**Module 8: Portfolio Project Option 2**

Joshua Humphreys

Colorado State University Global

Course Code:MIS407

Dr. Mario Missakian

October 10, 2021

**Preface**

I selected option 2 for the portfolio project. In the following pages I will present the steps I took in completing this assignment following the database development lifecycle (Mannino, 2019). I have included all the relevant SQL code as well as screenshots from PostGreSQL demonstrating successful completion of all project requirements. I have also included a metadata file in my submission. Following the development steps, I have included a brief discussion of my key learnings from this assignment.

**Preliminary Investigation**

The goal of this assignment was to develop a transactional database for a hardware store. The hardware store provides a tool rental service. In recent times the hardware store has tracked their tool rentals using a spreadsheet. However, due to recent growth, the company has found the spreadsheet ineffective. The hardware store rents a large variety of tools for different applications. The store purchases their tools from a few different suppliers. The store wants to be able to track their inventory of tools and catalog them based on various characteristics. Additionally, the store would like to keep a record of their customers and transactions. One specific and important detail of the store’s business model is that a customer is only allowed to possess up to three tools at a time.

**Systems Analysis**

Based on my systems analysis, I produced the following Entity Relationship diagram.

**Figure 1.**

*Entity Relationship Diagram for Hardware Store Database*

A picture containing letter

Description automatically generated

**Systems Design**

During the design phase I did notice a few business rules that would have created some redundancy in the database. Namely the business assigned account numbers to customers and tool numbers to each of the tools. In the design and implementation of the database it was necessary to identify primary keys for the customer and tool entities. My recommendation to the business would have been to stop using the old numbers to track those entities and use the primary keys that the DBMS will assign to those entities. By using the surrogate keys assigned by the DBMS as account numbers and tool ids we can ensure that those numbers are unique and not null. This may present an issue with migrating the old data to the new transactional database. Certainly, this would be a topic of discussion during database development. However, for the purposes of this assignment, I assumed the business would agree with me that it is best to just eliminate the old numbers and use the keys provided by the DBMS.

**Systems Implementation**

The following is the DDL SQL code I used in PostgreSQL to build the database. The script that I used to create the tables came directly from the documentation included with PostgreSQL provided by The PostgreSQL Global Development Group (2021). The most challenging piece of implementing the database came with creating the trigger and function that enforce the max rental business rule. Abdelhalim (2020) provided some excellent examples of how to implement triggers. Between her work and the documentation provided by The PostgreSQL Global Development Group (2021) I was able to successfully implement the trigger. One of the central components necessary to implementing the trigger function was the COUNT(\*) function included in PostgreSQL. I found the tutorial provided by PostgreSQL Tutorial (2021) very helpful in understanding how to implement the function in my trigger function.

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Figure 2.**

*Screenshot of successful database creation*

Graphical user interface, text, application

Description automatically generated

The following is the DML SQL script I used to populate the database.

Text

Description automatically generated

Table

Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence

**Figure 3.**

*Screenshot demonstrating successful creation of trigger to limit rentals*

Graphical user interface, text, application, Word

Description automatically generated

**Figure 4.**

*Screenshot demonstrating successful execution of SQL script to display contents of all tables*

A screenshot of a computer

Description automatically generated

**Figure 5.**

*Screenshot demonstrating successful execution of SQL script to Retrieve all the customers' names, account numbers, and addresses (street and zip code only), sorted by account number*

A screenshot of a computer

Description automatically generated

**Figure 6.**

*Screenshot demonstrating successful execution of SQL script* *to retrieve all the tools rented in the last 30 days and sort in chronological rental date order*

Graphical user interface, text, application

Description automatically generated

**Figure 7.**

*Screenshot demonstrating successful execution of SQL script* *to update a customer name to change their maiden name to married name*

Graphical user interface, text, application, email

Description automatically generated

**Figure 8.**

*Screenshot demonstrating successful execution of SQL script to* *delete a customer from the database*

A screenshot of a computer

Description automatically generated

**Conclusion**

In Mannino’s (2019) discussion of the information systems development lifecycle he states, “for most systems, the boundary between phases is blurred and there is considerable backtracking between phases” (p. 27). Throughout the completion of this project, that fact became very apparent. I initially set out with an ERD that was overly complicated and difficult to implement. I found myself going back to my ERD and adjusting the design to make implementation more practical. I also found myself reviewing the business rules consistently and conducting analysis throughout the process in order to ensure the database met the objectives the company set forth. Ultimately, my biggest takeaway from this project is the incredible power of relational databases and the PostgreSQL DBMS.

**References**

Abdelhalim, A. (2020). *Interesting 5 Functions and 5 Triggers With PostgreSQL.* Analyics Vidhya. <https://medium.com/analytics-vidhya/interesting-5-functions-and-5-triggers-with-postgresql-52700d554192>

Juba, S., and Volkov, A. (2017*). Learning PostgreSQL 10: A beginner’s guide to building high-performance PostgreSQL database solutions* (2nd ed.). Packt Publishing.

Mannino, M.V. (2019). *Database Design, Application Development, and Administration* (7th ed.). Chicago Business Press.

The PostgreSQL Global Development Group. (2021). *PostgreSQL 10.18 Documentation*

PostgreSQL Tutorial.(2021). *PostgreSQL COUNT Function.* Postgresql Tutorial. https://www.postgresqltutorial.com/postgresql-count-function/