PostgreSQL commands

Access the PostgreSQL server from **psql** with a specific user:

|  |  |
| --- | --- |
| 1 | psql -U [username]; |

For example, the following command uses the postgres user to access the PostgreSQL database server:

|  |  |
| --- | --- |
| 1 | psql -U postgres |

Connect to a specific database:

|  |  |
| --- | --- |
| 1 | \c database\_name; |

For example, the following command connects to the dvdrental database:

|  |  |
| --- | --- |
| 1  2 | \c dvdrental;  You are now connected to database "dvdrental" as user "postgres". |

To quit the psql:

|  |  |
| --- | --- |
| 1 | \q |

[List all databases](http://www.postgresqltutorial.com/postgresql-show-databases/) in the PostgreSQL database server

|  |  |
| --- | --- |
| 1 | \l |

List all schemas:

|  |  |
| --- | --- |
| 1 | \dn |

List all [stored procedures](http://www.postgresqltutorial.com/postgresql-stored-procedures/) and functions:

|  |  |
| --- | --- |
| 1 | \df |

List all [views](http://www.postgresqltutorial.com/postgresql-views/):

|  |  |
| --- | --- |
| 1 | \dv |

[Lists all tables](http://www.postgresqltutorial.com/postgresql-show-tables/) in a current database.

|  |  |
| --- | --- |
| 1 | \dt |

Or to get more information on tables in the current database:

|  |  |
| --- | --- |
| 1 | \dt+ |

Get detailed information on a table.

|  |  |
| --- | --- |
| 1 | \d+ table\_name |

Show a [stored procedure](http://www.postgresqltutorial.com/postgresql-stored-procedures/) or function code:

|  |  |
| --- | --- |
| 1 | \df+ function\_name |

Show query output in the pretty-format:

|  |  |
| --- | --- |
| 1 | \x |

List all users:

|  |  |
| --- | --- |
| 1 | \du |

Create a new [role](http://www.postgresqltutorial.com/postgresql-roles/):

|  |  |
| --- | --- |
| 1 | CREATE ROLE role\_name; |

Create a new role with a username and password:

|  |  |
| --- | --- |
| 1 | CREATE ROLE username NOINHERIT LOGIN PASSWORD password; |

Change role for the current session to the new\_role:

|  |  |
| --- | --- |
| 1 | SET ROLE new\_role; |

Allow role\_1 to set its role as role\_2:

|  |  |
| --- | --- |
| 1 | GRANT role\_2 TO role\_1; |

Managing databases

[Create a new database](http://www.postgresqltutorial.com/postgresql-create-database/):

|  |  |
| --- | --- |
| 1 | CREATE DATABASE [IF NOT EXISTS] db\_name; |

[Delete a database permanently](http://www.postgresqltutorial.com/postgresql-drop-database/):

|  |  |
| --- | --- |
| 1 | DROP DATABASE [IF EXISTS] db\_name; |

Managing tables

[Create a new table](http://www.postgresqltutorial.com/postgresql-create-table/) or a [temporary table](http://www.postgresqltutorial.com/postgresql-temporary-table/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | CREATE [TEMP] TABLE [IF NOT EXISTS] table\_name(     pk SERIAL PRIMARY KEY,     c1 type(size) NOT NULL,     c2 type(size) NULL,     ...  ); |

[Add a new column](http://www.postgresqltutorial.com/postgresql-add-column/) to a table:

|  |  |
| --- | --- |
| 1 | ALTER TABLE table\_name ADD COLUMN new\_column\_name TYPE; |

[Drop a column](http://www.postgresqltutorial.com/postgresql-drop-column/) in a table:

|  |  |
| --- | --- |
| 1 | ALTER TABLE table\_name DROP COLUMN column\_name; |

[Rename a column](http://www.postgresqltutorial.com/postgresql-rename-column/):

|  |  |
| --- | --- |
| 1 | ALTER TABLE table\_name RENAME column\_name TO new\_column\_name; |

Set or remove a default value for a column:

|  |  |
| --- | --- |
| 1 | ALTER TABLE table\_name ALTER COLUMN [SET DEFAULT value | DROP DEFAULT] |

Add a [primary key](http://www.postgresqltutorial.com/postgresql-primary-key/)to a table.

|  |  |
| --- | --- |
| 1 | ALTER TABLE table\_name ADD PRIMARY KEY (column,...); |

Remove the primary key from a table.

|  |  |
| --- | --- |
| 1  2 | ALTER TABLE table\_name  DROP CONSTRAINT primary\_key\_constraint\_name; |

[Rename a table](http://www.postgresqltutorial.com/postgresql-rename-table/).

|  |  |
| --- | --- |
| 1 | ALTER TABLE table\_name RENAME TO new\_table\_name; |

[Drop a table](http://www.postgresqltutorial.com/postgresql-drop-table/) and its dependent objects:

|  |  |
| --- | --- |
| 1 | DROP TABLE [IF EXISTS] table\_name CASCADE; |

Managing views

[Create a view](http://www.postgresqltutorial.com/managing-postgresql-views/):

|  |  |
| --- | --- |
| 1  2 | CREATE OR REPLACE view\_name AS  query; |

[Create a recursive view](http://www.postgresqltutorial.com/postgresql-recursive-view/):

|  |  |
| --- | --- |
| 1  2 | CREATE RECURSIVE VIEW view\_name(columns) AS  SELECT columns; |

[Create a materialized view](http://www.postgresqltutorial.com/postgresql-materialized-views/):

|  |  |
| --- | --- |
| 1  2  3  4 | CREATE MATERIALIZED VIEW view\_name  AS  query  WITH [NO] DATA; |

Refresh a materialized view:

|  |  |
| --- | --- |
| 1 | REFRESH MATERIALIZED VIEW CONCURRENTLY view\_name; |

Drop a view:

|  |  |
| --- | --- |
| 1 | DROP VIEW [ IF EXISTS ] view\_name; |

Drop a materialized view:

|  |  |
| --- | --- |
| 1 | DROP MATERIALIZED VIEW view\_name; |

Rename a view:

|  |  |
| --- | --- |
| 1 | ALTER VIEW view\_name RENAME TO new\_name; |

Managing indexes

Creating an index with the specified name on a table

|  |  |
| --- | --- |
| 1  2 | CREATE [UNIQUE] INDEX index\_name  ON table (column,...) |

Removing a specified index from a table

|  |  |
| --- | --- |
| 1 | DROP INDEX index\_name; |

Querying data from tables

Query all data from a table:

|  |  |
| --- | --- |
| 1 | SELECT \* FROM table\_name; |

Query data from specified columns of all rows in a table:

|  |  |
| --- | --- |
| 1  2 | SELECT column, column2….  FROM table; |

Query data and select only unique rows:

|  |  |
| --- | --- |
| 1  2 | SELECT DISTINCT (column)  FROM table; |

Query data from a table with a filter:

|  |  |
| --- | --- |
| 1  2  3 | SELECT \*  FROM table  WHERE condition; |

Assign an [alias](http://www.postgresqltutorial.com/postgresql-alias/) to a column in the result set:

|  |  |
| --- | --- |
| 1  2 | SELECT column\_1 AS new\_column\_1, ...  FROM table; |

Query data using the [LIKE](http://www.postgresqltutorial.com/postgresql-like/) operator:

|  |  |
| --- | --- |
| 1  2 | SELECT \* FROM table\_name  WHERE column LIKE '%value%' |

Query data using the [BETWEEN](http://www.postgresqltutorial.com/postgresql-between/) operator:

|  |  |
| --- | --- |
| 1  2 | SELECT \* FROM table\_name  WHERE column BETWEEN low AND high; |

Query data using the [IN](http://www.postgresqltutorial.com/postgresql-in/) operator:

|  |  |
| --- | --- |
| 1  2 | SELECT \* FROM table\_name  WHERE column IN (value1, value2,...); |

Constrain the returned rows with the [LIMIT](http://www.postgresqltutorial.com/postgresql-limit/) clause:

|  |  |
| --- | --- |
| 1  2  3 | SELECT \* FROM table\_name  LIMIT limit OFFSET offset  ORDER BY column\_name; |

Query data from multiple using the [inner join](http://www.postgresqltutorial.com/postgresql-inner-join/), [left join](http://www.postgresqltutorial.com/postgresql-left-join/), [full outer join](http://www.postgresqltutorial.com/postgresql-full-outer-join/), [cross join](http://www.postgresqltutorial.com/postgresql-cross-join/) and [natural join](http://www.postgresqltutorial.com/postgresql-natural-join/):

|  |  |
| --- | --- |
| 1  2  3 | SELECT \*  FROM table1  INNER JOIN table2 ON conditions |
| 1  2  3 | SELECT \*  FROM table1  LEFT JOIN table2 ON conditions |

|  |  |
| --- | --- |
| 1  2  3 | SELECT \*  FROM table1  FULL OUTER JOIN table2 ON conditions |
| 1  2  3 | SELECT \*  FROM table1  CROSS JOIN table2; |

|  |  |
| --- | --- |
| 1  2  3 | SELECT \*  FROM table1  NATURAL JOIN table2; |

Return the number of rows of a table.

|  |  |
| --- | --- |
| 1  2 | SELECT COUNT (\*)  FROM table\_name; |

Sort rows in ascending or descending order

|  |  |
| --- | --- |
| 1  2  3 | SELECT column, column2, ...  FROM table  ORDER BY column ASC [DESC], column2 ASC [DESC],...; |

Group rows using [GROUP BY](http://www.postgresqltutorial.com/postgresql-group-by/) clause.

|  |  |
| --- | --- |
| 1  2  3 | SELECT \*  FROM table  GROUP BY column\_1, column\_2, ...; |

Filter groups using the [HAVING](http://www.postgresqltutorial.com/postgresql-having/) clause.

|  |  |
| --- | --- |
| 1  2  3  4 | SELECT \*  FROM table  GROUP BY column\_1  HAVING condition; |

Set operations

Combine the result set of two or more queries with [UNION](http://www.postgresqltutorial.com/postgresql-union/) operator:

|  |  |
| --- | --- |
| 1  2  3 | SELECT \* FROM table1  UNION  SELECT \* FROM table2; |

Minus a result set using [EXCEPT](http://www.postgresqltutorial.com/postgresql-tutorial/postgresql-except/) operator:

|  |  |
| --- | --- |
| 1  2  3 | SELECT \* FROM table1  EXCEPT  SELECT \* FROM table2; |

Get intersection of the result sets of two queries:

|  |  |
| --- | --- |
| 1  2  3 | SELECT \* FROM table1  INTERSECT  SELECT \* FROM table2; |

Modifying data

[Insert a new row into a table](http://www.postgresqltutorial.com/postgresql-insert/):

|  |  |
| --- | --- |
| 1  2 | INSERT INTO table(column1,column2,...)  VALUES(value\_1,value\_2,...); |

Insert multiple rows into a table:

|  |  |
| --- | --- |
| 1  2  3  4 | INSERT INTO table\_name(column1,column2,...)  VALUES(value\_1,value\_2,...),        (value\_1,value\_2,...),        (value\_1,value\_2,...)... |

[Update](http://www.postgresqltutorial.com/postgresql-update/) data for all rows:

|  |  |
| --- | --- |
| 1  2  3 | UPDATE table\_name  SET column\_1 = value\_1,      ...; |

Update data for a set of rows specified by a condition in the WHERE clause.

|  |  |
| --- | --- |
| 1  2  3  4 | UPDATE table  SET column\_1 = value\_1,      ...  WHERE condition; |

[Delete all rows](http://www.postgresqltutorial.com/postgresql-delete/) of a table:

|  |  |
| --- | --- |
| 1 | DELETE FROM table\_name; |

Delete specific rows based on a condition:

|  |  |
| --- | --- |
| 1  2 | DELETE FROM table\_name  WHERE condition; |

Performance

Show the query plan for a query:

|  |  |
| --- | --- |
| 1 | EXPLAIN query; |

Show and execute the query plan for a query:

|  |  |
| --- | --- |
| 1 | EXPLAIN ANALYZE query; |

Collect statistics:

|  |  |
| --- | --- |
| 1 | ANALYZE table\_name; |