

PostgreSQL Common Queries

Brief overview of selecting and querying data in PostgreSQL. Many of these commands will be similar/identical to commands run in other database servers e.g. SQL Server, Oracle.

Using the pgAdmin 4 GUI

The Graphical User Interface can be used to do various basic queries such as return a certain number of rows from a table, or allow basic editing of the tables. For example:

1. In the pgAdmin explorer navigate in the tree to Databases > Training > Schemas > Public > Tables
2. Right click on the postcodes table and select View All Rows

For the rest of the exercises we will be using basic SQL commands and running these using the query tool.

1. Ensure a database is selected in the left hand servers list.
2. Select Tools > Query Tool.
3. Queries can then be written in the editor and run by hitting F5 or clicking the run (lightning) icon.

To run specific commands in the query editor, highlight the line you wish to run. Otherwise the tool will attempt to run everything in the editor window.

To run sequential commands in the query tool, ensure that the commands are separated by a semi colon.

Selecting data

Select particular columns from the data table. In the case below, just the postcode and positional quality indicator.

- [SELECT statement documentation](#)

```
SELECT postcode, positional_quality_indicator FROM postcodes;
```

Where clauses

Filter the data by specifying values that columns must conform to. In the case below, where the postcode begins 'EX'.

- [WHERE clause featured in SELECT documentation](#)

```
SELECT * FROM postcodes WHERE postcode LIKE 'EX%';
```

Order By

Modify the order in which the data is returned. In the case below, postcodes sorted in reverse alphabetical order.

- [ORDER BY documentation](#)

```
SELECT postcode FROM postcodes ORDER BY postcode DESC;
```

Group By

Commonly used for aggregating and counting data. In the case below, count postcodes by first two letters.

- [GROUP BY featured in SELECT documentation](#)

```
SELECT SUBSTRING(postcode, 1, 3), COUNT(*) FROM postcodes GROUP BY SUBSTRING(postcode, 1, 3);
```

Update Data

SQL Update commands will update data in a database table. It is important to use the WHERE clause with update statements as they will otherwise update every row. For example, updating a postcode to another postcode:

- [UPDATE statement documentation](#)

```
UPDATE postcodes
SET postcode = 'EX1 4BA'
WHERE postcode = 'EX1 4BB'
```

Automating commands

When you have a set of SQL commands that you want to run outside of using any client this can be done using the psql.exe application and passing it a file of SQL commands.

- [psql documentation](#)

```
C:\Program Files\PostgreSQL\9.6\bin\psql.exe -f "mycommands.sql" -d mydatabase
```