Working with Functions

Scenario

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

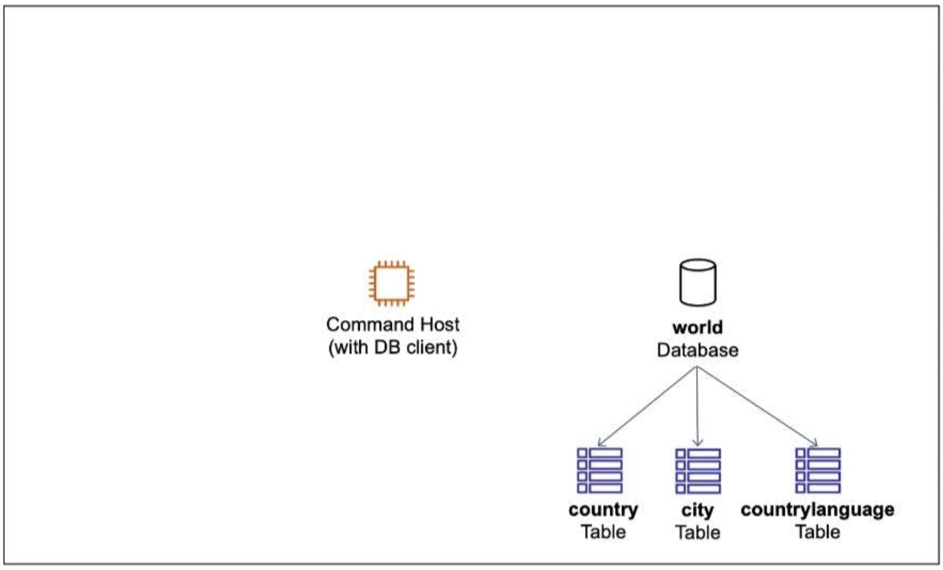
Lab overview and objectives

This lab demonstrates how to use some common database functions with the **SELECT** statement and **WHERE** clause.

After completing this lab, you should be able to:

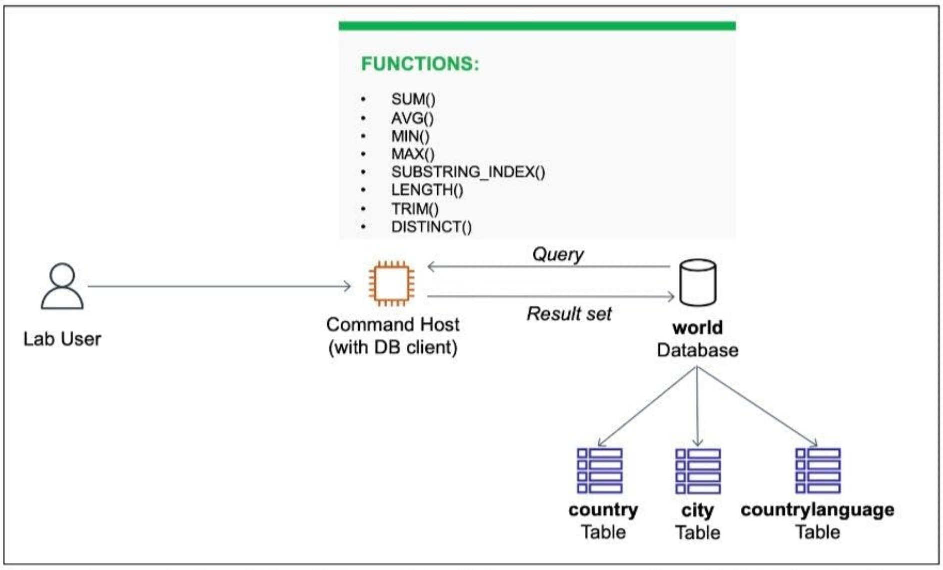
- Use aggregate functions **SUM()**, **MIN()**, **MAX()**, and **AVG()** to summarize data
- Use the **SUBSTRING_INDEX()** function to split strings
- Use the **LENGTH()** and **TRIM()** functions to determine the length of a string
- Use the **DISTINCT()** function to filter duplicate records
- Use functions in the **SELECT** statement and **WHERE** clause

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 would have used the **SELECT** statement and **WHERE** clause with some common database functions:



A lab user is connected to a database instance. It also displays some commonly used SQL database functions

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. Under **Compute**, choose **EC2**.

6. In the 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:

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

   ℹ **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'
```

   ℹ **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.

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.

```
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 functions. You use a function to process and manipulate data in a query. There are a wide range of SQL functions, and this lab reviews a subset of commonly used functions.

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, contact your instructor.

13. To review the table schema, data, and number of rows in the **country** table, run the following query.

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

14. The following query demonstrates how to use aggregate functions **SUM()**, **MIN()**, **MAX()**, and **AVG()** to summarize data. Because the query does not include a **WHERE** condition, the functions aggregate data from all records in the **country** table. Run the following query.

```
SELECT sum(Population), avg(Population), max(Population), min(Population),
count(Population) FROM world.country;
```

- SUM() adds all the population values together.
- AVG() generates an average across all the population values.
- MAX() finds the row with the highest population value.
- MIN() finds the row with the lowest population value.
- COUNT() finds the number of rows with a population value.

15. In some cases, you might need to split a string. The following query uses **SUBSTRING_FUNCTION()** to spilt a string where a space occurs. Run the following query.

```
SELECT Region, substring_index(Region, " ", 1) FROM world.country;
```

After you run the query, you notice that the second column includes the beginning of each region name.

16. Sometimes you may need to search rows using a string fragment. The following query includes **SUBSTRING_FUNCTION()** as part of a condition in the **WHERE** clause to filter records that include **Southern** in the first part of the region name. Run the following query.

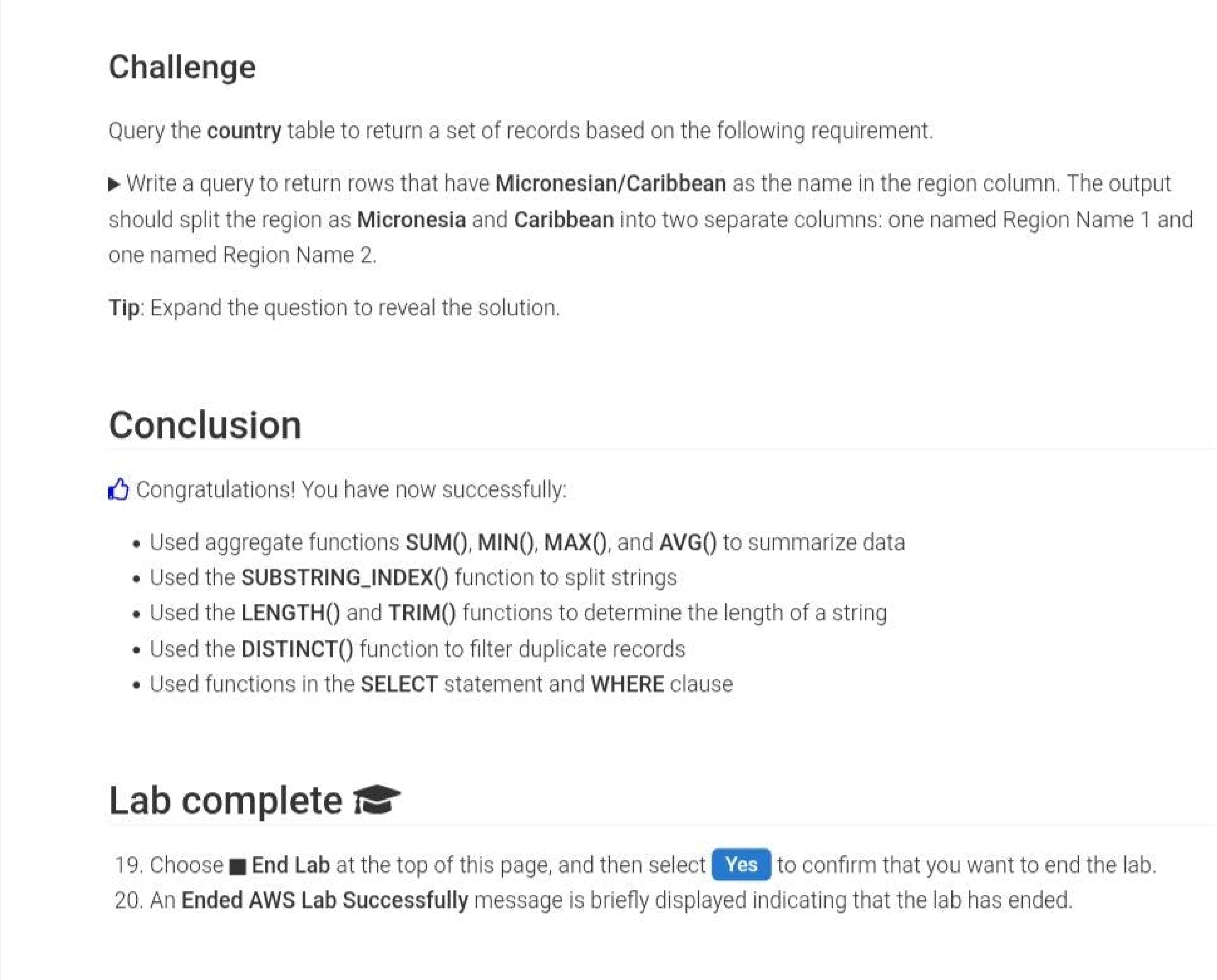
```
SELECT Name, Region from world.country WHERE substring_index(Region, " ", 1) =
"Southern";
```

17. You can use the **LENGTH()** and **TRIM()** functions to determine how many characters are in a string. **TRIM()** clears leading and trailing blank spaces, and the **LENGTH()** function returns a count of the remaining characters. The next example returns only regions that have fewer than 10 characters in their names. Run the following query.

```
SELECT Region FROM world.country WHERE LENGTH(TRIM(Region)) < 10;
```

18. You might have noticed duplicate records in the previous example. You can use the **DISTINCT()** function to filter the duplicates. Run the following query.

```
SELECT DISTINCT(Region) FROM world.country WHERE LENGTH(TRIM(Region)) < 10;
```



Challenge

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

▶ Write a query to return rows that have **Micronesian/Caribbean** as the name in the region column. The output should split the region as **Micronesia** and **Caribbean** into two separate columns: one named Region Name 1 and one named Region Name 2.

Tip: Expand the question to reveal the solution.

Conclusion

👍 Congratulations! You have now successfully:

- Used aggregate functions **SUM()**, **MIN()**, **MAX()**, and **AVG()** to summarize data
- Used the **SUBSTRING_INDEX()** function to split strings
- Used the **LENGTH()** and **TRIM()** functions to determine the length of a string
- Used the **DISTINCT()** function to filter duplicate records
- Used functions in the **SELECT** statement and **WHERE** clause

Lab complete 🎓

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