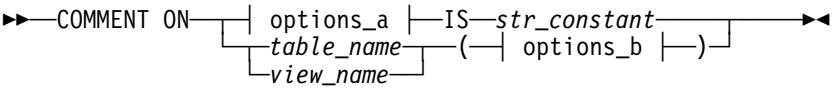
Extended CLOSE (P)

Closes the cursor identified by *cursor-variable*.

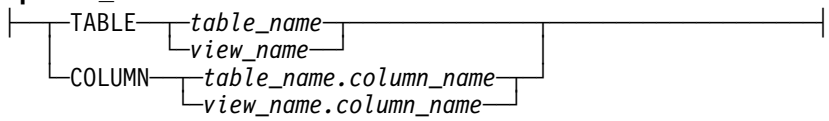


COMMENT ON (I,P)

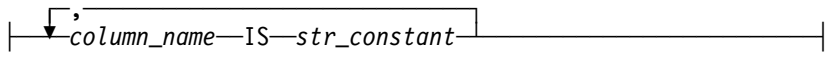
Adds or replaces comments in the catalog descriptions of tables, views, or columns.



options_a:

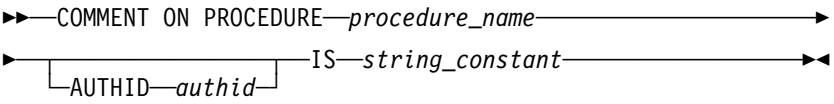


options_b:



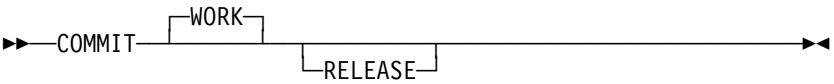
COMMENT ON PROCEDURE (I,P)

Adds or replaces comments in the catalog descriptions of stored procedures.



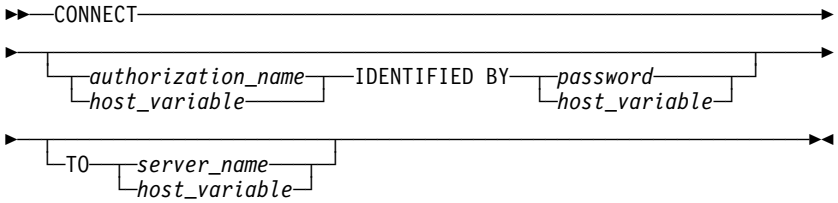
COMMIT (I,P)

Ends the current logical unit of work and commits any changes.



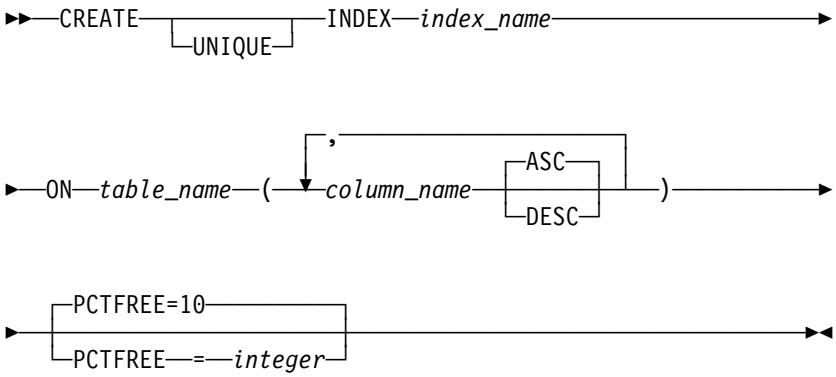
CONNECT (I,P)

Connects an application process or a user, or both, to an application server.



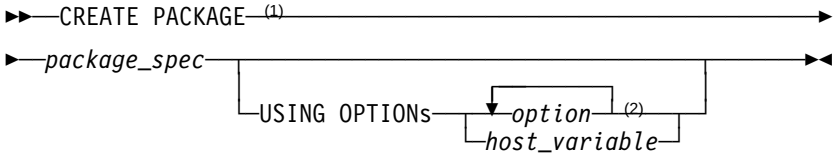
CREATE INDEX (I,P)

Creates an index on one or more columns of a table.



CREATE PACKAGE (P)

Creates a package.



Notes:

1 PROGRAM is equivalent to PACKAGE, and is provided for compatibility with some older versions of the SQL/DS product.

2 An option may be specified only once.

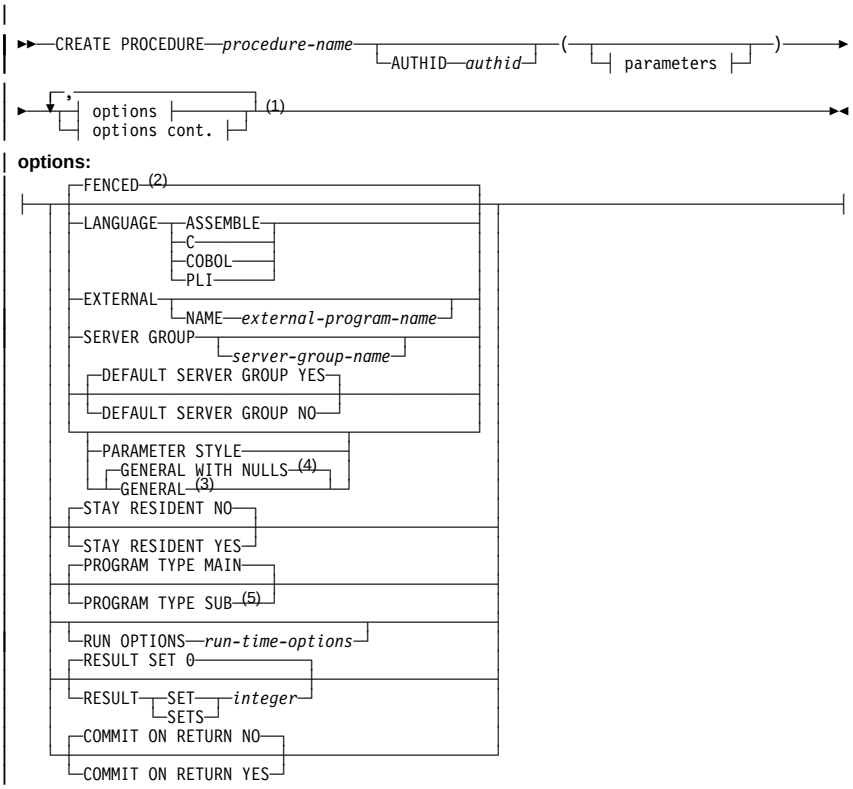
Using Options

Table 1.

CCSIDSbcs (integer)	NOCHECK | CHECK | ERROR
CCSIDMixed (integer)	NODESCRIBE | DESCRIBE
CCSIDGraphic (integer)	NOEXIST | EXIST
CHARSUB (Sbcs | Mixed | Bit)	NOMODIFY | MODIFY
DATE (ISO | USA | EUR | JIS | LOCAL)	OWner (authorization-name)
EXPLAIN (NO | YES)	QUALifier (collection-id)
ISOLation (RR | RS | CS | UR | USER)	RELease (COMMIT | DEALLOCATE)
KEEP | REVOKE	REPLACE | NEW
LABEL (label-text)	TIME (ISO | USA | EUR | JIS | LOCAL)
NOBLock | BLock | SBLock	

CREATE PROCEDURE (I,P)

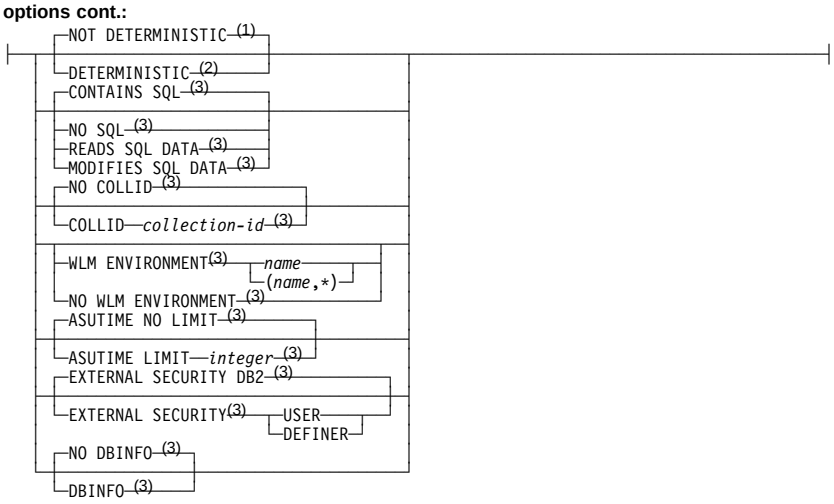
Defines a stored procedure.



Notes:

1. One or more clauses may be specified, however each clause may be specified at most once.
2. This parameter is included for compatibility with the DB2 family. If specified, it is ignored.
3. As an alternative to GENERAL, SIMPLE CALL may be used. This is for compatibility within the DB2 family.
4. As an alternative to GENERAL WITH NULLS, SIMPLE CALL WITH NULLS may be used. This is for compatibility within the DB2 family.
5. Currently, DB2 Server for VSE & VM supports stored procedures written as main programs only.

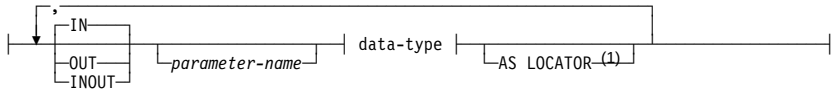
options cont.:



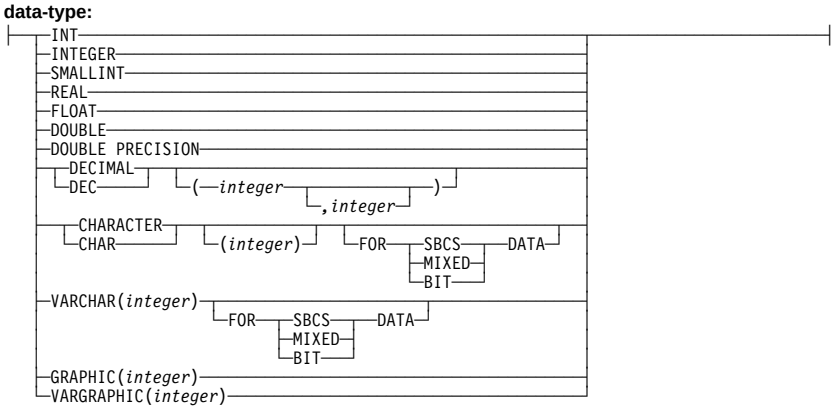
Notes:

1. VARIANT may be specified as an alternative to NOT DETERMINISTIC. This is for compatibility within the DB2 family.
2. NOT VARIANT may be specified as an alternative to DETERMINISTIC. This is for compatibility within the DB2 family.
3. This parameter is included for compatibility with the DB2 family. If specified, it is ignored.

parameters:



data-type:

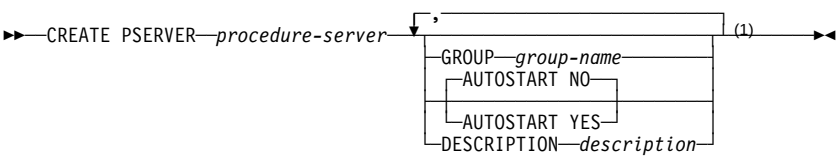


Note:

1 This parameter is included for compatibility with the DB2 family. If specified, it is ignored.

CREATE PSERVER (I,P)

Defines a stored procedure server.

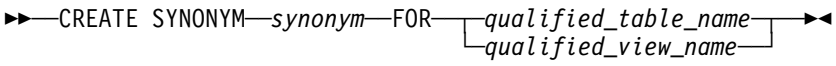


Note:

1 One or more clauses may be specified, however each clause may be specified at most once.

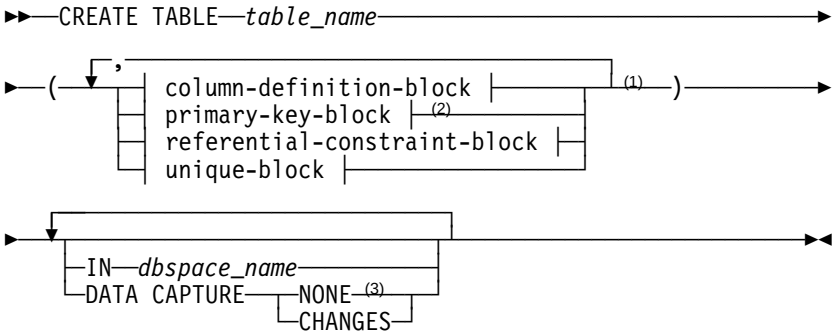
CREATE SYNONYM (I,P)

Defines an alternative name for a table or view.



CREATE TABLE (I,P)

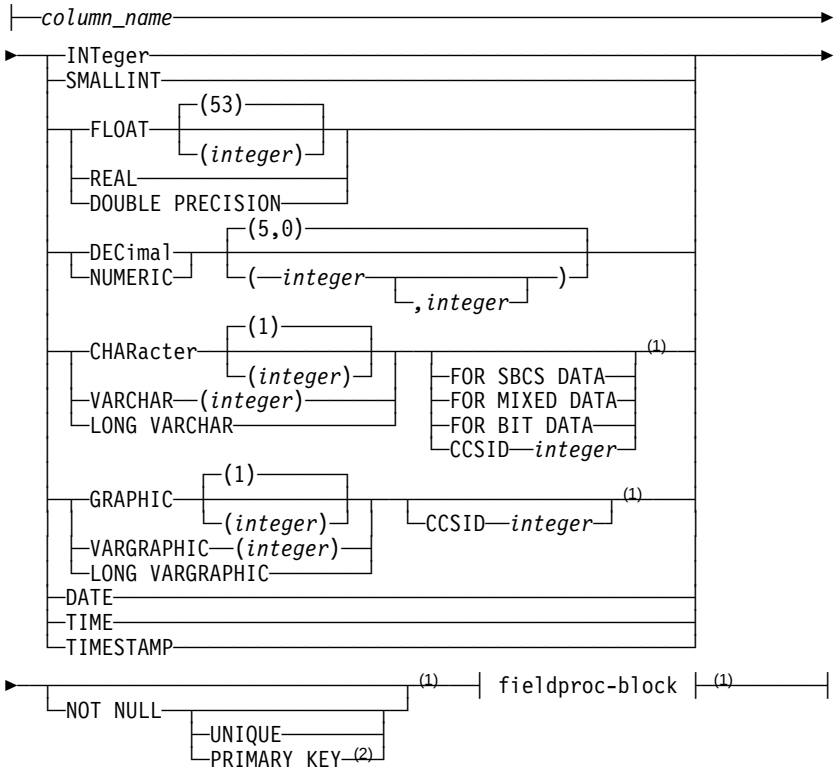
Creates a new table.



Notes:

1. There can be up to 255 columns in a table.
2. Only one primary key may be defined (either in a primary-key-block or as a column attribute).
3. The same clause must not be specified more than once.

column-definition-block:

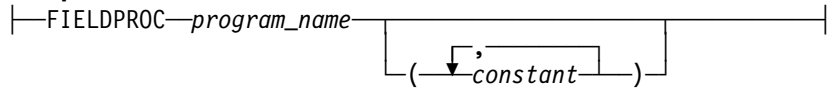


Notes:

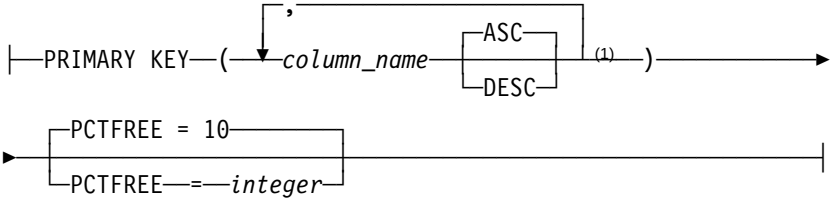
1 These clauses may be specified in any order.

2 Only one primary key may be defined (either in a primary-key-block or as a column attribute).

fieldproc-block:

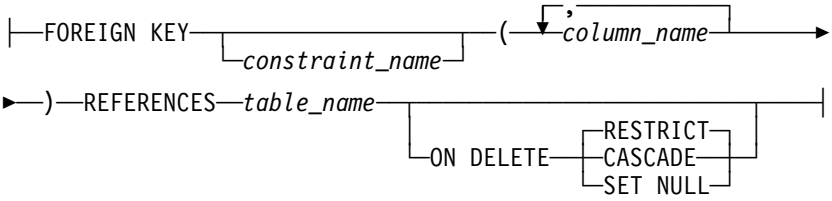


primary-key-block:

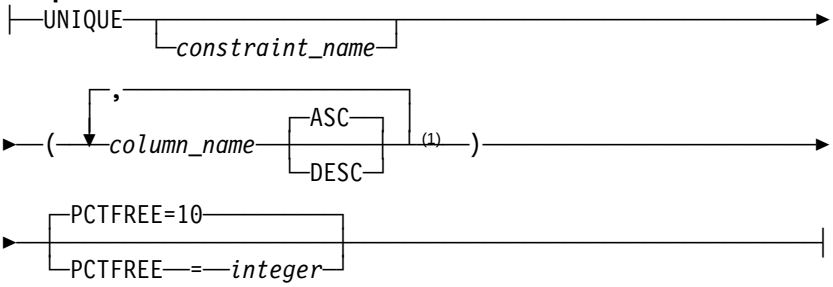


Note:
1 A PRIMARY KEY can have up to 16 columns.

referential-constraint-block:



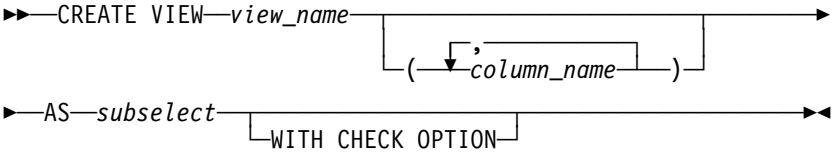
unique-block:



Note:
1 There can be up to 16 columns on a unique constraint.

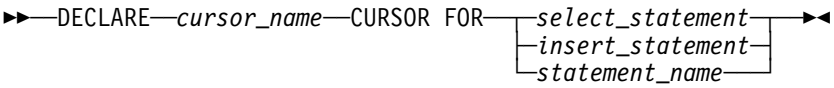
CREATE VIEW (I,P)

Creates a view on one or more tables or views.



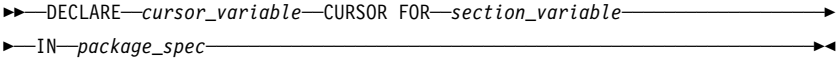
DECLARE CURSOR (P)

Declares the cursor that you can use to fetch or put the results of a prepared statement.



Extended DECLARE CURSOR (P)

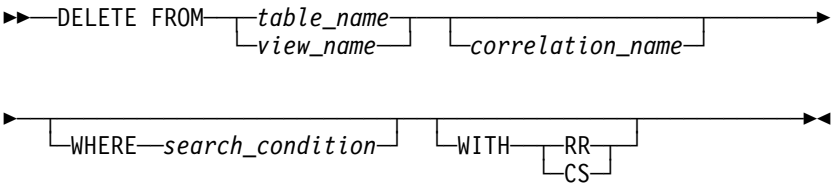
Declares the cursor that you can use to fetch or put the results of a prepared statement.



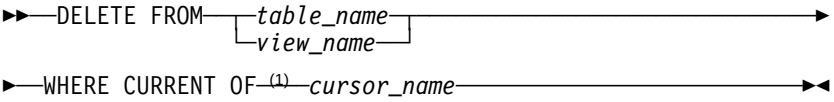
DELETE (I,P)

Deletes one or more rows from a table or view. Deleting a row from a view deletes the row from the table on which the view is based.

Searched delete (I,P)



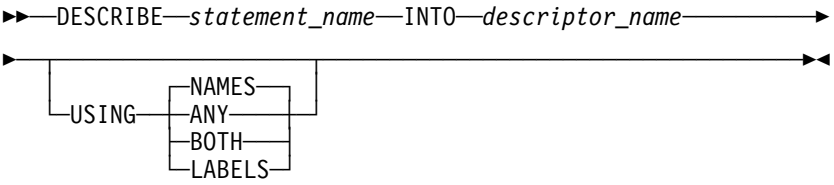
Positioned delete (P)



Note:
[1] A Positioned DELETE in FORTRAN, and programs prepared using Extended dynamic SQL cannot be used with DRDA protocol.

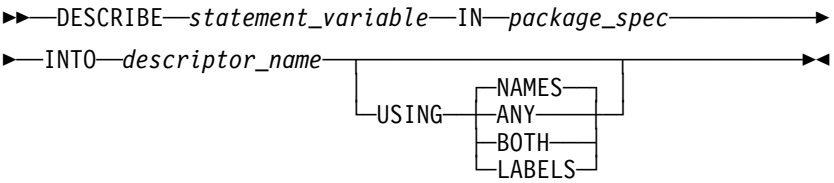
DESCRIBE (P)

Retrieves information about an SQL select-statement previously prepared with a PREPARE statement.



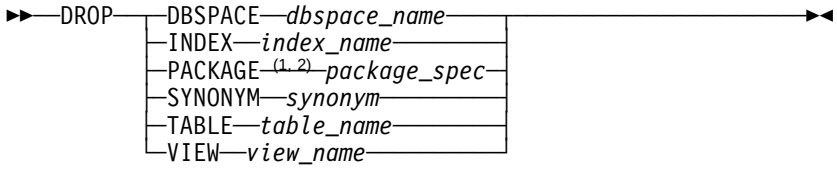
Extended DESCRIBE (P)

Retrieves information about an SQL SELECT statement previously prepared with an Extended PREPARE statement.



DROP (I,P)

Deletes an object. Objects that are directly or indirectly dependent on that object are also deleted.

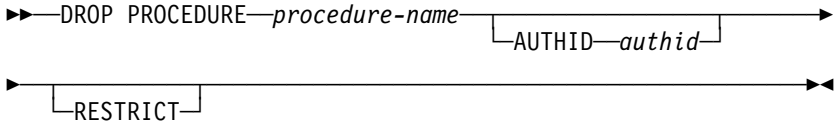


Notes:

1. PROGRAM is equivalent to PACKAGE and is provided for compatibility with prior releases of SQL/DS.
2. DROP PACKAGE cannot support a qualified structure subfield name. A host structure subfield name can be used as normal host variables, but must be unqualified. If being unqualified results in an ambiguous reference, the subfield cannot be used.

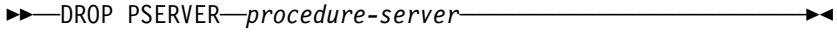
DROP PROCEDURE (I,P)

Removes a stored procedure.



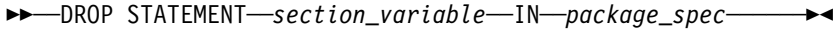
DROP PSERVER (I,P)

Removes a stored procedure pserver.



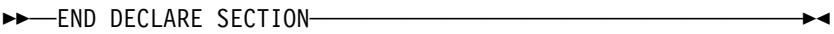
DROP STATEMENT (P)

Selectively deletes a statement from a package.



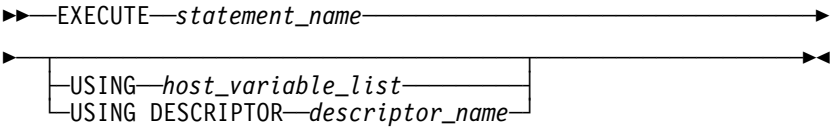
END DECLARE SECTION (P)

Marks the end of a *host-variable* declare section.



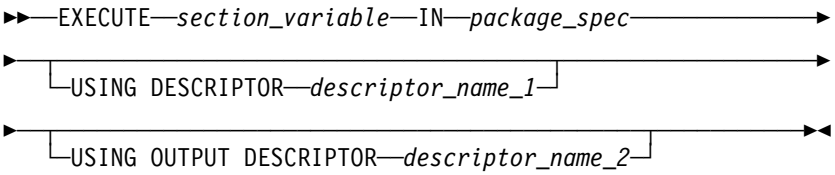
EXECUTE (P)

Executes a prepared SQL statement.



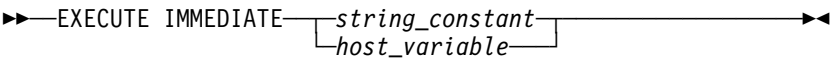
Extended EXECUTE (P)

Executes a statement previously prepared by an Extended PREPARE statement.



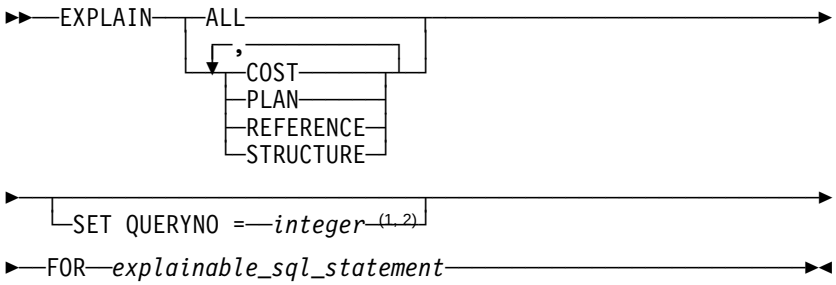
EXECUTE IMMEDIATE (P)

Prepares an executable form of an SQL statement from a character string form of the statement, executes the SQL statement, and then destroys the executable form.



EXPLAIN (I,P)

Retrieves information about the access path chosen for the execution of the SQL query, and about the structure and execution performance of a DELETE, INSERT, UPDATE or select-statement.



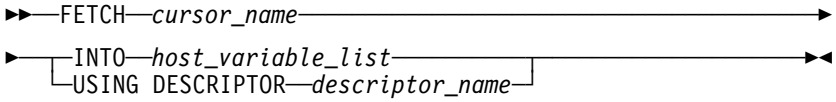
Notes:

1 The QUERYNO can be up to 2,147,483,647.

2 The integer constant must not be preceded by a sign.

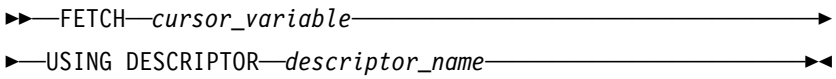
FETCH (P)

Positions a cursor on the next row of its result table and assigns the values of that row to the host variables.



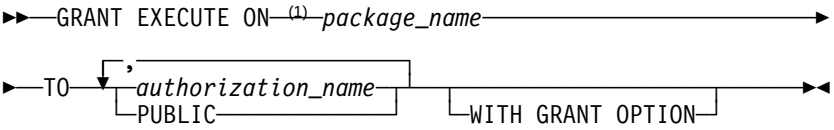
Extended FETCH (P)

Positions a cursor on the next row of its result table and assigns the values of that row to the host variables.



GRANT Package Privileges (I,P)

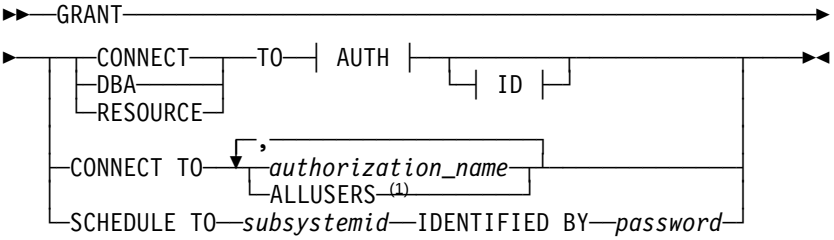
Grants the privilege to execute a package.



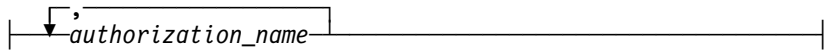
Note:
[1] RUN can be specified as a synonym for EXECUTE to support applications developed for previous releases of SQL/DS.

GRANT System Authorities (I,P)

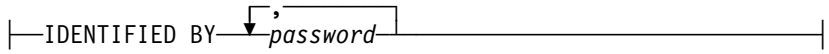
Grants authorities to users and changes passwords.



AUTH:



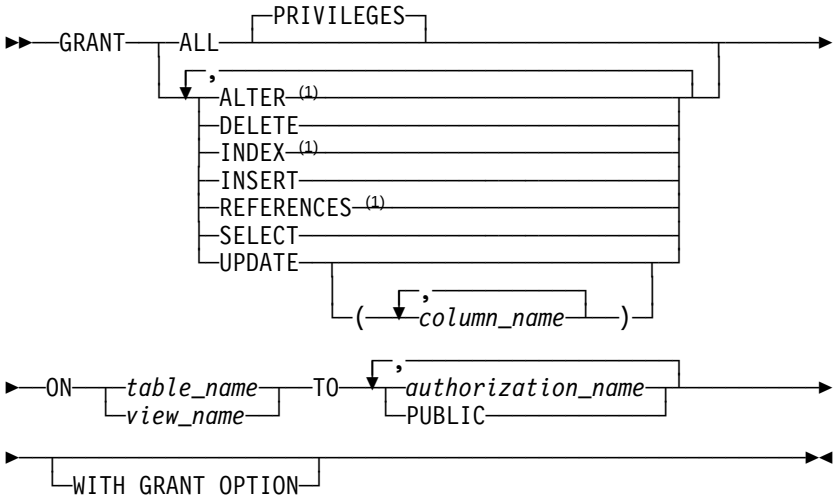
ID:



Note:
[1] ALLUSERS can only be specified once and is not applicable to a VSE application server.

GRANT Table Privileges (I,P)

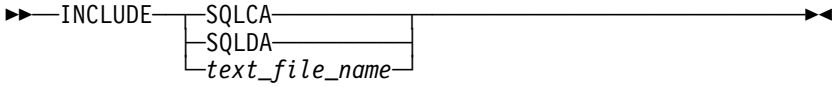
Grants privileges on a table or view.



Note:
[1] ALTER, INDEX, and REFERENCES do not apply to views.

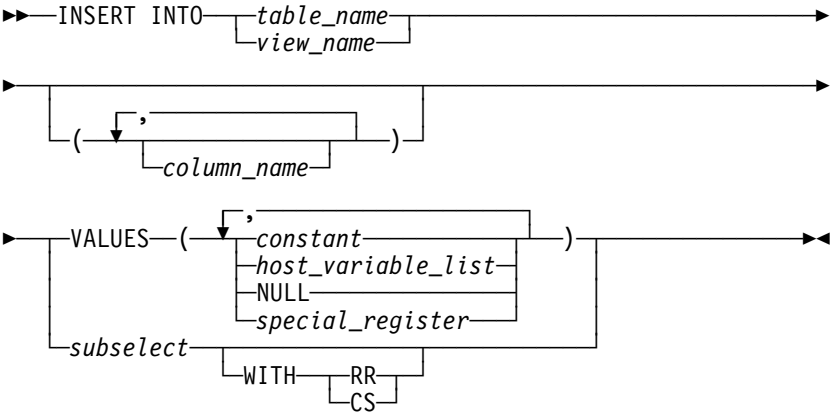
INCLUDE (P)

Inserts declarations or statements into a source program.



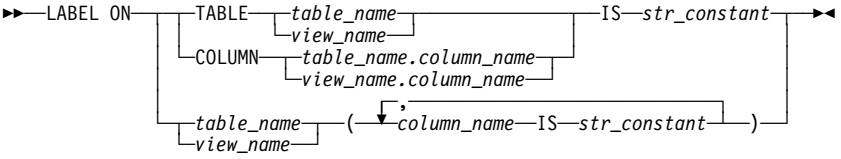
INSERT (I,P)

Inserts rows into a table or view. Inserting a row in a view inserts the row into the tables on which the view is based.



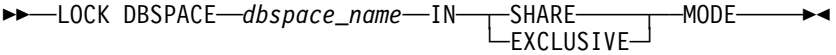
LABEL ON (I,P)

Adds or replaces labels in the catalog descriptions of tables, views, or columns.



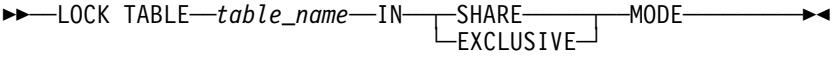
LOCK DBSPACE (I,P)

Acquires a shared or exclusive lock on the *dbspace-name* specified.



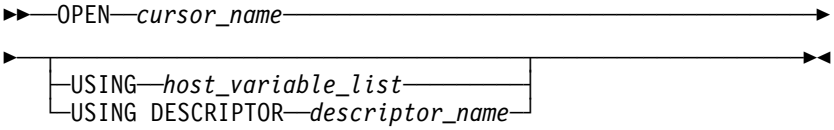
LOCK TABLE (I,P)

Acquires a shared or exclusive lock on the named table.



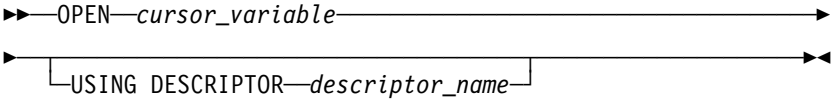
OPEN (P)

Opens a cursor.



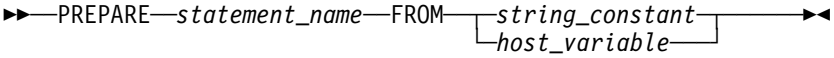
Extended OPEN CURSOR (P)

Opens a cursor.



PREPARE (P)

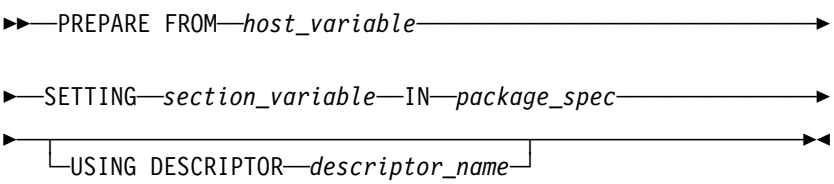
Prepares an SQL statement for execution from a character string form of the statement.



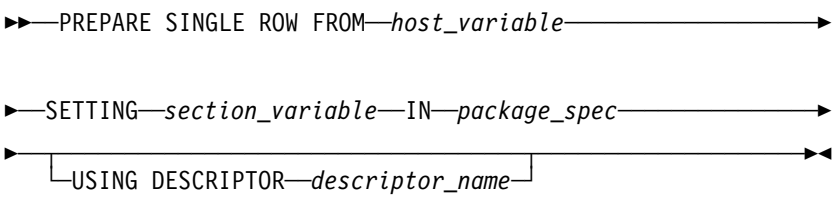
Extended PREPARE (P)

Basic Extended PREPARE

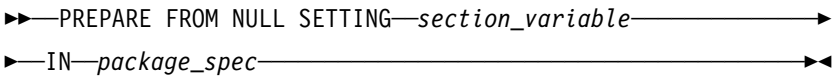
Adds an SQL statement to an existing package.



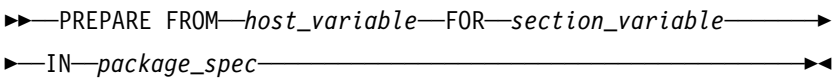
Single Row Extended PREPARE
Indicates that the SELECT statement in the *host-variable* is a single row SELECT.



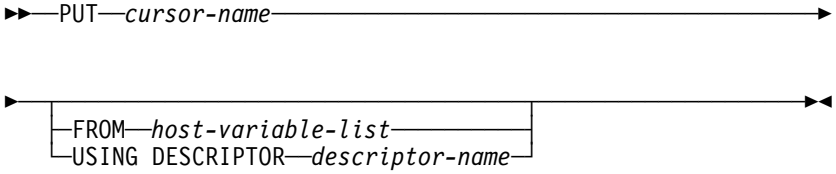
Empty Extended PREPARE
Allows for the creation of an indefinite section in a program.



Temporary Extended PREPARE
Prepares the statement in the *host-variable* and associates the output with a previously created indefinite section.

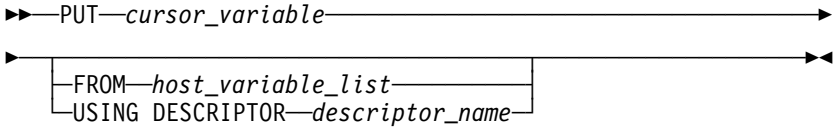


PUT (P)
Inserts a row into a table.



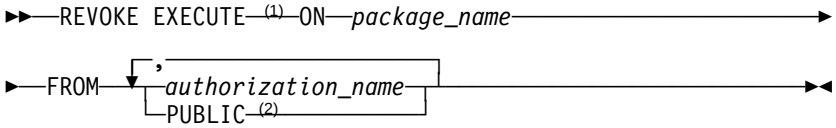
Extended PUT (P)

Inserts a row into a table.



REVOKE Package Privileges (I,P)

Revokes the privilege to execute a package.



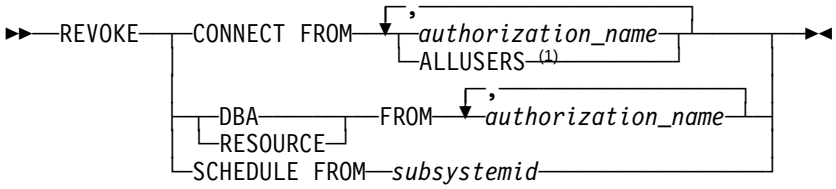
Notes:

1 RUN can be used as a synonym for EXECUTE and is provided for compatibility with previous versions of SQL/DS.

2 PUBLIC is specified only once.

REVOKE System Authorities (I,P)

Revokes system authorities.



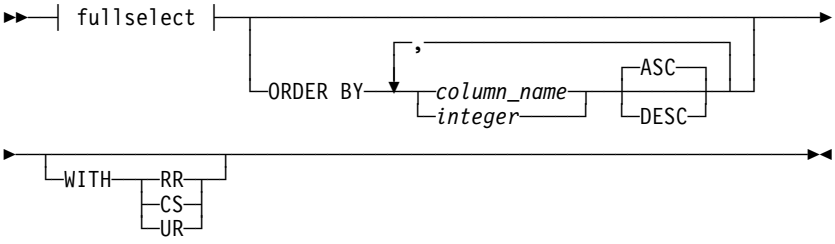
Note:

1 ALLUSERS can only be specified once.

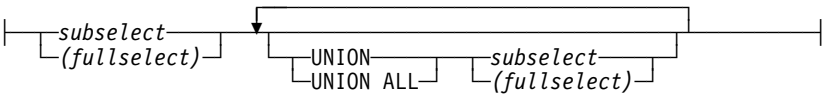
REVOKE Table Privileges (I,P)

Revokes privileges on a table or view.

interactive-select-statement



fullselect:

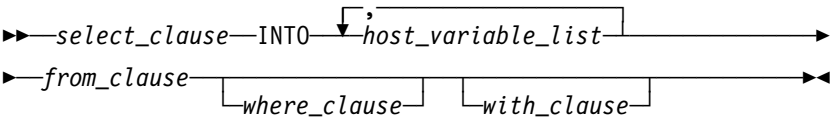


ROLLBACK (I,P)

Ends a logical unit of work without committing any changes.

SELECT INTO (P)

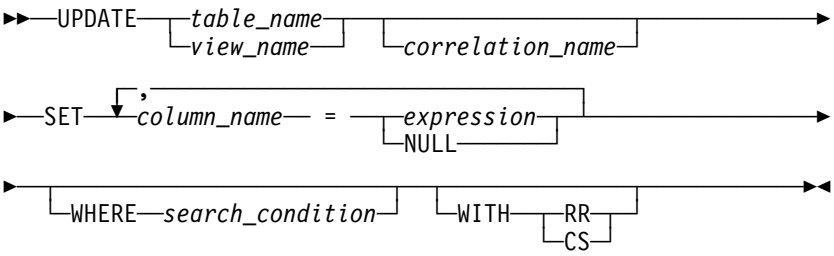
Produces a result table consisting of one row, and assigns the values in that row to host variables.



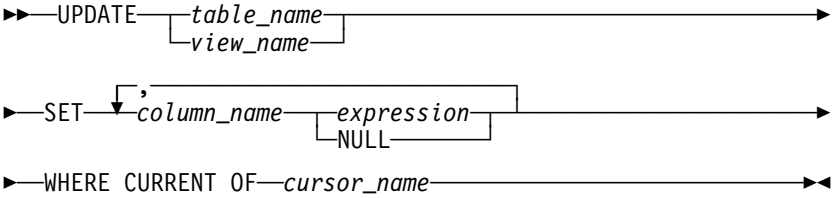
UPDATE (I,P)

Updates one or more column values in one or more rows of a table or view. Updating a row of a view updates a row of its base tables.

Searched update (I,P)

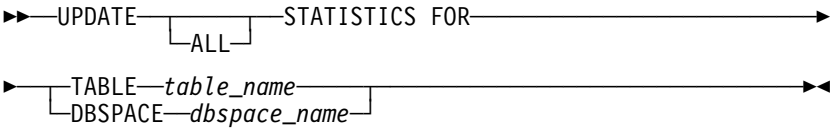


Positioned update (P)



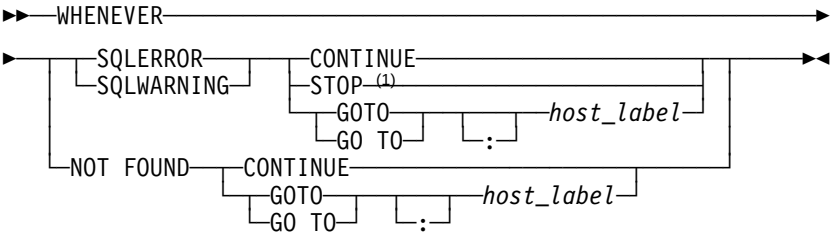
UPDATE STATISTICS (I,P)

Updates the statistics in the catalog about the tables and indexes.



WHENEVER (P)

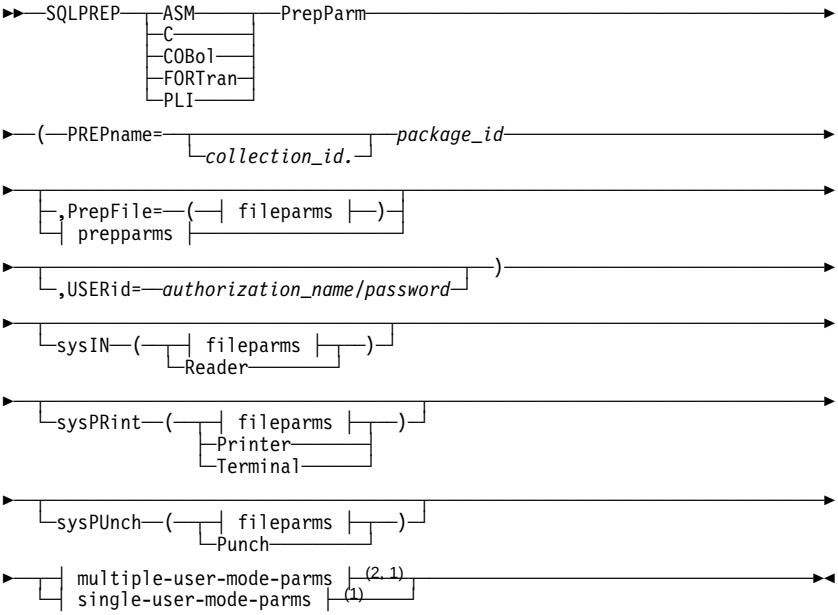
Specifies the action to be taken when a specified exception condition occurs.



Note:
[1] STOP is not valid for C and FORTRAN.

Preprocessing the Program

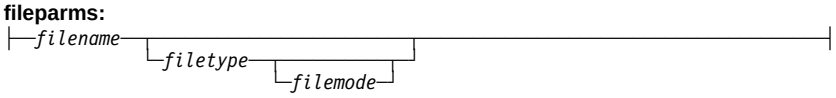
Program Preparation Command - VM Users



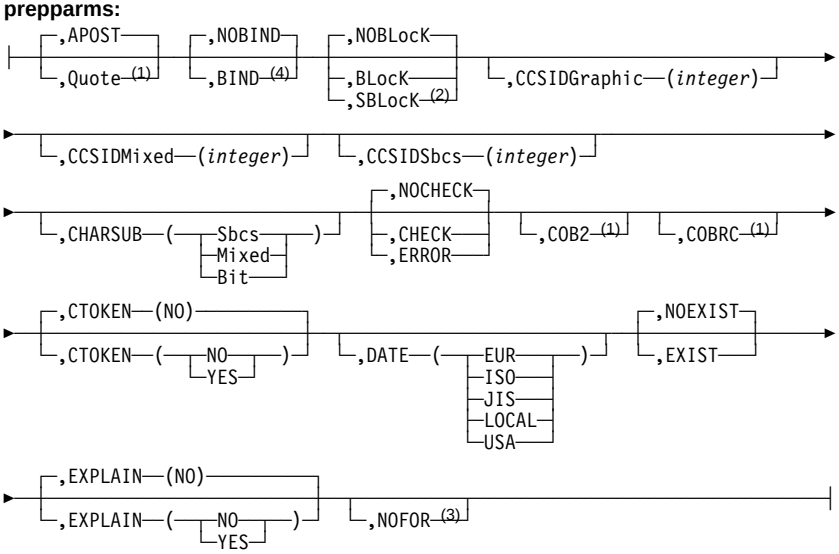
Notes:
[1] Valid for DB2 Server for VM only.

[2] Optional for multiple-user-mode.

fileparms:



prepparms:



Notes:

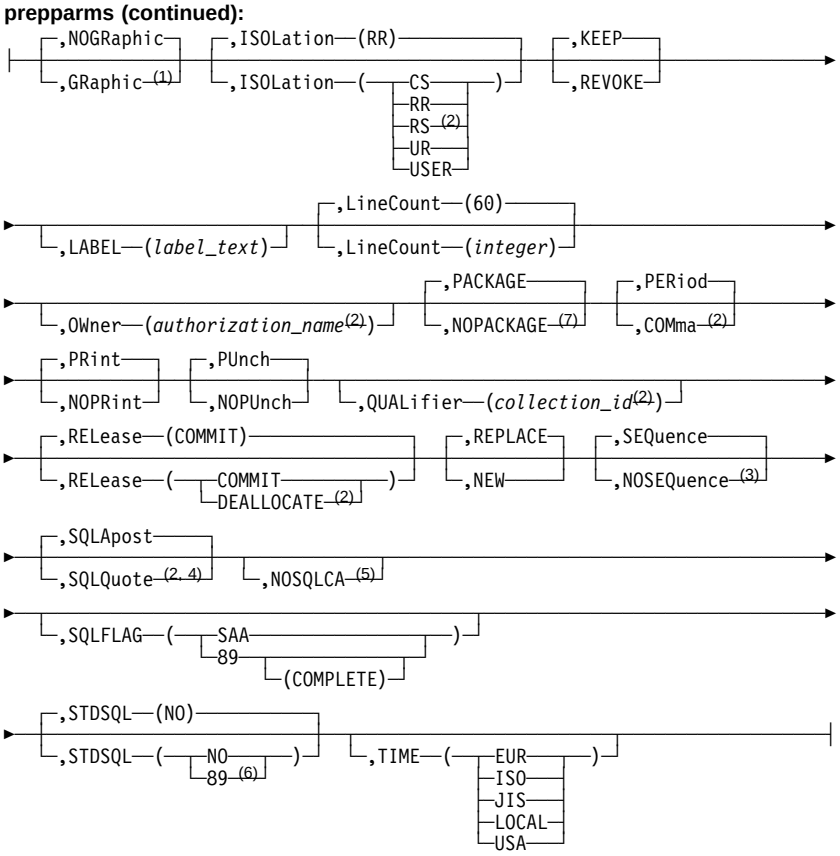
1 COBOL only.

2 Not meaningful for DB2 Server for VSE.

3 Implied if STDSQL(89) is specified.

4 Valid for DB2 Server for VSE only.

prepparms (continued):



Notes:

1 COBOL and PL/I only.

2 Only meaningful for a non-DB2 Server for VM or -DB2 Server for VSE application server.

3 C only.

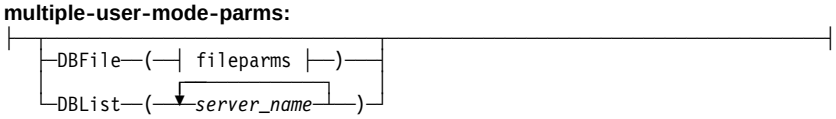
4 COBOL only.

5 Implied if STDSQL(89) is specified.

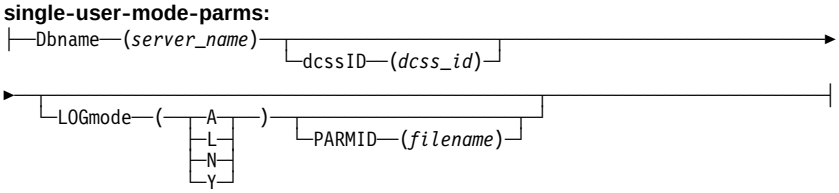
6 86 is a synonym for 89.

7 Valid for DB2 Server for VSE only.

multiple-user-mode-parms:

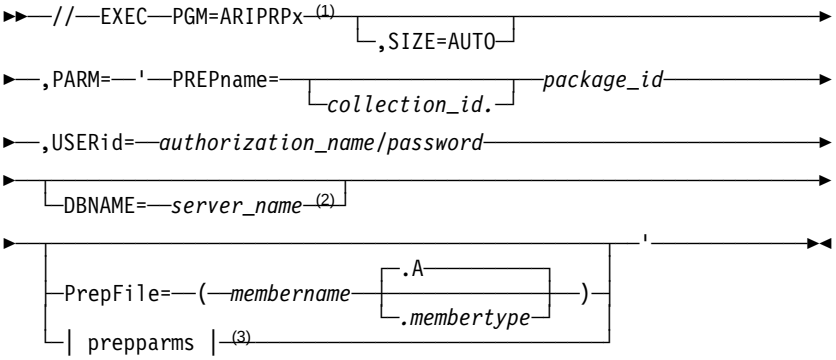


single-user-mode-parms:



Program Preparation Command - VSE Users

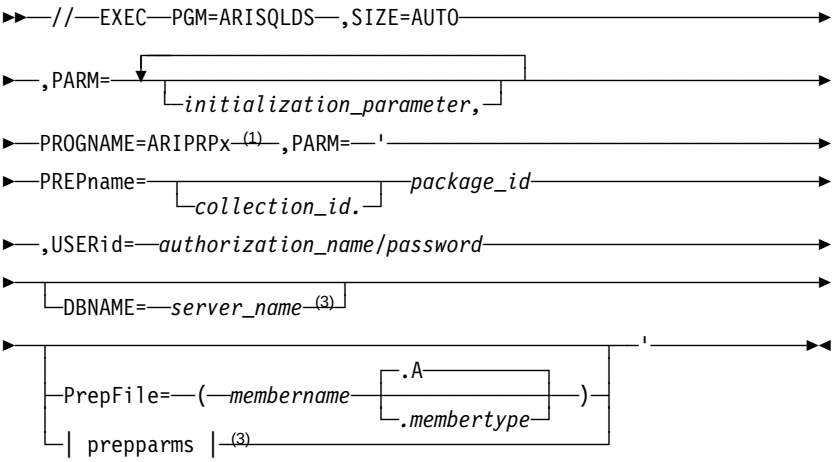
Multiple User Mode



Notes:

1. The x represents A for assembler, B for C, C for COBOL, F for FORTRAN, and P for PL/I.
2. From 1 to 18 characters, and identifies the DBNAME for the application server.
3. Same as `prepparms` for VM on page “Program Preparation Command - VM Users” on page 58.

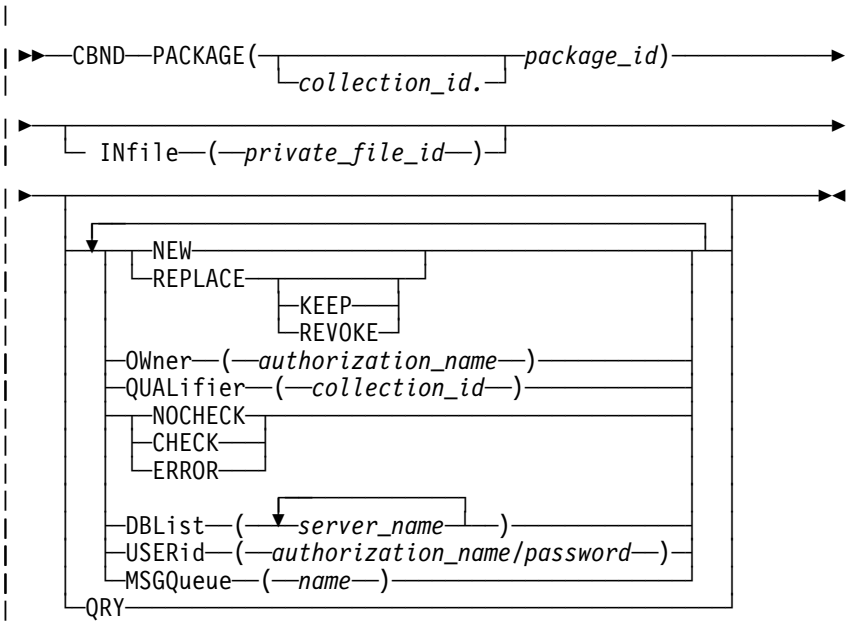
Single User Mode



Notes:

1. x represents A for assembler, B for C, C for COBOL, F for FORTRAN and P for PL/I.
2. Same as `prepparms` for VM on page “Program Preparation Command - VM Users” on page 58.
3. From 1 to 18 characters, and identifies the DBNAME for the application server.

Program BIND Command - VSE Users



Note: For any options not specified on the CBND transaction, the default option will be the option specified when the package was preprocessed, unless otherwise noted.

Interactive SQL Commands

Starting and Stopping ISQL - VM Users

To use ISQL in a VM environment, follow these steps:

1. Log on to VM.
2. IPL CMS.
3. Start ISQL.

To start ISQL, type the following 4-character transaction identifier and press ENTER:

```
ISQL
```

To stop communication with the DB2 Server for VM database manager through ISQL, type the following command in the input area and press ENTER:

```
EXIT
```

To end your VM terminal session, type the following command and press ENTER.

```
LOGOFF
```

Starting and Stopping ISQL - VSE Users

To start ISQL, type the following 4-character transaction identifier and press ENTER:

```
ISQL
```

If your installation has not defined a default authorization name, enter the authorization-name and password on the ISQL welcome panel, or use the explicit SQL CONNECT statement as follows:

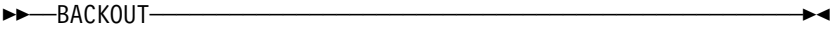
```
CONNECT authorization-name IDENTIFIED BY password
```

To stop communication with the DB2 Server for VSE database manager through ISQL, type the following command in the input area and press ENTER:

```
EXIT
```

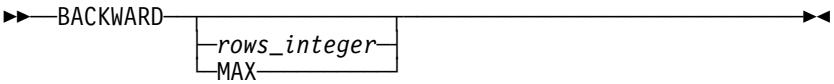
BACKOUT

Nullifies changes made since the last SAVE command or, if no previous SAVE command was issued, since the start of the INPUT command. Use this command only while applying the INPUT command with AUTOCOMMIT on.



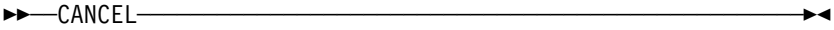
BACKWARD

Displays rows that occur before those that are currently displayed.



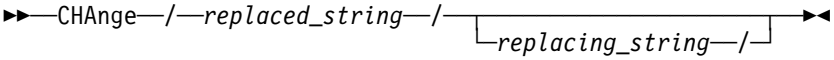
CANCEL

Cancels a command, an SQL statement, or a logical unit of work that is in progress.



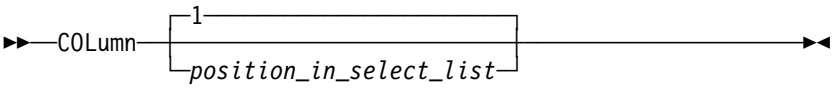
CHANGE

Modifies the current SQL statement in the command buffer and displays the result.



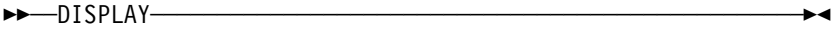
COLUMN

Displays the result of the interactive select-statement so that it begins with the specified column at the left edge of the display.



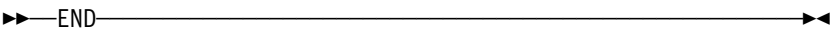
DISPLAY

Shows the results of the associated interactive select-statement on the display. This command can only be issued from a routine.



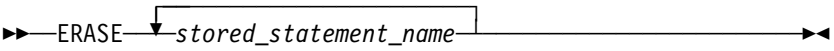
END

Ends the display of either a SELECT, COUNTER, SHOW or INPUT command, or a DISPLAY command in a routine.



ERASE

Erases stored SQL statements.

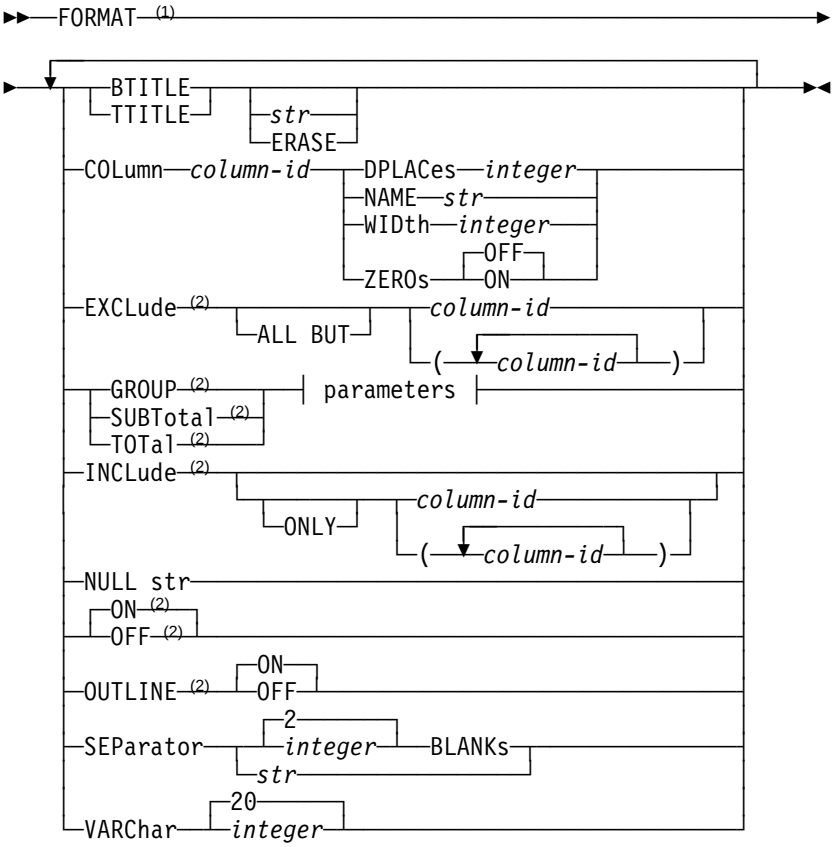


EXIT

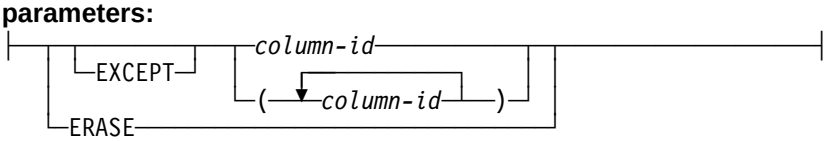
Ends the current ISQL terminal session.

FORMAT

Controls the format of the query result that is displayed.



parameters:



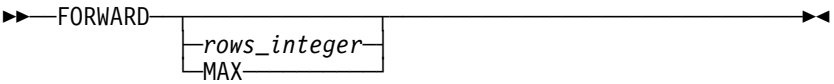
Notes:

1 Formatting can be performed only on the first 45 columns of a query result.

2 Any formatting command containing the keywords EXCLude, GROUP, INCLude, ON/OFF, OUTLINE, SUBTotal, or TOTal, causes the query to be reexecuted.

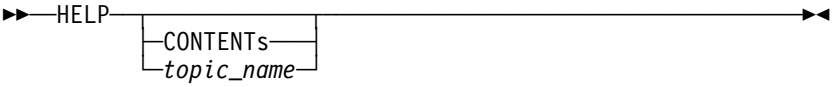
FORWARD

Moves your display forward through a query result.



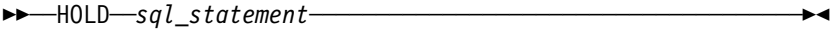
HELP

Retrieves online documentation (available as an option during installation of the database manager).



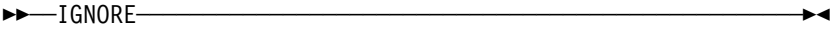
HOLD

Prevents an SQL statement from being processed when it is entered.



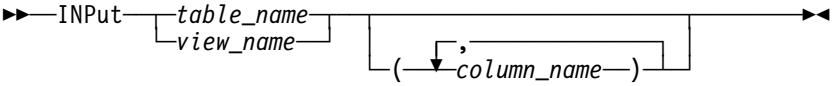
IGNORE

Nullifies a partially entered, multiple line command.



INPUT

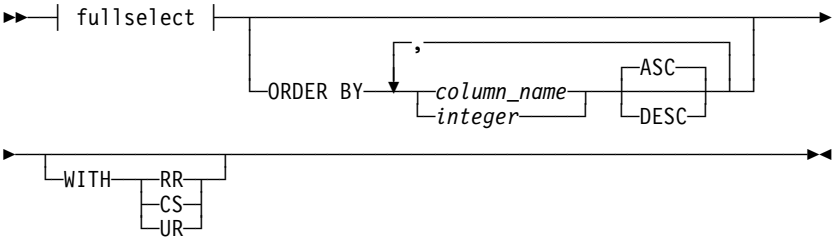
Inserts one or more rows of data into a table or view.



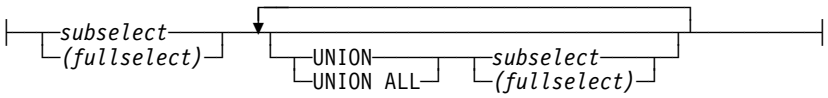
Interactive Select

The Interactive Select statement retrieves data from a table.

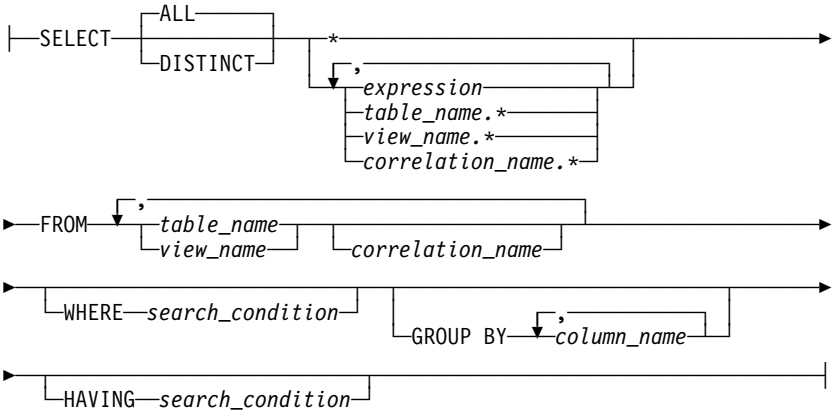
interactive-select-statement



fullselect:

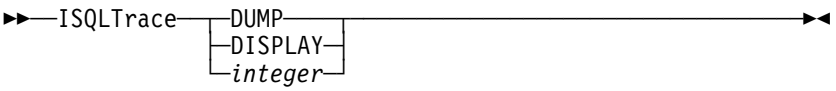


subselect:



ISQLTRACE

Traces activities within ISQL.

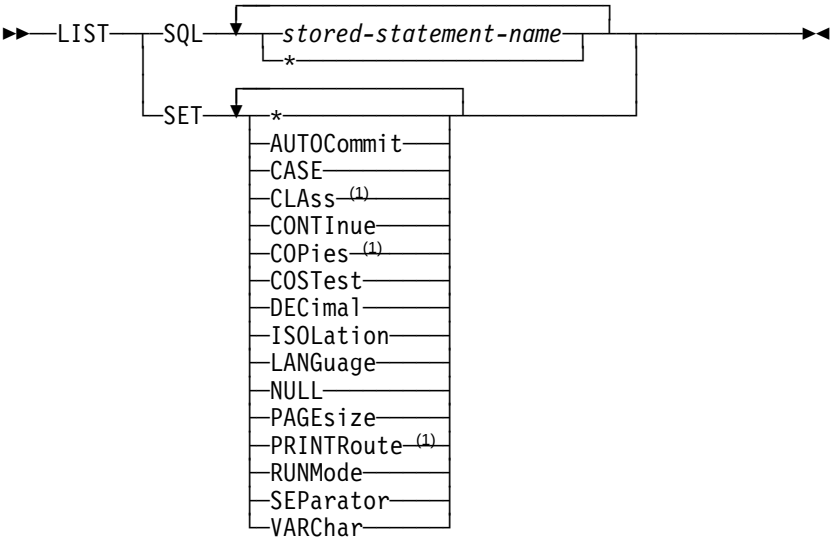


LEFT

Starts the display *integer* columns to the left, counting from the leftmost column on the display.

LIST

Lists information about stored SQL statements, or lists the settings of certain operational characteristics set by the SET command.

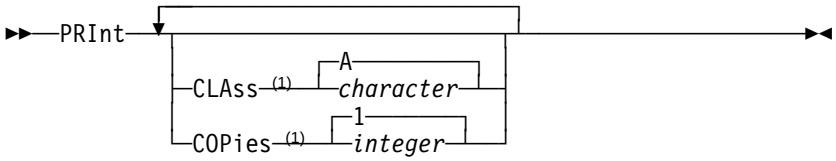


Note:

1 The parameters CLAss, COPies, and PRINTRoute are not applicable to VM.

PRINT - VM Users

Requests printed copies of a query result by sending it to the system printer.

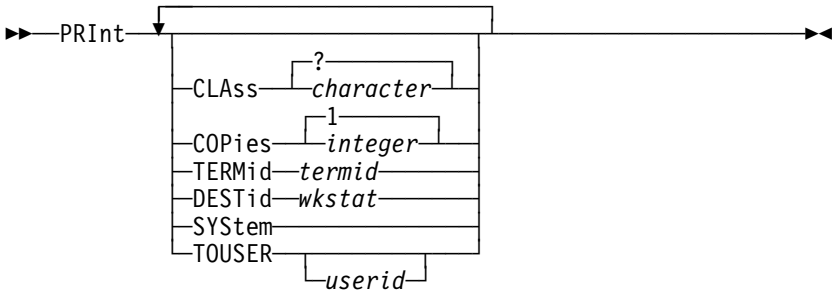


Note:

[1] The value for CLASS and COPIES specified in the PRINT command remain in effect until changed by a subsequent PRINT or CP SPOOL command.

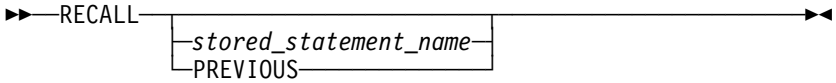
PRINT - VSE Users

Requests printed copies of a query result by sending it to the system printer, POWER remote printer, or CICS/VSE terminal.



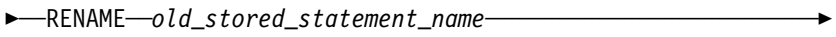
RECALL

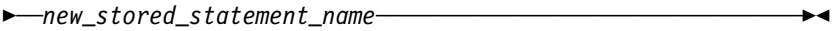
Retrieves a stored SQL statement.



RENAME

Renames a stored SQL statement.



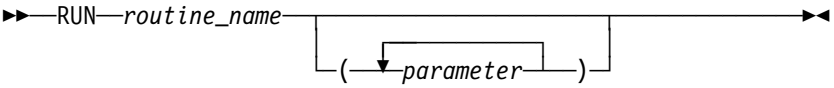


RIGHT

Starts the display *integer* columns to the right, counting from the leftmost column of the display.

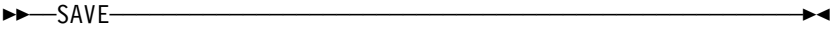
RUN

Initiates the processing of a routine. Multiple parameters are separated by blanks. Enclose a parameter in single quotation marks if it contains a blank.



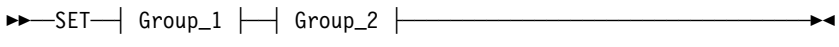
SAVE

Saves all changes since the last SAVE command or, if no SAVE command was issued, since the start of the INPUT command. SAVE is used while you are using the INPUT command with AUTOCOMMIT on only.

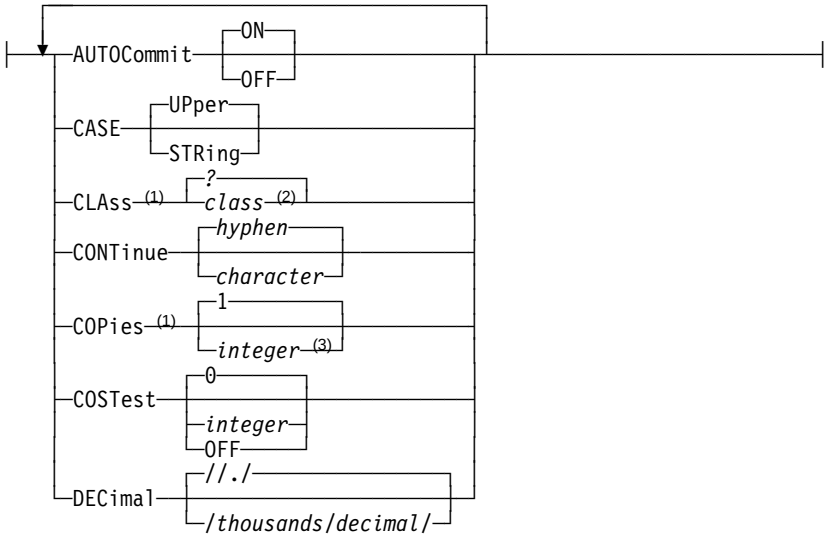


SET

Sets specified operational characteristics.



Group_1:



Notes:

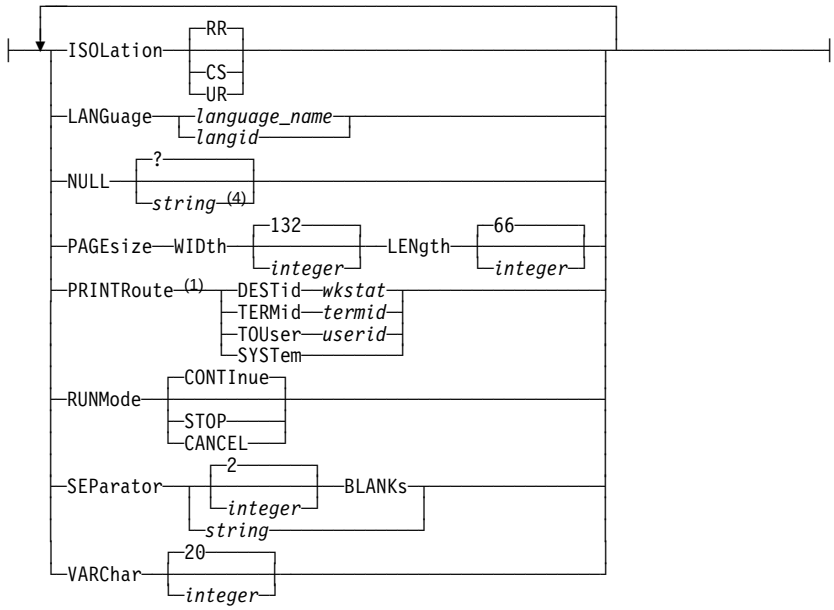
1 The parameters CLAss, COPies, and PRINTRoute are not applicable to VM.

2 Output class wanted (letters from A to Z).

3 Number of copies to be printed.

4 A maximum of 20 characters can be used for null values.

Group_2:



Notes:

1 The parameters CLAss, COPies, and PRINTRoute are not applicable to VM.

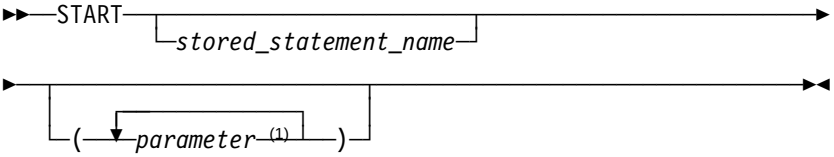
2 Output class wanted, (letters from A to Z).

3 Number of copies to be printed.

4 A maximum of 20 characters can be used for null values.

START

Processes the current SQL statement or a stored SQL statement.

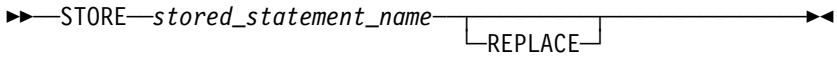


Note:

1 Enclose a parameter in single quotation marks when it contains a blank.

STORE

Saves the current SQL statement for later use. The statement remains stored until erased.

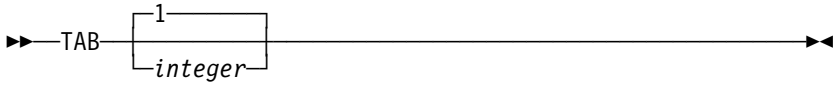


Note:

Related display formatting information can also be stored with the statement, however, some limitations exist. Refer to the *DB2 Server for VSE & VM Interactive SQL Guide and Reference* manual for more information about storing display information.

TAB

Displays all characters of a column that are too wide to fit on the display.



Note:

The TAB command is valid only for CHAR and VARCHAR columns.

ISQL Program Function Keys

The following default Program Function (PF) keys are provided through ISQL.

PF1, PF13 Issues a HELP command, which retrieves an explanation of the use of online help information and provides a list of topics available.

PF2, PF14 Issues a START command, which starts the command in the SQL command buffer (the **current** SQL command).

PF3, PF15 Issues an END command, which ends the display of a query.

PF4, PF16 Issues a PRINT command, which requests the currently displayed query result to be printed on the system or workstation printer.

PF5, PF17 Issues a RECALL command, which displays the contents of the SQL command buffer.

PF6, PF18 Not assigned.

PF7, PF19 Issues a BACKWARD command.

PF8, PF20 Issues a FORWARD command, which can be used to scroll through the query result half a screen at a time.

PF9, PF21 Issues a HOLD command, which prevents an SQL command from being processed when it is entered.

PF10, PF22 Issues a LEFT 1 command, which moves your view of the query result one column to the left.

PF11, PF23 Issues a RIGHT 1 command, which moves your view of the query result one column to the right.

PF12, PF24 Performs the RETRIEVE function, which moves the previously entered line into the input area.

Note: The CP SET command can be used to set the PF key functions.

CMS Subset VM Users

This CMS subset section applies to VM users. There is no equivalent section for VSE.

CMS or CP commands can be entered during an ISQL session. Supported commands from the CMS subset environment can be entered without terminating the ISQL session. To enter the CMS subset environment, type CMS on the command line, and press ENTER. Refer to Figure 1.

While in the CMS subset environment, you should not use any commands, programs, or execs that access the database manager. If they are used, the results are unpredictable and error conditions could occur.

To exit from the CMS environment, type RETURN in the command line, and press ENTER.

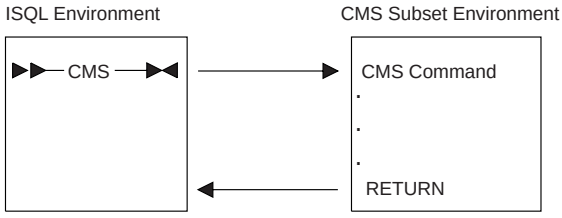


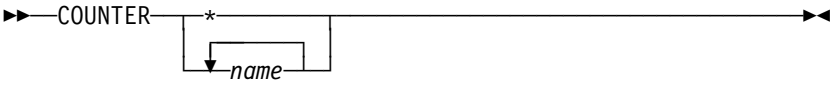
Figure 1. Using the CMS Subset Environment

Operator Commands

The following operator commands can be used within ISQL, or entered from the database machine operator console. For the complete set of operator commands, see the *DB2 Server for VSE & VM Operation* manual.

COUNTER

Used primarily to monitor system performance.

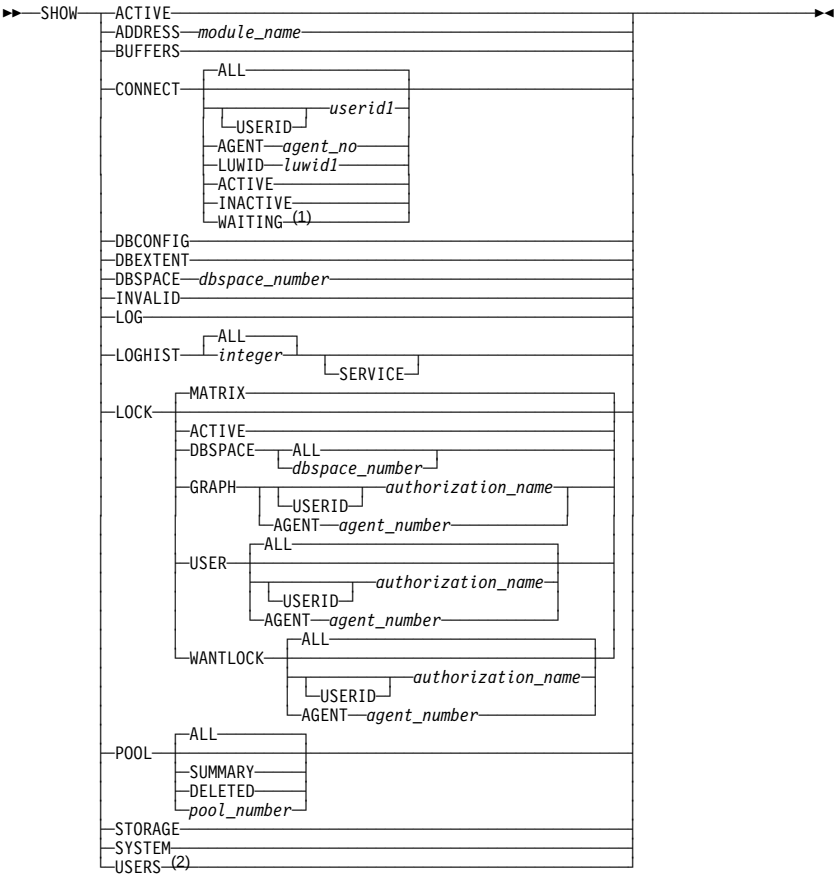


Valid names are:

BEGINLUW	DBSSCALL	LDIRBUFF	PAGEREAD
CHKPOINT	DEADLCK	LOCKLMT	PAGWRITE
DASDIO	DIRREAD	LOGREAD	RDSCALL
DASDREAD	DIRWRITE	LOGWRITE	ROLLBACK
DASDWRIT	ESCALATE	LPAGBUFF	WAITLOCK

SHOW

Monitors system activity.



Notes:

1 The SHOW CONNECT WAITING is not supported in the VSE environment.

2 The keyword USERS is not applicable to a VSE system.

Database Services Utility Commands

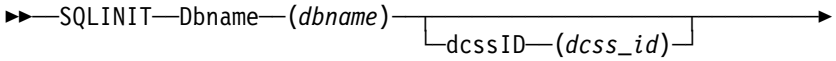
The Database Services Utility (DBS Utility) processes commands that are unique to the DBS Utility and certain SQL statements.

Starting and Stopping the DBS Utility

Before using the DB2 Server for VM or DB2 Server for VSE database manager for any application, initialize your database. The method of initialization depends on whether the DBS Utility is run in multiple user mode or single user mode.

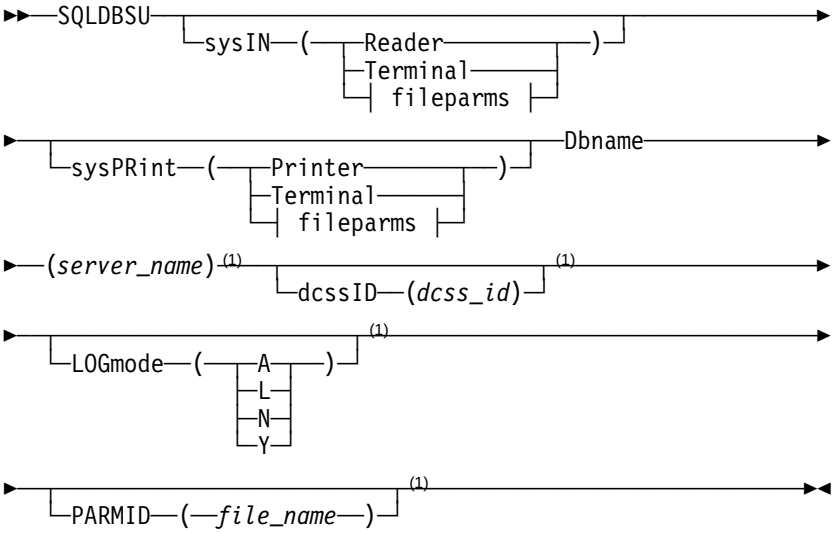
Starting the DBS Utility - VM Users

SQLINIT EXEC: This portion of the SQLINIT command should be issued before the SQLDBSU EXEC to initialize DB2 Server for VM database access and load the multiple user support system routines.

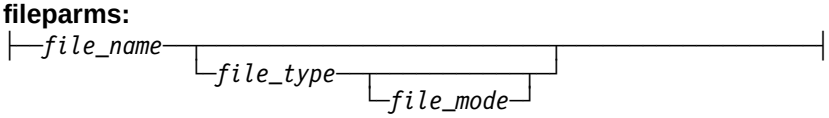


For more information about using the SQLINIT EXEC, see the *DB2 Server for VSE Database Administration* or *DB2 Server for VM Database Administration* manual.

SQLDBSU EXEC: Used to invoke the DBS Utility in single or multiple user mode.



fileparms:



Note:

1 Only applicable to single user mode.

Exiting from the DBS Utility - VM Users

Batch Mode: If a control file is supplied with the SYSIN option, an exit is made from the utility automatically after all the commands in the control file are processed.

Interactive Mode: If a control file is not supplied, the DBS Utility is used interactively. To exit, type the following command, and press ENTER.

```
EXIT;
```

Any uncommitted work is committed, and an exit is made from the utility to the conversational monitor system (CMS).

Starting the DBS Utility - VSE Users

Multiple User Mode: To invoke the DBS Utility, use the following EXEC statement as part of the JCL:

```
// EXEC PGM=ARIDBS,SIZE=AUTO
```

Single User Mode: To invoke the DBS Utility, use either:

```
// EXEC PROC=ARISDBSD
```

or

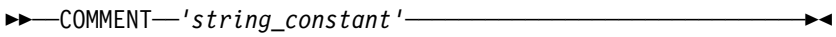
```
// EXEC ARISQLDS,SIZE=AUTO,                    X
//   PARM='SYSMODE=S,LOGMODE=N,PROGNAME=ARIDBS'
```

Exiting from the DBS Utility - VSE Users

The DBS Utility automatically ends after all commands in the control file are processed; that is, when /* is encountered.

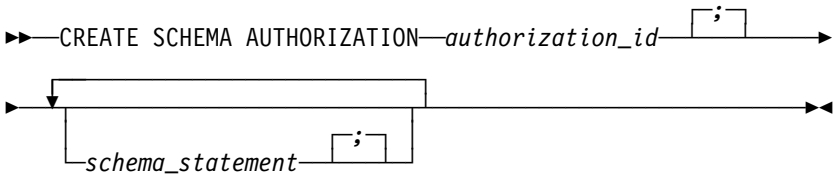
COMMENT

Documents input by supplying the Database Services Utility COMMENT commands at appropriate points within the Database Services Utility control command input stream. The utility displays the comments in the Database Services Utility message file listing.



CREATE SCHEMA

The sequential SCHEMA input file contains one CREATE SCHEMA statement, which is the first statement in the file.



DATALOAD

Loads or adds rows into existing tables from data in a sequential input file. DATALOAD and its subcommands are on more than one input record with each record in general containing data for a table row. Refer to the syntax diagram on page 83.

Table-Column-ID-Subcommand (TCI)

Identifies the location of the input records of the data for a table column. Each table-column-id-subcommand occupies a separate input record. Data must be in the same record positions in all records that relate to the table. Refer to the syntax diagram on page 83.

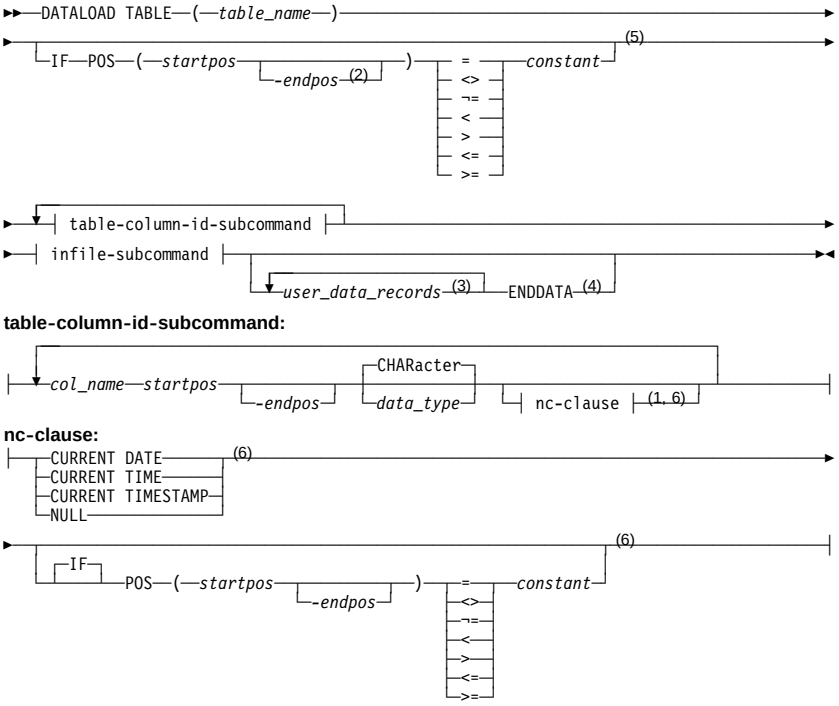


table-column-id-subcommand:

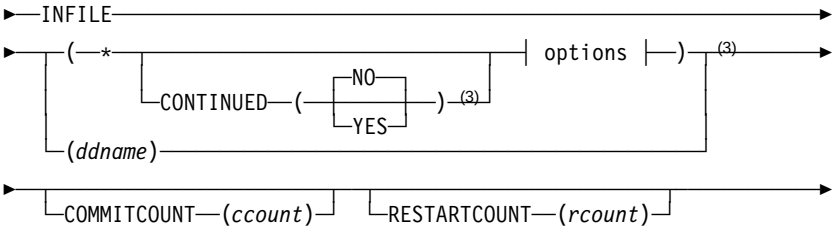
nc-clause:

Notes:

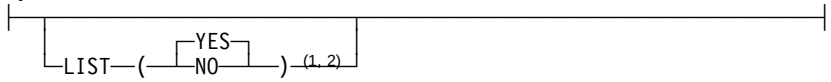
1. nc-clause is the short form for null-current-clause.
2. No blanks are allowed between startpos, hyphen, and endpos.
3. The *user-data-records* contain the data referenced by the preceding DATALOAD subcommands.
4. ENDDATA identifies the end of user-supplied data embedded in the control file. The command is valid if the previous DBS Utility command was an INFILE(*) subcommand.
5. These options must appear on the same physical line as DATALOAD TABLE.
6. This clause must appear on the same physical line.

Infile-subcommand
Identifies the file containing the data referenced by the preceding DATALOAD and TCI subcommands. When INFILE is followed by an asterisk, the data is in the control file and immediately follows the subcommand.

infile-subcommand - VM

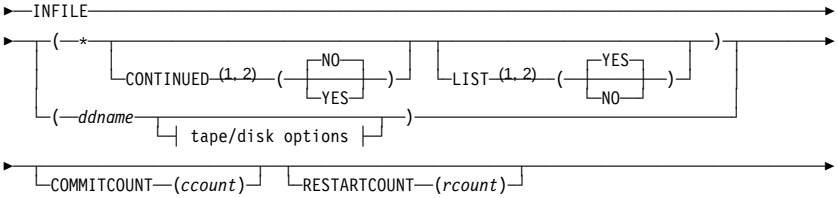


options:



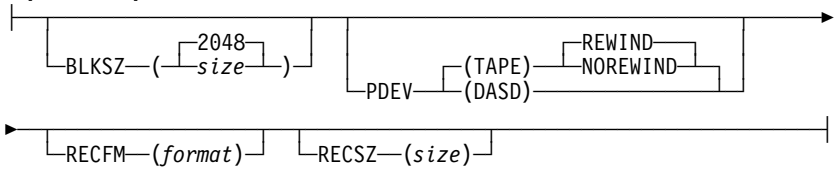
Notes:
1. No blanks are allowed between the keywords CONTINUED and LIST and the value specified for CONTINUED and LIST.
2. LIST can be specified before CONTINUED.
3. These options must appear on the same physical line as INFILE.

infile-subcommand - VSE



Notes:
1. No blanks are allowed between the keywords CONTINUED and LIST and the value specified for CONTINUED and LIST. LIST can be specified before CONTINUED.
2. LIST can be specified before CONTINUED.

tape/disk options:

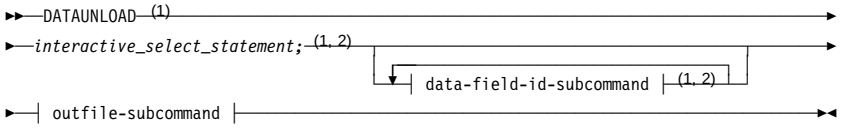


DATAUNLOAD

Selectively unloads data from tables and views to a user-defined sequential file of data.

Data-Field-Identification Subcommand

Identifies the location in the output record where the data for a column in the select-list parameter should be placed, and identifies the output record data-field data-type.

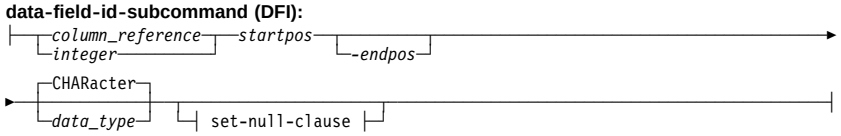


Notes:

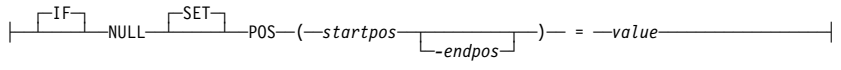
1 Each of these must be a separate record.

2 Both must appear on the same physical line.

data-field-id-subcommand (DFI):



set-null-clause:

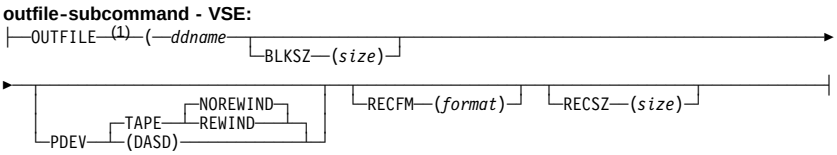


outfile-subcommand - VM:

Note:

1 The outfile-subcommand identifies the sequential output file that contains the data referenced by the preceding DATAUNLOAD subcommands.

outfile-subcommand - VSE:



Note:

1 The outfile-subcommand identifies the sequential output file that contains the data referenced by the preceding DATAUNLOAD subcommands.

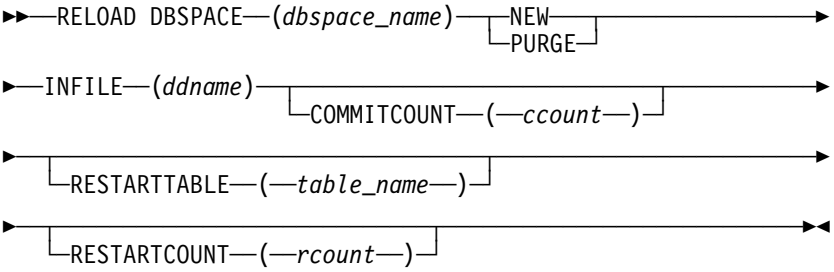
REBIND PACKAGE

Rebinds an existing package.

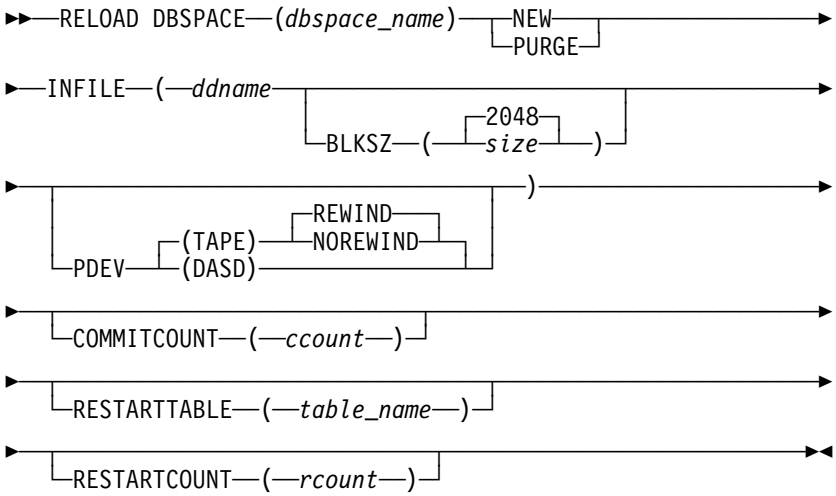
RELOAD DBSPACE

Identifies a RELOAD DBSPACE request and identifies a DBSPACE to be loaded.

VM Users



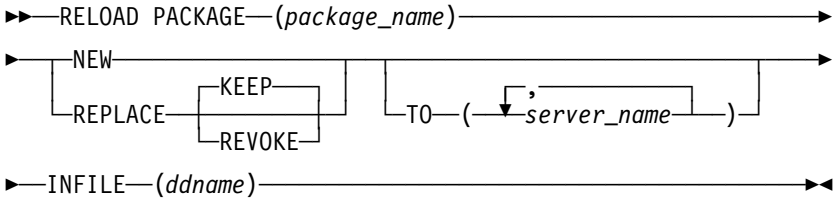
VSE Users



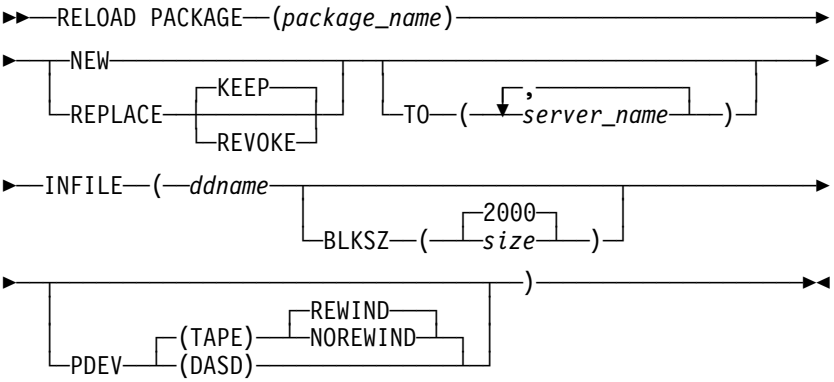
RELOAD PACKAGE

Identifies a RELOAD PACKAGE request and a package to be loaded. The UNLOAD PACKAGE output file becomes the input file for the RELOAD PACKAGE command.

VM Users



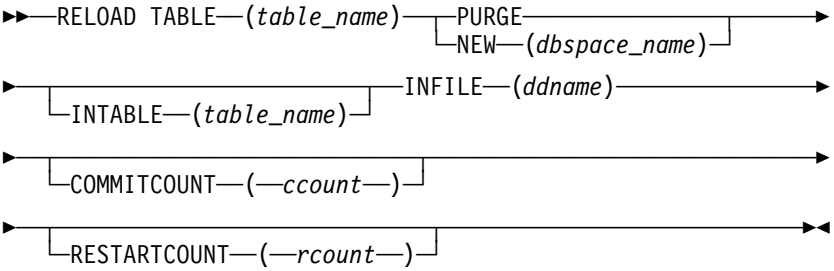
VSE Users



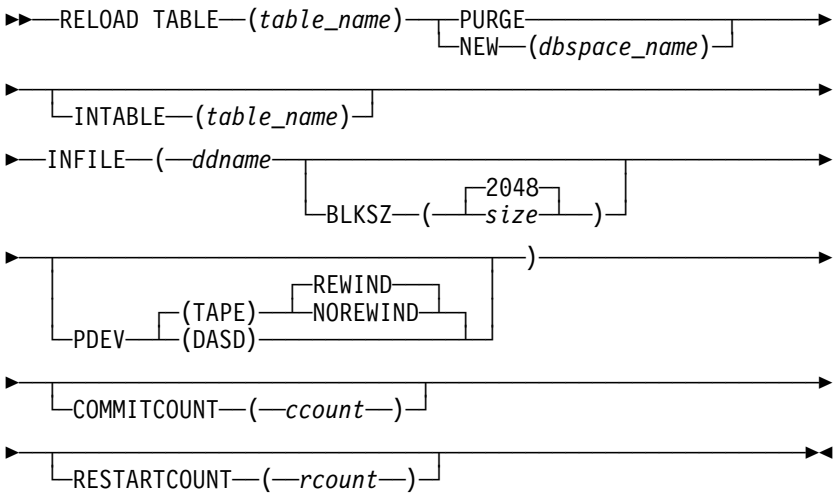
RELOAD TABLE

Identifies a RELOAD TABLE request and a table to be loaded.

VM Users

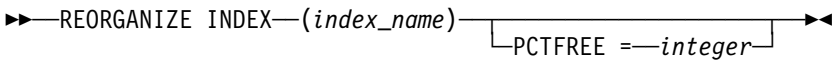


VSE Users



REORGANIZE INDEX

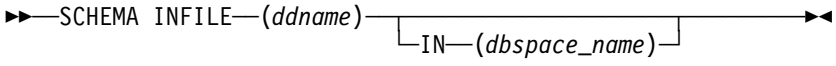
Corrects index fragmentation and skewing of index key values without first having to drop the index and then create it by using the DROP INDEX and CREATE INDEX SQL statements.



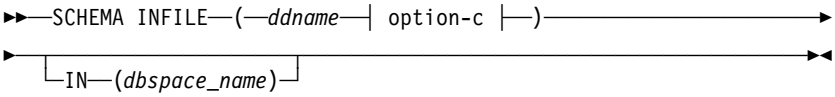
SCHEMA

Specifies an authorization ID and a list of table, view, and privilege definitions using the syntax of the SQL CREATE TABLE, CREATE VIEW, and GRANT statements. The SCHEMA command reads the text of a schema file and processes the statements in it.

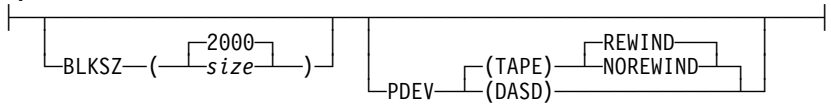
VM Users



VSE Users

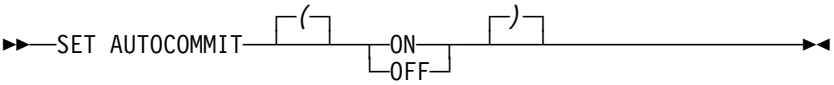


option-c:



SET AUTOCOMMIT

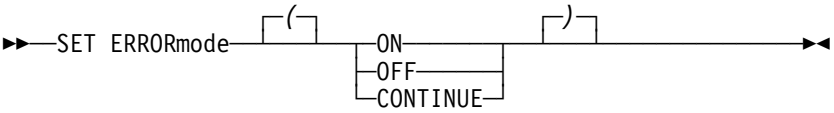
Activates or suppresses the automatic execution of the SQL COMMIT WORK statements. The SET AUTOCOMMIT command cannot span input records.



SET ERRORMODE

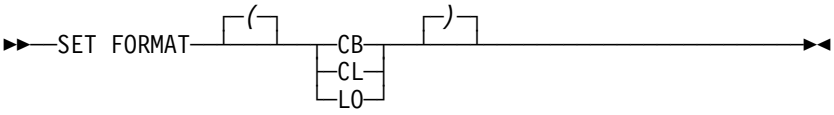
- Suspends the normal DBS Utility actions taken after a command processing error is detected and causes the DBS Utility to continue processing commands after an error has occurred
- Forces the DBS Utility to enter error mode processing
- Resumes normal DBS Utility command processing.

The SET ERRORMODE command cannot span input records. If you do not supply a SET ERRORMODE command in the input records, the DBS Utility operates as if you issued SET ERRORMODE OFF.



SET FORMAT

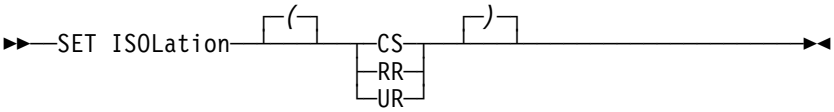
Identifies whether the Database Services Utility should use column-or-block format, column-or-list format, or only list format for SQL select-statement results. If not specified, Database Services Utility processing uses column-or-block format for SQL select-statement output.



SET ISOLATION

Controls the isolation level used for Database Services Utility processing. Each time the Database Services Utility runs, the isolation level is initialized to repeatable read (RR). SQL processing through the Database Services Utility is performed at the RR isolation level until a SET ISOLATION command is encountered.

The other isolation level settings are cursor stability (CS) and uncommitted read (UR).

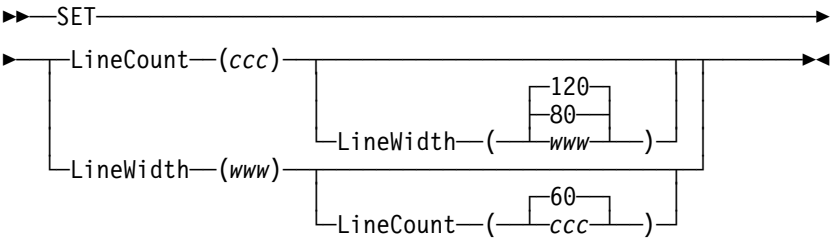


SET LINECOUNT (LINEWIDTH)

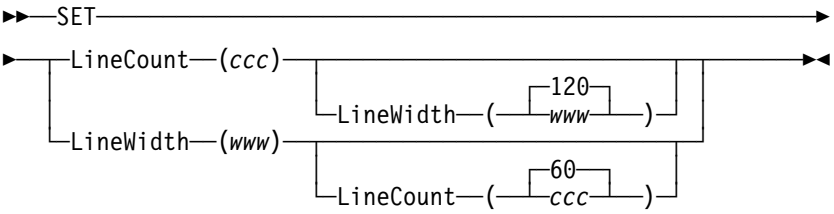
Defines the number of lines per page for Database Services Utility message file output and the number of print data positions used in each Database Services Utility message file record containing SQL SELECT statement output.

The SET LINECOUNT command cannot span input records. You must specify either the LINEWIDTH(*www*) parameter or the LINECOUNT(*ccc*) parameter, or both, to prevent a Database Services Utility processing error from occurring.

VM Users

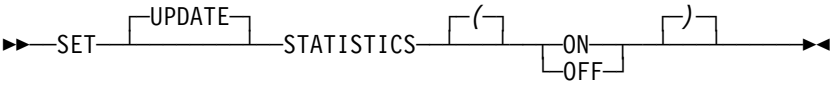


VSE Users



SET UPDATE STATISTICS

Controls the automatic UPDATE STATISTICS processing performed during Database Services Utility RELOAD TABLE, RELOAD DBSPACE, and DATALOAD TABLE command processing. The command cannot span input records. If you do not supply a SET UPDATE STATISTICS command in the input records, the Database Services Utility operates as if you issued SET UPDATE STATISTICS ON.

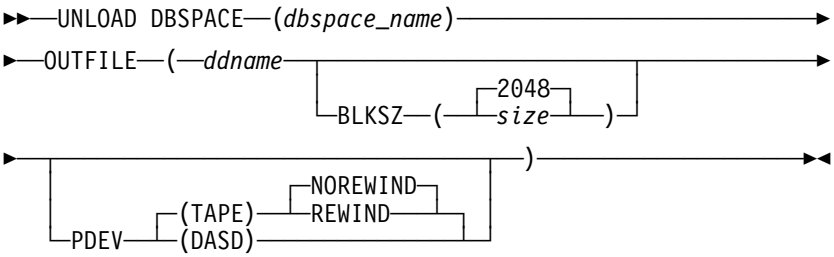


UNLOAD DBSPACE

Unloads all tables of the specified DBSPACE to a sequential output file.

VM Users

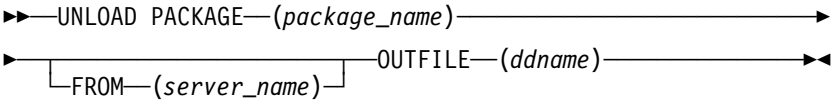
VSE Users



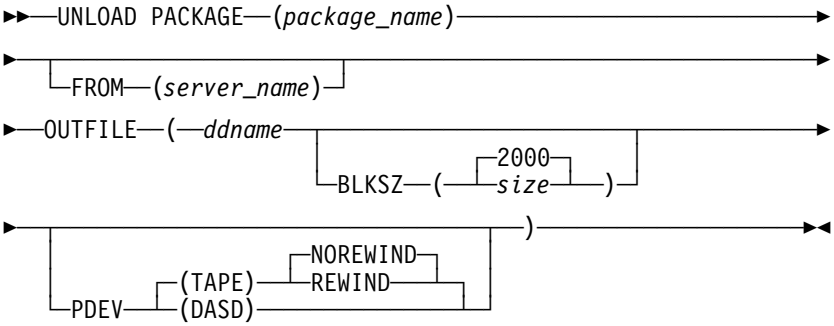
UNLOAD PACKAGE

Unloads a specific package to a file.

VM Users



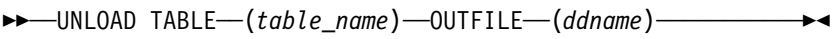
VSE Users



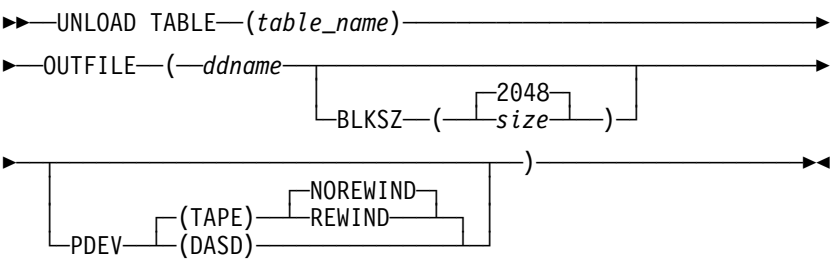
UNLOAD TABLE

Unloads a specific table or view to an output file.

VM Users



VSE Users



SQLCA and SQLDA

SQL Communication Area (SQLCA)

The SQLCA is a collection of variables that are updated at the end of the execution of every SQL statement.

Note: The field names are those provided by the SQL INCLUDE statement for Assembler, COBOL and PL/I. The names for C and FORTRAN are similar.

Field Name and Data Type	Description
SQLCAID CHAR(8)	The constant SQLCA.
SQLCABC INTEGER	Length of SQLCA. Always 136.
SQLCODE INTEGER	Negative: error condition, Zero: successful execution, Positive: warning condition.
SQLERRML SMALLINT	Length of SQLERRMC.
SQLERRMC VARCHAR(70)	Zero, one, or more tokens separated by X'FF'.

Field Name and Data Type	Description
SQLERRP CHAR(8)	Characters 1 to 3 identify the product. All 8 characters identify the module if the SQLCODE < 0.
SQLERRD(1) INTEGER	RDS return code.
SQLERRD(2) INTEGER	DBSS return code.
SQLERRD(3) INTEGER	Number of rows affected by INSERT, UPDATE or DELETE.
SQLERRD(4) INTEGER	Rough estimate of resources.
SQLERRD(5) INTEGER	For DELETE with RI: number of dependent rows. For local time exit: function number.
SQLERRD(6) INTEGER	Reserved.
SQLWARN array: SQLWARN0 CHAR(1)	'' if all indicators blank, 'S' if SQLWARN6 = 'S', 'W' otherwise.
SQLWARN1 CHAR(1)	'W' if value truncated when assigned to the host variable. 'Z' if invalid mixed data value is truncated.
SQLWARN2 CHAR(1)	'W' if null values eliminated from function result.
SQLWARN3 CHAR(1)	'W' if not enough host variables for column values.
SQLWARN4 CHAR(1)	'W' if WHERE missing from prepared UPDATE or DELETE.
SQLWARN5 CHAR(1)	'W' indicates performance degradation.
SQLWARN6 CHAR(1)	'S' if database manager is in unusable state due to error. 'W' if database manager was forced to terminate LUW.
SQLWARN7 CHAR(1)	'W' if adjustment made for last day of month. 'Z' if loss of digits on decimal division.

Field Name and Data Type	Description
SQLWARN8 CHAR(1)	'W' if statement disqualified for blocking. 'Z' if substitute character used on conversion.
SQLWARN9 CHAR(1)	'W' if not enough storage for blocking.
SQLWARNA CHAR(1)	'V' if error in converting SQLCA at application requester. 'W' if blocking factor could not be maintained.
SQLSTATE CHAR(5)	Standard cross-product return code.

SQL Descriptor Area (SQLDA)

An SQLDA is a collection of variables that is required for execution of the DESCRIBE statement, and can optionally be used by the OPEN, FETCH, EXECUTE, PUT, and Extended PREPARE statements.

Note: The field names in the following table are those provided by the SQL INCLUDE statement for Assembler and PL/I. The names for C are the same except that they are in lower case.

Field Name Data Type	Use in DESCRIBE	Use in Other Statements
SQLDAID CHAR(8)	The constant SQLDA.	Not used.
SQLDABC INTEGER	Length of SQLDA, equal to SQLN*44+16.	Same.
SQLN SMALLINT	Number of occurences of SQLVAR.	Same.
SQLD SMALLINT	Number of columns being described (times 2 if BOTH specified). Zero for a non-SELECT statement.	Number of host variables described by occurrences of SQLVAR.

Field Name Data Type	Use in DESCRIBE	Use in Other Statements
SQLVAR array:		
SQLTYPE SMALLINT	384/385 date	384/385 char containing date
	388/389 time	388/389 char containing time
	392/393 timestamp	392/393 char containing timestamp
	448/449 short varchar	448/449 short varchar
	452/453 character	452/453 character
	456/457 long varchar	456/457 long varchar
	460/461	460/461 NUL-terminated string
	464/465 short vargraphic	464/465 short vargraphic
	468/469 graphic	468/469 graphic
	472/473 long vargraphic	472/473 long vargraphic
	480/481 float	480/481 float
	484/485 packed decimal	484/485 packed decimal
	496/497 large integer	496/497 large integer
	500/501 small integer	500/501 small integer
	504/505	504/505 DISPLAY SIGN LEADING SEPARATE
SQLLEN SMALLINT	External length of column.	External length of host variable.
SQLDATA CHAR(4) or pointer	SBCS: X'0000' ccsid, mixed: X'0000' ccsid, bit: X'0000FFFF', graphic: X'0000' ccsid, otherwise: unused.	Address of host variable.
SQLIND CHAR(4) or pointer	1st byte set for character data: SBCS: X'01' MIXED: X'02' BIT: X'FF', otherwise: unused.[1]	Address of indicator variable, if there is one.
SQLNAME VARCHAR(30)	Name or label of the column.	SBCS: X'0000' ccsid, mixed: X'0000' ccsid, bit: X'0000FFFF', graphic: X'0000' ccsid, otherwise: unused.

[1] Not available for the DRDA protocol.

Catalog Tables

The DB2 Server for VM and DB2 Server for VSE database management systems maintain a set of tables, called catalog tables, that store information about the database. The catalog tables are automatically updated by the database manager during normal operation and in response to SQL data definition and control statements. Following is a Roadmap.

Roadmap

Item	Catalog Table	Page
authorization	SYSUSERAUTH	103
	SYSUSERLIST	103
character conversion	SYSSTRINGS	103
character set	SYSCHARSETS	100
coded character set identifiers	SYSCCSIDS	100
	SYSSTRINGS	103
column	SYSCOLUMNS	100
	SYSKEYCOLS	101
column update privilege	SYSCOLAUTH	100
column with field procedure	SYSFIELDS	101
constraint	SYSKEYS	101
dbspace	SYSDBSPACES	101
	SYSUSAGE	103
	SYSDROP	101
dbspace waiting to be dropped	SYSDROP	101
default	SYSOPTIONS	102
dropped dbspace	SYSDROP	101
dropped table	SYSDROP	101
field procedures	SYSFPARMS	101
	SYSFIELDS	101
foreign key	SYSKEYS	101
index	SYSINDEXES	101
	SYSUSAGE	103
index column statistics	SYSCOLSTATS	100
	SYSCOLUMNS	100
	SYSINDEXES	101
key	SYSKEYS	101
key column	SYSKEYCOLS	101
language for character set	SYSLANGUAGE	102
option	SYSOPTIONS	102
package	SYSACCESS	99
	SYSUSAGE	103
package run privilege	SYSPROGAUTH	102
password	SYSUSERAUTH	103

Item	Catalog Table	Page
privilege	SYSCOLAUTH	100
	SYSPROGAUTH	102
	SYSTABAUTH	103
primary key	SYSKEYS	101
statistics	SYSCATALOG	99
	SYSCOLSTATS	100
	SYSCOLUMNS	100
	SYSDBSPACES	101
	SYSINDEXES	101
stored procedures	SYSPARMS	102
	SYSPSERVERS	103
	SYSROUTINES	102
synonym	SYSSYNONYMS	103
table	SYSCATALOG	99
	SYSCOLUMNS	100
	SYSUSAGE	103
table privilege	SYSTABAUTH	103
table waiting to be dropped	SYSDROP	101
unique constraint	SYSKEYS	101
view	SYSVIEWS	104
	SYSCATALOG	99
	SYSCOLUMNS	100
	SYSACCESS	99
	SYSUSAGE	103
view privilege	SYSTABAUTH	103

Catalog Table Descriptions

Following is a description of the catalog tables:

SYSACCESS Records information about the tables in which packages are stored.

CONSTKN	FIRSTROW	TABID	TNAME
CREATOR	LINKID	TABTYPE	VALID
DBSPACENO	PLABEL	TIMESTAMP	

SYSCATALOG Contains a row for each table or view in the database, including one for itself and one for each catalog table.

AVGROWLEN	DBSPACENO	NCOLS	ROWCOUNT
CLUSTERROW	DEPENDENTS	NOVERFLOW	TABID
CLUSTERTYPE	INACTIVE	NPAGES	TABLETYPE
CREATOR	LFDDBSPACE	PARENTS	TLABEL
DATACAPTURE	LFDLINK	PCTPAGES	TNAME
DBSPACENAME	LFDTABID	REMARKS	

SYSCCSIDS Contains a row for every CCSID supported by the database manager.

CCSID	DBCSID	SUBTYPE
CHARNAME	SBCSID	

SYSCHARSETS Rows contain information about various EBCDIC character sets; information is based on what is specified in the CHARNAME initialization parameter.

CHARCLASS	CHARTRANS	NAME

SYSCOLAUTH Records grants of the UPDATE privilege on tables and views when the privilege is granted on a column-by-column basis.

COLNAME	GRANTEE	TIMESTAMP
CREATOR	GRANTOR	TNAME

SYSCOLSTATS Keeps column statistics for a column which is the first column of an index.

CNAME	FREQ2PCT	TNAME	VAL90
CREATOR	FREQ1VAL	VAL10	
FREQ1PCT	FREQ2VAL	VAL50	

SYSCOLUMNS Contains a more detailed description of the database than SYSCATALOG.

AVGCOLLEN	COLINFO	HIGH2KEY	REMARKS
CCSID	COLNO	LENGTH	SUBTYPE
CLABEL	COLTYPE	LOW2KEY	SYSLENGTH
CNAME	CREATOR	NULLS	TNAME
COLCOUNT	FLDPROC	ORDERFIELD	

SYSDBSPACES Contains a row for each PUBLIC and PRIVATE DBSPACE in the database, including those DBSPACEs that no user has yet acquired.

DBSPACENAME	FREEPCT	NPAGES	OWNER
DBSPACENO	LOCKMODE	NRHEADER	PCTINDX
DBSPACETYPE	NACTIVE	NTABS	POOL

SYSDROP Contains a list of objects waiting to be dropped.

DBSPACENO	QUALF	TABID

SYSFIELDS Contains a row for each column that has a field procedure associated with it.

CNAME	FLDLENGTH	FPNAME	TNAME
COLNO	FLDTYPE	FPPARMLIST	
CREATOR	FPEXITPARML	FPWORKAREA	

SYSFPARMS Holds the field procedure value block contents for each field procedure.

CNAME	FPEXITPARM	SEQNO
CREATOR	FPNAME	TNAME

SYSINDEXES Contains a row for every index currently in existence, including the indexes that the database manager maintains on its own catalog tables.

CLUSTER	FIRSTKEYCOUNT	INDEXTYPE	NLEAF
CLUSTERRATIO	FULLKEYCOUNT	IPCTFREE	NLEVELS
COLNAMES	ICREATOR	KEYLEN	RELEASE
COLNUMBERS	IID	KEYTYPE	TNAME
CREATOR	INAME	LOCKMODE	

SYSKEYCOLS Contains a row for every column in every key.

CCSID	FLDPROC	KEYTYPE	TCREATOR
CNAME	KEYNAME	SYSLENGTH	TIMESTAMP
DATACODE	KEYORD	TABLEORD	TNAME

SYSKEYS Contains a row for each primary and each foreign key.

DELETERULE	KEYNAME	REFTNAME	TIMESTAMP
INAME	KEYTYPE	STATUS	TNAME
KEYCOLS	REFTCREATOR	TCREATOR	

SYSLANGUAGE Contains the names of all national languages currently installed, a unique four-character code for each language, and a brief description of each language.

LANGID	LANGKEY	LANGUAGE	REMARKS

SYSOPTIONS Contains the options and defaults that may be implemented for this database.

REMARKS	SQLOPTION	VALUE

The following named rows describe the options and defaults that may be implemented for this database:

CCSIDGRAPHIC	DATE	MCCSIDGRAPHIC
CCSIDMIXED	DBCS	MCCSIDMIXED
CCSIDSBCS	DEFAULT LANGUAGE	MCCSIDSBCS
CHARNAME	LDATELEN	RELEASE
CHARSUB	LTIMELEN	TIME

| **SYSPARMS** Describes the parameters for the stored procedures
| defined.

NAME	AUTHID	PARMNAME	SUBTYPE
ROUTINEID	ROWTYPE	ORDINAL	CCSID
TYPENAME	DATATYPEID	LENGTH	SCALE

| **SYSPSERVERS** Defines the stored procedure servers where stored
| procedures run and puts them in groups.

PSERVER	SERVGROUP	AUTOSTART	DESCRIPTION

SYSPROGAUTH Records privileges of users to run programs, and to grant these privileges to other users.

CREATOR	GRANTOR	RUNAUTH
GRANTEE	PROGNAME	TIMESTAMP

| **SYSROUTINES** Specifies the load module or phase name and
| package name for a given stored procedure.

| NAME | AUTHID | LOADMOD | ROUTINEID
| PARMCOUNT | LANGUAGE | PARAMETERSTYLE | STAYRESIDENT
| PROGRAMTYPE | COMMITONRETURN | RESULTSETS | SERVGROUP
| DEFSERV | RUNOPTS | REMARKS |

SYSSTRINGS Contains a list of the valid combinations for source and target CCSID tags when using the remote unit of work feature.

ERRORBYTE	OUTCCSID	TRANSPROC	TRANSTAB2
INCCSID	SUBBYTE	TRANSTAB1	TRANSTYPE

SYSSYNONYMS Contains a row for every synonym currently in effect.

ALTNAME	CREATOR	TNAME	USERID

SYSTABAUTH Records privileges owned by users to access tables and views, and privileges on tables and views exercised by programs.

ALTERAUTH	GRANTOR	SCREATOR	TIMESTAMP
DELETEAUTH	INDEXAUTH	SELECTAUTH	TTNAME
GRANTEE	INSERTAUTH	STNAME	UPDATEAUTH
GRANTEETYPE	REFAUTH	TCREATOR	UPDATECOLS

SYSUSAGE Records dependencies of one database object on another.

BCREATOR	BTYPE	DNAME	TIMESTAMP
BNAME	DCREATOR	DTYPE	

SYSUSERAUTH Records special privileges of DBA, RESOURCE, SCHEDULE or CONNECT authority held by a user or a special privilege exercised by a program.

AUTHOR	NAME	RESOURCEAUTH
DBAAUTH	PASSWORD	SCHEDULEAUTH

SYSUSERLIST Records special privileges for access by users who do not have DBA authority.

AUTHOR	NAME	SCHEDULEAUTH
DBAAUTH	RESOURCEAUTH	

SYSVIEWS Contains the definitions of all views.

SEQNO	VIEWCHECK	VIEWNAME
VCREATOR	VIEWMAT	VIEWTEXT

Application Server Support for VSE

Up to 36 DB2 Server for VSE application servers can be active at the same time in your VSE system.

DBNAME Directory

The DBNAME directory is a user-definable directory of application server names, contained in an A-type source member called ARISDIRD. Each entry in this directory is an 80 byte record in the following format:

- Comment, column 1
- Transaction Program Name (TPN), columns 2 to 5
- Application Identifier (APPLID), columns 10 to 17
- System default marker (SYSDEFAULT), column 21
- DBNAME columns 22 to 39
- Partition name (PDEFAULT) columns 44 and 45
- Privileged (PRIVILEGE) column 50.

For more information about the DBNAME directory, refer to the *DB2 Server for VSE System Administration* manual.

SQL Reserved Words

Following is a list of SQL reserved words you should avoid using:

ACQUIRE
ADD
ALL
ALTER
AND
ANY
AS
ASC
AVG

BETWEEN
BY

CHAR
CHARACTER
COLUMN
COMMENT
COMMIT
CONCAT
CONNECT
COUNT
CREATE
CURRENT

DBA
DBSPACE
DELETE
DESC
DISTINCT
DOUBLE
DROP

EXCLUSIVE
EXECUTE
EXISTS
EXPLAIN

FIELDPROC
FOR
FROM

GRANT
GRAPHIC
GROUP

HAVING

IDENTIFIED
IN
INDEX
INSERT
INTO
IS

LIKE
LOCK
LONG

MAX
MIN
MODE

NAMED
NHEADER
NOT
NULL

OF
ON
OPTION
OR
ORDER

PACKAGE
PAGE
PAGES
PCTFREE
PCTINDEX
PRIVATE
PRIVILEGES
PROGRAM
PUBLIC

RESOURCE
REVOKE
ROLLBACK
ROW
RUN

SCHEDULE
SELECT
SET
SHARE
SOME
STATISTICS
STORPOOL
SUM
SYNONYM

TABLE
TO

UNION
UNIQUE
UPDATE
USER

VALUES
VIEW

WHERE
WITH
WORK

DBS Utility Reserved Words

In addition to the SQL reserved words, do not use the following words in Database Services Utility commands as the name for a table, view, column, or DBSPACE, unless they are enclosed in double quotation marks (").

DATALOAD
DATAUNLOAD
INFILE
INMOD
OUTFILE
REBIND
RELOAD
REORGANIZE
SCHEMA
UNLOAD

Notes

Communicating Your Comments to IBM

DB2® Server for VSE & VM
Quick Reference
Version 6 Release 1

Publication No. SC09-2670-00

If there is something you like—or dislike—about this book, please let us know. You can use one of the methods listed below to send your comments to IBM. If you want a reply, include your name, address, and telephone number. If you are communicating electronically, include the book title, publication number, page number, or topic you are commenting on.

The comments you send should only pertain to the information in this book and its presentation. To request additional publications or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give it to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - United States and Canada: 416-448-6161
  - Other countries: (+1)-416-448-6161
- If you prefer to send comments electronically, use the network ID listed below. Be sure to include your entire network address if you wish a reply.
  - Internet: torrcf@ca.ibm.com
  - IBMLink: toribm(torrcf)
  - IBM/PROFS: torolab4(torrcf)
  - IBMMAIL: ibmmail(caibmwt9)

Readers' Comments — We'd Like to Hear from You

DB2® Server for VSE & VM
Quick Reference
Version 6 Release 1

Publication No. SC09-2670-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____ _____

Phone No. _____ _____

PLACE
POSTAGE
STAMP
HERE

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 EGLINTON AVENUE EAST
NORTH YORK ONTARIO CANADA M3C 1H7

Fold and Tape **Please do not staple** Fold and Tape

Readers' Comments — We'd Like to Hear from You
SC09-2670-00

File Number: S370/4300-50
Program Number: 5648-A70

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC09-2670-00

Spine information:

DB2® Server for VSE & VM Quick Reference *Version 6 Release 1*