Database Table Operations

Scenario

The database operations team for an organization has configured a relational database instance. The team has asked you to practice creating and dropping (deleting) databases and tables.

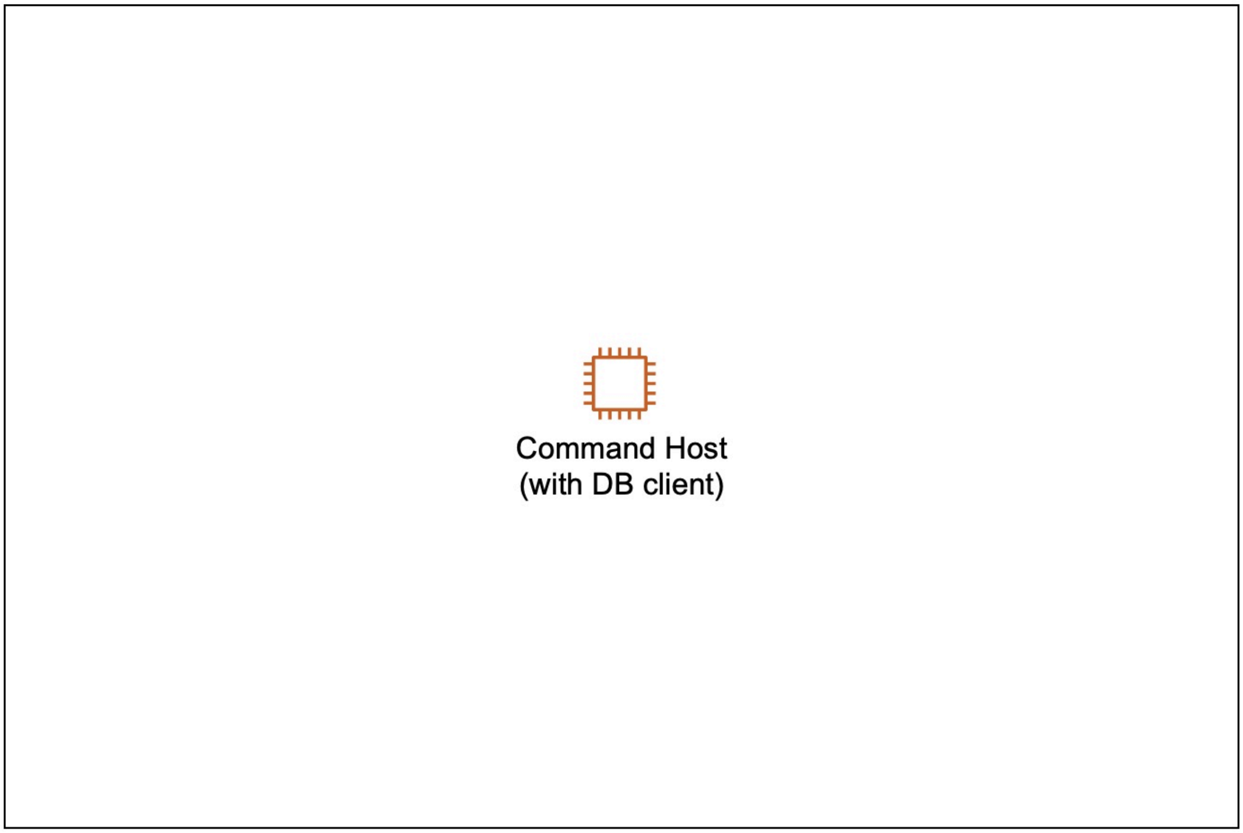
Lab overview and objectives

This lab demonstrates how to use some common database and table operations.

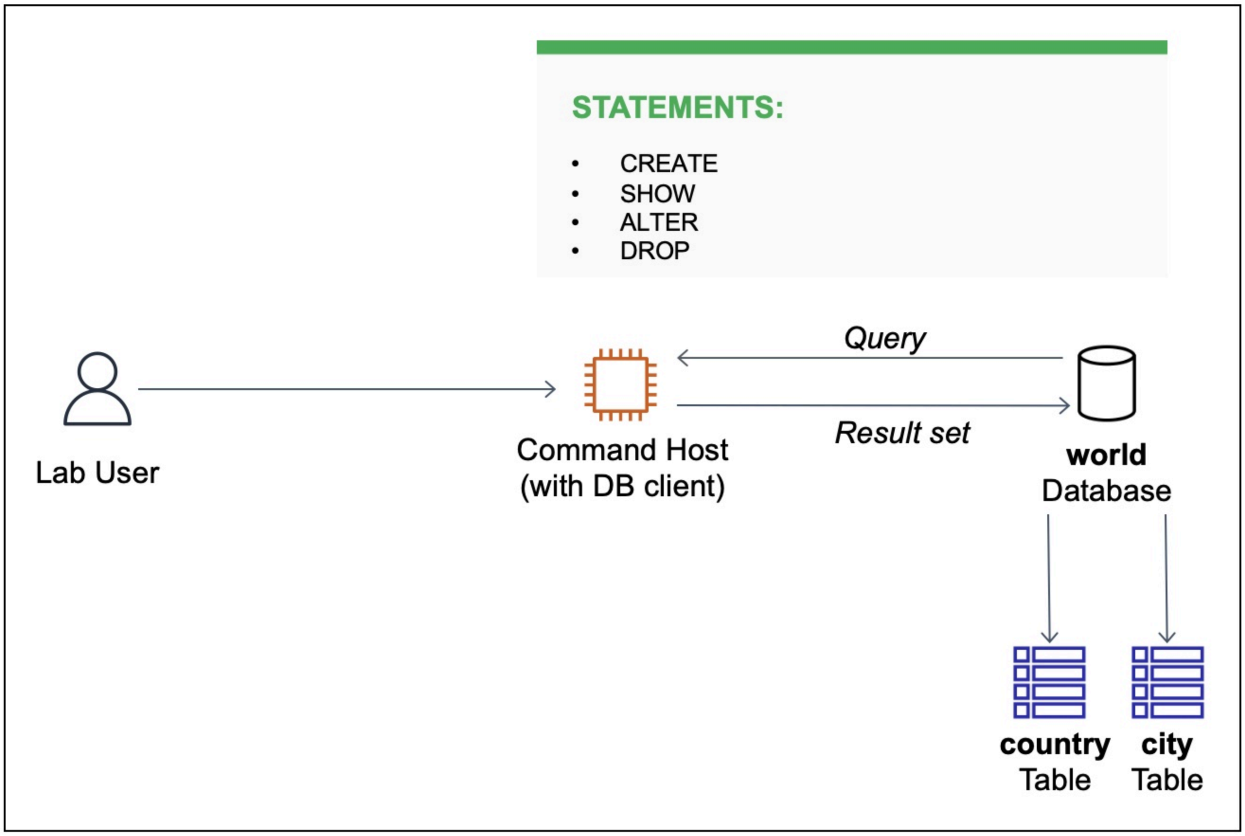
After completing this lab, you should be able to:

- Use the **CREATE** statement to create databases and tables
- Use the **SHOW** statement to view available databases and tables
- Use the **ALTER** statement to alter the structure of a table
- Use the **DROP** statement to delete databases and tables

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



Command Host
(with DB client)



Task 1: Connect to the Command Host

In this task, you connect to an EC2 instance configured with a database client. The client is used to run structured query language (SQL) queries against a relational database. This instance is referred to as the Command Host.

5. In the AWS Management Console, choose the **Services** menu. Choose **Compute**, and then choose **EC2**.
6. In the left navigation menu, 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 probably 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/
    ```

Recall Linux commands

The `sudo` (SuperUser DO) command is used to run a Linux command with the privileges of another user.

The `su` (switch user) command is used to switch to a different user. By default, it uses the root user if no user is specified.

The `cd` (change directory) command is used to change from the current directory to a new path.

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 relational 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'
    ```

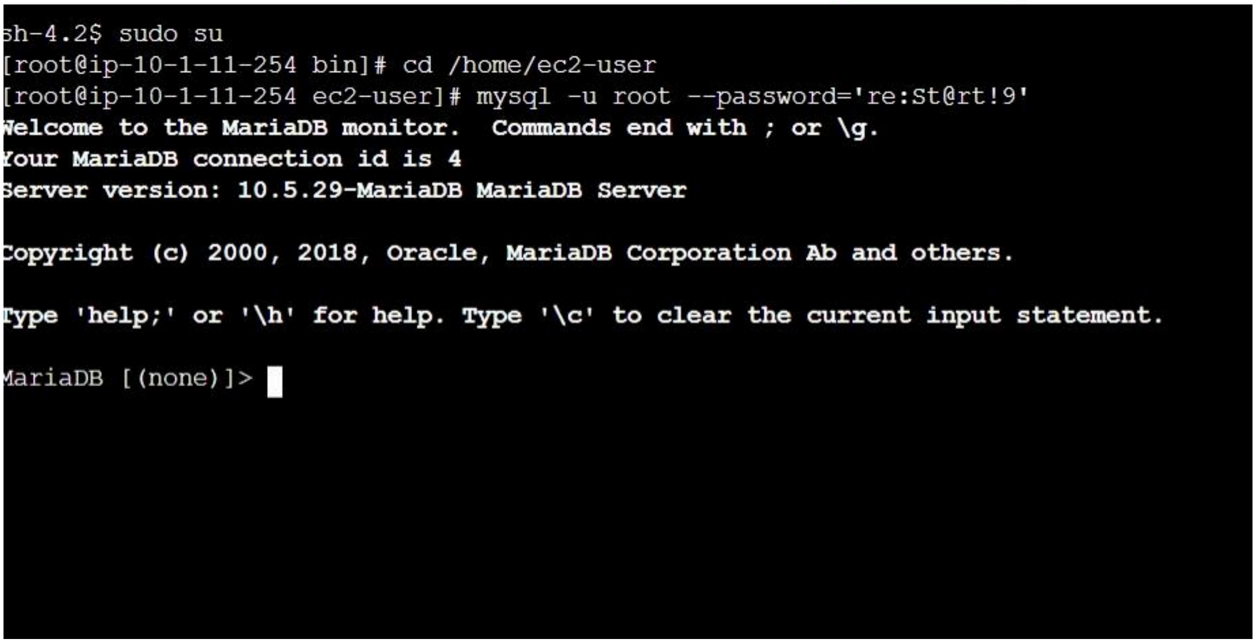
15. The MySQL command-line client is an SQL shell that you can use to interact with database engines.

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 Session Manager window is not responsive or if you need to reconnect to the database instance, then follow these steps:
    - Close the Session Manager window, and try to reconnect using the previous steps.
    - Run the following commands in the terminal.

    -
      ```
      sudo su
      cd /home/ec2-user/
      mysql -u root --password='re:St@rt!9'
      ```



```
sh-4.2$ sudo su
[root@ip-10-1-11-254 bin]# cd /home/ec2-user
[root@ip-10-1-11-254 ec2-user]# mysql -u root --password='re:St@rt!9'
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.5.29-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

Task 2: Create a database and a table

In this task, you create a database named **world** and a table named **country**. You then alter the **country** table.

12. To show the existing databases, run the following query.
13. `SHOW DATABASES;`
14. To determine the available database and to ensure that you are working with the correct database instance, use the **SHOW DATABASES;** command.
13. To create a new database named **world**, run the following command.
14. `CREATE DATABASE world;`
15. To verify that the **world** database has been created, run the following query.
16. `SHOW DATABASES;`
17. To store data in a database, the database needs to contain one or more tables. In an SQL database, a table needs a well-defined structure, known as a table schema. To create a table named **country**, run the following command.
18.
```
CREATE TABLE world.country (
  `Code` CHAR(3) NOT NULL DEFAULT '',
  `Name` CHAR(52) NOT NULL DEFAULT '',
  `Conitinent` enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South  America') NOT NULL DEFAULT 'Asia',
  `Region` CHAR(26) NOT NULL DEFAULT '',
  `SurfaceArea` FLOAT(10,2) NOT NULL DEFAULT '0.00',
  `IndepYear` SMALLINT(6) DEFAULT NULL,
  `Population` INT(11) NOT NULL DEFAULT '0',
  `LifeExpectancy` FLOAT(3,1) DEFAULT NULL,
  `GNP` FLOAT(10,2) DEFAULT NULL,
  `GNPOld` FLOAT(10,2) DEFAULT NULL,
  `LocalName` CHAR(45) NOT NULL DEFAULT '',
  `GovernmentForm` CHAR(45) NOT NULL DEFAULT '',
  `HeadOfState` CHAR(60) DEFAULT NULL,
  `Capital` INT(11) DEFAULT NULL,
  `Code2` CHAR(2) NOT NULL DEFAULT '',
  PRIMARY KEY (`Code`)
);
```

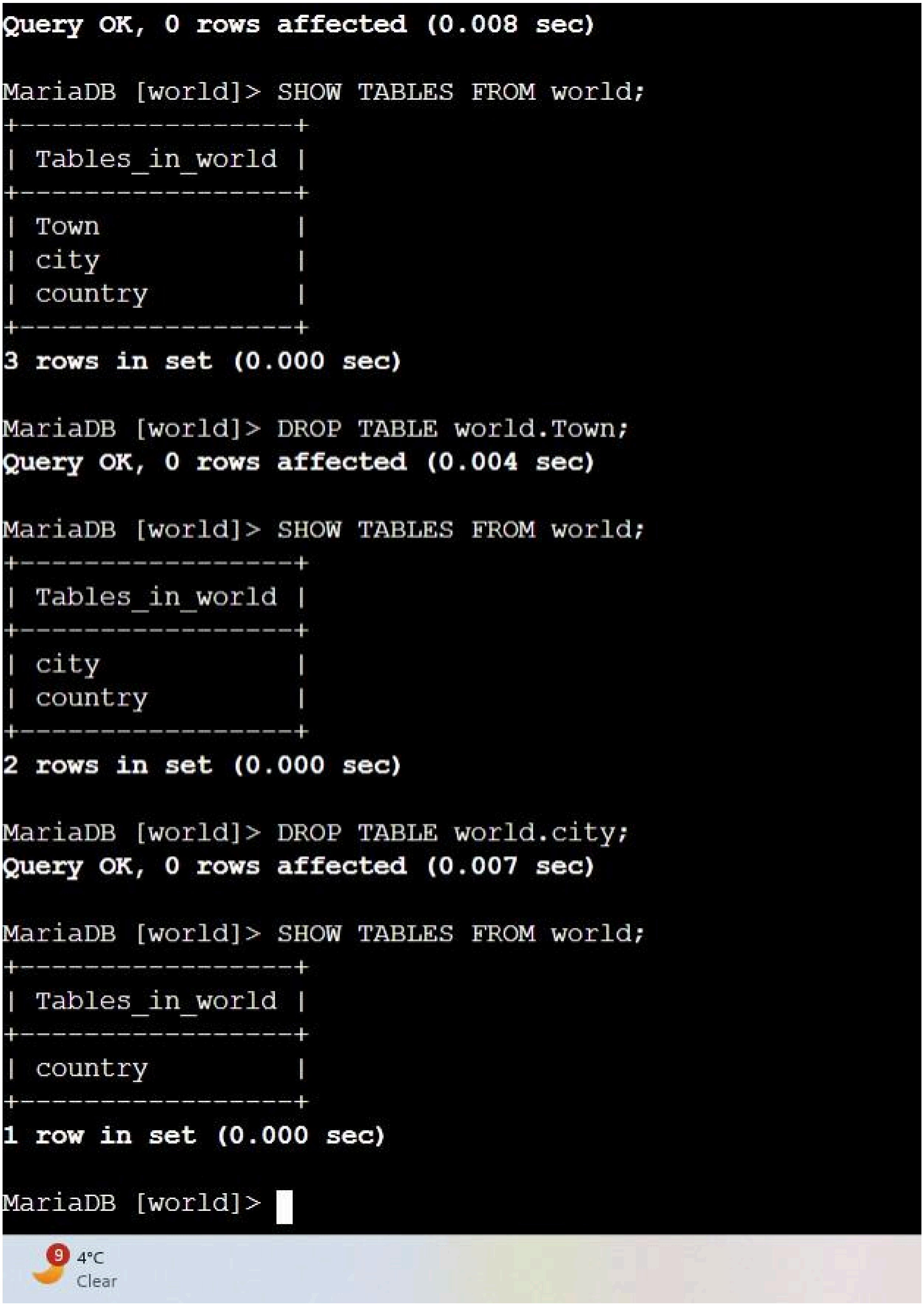
19. To verify that the **country** table was created, use the **SHOW TABLES;** command to list the tables in the database. You use the **USE** command to specify which database to run a query against. Run the following commands in your terminal.
20.
    ```
    USE world;
    SHOW TABLES;
    ```
21. Use the **SHOW COLUMNS** query to list all the columns on a table. Run the following query to list all columns and their properties in the **country** table.
22.
    ```
    SHOW COLUMNS FROM world.country;
    ```
23. **Note**: Notice that the **Continent** column is spelled incorrectly as **Conitinent**.
24. The **ALTER TABLE** command is used to alter the table's schema. To fix the incorrectly spelled **Continent** column, run the following command.
25.
    ```
    ALTER TABLE world.country RENAME COLUMN Conitinent TO Continent;
    ```
26. To verify that the **Continent** column name in the **country** table has been corrected, run the following query.
27.
    ```
    SHOW COLUMNS FROM world.country;
    ```
28.

Challenge 1

Create a table named **city** and add two columns named **Name** and **Region**. Both columns should use the **CHAR** data type.

```
CREATE TABLE world.city (`Name` CHAR(52), `Region` CHAR(26));
```

Tip: Expand the question to reveal the solution.



```
Query OK, 0 rows affected (0.008 sec)

MariaDB [world]> SHOW TABLES FROM world;
+-----------------+
| Tables_in_world |
+-----------------+
| Town            |
| city            |
| country         |
+-----------------+
3 rows in set (0.000 sec)

MariaDB [world]> DROP TABLE world.Town;
Query OK, 0 rows affected (0.004 sec)

MariaDB [world]> SHOW TABLES FROM world;
+-----------------+
| Tables_in_world |
+-----------------+
| city            |
| country         |
+-----------------+
2 rows in set (0.000 sec)

MariaDB [world]> DROP TABLE world.city;
Query OK, 0 rows affected (0.007 sec)

MariaDB [world]> SHOW TABLES FROM world;
+-----------------+
| Tables_in_world |
+-----------------+
| country         |
+-----------------+
1 row in set (0.000 sec)

MariaDB [world]> █
```

Task 3: Delete a database and tables

In this task, you delete the **world** database and **country** table.

20. The **DROP TABLE** command is used to delete (drop) a table in a database. Once a table has been dropped, it cannot be recovered unless a backup is available. To drop the **city** table, run the following command.
21. `DROP TABLE world.city;`

Challenge 2

Write a query to drop the **country** table.

```
DROP TABLE world.country;
```

Tip: Expand the question to reveal the solution.

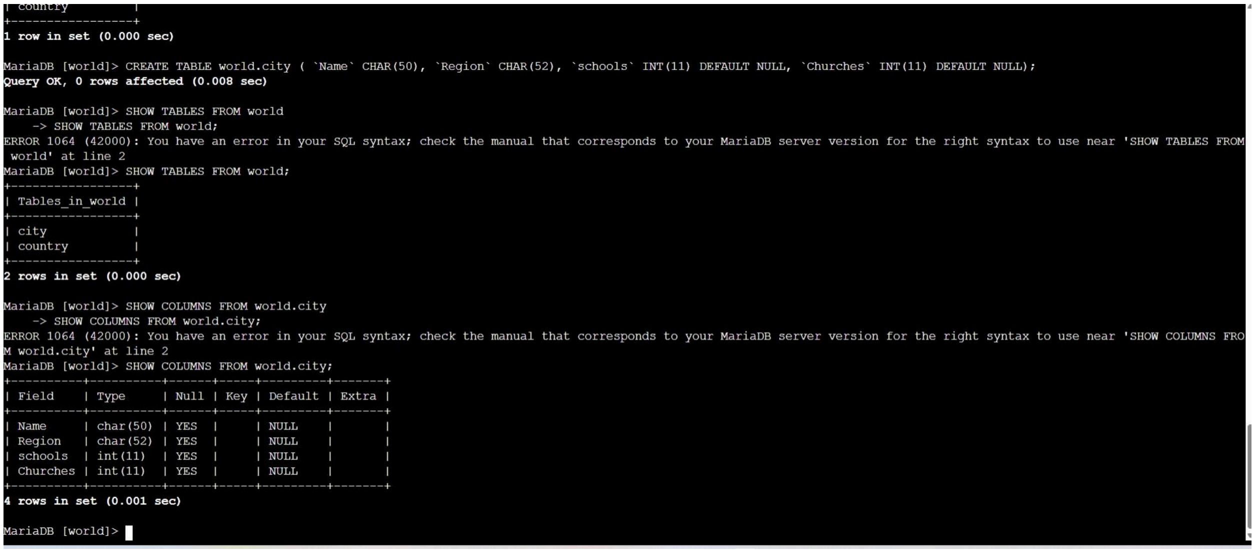
21. To verify that both tables have been dropped, run the following query.
22. `SHOW TABLES;`
23. To drop the **world** database, run the following command.
24. `DROP DATABASE world;`
25. To verify that the **world** database has been deleted, run the following query.
26. `SHOW DATABASES;`

Conclusion

Congratulations! You now have successfully:

- Used the **CREATE** statement to create databases and tables

- Used the **SHOW** statement to view available databases and tables
- Used the **ALTER** statement to alter the structure of a table
- Used the **DROP** statement to delete databases and tables



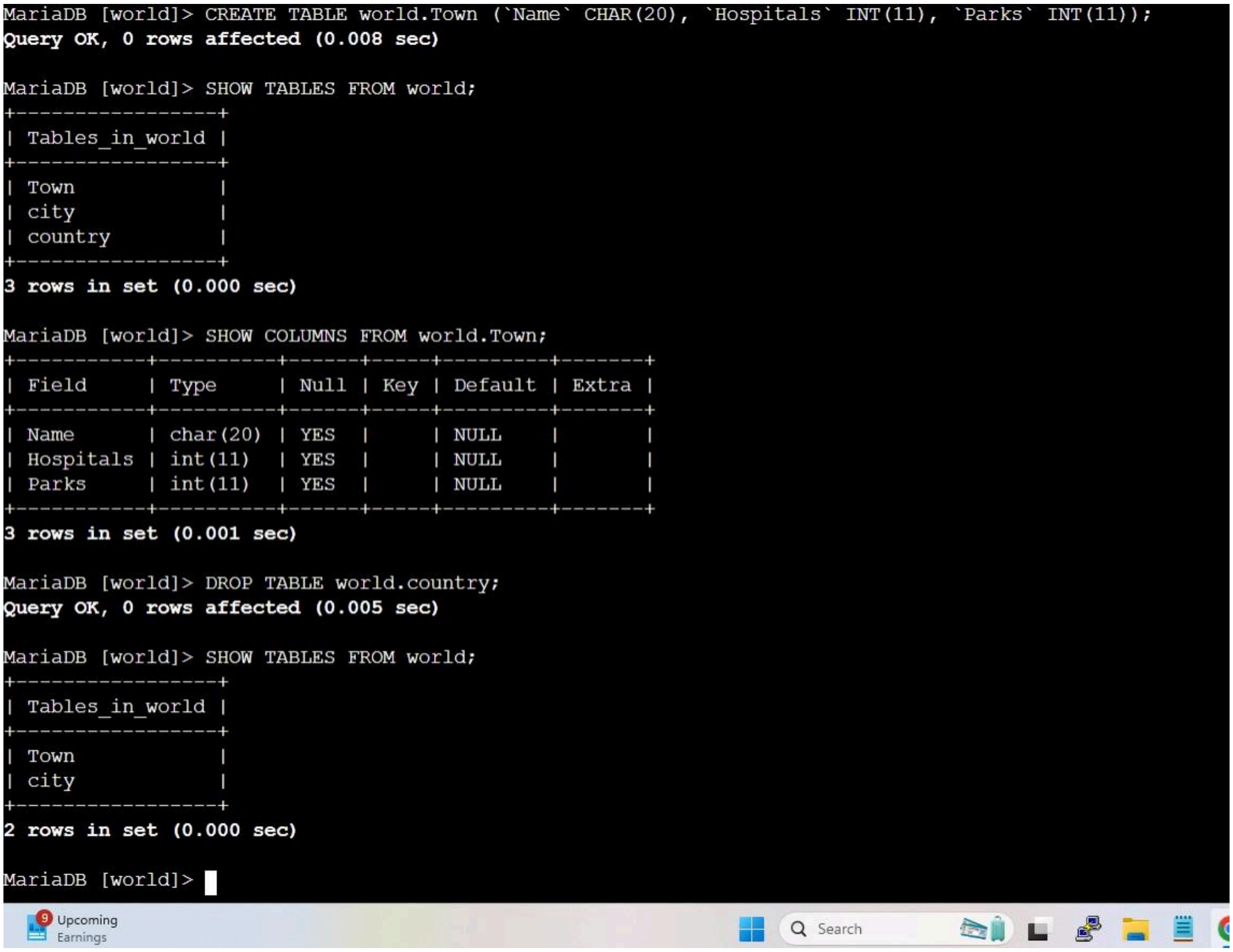
```
| country         |
+-----------------+
1 row in set (0.000 sec)

MariaDB [world]> CREATE TABLE world.city ( `Name` CHAR(50), `Region` CHAR(52), `schools` INT(11) DEFAULT NULL, `Churches` INT(11) DEFAULT NULL);
Query OK, 0 rows affected (0.008 sec)

MariaDB [world]> SHOW TABLES FROM world
    -> SHOW TABLES FROM world;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'SHOW TABLES FROM
 world' at line 2
MariaDB [world]> SHOW TABLES FROM world;
+-----------------+
| Tables_in_world |
+-----------------+
| city            |
| country         |
+-----------------+
2 rows in set (0.000 sec)

MariaDB [world]> SHOW COLUMNS FROM world.city
    -> SHOW COLUMNS FROM world.city;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'SHOW COLUMNS FRO
M world.city' at line 2
MariaDB [world]> SHOW COLUMNS FROM world.city;
+----------+----------+------+-----+---------+-------+
| Field    | Type     | Null | Key | Default | Extra |
+----------+----------+------+-----+---------+-------+
| Name     | char(50) | YES  |     | NULL    |       |
| Region   | char(52) | YES  |     | NULL    |       |
| schools  | int(11)  | YES  |     | NULL    |       |
| Churches | int(11)  | YES  |     | NULL    |       |
+----------+----------+------+-----+---------+-------+
4 rows in set (0.001 sec)

MariaDB [world]>
```



```
MariaDB [world]> CREATE TABLE world.Town (`Name` CHAR(20), `Hospitals` INT(11), `Parks` INT(11));
Query OK, 0 rows affected (0.008 sec)

MariaDB [world]> SHOW TABLES FROM world;
+-----------------+
| Tables_in_world |
+-----------------+
| Town            |
| city            |
| country         |
+-----------------+
3 rows in set (0.000 sec)

MariaDB [world]> SHOW COLUMNS FROM world.Town;
+-----------+----------+------+-----+---------+-------+
| Field     | Type     | Null | Key | Default | Extra |
+-----------+----------+------+-----+---------+-------+
| Name      | char(20) | YES  |     | NULL    |       |
| Hospitals | int(11)  | YES  |     | NULL    |       |
| Parks     | int(11)  | YES  |     | NULL    |       |
+-----------+----------+------+-----+---------+-------+
3 rows in set (0.001 sec)

MariaDB [world]> DROP TABLE world.country;
Query OK, 0 rows affected (0.005 sec)

MariaDB [world]> SHOW TABLES FROM world;
+-----------------+
| Tables_in_world |
+-----------------+
| Town            |
| city            |
+-----------------+
2 rows in set (0.000 sec)

MariaDB [world]>
```