Performing a Conditional Search

Scenario

The database operations team has created a relational database named **world** containing three tables: **city**, **country**, and **countrylanguage**. To help the team, you will write a few queries to search for records in the **country** table by using the **SELECT** statement and a **WHERE** clause.

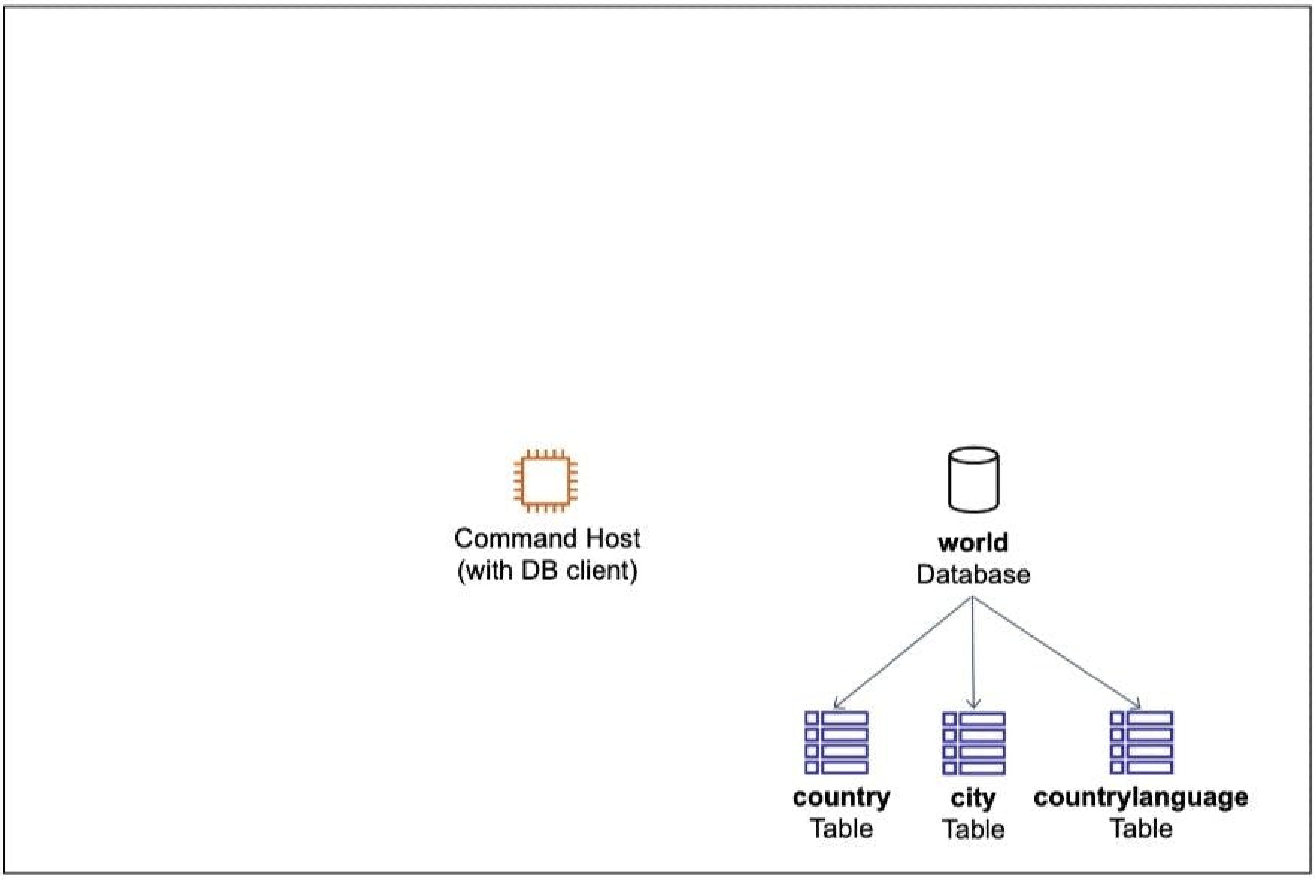
Lab overview and objectives

This lab demonstrates how to use the **SELECT** statement and a **WHERE** clause to filter records with a conditional search.

After completing this lab, you should be able to:

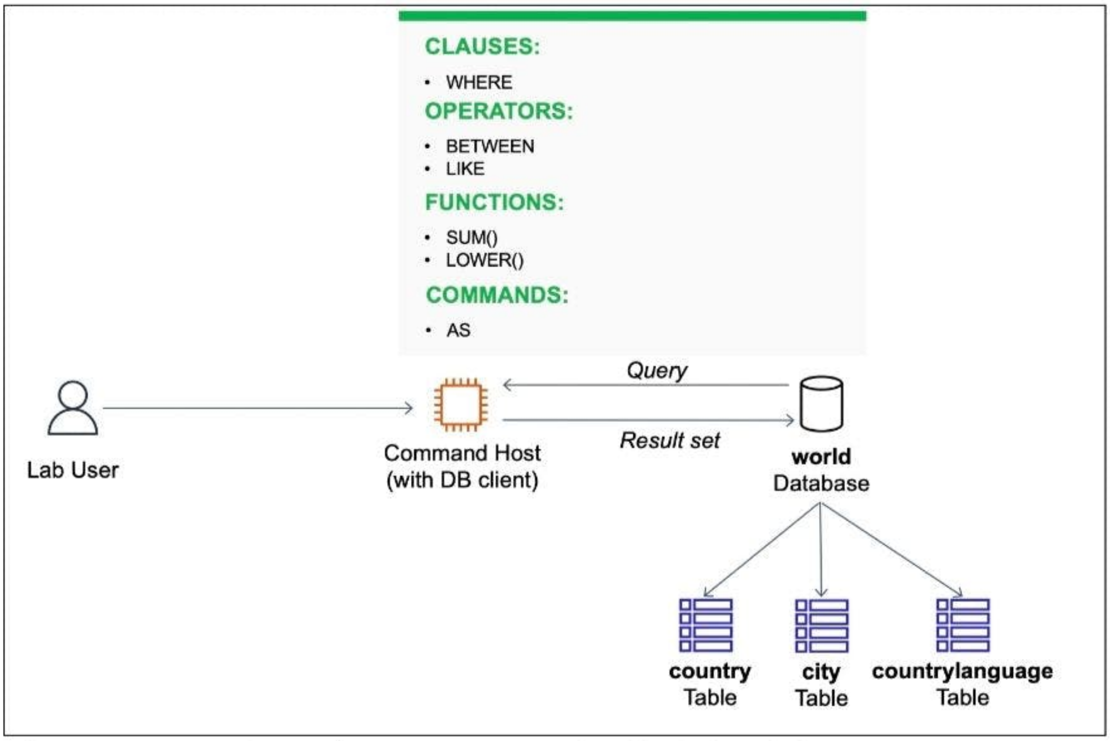
- Write a search condition by using the **WHERE** clause
- Use the **BETWEEN** operator
- Use the **LIKE** operator with wildcard characters
- Use the **AS** operator to create a column alias
- Use functions in a **SELECT** statement
- Use functions in a **WHERE** clause

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



*The Command Host instance has a DB client installed. You will use the Command Host to query the **world** database, which contains three tables.*

At the end of this lab, you will have learned how to use the **WHERE** clause, **BETWEEN** operator, and **LIKE** function to filter records:



A lab user connects to a Command Host instance to query the tables in the ***world*** *database.*

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 lab status can be interpreted as follows:

   - 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 signs 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 Amazon Elastic Compute Cloud (Amazon EC2) instance containing a database client, which you will use to connect to a database. This instance will be 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 might 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. 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 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'
      ```

Task 2: Query the world database

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

12. To show the existing databases, run the following query.

    ```
    SHOW DATABASES;
    ```

    Verify that a database named **world** is available. If the **world** database is not available, then 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;
    ```

    By reducing the number of records, the result set would be smaller and easier to work with. To limit the number of records returned, you can use a **WHERE** clause to define the conditions that records must match.

    Use the **AND** operator to combine two conditions. Each record is checked against both conditions before it's included in the result set. You can use the **>** operator and **=** operator to query values that are greater than or equal to a certain value. Similarly, you can combine the **<** operator and **=** operator to query values that are less than or equal to a certain value.
14. To reduce the number of records in the result set by using a **WHERE** clause and the **AND** operator, run the following query.

    ```
    SELECT Name, Capital, Region, SurfaceArea, Population FROM world.country WHERE
    Population >= 50000000 AND Population <= 100000000;
    ```

    When searching for records by using a range condition, you can use the **BETWEEN** operator instead of the **>=** operator and **<=** operator. By using the **BETWEEN** operator, the query is easier to read. Note that the operator is inclusive, meaning that the beginning and ending values are included.

15. To return the same records as the previous result set by using the **BETWEEN** operator, run the following query.

    ```
    SELECT Name, Capital, Region, SurfaceArea, Population FROM world.country WHERE
    Population BETWEEN 50000000 AND 100000000;
    ```

    You can use the **LIKE** function to search for a string pattern. The following query will return records that include the string *Europe* in the **Region** column. The percent symbol (%) is a wildcard character that represents any number of characters before or after the word *Europe*. The query will aggregate the population of all European countries by using the **SUM** function.

16. To return the population of all European countries by using the **LIKE** function and **SUM** function, run the following query.

    ```
    SELECT sum(Population) from world.country WHERE Region LIKE "%Europe%";
    ```

    In the previous query, the **SELECT** clause included a **SUM** function. In the following query, the **SUM** function is still used to calculate the total population of *Europe*. The query also includes a *column alias*, which makes the output easier to read. To define the column alias, the **AS** command is used in the **SELECT** statement.

17. To return the same information as the previous query with the column alias, run the following query.

    ```
    SELECT sum(population) as "Europe Population Total" from world.country WHERE region
    LIKE "%Europe%";
    ```

    Note that SQL is not a case-sensitive language. You can use either **SELECT** or **select** when writing a query. However, databases that you query might be configured with a case-sensitive collation. If the database was case sensitive, you would not be able to query a column named **Population** by using the following: `select population from world.country`

    Even though the database used in this lab is not case sensitive, we recommended making your queries consistent with the naming convention that is used in the database.

    The following example demonstrates how to perform a case-sensitive search. Depending on the database configuration, when comparing **Central** to **central**, the outcome might be false, because the strings don't use the same case. To solve this problem, you can use the **LOWER** function in the **WHERE** clause to compare the strings both as lowercase.

18. To perform a case-sensitive search by using the **LOWER** function, run the following query.

    ```
    SELECT Name, Capital, Region, SurfaceArea, Population from world.country WHERE
    LOWER(Region) LIKE "%central%";
    ```

Challenge

Write a query to return the sum of the surface area and sum of the population of North America.

▶ Query the table first to determine which columns and operators to use.

Tip: Expand the question to reveal the solution.

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.