pgEdge MCP Server

Setup Guide for Windows 11

Connecting Claude Desktop to Your Local PostgreSQL Database

Before You Start — What You Need

- Claude Desktop — already installed ☑
- PostgreSQL — already installed locally ☑
- pgAdmin — already installed with PostgreSQL ☑
- A web browser (Edge or Chrome is fine)
- About 20–30 minutes of your time

General Setup Guide — fill in your own values as you complete each step

Version 1.0 | Public Edition — replace all YOUR_USERNAME, YOUR_PASSWORD, and YOUR_DATABASE_NAME placeholders with your real values

What Is This Guide?

This guide walks you through connecting Claude Desktop (the AI assistant you already use) to your local PostgreSQL database, step by step. Once set up, you can ask Claude things like:

- "What tables are in my database_name database?"
- "Show me the schema for my database."

The bridge that makes this possible is called the pgEdge MCP Server. Think of it like a secure interpreter that sits between Claude and your database, letting Claude read your data without being able to accidentally change or delete anything.

Acronym Decoder — Plain English Translations

• MCP (Model Context Protocol) — A standard created by Anthropic so AI tools like Claude can connect to outside data sources, like your database.

• LLM (Large Language Model) — The AI engine behind Claude. It's trained on text and understands natural language.

• SQL (Structured Query Language) — The language used to talk to databases. Example: SELECT * FROM operators;

• PostgreSQL (often called Postgres) — A free, open-source database system. This is what you have installed.

• pgAdmin — A visual tool (already installed) that lets you manage your PostgreSQL database without typing SQL commands.

• pgEdge — The company that built the MCP Server software you are installing.

• localhost — Your own PC. When a program connects to 'localhost,' it means it stays on your computer.

• Port 5432 — The 'door number' PostgreSQL listens on. Default and standard — you likely do not need to change this.

• Binary / .exe — An executable program file. No installation wizard needed — just download and run.

• .pgpass — A hidden config file that stores your database password so you don't have to type it every time.

• JSON (JavaScript Object Notation) — A text format used for settings/config files. Looks like: { "key": "value" }

• Read-only — Claude can look at your data but cannot add, edit, or delete anything.

Step 1 — Create a Folder for the MCP Server

What & Why

We are creating a dedicated folder to store the MCP Server program file. Keeping it separate from other programs makes it easy to find and update later.

Folder to Create

```
Location:  C:\pgEdge\MCP

This becomes the home for the MCP Server program file.
```

How To Do It

1. Press the Windows key + E to open File Explorer.
2. In the left panel, click on This PC, then double-click Local Disk (C:).
3. Right-click on an empty space in the right panel.
4. Select New > Folder. Name it: pgEdge
5. Double-click to open the new pgEdge folder.
6. Right-click again > New > Folder. Name this one: MCP

☑ **Tip:** You now have the folder C:\pgEdge\MCP ready and waiting.

📷 **SCREENSHOT REFERENCE**

How to create a new folder in Windows 11 File Explorer

https://support.microsoft.com/en-us/windows/find-and-open-file-explorer-ef370130-1cca-9dc5-e0df-2f7416fe1cb1

Step 2 — Download the pgEdge MCP Server Program

What & Why

The pgEdge MCP Server is a single program file (.exe) — no installation wizard needed. It acts as the secure bridge between Claude and your PostgreSQL database. It is open-source software, meaning it is free and its code is publicly available for review.

Where to Download

Copy and paste this URL directly into your browser address bar:

```
https://github.com/pgEdge/pgedge-postgres-mcp/releases
```

Which file to download — Windows 11 (64-bit)

- Look for the file named exactly: pgedge-postgres-mcp-server_[version]_windows_x86_64.zip
- Example (as of Feb 2025):   pgedge-postgres-mcp-server_1.0.0-beta3a_windows_x86_64.zip
- File size is approximately 8-9 MB.
-
- There are many files on the page — here is how to pick the right one:
- • Must contain 'mcp-server' (not 'mcp-cli' or 'mcp-kb-builder')
- • Must contain 'windows' (not darwin = Mac, not linux)
- • Must contain 'x86_64' (this is the standard 64-bit Windows format)
- • Will be a .zip file (not .tar.gz — those are for Mac and Linux)
-
- Save the .zip file to: C:\pgEdge\MCP\

Steps

7. Open your web browser (Edge or Chrome).
8. Go to the URL above.
9. Scroll to the latest release (highest version number at the top).
10. Click Assets to expand the list of files.
11. Click the file ending in -windows-amd64.exe to download it.
12. When prompted, choose Save As and navigate to C:\pgEdge\MCP.
13. If Windows shows a security warning, click Keep (the file is safe).

⚠ **Important:** If the file downloads as a .zip, right-click it > Extract All > extract to C:\pgEdge\MCP.

📸 **SCREENSHOT REFERENCE**

Downloading a file from GitHub Releases (Assets section)

https://zapier.com/blog/how-to-download-from-github/

Step 3 — Verify the MCP Server File Works

What & Why

Before we configure anything, let us confirm the downloaded file runs correctly on your PC. We do this using the Command Prompt (CMD) — a basic text-based tool built into Windows that lets you run programs by typing commands.

How To Do It

14. Open File Explorer (Windows key + E).
15. Navigate to C:\pgEdge\MCP where you saved the .zip file.
16. Right-click the .zip file and select Extract All...
17. In the dialog box, set the destination to C:\pgEdge\MCP and click Extract.
18. Open the extracted folder — look for a file ending in .exe (it may be inside a subfolder). Copy or move that .exe file directly into C:\pgEdge\MCP.

✅ **Tip:** After extracting, your C:\pgEdge\MCP folder should contain an .exe file. That is the MCP Server program.

Verify It Works — Using Command Prompt

19. Press Windows key + R on your keyboard.
20. Type cmd and press Enter. A black window (Command Prompt) opens.
21. Type the following and press Enter:

```
cd C:\pgEdge\MCP
```

22. Type the name of the extracted program and press Enter:

```
pgedge-postgres-mcp.exe
```

23. You should see the following two lines:

```
Database: Not configured
Mode: STDIO
```

☑ **Tip:** This is the correct and expected response at this stage! It means the program is working. 'Not configured' simply means we have not yet pointed it at your database — that happens in Steps 4, 5, and 6.

⚠ **Important:** If you see 'not recognized as an internal or external command,' the .exe file is not in C:\pgEdge\MCP or the filename is different. Check the folder contents in File Explorer.

📷 **SCREENSHOT REFERENCE**

Running a command in Windows Command Prompt

https://www.ionos.com/digitalguide/server/tools/how-to-open-command-prompt/

Step 4 — Create a Dedicated Database User for Claude in pgAdmin

What & Why

We will create a dedicated database user called claude with full read and write access to your database. Since this is your personal PC and you are the administrator, Claude needs full access to be genuinely useful — reading data, writing records, updating rows, creating tables, and helping you build and test code against your live database.

We still create a separate user rather than using your main postgres superuser, so it is easy to see what Claude is doing and adjust permissions later if ever needed.

Key Terms for This Step

• Login Role — A user account in PostgreSQL. Like a Windows user account, but for your database.

• Superuser — The highest level of database access. Can do anything. Your postgres user is a superuser.

• GRANT — A SQL command that gives a user permission to do something.

• SCHEMA — A namespace/container inside your database that holds tables. The default is called public.

• ALL PRIVILEGES — Grants full read, write, create, update, and delete permissions.

• SEQUENCES — The auto-incrementing ID counters used by your tables. Needed so Claude can insert new rows that auto-generate an ID.

Part A — Create the Claude User

24. Open pgAdmin from the Start menu (search for pgAdmin).
25. In the left panel (Browser), expand: Servers > [your server name] > Login/Group Roles.
26. Right-click on Login/Group Roles and select Create > Login/Group Role...
27. In the dialog box that opens:

- General tab — Name field: type claude
- Definition tab — Password field: type a strong password (write it down — you will need it in Step 5)
- Privileges tab — Turn ON: Can login?, Create databases?, Inherit rights from parent roles?
- Privileges tab — Leave Superuser OFF (not needed)

28. Click Save.

📷 **SCREENSHOT REFERENCE**

Creating a Login Role in pgAdmin (General and Privileges tabs)

https://www.pgadmin.org/docs/pgadmin4/latest/role_dialog.html

Part B — Grant Full Access to Your Database

Now we give the claude user full permissions on your database. Run these commands in pgAdmin's Query Tool. Replace YOUR_DATABASE_NAME with your actual database name (e.g., database_name).

29. In pgAdmin, right-click your database name (listed under Databases in the left panel).
30. Select Query Tool.
31. Paste the following commands and press F5 to run:

```
GRANT ALL PRIVILEGES ON DATABASE YOUR_DATABASE_NAME TO claude;

GRANT ALL PRIVILEGES ON SCHEMA public TO claude;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO claude;

GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO claude;

ALTER DEFAULT PRIVILEGES IN SCHEMA public
  GRANT ALL PRIVILEGES ON TABLES TO claude;

ALTER DEFAULT PRIVILEGES IN SCHEMA public
  GRANT ALL PRIVILEGES ON SEQUENCES TO claude;
```

32. You should see a success message in the Messages panel below.

☑ **Tip:** Run each command one at a time if you prefer — just highlight one block and press F5.

☑ **Tip:** ALL PRIVILEGES covers SELECT (read), INSERT, UPDATE, DELETE (write), and more. This means Claude Desktop, Claude Code, and Claude Cowork can all fully assist with your database work.

📷 **SCREENSHOT REFERENCE**

Running SQL in pgAdmin Query Tool

https://www.pgadmin.org/docs/pgadmin4/latest/query_tool.html

Step 5 — Create the .pgpass Password File

What & Why

The .pgpass file (short for PostgreSQL Password) is a hidden text file that stores your database credentials (username + password) securely. By putting the password here, you avoid having it visible in your settings files. PostgreSQL automatically checks this file when connecting.

Where to Create It

```
Folder:  C:\Users\YOUR_USERNAME\AppData\Roaming\postgresql\
File:    pgpass.conf

Full path:  C:\Users\YOUR_USERNAME\AppData\Roaming\postgresql\pgpass.conf
```

⚠ **Important:** The AppData folder is hidden by Windows by default. You will NOT see it in File Explorer unless you turn on hidden items first. Follow Part A below before anything else.

Part A — Show Hidden Folders in Windows

33. Open File Explorer (Windows key + E).
34. Click View in the top menu bar.
35. Click Show in the dropdown.
36. Click Hidden items to put a checkmark next to it.

☑ **Tip:** You only need to do this once. Windows will remember the setting. The AppData folder will now be visible inside C:\Users\YOUR_USERNAME\

Part B — Create the postgresql Folder

37. In the File Explorer address bar, type the following and press Enter:

```
C:\Users\YOUR_USERNAME
```

38. You should now see the AppData folder. Double-click it.
39. Double-click the Roaming folder.
40. Right-click in empty space > New > Folder.
41. Name the folder exactly: postgresql
42. Double-click to open the new postgresql folder.

Part C — Create the pgpass.conf File

43. Inside the postgresql folder, right-click in empty space > New > Text Document.
44. Name the file pgpass.conf — make sure to delete the .txt extension completely.

⚠ **Important:** Windows will warn you about changing the file extension. Click Yes — this is intentional.

45. Right-click pgpass.conf > Open with > Notepad.
46. Type the following single line, replacing the placeholders with your real values:

```
localhost:5432:YOUR_DATABASE_NAME:claude:YOUR_PASSWORD
```

Example with real values:

```
localhost:5432:database_name:claude:MyStr0ngP@ssw0rd!
```

47. Press Ctrl + S to save, then close Notepad.

☑ **Tip:** %APPDATA% is a Windows shortcut that points to C:\Users\YOUR_USERNAME\AppData\Roaming\ — you can type either one in the address bar and Windows understands both.

📷 **SCREENSHOT REFERENCE**

How to show hidden files and folders in Windows 11

https://support.microsoft.com/en-us/windows/show-hidden-files-0320fe58-0117-fd59-6851-9b7f9840fdb2

📷 **SCREENSHOT REFERENCE**

How to use the .pgpass / pgpass.conf file in PostgreSQL on Windows

https://tableplus.com/blog/2019/09/how-to-use-pgpass-in-postgresql.html

Step 6 — Configure Claude Desktop to Use the MCP Server

What & Why

Claude Desktop uses a settings file called claude_desktop_config.json to know about any MCP servers it should connect to. We need to add the pgEdge MCP Server to this file.

> ⚠ **Important:** IMPORTANT: Claude Desktop on this PC is installed as a Windows Package app. This means the real config file is NOT in the locations most online guides describe. The correct path is inside the Packages folder — see below.

The THREE Claude folders on this PC — and which one actually matters

- AppData\Roaming\Claude — Created manually during setup. Claude Desktop does NOT use this.
- AppData\Local\Claude — Contains logs and a config file, but Claude Desktop does NOT use this for MCP either.
- AppData\Local\Packages\Claude_[YOUR_PACKAGE_ID]\LocalCache\Roaming\Claude
- — THE REAL location. This is where the config file must be edited.
- How we found this: Searched for claude.exe across the entire user folder using Command Prompt.

Where the Real Config File Lives

```
Full path (broken across two lines for readability):
C:\Users\YOUR_USERNAME\AppData\Local\Packages\

Claude_[YOUR_PACKAGE_ID]\LocalCache\Roaming\Claude\claude_desktop_config.json

Shortcut to type in File Explorer address bar:
%LOCALAPPDATA%\Packages\Claude_[YOUR_PACKAGE_ID]\
  LocalCache\Roaming\Claude
```

Part A — Find Your Package ID and Navigate to the Real Config File

Because Claude Desktop is installed as a Windows Package app, its folder name contains a unique Package ID. Here is how to find yours:

```
Step 1 — Open Command Prompt (Windows key + R, type cmd, press Enter)

Step 2 — Paste this command and press Enter:
  dir /s /b "C:\Users\YOUR_USERNAME" 2>nul | findstr /i "claude.exe"

Step 3 — Look for a path like this in the results:
  C:\Users\YOUR_USERNAME\AppData\Local\Packages\Claude_XXXXXXXXX\...

Step 4 — The Claude_XXXXXXXXX part is your Package ID. Note it down.
         Your full config folder path will be:
  %LOCALAPPDATA%\Packages\Claude_XXXXXXXXX\LocalCache\Roaming\Claude
```

48. Open File Explorer (Windows key + E).
49. Click in the address bar, type your full path from above, and press Enter:

```
%LOCALAPPDATA%\Packages\Claude_[YOUR_PACKAGE_ID]\LocalCache\Roaming\Claude
```

50. You will see a file called claude_desktop_config.json already exists here. This is the real one.

Part B — Edit the Config File

51. Right-click claude_desktop_config.json > Open with > Notepad.
52. You will see it already contains some preferences settings. You must KEEP those and ADD the mcpServers section alongside them.
53. Replace ALL the content with the following, substituting YOUR_DATABASE_NAME and YOUR_PASSWORD with your real values:

```
{
  "preferences": {
    "coworkScheduledTasksEnabled": true,
    "ccdScheduledTasksEnabled": true,
    "sidebarMode": "chat",
    "bypassPermissionsModeEnabled": true,
    "dockBounceEnabled": true,
    "coworkWebSearchEnabled": true
  },
  "mcpServers": {
    "local-postgres": {
      "command": "C:\\pgEdge\\MCP\\pgedge-postgres-mcp.exe",
      "args": [],
      "env": {
        "PGHOST": "localhost",
        "PGPORT": "5432",
        "PGDATABASE": "YOUR_DATABASE_NAME",
        "PGUSER": "claude",
        "PGPASSWORD": "YOUR_PASSWORD",
        "PGSSLMODE": "disable"
      }
    }
  }
}
```

💡 **Note:** Replace YOUR_DATABASE_NAME with database_name and YOUR_PASSWORD with the password you set for the claude user in Step 4.

☑ **Tip:** The double backslashes \\ in the file path are required in JSON. One backslash has special meaning in JSON so we write two to represent one real backslash.

54. Press Ctrl + S to save and close Notepad.

📷 **SCREENSHOT REFERENCE**

Editing a JSON config file with Notepad on Windows

https://4sysops.com/archives/edit-json-files-with-notepad/

Step 7 — Restart Claude Desktop and Test the Connection

What & Why

Claude Desktop reads its config file when it starts up. We need to fully close and reopen it so it picks up the new MCP Server settings. Then we will verify everything is connected properly.

How To Restart Claude Completely

55. Right-click the Claude icon in the System Tray (bottom-right corner of your taskbar, near the clock).
56. Select Quit or Exit Claude.
57. If Claude is not in the tray, press Ctrl + Shift + Esc to open Task Manager.
58. In Task Manager, look for Claude in the Processes list.
59. Click on it, then click End Task.
60. Wait 5 seconds, then reopen Claude Desktop from the Start menu or desktop shortcut.

How to Confirm the MCP Connection Is Active

61. In Claude Desktop, start a new conversation.
62. Look for a hammer icon (🔨) near the message input area. NOTE: On Claude Desktop version 1.1.6679 (Windows Package install), the hammer icon does NOT appear. Skip to step 3.
63. Click the + button in the lower left corner of the chat input area.
64. Look for local-postgres in the list that appears. If it shows as active/connected, the MCP server is working.
65. Type this test message and press Enter:

```
What tables are in my database?
```

66. Claude should respond with a list of tables from your database.

☑ **Tip:** Once connected, you can ask Claude to list your tables, describe your schema, query your data, and help you write SQL.

☑ **Tip:** You can also try: 'How many rows are in my table?' or 'Show me the schema for my database.'

Troubleshooting — If local-postgres Does Not Appear in the + Menu

- Check you edited the RIGHT config file — it must be in:
- AppData\Local\Packages\Claude_[YOUR_PACKAGE_ID]\

- LocalCache\Roaming\Claude\claude_desktop_config.json
- Do NOT edit files in AppData\Roaming\Claude or AppData\Local\Claude — Claude Desktop ignores those.
- Check JSON syntax: Go to jsonlint.com and paste your config file contents to validate it.
- Make sure the preferences section is still intact — the mcpServers section is added alongside it, not replacing it.
- Check the file path: Make sure C:\pgEdge\MCP\pgedge-postgres-mcp.exe actually exists.
- Check PostgreSQL is running: Open pgAdmin — if it connects, PostgreSQL is running.
- Full restart required: System tray Quit > Task Manager End Task > Restart PC > Reopen Claude.

Quick Reference — File Locations

What	File Path
MCP Server program file	`C:\pgEdge\MCP\pgedge-postgres-mcp.exe`
Password file (.pgpass)	`C:\Users\YOUR_USERNAME\AppData\Roaming\postgresql\pgpass.conf`
Claude Desktop config (line 1)	`...\Packages\Claude_[YOUR_PACKAGE_ID]\`
Claude Desktop config (line 2)	`LocalCache\Roaming\Claude\claude_desktop_config.json`
PostgreSQL default port	`5432`
Database user created	`claude  (full read/write access)`

My Notes — Setup Record

Fill in these values as you complete each step. Keep this document somewhere safe for future reference.

```
My database name:          ________________________________

Windows username:          YOUR_USERNAME  (e.g. john, jane, etc.)

MCP Server filename:       ________________________________

MCP Server version:        ________________________________

Database user created:     claude  (full read/write access)
```

```
Claude Desktop version:    ________________________________

Steps completed:   [ ] 1  [ ] 2  [ ] 3  [ ] 4  [ ] 5  [ ] 6  [ ] 7

Date completed:    ________________________________

Key lessons learned during this setup:

  1. Claude Desktop may be installed as a Windows Package app on your PC.
     If so, the real config file is NOT in AppData\Roaming\Claude or
AppData\Local\Claude.
     AppData\Local\Packages\Claude_[YOUR_PACKAGE_ID]\
     LocalCache\Roaming\Claude\

  2. The hammer icon does NOT appear in Claude Desktop v1.1.6679.
     Use the + button in the chat input to verify local-postgres is connected.

  3. The downloaded file is a .zip not a .exe. Must extract after downloading.
     The extracted program is named: pgedge-postgres-mcp.exe

  4. AppData is a hidden folder. Must enable View > Show > Hidden items first.

  5. The config file must include BOTH the existing preferences section
     AND the new mcpServers section — do not replace one with the other.

  6. PGPASSWORD must be included directly in the config file env section.
     The .pgpass file alone was not sufficient for this version.
```

Setup complete! Ask Claude: 'What tables are in my database?' to verify the connection at any time.