Joshua Merren

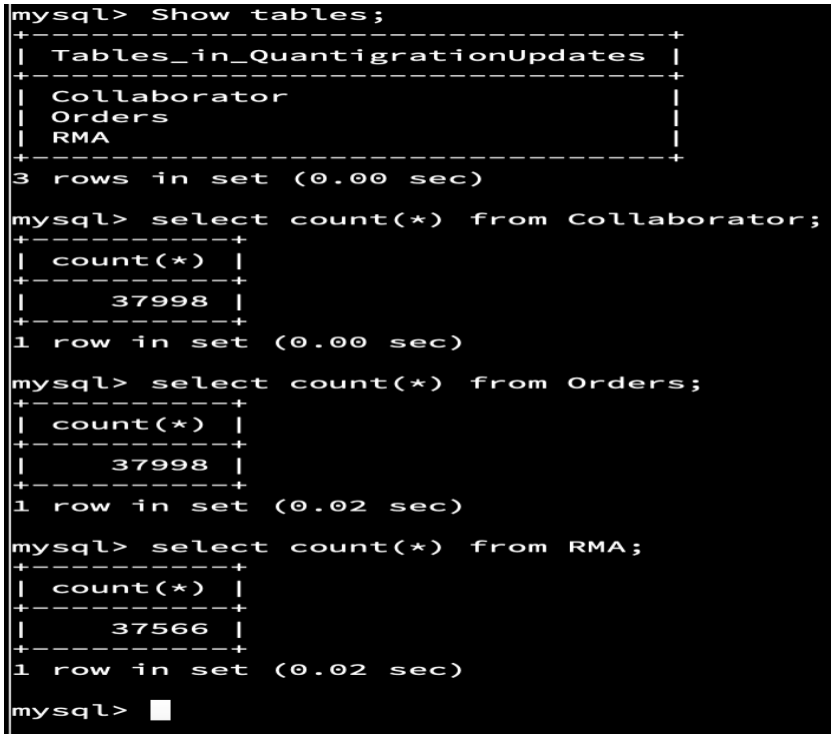
Professor Lockwood

DAD-220 Intro to Struct Database Env

18 February 2024

7-1 Project Two: Analyzing Databases

1. Begin by writing SQL commands to **capture usable data** (which you've preloaded into Codio) for your analysis.



```
mysql> Show tables;
+-----------------------------------+
| Tables_in_QuantigrationUpdates    |
+-----------------------------------+
| Collaborator                      |
| Orders                            |
| RMA                               |
+-----------------------------------+
3 rows in set (0.00 sec)

mysql> select count(*) from Collaborator;
+----------+
| count(*) |
+----------+
|    37998 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from Orders;
+----------+
| count(*) |
+----------+
|    37998 |
+----------+
1 row in set (0.02 sec)

mysql> select count(*) from RMA;
+----------+
| count(*) |
+----------+
|    37566 |
+----------+
1 row in set (0.02 sec)

mysql>
```

Code used: Show tables;

Select count(*) from Collaborator;

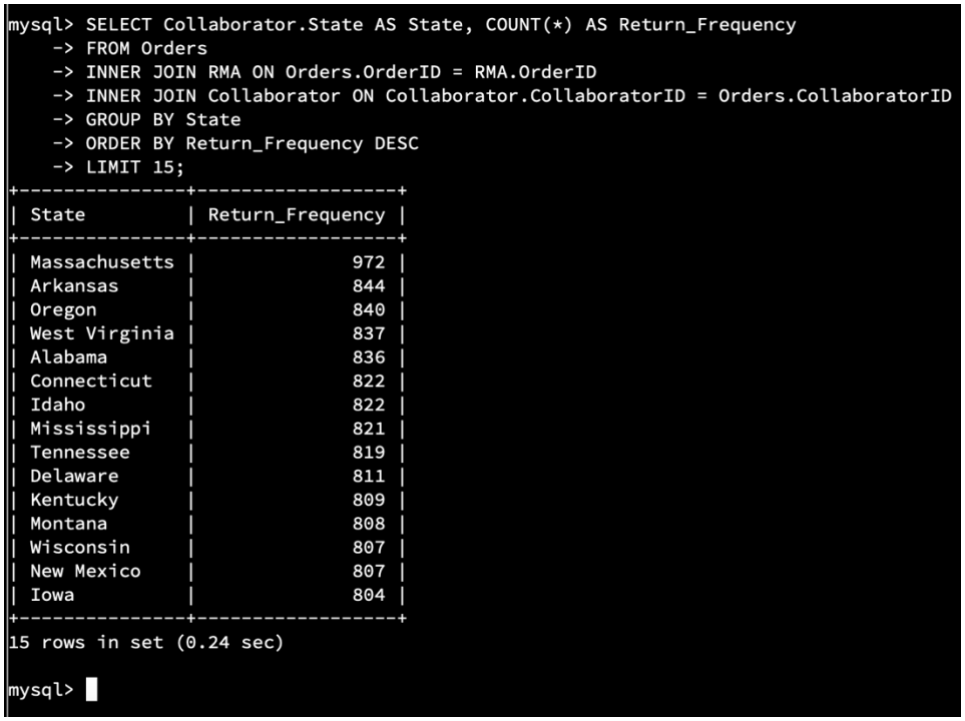
Select count(*) from Orders;

Select count(*) from RMA;

The screenshot above shows I have accessed the QuantigrationUpdates Database from the last module. I shown the tables I will be using and the amount of rows in each table.

2. Specifically, the product manager wants you to analyze the following:

   - **Analyze** the **number of returns by state** and describe your findings in your report.



```
mysql> SELECT Collaborator.State AS State, COUNT(*) AS Return_Frequency
    -> FROM Orders
    -> INNER JOIN RMA ON Orders.OrderID = RMA.OrderID
    -> INNER JOIN Collaborator ON Collaborator.CollaboratorID = Orders.CollaboratorID
    -> GROUP BY State
    -> ORDER BY Return_Frequency DESC
    -> LIMIT 15;
+---------------+------------------+
| State         | Return_Frequency |
+---------------+------------------+
| Massachusetts |              972 |
| Arkansas      |              844 |
| Oregon        |              840 |
| West Virginia |              837 |
| Alabama       |              836 |
| Connecticut   |              822 |
| Idaho         |              822 |
| Mississippi   |              821 |
| Tennessee     |              819 |
| Delaware      |              811 |
| Kentucky      |              809 |
| Montana       |              808 |
| Wisconsin     |              807 |
| New Mexico    |              807 |
| Iowa          |              804 |
+---------------+------------------+
15 rows in set (0.24 sec)

mysql>
```

Code used: SELECT Collaborator.State AS State, COUNT(*) AS Return_Frequency

-> FROM Orders

-> INNER JOIN RMA ON Orders.OrderID = RMA.OrderID

-> INNER JOIN Collaborator ON Collaborator.CollaboratorID = Orders.CollaboratorID

-> GROUP BY State

-> ORDER BY Return_Frequency DESC

-> LIMIT 15;

This screenshot show the command I used to show the frequency of returns by state. As you can see Massachusetts has the most returns at 972, followed by Arkansas with 844. The top five

states are the states that have the highest percentage of returns and will have a higher number of returns. The screenshot below shows the percentage by state.

```
mysql> SELECT Collaborator.State AS State,
    ->        (COUNT(*) * 100 / (SELECT COUNT(*) FROM Orders)) AS Return_Percentage
    -> FROM Orders
    -> INNER JOIN RMA ON Orders.OrderID = RMA.OrderID
    -> INNER JOIN Collaborator ON Collaborator.CollaboratorID = Orders.CollaboratorID
    -> GROUP BY State
    -> ORDER BY Return_Percentage DESC;
+---------------+-------------------+
| State         | Return_Percentage |
+---------------+-------------------+
| Massachusetts |            2.5580 |
| Arkansas      |            2.2212 |
| Oregon        |            2.2106 |
| West Virginia |            2.2027 |
| Alabama       |            2.2001 |
| Idaho         |            2.1633 |
| Connecticut   |            2.1633 |
| Mississippi   |            2.1606 |
| Tennessee     |            2.1554 |
```

- **Analyze** the **percentage of returns by product type** and describe your findings in your report.

Code used: SELECT Orders.SKU,
-> MAX(Orders.Description) AS Description,
-> (COUNT(*) * 100 / (SELECT COUNT(*) FROM Orders
-> INNER JOIN RMA ON Orders.OrderID = RMA.OrderID))
-> AS Return_Percent

```
mysql> SELECT Orders.SKU,
    ->        MAX(Orders.Description) AS Description,
    ->        (COUNT(*) * 100 / (SELECT COUNT(*) FROM Orders
    ->                           INNER JOIN RMA ON Orders.OrderID = RMA.OrderID))
    ->        AS Return_Percent
    -> FROM Orders
    -> INNER JOIN RMA ON Orders.OrderID = RMA.OrderID
    -> GROUP BY SKU, Description
    -> HAVING COUNT(*) > 1  -- Filter groups with only one order
    -> ORDER BY Return_Percent DESC;
+------------+-----------------------------------------------------+----------------+
| SKU        | Description                                         | Return_Percent |
+------------+-----------------------------------------------------+----------------+
|            |        22.0465 |/100/1000 BaseT 48 port             |                |
|            |        16.2860 |ch 40GigE SFP+ 48 port              |                |
|            |         11.4119 |h 10GigE SFP+ 48 port              |                |
|            |        11.3081 |0/100/1000 BaseT 8 port             |                |
|            |         11.2628 |h 10GigE SFP+ 24 Port              |                |
| ADV-48-10F | Advanced Switch 10 GigE Copper/Fiber 44 port coppe  |        10.9780 |
|            |         10.9727 |10GigE Copper 24 port              |                |
|            |         5.6461 |ch 40GigE SFP+ 24 port              |                |
|            |         0.0878 |/100/1000 BaseT 24 port             |                |
+------------+-----------------------------------------------------+----------------+
```

```
-> FROM Orders
-> INNER JOIN RMA ON Orders.OrderID = RMA.OrderID
-> GROUP BY SKU
-> ORDER BY Return_Percent DESC;
```

This screenshot shows the following command that returns The percentages of the type of product. I could not get this to work properly I even went as far as droping the database and starting from scratch on the whole database and still couldn't get it to work. This was the best I could get it and it shows the highest percentage is 22.8465, which is BAS-48-1-C. It goes all the way down to a percentage of 0.0878.

3. In your report, clearly **summarize your analysis of the data for stakeholders**. Include screenshots of the results of each query. When summarizing results, you may want to consider the following questions:

    - How does the data provide the product manager with usable information?

The data presented above shows the product manager with some of the breakdowns of where the most returns are coming in from by the state/location. It also gives some insight into what items are being returned the most by count and percentages. Further investigation will need to happen to exactly figure out the root cause of why the top item is returned so much.

    - What are the potential flaws in the data that has been presented?

The data does not necessarily determine the reason for the return of each item. The product manager will have to investigate the product further to determine why a certain product might have a higher rate of return compared to a product with a low rate of return. Additionally, this data only shows rates of returns for the company's products. This does not account for

competitor products and how our company compares to the competitor. This means there will need to be additional analysis of the competitor and how we stack up.

- Are there any limitations on your conclusions, or any other ways of looking at it that you haven't considered? Clearly communicate your findings to stakeholders.

This data provides much insight and gives the company a direction to look in for improvements but is limited in depth. While it is visible that products are being returned, we need to include information on why they are being returned. With further analysis of return reasons or positive reviews, the method for improvement will be more accurate.