Selecting Data from a Database

Scenario

The database operations team has created a relational database named **world** containing three tables: **city**, **country**, and **countrylanguage**. Based on specific use cases defined in the lab exercise, your write a few queries using database operators and the **SELECT** statement.

Lab overview and objectives

This lab demonstrates how to use some common database operators and the **SELECT** statement.

After completing this lab, you should be able to:

- Use the **SELECT** statement to query a database
- Use the **COUNT()** function
- Use the following operators to query a database:
  - **<**
  - **>**
  - **=**
  - **WHERE**
  - **ORDER BY**
  - **AND**

When you start the lab, the following resources are already created for you:
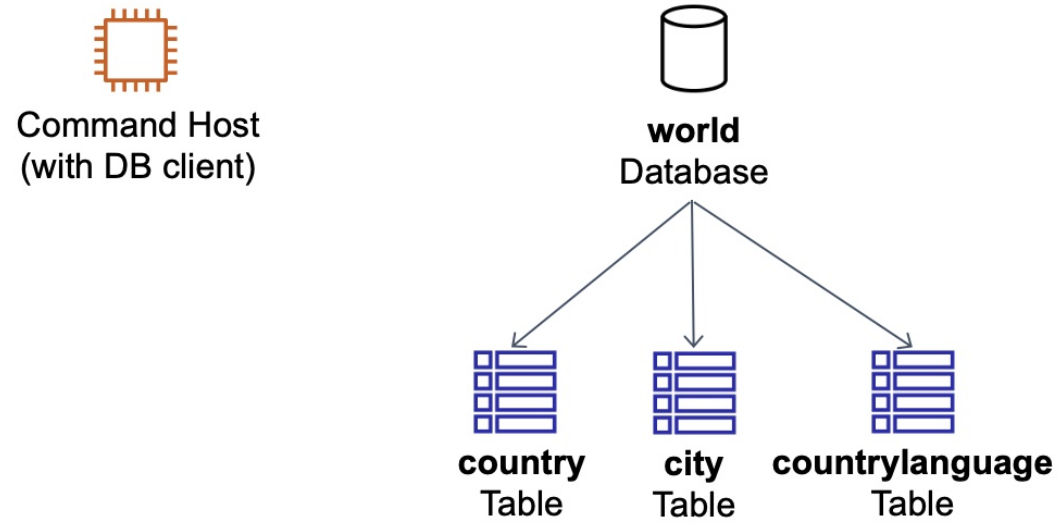
A Command Host instance and world database containing three tables

At the end of this lab, you will have used the **SELECT** statement and some common database operators:
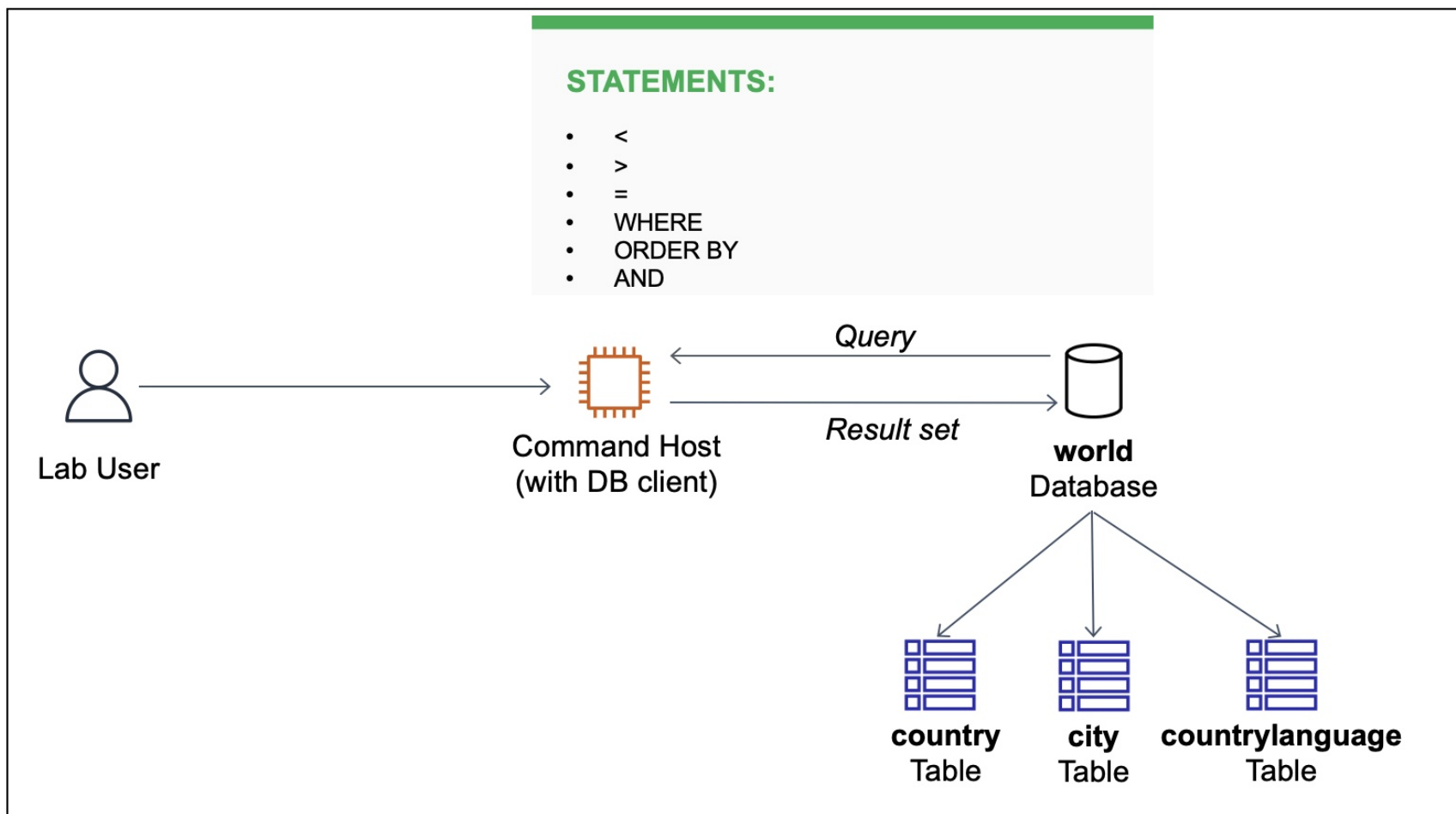
A lab user is connected to a database instance. It also displays some commonly used database operations.

Sample data in this course is taken from Statistics Finland, general regional statistics, February 4, 2022.

Duration

This lab requires approximately **45 minutes** to complete.

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that you need to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that this lab describes.

Accessing the AWS Management Console

1. At the upper-right corner of these instructions, choose ▶ **Start Lab**

   Troubleshooting tip: If you get an **Access Denied** error, close the error box, and choose ▶ **Start Lab** again.

2. The following information indicates the lab status:

   - A red circle next to **AWS** 🔴 at the upper-left corner of this page indicates that the lab has not been started.
   - A yellow circle next to **AWS** 🟡 at the upper-left corner of this page indicates that the lab is starting.
   - A green circle next to **AWS** 🟢 at the upper-left corner of this page indicates that the lab is ready.

   Wait for the lab to be ready before proceeding.

3. At the top of these instructions, choose the green circle next to **AWS** 🟢

   This option opens the AWS Management Console in a new browser tab. The system automatically sign you in.

   Tip: If a new browser tab does not open, a banner or icon at the top of your browser will indicate that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and choose **Allow pop-ups**.

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you should be able to see both browser tabs at the same time so that you can follow the lab steps.

   ⚠️ **Do not change the lab Region unless specifically instructed to do so**.

Task 1: Connect to the Command Host

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. 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 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:

```
sudo su
cd /home/ec2-user/
```

ⓘ 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.

11. To connect to the database server, run the following command in the terminal. A password was configured when the database was installed.

```
mysql -u root --password='re:St@rt!9'
```

ⓘ Tip: At any stage of the lab, if the Sessions Manager window is not responsive or if you need to reconnect to the database, then follow these steps:

- Close the Sessions 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'
```

Task 2: Query the world database

In this task, you query the **world** database using various **SELECT** statements and database operators.

12. To show the existing databases, enter the following command in the terminal.

```
SHOW DATABASES;
```

Verify that a database named **world** is available. If the **world** database is not available, then contact your instructor.

13. To list all rows and columns in the **country** table, run the following query.

```
SELECT * FROM world.country;
```

14. To query the number of rows in a table, you can use the **COUNT()** function in a **SELECT** statement. To count all the rows in table, you can use **COUNT(*)**. To count the number of rows that have a value in a specific column, include the column name as a parameter in the COUNT() function: for example, COUNT(Population). To list the number of rows in the **country** table, run the following query.

```
SELECT COUNT(*) FROM world.country;
```

15. To list all columns in the **country** table, run the following query. You run this query to understand the table schema.

```
SHOW COLUMNS FROM world.country;
```

16. To query specific columns in the **world** table, run the following query to return a result set that includes the Name, Capital, Region, SurfaceArea, and Population columns.

```
SELECT Name, Capital, Region, SurfaceArea, Population FROM world.country;
```

17. Database column names are sometimes not user friendly. To add a more descriptive column name to the query output, you can use the **AS** option. Run the following query that includes this option.

```
SELECT Name, Capital, Region, SurfaceArea AS "Surface Area", Population FROM world.country;
```

If required, scroll to the top of the query results, and observe that the **SurfaceArea** column is displayed as **Surface Area**.

18. Ordered result sets are easier to view and work with. If you would like to order the output based on a column, you can use the **ORDER BY** option. In this example, you order the output based on the population.

```
SELECT Name, Capital, Region, SurfaceArea AS "Surface Area", Population FROM world.country ORDER BY Population;
```

The **ORDER BY** option orders data in ascending order.

19. To order data in descending order, use the **DESC** option with **ORDER BY**. Run the following command with this option.

```
SELECT Name, Capital, Region, SurfaceArea AS "Surface Area", Population FROM world.country ORDER BY Population
DESC;
```

20. You can add conditions to **SELECT** statements by using the **WHERE** clause. For example, to list all rows with a population greater than 50,000,000, run the following query.

```
SELECT Name, Capital, Region, SurfaceArea AS "Surface Area", Population FROM world.country WHERE Population >
50000000 ORDER BY Population DESC;
```

You have used the **>** comparison operator. Similarly, you can use other comparison operators to compare values.

21. You can construct a **WHERE** clause by using a number of conditions and operators.

The following query uses two conditions: all rows with a population greater than 50,000,000 and all rows with a population less than 100,000,000. The query includes the **AND** operator to indicate that both the conditions must be true. Run the following query in your terminal.

```
SELECT Name, Capital, Region, SurfaceArea AS "Surface Area", Population FROM world.country WHERE Population >
50000000 AND Population < 100000000 ORDER BY Population DESC;
```

For more information about comparison operators, see the **Additional resources** section at the end of the lab.

Challenge

Query the **country** table to return a set of records based on the following question.

▼ Which country in Southern Europe has a population greater than 50,000,000?

```
SELECT Name, Capital, Region, SurfaceArea AS "Surface Area", Population from world.country WHERE Population > 50000000
AND Region = "Southern Europe";
```

ⓘ **Tip**: Expand the question to reveal the solution.

Conclusion

👍 Congratulations! You have now successfully:

- Used the **SELECT** statement to query a database

- Used the **COUNT()** function

- Used the following operators to query a database:

  - <
  - >
  - =
  - **WHERE**
  - **ORDER BY**
  - **AND**

Lab complete 🎓

22. Choose ■ **End Lab** at the top of this page, and then select Yes to confirm that you want to end the lab.
23. An **Ended AWS Lab Successfully** message is briefly displayed indicating that the lab has ended.

Additional resources

- Country, city, and language data, Statistics Finland: The material was downloaded from Statistics Finland's interface service on February 4, 2022, with the license CC BY 4.0. The original data source is available from Statistics Finland.

- For more information about database functions and operators, see the following resources:

  - SELECT statements
  - Count function
  - Order By
  - Operators
  - Comparison Operators

For more information about AWS Training and Certification, see AWS Training and Certification.

Your feedback is welcome and appreciated.

If you would like to share any suggestions or corrections, please provide the details in our AWS Training and Certification Contact Form.