CS375-Databases and Information Retrieval

Project Report (Due: Dec. 13, 2022)

Group Members: Andrew Petrie, Onimoe Wilfred Chidera, Raj Wadia

Student Number(s):200389252, 200438059,200449347

Email: arp032@uregina.ca, cwo181@uregina.ca, rmw462@uregina.ca

Abstract

In the vast information space, where storing and retrieving data has become a vital part of how we communicate. Databases play the role of storage containers where all that data can be stored in an organized manner and accurately and efficiently accessed. However, there are certain planning procedures like conceptual schemas and building ER Diagrams and normalization rules that have to be followed to achieve data integrity. Our objective for this project is to follow these planning procedures and apply normalization rules when building the database for a typical eCommerce website with basic features.

Introduction

When we first learned that there would be a group project for this class we immediately sprang into action, brainstorming ideas which could display the knowledge we had obtained from this course. It was our group member Fred who suggested we try our hand at creating an eCommerce database, which would have all the required entities and relationships needed to adequately challenge us and test our new found abilities to create, maintain and deploy a large database. From conceptual schema to construction we successfully created a database we can say we’re proud of.

We learned a lot along the way and as a result we’ve reinforced our understanding of how databases work in the real world as well as picked up new skills and techniques that can be applied to the rest of our lives as we continue our careers in the fields of computer science.

Literature Review

For the literature used during development of this database we mostly referred to the course material. Looking over the chapters posted to urcourses and reviewing lecture notes, if necessary, we also looked at the textbook in cases were we needed a further in depth analysis of certain topics or ideas. If we ran into any situations in which there were no answers readily available in the previously mentioned literature, we referred to stack overflow and github for directions in which we could apply to our work and reach a solvable solution.

Problem Statement

Our problem was to create an authentic DBMS for a large eCommerce website such as amazon or ebay. This problem was chosen specifically for the significant challenges and hurdles it would pose to us and thus allowing our ability to create a functioning, efficient, and well maintained database to shine through. We stand by what we have created and believe we have achieved all the goals we set out for ourselves.

Methodology

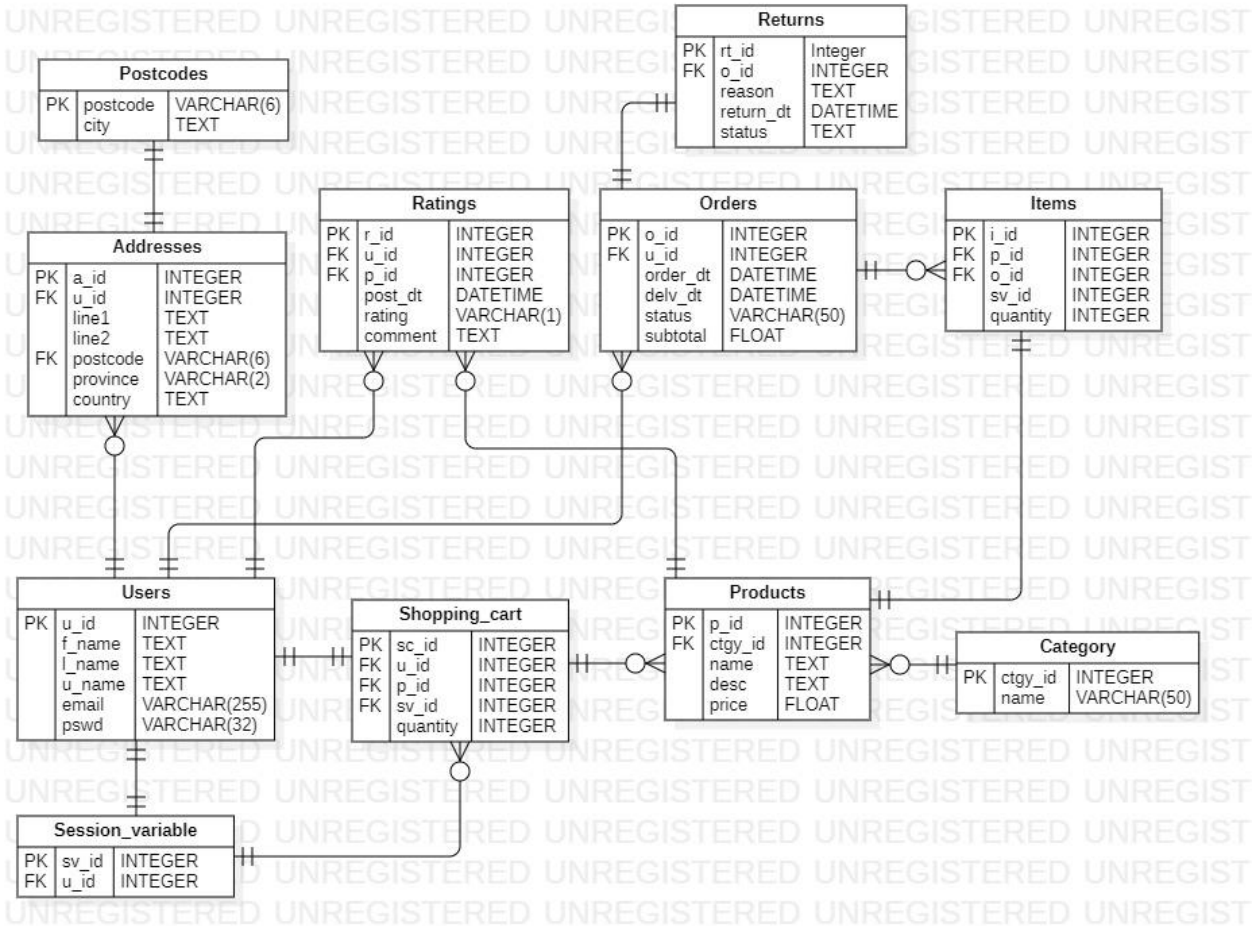
Required Analysis

To get started with building an actual database, we did an in-depth analysis on data required by the stakeholder(s), customers in our case. With that analysis we were successfully able to create a conceptual schema. Another part of the analysis was to select the appropriate tools and software with which we could host and run the

database. We chose MySQL as a querying language because we had prior experience working with it. There were several options available to host the database. Out of those options, we particularly chose Amazon Web Service (AWS) RDS to host MySQL server.

ER Diagram

From our analysis on data requirements, we produced an Entity Relation Diagram that acts as a schematic for our database. It also describes the type of relationship between two entities. Overall, it is a blueprint for our database.



Normalization

It is the process used to organize and structure the data in a database. It helps reduce the amount of duplicates across the tables in the database and reduces the redundancy and complexity of the database. For this database we enforced rules

required for the table to be in 3NF. Particularly, the Addresses table, previously, had postcode and city as attributes. However, due to transitive relationship between a_id -> postcode -> city, we could achieve 3NF. We create the Postcodes as a separate entity where postcode is a primary key and city as an attribute.

Result

All of our hard work and dedication resulted in a fully functioning DBMS that we can proudly stand by. All of this effort and time spent creating this database has helped further develop and absorb the information taught in class and as a result we are a far more capable and trained group of data analysts and computer scientists. This culmination of all our efforts has given us valuable insight and expertise in the quickly growing field of databases and information retrieval. We are entirely grateful for what we learned and how we've expanded our skillset as preparation for future projects and careers involving the use of databases.

Physically wise we have created an expansive DBMS with many queries as well as a front end for querying the database and retrieving information from it. It took many iterations and many trials and errors to accomplish all of our goals but the end result has been worth the effort.

IMG 1.1: Query of average product rating.

```
1
2     -- get avg ratings of all product
3     select avg(rating) as average_product_rating
4     from Ratings R
5
6     ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

average_product_rating
3.7500

IMG 1.2: Query of best selling product by category.

```
1     -- best selling product from each category
2     select S.ctg_name as "Category Name", Product_name as "Product Name", max(qty_sold) as "Quantity sold"
3
4     from
5     (select C.name as ctg_name , P.name as Product_name, sum(quantity) as qty_sold
6     from Products P
7     join Category C on P.ctgy_id = C.ctgy_id
8     join Items I on I.p_id = P.p_id
9     group by C.ctgy_id) S
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Category Name	Product Name	Quantity sold
Electronics	TV	8
Books	The Atomic Habits	2

IMG 1.3: Query of customer with highest orders.

```
1
2   -- Customer with highest orders
3   select S.FirstName as "First Name", LastName as "Last Name", max(NumberOfOrders) as "HigestOrderQty"
4
5   from
6   (select U.f_name as FirstName, U.l_name as LastName, count(o_id) as NumberOfOrders
7   from Orders O
8   join Users U on O.u_id = U.u_id
9   group by U.f_name) S
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

First Name	Last Name	HigestOrderQty
Fred	Onimoe	2

Conclusion

In conclusion, we have implemented a DBMS to solve the problem of customer data for eCommerce giants like Amazon or Ebay. In the industry, there is a high demand for DBMS systems that are reliable, recover from loss and quick. This prompted us to focus on efficiency, correctness and maintainability features while creating the DBMS. MySQL and AWS are amongst the software tools we used for the implementation. The end result is a good working database and an interface that can facilitate database interaction. Creating this project has given us hands-on experience of working in a team with different people and starting a project from design phase to deployment. This will no doubt give us an advantage when we graduate from school.

Further Work

To go above and beyond the requirements for this project we decided to include a simple front end website to aid in the visualization of how the database entities and their relationship works which can be found here: https://ruswal.github.io/Website/index.html . We also decided to include some additional procedures that were taught in class to demonstrate our capabilities and our understanding of the course material, such as, including a crash recovery system.

We are far from having completed a fully functioning amazon clone, however we proudly stand by the work we have accomplished and if given the time and resources we feel confident we could finish a completely functioning prototype based on the work we have done thus far.

References

E-commerce database design example:
https://fabric.inc/blog/ecommerce-database-design-example/

Class textbook:
R. Ramakrishnan and J. Gehrke, Database Management Systems, 3rd edition, McGraw Hill, 2003.

SQL command referencing:
https://www.w3schools.com/sql/