SQL Fundamentals Project – Basic Data Querying and Sorting

Introduction

In this project, I worked with a virtual machine (VM) environment to practice foundational SQL querying skills. The objective was to interact with a sample database and execute various SQL commands to retrieve, filter, and organize data. Using the SQL command-line interface. I ran queries to select specific columns from a table, displaying all available data using the * operator, and sort results using the ORDER table by keyword.

Throughout this project, I included screenshots of each step as evidence of the command executed and the resulting outputs. Alongside each screenshot, I provided explanations describing what each command did, how it affected the query results, and why it was used. Conducting the lab in a virtualized environment provided a safe and controlled space to experiment with queries and better understand how structured query language is used to manage analyse data.

Task 1: Displayed All Columns in the Database Table (Machines)

Objective:
To view all the available columns and rows within the Machines table using a basic SQL query

```
^ZMariaDB [organization]> Select *
    -> from machines;
+--------------+------------------+----------------+---------------+----------
---+
| device_id    | operating_system | email_client   | OS_patch_date | employee_
id |
+--------------+------------------+----------------+---------------+----------
---+
| a184b775c707 | OS 1             | Email Client 1 | 2021-09-01    |        11
56 |
| a192b174c940 | OS 2             | Email Client 1 | 2021-06-01    |        10
52 |
| a305b818c708 | OS 3             | Email Client 2 | 2021-06-01    |        11
82 |
| a317b635c465 | OS 1             | Email Client 2 | 2021-03-01    |        11
30 |
| a320b137c219 | OS 2             | Email Client 2 | 2021-03-01    |        10
00 |
```

Command: Select *
from machines;

Explanation: This command retrieved all the columns and rows from the table named Machines. The * symbol is used as a wildcard, meaning “select everything”. It displays every piece of data stored in the machines table, which helps to get an overview of its content before filtering or sorting specific information

Task 1.1: Retrieve Email client & Device ID from 3rd Row in the Machines table

Objective: To run a query that retrieves the device_id and email_client column from the Machines table, and identify which device ID and email client appear in the third row of the query results.

```
MariaDB [organization]> Select device_id, email_client
    -> from machines;
+--------------+----------------+
| device_id    | email_client   |
+--------------+----------------+
| a184b775c707 | Email Client 1 |
| a192b174c940 | Email Client 1 |
| a305b818c708 | Email Client 2 |
| a317b635c465 | Email Client 2 |
| a320b137c219 | Email Client 2 |
| a398b471c573 | Email Client 2 |
| a667b270c984 | Email Client 1 |
| a821b452c176 | Email Client 2 |
| a998b568c863 | Email Client 1 |
```

Command: select device_id, email_client
from machines;

Explanation: This command retrieves only the device_id and email_client columns from the Machines table. The output displays a list of device IDs with their corresponding email clients. By reviewing the results. I was able to identify the third record in the list, which was **email client 2 with device id a305b818c708**.

Task 1.2: Retrieve the Device ID, Operating System, and OS Patch Date Columns from the Machines Table

Objective: Run a query that displays only the device_id, operating_system and OS_patch_date columns from the Machines table, excluding all other data.

```
MariaDB [organization]> select device_id,operating_System,OS_patch_date
    -> from machines;
+--------------+------------------+---------------+
| device_id    | operating_System | OS_patch_date |
+--------------+------------------+---------------+
| a184b775c707 | OS 1             | 2021-09-01    |
| a192b174c940 | OS 2             | 2021-06-01    |
| a305b818c708 | OS 3             | 2021-06-01    |
| a317b635c465 | OS 1             | 2021-03-01    |
| a320b137c219 | OS 2             | 2021-03-01    |
| a398b471c573 | OS 3             | 2021-12-01    |
| a667b270c984 | OS 1             | 2021-03-01    |
```

Command: select device_id, operating_system, OS_patch_date
from machines;

Explanation: This command retrieves specific columns – device_id, operating_system, and os_patch_date – from the Machines table. By listing only the required column names, query filters out unnecessary data and

focuses on information relevant to system identification and patch status. This task reinforces how column selection in SQL can be used to extract precise datasets for analysis or reporting.

Task 2: Check Login Attempts from Expected Countries (United States, Canada, Mexico)

Objective: To run a query that displays only the event_id and country columns from the log_in_attempts table, allowing analysis of login events by country

```
MariaDB [organization]> select event_id,country
    -> from log_in_attempts;
+----------+---------+
| event_id | country |
+----------+---------+
|        1 | CAN     |
|        2 | CAN     |
|        3 | USA     |
|        4 | USA     |
|        5 | CANADA  |
|        6 | MEXICO  |
|        7 | CAN     |
|        8 | US      |
|        9 | MEX     |
|       10 | CANADA  |
```

Command: select event_id, country
from log_in_attempts;

Explanation: This query selects the event_id and country columns from the log_in_attempts table, filtering out all other columns. The results allow us to see each login attempt and the country from which it originated, providing a clear view of login activity across different regions.

Task 2.1: Retrieve Username, Login Date, and Login Time to Identify the Fifth Login Attempt

Objective: To query the username, login_date, and login_time columns from the log_in_attempts table in order to determine whether login attempts occurred outside the organization's working hours, and to identify the username returned in the fifth row of the results.

```
MariaDB [organization]> select username,login_date,login_time
    -> from log_in_attempts;
+----------+------------+------------+
| username | login_date | login_time |
+----------+------------+------------+
| jrafael  | 2022-05-09 | 04:56:27   |
| apatel   | 2022-05-10 | 20:27:27   |
| dkot     | 2022-05-09 | 06:47:41   |
| dkot     | 2022-05-08 | 02:00:39   |
| jrafael  | 2022-05-11 | 03:05:59   |
| arutley  | 2022-05-12 | 17:00:59   |
```

Command: select username, login_date, login_time
from log_in_attempts;

Explanation: The command retrieves the username, login_date, and login_time columns from the log_in_attempts table. Reviewing the output, we can determine the username of the 5th Login attempt is Jrafael with login_date 2022-05-11 & login_time 03:05:59

Task 3: Sort Login Attempts by Date and Time to Identify the First Recorded Login

Objective: To sort the data from the log_in_attempts table by the login_date & login_time column in ascending order, allowing identification of the first recorded login attempt and the corresponding username and date.

```
                      SELECT *
    -> FROM log_in_attempts
    -> ORDER BY login_date, login_time;
+----------+----------+------------+------------+---------+-----------------+-
--------+
| event_id | username | login_date | login_time | country | ip_address      |
success |
+----------+----------+------------+------------+---------+-----------------+-
--------+
|      117 | bsand    | 2022-05-08 | 00:19:11   | USA     | 192.168.197.187 |
      0 |
|       92 | pwashing | 2022-05-08 | 00:36:12   | US      | 192.168.247.219 |
      0 |
|        8 | bisles   | 2022-05-08 | 01:30:17   | US      | 192.168.119.173 |
      0 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |
      0 |
|       80 | cjackson | 2022-05-08 | 02:18:10   | CANADA  | 192.168.33.140  |
      1 |
|       43 | mcouliba | 2022-05-08 | 02:35:34   | CANADA  | 192.168.16.208  |
      0 |
|      184 | alevitsk | 2022-05-08 | 03:09:48   | CAN     | 192.168.33.70   |
```

Command: Select*
From Log_in_attempts
Order by login_date, login_time;

Explanation: This query retrieves the username, login_date, and login_time columns from the log_in_attempts table, then orders the results first by login_date and then by login_time in ascending order.

From the output, we find that the username “bsand” was the first recorded login, which occurred at 00:19:11 on 2022-05-08, in the USA. This demonstrates how SQL can be used to organize and Alyse login data to identify the earliest recorded access in a system.

Conclusion/What I Learned

During this project, I gained practical experience with key SQL concepts and commands, including:

- Using the **Select** statement to retrieve data from tables.
- Applying the ***** wildcard to display all columns in a table.
- Selecting specific columns to extract targeted information.
- Using the **Order** by keyword to sort query results by date and time.
- Identifying specific rows of interest (e.g., first or third record)
- Analysing login data to find **potential irregularities** and **track user activity.**

This project helped strengthen my understanding of how SQL can be used to query, organize and analyse data – skills that are highly relevant to information security and cybersecurity analysis.