Git Bash + PostgreSQL Survival Guide

This guide provides a concise reference for using Git Bash in combination with PostgreSQL. It includes essential Unix-style commands, Postgres connection and script execution techniques, and best practices for managing database roles, permissions, and automation via Bash.

# 1. Git Bash Basics

Common Unix-style commands used within Git Bash on Windows:

|  |  |  |
| --- | --- | --- |
| Action | Command | Description |
| Where am I? | pwd | Displays the current working directory. |
| List all files | ls -la | Lists all files including hidden ones. |
| Change directory | cd /c/Users/YourName/ | Navigates to the specified path. |
| Make a directory | mkdir foldername | Creates a new directory. |
| Delete a file | rm file.txt | Deletes the specified file. |
| Delete folder | rm -r foldername | Deletes the folder and all its contents. |

# 2. PostgreSQL in Git Bash

Commands and procedures for interacting with PostgreSQL from Git Bash.

## Connecting to PostgreSQL

To connect to a PostgreSQL database:

psql -h hostname -U username -d databasename

## Running SQL Scripts

To run a SQL script from file:

\i ./path/to/script.sql

## Useful psql Meta Commands

|  |  |
| --- | --- |
| Command | Purpose |
| \dn | List all schemas |
| \dt | List tables in current schema |
| \df | List functions |
| \c dbname | Connect to another database |
| \q | Quit the psql interface |

# 3. Automating SQL Execution in Bash

Sample Bash loop to execute multiple SQL scripts:

for file in ./scripts/\*.sql; do  
 echo "Running $file"  
 psql -U username -d dbname -f "$file"  
done

# 4. PostgreSQL Function Execution

|  |  |
| --- | --- |
| Function | Example Call |
| SetAccountSetup | SELECT deploy.SetAccountSetup(...); |
| SetAccountRevoke | SELECT deploy.SetAccountRevoke(...); |
| SetPassword | SELECT info.SetPassword('user', 'newpass'); |
| DropSchemaWithCleanup | SELECT deploy.DropSchemaWithCleanup('schema', TRUE); |
| CreateSchemaWithPermissions | SELECT deploy.CreateSchemaWithPermissions(...); |

# 5. Safety Tips

* • Avoid using 'DROP ROLE postgres'.
* • CREATE DATABASE must be executed outside a transaction block.
* • PostgreSQL does not support CREATE SCHEMA IF NOT EXISTS directly.
* • Always validate \set variables before using them in \connect.