What is **psql** in PostgreSQL?

`psql` is the **interactive command-line tool** for working with PostgreSQL.

It allows you to:

- Execute SQL queries
- Manage databases and users
- Run scripts
- View/query data quickly from the terminal

Think of `psql` as a **terminal interface (client)** to communicate with the PostgreSQL server.

Why do we use `psql`?

We use `psql` because it is:

- ⚡ **Fast** – no GUI needed
- 🧰 **Powerful** – supports SQL + special commands
- 🔧 **Administrative** – manage users, roles, databases
- 📜 **Scriptable** – automate tasks using `.sql` files

🪟 How to Access `psql` in Windows

✅ 1. Install PostgreSQL (if not installed)

Download from:

- https://www.postgresql.org/download/windows/

During installation:

- Set **password for postgres user**
- Keep default port **5432**
- Make sure **Command Line Tools (psql)** is selected

⚙️ 2. Open psql in Windows

Method 1: Using SQL Shell (psql) (Easiest)

1. Click **Start Menu**
2. Search:

```
SQL Shell (psql)
```

3. Open it

You will see prompts like:

```
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password:
```

👉 Press Enter for defaults and enter password.

Method 2: Using Command Prompt (CMD)

Step 1: Open CMD

Press:

```
Win + R → type cmd → Enter
```

Step 2: Run psql

```
psql -U postgres
```

If it does not work, use full path:

```
"C:\Program Files\PostgreSQL\16\bin\psql.exe" -U postgres
```

(Version folder may be 15, 14, etc.)

Method 3: Add psql to PATH (Recommended)

So you can run `psql` from anywhere:

Steps:

1. Copy this path:

```
C:\Program Files\PostgreSQL\16\bin
```

2. Go to:

```
Control Panel → System → Advanced System Settings → Environment Variables
```

3. Edit **Path**
4. Add new entry:

```
C:\Program Files\PostgreSQL\16\bin
```

for more details, see Fix 'psql' is not recognized on Windows

5. Restart CMD

Now just type:

```
psql -U postgres
```

🔐 3. Login Examples

Login as postgres user:

```
psql -U postgres
```

Login with database:

```
psql -U postgres -d university_db
```

Login with host & port:

```
psql -U postgres -h localhost -p 5432
```

📂 4. Useful First Commands after login

```
\l        -- list databases
\c postgres   -- connect database
\dt       -- list tables
\du       -- list users
\q        -- quit
```

⚠️ Common Problems & Fixes

❌ “psql is not recognized”

✓ Fix: Add PostgreSQL `bin` folder to PATH

❌ Password error

✓ Fix:

- Check password used during installation
- Reset postgres password if needed

❌ Connection failed

✓ Fix:

- Ensure PostgreSQL service is running:

```
Services → PostgreSQL → Start
```

How to Login into `psql`

1. Login with **postgres (default superuser)**

```
psql -U postgres
```

If PostgreSQL is running locally:

```
psql -U postgres -h localhost -p 5432
```

👉 You will be prompted for a password.

---

2. Login into a **specific database**

```
psql -U postgres -d mydatabase
```

---

3. Login with a **new user**

First create a user:

```
CREATE ROLE student_user WITH LOGIN PASSWORD '1234';
```

Then login:

```
psql -U student_user -d mydatabase
```

Or:

```
psql -U student_user -h localhost -d mydatabase
```

---

4. Login using **connection string**

```
psql "postgresql://student_user:1234@localhost:5432/mydatabase"
```

📘 `psql` Backslash Commands (Meta-Commands) with Examples

In **psql (PostgreSQL interactive terminal)**, backslash commands (`\`) are used to manage and explore databases.

👉 Key point:

- Executed by **psql client (not SQL server)**
- Start with `\`
- Very useful for learning and administration

🔍 1. Essential Discovery Commands (with Examples)

📌 `\l` → List all databases

```
\l
```

👉 Shows all databases on the PostgreSQL server

📌 `\dn` → List schemas

```
\dn
```

👉 Example output:

- public
- information_schema

📌 \dt → List tables

```
\dt
```

👉 Example:

```
List of relations
Schema | Name     | Type  | Owner
-------+----------+-------+-------
public | students | table | postgres
```

📌 \di → List indexes

```
\di
```

👉 Shows indexes created on tables

📌 \dv → List views

```
\dv
```

👉 Displays all views in current database

📌 \df → List functions

```
\df
```

👉 Shows stored functions

📌 \du → List users/roles

```
\du
```

👉 Example:

```
Role name | Attributes
----------+----------------
```

```
postgres  | Superuser
student   | Create DB
```

🧭 2. Navigation & Connection Commands

📌 `\c database_name` → Connect to database

```
\c university_db
```

👉 Output:

```
You are now connected to database "university_db"
```

📌 `\conninfo` → Show connection details

```
\conninfo
```

👉 Example:

```
You are connected to database "university_db" as user "postgres" on host "localhost"
at port "5432"
```

📌 `\q` → Quit psql

```
\q
```

👉 Exits PostgreSQL terminal

📊 3. Inspecting Objects (Very Important for Students)

📌 `\d table_name` → Describe table

```
\d students
```

👉 Example output:

```
Column | Type         | Constraints
-------+--------------+-------------
id     | integer      | primary key
name   | varchar(50)  |
```

📌 `\d+ table_name` → Detailed table info

```
\d+ students
```

👉 Shows:

- Column details
- Indexes
- Table size
- Comments

🖥️ 4. System & Output Commands

📌 `\x` → Expanded view toggle

```
\x
SELECT * FROM students;
```

👉 Output becomes vertical (useful for many columns):

```
id   : 1
name : Ali
```

📌 `\timing` → Show query execution time

```
\timing
SELECT * FROM students;
```

👉 Output:

```
Time: 1.234 ms
```

📌 `\i filename.sql` → Run SQL file

```
\i school.sql
```

👉 Executes all SQL inside file:

```
CREATE TABLE students (...);
INSERT INTO students VALUES (...);
```

📌 `\! command` → Run OS command

```
\! ls
```

👉 Example output:

```
school.sql
data.sql
```

Other example:

```
\! pwd
```

❓ 5. Help Commands (Must Know)

📌 `\?` → All psql commands

```
\?
```

👉 Shows full list of meta-commands

📌 `\h` → SQL help

```
\h
```

📌 `\h command` → Specific SQL help

```
\h CREATE TABLE
```

👉 Shows syntax:

```
CREATE TABLE table_name (
    column_name data_type
);
```

🧪 Examples

```
\l                          -- list databases
\c university_db            -- connect database
\dn                         -- show schemas
\dt                         -- show tables

\d students                -- describe table
SELECT * FROM students;

\x                          -- expanded view
\timing                     -- enable timing

\! ls                       -- check files
\i insert_data.sql         -- run SQL file
```

⭐ Summary

✓ \ commands = psql meta-commands ✓ Used for:

- Database navigation
- Table inspection
- System control
- Debugging & learning
