

PROG8850 MySQL CI/CD Automation Project

This project demonstrates automated MySQL schema management using Python and GitHub Actions.

It provides a reproducible environment for managing database changes and testing them in CI/CD pipelines.

Features

- Automated MySQL setup using GitHub Actions services
 - Schema management via SQL scripts (`schema_changes.sql`, `adddept_table.sql`)
 - Python automation with `mysql-connector-python`
 - CI/CD pipeline that runs on every push to the `main` branch
-

Project Structure

.

README.md

schema_changes.sql # Main schema changes (projects table, etc.)

adddept_table.sql # Adds departments table to companydb

run_sql_script.py # Python script to execute SQL scripts

.github/

workflows/

ci_cd_pipeline.yml # GitHub Actions workflow

Getting Started

1. Clone the Repository

```
git clone https://github.com/your-username/your-repo.git
```

```
cd your-repo
```

2. Local Setup (Optional)

- Install MySQL and Python 3.8+ on your machine.

- Install Python dependencies:

```
pip install mysql-connector-python
```

- Run the SQL scripts locally:

```
python run_sql_script.py
```

3. Running the CI/CD Pipeline

- Make sure your changes are committed and pushed to the `main` branch:

```
git add .
```

```
git commit -m "Describe your changes"
```

```
git push origin main
```

- The workflow in `.github/workflows/ci_cd_pipeline.yml` will automatically:

- Start a MySQL service

- Wait for MySQL to be ready

- Set up Python

- Install dependencies

- Run your SQL scripts using ``run_sql_script.py``
-

Checking Workflow Results

1. Go to your repository on GitHub.
 2. Click the **Actions** tab.
 3. Select the latest workflow run (named "MySQL CI").
 4. Review the logs for each step to ensure all SQL commands executed successfully.
-

SQL Scripts

- ``schema_changes.sql``: Creates the ``projects`` table and adds a ``budget`` column.
 - ``adddept_table.sql``: Creates the ``companydb`` database and the ``departments`` table.
-

Python Script

- ``run_sql_script.py``: Reads and executes each SQL script, statement by statement, using ``mysql-connector-python``. Handles duplicate column errors gracefully.
-

Credentials

The workflow and scripts use these MySQL credentials by default:

- Host: 127.0.0.1

- User: root
- Password: Secret5555
- Database: test

You can change these in the workflow file or Python script as needed.

Shutting Down (Local Only)

If you used Ansible or Docker for local setup, you can tear down the environment with:

ansible-playbook down.yml

Author

Keerthana Garimella