

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.

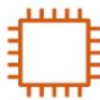
Lab overview and objectives

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

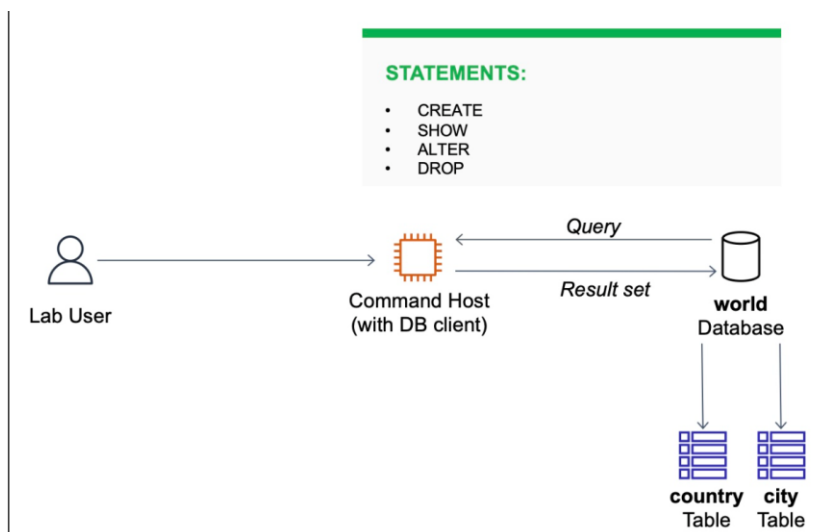
When you start the lab, the following resources are already created for you:



Command Host
(with DB client)

A database client is installed on an instance.

At the end of this lab, you will have completed some common database and table operations:



A lab user creates a database and tables. Other displayed statements are SHOW, ALTER, and DROP. Sample data in this course is taken from Statistics Finland, general regional statistics, February 4, 2022.

1. Connect to the Command Host

I connected to a remote computer (called a Command Host) on Amazon Web Services (AWS).

To do this, I:

1. Logged into AWS and found the Command Host computer.
2. Opened a connection to the computer, which gave me a terminal window.
3. Told the computer to give me access to certain tools and files.
4. Connected to a database on the computer using a special command.

The first screenshot shows the AWS Management Console 'Instances' page. A table lists the instances, with 'Command Host' (ID: i-00104903050380644) in a 'Running' state. Below the table, the 'Details' tab for this instance is selected, showing its public IPv4 address (52.12.87.244) and public DNS (ec2-52-12-87-244.us-west-2.compute.amazonaws.com).

The second screenshot shows the 'Connect to instance' page for the same instance. The 'EC2 Instance Connect' tab is active. Under 'Connection type', 'Connect using a Public IP' is selected. The 'Public IPv4 address' is 52.12.87.244. The 'Username' is set to 'ec2-user'. A note at the bottom states: 'Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.'

MOKGADI SELEPE (AWS Re-start)

The screenshot displays the AWS Management Console interface for connecting to an EC2 instance. The top navigation bar shows the AWS logo, a search bar, and the account ID: 2607-4886-3858. The breadcrumb trail indicates the path: EC2 > Instances > i-00104903050380644 > Connect to instance.

The main section is titled 'Connect' with a sub-header 'Connect to an instance using the browser-based client.' Below this, there are four tabs: 'EC2 Instance Connect', 'Session Manager' (selected), 'SSH client', and 'EC2 serial console'. A blue banner at the top of the 'Session Manager' tab reads: 'Introducing Systems Manager just-in-time node access. Move towards zero standing privileges by requiring operators to request access before remotely connecting to instances. [Learn more](#)'. A 'Try for free' button is also present.

Under the 'Session Manager usage:' section, there are four bullet points:

- Connect to your instance without SSH keys, a bastion host, or opening any inbound ports.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager [Preferences](#) page.

At the bottom of the 'Connect' section, there are 'Cancel' and 'Connect' buttons.

Below the 'Connect' section, the 'CloudShell' interface is visible. It shows the session ID: user4473058=Mokgadi_Selepe-4062mceplcnp5646gn5cy3gs4 and the instance ID: i-00104903050380644. The terminal window displays the following commands and output:

```
sh-4.2$ sudo su
[root@ip-10-1-11-101 bin]# cd /home/ec2-user/
[root@ip-10-1-11-101 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 5
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)]>
```

2. Create a database and a table

I created a database and a table.

1. Checked existing databases: Listed the databases I already had.
2. Created a new database: Made a new database named "world".
3. Created a table: Made a table named "country" with many columns (like "Code", "Name", "Continent", etc.) to store data.
4. Found a mistake: The "Continent" column was misspelled as "Conitinent".
5. Fixed the mistake: Corrected the spelling of the "Continent" column.

In short, I:

- Created a database named "world"
- Created a table named "country" with many columns
- Fixed a spelling mistake in one of the column names

Now I have a database and table set up and ready to use!

```
Session ID: user4473058-Mokgadi_Selepe-2uy9hdbku3n4ucs8nmoto2fci Shortcuts Instance ID: i-00104903050380644 Terminate

sh-4.2$ sudo su
[root@ip-10-1-11-101 bin]# cd /home/ec2-user/
[root@ip-10-1-11-101 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 5
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)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.001 sec)

MariaDB [(none)]> CREATE DATABASE world;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| world |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]>

Session ID: user4473058-Mokgadi_Selepe-2uy9hdbku3n4ucs8nmoto2fci Shortcuts Instance ID: i-00104903050380644 Terminate

+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| world |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]> CREATE TABLE world.country (
  -> `Code` CHAR(3) NOT NULL DEFAULT '',
  -> `Name` CHAR(52) NOT NULL DEFAULT '',
  -> `Continent` 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`)
  -> );
Query OK, 0 rows affected (0.010 sec)

MariaDB [(none)]> USE world;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [world]> SHOW TABLES;
```

MOKGADI SELEPE (AWS Re-start)

```
MariaDB [(none)]> USE world;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

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

+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| Code  | char(3) | NO | PRI |          |       |
| Name  | char(52) | NO |      |          |       |
| Continent | enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America') | NO |      | Asia |       |
| Region | char(26) | NO |      |          |       |
| SurfaceArea | float(10,2) | NO |      | 0.00 |       |
| IndepYear | smallint(6) | YES |      | NULL |       |
| Population | int(11) | NO |      | 0 |       |
| LifeExpectancy | float(3,1) | YES |      | NULL |       |
| GNP | float(10,2) | YES |      | NULL |       |
| GNPOld | float(10,2) | YES |      | NULL |       |
| LocalName | char(45) | NO |      |          |       |
| GovernmentForm | char(45) | NO |      |          |       |
| HeadOfState | char(60) | YES |      | NULL |       |
| Capital | int(11) | YES |      | NULL |       |
| Code2 | char(2) | NO |      |          |       |
+-----+
15 rows in set (0.000 sec)

MariaDB [world]>

+-----+
| GNPOld | float(10,2) | YES |      | NULL |       |
| LocalName | char(45) | NO |      |          |       |
| GovernmentForm | char(45) | NO |      |          |       |
| HeadOfState | char(60) | YES |      | NULL |       |
| Capital | int(11) | YES |      | NULL |       |
| Code2 | char(2) | NO |      |          |       |
+-----+
15 rows in set (0.000 sec)

MariaDB [world]> ALTER TABLE world.country RENAME COLUMN Conitinent TO Continent;
Query OK, 0 rows affected (0.005 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [world]> SHOW COLUMNS FROM world.country;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| Code  | char(3) | NO | PRI |          |       |
| Name  | char(52) | NO |      |          |       |
| Continent | enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America') | NO |      | Asia |       |
| Region | char(26) | NO |      |          |       |
| SurfaceArea | float(10,2) | NO |      | 0.00 |       |
| IndepYear | smallint(6) | YES |      | NULL |       |
| Population | int(11) | NO |      | 0 |       |
| LifeExpectancy | float(3,1) | YES |      | NULL |       |
| GNP | float(10,2) | YES |      | NULL |       |
| GNPOld | float(10,2) | YES |      | NULL |       |
| LocalName | char(45) | NO |      |          |       |
| GovernmentForm | char(45) | NO |      |          |       |
| HeadOfState | char(60) | YES |      | NULL |       |
| Capital | int(11) | YES |      | NULL |       |
| Code2 | char(2) | NO |      |          |       |
+-----+
15 rows in set (0.001 sec)

MariaDB [world]>
```

Challenge: 1

```
MariaDB [world]> CREATE TABLE world.city (`Name` CHAR(52), `Region` CHAR(26));
Query OK, 0 rows affected (0.008 sec)

MariaDB [world]> SHOW COLUMNS FROM world.city;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| Name  | char(52) | YES |      | NULL |       |
| Region | char(26) | YES |      | NULL |       |
+-----+
2 rows in set (0.001 sec)

MariaDB [world]>
```

I created a table named "city" with two columns:

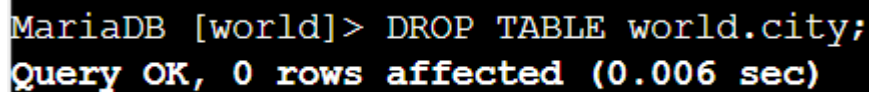
1. Name: This column can hold up to 52 characters (letters, numbers, etc.).
2. Region: This column can hold up to 26 characters.

Both columns use the CHAR data type, which means they store fixed-length strings.

The table is created in the "world" database.

The code used to create this table is:

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



```
MariaDB [world]> DROP TABLE world.city;  
Query OK, 0 rows affected (0.006 sec)
```

I deleted a table.

The table that was deleted is named "city" and it's located in the "world" database.

The command used to delete the table is:

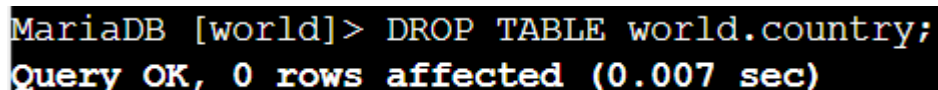
```
DROP TABLE world.city;
```

What this means:

- The table "city" is now gone and can't be used anymore.
- Any data stored in the "city" table is deleted and can't be recovered unless there's a backup.

Note that the entire "world" database wasn't deleted, just the "city" table within it. If you want to delete the entire database, you would use a different command, like DROP DATABASE world;.

Challenge: 2



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

I deleted the "country" table.

The table that was deleted is named "country" and it's located in the "world" database.

MOKGADI SELEPE (AWS Re-start)

The command used to delete the table is:

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

What this means:

- The "country" table is now gone and can't be used anymore.
- Any data stored in the "country" table is deleted and can't be recovered.

The table is permanently removed from the database.

```
Region C...' at line 2
MariaDB [world]> CREATE TABLE world.city (`Name` CHAR(52), `Region` CHAR(26));
Query OK, 0 rows affected (0.008 sec)

MariaDB [world]> SHOW COLUMNS FROM world.city;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Name  | char(52) | YES  |     | NULL    |       |
| Region | char(26) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [world]> DROP TABLE world.city;
Query OK, 0 rows affected (0.006 sec)

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

MariaDB [world]> SHOW TABLES;
Empty set (0.000 sec)

MariaDB [world]> DROP DATABASE world;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql          |
| performance_schema |
+-----+
3 rows in set (0.000 sec)

MariaDB [(none)]>
```

I did three things:

1. Verified tables are deleted: I checked if the tables ("country" and "city") were deleted from the "world" database by running SHOW TABLES;.
2. Deleted the "world" database: I used the command DROP DATABASE world; to delete the entire "world" database.

3. Verified database deletion: I checked if the "world" database was successfully deleted by running SHOW DATABASES;.

What this means:

- The "world" database and all its tables (including "country" and "city") are now gone and can't be used anymore.
- All data stored in the "world" database is deleted and can't be recovered.

Everything is removed and cleaned up!

Conclusion

I completed a task and learned some things:

I successfully:

1. Created databases and tables using the CREATE statement.
2. Viewed databases and tables using the SHOW statement.
3. Changed a table's structure using the ALTER statement.
4. Deleted databases and tables using the DROP statement.

What this means:

I now know the basics of managing databases and tables using SQL commands. I can create, view, modify, and delete databases and tables.
