

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

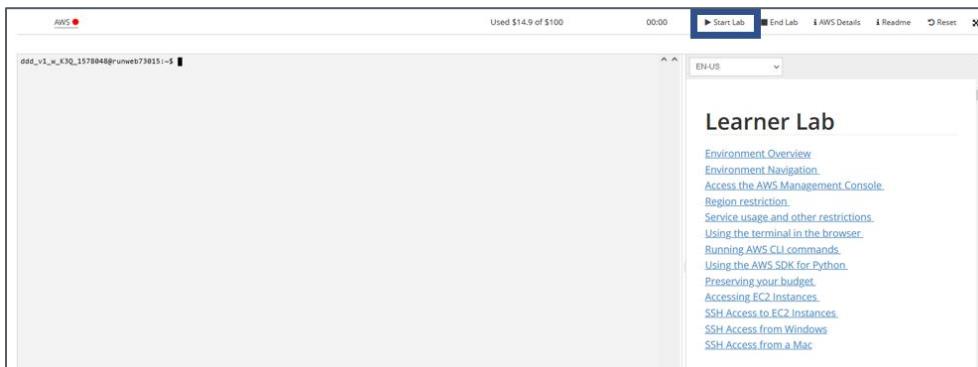
In this article, you will:

- 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

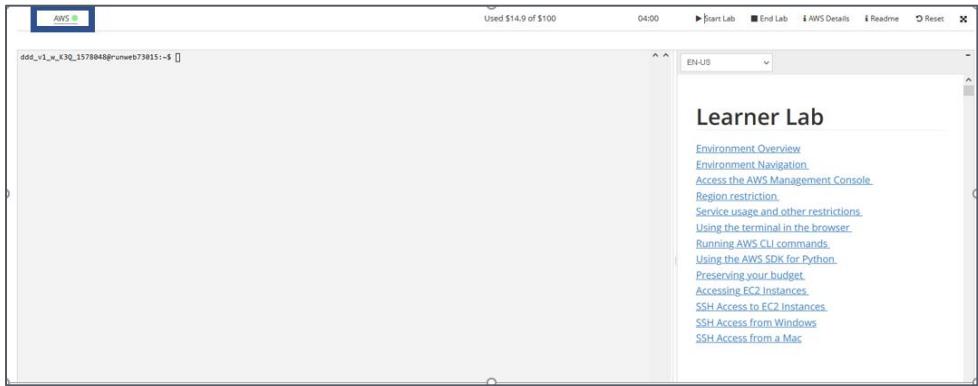
Prerequisites

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:

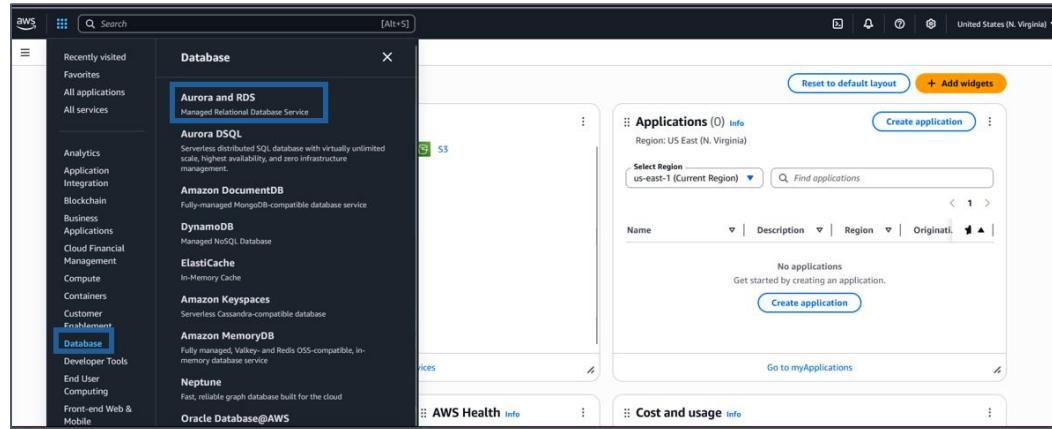


Implementation

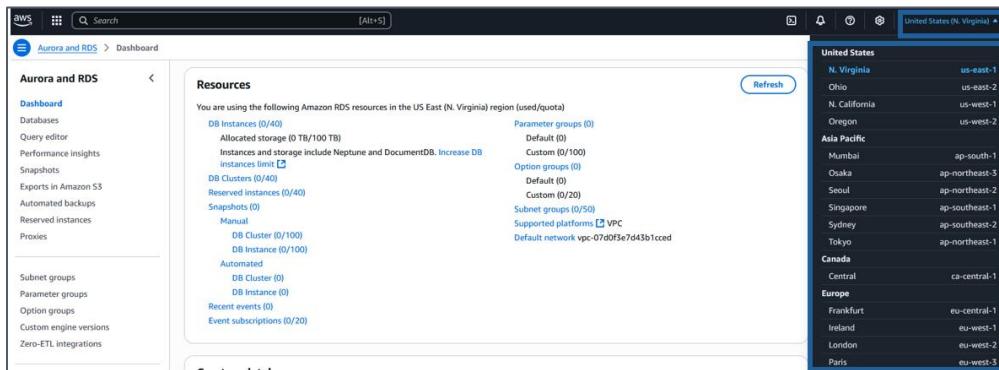
Create a MySQL DB Instance

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.

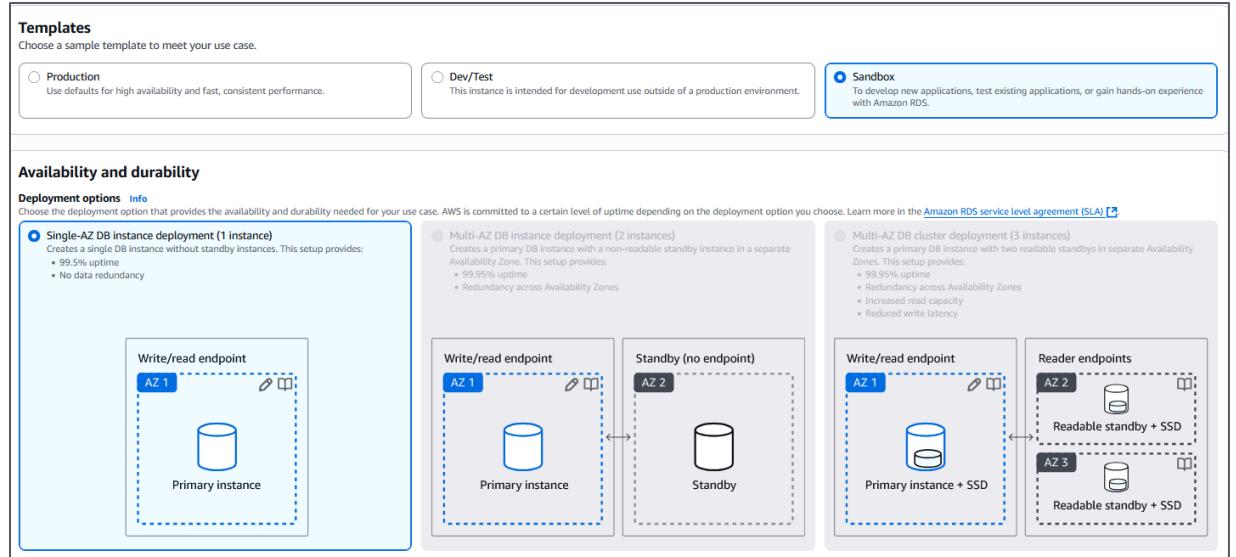


- In the **Create a database** section, choose **Create a database**.

The screenshot shows the Aurora and RDS Dashboard. On the left, there's a sidebar with links like Dashboard, Databases, Query editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, and Recommendations (0). The main area is titled 'Resources' and lists various Amazon RDS resources in the US East (N. Virginia) region. It includes sections for DB Instances (0/40), DB Clusters (0/40), Reserved instances (0/40), Snapshots (0), and Subnet groups (0/50). There are also links for Parameter groups (0), Option groups (0), and Default network vpc-07d0f3e7d43b1cced. A 'Create a database' section contains a 'Create a database' button and a note about using a backup from Amazon S3 to restore and create a new Aurora MySQL or MySQL database. A 'Restore from S3' button is also present.

- 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.

The screenshot shows the 'Create database' configuration page. It starts with a 'Choose a database creation method' section with 'Standard create' selected (highlighted in blue). Below it is an 'Easy create' section with a note about recommended best-practice configurations. The next section is 'Engine options' under 'Engine type'. It lists several database engines with icons: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL (selected), PostgreSQL, MariaDB, Oracle, Microsoft SQL Server, and IBM Db2. Under 'Edition', 'MySQL Community' is selected. In the 'Engine version' section, 'MySQL 8.0.42' is chosen. At the bottom, there are two filter options: 'Show only versions that support the Multi-AZ DB cluster' (selected) and 'Show only versions that support the Amazon RDS Optimized Writes' (selected). A note at the bottom states: '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.'



- 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, 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

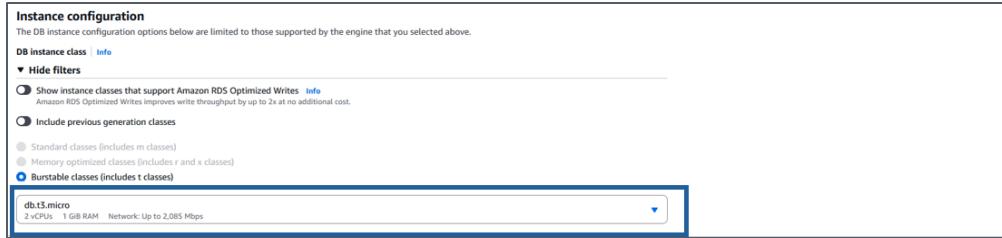
Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

- 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.

Master password [Info](#)

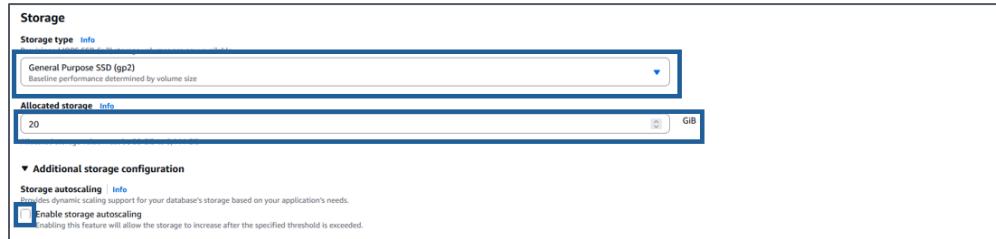
Password strength **Strong**
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / " @

Confirm master password [Info](#)



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.



- 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.

Compute resource

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC)

Default VPC (vpc-07d0f5e7d43b1cced)
6 Subnets, 6 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

DB subnet group

Choose the DB subnet group. The DB subnet group defines which subnet and IP range the DB instance uses in the VPC that you selected.

default

Public access

Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall)

Choose existing
Choose existing VPC security groups.

Create new
Create new VPC security group

New VPC security group name: mysql-db-demog

Availability Zone

No preference

RDS Proxy

RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

Create an RDS Proxy

Certificate authority - optional

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
Expiration: May 25, 2021

If you don't select a certificate authority, RDS chooses one for you.

Additional configuration

Database port

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.

Database authentication

Database authentication options

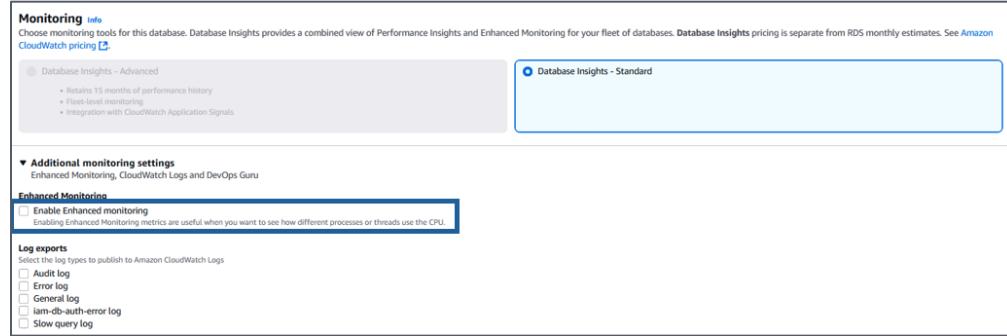
Password authentication
Authenticates using database passwords.

Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

Password and Kerberos authentication
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.



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**.

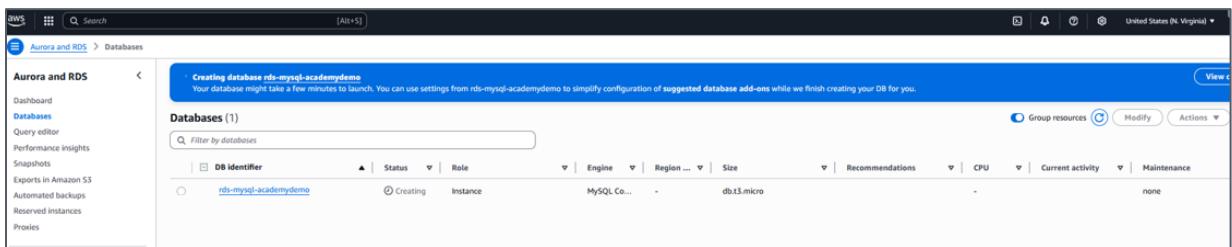
Deletion protection: Turn off **Enable deletion protection** for this article. When this option is enabled, you're prevented from accidentally deleting the database.

Choose **Create Database**.

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.



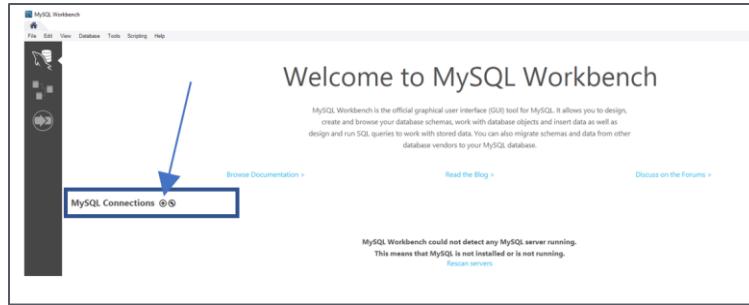
Connect to the Database using MySQL Workbench

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.

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.

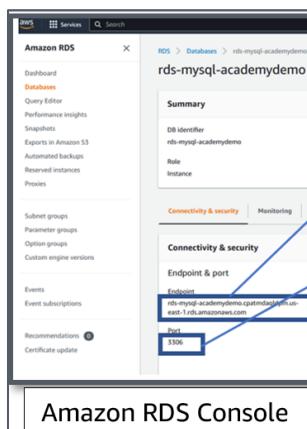
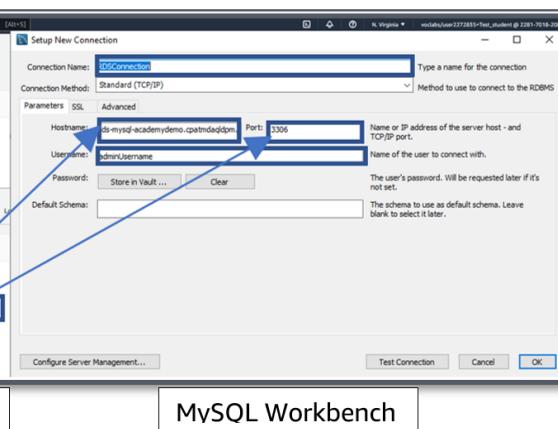
- Launch the MySQL Workbench application and go to **MySQL Connections > +** from the home screen.



- A dialog box appears. Enter the following:

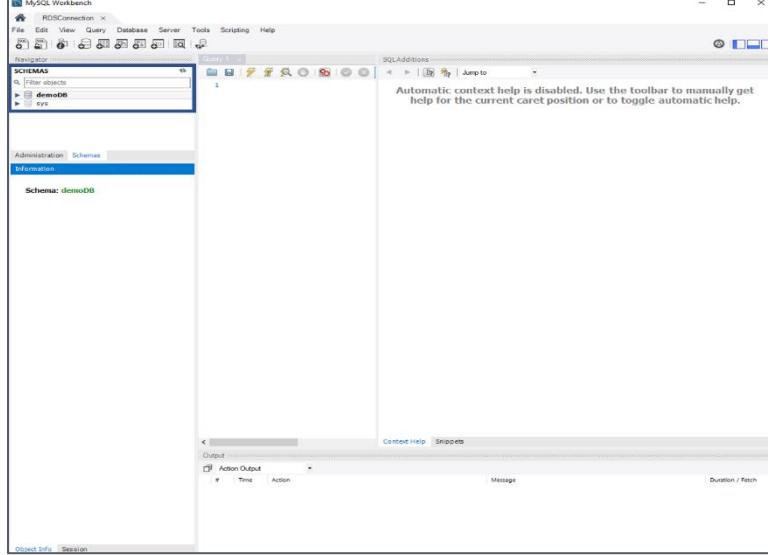
- Hostname:** You can find your hostname on the Amazon RDS console as shown in the screenshot.
- Port:** The default value should be 3306.
- Username:** Type in the username you created for the Amazon RDS database. In this article, it is 'adminUsername.'
- Password:** Choose **Store in Vault** (or Store 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, RDSCConnection). MacOS users might not need to specify a name for their connection.

Choose OK.

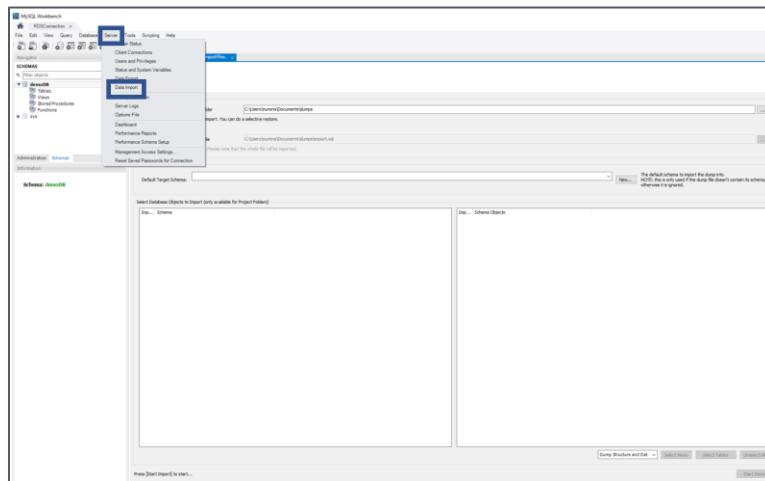
	
---	--

Create tables, insert data, and run queries

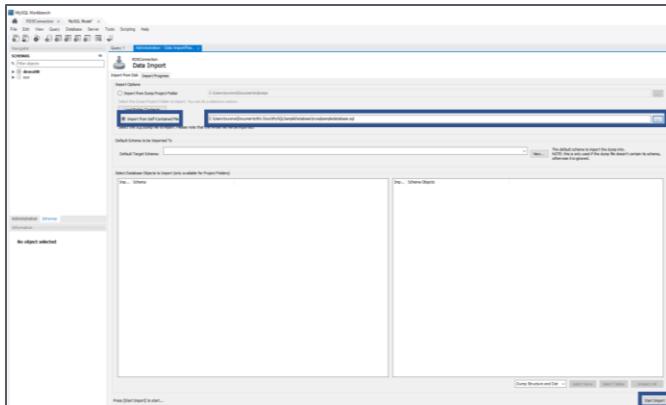
- a. Congratulations!!! Now that you are connected to your database, you will see various schema objects available in the database. Now you can create tables, insert data, and run queries.



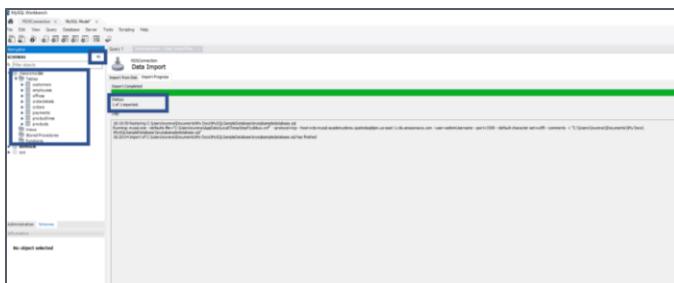
- b. Let's create a new database, create some tables, and insert some data into the tables. The **classicmodels** database used in this demo is available on the [MySQL Sample Database website](#). You can download the database and import it into your MySQL DB instance.
- c. To import the **classicmodels**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



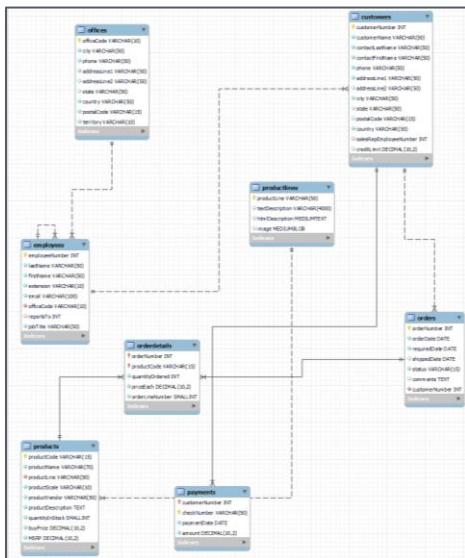
- d. A **Data Import** box appears. Select **Import from self-contained File**, then click the button in front of it to choose the **classicmodels** script you downloaded to your computer from step b. Click **Start Import** to import the script.



- 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:

The screenshot shows the SQL Server Management Studio interface. The left pane displays the Navigator and Schema browser for the 'classicmodels' database. The right pane contains a 'Query Grid' window with the following SQL code:

```
SELECT firstname, lastname, jobTitle
FROM employees;
```

The results grid shows the following data:

firstname	lastname	jobTitle
Janet	Landry	President
Mary	Patterson	VP Sales
Jeff	Pirelli	VP Marketing
William	Patterson	Sales Manager (APAC)
Gerard	Bondur	Sale Manager (EMEA)
Patricia	Brennan	Sale Manager (NA)
Leslie	Jennerig	Sales Rep
Steve	Thompson	Sales Rep
Julie	Pirelli	Sales Rep
Steven	Patterson	Sales Rep
George	Vernon	Sales Rep
Lou	Bondur	Sales Rep
Gerard	Hernandez	Sales Rep
Pamela	Castillo	Sales Rep
Beth	Brennan	Sales Rep
Berry	Jones	Sales Rep
Andy	Pivter	Sales Rep
Peter	Marsh	Sales Rep
Tim	King	Sales Rep
James	Nalli	Sales Rep
Yoshimi	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.

The screenshot shows the SQL Server Management Studio interface. The left pane displays the Navigator and Schema browser for the 'classicmodels' database. The right pane contains a 'Query Grid' window with the following SQL code:

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

The results grid shows the following data:

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	58841.35
10204	Shipped	58793.53
10126	Shipped	57131.92
10222	Shipped	56822.65
10142	Shipped	56652.56
10390	Shipped	55902.50
10312	Shipped	55639.66
10135	Shipped	55601.64
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
10407	On Hold	52329.55

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

The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays the schema structure under 'classicmodels'. The central area contains a 'Query 1' window with the following SQL code:

```

1 • SELECT MAX(items), MIN(items), FLOOR(AVG(items))
2 FROM (SELECT orderNumber, COUNT(orderNumber) AS items
3       FROM orderdetails
4     GROUP BY orderNumber) AS lineitems;

```

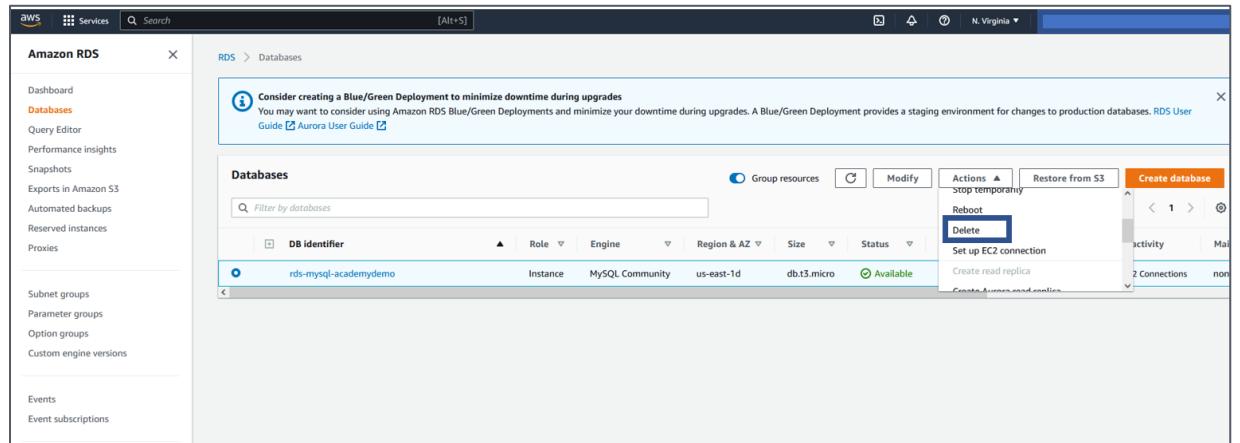
Below the query window is a 'Result Grid' showing the output of the query:

	MAX(items)	MIN(items)	FLOOR(AVG(items))
18	1	9	

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.

