

Provisioning Amazon RDS database Using AWS Academy Learner Lab

Table of Contents

Introduction	2
Tasks	2
Prerequisites	2
Implementation	3
Create a MySQL DB Instance.....	3
Connect to the Database using MySQL Workbench.....	9
Create tables, insert data, and run queries	11
Delete the database instance	14

Introduction

In this article, you will learn how to create a MySQL database system, connect to the database, run queries against the database, and delete the database instance. We will do this using the Amazon Relational Database Service (Amazon RDS) within the AWS Academy Learner Lab (LL).

Tasks

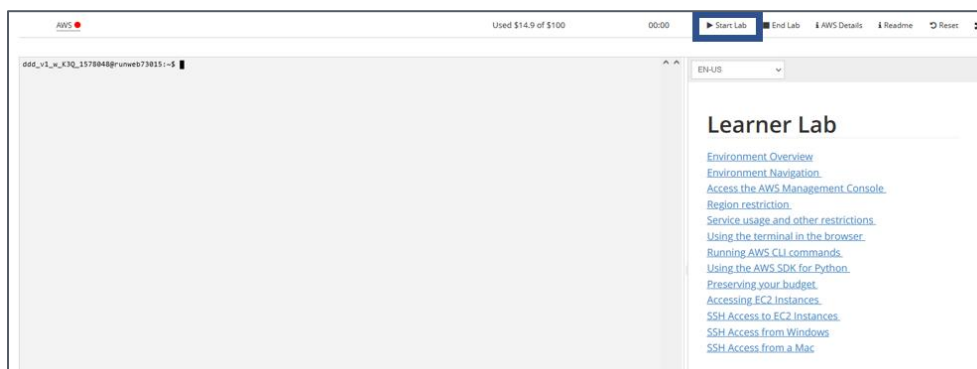
In this article, you will:

- Create an environment to run a MySQL database
- Connect to the database using the MySQL Workbench
- Create tables, insert data, and run queries
- Delete the database instance

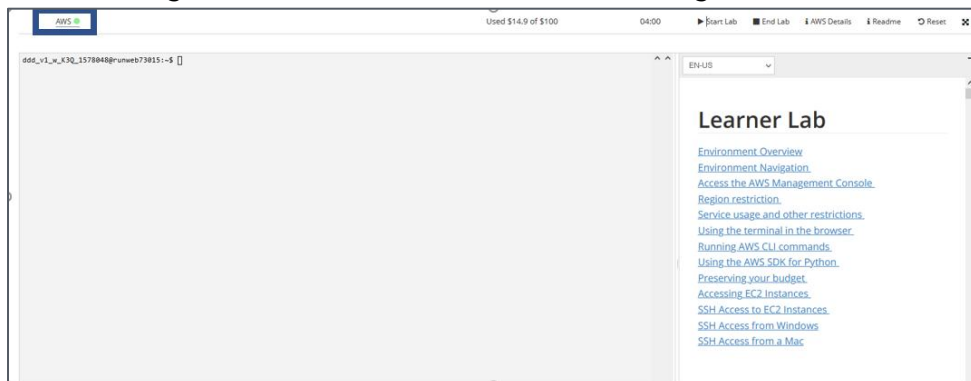
Prerequisites

Before continuing with the following activities, you will need:

- **Access to the AWS Academy Learner (LL) course:** From the Student View of the LL, click the Start Lab button as shown in the screenshot below:



- After starting the lab, wait until the AWS icon turns green as shown below:

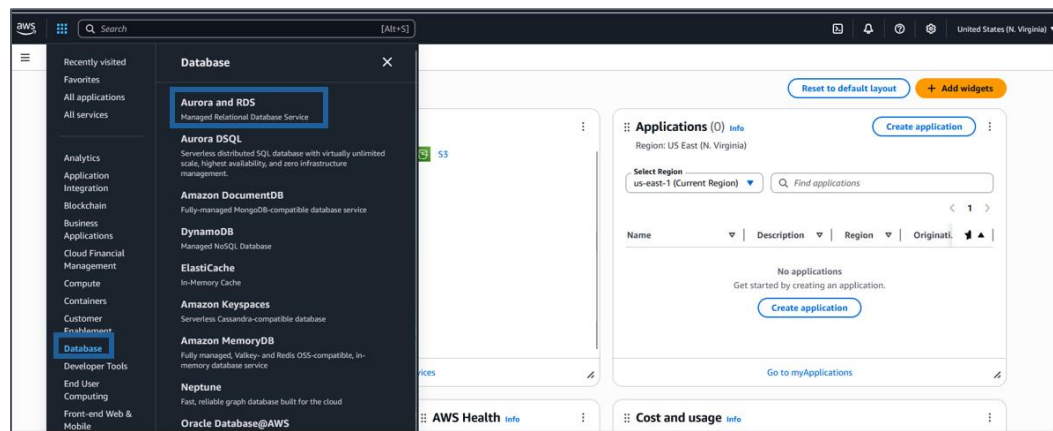


Implementation

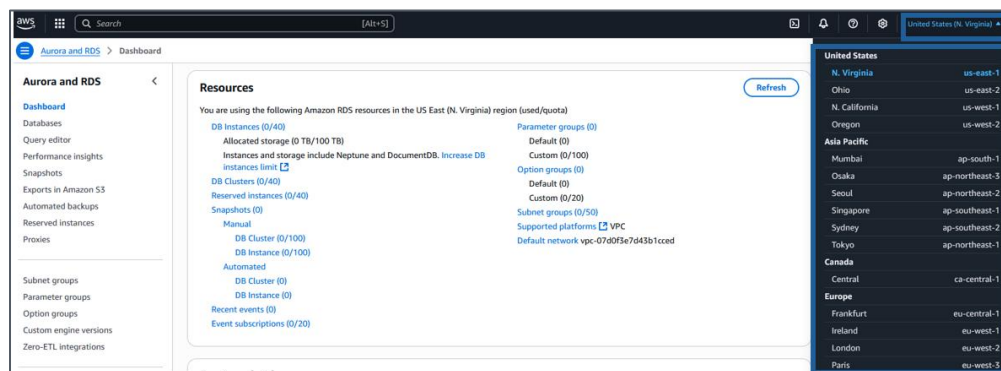
Create a MySQL DB Instance

In this step, we will use Amazon RDS to create a MySQL DB Instance with db.t3.micro DB instance class, 20 GB of storage, and automated backups disabled. All of these are Free Tier eligible.

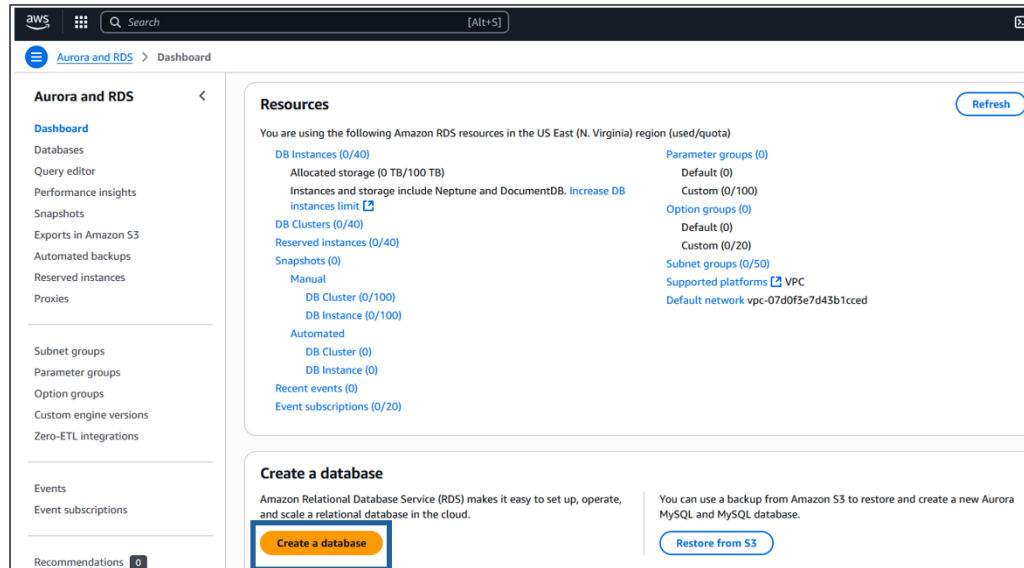
- Click the AWS icon as shown above to open the *AWS Management Console* in a new browser tab. When the console opens, click on **Services (square icon)** to select **Database** from the left navigation pane and choose **Aurora and RDS** to open the **Amazon Aurora and RDS console**.



- In the top right corner of the Amazon RDS console, select the Region in which you want to create the DB instance.



- In the **Create a database** section, choose **Create a database**.



- d. You now have options to select your engine. For this article, choose the MySQL icon, leave the default value of edition and engine version, and select the **Sandbox** template.

Create database [info](#)


Choose a database creation method


☒ **Standard create**
You set all of the configuration options, including ones for availability, security, backups, and maintenance.


☐ **Easy create**
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.


Engine options


Engine type [info](#)


☐ Aurora (MySQL Compatible) 


☐ Aurora (PostgreSQL Compatible) 


☒ **MySQL** 

☐ PostgreSQL 

☐ MariaDB 

☐ Oracle 

☐ Microsoft SQL Server 

☐ IBM Db2 

Edition

☒ **MySQL Community**

Engine version [info](#)

View the engine versions that support the following database features.

Hide filters

☐ Show only versions that support the Multi-AZ DB cluster [info](#)
Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

☐ Show only versions that support the Amazon RDS Optimized Writes [info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine version

MySQL 8.0.42

☐ **Enable RDS Extended Support** [info](#)
Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#).

Templates
Choose a sample template to meet your use case.

☐ **Production**
Use defaults for high availability and fast, consistent performance.

☐ **Dev/Test**
This instance is intended for development use outside of a production environment.

☒ **Sandbox**
To develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

Availability and durability
Deployment options Info
Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the [Amazon RDS service level agreement \(SLA\)](#).

☒ **Single-AZ DB instance deployment (1 instance)**
Creates a single DB instance without standby instances. This setup provides:
 • 99.95% uptime
 • No data redundancy

☐ **Multi-AZ DB instance deployment (2 instances)**
Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:
 • 99.95% uptime
 • Redundancy across Availability Zones

☐ **Multi-AZ DB cluster deployment (3 instances)**
Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
 • 99.95% uptime
 • Redundancy across Availability Zones
 • Increased read capacity
 • Reduced write latency

- e. You will now configure your DB instance. The list below shows the example settings you can use for this article:

Settings:

- **DB instance identifier:** Type a name for the DB instance that is unique for your account in the Region that you selected. For this article, we will name it **rds-mysql-academyDemo**.
- **Master username:** Type a username that you will use to log in to your DB instance. We will use **adminUsername** in this example.
- **Credentials management:** Choose Self managed option, then specify your
 - **Master password:** Type a password that contains from 8 to 41 printable ASCII characters (excluding /, ", and @) for your master user password.
 - **Confirm password:** Retype your password

Settings

DB instance identifier Info
rds-mysql-academyDemo

▼ Credentials Settings

Master username Info
adminUsername

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

☐ **Managed in AWS Secrets Manager - most secure**
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

☒ **Self managed**
Create your own password or have RDS create a password that you manage.

☐ **Auto generate password**
Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

Password strength Info
Strong
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / " * @

Confirm master password Info

Instance configuration
The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

▼ **Hide filters**

☐ Show instance classes that support Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

☐ Include previous generation classes

☐ Standard classes (includes m classes)

☐ Memory optimized classes (includes r and x classes)

☒ **Burstable classes (includes t classes)**

db.t3.micro
2 vCPUs 1 GiB RAM Network: Up to 2,085 Mbps

Instance specifications:

- **DB instance class:** Select **db.t3.micro** — 2 vCPU, 1 GiB RAM. This equates to 1 GB memory and 2 vCPU.
- **Storage type:** Select General Purpose SSD (gp2).
- **Allocated storage:** Select 20 to allocate 20 GB of storage for your database. Please check the LL service restrictions for Amazon RDS before you scale up.
- **Enable storage autoscaling:** If your workload is cyclical or unpredictable, you would enable storage autoscaling to enable Amazon RDS to automatically scale up your storage when needed. This option does not apply to this demo.
- **Multi-AZ deployment:** Using a Multi-AZ deployment will automatically provision and maintain a synchronous standby replica in a different Availability Zone. This feature is not available for **Sandbox** template.

Storage

Storage type [Info](#)

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage [Info](#)

20 GB

▼ **Additional storage configuration**

Storage autoscaling [Info](#)
Provides dynamic scaling support for your database's storage based on your application's needs.

☒ **Enable storage autoscaling**
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

- f. You are now in the **Connectivity** section where you can provide information that Amazon RDS needs to launch your MySQL DB instance. The list below shows settings for our demoDB instance.

Connectivity

- **Compute resource:** Choose **Don't connect to an EC2 compute resource**. You can manually set up a connection to a compute resource later.
- **Virtual Private Cloud (VPC):** Select **Default VPC**.
- **DB subnet group:** Choose the **default** subnet group.

- **Public accessibility:** Choose **Yes**. This will allocate an IP address for your database instance so that you can directly connect to the database from your own device.
- **VPC security group (firewall):** Select **Create new**. Then specify a name for your security group. This will create a security group that will allow connection from the IP address of the device that you are currently using to the database created.
- **Availability Zone:** Choose **No preference**.
- **RDS Proxy:** By using Amazon RDS Proxy, you can allow your applications to pool and share database connections to improve their ability to scale. Leave the **RDS Proxy** unchecked.

Additional connectivity configuration

- **Port:** Leave the default value of 3306.

The image shows two screenshots of the Amazon RDS console configuration pages. The top screenshot is the 'Connectivity' page, which includes sections for 'Compute resource' (with options to connect to an EC2 instance or not), 'Virtual private cloud (VPC)' (with a dropdown menu showing 'Default VPC'), 'DB subnet group' (with a dropdown menu showing 'default'), and 'Public access' (with a radio button selected for 'Yes'). The bottom screenshot is the 'VPC security group (firewall)' page, which includes a 'Choose existing' or 'Create new' option (with 'Create new' selected), a 'New VPC security group name' field (containing 'mysql-db-dm0jg'), an 'Availability Zone' dropdown (set to 'No preference'), an 'RDS Proxy' section (with 'Create an RDS Proxy' unchecked), a 'Certificate authority - optional' dropdown (set to 'rds-ca-rsa2048-g1 (default)'), and an 'Additional configuration' section with a 'Database port' field (set to '3306').

- g. Amazon RDS supports several ways to authenticate database users. Select **Password authentication** from the list of options if it is not selected by default.

The image shows the 'Database authentication' configuration page in the Amazon RDS console. It includes a section titled 'Database authentication options' with three radio button options: 'Password authentication' (selected), 'Password and IAM database authentication', and 'Password and Kerberos authentication'. Below the 'Password authentication' option, it states 'Authenticates using database passwords.' Below the 'Password and IAM database authentication' option, it states 'Authenticates using the database password and user credentials through AWS IAM users and roles.' Below the 'Password and Kerberos authentication' option, it states 'Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.'

Monitoring

- The **Database Insight - Standard** is selected here for you by default because of the Sandbox template.
- **Enhanced monitoring:** Leave **Enable enhanced monitoring** unchecked as this setting is not supported within the LL.

Monitoring info
Choose monitoring tools for this database. Database Insights provides a combined view of Performance Insights and Enhanced Monitoring for your fleet of databases. Database Insights pricing is separate from RDS monthly estimates. See [Amazon CloudWatch pricing](#).

☐ Database Insights - Advanced
 • Retains 15 months of performance history
 • Fleet-level monitoring
 • Integration with CloudWatch Application Signals

☒ Database Insights - Standard

▼ **Additional monitoring settings**
 Enhanced Monitoring, CloudWatch Logs and DevOps Guru

Enhanced Monitoring
☐ **Enable Enhanced monitoring**
 Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Log exports
 Select the log types to publish to Amazon CloudWatch Logs

☐ Audit log
☐ Error log
☐ General log
☐ iam-db-auth-error log
☐ Slow query log

In the **Additional configuration** section:

Database options

- **Database name:** Enter a database name that is 1 to 64 alphanumeric characters. If you do not provide a name, Amazon RDS will not automatically create a database on the DB instance you are creating.
- **DB parameter group:** Leave the default value.
- **Option group:** Leave the default value. Amazon RDS uses option groups to enable and configure additional features.
- **Encryption:** This option is not available in the LL. Therefore, uncheck the Enable encryption setting.

Backup

- **Enable automated backups:** For this article, we will uncheck this setting.

Maintenance

- **Auto minor version upgrade:** Select **Enable auto minor version upgrade** to receive automatic updates when they become available.
- **Maintenance Window:** Select **No preference**.

Deletion protection: Turn off **Enable deletion protection** for this article. When this option is enabled, you're prevented from accidentally deleting the database.

Choose **Create Database**.

Additional configuration
Database options, encryption turned off, backup turned off, backtrace turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options
Initial database name:
DB parameter group:
Option group:

Backup
☐ Enable automated backups
☐ Enable encryption
☐ Enable deletion protection

Maintenance
☒ Enable auto minor version upgrade
☐ Choose a window
☒ No preference

Enable deletion protection
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

Create database

h. Your DB instance is now being created.

Note: Depending on the DB instance class and storage allocated, it could take several minutes for the new DB instance to become available.

The new DB instance appears in the list of DB instances on the RDS console. The DB instance will have a status of **creating** until the DB instance is created and ready for use. When the state changes to **available**, you can connect to a database on the DB instance. Feel free to move on to the next step as you wait for the DB instance to become available.

DB Identifier	Status	Role	Engine	Region	Size	Recommendations	CPU	Current activity	Maintenance
rds-mysql-academydemo	Creating	Instance	MySQL Co...	-	db.t3.micro	-	-	-	none

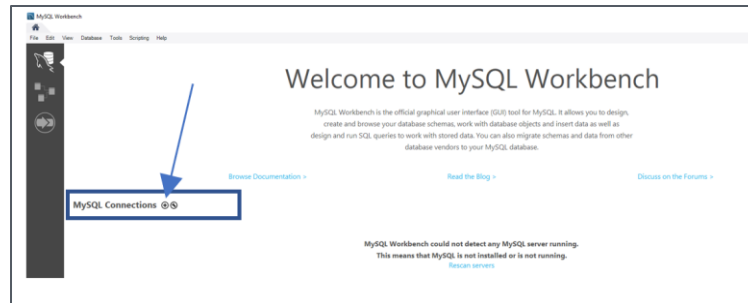
Connect to the Database using MySQL Workbench

Once the database instance creation is complete and the status changes to **available**, you can connect to the database on the RDS instance using any standard SQL client, such as MySQL

Workbench. If you do not have MySQL Workbench installed on your computer, please go to the [Download MySQL Workbench](#) page to download and install it.

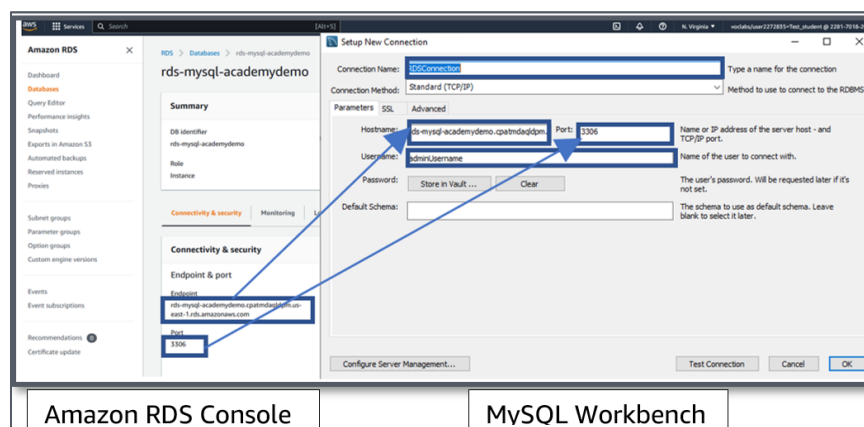
In this step, we will connect to the database you created using MySQL Workbench.

- a. Launch the MySQL Workbench application and go to **MySQL Connections > +** from the home screen.



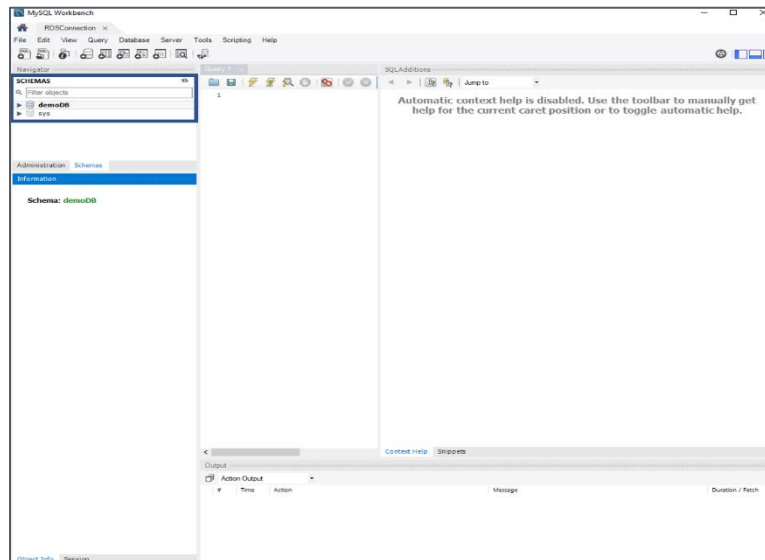
- b. A dialog box appears. Enter the following:
 - **Hostname:** You can find your hostname on the Amazon RDS console as shown in the screenshot.
 - **Port:** The default value should be 3306.
 - **Username:** Type in the username you created for the Amazon RDS database. In this article, it is 'adminUsername.'
 - **Password:** Choose **Store in Vault** (or Store in Keychain on MacOS) and enter the password that you used when creating the Amazon RDS database.
 - **Connection Name:** Please provide a suitable name for your connection (For example, RDSConnection). MacOS users might not need to specify a name for their connection.

Choose **OK**.

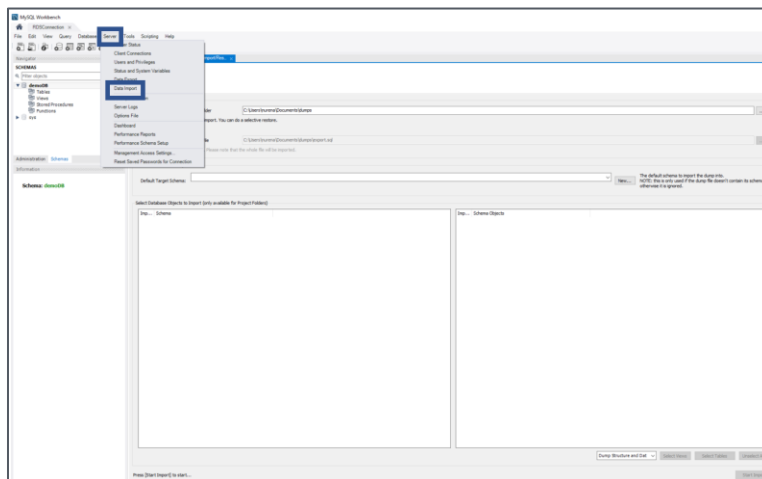


Create tables, insert data, and run queries

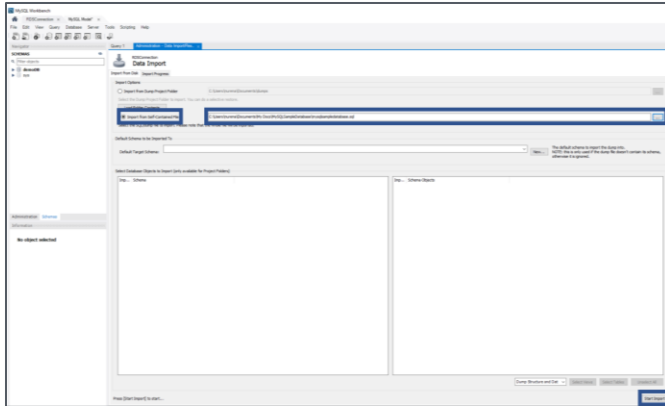
- a. Congratulations!!! Now that you are connected to your database, you will see various schema objects available in the database. Now you can create tables, insert data, and run queries.



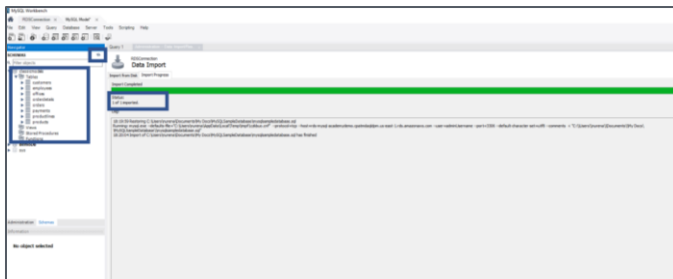
- b. Let's create a new database, create some tables, and insert some data into the tables. The **classicmodels** database used in this demo is available on the [MySQL Sample Database website](#). You can download the database and import it into your MySQL DB instance.
- c. To import the **classicmodels** script into your database, Go to **Server > Data Import** from the menu bar.



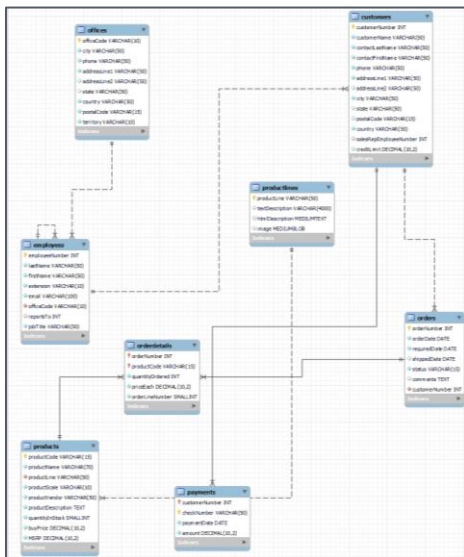
- d. A **Data Import** box appears. Select **Import from self-contained File**, then click the button in front of it to choose the **classicmodels** script you downloaded to your computer from step b. Click **Start Import** to import the script.



- e. Once the data has imported successfully, **refresh** the **SCHEMAS** and a new **schema object** (a database) will appear.



- f. If you like, you can use the MySQL Workbench **Reverse Engineer Database Wizard** to create the Entity Relationship diagram (ERD) of your database. Go to **Database** menu > **Reverse Engineer Database** > **Select your Stored Connection** (for example, RDSConnection) > **Click Next** > **Click Next** > **Check your database name** (classicmodels) > **Click Next** > **Click Next** > **Click Execute** > **Click Next** > **Click Finish**.



- g. It is time to run some queries against your database.

- **Query 1:** Write a query to get the first name, last name, and job title of all the employees:

Query 1

```
1 SELECT first_name, last_name, job_title
2 FROM employees;
```

first_name	last_name	job_title
Diane	Murphy	President
Mary	Patterson	VP Sales
Jeff	Finnell	VP Marketing
William	Patterson	Sales Manager (APAC)
Gerard	Bondur	Sales Manager (EMEA)
Anthony	Bow	Sales Manager (NA)
Leslie	Jernigan	Sales Rep
Leslie	Thompson	Sales Rep
Julie	Finnell	Sales Rep
Steve	Patterson	Sales Rep
Pamela	Stang	Sales Rep
George	Venau	Sales Rep
Lou	Bondur	Sales Rep
Gerard	Hernandez	Sales Rep
Pamela	Castillo	Sales Rep
Larry	Bott	Sales Rep
Barry	Jones	Sales Rep
Andy	Foster	Sales Rep
Peter	Marsh	Sales Rep
Tim	King	Sales Rep
Hans	Kato	Sales Rep
Yoshiko	Kato	Sales Rep
Martin	Gerard	Sales Rep

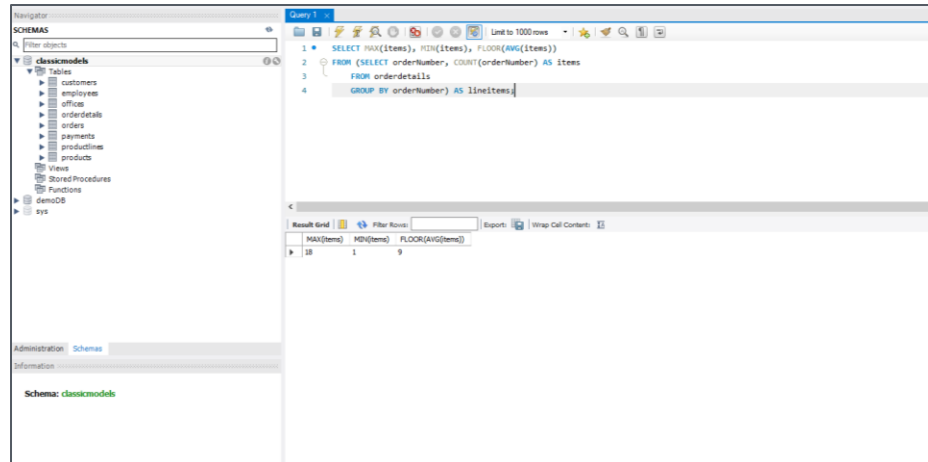
- **Query 2:** Write a query that returns the order number, order status, and total sales per order. Note that there is no column named totals sales in either orders and orderdetails tables.

Query 1

```
1 SELECT a.orderNumber, a.status, SUM(quantityOrdered * priceEach) total
2 FROM orders a
3 JOIN orderdetails b
4 ON a.orderNumber = b.orderNumber
5 GROUP BY orderNumber
6 ORDER BY total DESC; -- Just using this to sort result in descending order
```

orderNumber	status	total
10165	Shipped	67392.85
10287	Shipped	61402.00
10310	Shipped	61234.67
10212	Shipped	59830.55
10207	Shipped	59265.14
10127	Shipped	58941.35
10204	Shipped	58793.53
10126	Shipped	57131.92
10222	Shipped	56821.65
10142	Shipped	56052.56
10390	Shipped	55902.50
10312	Shipped	55639.66
10135	Shipped	55601.84
10192	Shipped	55425.77
10181	Shipped	55069.55
10159	Shipped	54682.68
10105	Shipped	53959.21
10314	Shipped	53745.34
10304	Shipped	53116.99
10306	Shipped	52825.29
10185	Shipped	52548.49
10419	Shipped	52420.07
10402	On Hold	52330.88

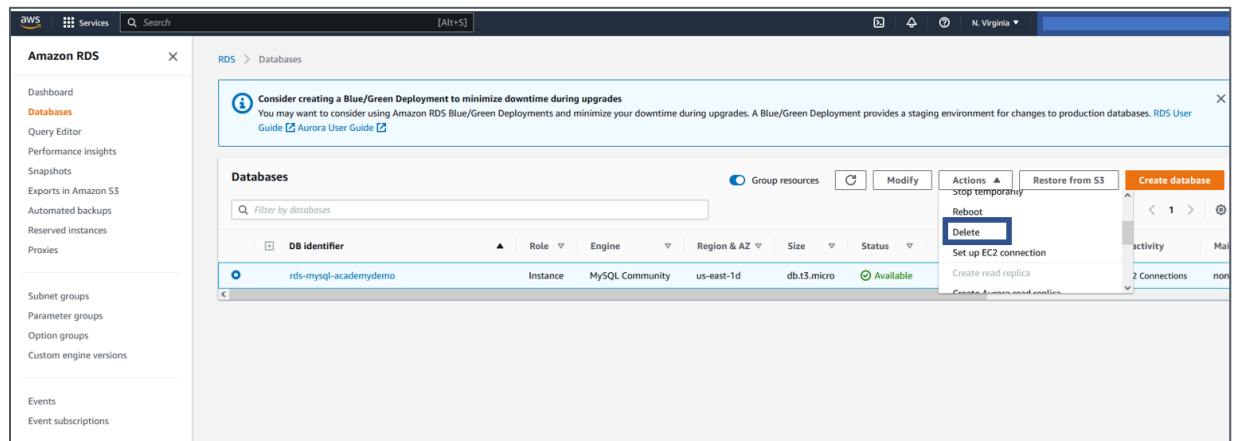
- **Query 3:** Write a query that returns the maximum, minimum, and average number of items in sales orders.



Delete the database instance

It is a best practice to delete instances that you are no longer using so that you don't keep getting charged for them.

- Go back to the Amazon RDS console. Select **Databases**, choose the instance that you want to delete, and then select **Delete** from the **Actions** dropdown menu.



- You are asked to create a final snapshot and to confirm the deletion. In this case, do not create a final snapshot, acknowledge that you want to delete the instance, and then choose Delete.

Note: Deleting your DB instance may take a few minutes.

Delete rds-mysql-academydemo instance?

×


Are you sure you want to Delete the **rds-mysql-academydemo** DB Instance?

☐ Create final snapshot?
Determines whether a final DB Snapshot is created before the DB instance is deleted.

☒ I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.

To confirm deletion, type *delete me* into the field

delete me

 We strongly recommend taking a final snapshot before instance deletion since after your instance is deleted, automated backups will no longer be available.

CancelDelete