PostgreSQL Database Setup and Hosting Guide

Understanding PostgreSQL vs Microsoft SQL Server

- PostgreSQL (Postgres) and Microsoft SQL Server (MSSQL) are different systems.
- PostgreSQL is open-source, cross-platform, and uses PL/pgSQL.
- Microsoft SQL Server is proprietary, mostly Windows-based, and uses T-SQL.

You do *not* need Microsoft SQL Server to run PostgreSQL.

---

1. Install PostgreSQL

Option A: Local Installation

1. Download from https://www.postgresql.org/download/
2. Choose your operating system (Windows, macOS, or Linux).
3. During setup, note:
    - Port (default: `5432`)
    - Username (default: `postgres`)
    - The password you create

Option B: Cloud Installation

Use a managed PostgreSQL service such as:

- Supabase
- Neon.tech
- Render
- Amazon RDS
- Azure Database for PostgreSQL

---

2. Windows-Only Setup Notes

Add PostgreSQL to the System PATH

To run `psql` from the Command Prompt or PowerShell without specifying the full path:

1. Press **Windows + S**, type **Environment Variables**, and open *Edit the system environment variables*.
2. Click **Environment Variables**.
3. Under *System variables*, select **Path** → click **Edit**.
4. Add the path to your PostgreSQL `bin` directory, for example:

```
C:\Program Files\PostgreSQL\15\bin
```

5. Click **OK** and restart your terminal.

You can now verify PostgreSQL is in your PATH:

```
psql --version
```

Using pgAdmin (Graphical Interface)

- pgAdmin is installed automatically with PostgreSQL on Windows.
- Launch it from the Start menu.
- Connect to your local server (default: `localhost`, port `5432`).
- You can:
  - Create databases and tables through a visual UI.
  - Run SQL queries in the query tool.
  - Backup and restore databases.

Start or Stop the PostgreSQL Service

If PostgreSQL doesn't start automatically, you can manage it manually:

```
net start postgresql-x64-15
net stop postgresql-x64-15
```

(The version number may differ based on your installation.)

---

3. Create a Database

Open Command Prompt or PowerShell and run:

```
psql -U postgres
```

Inside the PostgreSQL shell:

```
CREATE DATABASE my_database;
\c my_database;
```

---

4. Create Tables and Insert Data

```
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
```

```
    name VARCHAR(100),
    email VARCHAR(100) UNIQUE
);

INSERT INTO users (name, email)
VALUES ('Paul Namalomba', 'paul@example.com');

SELECT * FROM users;
```

---

5. Host the Database on a Server

If PostgreSQL is installed locally, you can enable remote access.

(a) Edit Configuration Files

File: `postgresql.conf`

```
listen_addresses = '*'
```

File: `pg_hba.conf`

```
host    all    all    0.0.0.0/0    md5
```

Then restart PostgreSQL:

```
sudo systemctl restart postgresql
```

(b) Open Port 5432

Allow inbound connections on TCP port 5432 through your firewall.

(c) Connect Remotely

```
psql -h your.server.ip -U postgres -d my_database
```

---

6. Cloud Hosting (Easier Option)

You can skip server setup by using a hosted provider.

Provider	Description

Provider	Description
Supabase	A backend-as-a-service platform built around PostgreSQL. It provides a hosted PostgreSQL database, authentication, storage, and auto-generated APIs. Ideal for web and mobile apps.
Render	A cloud hosting platform that offers free and paid PostgreSQL instances, along with app and API hosting. Similar to Heroku but simpler and more modern.
Neon.tech	Serverless PostgreSQL hosting that allows branching, scaling, and free usage tiers.

7. Example: Set Up PostgreSQL on Supabase

1. Visit https://supabase.com
2. Sign up with GitHub or email.
3. Create a **New Project**.
4. Choose:
    - **Project Name** (e.g., `mydatabase`)
    - **Region** (closest to you)
    - **Database Password** (you'll use this to connect)
5. After setup, go to **Project Settings** → **Database**.
6. Copy your **connection string**, which looks like this:

```
postgresql://postgres:your_password@db.supabase.co:5432/postgres
```

7. Connect to it via `psql` or in your app:

```
psql "postgresql://postgres:your_password@db.supabase.co:5432/postgres"
```

---

8. Example: Set Up PostgreSQL on Render

1. Go to https://render.com
2. Sign up and click **New +** → **PostgreSQL**.
3. Choose:
    - **Name** for your database.
    - **Region**.
    - Select **Free Tier** (for testing).
4. Render will create your database and show a **Connection URL** like:

```
postgresql://renderuser:password@dpg-someid.render.com/mydb
```

5. Use this connection string in your app or connect with `psql`.

---

9. Connect Your Application

Example (Python)

```
import psycopg2

conn = psycopg2.connect(
    dbname="my_database",
    user="postgres",
    password="mypassword",
    host="your.server.ip",
    port="5432"
)

cur = conn.cursor()
cur.execute("SELECT * FROM users;")
print(cur.fetchall())
cur.close()
conn.close()
```

Summary

Step	Action	Tools
1	Install PostgreSQL	Local or Cloud
2	Windows Setup	PATH, pgAdmin
3	Create Database	psql, pgAdmin
4	Add Tables/Data	SQL Commands
5	Configure Server Access	postgresql.conf, pg_hba.conf
6	Optional Cloud Hosting	Supabase, Render, Neon
7	Connect Application	Python, Node.js, etc.

Recommended Next Steps

- Learn SQL basics (`SELECT`, `INSERT`, `UPDATE`, `DELETE`).
- Set up user roles and permissions for security.
- Create backups using `pg_dump`.
- Enable SSL/TLS if hosting publicly.