

Here's what's happening:

- You installed `dbt-postgres` successfully inside your venv.
- The executables (`dbt.exe`, `dbt-script.py`) were placed in the **Windows-style path**:

Code >

Copy

```
venv\Scripts\
```

- But since you're running **bash**, it doesn't automatically look in `venv\Scripts\` for executables. That's why `dbt --version` says "command not found," even though dbt is installed.

How to fix it

1. Run dbt directly from the Scripts folder

In Git Bash:

Bash >

Copy

```
./venv/Scripts/dbt.exe --version
```

That should show the dbt version.

2. Add the Scripts folder to PATH in bash

Temporarily:

Bash >

Copy

```
export PATH=$PATH:/c/1Programming/01\ Data\ Engineer/01\ DE\ Project\ FreeCodeCamp/01\
```

Then try:

Bash >

Copy

```
dbt --version
```

If it works, you can add that `export PATH=...` line to your `~/.bashrc` so it's always available.

3. Alternative: use PowerShell terminal in VS Code

If you switch your VS Code terminal to **PowerShell**, activation will automatically put

`venv\Scripts` on PATH, so `dbt --version` works without extra steps:

Powershell >

Copy

```
venv\Scripts\Activate.ps1  
dbt --version
```

Recommended approach

Since you're on Windows, dbt works more smoothly in **PowerShell** than Git Bash. If you prefer bash, you'll need to manually add the `venv\Scripts` folder to PATH as shown above.

So: dbt is installed, but bash doesn't know where to find it. Either call it directly with `./venv/Scripts/dbt.exe` or add `venv\Scripts` to PATH.

Do you want me to give you the exact line to put in your VS Code **bash profile** so dbt is always available without typing the long path?