Sure! Here are some common PostgreSQL queries and explanations of what each one does:

**1. Creating a Database**

sql

Copy code

CREATE DATABASE mydatabase;

* **What it does**: This query creates a new database named mydatabase.

**2. Creating a Table**

sql

Copy code

CREATE TABLE employees (

id SERIAL PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100) UNIQUE,

hire\_date DATE NOT NULL,

salary NUMERIC(10, 2)

);

* **What it does**: This query creates a new table named employees with columns for id, first\_name, last\_name, email, hire\_date, and salary. The id column is a serial (auto-incrementing) primary key. The email column must be unique, and hire\_date cannot be null.

**3. Inserting Data into a Table**

sql

Copy code

INSERT INTO employees (first\_name, last\_name, email, hire\_date, salary)

VALUES ('John', 'Doe', 'john.doe@example.com', '2023-01-15', 60000.00);

* **What it does**: This query inserts a new row into the employees table with the specified values for first\_name, last\_name, email, hire\_date, and salary.

**4. Selecting Data from a Table**

sql

Copy code

SELECT \* FROM employees;

* **What it does**: This query selects all columns and rows from the employees table.

**5. Selecting Specific Columns**

sql

Copy code

SELECT first\_name, last\_name FROM employees;

* **What it does**: This query selects only the first\_name and last\_name columns from the employees table.

**6. Filtering Data with WHERE**

sql

Copy code

SELECT \* FROM employees WHERE salary > 50000;

* **What it does**: This query selects all columns from the employees table where the salary is greater than 50,000.

**7. Updating Data**

sql

Copy code

UPDATE employees SET salary = 65000 WHERE id = 1;

* **What it does**: This query updates the salary of the employee with id 1 to 65,000.

**8. Deleting Data**

sql

Copy code

DELETE FROM employees WHERE id = 2;

* **What it does**: This query deletes the row from the employees table where the id is 2.

**9. Adding a New Column**

sql

Copy code

ALTER TABLE employees ADD COLUMN department VARCHAR(50);

* **What it does**: This query adds a new column named department to the employees table.

**10. Dropping a Column**

sql

Copy code

ALTER TABLE employees DROP COLUMN department;

* **What it does**: This query drops the department column from the employees table.

**11. Creating an Index**

sql

Copy code

CREATE INDEX idx\_last\_name ON employees (last\_name);

* **What it does**: This query creates an index on the last\_name column of the employees table to speed up searches based on last\_name.

**12. Joining Tables**

sql

Copy code

SELECT employees.first\_name, employees.last\_name, departments.department\_name

FROM employees

JOIN departments ON employees.department\_id = departments.id;

* **What it does**: This query selects the first\_name and last\_name from the employees table and the department\_name from the departments table, joining them on the department\_id field.

**13. Aggregating Data**

sql

Copy code

SELECT department\_id, AVG(salary) AS average\_salary

FROM employees

GROUP BY department\_id;

* **What it does**: This query calculates the average salary for each department\_id in the employees table and groups the results by department\_id.

**14. Using Subqueries**

sql

Copy code

SELECT first\_name, last\_name

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

* **What it does**: This query selects the first\_name and last\_name of employees whose salary is greater than the average salary of all employees.

**15. Creating a View**

sql

Copy code

CREATE VIEW high\_earners AS

SELECT first\_name, last\_name, salary

FROM employees

WHERE salary > 70000;

* **What it does**: This query creates a view named high\_earners that includes the first\_name, last\_name, and salary of employees whose salary is greater than 70,000.

**16. Dropping a Table**

sql

Copy code

DROP TABLE employees;

* **What it does**: This query deletes the employees table and all of its data.

These are some of the most common SQL queries used in PostgreSQL, each serving a different purpose in database management. If you have any specific queries in mind or need further details on any of these, feel free to ask!