Assignment 2: Database Automation

Nidhun Murali – 8981611

Cloud Development and Operations

Conestoga College, Waterloo Campus

Database Automation

PROG8850 - Spring 2025 - Section 1

6th June 2025

Repo Link: https://github.com/nmrepos/PROG8850Assignment2

Question 1

Q1 Link: https://github.com/nmrepos/PROG8850Assignment2/tree/question1

SQL Code

```sql
-- Creating the database
CREATE DATABASE IF NOT EXISTS projectdb;
USE projectdb;

-- Creating the new table named 'projects'
CREATE TABLE IF NOT EXISTS projects (
    project_id INT AUTO_INCREMENT PRIMARY KEY,
    project_name VARCHAR(255) NOT NULL,
    start_date DATE,
    end_date DATE
);

-- Add a new column 'budget' to the existing 'projects' table
ALTER TABLE projects ADD COLUMN budget DECIMAL(10, 2);
```

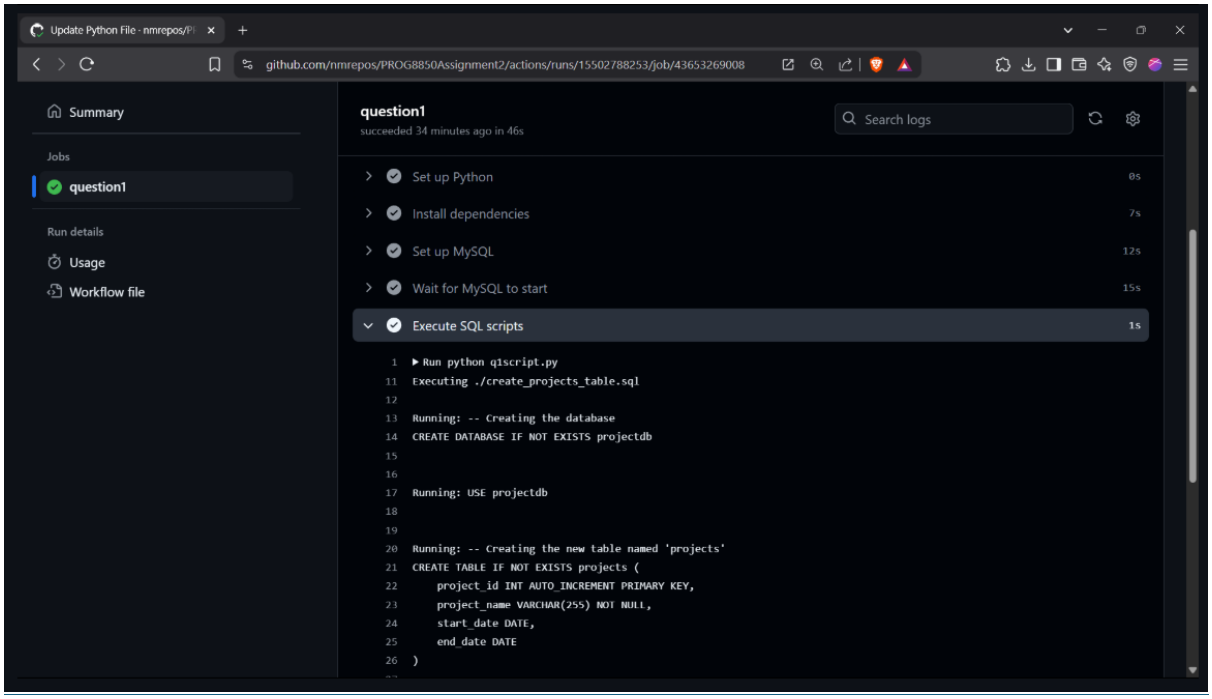
Python Script to Run the SQL Script

```
import mysql.connector
import os
import glob

# Connect to MySQL
connection = mysql.connector.connect(
  host=os.environ['MYSQL_HOST'],
  user=os.environ['MYSQL_USER'],
  password=os.environ['MYSQL_PASSWORD'],
  database=os.environ['MYSQL_DATABASE']
)

cursor = connection.cursor()

# Path to the single SQL file
sql_file = './create_projects_table.sql'
print(f"Executing {sql_file}")

with open(sql_file, 'r') as file:
  sql_script = file.read()

# Split script into commands
commands = sql_script.split(';')

for command in commands:
```

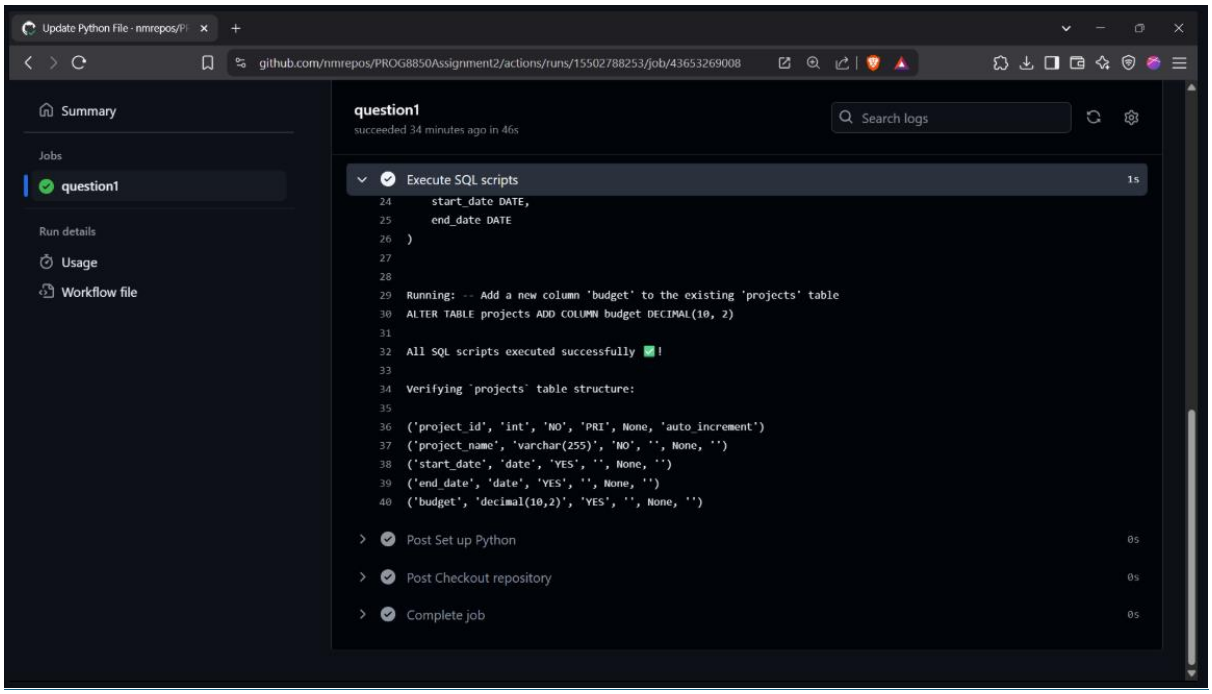
```
    command = command.strip()
    if command:
        print(f"\nRunning: {command}\n")
        cursor.execute(command)



# Commit changes
connection.commit()
print("All SQL scripts executed successfully ✅!")


print("\nVerifying `projects` table structure:\n")
cursor.execute("DESCRIBE projects;")
for row in cursor.fetchall():
    print(row)

# Close connection
cursor.close()
connection.close()
```

Screenshots

Screenshot 1 – Running the Python Automation Script



Screenshot 2 – Verifying the mycompany Database



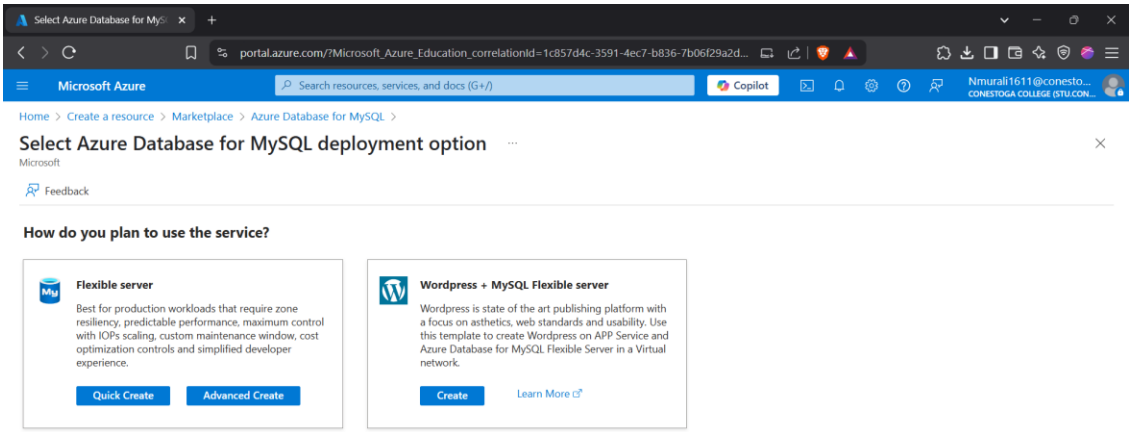
Q2: Azure MySQL Setup and GitHub Actions Configuration

Q2 Github Repo Link: https://github.com/nmrepos/PROG8850Assignment2/tree/question2

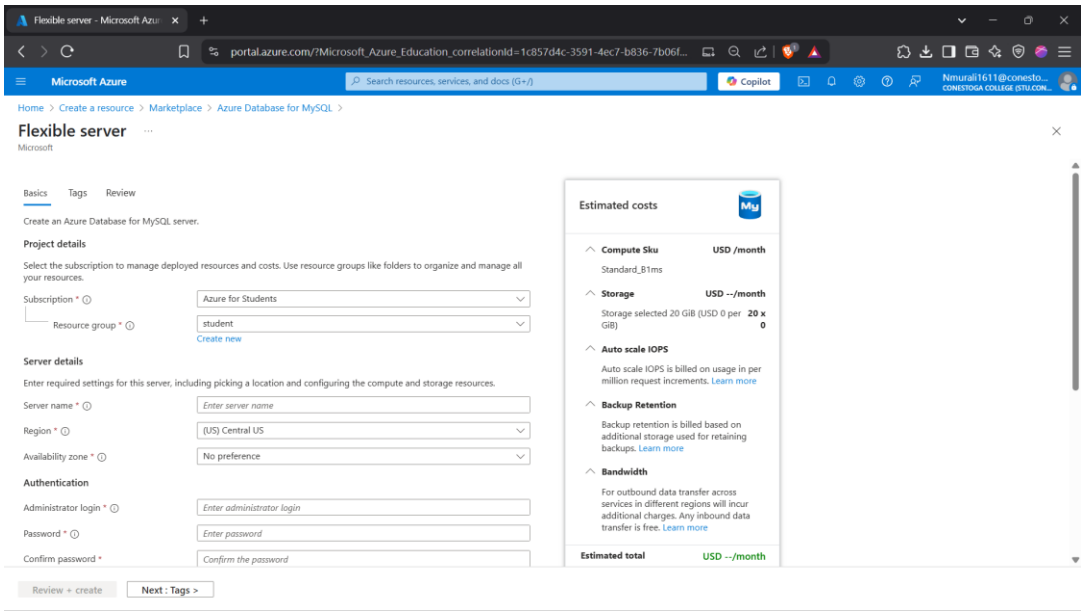
Setting Up Azure MySQL Database

Step 1: Create an Azure MySQL Database Instance

1. Log in to the Azure Portal at https://portal.azure.com
2. Click on "Create a resource" in the left-hand menu
3. Search for "Azure Database for MySQL" and select it
4. Click "Create" and choose "Flexible server Quick Create" as the deployment option

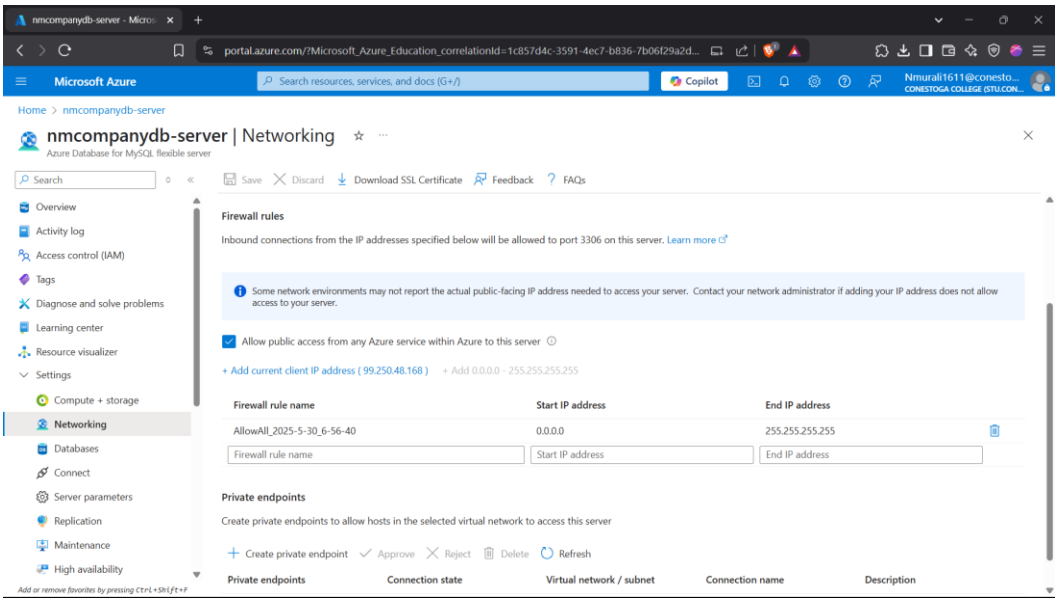


5. Fill in the required details:
    - Subscription: select your Azure subscription
    - Resource group: create a new one or select an existing group
    - Server name: enter a unique name (e.g., nmcompanydb-server)
    - Location: choose a region (US Central)
    - Admin username: create an admin username
    - Password: create and note down a secure password
    - Select Workload Details – Dev/Test
6. Click "Review + create" and then "Create" to deploy the server



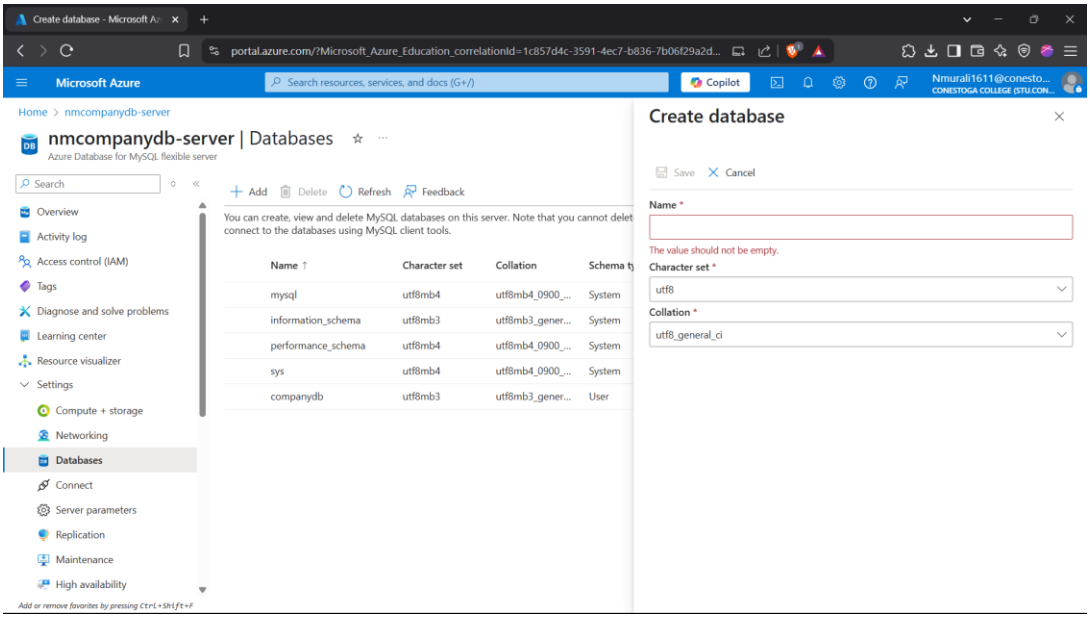
Step 2: Configure Firewall Settings

1. Navigate to your MySQL server resource
2. Under “Settings” click “Networking”
3. Enable “Allow access to Azure services”
4. Add a firewall rule:
    - Name: AllowAll-IP (I am using github actions , if you are using the local machine you can select yout IP, More Secure)
    - Start IP: 0.0.0.0
    - End IP: 255.255.255.255
5. Click “Save”



Step 3: Create the “companydb” Database

1. Under “Settings” click “Databases”
2. Click “+ Add”
3. Enter “companydb” as the name and click “OK”
4. Confirm it appears in the list

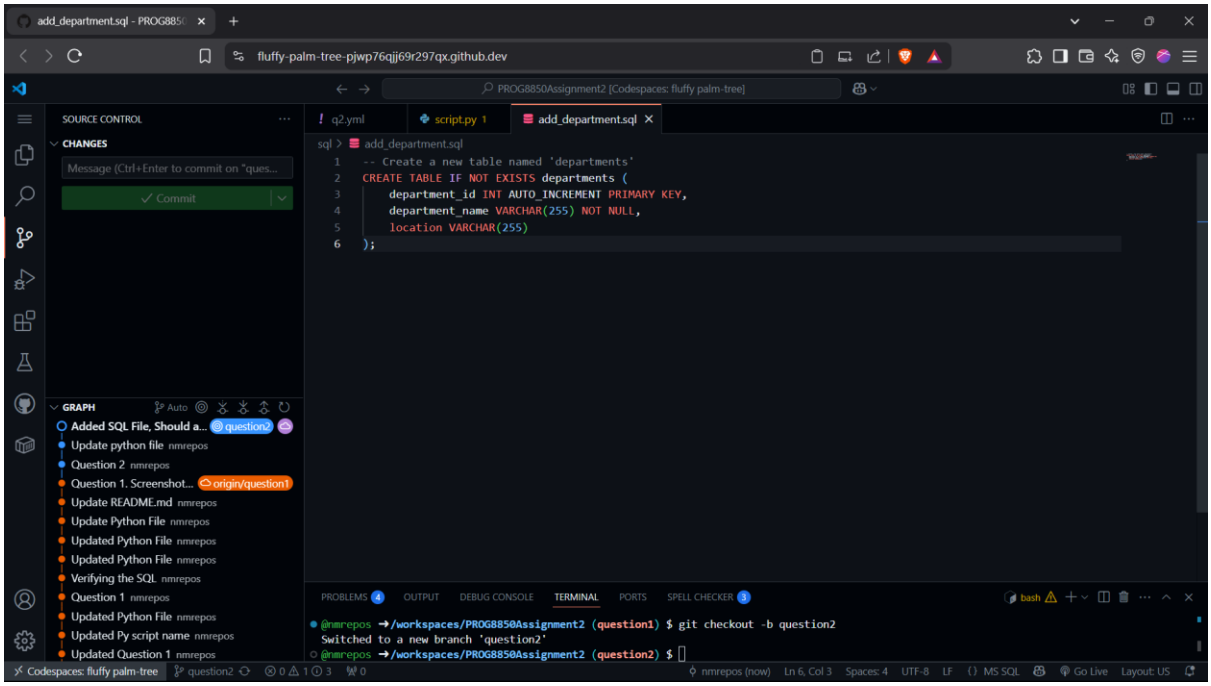


Setting Up GitHub Actions

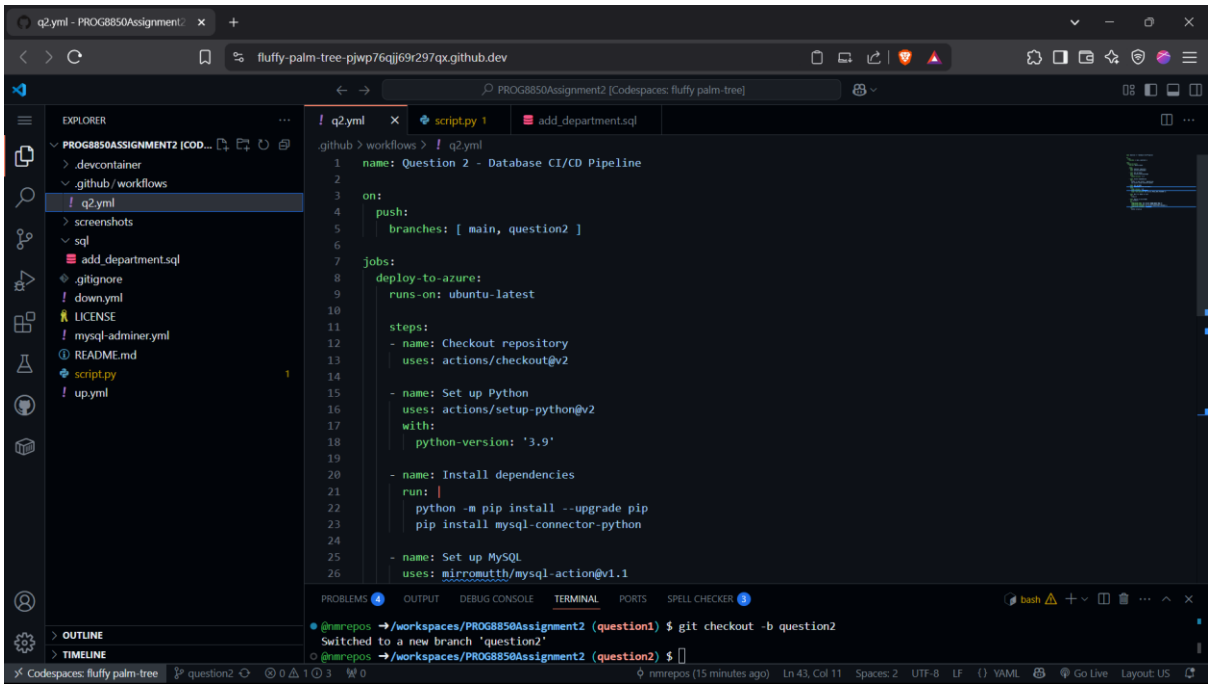
Step 1: Create a GitHub Repository

Step 2: Set Up Repository Structure

1. Open Codespace
2. Make directories:
3. mkdir -p sql .github/workflows
4. Create the SQL scripts in sql folder

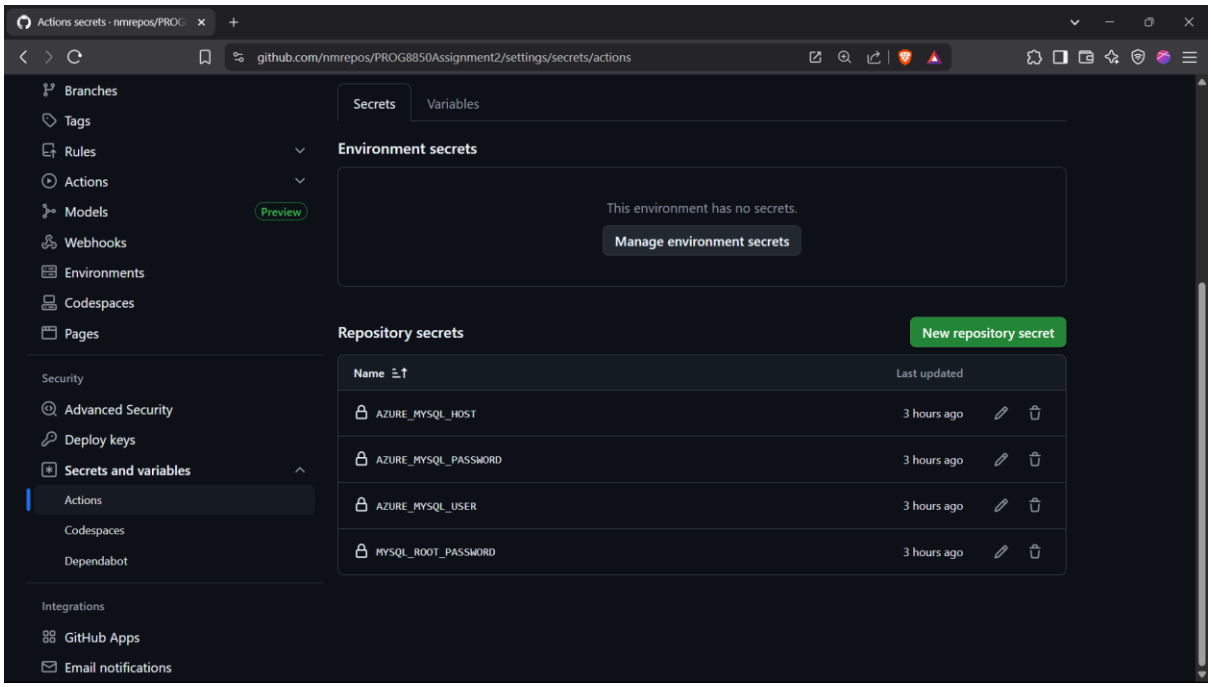


5. Create the workflow in .github/workflows/ folder



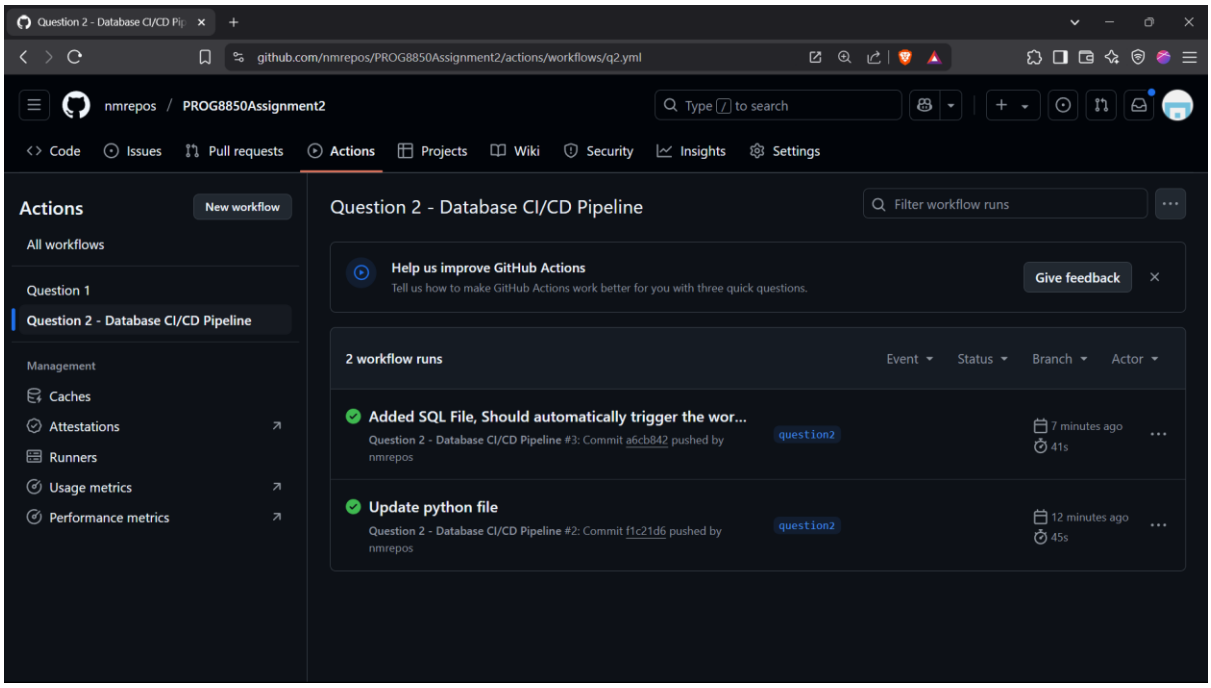
Step 3: Configure GitHub Secrets

1. In your repo on GitHub go to Settings → Secrets and variables → Actions
2. Add these secrets:
    - MYSQL_ROOT_PASSWORD (for local MySQL tests)
    - AZURE_MYSQL_HOST (e.g., nmcompanydb-server.mysql.database.azure.com)
    - AZURE_MYSQL_USER (Azure admin username)
    - AZURE_MYSQL_PASSWORD (Azure admin password)
3. Click “Add secret” for each

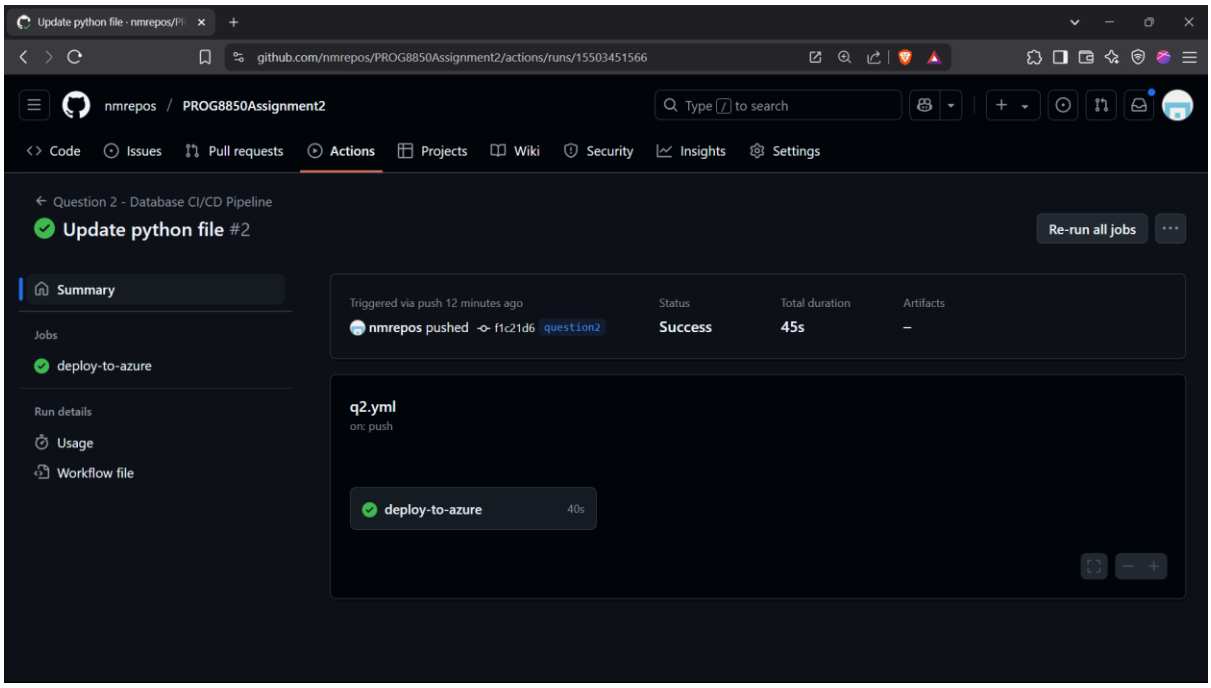


Testing the Workflow

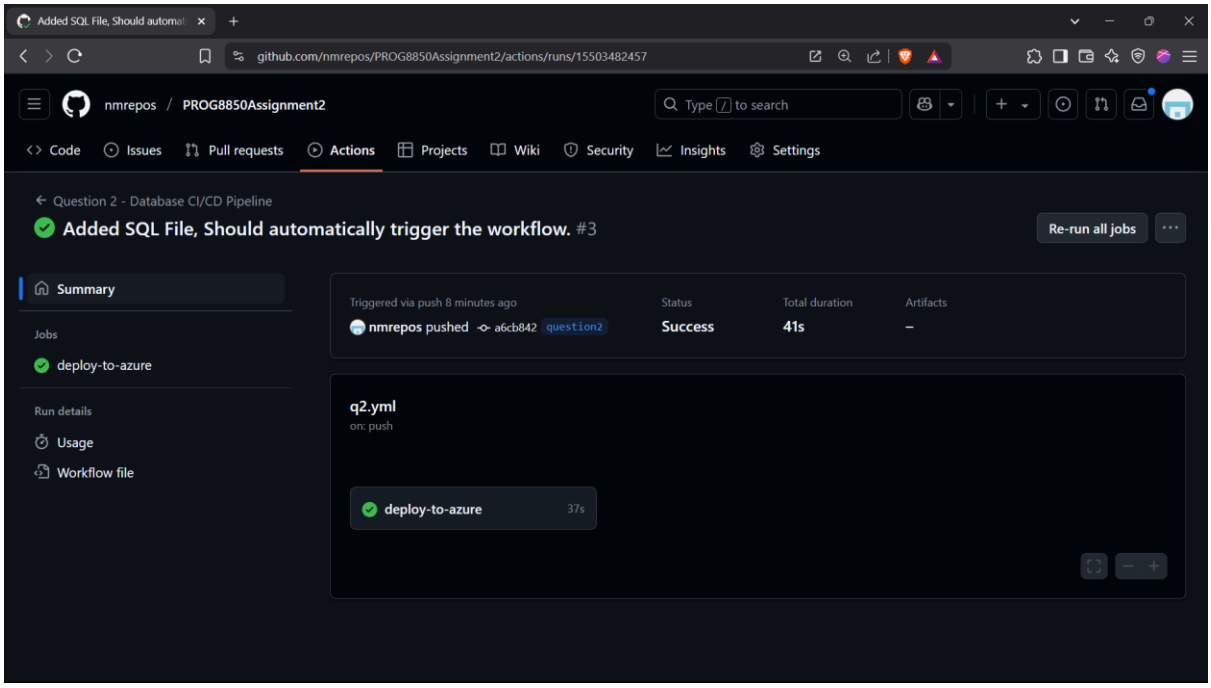
Step 1: Monitor Workflow Execution

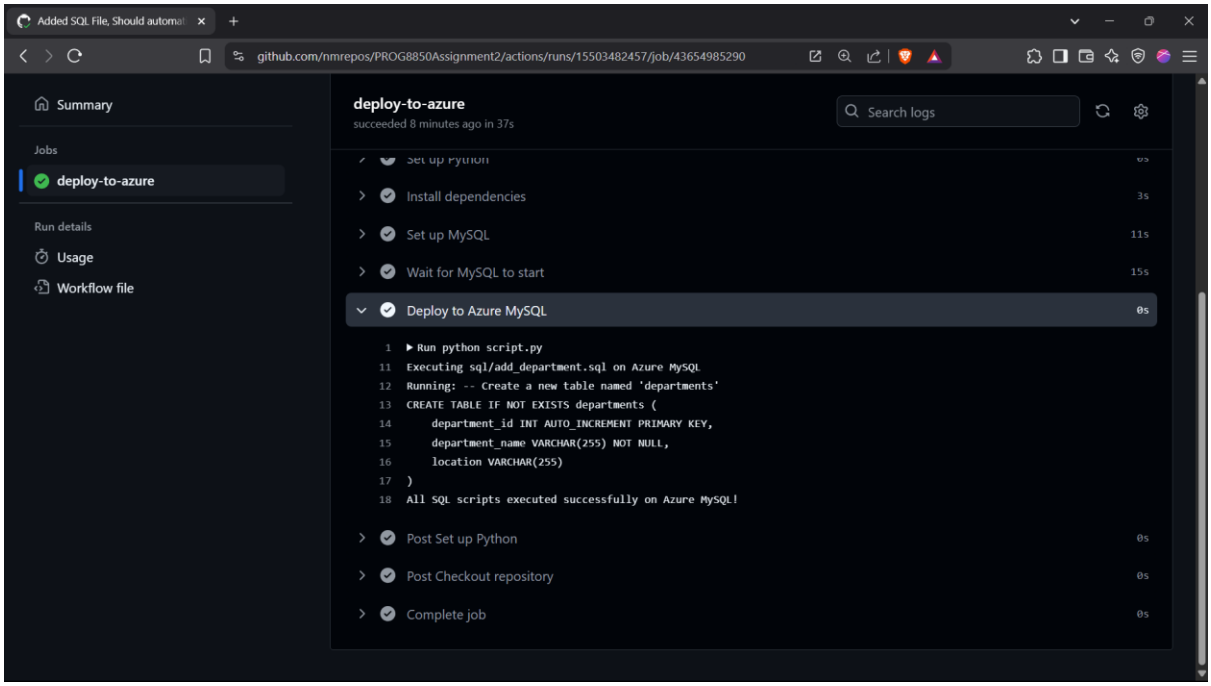


Before Pushing SQL File

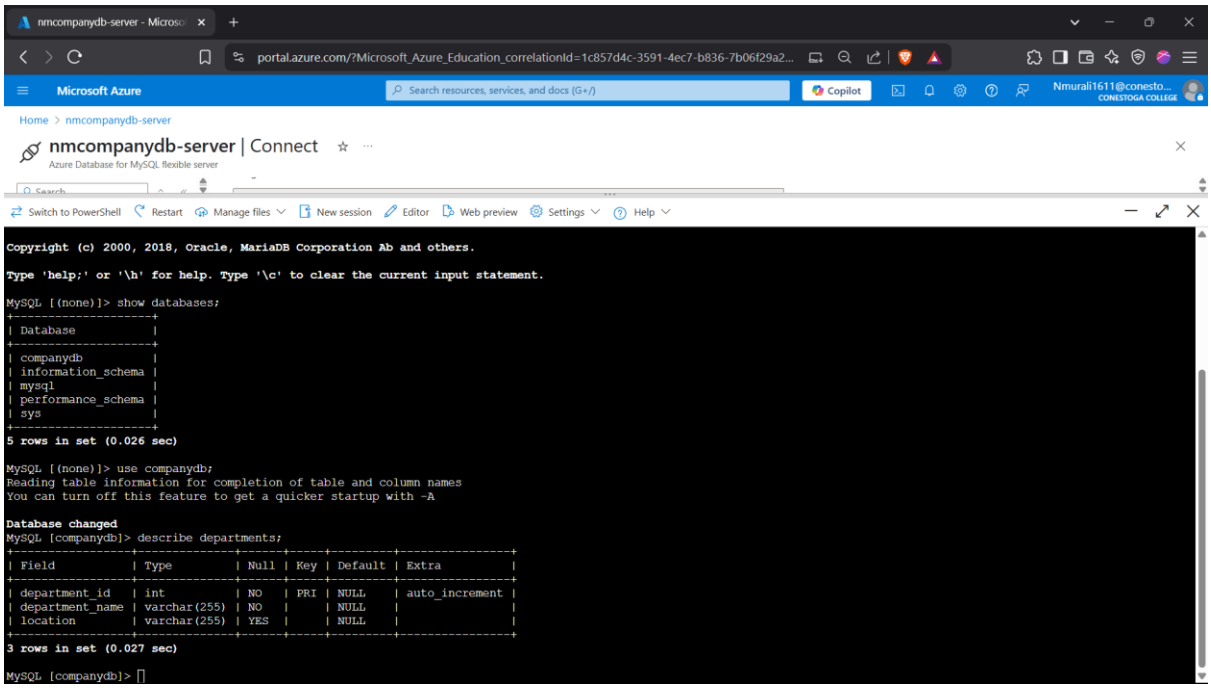


After Pushing SQL File





Verifying the Azure DB for the Changes



Result

The table ‘departments’ with columns ‘department_id’, ‘department_name’, and ‘location’ to the ‘companydb’ database in AZURE.

Final Outcome

Whenever a SQL file is pushed to the SQL folder, it should automatically trigger the workflow and apply the change to the Azure Database.

Github Repo Link: https://github.com/nmrepos/PROG8850Assignment2.git