

# SQL SELECT Statement

## 1. What is SELECT in SQL?

- SELECT is used to retrieve data from database tables.
- The result is stored in a **result-set** (virtual table).

## 2. Basic SELECT Syntax

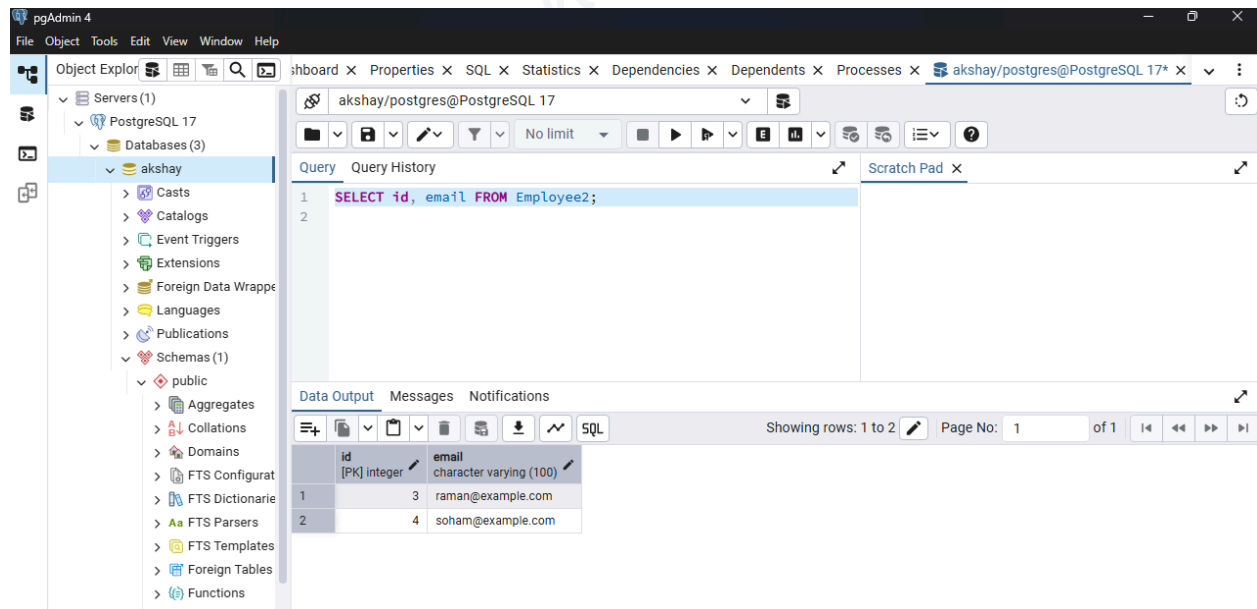
SELECT column1, column2, ... FROM table\_name;

To get all columns:

SELECT \* FROM table\_name;

**Example:**

SELECT id, email FROM Employee2;



```
SELECT * FROM customers;
```

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows a tree view of the database structure, including Servers, Databases, and Schemas. The main pane displays a SQL query: `SELECT * FROM customer;`. Below the query, the Data Output tab shows the results of the query. The results are displayed in a table with 9 rows and 8 columns: customer\_id, customer\_name, segment, age, country, city, and state. The table shows data for various customers, including Claire Gute, Darrin Van Huff, Sean O'Donnell, Brosina Hoffman, Andrew Allen, Irene Maddox, Harold Pawlan, Pete Kriz, and Alejandro Grove.

customer_id	customer_name	segment	age	country	city	state	
1	CG-12520	Claire Gute	Consumer	67	United States	Henderson	Kentucky
2	DV-13045	Darrin Van Huff	Corporate	31	United States	Los Angeles	California
3	SO-20335	Sean O'Donnell	Consumer	65	United States	Fort Lauderdale	Florida
4	BH-11710	Brosina Hoffman	Consumer	20	United States	Los Angeles	California
5	AA-10480	Andrew Allen	Consumer	50	United States	Concord	North Carolina
6	IM-15070	Irene Maddox	Consumer	66	United States	Seattle	Washington
7	HP-14815	Harold Pawlan	Home Office	20	United States	Fort Worth	Texas
8	PK-19075	Pete Kriz	Consumer	46	United States	Madison	Wisconsin
9	AG-10270	Alejandro Grove	Consumer	18	United States	West Jordan	Utah

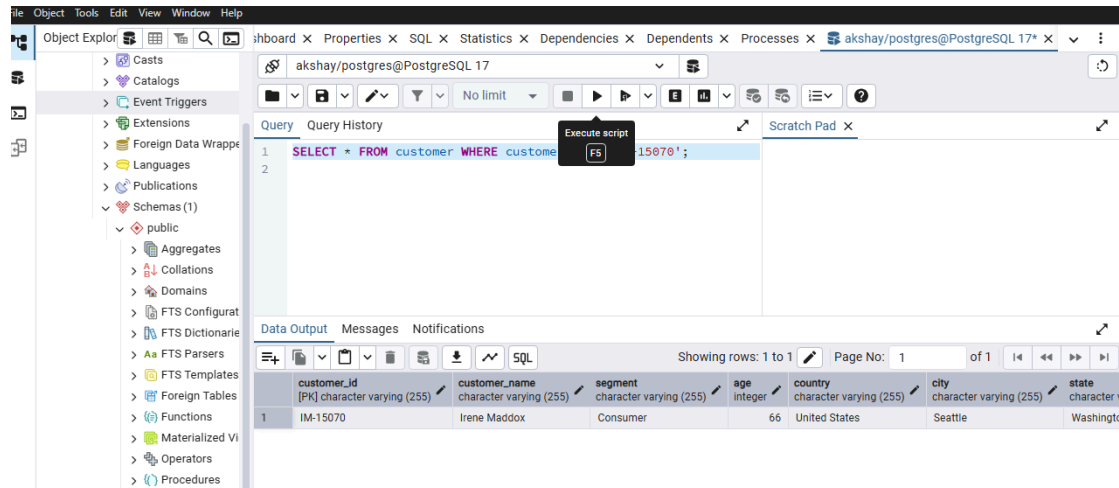
### 3. SELECT with WHERE Clause

- Filters data based on specific conditions.

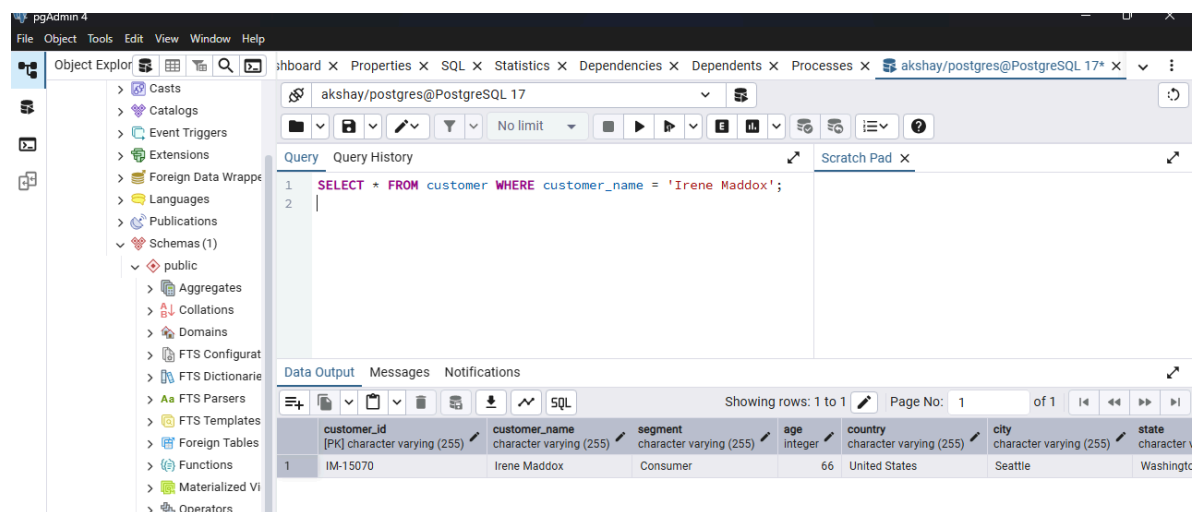
```
SELECT * FROM table_name WHERE condition;
```

#### Example:

```
SELECT * FROM customer WHERE customer_id = 'IM-15070';
```



**SELECT \* FROM customers WHERE customer\_name = 'Irene Maddox';**



## 4. SELECT DISTINCT

- Removes duplicate values from the result.

SELECT DISTINCT column\_name FROM table\_name;

**Example:**

**SELECT DISTINCT state FROM customer;**

## Output:

The screenshot shows the DBeaver interface with a SQL query executed in the 'Query' tab. The query is `SELECT DISTINCT state FROM customer;`. The 'Data Output' tab shows the results of the query, displaying a table with 10 rows of distinct states.

state
1 Oklahoma
2 Colorado
3 North Carolina
4 Mississippi
5 Florida
6 Delaware
7 Nevada
8 Louisiana
9 New York
10 New Jersey

## 5. SELECT COUNT()

- Returns the number of rows (records).

1. `SELECT COUNT(column_name) FROM table_name;`

### Example:

`SELECT COUNT(state) FROM customer;`

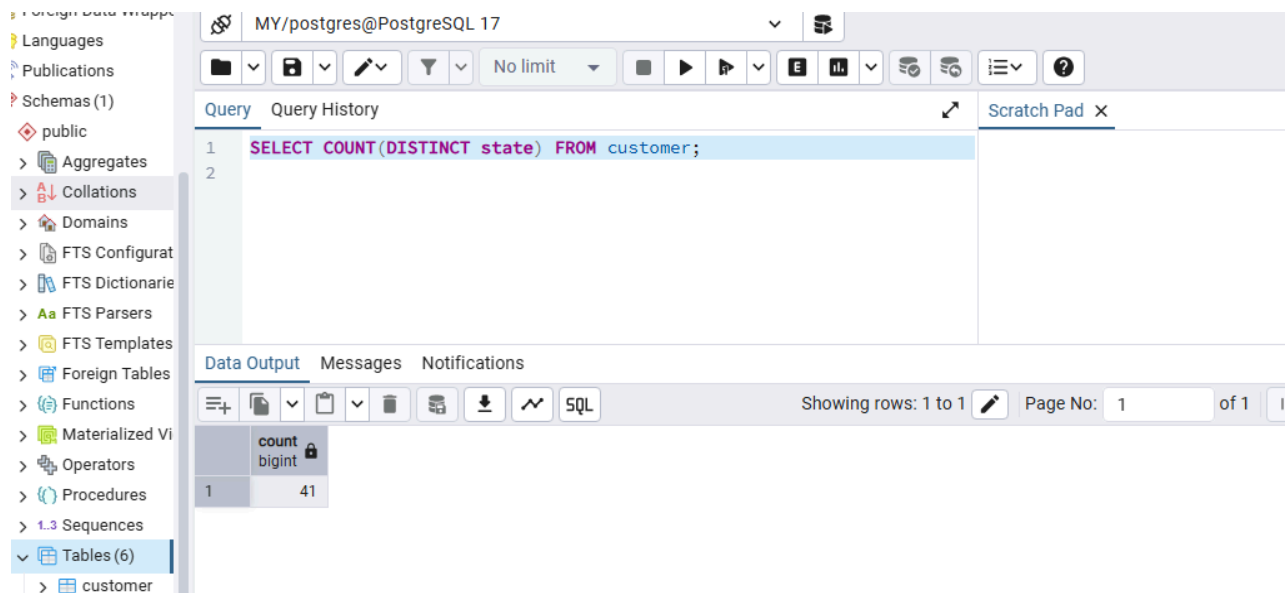
The screenshot shows the DBeaver interface with a SQL query executed in the 'Query' tab. The query is `SELECT COUNT(state) FROM customer;`. The 'Data Output' tab shows the results of the query, displaying a table with 1 row showing the count of records.

count
1 793

2. `SELECT COUNT(DISTINCT column_name) FROM table_name WHERE condition;`

### Example:

`SELECT COUNT(DISTINCT state) FROM customer;`



The screenshot shows a PostgreSQL client interface. On the left is a sidebar with a tree view of database objects: Languages, Publications, Schemas (1), public, Aggregates, Collations, Domains, FTS Configurat, FTS Dictionarie, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Vi, Operators, Procedures, Sequences, and Tables (6). The 'customer' table is selected under Tables (6). The main window has a title bar 'MY/postgres@PostgreSQL 17' and a toolbar with icons for file operations, query execution, and settings. Below the toolbar is a 'Query' tab with the text: `1 SELECT COUNT(DISTINCT state) FROM customer;`  
`2`. To the right of the query is a 'Scratch Pad' tab. Below the query editor is a 'Data Output' tab showing a table with two columns: 'count' and 'bigint'. The first row of data shows '1' and '41'. The status bar at the bottom indicates 'Showing rows: 1 to 1', 'Page No: 1', and 'of 1'.

## 6. SELECT with IN

Used to filter data based on a list of **multiple values**.

`SELECT column1, column2`

`FROM table_name`

`WHERE column_name IN (value1, value2, ...);`

### Example 1 :

`SELECT * FROM Products WHERE Category IN ('Furniture', 'Technology');`

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane displays the database structure, with 'Tables (6)' under the 'public' schema selected. The main pane shows a SQL query editor with the following query:

```
SELECT * FROM Product WHERE Category IN ('Furniture', 'Technology');
```

Below the query editor, the 'Data Output' pane displays the results of the query. The results are shown in a table with the following columns: product\_id, category, sub\_category, and product\_name. The table contains 10 rows of data.

product_id	category	sub_category	product_name
FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase
FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs Rounded Back
FUR-TA-10000577	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table
FUR-FU-10001487	Furniture	Furnishings	Eldon Expressions Wood and Plastic Desk Accessories Cherry Wood
TEC-PH-10002275	Technology	Phones	Mitel 5320 IP Phone VoIP phone
FUR-TA-10001539	Furniture	Tables	Chromcraft Rectangular Conference Tables
TEC-PH-10002033	Technology	Phones	Konftel 250 Conference phone Charcoal black
TEC-PH-10001949	Technology	Phones	Cisco SPA 501G IP Phone
FUR-CH-10002774	Furniture	Chairs	Global Deluxe Stacking Chair Gray
TEC-AC-10003027	Technology	Accessories	Imation 8GB Mini TravelDrive USB 2.0 Flash Drive

Fetches all products that belong to either the Furniture or Technology category.

## Example 2 :

**SELECT \* FROM Customer WHERE Region IN ('West', 'East');**

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane displays the database structure, with 'Tables (6)' under the 'public' schema selected. The main pane shows a SQL query editor with the following query:

```
SELECT * FROM Customer WHERE Region IN ('West', 'East');
```

Below the query editor, the 'Data Output' pane displays the results of the query. The results are shown in a table with the following columns: customer\_id, country, city, state, postal\_code, region, and test. The table contains 10 rows of data.

customer_id	country	city	state	postal_code	region	test
31	United States	Los Angeles	California	90036	West	[null]
20	United States	Los Angeles	California	90032	West	[null]
66	United States	Seattle	Washington	98103	West	[null]
18	United States	West Jordan	Utah	84084	West	[null]
66	United States	San Francisco	California	94109	West	[null]
41	United States	Philadelphia	Pennsylvania	19140	East	[null]
34	United States	Orem	Utah	84057	West	[null]
21	United States	Los Angeles	California	90049	West	[null]
48	United States	Philadelphia	Pennsylvania	19140	East	[null]
51	United States	Los Angeles	California	90049	West	[null]

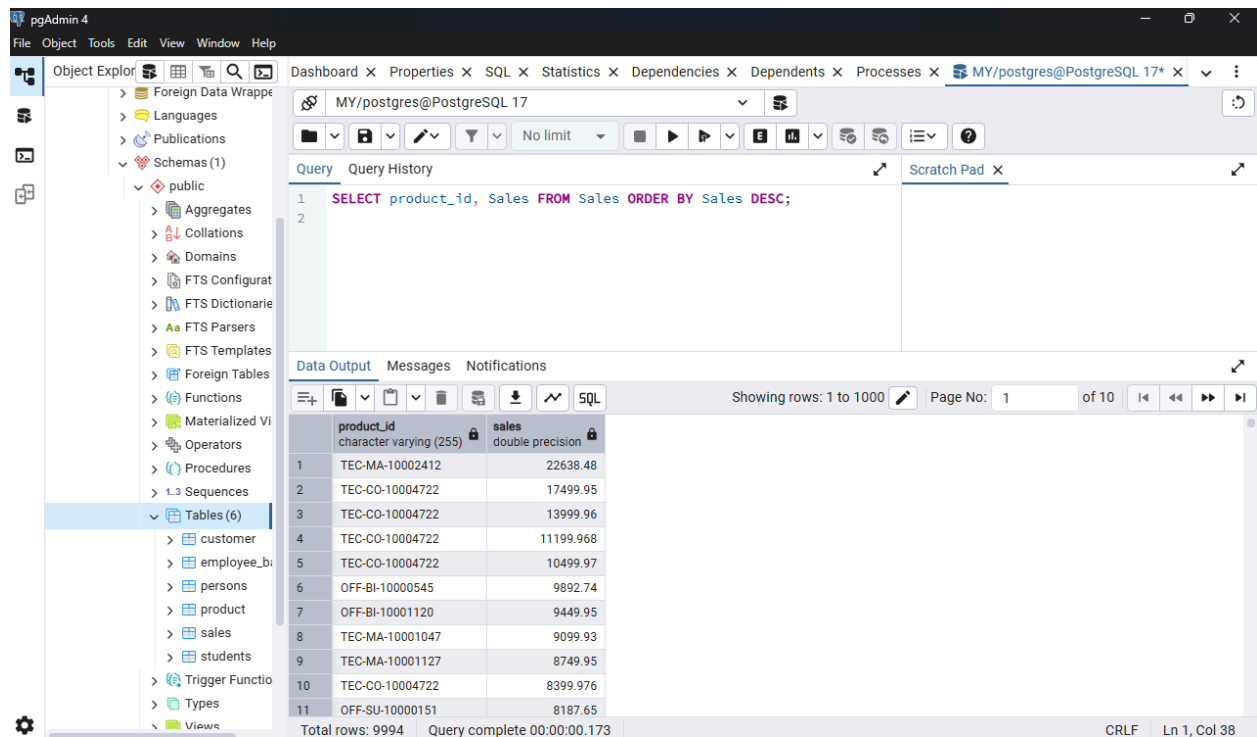
Retrieves all customers located in the West or East regions.

## 7. SELECT with ORDER BY

Used to **sort** the result set by one or more columns.

**Example:**

**SELECT product\_id , Sales FROM Sales ORDER BY Sales DESC;**



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'Tables (6)' folder expanded under the 'public' schema. The main window shows a SQL query editor with the following query:

```
1 SELECT product_id, Sales FROM Sales ORDER BY Sales DESC;
```

The query results are displayed in the 'Data Output' tab, showing 11 rows of data. The columns are 'product\_id' (character varying (255)) and 'sales' (double precision). The results are sorted in descending order of sales.

product_id	sales
TEC-MA-10002412	22638.48
TEC-CO-10004722	17499.95
TEC-CO-10004722	13999.96
TEC-CO-10004722	11199.968
TEC-CO-10004722	10499.97
OFF-BI-10000545	9892.74
OFF-BI-10001120	9449.95
TEC-MA-10001047	9099.93
TEC-MA-10001127	8749.95
TEC-CO-10004722	8399.976
OFF-SU-10000151	8187.65

The status bar at the bottom indicates 'Total rows: 9994' and 'Query complete 00:00:00.173'.

## Summary Table:

Feature	Example SQL Command
All columns	SELECT * FROM table_name;
Specific columns	SELECT name, age FROM table_name;
Filtered data	SELECT * FROM table_name WHERE age > 25;
Unique values	SELECT DISTINCT city FROM table_name;
Count total rows	SELECT COUNT(*) FROM table_name;
Count unique values	SELECT COUNT(DISTINCT salary) FROM table_name;
Match multiple values	SELECT * FROM table_name WHERE name IN ('A', 'B');



# SQL SELECT – Practice Questions

## 1. Basic SELECT

1. Show all columns from the Products table.
2. Retrieve only Product\_ID, Product\_Name, and Category from the Products table.
3. Display all records from the Customers table.

## 2. SELECT with WHERE

4. Find all orders placed by customer ID 'CA-2017-152156'.
5. Show all customers from the 'East' region.
6. Retrieve products that belong to the category 'Technology'.
7. Show all products with sales greater than 500.

## 3. SELECT DISTINCT

8. List all unique product categories.
9. Show distinct customer segments.
10. Display all distinct regions from the Customers table.

## 4. SELECT COUNT()

11. Count the total number of orders.
12. Count how many unique customers are in the database.
13. Count how many products belong to the category 'Furniture'.
14. Count the number of orders placed in the 'West' region.

## 5. SELECT with IN

- 15. Show all products in the 'Furniture' or 'Office Supplies' categories.
- 16. Retrieve all customers located in 'South' or 'Central' regions.
- 17. Display orders with IDs in the list: 'CA-2015-100111', 'CA-2016-123456', 'CA-2017-654321'.

## 6. SELECT with ORDER BY

- 18. List all customers sorted by Customer\_Name in ascending order.
- 19. Display all products sorted by Sales in descending order.
- 20. Show orders sorted by Order\_Date from oldest to newest.

You can download the complete set of SQL notes and practice files from this GitHub repository:

👉 [SQL-resources-and-tutorials by akshay-dhage](#)