Command: **ABORT**
Description: **abort** the current **transaction**
Syntax:
ABORT [**WORK** | **TRANSACTION**]

Command: **ALTER GROUP**
Description: **add** users **to** a **group or** remove users **from** a **group**
Syntax:
ALTER GROUP name **ADD USER** username [, ...]
ALTER GROUP name **DROP USER** username [, ...]

Command: **ALTER TABLE**
Description: change the definition **of** a **table**
Syntax:
ALTER TABLE [**ONLY**] **table** [*]
ADD [**COLUMN**] **column type** [column_constraint [...]]
ALTER TABLE [**ONLY**] **table** [*]
ALTER [**COLUMN**] **column { SET DEFAULT** value | **DROP DEFAULT }**
ALTER TABLE [**ONLY**] **table** [*]
ALTER [**COLUMN**] **column SET STATISTICS** integer
ALTER TABLE [**ONLY**] **table** [*]
RENAME [**COLUMN**] **column TO** newcolumn
ALTER TABLE table
RENAME TO new_table
ALTER TABLE table
ADD table_constraint_definition
ALTER TABLE [**ONLY**] **table**
DROP CONSTRAINT constraint { RESTRICT | **CASCADE }**
ALTER TABLE table
OWNER TO new_owner

Command: **ALTER USER**
Description: change a **database user** account
Syntax:
ALTER USER username [[**WITH**] **option** [...]]

where option can be:

[**ENCRYPTED** | **UNENCRYPTED**] **PASSWORD** ´**password**´
| **CREATEDB** | **NOCREATEDB**
| **CREATEUSER** | **NOCREATEUSER**
| **VALID UNTIL** ´abstime´

Command: **ANALYZE**
Description: collect **statistics** about a **database**
Syntax:
ANALYZE [**VERBOSE**] [**table** [(**column** [, ...])]]

Command: **BEGIN**
Description: **start** a **transaction** block

Syntax:
BEGIN [**WORK** | **TRANSACTION**]

Command: **CHECKPOINT**
Description: **force** a **transaction** log **checkpoint**
Syntax:
CHECKPOINT

Command: **CLOSE**
Description: **close** a **cursor**
Syntax:
CLOSE cursor

Command: **CLUSTER**
Description: **cluster** a **table** according **to** an **index**
Syntax:
CLUSTER indexname **ON** tablename

Command: **COMMENT**
Description: define **or** change the **comment of** an object
Syntax:
COMMENT ON
[
[**DATABASE** | **INDEX** | **RULE** | **SEQUENCE** | **TABLE** | **TYPE** | **VIEW**] object_name |
COLUMN table_name.column_name |
AGGREGATE agg_name (agg_type) |
FUNCTION func_name (arg1, arg2, ...) |
OPERATOR op (leftoperand_type rightoperand_type) |
TRIGGER trigger_name **ON** table_name
] **IS** ´text´

Command: **COMMIT**
Description: **commit** the current **transaction**
Syntax:
COMMIT [**WORK** | **TRANSACTION**]

Command: **COPY**
Description: **copy** data **between** files **and** tables
Syntax:
COPY [**BINARY**] **table** [**WITH OIDS**]
FROM { ´filename´ | **stdin }**
[[**USING**] **DELIMITERS** ´delimiter´]
[**WITH NULL AS** ´**null** string´]
COPY [**BINARY**] **table** [**WITH OIDS**]
TO { ´filename´ | **stdout }**
[[**USING**] **DELIMITERS** ´delimiter´]
[**WITH NULL AS** ´**null** string´]

Command: **CREATE AGGREGATE**
Description: define a **new aggregate function**

Syntax:
CREATE AGGREGATE name (BASETYPE = input_data_type,
SFUNC = sfunc, STYPE = state_type
[, FINALFUNC = ffunc]
[, INITCOND = initial_condition])

Command: **CREATE CONSTRAINT TRIGGER**
Description: define a **new constraint trigger**
Syntax:
CREATE CONSTRAINT TRIGGER name
AFTER events **ON**
relation **constraint** attributes
FOR EACH ROW EXECUTE PROCEDURE func ´(´ args ´)´

Command: **CREATE DATABASE**
Description: **create** a **new database**
Syntax:
CREATE DATABASE name
[**WITH** [**LOCATION** = ´dbpath´]
[**TEMPLATE** = **template**]
[**ENCODING** = **encoding**]]

Command: **CREATE FUNCTION**
Description: define a **new function**
Syntax:
CREATE [**OR REPLACE**] **FUNCTION** name ([argtype [, ...]])
RETURNS rettype
AS ´definition´
LANGUAGE langname
[**WITH** (attribute [, ...])]
CREATE [**OR REPLACE**] **FUNCTION** name ([argtype [, ...]])
RETURNS rettype
AS ´obj_file´, ´link_symbol´
LANGUAGE langname
[**WITH** (attribute [, ...])]

Command: **CREATE GROUP**
Description: define a **new user group**
Syntax:
CREATE GROUP name [[**WITH**] **option** [...]]

where option can be:

SYSID gid
| **USER** username [, ...]

Command: **CREATE INDEX**
Description: define a **new index**
Syntax:
CREATE [**UNIQUE**] **INDEX** index_name **ON table**

[**USING** acc_method] (**column** [ops_name] [, ...])
[**WHERE** predicate]
CREATE [**UNIQUE**] **INDEX** index_name **ON table**
[**USING** acc_method] (func_name(**column** [, ...]) [ops_name])
[**WHERE** predicate]

Command: **CREATE LANGUAGE**
Description: define a **new procedural language**
Syntax:
CREATE [**TRUSTED**] [**PROCEDURAL**] **LANGUAGE** langname
HANDLER call_handler

Command: **CREATE OPERATOR**
Description: define a **new operator**
Syntax:
CREATE OPERATOR name (**PROCEDURE** = func_name
[, LEFTARG = lefttype
] [, RIGHTARG = righttype]
[, COMMUTATOR = com_op] [, NEGATOR = neg_op]
[, **RESTRICT** = res_proc] [, **JOIN** = join_proc]
[, HASHES] [, SORT1 = left_sort_op] [, SORT2 = right_sort_op])

Command: **CREATE RULE**
Description: define a **new** rewrite **rule**
Syntax:
CREATE RULE name **AS ON** event
TO object [**WHERE** condition]
DO [**INSTEAD**] **action**

where action can be:

NOTHING
|
query
|
(query ; query ...)
|
[query ; query ...]

Command: **CREATE SEQUENCE**
Description: define a **new sequence** generator
Syntax:
CREATE [**TEMPORARY** | **TEMP**] **SEQUENCE** seqname [**INCREMENT increment**]
[**MINVALUE minvalue**] [**MAXVALUE maxvalue**]
[**START start**] [**CACHE cache**] [**CYCLE**]

Command: **CREATE TABLE**
Description: define a **new table**
Syntax:
CREATE [[**LOCAL**] **{ TEMPORARY** | **TEMP }**] **TABLE** table_name (

{ column_name data_type [**DEFAULT** default_expr] [column_constraint [, ...]]
| table_constraint **}** [, ...]
)
[**INHERITS** (parent_table [, ...])]
[**WITH OIDS** | **WITHOUT OIDS**]

where column_constraint **is**:

[**CONSTRAINT** constraint_name]
{ NOT NULL | **NULL** | **UNIQUE** | **PRIMARY KEY** |
CHECK (expression) |
REFERENCES reftable [(refcolumn)] [**MATCH FULL** | **MATCH PARTIAL**]
[**ON DELETE action**] [**ON UPDATE action**] **}**
[**DEFERRABLE** | **NOT DEFERRABLE**] [**INITIALLY DEFERRED** | **INITIALLY IMMEDIATE**

and table_constraint **is**:

[**CONSTRAINT** constraint_name]
{ UNIQUE (column_name [, ...]) |
PRIMARY KEY (column_name [, ...]) |
CHECK (expression) |
FOREIGN KEY (column_name [, ...]) **REFERENCES** reftable [(refcolumn [, ...])]
[**MATCH FULL** | **MATCH PARTIAL**] [**ON DELETE action**] [**ON UPDATE action**] **}**
[**DEFERRABLE** | **NOT DEFERRABLE**] [**INITIALLY DEFERRED** | **INITIALLY IMMEDIATE**

Command: **CREATE TABLE AS**
Description: **create** a **new table from** the results **of** a query
Syntax:
CREATE [[**LOCAL**] **{ TEMPORARY** | **TEMP }]** **TABLE** table_name [(column_name [, ...])]
AS query

Command: **CREATE TRIGGER**
Description: define a **new trigger**
Syntax:
CREATE TRIGGER name **{ BEFORE** | **AFTER } { event** [**OR** ...] **}**
ON table FOR EACH { ROW | **STATEMENT }**
EXECUTE PROCEDURE func (arguments)

Command: **CREATE TYPE**
Description: define a **new** data **type**
Syntax:
CREATE TYPE typename (INPUT = input_function, OUTPUT = output_function
, INTERNALLENGTH = **{** internallength | VARIABLE **}**
[, EXTERNALLENGTH = **{** externallength | VARIABLE **}**]
[, **DEFAULT** = **default**]
[, ELEMENT = element] [, DELIMITER = delimiter]
[, SEND = send_function] [, RECEIVE = receive_function]
[, PASSEDBYVALUE]
[, ALIGNMENT = alignment]
[, STORAGE = storage]

)

Command: **CREATE USER**
Description: define a **new database user** account
Syntax:
CREATE USER username [[**WITH**] **option** [...]]

where option can be:

SYSID uid
| [**ENCRYPTED** | **UNENCRYPTED**] **PASSWORD** ´**password**´
| **CREATEDB** | **NOCREATEDB**
| **CREATEUSER** | **NOCREATEUSER**
| **IN GROUP** groupname [, ...]
| **VALID UNTIL** ´abstime´

Command: **CREATE VIEW**
Description: define a **new view**
Syntax:
CREATE VIEW view [(**column** name list)] **AS SELECT** query

Command: **DECLARE**
Description: define a **cursor**
Syntax:
DECLARE cursorname [**BINARY**] [**INSENSITIVE**] [**SCROLL**]
CURSOR FOR query
[**FOR { READ ONLY** | **UPDATE** [**OF column** [, ...]]]

Command: **DELETE**
Description: **delete** rows **of** a **table**
Syntax:
DELETE FROM [**ONLY**] **table** [**WHERE** condition]

Command: **DROP AGGREGATE**
Description: remove a **user**-defined **aggregate function**
Syntax:
DROP AGGREGATE name (**type**)

Command: **DROP DATABASE**
Description: remove a **database**
Syntax:
DROP DATABASE name

Command: **DROP FUNCTION**
Description: remove a **user**-defined **function**
Syntax:
DROP FUNCTION name ([**type** [, ...]])

Command: **DROP GROUP**
Description: remove a **user group**

Syntax:
DROP GROUP name

Command: **DROP INDEX**
Description: remove an **index**
Syntax:
DROP INDEX index_name [, ...]

Command: **DROP LANGUAGE**
Description: remove a **user**-defined **procedural language**
Syntax:
DROP [**PROCEDURAL**] **LANGUAGE** name

Command: **DROP OPERATOR**
Description: remove a **user**-defined **operator**
Syntax:
DROP OPERATOR id (lefttype | **NONE** , righttype | **NONE**)

Command: **DROP RULE**
Description: remove a rewrite **rule**
Syntax:
DROP RULE name [, ...]

Command: **DROP SEQUENCE**
Description: remove a **sequence**
Syntax:
DROP SEQUENCE name [, ...]

Command: **DROP TABLE**
Description: remove a **table**
Syntax:
DROP TABLE name [, ...]

Command: **DROP TRIGGER**
Description: remove a **trigger**
Syntax:
DROP TRIGGER name **ON table**

Command: **DROP TYPE**
Description: remove a **user**-defined data **type**
Syntax:
DROP TYPE typename [, ...]

Command: **DROP USER**
Description: remove a **database user** account
Syntax:
DROP USER name

Command: **DROP VIEW**
Description: remove a **view**

Syntax:
DROP VIEW name [, ...]

Command: **END**
Description: **commit** the current **transaction**
Syntax:
END [**WORK** | **TRANSACTION**]

Command: **EXPLAIN**
Description: **show** the execution plan **of** a **statement**
Syntax:
EXPLAIN [**ANALYZE**] [**VERBOSE**] query

Command: **FETCH**
Description: retrieve rows **from** a **table using** a **cursor**
Syntax:
FETCH [direction] [count] **{ IN** | **FROM } cursor**
FETCH [**FORWARD** | **BACKWARD** | **RELATIVE**] [# | **ALL** | **NEXT** | **PRIOR**] **{ IN** | **FROM**

Command: **GRANT**
Description: define **access privileges**
Syntax:
GRANT { { SELECT | **INSERT** | **UPDATE** | **DELETE** | **RULE** | **REFERENCES** | **TRIGGER } [**...
ON [**TABLE**] objectname [, ...]
TO { username | **GROUP** groupname | **PUBLIC }** [, ...]

Command: **INSERT**
Description: **create new** rows **in** a **table**
Syntax:
INSERT INTO table [(**column** [, ...])]
{ DEFAULT VALUES | **VALUES** (expression [, ...]) | **SELECT** query **}**

Command: **LISTEN**
Description: **listen for** a notification
Syntax:
LISTEN name

Command: **LOAD**
Description: **load or** reload a shared library file
Syntax:
LOAD ´filename´

Command: **LOCK**
Description: explicitly **lock** a **table**
Syntax:
LOCK [**TABLE**] name [, ...]
LOCK [**TABLE**] name [, ...] **IN** lockmode **MODE**

where lockmode **is** one **of**:

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

Command: **MOVE**
Description: **position** a **cursor on** a specified **row of** a **table**
Syntax:
MOVE [direction] [count]
{ IN | FROM } cursor

Command: **NOTIFY**
Description: generate a notification
Syntax:
NOTIFY name

Command: **REINDEX**
Description: rebuild corrupted indexes
Syntax:
REINDEX { TABLE | DATABASE | INDEX } name [**FORCE**]

Command: **RESET**
Description: restore the value **of** a run-**time** parameter **to** a **default** value
Syntax:
RESET variable

Command: **REVOKE**
Description: remove **access privileges**
Syntax:
REVOKE { { SELECT | INSERT | UPDATE | DELETE | RULE | REFERENCES | TRIGGER } [
ON [**TABLE**] object [, ...]
FROM { username | **GROUP** groupname | **PUBLIC }** [, ...]

Command: **ROLLBACK**
Description: **abort** the current **transaction**
Syntax:
ROLLBACK [**WORK** | **TRANSACTION**]

Command: **SELECT**
Description: retrieve rows **from** a **table or view**
Syntax:
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**] [, ...]]
[**FOR UPDATE** [**OF** tablename [, ...]]]
[**LIMIT {** count | **ALL } **]
[**OFFSET start**]

where from_item can be:

[**ONLY**] table_name [*]
[[**AS**] alias [(column_alias_list)]]
|
(**select**)
[**AS**] alias [(column_alias_list)]
|
from_item [**NATURAL**] join_type from_item
[**ON** join_condition | **USING** (join_column_list)]

Command: **SELECT INTO**
Description: **create** a **new table from** the results **of** a query
Syntax:
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**] [, ...]]
[**FOR UPDATE** [**OF** tablename [, ...]]]
[**LIMIT** [**start** ,] **{** count | **ALL } **]
[**OFFSET start**]

where from_item can be:

[**ONLY**] table_name [*]
[[**AS**] alias [(column_alias_list)]]
|
(**select**)
[**AS**] alias [(column_alias_list)]
|
from_item [**NATURAL**] join_type from_item
[**ON** join_condition | **USING** (join_column_list)]

Command: **SET**
Description: change a run-**time** parameter
Syntax:
SET variable **{ TO** | = **} {** value | ´value´ | **DEFAULT }**
SET TIME ZONE { ´timezone´ | **LOCAL** | **DEFAULT }**

Command: **SET CONSTRAINTS**
Description: **set** the **constraint mode of** the current **transaction**
Syntax:
SET CONSTRAINTS { ALL | **constraint** [, ...] **} { DEFERRED** | **IMMEDIATE }**

Command: **SET SESSION AUTHORIZATION**
Description: **set** the **session user** identifier **and** the current **user** identifier **of** the current **session**
Syntax:
SET SESSION AUTHORIZATION ´username´

Command: **SET TRANSACTION**
Description: **set** the **characteristics of** the current **transaction**
Syntax:
SET TRANSACTION ISOLATION LEVEL { READ COMMITTED | SERIALIZABLE }
SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL
{ READ COMMITTED | SERIALIZABLE }

Command: **SHOW**
Description: **show** the value **of** a run-**time** parameter
Syntax:
SHOW name

Command: **TRUNCATE**
Description: empty a **table**
Syntax:
TRUNCATE [**TABLE**] name

Command: **UNLISTEN**
Description: stop listening **for** a notification
Syntax:
UNLISTEN { notifyname | * **}**

Command: **UPDATE**
Description: **update** rows **of** a **table**
Syntax:
UPDATE [**ONLY**] **table SET** col = expression [, ...]
[**FROM** fromlist]
[**WHERE** condition]

Command: **VACUUM**
Description: garbage-collect **and** optionally **analyze** a **database**
Syntax:
VACUUM [**FULL**] [**FREEZE**] [**VERBOSE**] [**table**]
VACUUM [**FULL**] [**FREEZE**] [**VERBOSE**] **ANALYZE** [**table** [(**column** [, ...])]]