This chapter provides a list of the PostgreSQL SQL commands, followed by the precise syntax rules for each of these commands. This set of commands is taken from the psql command-line tool. Now that you have Postgres installed, open the psql as −

**Program Files → PostgreSQL 9.2 → SQL Shell(psql).**

Using psql, you can generate a complete list of commands by using the \help command. For the syntax of a specific command, use the following command −

postgres-# \help <command\_name>

The SQL Statement

An SQL statement is comprised of tokens where each token can represent either a keyword, identifier, quoted identifier, constant, or special character symbol. The table given below uses a simple SELECT statement to illustrate a basic, but complete, SQL statement and its components.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **SELECT** | **id, name** | **FROM** | **states** |
| Token Type | Keyword | Identifiers | Keyword | Identifier |
| Description | Command | Id and name columns | Clause | Table name |

PostgreSQL SQL commands

ABORT

Abort the current transaction.

ABORT [ WORK | TRANSACTION ]

ALTER AGGREGATE

Change the definition of an aggregate function.

ALTER AGGREGATE name ( type ) RENAME TO new\_name

ALTER AGGREGATE name ( type ) OWNER TO new\_owner

ALTER CONVERSION

Change the definition of a conversion.

ALTER CONVERSION name RENAME TO new\_name

ALTER CONVERSION name OWNER TO new\_owner

ALTER DATABASE

Change a database specific parameter.

ALTER DATABASE name SET parameter { TO | = } { value | DEFAULT }

ALTER DATABASE name RESET parameter

ALTER DATABASE name RENAME TO new\_name

ALTER DATABASE name OWNER TO new\_owner

ALTER DOMAIN

Change the definition of a domain specific parameter.

ALTER DOMAIN name { SET DEFAULT expression | DROP DEFAULT }

ALTER DOMAIN name { SET | DROP } NOT NULL

ALTER DOMAIN name ADD domain\_constraint

ALTER DOMAIN name DROP CONSTRAINT constraint\_name [ RESTRICT | CASCADE ]

ALTER DOMAIN name OWNER TO new\_owner

ALTER FUNCTION

Change the definition of a function.

ALTER FUNCTION name ( [ type [, ...] ] ) RENAME TO new\_name

ALTER FUNCTION name ( [ type [, ...] ] ) OWNER TO new\_owner

ALTER GROUP

Change a user group.

ALTER GROUP groupname ADD USER username [, ... ]

ALTER GROUP groupname DROP USER username [, ... ]

ALTER GROUP groupname RENAME TO new\_name

ALTER INDEX

Change the definition of an index.

ALTER INDEX name OWNER TO new\_owner

ALTER INDEX name SET TABLESPACE indexspace\_name

ALTER INDEX name RENAME TO new\_name

ALTER LANGUAGE

Change the definition of a procedural language.

ALTER LANGUAGE name RENAME TO new\_name

ALTER OPERATOR

Change the definition of an operator.

ALTER OPERATOR name ( { lefttype | NONE }, { righttype | NONE } )

OWNER TO new\_owner

ALTER OPERATOR CLASS

Change the definition of an operator class.

ALTER OPERATOR CLASS name USING index\_method RENAME TO new\_name

ALTER OPERATOR CLASS name USING index\_method OWNER TO new\_owner

ALTER SCHEMA

Change the definition of a schema.

ALTER SCHEMA name RENAME TO new\_name

ALTER SCHEMA name OWNER TO new\_owner

ALTER SEQUENCE

Change the definition of a sequence generator.

ALTER SEQUENCE name [ INCREMENT [ BY ] increment ]

[ MINVALUE minvalue | NO MINVALUE ]

[ MAXVALUE maxvalue | NO MAXVALUE ]

[ RESTART [ WITH ] start ] [ CACHE cache ] [ [ NO ] CYCLE ]

ALTER TABLE

Change the definition of a table.

ALTER TABLE [ ONLY ] name [ \* ]

action [, ... ]

ALTER TABLE [ ONLY ] name [ \* ]

RENAME [ COLUMN ] column TO new\_column

ALTER TABLE name

RENAME TO new\_name

Where *action* is one of the following lines −

ADD [ COLUMN ] column\_type [ column\_constraint [ ... ] ]

DROP [ COLUMN ] column [ RESTRICT | CASCADE ]

ALTER [ COLUMN ] column TYPE type [ USING expression ]

ALTER [ COLUMN ] column SET DEFAULT expression

ALTER [ COLUMN ] column DROP DEFAULT

ALTER [ COLUMN ] column { SET | DROP } NOT NULL

ALTER [ COLUMN ] column SET STATISTICS integer

ALTER [ COLUMN ] column SET STORAGE { PLAIN | EXTERNAL | EXTENDED | MAIN }

ADD table\_constraint

DROP CONSTRAINT constraint\_name [ RESTRICT | CASCADE ]

CLUSTER ON index\_name

SET WITHOUT CLUSTER

SET WITHOUT OIDS

OWNER TO new\_owner

SET TABLESPACE tablespace\_name

ALTER TABLESPACE

Change the definition of a tablespace.

ALTER TABLESPACE name RENAME TO new\_name

ALTER TABLESPACE name OWNER TO new\_owner

ALTER TRIGGER

Change the definition of a trigger.

ALTER TRIGGER name ON table RENAME TO new\_name

ALTER TYPE

Change the definition of a type.

ALTER TYPE name OWNER TO new\_owner

ALTER USER

Change a database user account.

ALTER USER name [ [ WITH ] option [ ... ] ]

ALTER USER name RENAME TO new\_name

ALTER USER name SET parameter { TO | = } { value | DEFAULT }

ALTER USER name RESET parameter

Where *option* can be −

[ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'

| CREATEDB | NOCREATEDB

| CREATEUSER | NOCREATEUSER

| VALID UNTIL 'abstime'

ANALYZE

Collect statistics about a database.

ANALYZE [ VERBOSE ] [ table [ (column [, ...] ) ] ]

BEGIN

Start a transaction block.

BEGIN [ WORK | TRANSACTION ] [ transaction\_mode [, ...] ]

Where *transaction\_mode* is one of −

ISOLATION LEVEL {

SERIALIZABLE | REPEATABLE READ | READ COMMITTED

| READ UNCOMMITTED

}

READ WRITE | READ ONLY

CHECKPOINT

Force a transaction log checkpoint.

CHECKPOINT

CLOSE

Close a cursor.

CLOSE name

CLUSTER

Cluster a table according to an index.

CLUSTER index\_name ON table\_name

CLUSTER table\_name

CLUSTER

COMMENT

Define or change the comment of an object.

COMMENT ON {

TABLE object\_name |

COLUMN table\_name.column\_name |

AGGREGATE agg\_name (agg\_type) |

CAST (source\_type AS target\_type) |

CONSTRAINT constraint\_name ON table\_name |

CONVERSION object\_name |

DATABASE object\_name |

DOMAIN object\_name |

FUNCTION func\_name (arg1\_type, arg2\_type, ...) |

INDEX object\_name |

LARGE OBJECT large\_object\_oid |

OPERATOR op (left\_operand\_type, right\_operand\_type) |

OPERATOR CLASS object\_name USING index\_method |

[ PROCEDURAL ] LANGUAGE object\_name |

RULE rule\_name ON table\_name |

SCHEMA object\_name |

SEQUENCE object\_name |

TRIGGER trigger\_name ON table\_name |

TYPE object\_name |

VIEW object\_name

}

IS 'text'

COMMIT

Commit the current transaction.

COMMIT [ WORK | TRANSACTION ]

COPY

Copy data between a file and a table.

COPY table\_name [ ( column [, ...] ) ]

FROM { 'filename' | STDIN }

[ WITH ]

[ BINARY ]

[ OIDS ]

[ DELIMITER [ AS ] 'delimiter' ]

[ NULL [ AS ] 'null string' ]

[ CSV [ QUOTE [ AS ] 'quote' ]

[ ESCAPE [ AS ] 'escape' ]

[ FORCE NOT NULL column [, ...] ]

COPY table\_name [ ( column [, ...] ) ]

TO { 'filename' | STDOUT }

[ [ WITH ]

[ BINARY ]

[ OIDS ]

[ DELIMITER [ AS ] 'delimiter' ]

[ NULL [ AS ] 'null string' ]

[ CSV [ QUOTE [ AS ] 'quote' ]

[ ESCAPE [ AS ] 'escape' ]

[ FORCE QUOTE column [, ...] ]

CREATE AGGREGATE

Define a new aggregate function.

CREATE AGGREGATE name (

BASETYPE = input\_data\_type,

SFUNC = sfunc,

STYPE = state\_data\_type

[, FINALFUNC = ffunc ]

[, INITCOND = initial\_condition ]

)

CREATE CAST

Define a new cast.

CREATE CAST (source\_type AS target\_type)

WITH FUNCTION func\_name (arg\_types)

[ AS ASSIGNMENT | AS IMPLICIT ]

CREATE CAST (source\_type AS target\_type)

WITHOUT FUNCTION

[ AS ASSIGNMENT | AS IMPLICIT ]

CREATE CONSTRAINT TRIGGER

Define a new constraint trigger.

CREATE CONSTRAINT TRIGGER name

AFTER events ON

table\_name constraint attributes

FOR EACH ROW EXECUTE PROCEDURE func\_name ( args )

CREATE CONVERSION

Define a new conversion.

CREATE [DEFAULT] CONVERSION name

FOR source\_encoding TO dest\_encoding FROM func\_name

CREATE DATABASE

Create a new database.

CREATE DATABASE name

[ [ WITH ] [ OWNER [=] db\_owner ]

[ TEMPLATE [=] template ]

[ ENCODING [=] encoding ]

[ TABLESPACE [=] tablespace ]

]

CREATE DOMAIN

Define a new domain.

CREATE DOMAIN name [AS] data\_type

[ DEFAULT expression ]

[ constraint [ ... ] ]

Where *constraint* is −

[ CONSTRAINT constraint\_name ]

{ NOT NULL | NULL | CHECK (expression) }

CREATE FUNCTION

Define a new function.

CREATE [ OR REPLACE ] FUNCTION name ( [ [ arg\_name ] arg\_type [, ...] ] )

RETURNS ret\_type

{ LANGUAGE lang\_name

| IMMUTABLE | STABLE | VOLATILE

| CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT | STRICT

| [ EXTERNAL ] SECURITY INVOKER | [ EXTERNAL ] SECURITY DEFINER

| AS 'definition'

| AS 'obj\_file', 'link\_symbol'

} ...

[ WITH ( attribute [, ...] ) ]

CREATE GROUP

Define a new user group.

CREATE GROUP name [ [ WITH ] option [ ... ] ]

Where option can be:

SYSID gid

| USER username [, ...]

CREATE INDEX

Define a new index.

CREATE [ UNIQUE ] INDEX name ON table [ USING method ]

( { column | ( expression ) } [ opclass ] [, ...] )

[ TABLESPACE tablespace ]

[ WHERE predicate ]

CREATE LANGUAGE

Define a new procedural language.

CREATE [ TRUSTED ] [ PROCEDURAL ] LANGUAGE name

HANDLER call\_handler [ VALIDATOR val\_function ]

CREATE OPERATOR

Define a new operator.

CREATE OPERATOR name (

PROCEDURE = func\_name

[, LEFTARG = left\_type ] [, RIGHTARG = right\_type ]

[, COMMUTATOR = com\_op ] [, NEGATOR = neg\_op ]

[, RESTRICT = res\_proc ] [, JOIN = join\_proc ]

[, HASHES ] [, MERGES ]

[, SORT1 = left\_sort\_op ] [, SORT2 = right\_sort\_op ]

[, LTCMP = less\_than\_op ] [, GTCMP = greater\_than\_op ]

)

CREATE OPERATOR CLASS

Define a new operator class.

CREATE OPERATOR CLASS name [ DEFAULT ] FOR TYPE data\_type

USING index\_method AS

{ OPERATOR strategy\_number operator\_name [ ( op\_type, op\_type ) ] [ RECHECK ]

| FUNCTION support\_number func\_name ( argument\_type [, ...] )

| STORAGE storage\_type

} [, ... ]

CREATE RULE

Define a new rewrite rule.

CREATE [ OR REPLACE ] RULE name AS ON event

TO table [ WHERE condition ]

DO [ ALSO | INSTEAD ] { NOTHING | command | ( command ; command ... ) }

CREATE SCHEMA

Define a new schema.

CREATE SCHEMA schema\_name

[ AUTHORIZATION username ] [ schema\_element [ ... ] ]

CREATE SCHEMA AUTHORIZATION username

[ schema\_element [ ... ] ]

CREATE SEQUENCE

Define a new sequence generator.

CREATE [ TEMPORARY | TEMP ] SEQUENCE name

[ INCREMENT [ BY ] increment ]

[ MINVALUE minvalue | NO MINVALUE ]

[ MAXVALUE maxvalue | NO MAXVALUE ]

[ START [ WITH ] start ] [ CACHE cache ] [ [ NO ] CYCLE ]

CREATE TABLE

Define a new table.

CREATE [ [ GLOBAL | LOCAL ] {

TEMPORARY | TEMP } ] TABLE table\_name ( {

column\_name data\_type [ DEFAULT default\_expr ] [ column\_constraint [ ... ] ]

| table\_constraint

| LIKE parent\_table [ { INCLUDING | EXCLUDING } DEFAULTS ]

} [, ... ]

)

[ INHERITS ( parent\_table [, ... ] ) ]

[ WITH OIDS | WITHOUT OIDS ]

[ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]

[ TABLESPACE tablespace ]

Where *column\_constraint* is −

[ CONSTRAINT constraint\_name ] {

NOT NULL |

NULL |

UNIQUE [ USING INDEX TABLESPACE tablespace ] |

PRIMARY KEY [ USING INDEX TABLESPACE tablespace ] |

CHECK (expression) |

REFERENCES ref\_table [ ( ref\_column ) ]

[ MATCH FULL | MATCH PARTIAL | MATCH SIMPLE ]

[ ON DELETE action ] [ ON UPDATE action ]

}

[ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]

And *table\_constraint* is −

[ CONSTRAINT constraint\_name ]

{ UNIQUE ( column\_name [, ... ] ) [ USING INDEX TABLESPACE tablespace ] |

PRIMARY KEY ( column\_name [, ... ] ) [ USING INDEX TABLESPACE tablespace ] |

CHECK ( expression ) |

FOREIGN KEY ( column\_name [, ... ] )

REFERENCES ref\_table [ ( ref\_column [, ... ] ) ]

[ MATCH FULL | MATCH PARTIAL | MATCH SIMPLE ]

[ ON DELETE action ] [ ON UPDATE action ] }

[ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]

CREATE TABLE AS

Define a new table from the results of a query.

CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } ] TABLE table\_name

[ (column\_name [, ...] ) ] [ [ WITH | WITHOUT ] OIDS ]

AS query

CREATE TABLESPACE

Define a new tablespace.

CREATE TABLESPACE tablespace\_name [ OWNER username ] LOCATION 'directory'

CREATE TRIGGER

Define a new trigger.

CREATE TRIGGER name { BEFORE | AFTER } { event [ OR ... ] }

ON table [ FOR [ EACH ] { ROW | STATEMENT } ]

EXECUTE PROCEDURE func\_name ( arguments )

CREATE TYPE

Define a new data type.

CREATE TYPE name AS

( attribute\_name data\_type [, ... ] )

CREATE TYPE name (

INPUT = input\_function,

OUTPUT = output\_function

[, RECEIVE = receive\_function ]

[, SEND = send\_function ]

[, ANALYZE = analyze\_function ]

[, INTERNALLENGTH = { internal\_length | VARIABLE } ]

[, PASSEDBYVALUE ]

[, ALIGNMENT = alignment ]

[, STORAGE = storage ]

[, DEFAULT = default ]

[, ELEMENT = element ]

[, DELIMITER = delimiter ]

)

CREATE USER

Define a new database user account.

CREATE USER name [ [ WITH ] option [ ... ] ]

Where *option* can be −

SYSID uid

| [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'

| CREATEDB | NOCREATEDB

| CREATEUSER | NOCREATEUSER

| IN GROUP group\_name [, ...]

| VALID UNTIL 'abs\_time'

CREATE VIEW

Define a new view.

CREATE [ OR REPLACE ] VIEW name [ ( column\_name [, ...] ) ] AS query

DEALLOCATE

Deallocate a prepared statement.

DEALLOCATE [ PREPARE ] plan\_name

DECLARE

Define a cursor.

DECLARE name [ BINARY ] [ INSENSITIVE ] [ [ NO ] SCROLL ]

CURSOR [ { WITH | WITHOUT } HOLD ] FOR query

[ FOR { READ ONLY | UPDATE [ OF column [, ...] ] } ]

DELETE

Delete rows of a table.

DELETE FROM [ ONLY ] table [ WHERE condition ]

DROP AGGREGATE

Remove an aggregate function.

DROP AGGREGATE name ( type ) [ CASCADE | RESTRICT ]

DROP CAST

Remove a cast.

DROP CAST (source\_type AS target\_type) [ CASCADE | RESTRICT ]

DROP CONVERSION

Remove a conversion.

DROP CONVERSION name [ CASCADE | RESTRICT ]

DROP DATABASE

Remove a database.

DROP DATABASE name

DROP DOMAIN

Remove a domain.

DROP DOMAIN name [, ...] [ CASCADE | RESTRICT ]

DROP FUNCTION

Remove a function.

DROP FUNCTION name ( [ type [, ...] ] ) [ CASCADE | RESTRICT ]

DROP GROUP

Remove a user group.

DROP GROUP name

DROP INDEX

Remove an index.

DROP INDEX name [, ...] [ CASCADE | RESTRICT ]

DROP LANGUAGE

Remove a procedural language.

DROP [ PROCEDURAL ] LANGUAGE name [ CASCADE | RESTRICT ]

DROP OPERATOR

Remove an operator.

DROP OPERATOR name ( { left\_type | NONE }, { right\_type | NONE } )

[ CASCADE | RESTRICT ]

DROP OPERATOR CLASS

Remove an operator class.

DROP OPERATOR CLASS name USING index\_method [ CASCADE | RESTRICT ]

DROP RULE

Remove a rewrite rule.

DROP RULE name ON relation [ CASCADE | RESTRICT ]

DROP SCHEMA

Remove a schema.

DROP SCHEMA name [, ...] [ CASCADE | RESTRICT ]

DROP SEQUENCE

Remove a sequence.

DROP SEQUENCE name [, ...] [ CASCADE | RESTRICT ]

DROP TABLE

Remove a table.

DROP TABLE name [, ...] [ CASCADE | RESTRICT ]

DROP TABLESPACE

Remove a tablespace.

DROP TABLESPACE tablespace\_name

DROP TRIGGER

Remove a trigger.

DROP TRIGGER name ON table [ CASCADE | RESTRICT ]

DROP TYPE

Remove a data type.

DROP TYPE name [, ...] [ CASCADE | RESTRICT ]

DROP USER

Remove a database user account.

DROP USER name

DROP VIEW

Remove a view.

DROP VIEW name [, ...] [ CASCADE | RESTRICT ]

END

Commit the current transaction.

END [ WORK | TRANSACTION ]

EXECUTE

Execute a prepared statement.

EXECUTE plan\_name [ (parameter [, ...] ) ]

EXPLAIN

Show the execution plan of a statement.

EXPLAIN [ ANALYZE ] [ VERBOSE ] statement

FETCH

Retrieve rows from a query using a cursor.

FETCH [ direction { FROM | IN } ] cursor\_name

Where *direction* can be empty or one of −

NEXT

PRIOR

FIRST

LAST

ABSOLUTE count

RELATIVE count

count

ALL

FORWARD

FORWARD count

FORWARD ALL

BACKWARD

BACKWARD count

BACKWARD ALL

GRANT

Define access privileges.

GRANT { { SELECT | INSERT | UPDATE | DELETE | RULE | REFERENCES | TRIGGER }

[,...] | ALL [ PRIVILEGES ] }

ON [ TABLE ] table\_name [, ...]

TO { username | GROUP group\_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { { CREATE | TEMPORARY | TEMP } [,...] | ALL [ PRIVILEGES ] }

ON DATABASE db\_name [, ...]

TO { username | GROUP group\_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { CREATE | ALL [ PRIVILEGES ] }

ON TABLESPACE tablespace\_name [, ...]

TO { username | GROUP group\_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }

ON FUNCTION func\_name ([type, ...]) [, ...]

TO { username | GROUP group\_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { USAGE | ALL [ PRIVILEGES ] }

ON LANGUAGE lang\_name [, ...]

TO { username | GROUP group\_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]

GRANT { { CREATE | USAGE } [,...] | ALL [ PRIVILEGES ] }

ON SCHEMA schema\_name [, ...]

TO { username | GROUP group\_name | PUBLIC } [, ...] [ WITH GRANT OPTION ]

INSERT

Create new rows in a table.

INSERT INTO table [ ( column [, ...] ) ]

{ DEFAULT VALUES | VALUES ( { expression | DEFAULT } [, ...] ) | query }

LISTEN

Listen for a notification.

LISTEN name

LOAD

Load or reload a shared library file.

LOAD 'filename'

LOCK

Lock a table.

LOCK [ TABLE ] name [, ...] [ IN lock\_mode MODE ] [ NOWAIT ]

Where *lock\_mode* is one of −

ACCESS SHARE | ROW SHARE | ROW EXCLUSIVE | SHARE UPDATE EXCLUSIVE

| SHARE | SHARE ROW EXCLUSIVE | EXCLUSIVE | ACCESS EXCLUSIVE

MOVE

Position a cursor.

MOVE [ direction { FROM | IN } ] cursor\_name

NOTIFY

Generate a notification.

NOTIFY name

PREPARE

Prepare a statement for execution.

PREPARE plan\_name [ (data\_type [, ...] ) ] AS statement

REINDEX

Rebuild indexes.

REINDEX { DATABASE | TABLE | INDEX } name [ FORCE ]

RELEASE SAVEPOINT

Destroy a previously defined savepoint.

RELEASE [ SAVEPOINT ] savepoint\_name

RESET

Restore the value of a runtime parameter to the default value.

RESET name

RESET ALL

REVOKE

Remove access privileges.

REVOKE [ GRANT OPTION FOR ]

{ { SELECT | INSERT | UPDATE | DELETE | RULE | REFERENCES | TRIGGER }

[,...] | ALL [ PRIVILEGES ] }

ON [ TABLE ] table\_name [, ...]

FROM { username | GROUP group\_name | PUBLIC } [, ...]

[ CASCADE | RESTRICT ]

REVOKE [ GRANT OPTION FOR ]

{ { CREATE | TEMPORARY | TEMP } [,...] | ALL [ PRIVILEGES ] }

ON DATABASE db\_name [, ...]

FROM { username | GROUP group\_name | PUBLIC } [, ...]

[ CASCADE | RESTRICT ]

REVOKE [ GRANT OPTION FOR ]

{ CREATE | ALL [ PRIVILEGES ] }

ON TABLESPACE tablespace\_name [, ...]

FROM { username | GROUP group\_name | PUBLIC } [, ...]

[ CASCADE | RESTRICT ]

REVOKE [ GRANT OPTION FOR ]

{ EXECUTE | ALL [ PRIVILEGES ] }

ON FUNCTION func\_name ([type, ...]) [, ...]

FROM { username | GROUP group\_name | PUBLIC } [, ...]

[ CASCADE | RESTRICT ]

REVOKE [ GRANT OPTION FOR ]

{ USAGE | ALL [ PRIVILEGES ] }

ON LANGUAGE lang\_name [, ...]

FROM { username | GROUP group\_name | PUBLIC } [, ...]

[ CASCADE | RESTRICT ]

REVOKE [ GRANT OPTION FOR ]

{ { CREATE | USAGE } [,...] | ALL [ PRIVILEGES ] }

ON SCHEMA schema\_name [, ...]

FROM { username | GROUP group\_name | PUBLIC } [, ...]

[ CASCADE | RESTRICT ]

ROLLBACK

Abort the current transaction.

ROLLBACK [ WORK | TRANSACTION ]

ROLLBACK TO SAVEPOINT

Roll back to a savepoint.

ROLLBACK [ WORK | TRANSACTION ] TO [ SAVEPOINT ] savepoint\_name

SAVEPOINT

Define a new savepoint within the current transaction.

SAVEPOINT savepoint\_name

SELECT

Retrieve rows from a table or view.

SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]

\* | expression [ AS output\_name ] [, ...]

[ FROM from\_item [, ...] ]

[ WHERE condition ]

[ GROUP BY expression [, ...] ]

[ HAVING condition [, ...] ]

[ { UNION | INTERSECT | EXCEPT } [ ALL ] select ]

[ ORDER BY expression [ ASC | DESC | USING operator ] [, ...] ]

[ LIMIT { count | ALL } ]

[ OFFSET start ]

[ FOR UPDATE [ OF table\_name [, ...] ] ]

Where *from\_item* can be one of:

[ ONLY ] table\_name [ \* ] [ [ AS ] alias [ ( column\_alias [, ...] ) ] ]

( select ) [ AS ] alias [ ( column\_alias [, ...] ) ]

function\_name ( [ argument [, ...] ] )

[ AS ] alias [ ( column\_alias [, ...] | column\_definition [, ...] ) ]

function\_name ( [ argument [, ...] ] ) AS ( column\_definition [, ...] )

from\_item [ NATURAL ] join\_type from\_item

[ ON join\_condition | USING ( join\_column [, ...] ) ]

SELECT INTO

Define a new table from the results of a query.

SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]

\* | expression [ AS output\_name ] [, ...]

INTO [ TEMPORARY | TEMP ] [ TABLE ] new\_table

[ FROM from\_item [, ...] ]

[ WHERE condition ]

[ GROUP BY expression [, ...] ]

[ HAVING condition [, ...] ]

[ { UNION | INTERSECT | EXCEPT } [ ALL ] select ]

[ ORDER BY expression [ ASC | DESC | USING operator ] [, ...] ]

[ LIMIT { count | ALL } ]

[ OFFSET start ]

[ FOR UPDATE [ OF table\_name [, ...] ] ]

SET

Change a runtime parameter.

SET [ SESSION | LOCAL ] name { TO | = } { value | 'value' | DEFAULT }

SET [ SESSION | LOCAL ] TIME ZONE { time\_zone | LOCAL | DEFAULT }

SET CONSTRAINTS

Set constraint checking modes for the current transaction.

SET CONSTRAINTS { ALL | name [, ...] } { DEFERRED | IMMEDIATE }

SET SESSION AUTHORIZATION

Set the session user identifier and the current user identifier of the current session.

SET [ SESSION | LOCAL ] SESSION AUTHORIZATION username

SET [ SESSION | LOCAL ] SESSION AUTHORIZATION DEFAULT

RESET SESSION AUTHORIZATION

SET TRANSACTION

Set the characteristics of the current transaction.

SET TRANSACTION transaction\_mode [, ...]

SET SESSION CHARACTERISTICS AS TRANSACTION transaction\_mode [, ...]

Where *transaction\_mode* is one of −

ISOLATION LEVEL { SERIALIZABLE | REPEATABLE READ | READ COMMITTED

| READ UNCOMMITTED }

READ WRITE | READ ONLY

SHOW

Show the value of a runtime parameter.

SHOW name

SHOW ALL

START TRANSACTION

Start a transaction block.

START TRANSACTION [ transaction\_mode [, ...] ]

Where *transaction\_mode* is one of −

ISOLATION LEVEL { SERIALIZABLE | REPEATABLE READ | READ COMMITTED

| READ UNCOMMITTED }

READ WRITE | READ ONLY

TRUNCATE

Empty a table.

TRUNCATE [ TABLE ] name

UNLISTEN

Stop listening for a notification.

UNLISTEN { name | \* }

UPDATE

Update rows of a table.

UPDATE [ ONLY ] table SET column = { expression | DEFAULT } [, ...]

[ FROM from\_list ]

[ WHERE condition ]

VACUUM

Garbage-collect and optionally analyze a database.

VACUUM [ FULL ] [ FREEZE ] [ VERBOSE ] [ table ]

VACUUM [ FULL ] [ FREEZE ] [ VERBOSE ] ANALYZE [ table [ (column [, ...]