

Lab 8: AWS/VSCode Setup

GW CS 2541W: Database Systems and Team Projects - 2026

**Prof. Gabe Parmer, Bella Dayrit, Sydney Berritt, Max Eichholz, Aditya Arjun,
Emil Abbasov**

AWS RDS Setup

Create a MySQL Database Server in the Cloud



Today's lab you will be actively following along with the slides on your computer

Today's lab is going to be different from all the all the other lab's from the rest of the semester

Setting up AWS RDS is very finicky and this lab will require you to follow along step by step exactly. One singular mess up at an earlier step can completely mess up your configuration and ruin your set up for the **final project**, leading to unexpected bugs and issues.

We strongly ask you to pay very close attention and raise you hand immediately if you are unsure about any specific step of this process that we will be walking you through

Log into AWS Academy and click **Modules** → **AWS Academy Learner Lab**

▼ AWS Academy Learner Lab



Launch AWS Academy Learner Lab

▼ AWS Academy Learner Lab Resources



Demo - How to Access Learner Lab



Demo - General Troubleshooting Tips



Demo - How to Launch Services through AWS Console



Learner Lab Activity - CodeWhisperer

Click **Start Lab**, wait, then click **AWS** once green

ALLFv1-16958 > Modules > Learner Lab Foundatio... > Learner Lab - Foundational Services

Used \$0 of \$100 03:55 Start Lab End Lab AWS Details Readme Reset

Home Modules Discussions

EN-US

Learner Lab - Foundational Level

- [Environment Overview](#)
- [Environment Navigation](#)
- [Access the AWS Management Console](#)
- [Region restriction](#)
- [Service usage and other restrictions](#)
- [Using the terminal in the browser](#)
- [Running AWS CLI commands](#)
- [Using the AWS SDK for Python](#)
- [Preserving your budget](#)
- [Accessing EC2 Instance\(s\)](#)
- [SSH Access to EC2 Instances](#)
- [SSH access from Windows](#)
- [SSH access from a Mac](#)

Environment Overview

This Learner Lab provides a sandbox

Thumbs up my message in discord once your team has completed this step

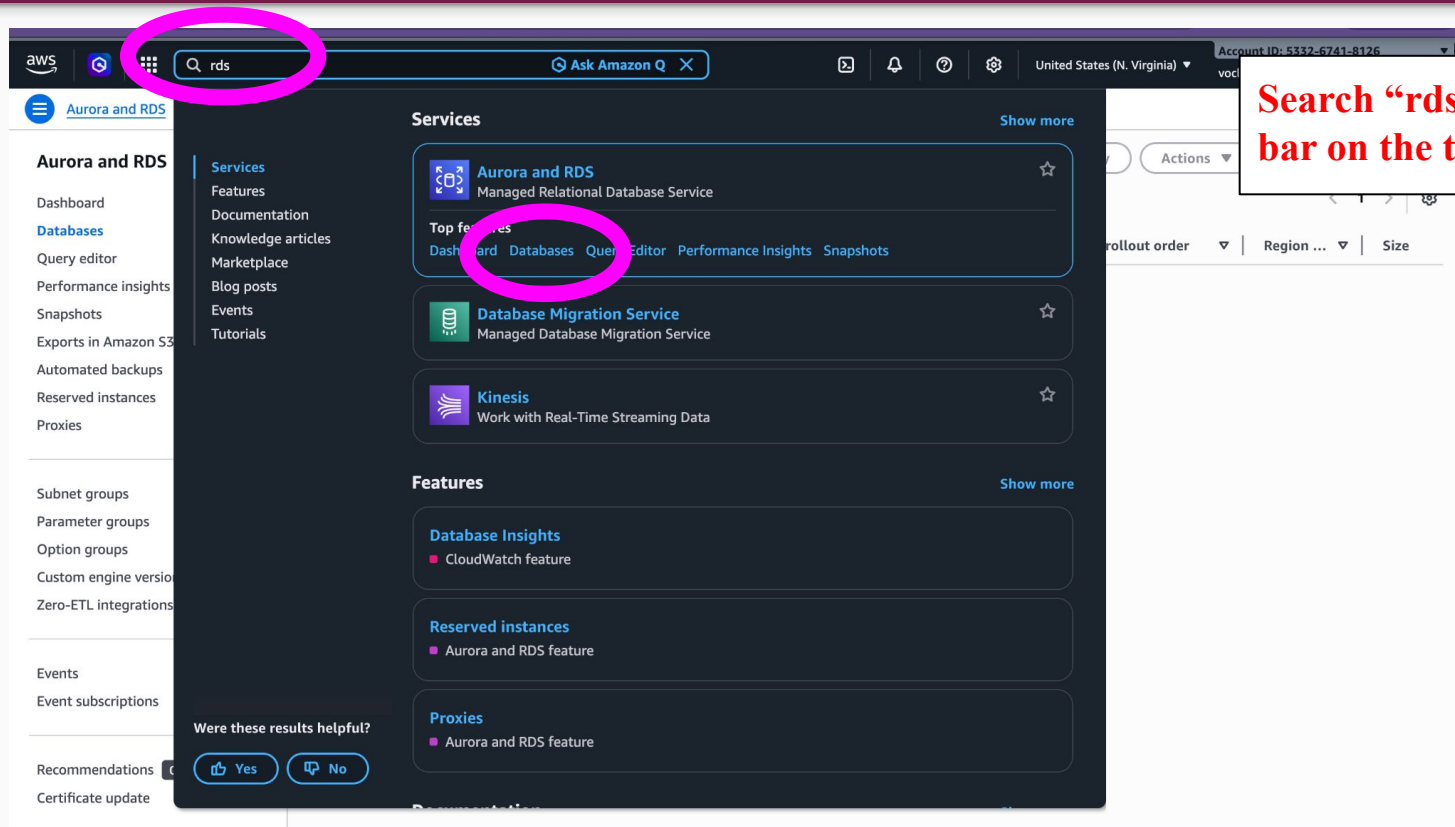
Thumbs down if struggling

Reset Student Leave Student View

63 You are currently logged into Student View

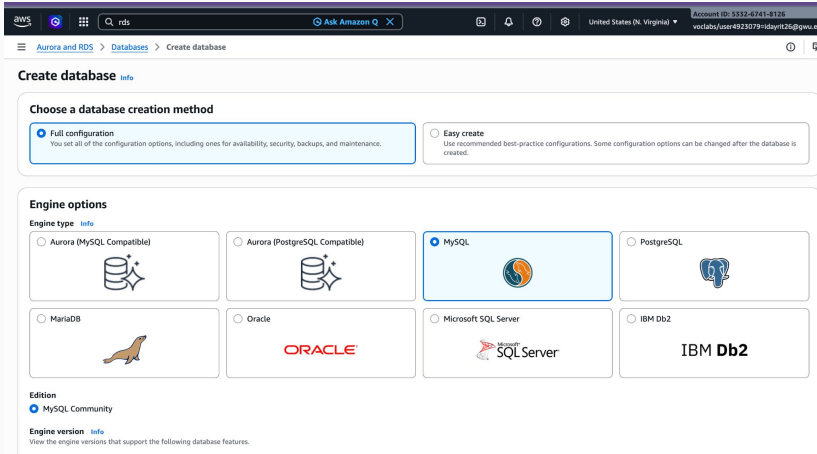
Resetting the test student will clear all history for this student, allowing you to view the course as a brand new student.

In AWS Management Console - Type "RDS" into the search bar, then click Databases



Search "rds" in the search bar on the top left !

Click Create Database. Use these options when filling out the database configuration



Method: Full Configuration

Engine Type: MySQL

Template: Sandbox

Instance Identifier: something like REGS26-LastName

Master Password: something secure that you write down!

Storage Autoscaling: Disable (click on additional storage configuration)

Public Access: YES

Additional Configuration

Initial DB name: university

Disable Automatic Backups

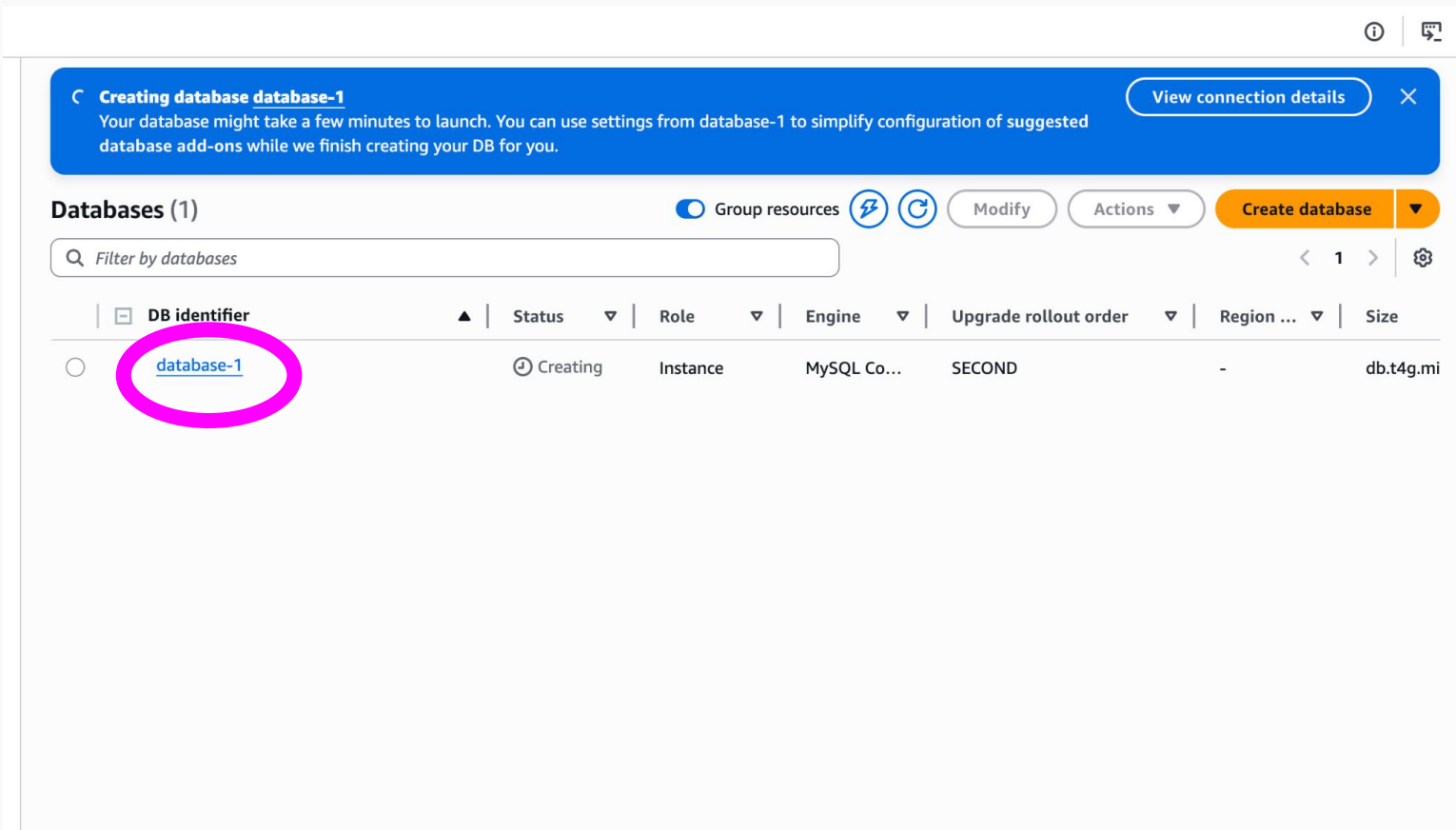
Disable Auto minor version upgrade

Then click **Create Database**

**Thumbs down if
struggling**

**Thumbs up my message in discord for your
team when you have complete this step**

After creating the database, click on the database you just created



The screenshot displays a database management interface. At the top, a blue notification banner reads: "Creating database database-1. Your database might take a few minutes to launch. You can use settings from database-1 to simplify configuration of suggested database add-ons while we finish creating your DB for you." A "View connection details" button is located on the right of the banner. Below the banner, the "Databases (1)" section features a search bar with the placeholder "Filter by databases", a "Group resources" toggle, and buttons for "Modify", "Actions", and "Create database". A table below lists the database resources with columns: DB identifier, Status, Role, Engine, Upgrade rollout order, Region, and Size. The first row shows a database named "database-1" with a status of "Creating", role of "Instance", engine of "MySQL Co...", rollout order of "SECOND", and size of "db.t4g.mi". The "database-1" text in the table is circled in pink.

DB identifier	Status	Role	Engine	Upgrade rollout order	Region ...	Size
database-1	Creating	Instance	MySQL Co...	SECOND	-	db.t4g.mi

Click on Endpoints after navigating to this page and scroll down to additional configurations

database-1



Modify

Actions ▾

Summary

DB identifier
database-1

Status
🕒 Creating

Role
Instance

Engine
MySQL Community

Recommendations

CPU
-

Class
db.t4g.micro

Current activity

Region & AZ
us-east-1d



Connectivity & security

Monitoring

Logs & events

Configuration

Zero-ETL integrations

Maintenance & backups



Connect using [Info](#)



Code snippets

Use when connecting through SDK, APIs, or third-party tools including agents.



CloudShell

Use for a quick access to AWS CLI that launches directly from the AWS Management Console.



Endpoints

Use when connecting through any IDE interface.

Database name

📄 university

Master username

📄 admin

Internet access gateway

🔌 Disabled

Endpoint type

Instance endpoint ▾

▼ **Additional configurations**

Click on the highlighted text under VPC security groups

▼ Additional configurations

Connectivity & security

Endpoint & port

Endpoint

 database-1.c1yyskuwmzuf.us-east-1.rds.amazonaws.com

Port

3306

Networking

Availability Zone

us-east-1d

VPC

[vpc-01e3933f9fbc3d313](#)

Subnet group

default-vpc-01e3933f9fbc3d313

Subnets

[subnet-04a155eb6df1f2973](#)

[subnet-04245dabc9d27f1f9](#)

[subnet-00ec4e0d4159a63d4](#)

[subnet-0bbfda3f64d6c0f79](#)

[subnet-06deab5d174433f2a](#)

[subnet-0c24652bee79f00c8](#)

Network type

IPv4

Security

VPC security groups

[default \(sg-09f5ad90391ea6903\)](#)

 Active

Publicly accessible

Yes

Certificate authority [Info](#)

rds-ca-rsa2048-g1

Certificate authority date

May 25, 2061, 19:34 (UTC-04:00)

DB instance certificate expiration date

March 20, 2027, 18:52 (UTC-04:00)

Add a new Inbound Rule - Port 3306 - Source - Custom - Enter IP 0.0.0.0/0

EC2 > Security Groups > sg-03846ac637a32371f - default > Edit inbound rules

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sg-089996e2a73bcdbe6	All traffic	All	All	Custom	
-	Custom TCP	TCP	3306	Custom	

[Add rule](#)

[Cancel](#) [Preview changes](#) [Save rules](#)

Thumbs down if struggling

Thumbs up my message in discord once your team has completed this step

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Next, navigate to your database dashboard to check your DB creation status. When available, copy host info.

The screenshot shows the AWS RDS console for a database instance named 'database-1'. The left sidebar contains navigation options like 'Dashboard', 'Databases', 'Query editor', and 'Performance insights'. The main content area is titled 'Connect using' and shows three options: 'Code snippets', 'CloudShell', and 'Endpoints' (which is selected). Below this, there are fields for 'Database name' (university), 'Master username' (admin), 'Internet access gateway' (Disabled), and 'Port' (3306). The 'Endpoint type' is set to 'Instance endpoint'. Under 'Additional configurations', the 'Connectivity & security' section is expanded, showing 'Endpoint & port' with the endpoint 'database-1.c1yyskuwmzuf.us-east-1.rds.amazonaws.com' circled in pink. Other details include 'Availability Zone' (us-east-1d), 'VPC' (vpc-01e3933f9fbc3d313), 'Subnet group' (default-vpc-01e3933f9fbc3d313), and 'Subnets' (subnet-04a155eb6df1f2973, subnet-04245dabc9d27f1f9, subnet-00ec4e0d4159a63d4, subnet-0bbfda3f64d6c0f79, subnet-06deab5d174433f2a). The 'Security' section shows 'VPC security groups' (default (sg-09f5ad90391ea6903) Active), 'Publicly accessible' (Yes), 'Certificate authority' (rds-ca-rsa2048-g1), 'Certificate authority date' (May 25, 2061, 19:34 (UTC-04:00)), and 'DB instance certificate expiration date'.

Aurora and RDS > **Databases** > database-1

Connect using | Info

- Code snippets
Use when connecting through SDK, APIs, or third-party tools including agents.
- CloudShell
Use for a quick access to AWS CLI that launches directly from the AWS Management Console.
- Endpoints
Use when connecting through any IDE interface.

Database name
university

Master username
admin

Internet access gateway
Disabled

Port
3306

Endpoint type
Instance endpoint

▼ **Additional configurations**

Connectivity & security

Endpoint & port

Endpoint
database-1.c1yyskuwmzuf.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1d

VPC
vpc-01e3933f9fbc3d313

Subnet group
default-vpc-01e3933f9fbc3d313

Subnets
subnet-04a155eb6df1f2973
subnet-04245dabc9d27f1f9
subnet-00ec4e0d4159a63d4
subnet-0bbfda3f64d6c0f79
subnet-06deab5d174433f2a

Security

VPC security groups
default (sg-09f5ad90391ea6903)
Active

Publicly accessible
Yes

Certificate authority | Info
rds-ca-rsa2048-g1

Certificate authority date
May 25, 2061, 19:34 (UTC-04:00)

DB instance certificate expiration date
May 25, 2061, 19:34 (UTC-04:00)

Test from AWS Academy Lab Console (other tab)

The screenshot shows the AWS Academy Lab Console interface. The browser tab is titled "Learner Lab - Foundational Services". The URL is <https://awsacademy.instructure.com/courses/16958/modules/items/1398292>. The page shows the "Learner Lab - Foundational Services" page with a terminal window open. The terminal output is as follows:

```
ddd_v1_w_9M5_1123374@runweb514666:~$ ls
ddd_v1_w_9M5_1123374@runweb514666:~$ mysql -h instructor-1.ck8ualavedvt.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor.  Copyright 2000-2013 Oracle and/or its affiliates.
Your MySQL connection id is 16
Server version: 8.0.27 Source distribution

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| performance_schema     |
| sys                     |
| university              |
+-----+
5 rows in set (0.07 sec)

mysql> use university;
Database changed
mysql> show tables;
Empty set (0.07 sec)

mysql>
```

Commands:
**mysql -h [endpoint] -u
admin -p**

Enter password

**mysql> show
databases;**

mysql> use university;

mysql> show tables;

**Thumbs up my message in
discord once your team**

Thumbs down if

MySQL vs SQLite

Code / System Differences



SQLite vs MySQL

SQLite: embedded database

- Simple, single file based DB integrated in your software
- Great for mobile apps, or single user applications

MySQL: full featured database server

- Independent server application
- Can be deployed across a cluster of servers
- Supports larger scale and stronger reliability guarantees

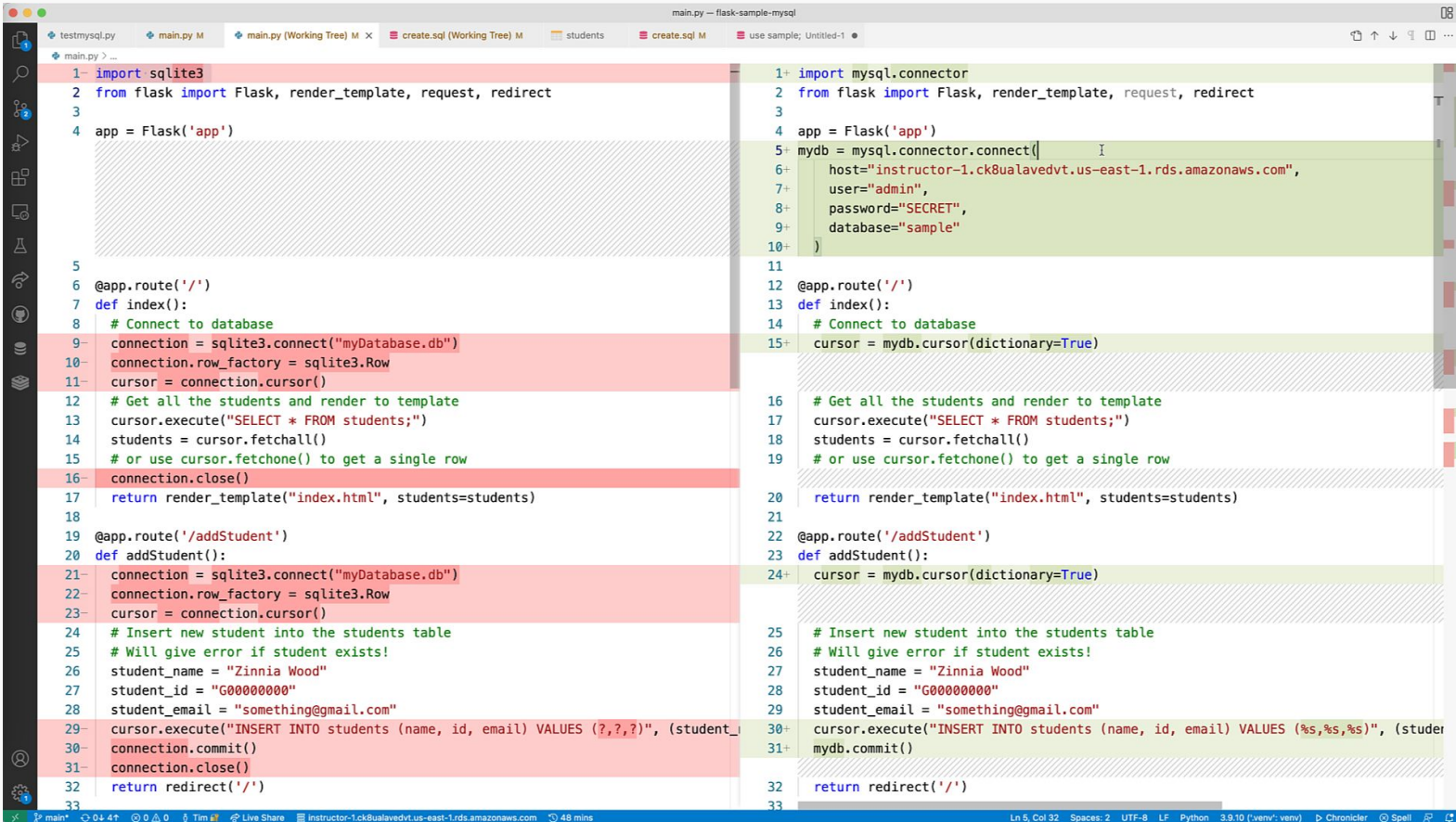
SQL Code Differences

```
create.sql -- flask-sample-mysql
testmysql.py main.py M main.py (Working Tree) M create.sql (Working Tree) M X students create.sql M use sample; Untitled-1
create.sql > ...
1- PRAGMA foreign_keys=off;
2
3 DROP TABLE IF EXISTS students;
4 CREATE TABLE students (
5   id      varchar(32) not null PRIMARY KEY,
6   name    varchar(50) not null,
7   email   varchar(50) not null
8 );
9-
10- PRAGMA foreign_keys=on;
11
12 INSERT INTO students VALUES ('G12345678', 'Jett Jacobs', 'jacobsemail@fakeemail.com');
13 INSERT INTO students VALUES ('G22489071', 'Alex Coleman', 'alexcoleman@fakeemail.com');
14 INSERT INTO students VALUES ('G82915273', 'Ethan Baron', 'ethan@fakeemail.com');
15 INSERT INTO students VALUES ('G22004676', 'Cat Meadows', 'cat@fakeemail.com');
16
17 UPDATE students SET name = 'Catherine Meadows' WHERE id = 'G22004676';

1- CREATE DATABASE sample
2-   DEFAULT CHARACTER SET = 'utf8mb4';
3- use sample;
4-
5- SET FOREIGN_KEY_CHECKS=0;
6
7 DROP TABLE IF EXISTS students;
8 CREATE TABLE students (
9   id      varchar(32) not null PRIMARY KEY,
10  name    varchar(50) not null,
11  email   varchar(50) not null
12 );
13- SET FOREIGN_KEY_CHECKS=1;
14
15 INSERT INTO students VALUES ('G12345678', 'Jett Jacobs', 'jacobsemail@fakeemail.com');
16 INSERT INTO students VALUES ('G22489071', 'Alex Coleman', 'alexcoleman@fakeemail.com');
17 INSERT INTO students VALUES ('G82915273', 'Ethan Baron', 'ethan@fakeemail.com');
18 INSERT INTO students VALUES ('G22004676', 'Cat Meadows', 'cat@fakeemail.com');
19
20 UPDATE students SET name = 'Catherine Meadows' WHERE id = 'G22004676';

Ln 9, Col 24 (11 selected) Spaces: 2 UTF-8 LF SQL Chronieler Spell
```

Python Code Differences



```
main.py - flask-sample-mysql
testmysql.py main.py M main.py (Working Tree) M x create.sql (Working Tree) M students create.sql M use sample; Untitled-1
main.py > ...
1- import sqlite3
2 from flask import Flask, render_template, request, redirect
3
4 app = Flask('app')
5
6 @app.route('/')
7 def index():
8     # Connect to database
9     connection = sqlite3.connect("myDatabase.db")
10    connection.row_factory = sqlite3.Row
11    cursor = connection.cursor()
12    # Get all the students and render to template
13    cursor.execute("SELECT * FROM students;")
14    students = cursor.fetchall()
15    # or use cursor.fetchone() to get a single row
16    connection.close()
17    return render_template("index.html", students=students)
18
19 @app.route('/addStudent')
20 def addStudent():
21    connection = sqlite3.connect("myDatabase.db")
22    connection.row_factory = sqlite3.Row
23    cursor = connection.cursor()
24    # Insert new student into the students table
25    # Will give error if student exists!
26    student_name = "Zinnia Wood"
27    student_id = "G00000000"
28    student_email = "something@gmail.com"
29    cursor.execute("INSERT INTO students (name, id, email) VALUES (?, ?, ?)", (student_name, student_id, student_email))
30    connection.commit()
31    connection.close()
32    return redirect('/')
33
1+ import mysql.connector
2 from flask import Flask, render_template, request, redirect
3
4 app = Flask('app')
5+ mydb = mysql.connector.connect({
6+     host="instructor-1.ck8ualavedvt.us-east-1.rds.amazonaws.com",
7+     user="admin",
8+     password="SECRET",
9+     database="sample"
10+ })
11
12 @app.route('/')
13 def index():
14     # Connect to database
15+    cursor = mydb.cursor(dictionary=True)
16
17 # Get all the students and render to template
18 cursor.execute("SELECT * FROM students;")
19 students = cursor.fetchall()
20 # or use cursor.fetchone() to get a single row
21
22 return render_template("index.html", students=students)
23
24 @app.route('/addStudent')
25 def addStudent():
26+    cursor = mydb.cursor(dictionary=True)
27
28 # Insert new student into the students table
29 # Will give error if student exists!
30 student_name = "Zinnia Wood"
31 student_id = "G00000000"
32 student_email = "something@gmail.com"
33+    cursor.execute("INSERT INTO students (name, id, email) VALUES (%s,%s,%s)", (student_name, student_id, student_email))
34+    mydb.commit()
35
36 return redirect('/')
37
Ln 5, Col 32 Spaces: 2 UTF-8 LF Python 3.9.10 ('venv': venv) > Chronidier Spell
```

MySQL vs SQLite Summary

MySQL allows you to run a database on a separate server than your application

—This helps with performance, scalability, and reliability

SQL query syntax is mostly the same

—Remember to **"use"** a database

Python code is mostly the same

—Create one MySQL connection in global scope, then create cursors for each route

—use **%s** to fill in query parameters instead of **?**

Python Virtual Environments

And installing Flask Library



What is a Python Virtual Environment ?

A virtual environment in Python is an isolated environment on your computer!

Allows you to manage project-specific dependencies without interfering with other projects or the original Python installation.

Think of a virtual environment as a separate container for each Python project. Each container:

- Has its own Python interpreter
- Has its own set of installed packages
- Is isolated from other virtual environments
- Can have different versions of the same package

Good for team projects when your teammates have different computers than you, and you don't need to worry about conflicting versions of packages

Extra Resources for Virtual Environments: https://www.w3schools.com/python/python_virtualenv.asp

Virtual Environment

Create environment

```
python3 -m venv .venv
```

Tell VS Code to use the environment when it shows a popup!

Activate environment

```
source .venv/bin/activate (mac)
```

```
.venv\Scripts\activate (windows)
```

(LATER, after installing all modules)

Save required package list

```
pip3 freeze > requirements.txt
```

Install from required package list (on another machine)

```
pip install -r requirements.txt
```

Use this to save a list of packages you have installed. Then add the file to your git repository

On another computer you could use this to load all the libraries needed by the repository

Check

Will say "(.venv)"
above every command
prompt in terminal if
working correctly

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS zsh + - [ ] [ ] ... ^ x
...
● sameenahmad@Sameens-MacBook-Pro ~ % python3 -m venv .venv
● sameenahmad@Sameens-MacBook-Pro ~ % source .venv/bin/activate
○ (.venv) sameenahmad@Sameens-MacBook-Pro ~ % █
```

Thumbs down if struggling

Thumbs up my message in discord once your team has completed this step

Flask Module Installation

Inside your virtual environment terminal:

```
pip3 install flask
```

MySQL Local Setup

VS Code Extension and Python Library



MySQL Extension for VS Code

A screenshot of the VS Code marketplace search results for the keyword 'mysql'. The search bar at the top contains the text 'mysql'. Below the search bar, several extensions are listed. The first extension is 'MySQL' by Jun Han, with 838K downloads and a 3.5 star rating. The second extension is 'MySQL Database manager for MySQL/MariaDB' by cweijan, with 203ms load time, 446,350 downloads, and a 5-star rating. The third extension is 'MySQL Syntax' by Jake Bathman, with 335K downloads and a 3-star rating. The fourth extension is 'SQLTools MySQL/MariaDB' by Matheus Teixeira, with 291K downloads and a 5-star rating. The fifth extension is 'mysql-inline-decorator' by odubuc, with 90K downloads and a 3-star rating. The sixth extension is 'MySQL Statement Runner' by Jared Black, with 36K downloads and a 5-star rating. The seventh extension is 'MySQL Autocomplete' by nespinozacr, with 7K downloads and a 5-star rating. The eighth extension is 'ES7 JavaScript/Node.js' by abrahamwilliam007, with 23K downloads and a 5-star rating. Each extension entry includes an 'Install' button and a dropdown arrow.

A screenshot of the MySQL extension page in VS Code. The extension is titled 'MySQL' by cweijan, version 4.8.6. It has 446,350 downloads and a 5-star rating (138 reviews). The extension is currently installed and enabled globally. A pink oval highlights the extension name and version. A callout box with a pink background and black text says 'Get the one by cweijan!'. Below the extension name, there are buttons for 'Disable', 'Uninstall', and a refresh icon. Below the buttons, it says 'This extension is enabled globally.' At the bottom of the page, there are tabs for 'Details', 'Feature Contributions', 'Changelog', and 'Runtime Status'. Below the tabs, there is a section titled 'Database Client for Visual Studio Code' with a badge for 'v4.8.6', 'installs 447.38K', 'stars 1.7k', 'rating 4.4/5 (139)', and 'license MIT'. Below this section, there is a paragraph of text: 'This project is a database client for VSCode, supports manager MySQL/MariaDB, PostgreSQL, SQLite, Redis, ClickHouse, 达梦, and ElasticSearch, and works as an SSH client, boost your maximum productivity!'. Below the paragraph, there is a link to the project site: 'Project site: [vscode-database-client](#), [中文文档](#)'. At the bottom of the page, there is a logo for 'Database Client' with a stack of three cylinders (yellow, orange, blue).

Click DB icon in left menu, Add Database, fill in your RDS connection info

The screenshot shows the 'Connect Server' dialog in DBeaver. The left sidebar has a database icon circled in pink. The dialog has fields for Name, Group, Connection Scope (Global selected), Read Only (disabled), and Sort (10). Under 'Server Type', MySQL is selected. The 'Host' field contains '127.0.0.1' and is annotated with 'RDS Endpoint'. The 'Port' field contains '3306'. The 'Username' field contains 'root' and is annotated with 'Admin information that you created earlier'. The 'Database' field contains 'Connection d'. Other fields include Connect Timeout (5000), Socket Path (Unix Socket Path), Time Zone (+00:00), and SSH Tunnel options (disabled). 'Connect' and 'Close' buttons are at the bottom.

Thumbs down if struggling

Click here to add the new database connection

Click on this icon to navigate to this extension

Thumbs up my message in discord once your team has completed this step

RDS Endpoint

Admin information that you created earlier

MySQL Python Module

Inside your virtual environment terminal:

```
pip3 install mysql-connector-python
```

Now save the packages

(Later, after installing all modules)

Save required package list

```
pip3 freeze > requirements.txt
```

Install from required package list (on another machine)

```
pip3 install -r requirements.txt
```

Use this to save a list of packages you have installed. Then add the file to your git repository

On another computer you could use this to load all the libraries needed by the repository