

PostgreSQL Basics

Learn SQL online at www.DataCamp.com

What is PostgreSQL?

PostgreSQL is an open-source relational database management system (RDBMS) known for its extensibility and standards compliance. Developed and maintained by a group of volunteers known as The PostgreSQL Global Development Group, it is popular across a wide range of organizations from enterprises to government departments. It has powerful data analysis capabilities.

Sample Data

The dataset contains details of the world's fastest production cars by 0 to 60 mph acceleration time. Each row contains one car model, and the table is named `cars`.

make	model	year	propulsion_type	time_to_60_mph_s	limited_production_count
Lamborghini	Huracán Performante	2018	ICE	2.2	
Ferrari	SF90 Stradale	2021	Hybrid	2	
Tesla	Model S Plaid	2021	Electric	1.98	
Porsche	918 Spyder	2015	Hybrid	2.1	918
Rimac	Nevera	2021	Electric	1.74	150
Porsche	911 Turbo S (992)	2020	ICE	2.1	

Querying tables

Get all the columns from a table using `SELECT *`

```
SELECT *
FROM cars
```

Get a column from a table by name using `SELECT col`

```
SELECT model
FROM cars
```

Get multiple columns from a table by name using `SELECT col1, col2`

```
SELECT make, model
FROM cars
```

Override column names with `SELECT col AS new_name`

```
SELECT make, model, propulsion_type AS engine_type
FROM cars
```

Override column names with `SELECT col AS new_name`

```
SELECT make, model, propulsion_type
AS engine_type
FROM cars
```

Arrange the rows in ascending order of values in a column with `ORDER BY col`

```
SELECT make, model, time_to_60_mph_s
FROM cars
ORDER BY time_to_60_mph_s
```

Arrange the rows in descending order of values in a column with `ORDER BY col DESC`

```
SELECT make, model, model_year
FROM cars
ORDER BY model_year DESC
```

Limit the number of rows returned with `LIMIT n`

```
SELECT *
FROM cars
LIMIT 2
```

Get unique values with `SELECT DISTINCT`

```
SELECT DISTINCT propulsion_type
FROM cars
```

Filtering Data

Filtering on numeric columns

Get rows where a number is greater than a value with `WHERE col > n`

```
SELECT make, model, time_to_60_mph_s
FROM cars
WHERE time_to_60_mph_s > 2.1
```

Get rows where a number is greater than or equal to a value with `WHERE col >= n`

```
SELECT make, model, time_to_60_mph_s
FROM cars
WHERE time_to_60_mph_s >= 2.1
```

Get rows where a number is less than a value with `WHERE col < n`

```
SELECT make, model, time_to_60_mph_s
FROM cars
WHERE time_to_60_mph_s < 2.1
```

Get rows where a number is less than or equal to a value with `WHERE col <= n`

```
SELECT make, model, time_to_60_mph_s
FROM cars
WHERE time_to_60_mph_s <= 2.1
```

Get rows where a number is equal to a value with `WHERE col = n`

```
SELECT make, model, time_to_60_mph_s
FROM cars
WHERE time_to_60_mph_s = 2.1
```

Get rows where a number is not equal to a value with `WHERE col <> n` or `WHERE col != n`

```
SELECT make, model, time_to_60_mph_s
FROM cars
WHERE time_to_60_mph_s <> 2.1
```

Get rows where a number is between two values (inclusive) with `WHERE col BETWEEN m AND n`

```
SELECT make, model, time_to_60_mph_s
FROM cars
WHERE time_to_60_mph_s BETWEEN 1.9 AND 2.1
```

Filtering on text columns

Get rows where text is equal to a value with `WHERE col = 'x'`

```
SELECT make, model, propulsion_type
FROM cars
WHERE propulsion_type = 'Hybrid'
```

Get rows where text is one of several values with `WHERE col IN ('x', 'y')`

```
SELECT make, model, propulsion_type
FROM cars
WHERE propulsion_type IN ('Electric', 'Hybrid')
```

Get rows where text contains specific letters with `WHERE col LIKE '%abc%'` (% represents any characters)

```
SELECT make, model, propulsion_type
FROM cars
WHERE propulsion_type LIKE '%ic%'
```

For case insensitive matching, use `WHERE col ILIKE '%abc%'`

```
SELECT make, model, propulsion_type
FROM cars
WHERE propulsion_type ILIKE '%ic%'
```

Filtering on multiple columns

Get the rows where one condition and another condition holds with `WHERE condn1 AND condn2`

```
SELECT make, model, propulsion_type, model_year
FROM cars
WHERE propulsion_type = 'Hybrid' AND model_year < 2020
```

Get the rows where one condition or another condition holds with `WHERE condn1 OR condn2`

```
SELECT make, model, propulsion_type, model_year
FROM cars
WHERE propulsion_type = 'Hybrid' OR model_year < 2020
```

Filtering on missing data

Get rows where values are missing with `WHERE col IS NULL`

```
SELECT make, model, limited_production_count
FROM cars
WHERE limited_production_count IS NULL
```

Get rows where values are not missing with `WHERE col IS NOT NULL`

```
SELECT make, model, limited_production_count
FROM cars
WHERE limited_production_count IS NOT NULL
```

Aggregating Data

Simple aggregations

Get the total number of rows `SELECT COUNT(*)`

```
SELECT COUNT(*)
FROM cars
```

Get the total value of a column with `SELECT SUM(col)`

```
SELECT SUM(limited_production_count)
FROM cars
```

Get the mean value of a column with `SELECT AVG(col)`

```
SELECT AVG(time_to_60_mph_s)
FROM cars
```

Get the minimum value of a column with `SELECT MIN(col)`

```
SELECT MIN(time_to_60_mph_s)
FROM cars
```

Get the maximum value of a column with `SELECT MAX(col)`

```
SELECT MAX(time_to_60_mph_s)
FROM cars
```

Grouping, filtering, and sorting

Get summaries grouped by values with `GROUP BY col`

```
SELECT propulsion_type, COUNT(*)
FROM cars
GROUP BY propulsion_type
```

Get summaries grouped by values, in order of summaries with `GROUP BY col ORDER BY smmry DESC`

```
SELECT propulsion_type, AVG(time_to_60_mph_s) AS mean_time_to_60_mph_s
FROM cars
GROUP BY propulsion_type
ORDER BY mean_time_to_60_mph_s
```

Get rows where values in a group meet a criterion with `GROUP BY col HAVING condn`

```
SELECT propulsion_type, AVG(time_to_60_mph_s) AS mean_time_to_60_mph_s
FROM cars
GROUP BY propulsion_type
HAVING mean_time_to_60_mph_s > 2
```

Filter before and after grouping with `WHERE condn_before GROUP BY col HAVING condn_after`

```
SELECT propulsion_type, AVG(time_to_60_mph_s) AS mean_time_to_60_mph_s
FROM cars
WHERE limited_production_count IS NOT NULL
GROUP BY propulsion_type
HAVING mean_time_to_60_mph_s > 2
```

PostgreSQL-Specific Syntax

Not all code works in every dialect of SQL. The following examples work in PostgreSQL, but are not guaranteed to work in other dialects.

Limit the number of rows returned, offset from the top with `LIMIT m OFFSET n`

```
SELECT *
FROM cars
LIMIT 2 OFFSET 3
```

PostgreSQL allows text concatenation with the `||` operator

```
SELECT make || ' ' || model AS make_and_model
FROM cars
```

Get the current date with `CURRENT_DATE` and the current datetime with `NOW()` or `CURRENT_TIME`

```
SELECT NOW(), CURRENT_DATE, CURRENT_TIME
```

List available tables by selecting from `pg_catalog.pg_tables`

```
SELECT *
FROM pg_catalog.pg_tables
```