Southern New Hampshire University

DAD 220 Module 7-1
Prepared on: 17 Apr., 2022
Prepared for: Prof. Aastha Agarwal
Prepared by: Alexander Ahmann

Contents

1 Introduction

Many for-profit companies rely on many empirical observations, expressed quantitatively, to make the optimal decisions that will lead to profit-maximisation. Our company, *Quantigration*, makes use of relational database systems to process copious volumes of such empirical data. In this report, I shall use the SQL programming language to analyse the *Quantigration* database in an effort to give business executives and managers useful information that will help them make better decisions.

2 Procedure

2.1 Statement of the data analysis problem

In a previous report of database engineering job (Ahmann, 2022), I have described the procedure that I employed to create a database for handling our customer in-

formation. Now, I will analyse customer information to assist business executives and product managers in making the optimal profit-maximising decisions. But before proceeding to analyse the customer information, I must define the problem.

What is required is for myself to analyse both the frequency of returns by state and the per cent of returns by product type. Using these summary statistics, I will then devise various conjectures that attempt to explain the causal agents or predictors that influence the resulting frequency of returns by state and the per cent of returns by product type. Finally, I will use these researches as a basis of further research and recommendations to the product managers, business executives and other concerned persons.

2.2 Overview of the database's schema

I have discussed in a writeup[1] the layout of `QuantigrationUpdates`, our company's database, with its E.R. diagram,[2] but I will briefly go over it in this subsection to "jog the memory" of anyone who read the report, or to inform anyone who did not get to read the writeup about the database's schema.

Fig 1 shows the `QuantigrationUpdates`' E.R. diagram. The `Collaborators`[3] table has a one-to-many relationship with the `Orders` table, and the `Orders` table has a one-to-many relationship with the `RMA` table. These one-to-many relationships are so that the `INNER JOIN`s can be performed between tables. The `Orders` table has a foreign key that references the `Collaborators` table's primary key, and the `RMA` table has a foreign key that references the `Orders` table's primary key.

`INNER JOIN`s allow an analyst to "merge" two or more tables together whilst retrieving or modifying table rows. With a discussion of the high-level visual representation of the tables dispensed with, I can now proceed to perform statistical analysis on the data set in question.

2.3 Analysis of frequency of returns by state

An interesting feature to analyse would be the frequency of returns by state. The exact SQL statement that I used to work out the solution to this problem is:

```
SELECT Collaborators.State, COUNT(RMA.RMAID) AS "Freq."
  FROM Collaborators
  INNER JOIN Orders
```

[1] In Ahmann (2022).

[2] The entity-relationship diagram.

[3] This table was previously called `Customers`, but was renamed to `Collaborators`. Because of this, the E.R. diagram, will show `Collaborators` as `Customers`.

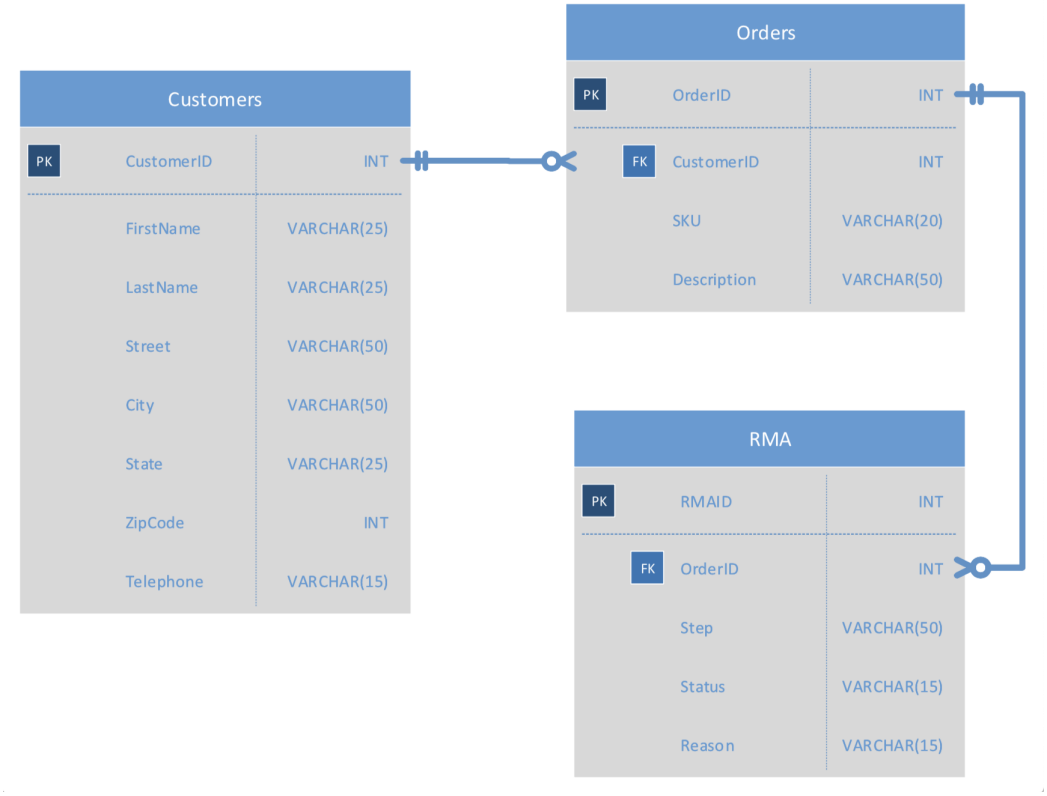

Figure 1: `QuantigrationUpdates`**' Entity-Relationship Diagram (Week 6 resources, n.d.).**

```
  ON Collaborators.CustomerID = Orders.CustomerID
INNER JOIN RMA
  ON Orders.OrderID = RMA.OrderID
GROUP BY Collaborators.State
ORDER BY COUNT(RMA.RMAID) DESC;
```

Fig. 2 demonstrates that my query was able to successfully run. It reports a total of 48 rows.

2.4 Analysis of per cent of returns by product type

Another interesting feature to analyse would be the per cent of returns by product type. The exact SQL statements that I used to work out the solution to this problem are:

```
mysql> SELECT Collaborators.State, COUNT(RMA.RMAID) AS "Freq."
    ->   FROM Collaborators
    ->   INNER JOIN Orders
    ->     ON Collaborators.CustomerID = Orders.CustomerID
    ->   INNER JOIN RMA
    ->     ON Orders.OrderID = RMA.OrderID
    ->   GROUP BY Collaborators.State
    ->   ORDER BY COUNT(RMA.RMAID) DESC;
+----------------+-------+
| State          | Freq. |
+----------------+-------+
| Massachusetts  |   988 |
| Arkansas       |   858 |
| West Virginia  |   851 |
| Oregon         |   850 |
| Alabama        |   845 |
| Idaho          |   839 |
| Connecticut    |   837 |
| Tennessee      |   835 |
| Mississippi    |   834 |
| Delaware       |   828 |
| Montana        |   825 |
| Kentucky       |   823 |
| Wisconsin      |   823 |
```

Figure 2: **An exerpt of the results of the first query.**

```
SELECT @total_entries := COUNT(*) FROM Orders
  INNER JOIN RMA ON Orders.OrderID = RMA.RMAID;
```

... and ...

```
SELECT Orders.SKU, COUNT(*) / @total_entries
    AS "%age return"
  FROM Orders
  INNER JOIN RMA
    ON Orders.OrderID = RMA.RMAID
  GROUP BY Orders.SKU
  ORDER BY COUNT(*) / @total_entries DESC;
```

Fig. 3 demonstrates that my query was able to successfully run. It reports a total of 9 rows.

2.5 "First impressions" of the analysis

Judging by the initial analysis, it seems that territories going near to the eastern part of the United States have the highest frequency of return rate and there does

```
mysql> SELECT Orders.SKU, COUNT(*) / @total_entries
    ->     AS "%age return"
    ->   FROM Orders
    ->   INNER JOIN RMA
    ->     ON Orders.OrderID = RMA.RMAID
    ->   GROUP BY Orders.SKU
    ->   ORDER BY COUNT(*) / @total_entries DESC;
+------------+-------------+
| SKU        | %age return |
+------------+-------------+
| BAS-48-1 C |      0.2259 |
| ENT-48-40F |      0.1620 |
| BAS-08-1 C |      0.1144 |
| ENT-48-10F |      0.1142 |
| ENT-24-10F |      0.1121 |
| ADV-24-10C |      0.1109 |
| ADV-48-10F |      0.1055 |
| ENT-24-40F |      0.0544 |
| BAS-24-1 C |      0.0005 |
+------------+-------------+
9 rows in set (0.07 sec)
```

Figure 3: **The results of the second query.**

not seem to be any practical differences regarding its North-South continuum. It may be "worth it" to allocate technicians to the states that have the highest frequency of returns to diagnose any problems. An initial look of the results of the SQL statement used to get per cent of returns by product reveals that there does not seem to be any significant difference between the prefix of the `SKU` product ID and the per cent of returns.

3 Summary

This report is detailing an initial exploratory data analysis of the data sets in question to work out both what territories in the United States experience the highest frequency of returns and per cents of returns by product type. It should be noted that this paper is merely "scratching the surface" in regards to what can be done with regards to further research. For example, A/B testing can be employed as something of an experiment to test out hypotheses' regarding what the causal agents of product returns are. This report merely demonstrates the usefulness of relational databases and the SQL programming language in an initial exploratory data analysis and further research will be needed.

References

Ahmann, A. (2022). *Module 6-1 Project*. DAD-220.

Week 6 resources (n.d.). *Quantigration E.R. diagram*. DAD-220.

Week 6 resources (n.d.). *Project One Guidelines and Rubric*. DAD-220.