Insert, Update, and Delete Data in a Database

Scenario

The database operations team has created a relational database called **world** containing three tables: **city**, **country**, and **countrylanguage**. You have to validate the configuration of the database by running **INSERT**, **UPDATE**, and **DELETE** statements on the **country** table.

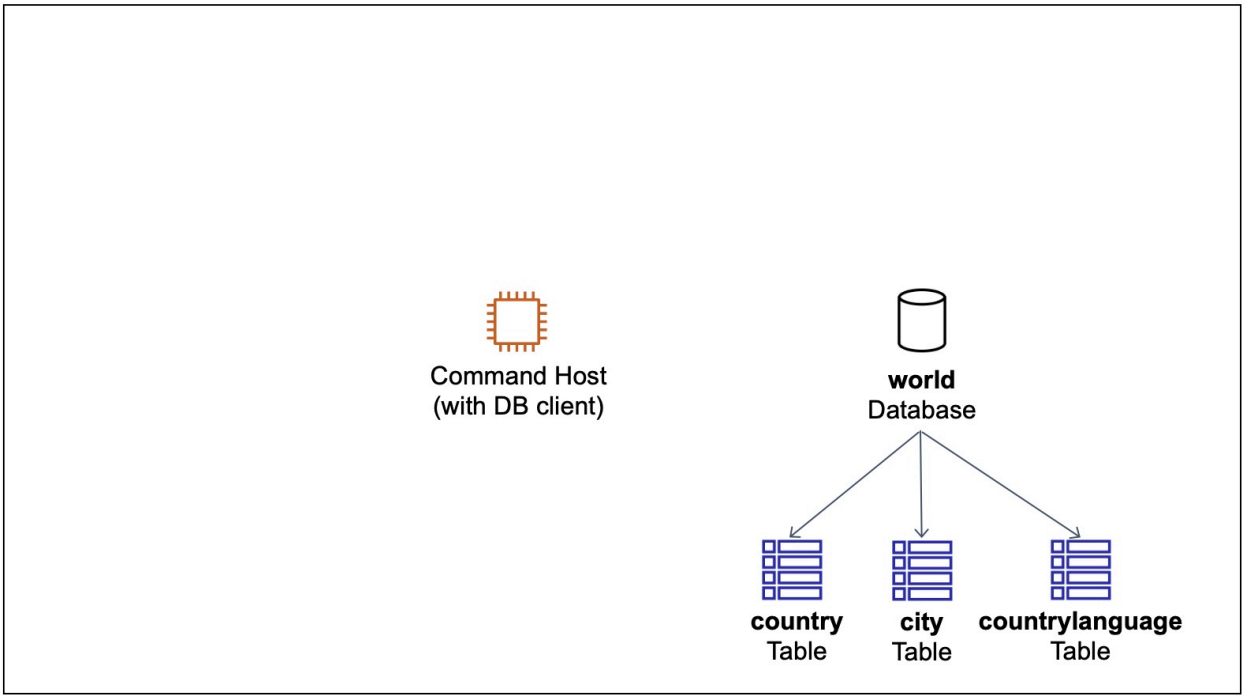
Lab overview and objectives

This lab demonstrates how to insert, update, delete, and import rows of data using structured query language (SQL).

After completing this lab, you will be able to:

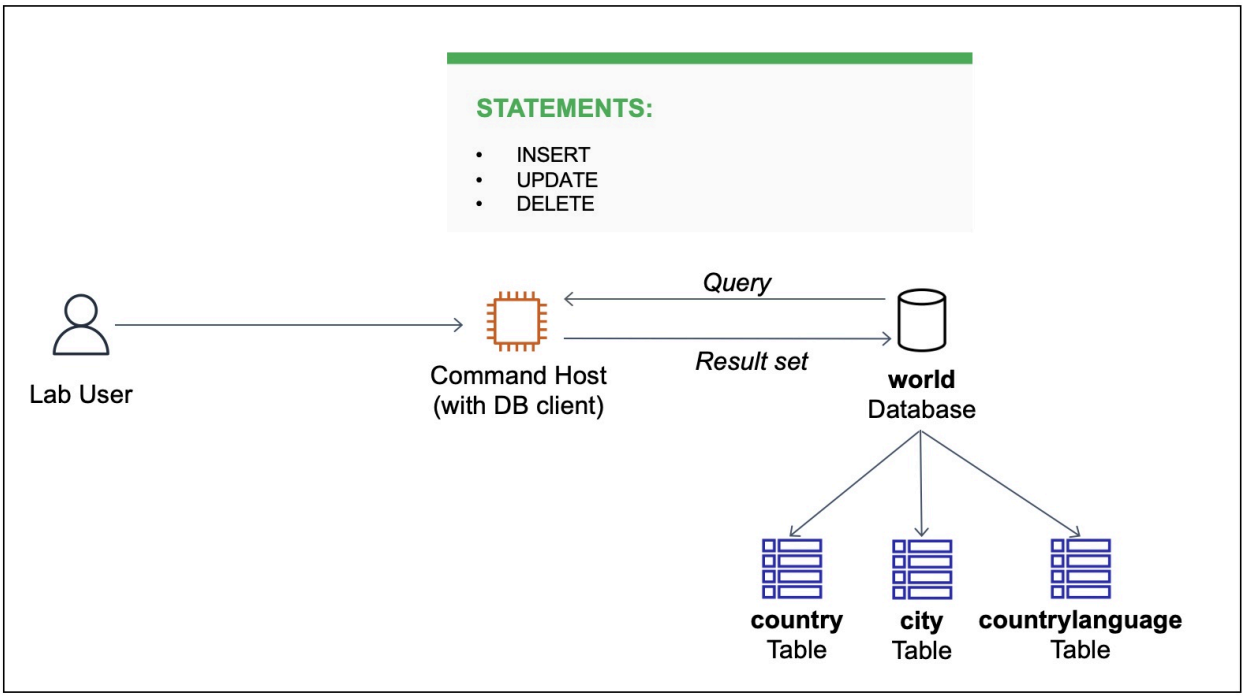
- Insert rows into a table
- Update rows in a table
- Delete rows from a table
- Import rows from a database backup file

When you start this lab, the following resources are already created for you:



A Command Host instance and world database containing three tables

At the end of this lab, your architecture will look like the following example:



Task 1: Connect to a database

In this task, you connect to an instance containing a database client, which is used to connect to a database. This instance is referred to as the Command Host.

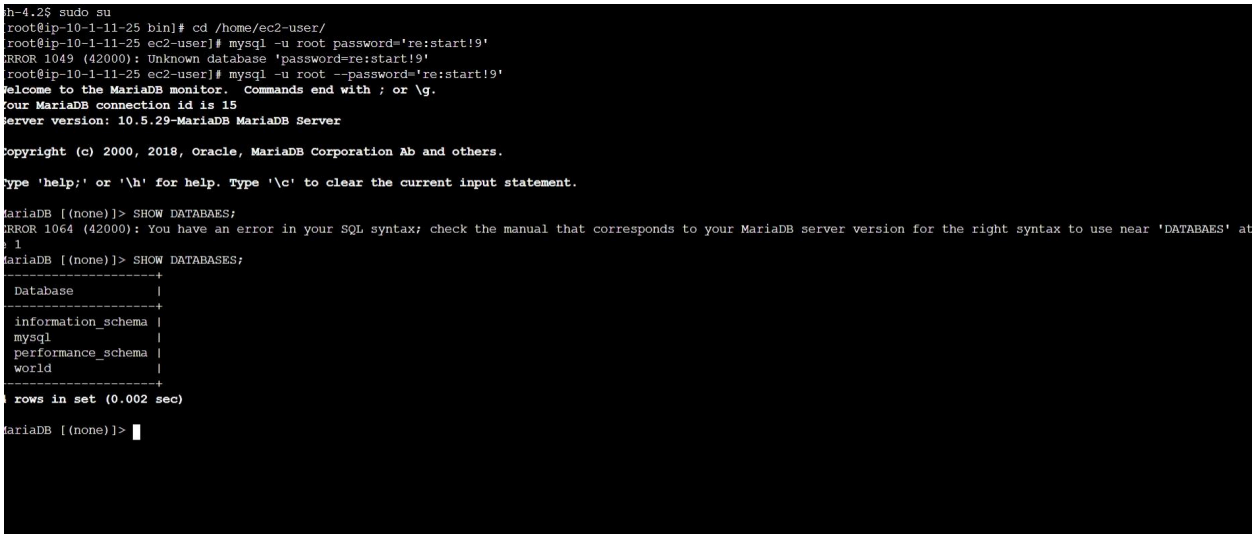
5. In the AWS Management Console, choose the **Services** menu. Under **Compute**, choose **EC2**.
6. In the left navigation pane, choose **Instances**.
7. Next to the instance labelled **Command Host**, select the check box and then choose **Connect**.
   Note: If you do not see the **Command Host**, the lab is possibly still being provisioned, or you may be using another Region.
8. For **Connect to instance**, choose the **Session Manager** tab.
9. Choose **Connect** to open a terminal window.
   Note: If the **Connect** button is not available, wait for a few minutes and try again.
10. To configure the terminal to access all required tools and resources, run the following command:
11. `sudo su`
    `cd /home/ec2-user/`
12. **Tips**:
    - Copy and paste the command into the Session Manager terminal window.

- If you are using a Windows system, press Shift+Ctrl+v to paste the command.

13. To connect to the database instance, run the following command in the terminal. A password was configured when the database was installed.
14. `mysql -u root --password='re:St@rt!9'`
15. The MySQL command-line client is a SQL shell that you can use to interact with database engines.

Switch	Description
-u or --user	The MySQL user name used to connect to a database instance
-p or --password	The MySQL password used to connect to a database instance

16. **Tip**: At any stage of the lab, if the Sessions Manager window is not responsive or if you need to reconnect to the database instance, then follow these steps:
    - Close the Sessions Manager window, and try to reconnect using the previous steps.
    - Run the following commands in the terminal.
17.
```
sudo su
cd /home/ec2-user/
mysql -u root --password='re:St@rt!9'
```
18. To show the existing databases, enter the following command in the terminal. Make a note of the currently available databases.
19. `SHOW DATABASES;`
20.



Task 2: Insert data into a table

In this task, you insert sample data into the **country** table.

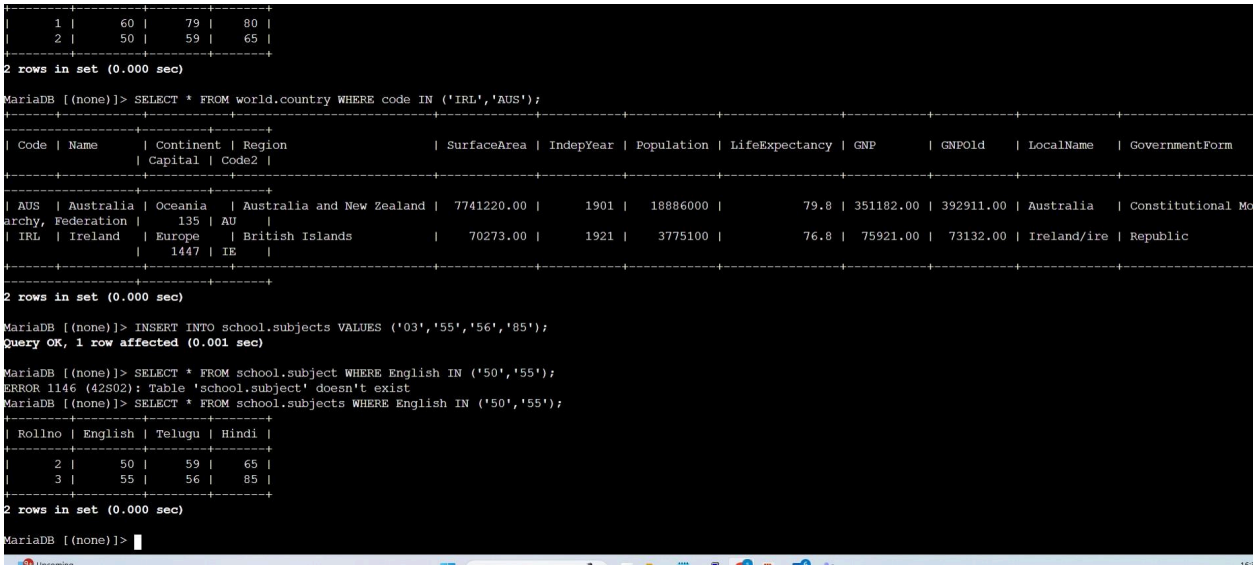
13. To verify that the **country** table exists, run the following command. The **SELECT** statement is used to identify the columns that should be included in the result set. The use of the * denotes all columns. The **FROM** clause is used in the following example to specify the database and table that is queried.
14. `SELECT * FROM world.country;`
15. To insert rows into the **country** table, run the following commands. The values in the **VALUES** clause need to be in the same order as defined by the table schema.

```
INSERT INTO world.country VALUES ('IRL','Ireland','Europe','British Islands',70273.00,1921,3775100,76.8,75921.00,73132.00,'Ireland/Éire','Republic',1447,'IE');
```

16. `INSERT INTO world.country VALUES ('AUS','Australia','Oceania','Australia and New Zealand',7741220.00,1901,18886000,79.8,351182.00,392911.00,'Australia','Constitutional Monarchy, Federation',135,'AU');`
17. To verify that two rows were successfully inserted into the **country** table, run the following query.
18. `SELECT * FROM world.country WHERE Code IN ('IRL', 'AUS');`

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOld	LocalName	GovernmentForm	Capital	Code2
AUS	Australia	Oceania	Australia and New Zealand	7741220	1901	18886000	79.8	351182	392911	Australia	Constitutional Monarchy, Federation	135	AU
IRL	Ireland	Europe	British Isla	70273	1921	3775100	76.8	75921	73132	Ireland/ Éire	Republic	1447	IE

			n d s										

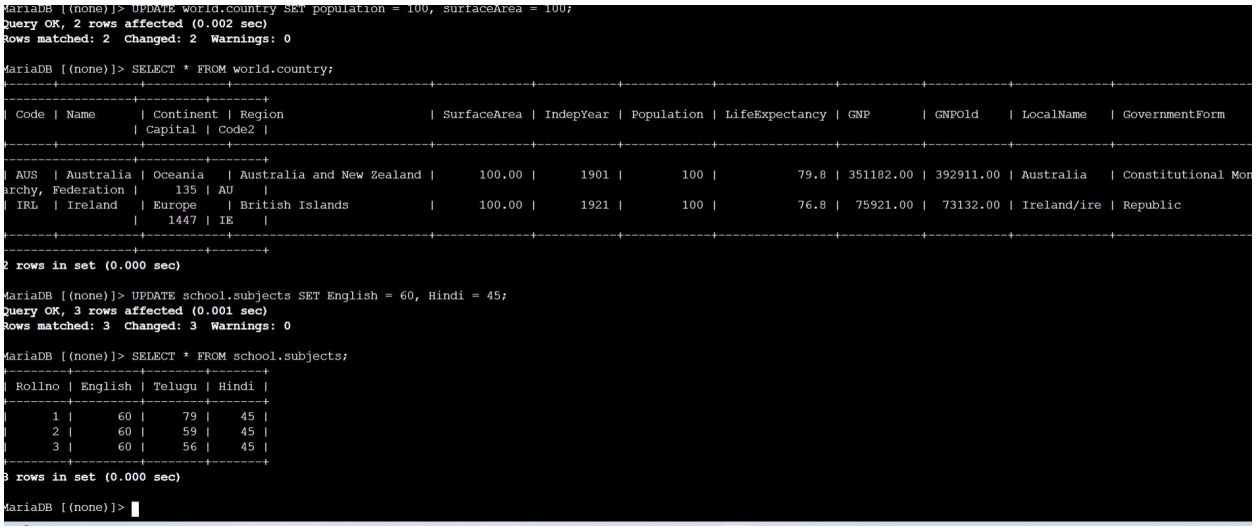


Task 3: Update rows in a table

In this task, you update both rows in the **country** table using an **UPDATE** statement.

16. To set the value in the **Population** column to 0 for both rows in the **country** table, run the following **UPDATE** statement.
17. `UPDATE world.country SET Population = 0;`
18. All rows are updated because the **UPDATE** statement does not include a **WHERE** condition. A **WHERE** clause uses conditions to filter rows returned by a query. The next lab introduces the **WHERE** clause.
19. To verify that the **Population** column in the **country** table was updated, run the following command.
20. `SELECT * FROM world.country;`
21. To update the **Population** and **SurfaceArea** columns for all rows in the **country** table, run the following **UPDATE** statement.
22. `UPDATE world.country SET Population = 100, SurfaceArea = 100;`

23. To verify that the **Population** and **SurfaceArea** columns in the **country** table were updated, run the following command.
24. `SELECT * FROM world.country;`

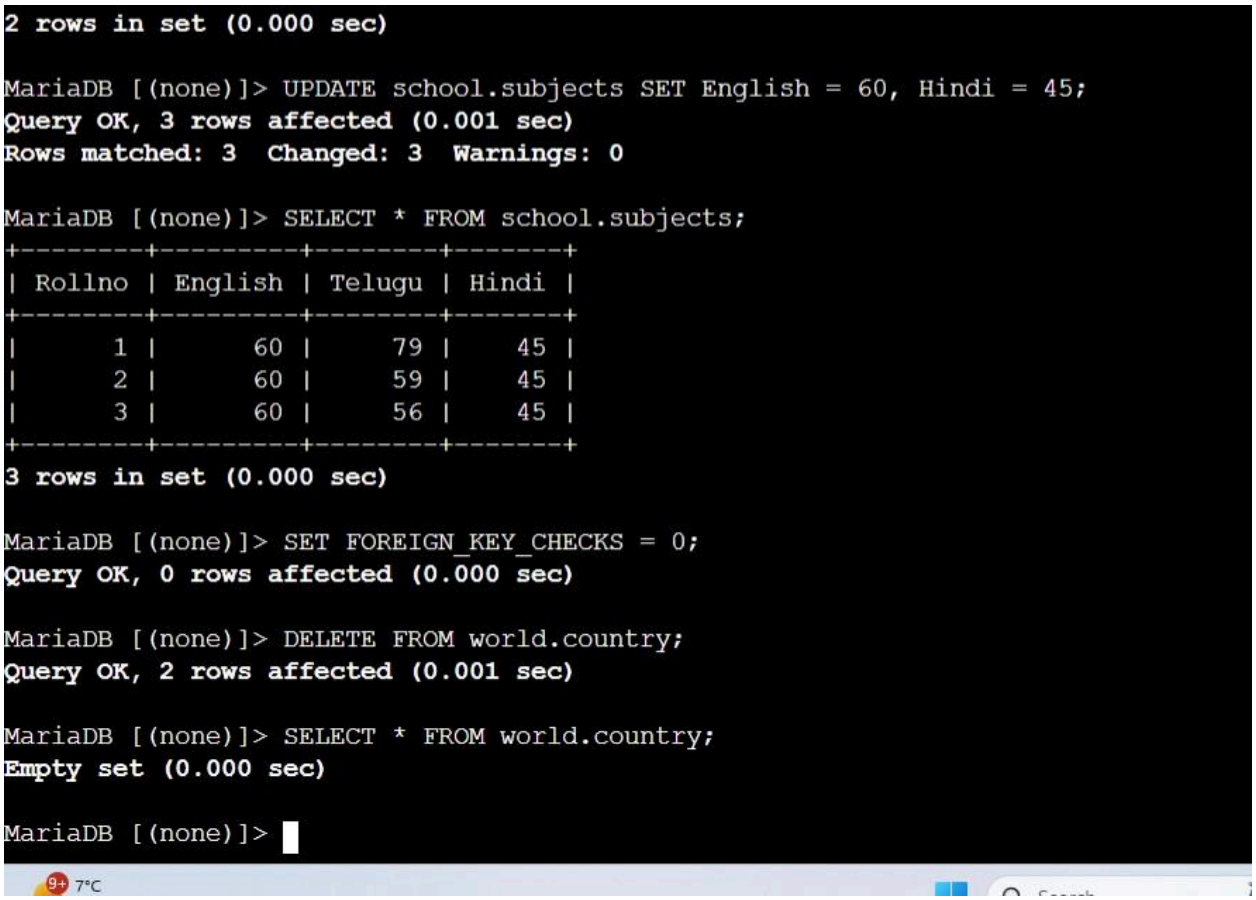


Task 4: Delete rows from a table

In this task, you delete rows in the **country** table using a **DELETE** statement.

Exercise caution when using data manipulation statements such as **UPDATE** and **DELETE** because these changes may not be reversible.

20. To delete **ALL** rows from the **country** table, run the following command.
21.
```
SET FOREIGN_KEY_CHECKS = 0;
DELETE FROM world.country;
```
22. Because the **DELETE** statement does not include a **WHERE** condition, all rows are deleted.
23. To verify that all rows have been deleted from the **country** table, run the following command.
24. `SELECT * FROM world.country;`



Task 5: Import data using an SQL file

In this task, you import sample data into the **country** table using an SQL file.

22. To exit the MySQL terminal, run the following command.
23. `QUIT;`
24. To verify that the **world.sql** file has been downloaded, run the following command.
25. `ls /home/ec2-user/world.sql`

Recall Linux commands

Use the `ls` (list) command to list the contents of a directory.

26.
27. It is time-consuming to insert individual rows into a table. You can create a SQL script file containing a group of SQL statements to quickly load data into a database. To load rows into the **country** table, run the following command.
28. `mysql -u root --password='re:St@rt!9' < /home/ec2-user/world.sql`
29. This database file adds two additional tables and inserts data into all three tables.
30. To reconnect to the database, run the following command.
31. `mysql -u root --password='re:St@rt!9'`
32. To verify that the script ran successfully, run the following command.
33. `USE world;`
    `SHOW TABLES;`
34. Observe that there are three tables named **city**, **country**, and **countrylanguage**.
35. To verify that the rows were loaded successfully, run the following command.
36. `SELECT * FROM country;`
37. Notice that there are more entries in the **country** table.
28. Similarly, use the **SELECT** statement to query the **city** and **countrylanguage** tables that were created when you imported the backup file.