# **Grocery Store Management System: Data Storage and Optimization**

Lewis University  
CPSC 50900: Database Systems   
Spring 2025 Term Project

BY

Prajwal Palabandla

[PrajwalPalabandla@lewisu.edu](mailto:PrajwalPalabandla@lewisu.edu)

GitHub Link:

Table of Contents

[A. Title Page 1](#_gjdgxs)

[Schedule of Milestones 1](#_30j0zll)

[B. Initial Proposal 2](#_1fob9te)

[C. Data Sources 2](#_3znysh7)

[D. Alternative Ways to Store the Data 3](#_tyjcwt)

[E. Conceptual and Logical Models 3](#_3dy6vkm)

[F. Physical Model 4](#_4d34og8)

[G. Populate the database with data 5](#_2s8eyo1)

[H. Data Manipulation Language (DML) Scripts 5](#_17dp8vu)

[I. Indexes 6](#_3rdcrjn)

[J. Views 6](#_26in1rg)

[K. Stored Programs (Stored Procedures, Stored Functions, Triggers) 6](#_lnxbz9)

[L. Transactions 7](#_35nkun2)

[M. Database Security 7](#_1ksv4uv)

[N. Locking and Concurrent Access 7](#_44sinio)

[O. Backing Up Your Database 8](#_2jxsxqh)

[P. Programming 8](#_z337ya)

[Q. Suggested Future Work 8](#_3j2qqm3)

[R. Activity Log 9](#_1y810tw)

The remaining sections – Triggers, Locking and Concurrency, Backup, and Programming, will be turned in with the final report, which is due March 16 at 11:59pm.

# 

# **B. Initial Proposal**

***Description: You will describe the data you aim to store. What data will be storing? Why are you interested in this data? Why is it important? Where will the data come from? Who will use this data? What kind of application do you plan to build with it?***

***Rubric: Your response to each of these six questions will be graded out of 3 points.***

* ***3 points: clear, complete descriptions that convey the importance and meaning of your data***
* ***2 points: mostly clear descriptions, although some additional data would have helped in some sections***
* ***1 point: necessary details are lacking in many of your responses.***

***You will also earn 2 additional points for coming up with a descriptive title for your project.***

***As you consider various ideas for your project, keep in mind that your database is going to have to store data for at least 8 different types of things. Each of these different “types of things” will become a table in the database you design and build. So, the idea can’t be so narrow that you can’t identify at least eight different types of things in it that you’d store data about.***

***Total points possible: 20***

The database contains multiple tables for organizing grocery-related data which allows effective data management together with easy accessibility. The system contains essential store data points which include product specifications (names with categories and prices and inventory counts), supplier business information and orders (IDs with item lists and amounts alongside total prices), employee records (IDs with duties and schedules) and financial transactions (methods with amounts and timestamps) and current item stock counts.

The data serves as a necessary foundation for grocery store operational management on a daily basis. The demand for precise tracking of numerous activities and inventory levels remains paramount in the grocery industry because effective management requires it. Researching this data enables us to boost operational efficiency while decreasing superfluous stock and inadequate inventories which leads to better customer satisfaction.

The business needs exact data at all times to perform stock management and execute order fulfillment processes although both require up-to-date data for accurate sales forecasting. The store enhances product rotation and offers popular items through effective monitoring of inventory and sales patterns which results in greater customer satisfaction.

The research data will originate from multiple sources that include supermarket transaction reports as well as supplier information systems and historical customer buying records. Users will access Kaggle's supermarket sales data as their initial testing sample to fill the database for development purposes.

The same data material will be used by different groups of stakeholders. Through the data stores will monitor their inventory stock and conduct sales assessments to ensure smooth order processing. Cashiers need to utilize purchase information from customers to perform transactions. A mobile application provides customers access to personalized data which includes product availability and order status. The store obtains inventory status updates from its suppliers based on their monitoring activities.

A complete grocery management application stands as the primary focus of development. The system incorporates three main functions including Point of Sale (POS) for dealing with transactions combined with inventory tracking capabilities which maintain stock levels and also includes customer analytics functionality that delivers personalized suggestions and promotional offers. Real-time data analytics along with decision-making based on data will maximize the store's efficiency and deliver better experiences to customers through this application.

# **C. Data Sources**

***Description: Gather your data in text files. The text files may be csv, tab-delimited, xml, json, or some other custom format. Not all the files need be of the same type. Identify what each file contains by indicating where it came from, explaining in detail how it is structured, and describing how you will reorganize the data into a relational database. Post your data files to your GitHub repository, and provide samples of the data in your Word doc.***

***Rubric: Your work will be graded as follows:***

* ***5 points: you gathered multiple data files that contain the data that will populate your databases. If you do not use multiple data files, you will not receive credit.***
* ***5 points: you described the contents of the data files in detail, including referencing their origin and explaining how they were structured.***
* ***3 points: you identify which fields you plan to include in your database, including their data types and any constraints you expect to impose on the data or steps you'll have to take to clean up the data.***
* ***2 points: you post the data files to your GitHub account and make it possible for me to see them.***

[*https://www.kaggle.com/code/bibeksubedi11/supermarket-data-visualization/input?select=supermarket\_sales+-+Sheet1.csv*](https://www.kaggle.com/code/bibeksubedi11/supermarket-data-visualization/input?select=supermarket_sales+-+Sheet1.csv)

*Total points possible: 15*

*The primary dataset is sourced from Kaggle’s supermarket sales data. It is a* CSV file *containing structured information on grocery store transactions.*

**Contents**

* **Product Details:** Product ID, category, unit price, and stock levels
* **Customer Information:** Customer ID, gender, payment type
* **Sales Transactions:** Date, time, product sold, quantity, price, discount
* **Payment Details:** Payment method, total amount paid
* **Employee Data:** Cashier handling the transaction

**Database Organization:**

* The CSV file will be cleaned and transformed into relational tables.
* Tables will be structured as **Products, Customers, Orders, Employees, Suppliers, and Payments**.
* Fields will be normalized to reduce redundancy and improve performance.
* Data types will include **VARCHAR (product names), INT (IDs, quantities), and DECIMAL (prices)**.

# D. Alternative Ways to Store the Data

*Description: We will study alternatives to storing data in a relational database. Some of the alternatives come from several decades ago, including the hierarchical and network models. Some are newer options, such as NoSQL databases that use JSON or some other encoding. Describe in detail how to store the data using two alternatives to relational databases. Be sure to describe how you would implement the alternatives and the advantages and disadvantages of each.*

*Rubric: Your work will be graded as follows*

* *5 points for clearly describing how your data could be stored using one alternative to relational databases and what the advantages and disadvantages of that approach would be.*
* *5 points for clearly describing how your data could be stored using another alternative to relational databases and what the advantages and disadvantages of that approach would be.*

*Total points possible: 10*

**NoSQL Database (Document-Based – MongoDB)**

Implementation:

The JSON-like documents serve as data storage units within MongoDB which operates as a document-based NoSQL system instead of using relational table frameworks. The application will establish different collections which contain information about Products Orders Customers , Employees and Payments. All relevant information will be stored regarding each document through nested documentation. An Order document maintains all information about customers together with their payment details along with purchased products inside one single record thus simplifying complex join operations.

**Advantages:**

The database model follows flexible schema design instead of relational tables because MongoDB enables adjustable data structure modifications through business growth.

MongoDB enables businesses to scale horizontally thus becoming a prime choice for dealing with big datasets with high transactions.

**Disadvantages:**

The limited join capabilities of MongoDB distinguish it from SQL databases since complex data relationships present challenges in management.

Data duplication through denormalization presents a disadvantage since MongoDB uses more storage space than relational databases.

**Hierarchical Database (XML-Based Storage)**

**Implementation:**

A hierarchical database arranges its data in XML format and establishes parent-child relationships within it. The database hierarchy for Products represents multiple Items at its child level while Orders maintain their Purchases as child entities.

**Advantages:**

The system delivers superior fast query capabilities as well as efficient batch processing requirements.

The hierarchical database model suits predefined data models because of its capability to handle structured relationships between data elements.

**Disadvantages:**

The structured parent-child relationship of the database presents challenges when you need to change schemas since it maintains a rigid structure.

The retrieval of particular data elements demands complete XML trees parsing when running complex queries.

ENTER YOUR ALTERNATIVE DATA STORAGE IDEAS HERE

# E. Conceptual and Logical Models

*Description: First, come up with a conceptual model. The conceptual model identifies the entity sets and the relationships among them. For each relationship, identify the connectivity and the participation (optional or mandatory).*

*Now that you know the entity sets, the next step is to develop the logical model by adding attributes. For each entity set, identify the attributes that describe the entity set. This may include references to other entity sets that are involved in relationships. Then, identify the functional dependencies that exist among them. For each functional dependency, identify the determinants and the fields they determine, like this:*

*determinant, or, determinants 🡪 attributes, they, determine*

*This becomes the basis for identifying your entity sets, which will become your tables when we move to the physical model in the next section. The attributes listed on the left of the arrows are candidates to become your primary key attributes. Attributes that are references to other entity sets are candidates to become the foreign keys.*

*For entity sets that have multi-attribute determinants, replace them with surrogate keys. This makes it easier to identify each entity in the set and to define foreign keys.*

*Then apply normalization to make sure that your design satisfies First, Second, and Third Normal forms. For 1st Normal Form, make sure that all attributes are indivisible. This may require adding an entity set that lists values that appear in comma-separated lists as individual entities. For 2nd Normal Form, make sure there are no partial dependencies (this won’t be a problem if all your entity sets have single-attribute determinants). Finally, make sure all your entity sets are in 3rd Normal Form. This means that you have to split transitive dependencies into separate entity sets and add relationships between the original entity set and the new ones.*

*Finally, draw the logical model as an ERD. At this point, your design will have entity sets, their relationships, and their attributes. M:N relationships are acceptable at this point, as we’ll remove them in the physical model.*

*Rubric: Your work will be graded as follows:*

* *5 points for identifying all entity sets*
* *5 points for writing each relationship between entity sets as two sentences and correctly identifying their connectivity and participation.*
* *5 points for adding attributes to entity sets and writing the functional dependencies correctly. Replace multi-attribute determinants with surrogate keys.*
* *4 points for performing the normalization steps. Make sure your design is in 3rd Normal Form.*
* *5 points for drawing the ERD for the logical model. At this point, the ERD will show entity sets, relationships, attributes, and primary identifiers. The design may include M:N relationships at this point. We’ll get rid of those in the physical model.*

*Total points possible: 24*

ENTER YOUR RELATIONAL DATABASE DESIGN DESCRIPTION HERE. INCLUDE A PICTURE OF YOUR ERD.

# F. Physical Model

*Description: This is where you will complete your database design. Add data types, including size constraints, uniqueness constraints, and auto-incrementing for all attributes. Implement relationships using foreign keys. Replace many-to-many relationships with two one-to-many relationships using bridge entity sets. Add additional entity sets that you think could be helpful for storing the acceptable values of particular attributes. (For example, if you were storing student data, valid student statuses might include Good Standing, Graduated, On Probation, Expelled. Put those in a table and create a relationship back to the student table). Draw the ERD for the physical model.*

*Using the final ERD, write the SQL DDL statements needed to create the database, its tables, and the relationships among them. Run these statements in MySQL to build your database. Provide screen shots that show the database you built in MySQL, including its tables and descriptions of some of the tables. To show a list of databases and a list of the tables in a particular database, use the show command. To see a description for a table, use the describe command.*

*Rubric: Your work will be graded as follows:*

* *3 points for introducing bridge entity sets (if necessary)*
* *3 points for adding data types and other constraints on the data.*
* *3 points for introducing other entity sets and their relationships that help enforce what values can be assigned to particular attributes (if necessary)*
* *5 points for drawing the ERD for the physical model. If you used Vertabelo, the resulting ERD must be free of errors and warnings*
* *6 points for generating the SQL scripts that build the database and then running the script in mysql. Demonstrate that the script built the database and its tables with screenshots that show that you ran the show and describe commands.*

*You will be penalized 4 points if your database doesn’t have at least 8 appropriately defined tables.*

*Total points possible: 20*

DESCRIBE THE STEPS YOU TOOK TO COMPLETE THE PHYSICAL MODEL. THEN SHOW THE ERD FOR THE PHYSICAL MODEL. THEN SHOW THE SQL COMMANDS THAT BUILD THE DATABASE. (FOR THIS, YOU MAY REFER TO A PARTICULAR FILE IN YOUR GITHUB REPOSITORY. MAKE SURE YOU INVITE ME AS A COLLABORATOR ON YOUR REPOSITORY SO THAT I CAN ACCESS THE SCRIPT.) FINALLY, SHOW SCREEN SHOTS THAT PROVE THAT YOU BUILT THE DATABASE AND ITS STRUCTURES IN MYSQL.

# G. Populate the database with data

*Description: You built the database in section F, and it now exists in mysql. Now populate it with your data. Take your original data source or sources and generate insert statements from them. Store the insert statements in a text file, and then use the mysql source command to run these insert statements to populate the various table structures. Generating the necessary insert statements may require writing Python scripts or manipulating Excel databases to convert the data from your original data sources.*

*Rubric: Your work will be grades as follows:*

* *Explain step-by-step and very clearly how you created the required SQL statements from your initial data. Write it as a set of instructions. 5 points*
* *Show the file of insert statements that you ran in MySQL. You may do this either by including the listing in this report or by identifying the file in your GitHub that contains the insert statements. Make sure I have access to your GitHub repository. 4 points*
* *Show screenshots of the data in your MySQL database. To do this, run select statements for each table and show screen shots of what is displayed: 5 points*

*Total points possible: 14*

ENTER YOUR DDL WORK HERE

# H. Data Manipulation Language (DML) Scripts

*Description: Write the SQL commands for twelve queries. Two queries should be insert statements, two should update statements, one should be a delete statement, one should be a simple select statement that selects a subset of the rows and columns from one table, two should be a select statements that select data from a joining of two tables, two should use summary functions to generate statistics about the data, one should be a multi-table query, and one should be another query of your choice. Show the queries and screenshots of the results in your Word document, and save your queries in a commented sql script to GitHub.*

*Rubric: Your work will be graded as follows:*

* *1 point each for the two insert statements*
* *1 point each for the two update statements*
* *1 point for the delete statement*
* *1 point for the simple select statement*
* *2 points each for the 2 join statements*
* *2 points each for the two that use summary statements*
* *2 points for the multi-table query*
* *2 points for the query of your choice.*
* *6 points for showing the query and a screenshot of the corresponding result set back-to-back for each of these queries in your Word document.*

*Total points possible: 24*

ENTER DML WORK HERE

# I. Indexes

*Description: Improve the performance of your design by adding indexes to various tables. Show the SQL needed to add the indexes. Explain why you chose the ones you added. Explain how you would demonstrate the impact the indexes had on the performance of various queries.*

*Rubric: Your work will be graded as follows:*

* *3 points for clearly defining at least three indexes and explaining why you chose them.*
* *3 points for showing the sql needed to generate the indexes*
* *2 points for explaining how you would demonstrate the performance improvement afforded by the indexes.*

*Total points possible: 8*

ENTER YOUR INDEX WORK HERE

# J. Views

*Description: Add two views to your database to provide easy access to combinations of data from multiple tables.*

*Rubric: Your work will be graded as follows:*

* *3 points for including the SQL for generating the two views in your Word document*
* *3 points for including screenshots for the data contained in each view in your Word document*
* *3 points for explaining why each view is a valuable addition to your database*

*Total points possible: 9*

ENTER YOUR WORK WITH VIEWS HERE

# K. Stored Programs (Stored Procedures, Stored Functions, Triggers)

*Description: Add a stored procedure, stored function or trigger to a table and demonstrate using it.*

*Rubric: Your work will be graded as follows:*

* *3 points for including the SQL for the stored program (procedure, function, or trigger in your Word document*
* *3 points for clearly explaining the purpose of the stored program*
* *3 points for a screenshot and explanation that shows the stored program in action.*

*Total points possible: 9*

ENTER YOUR WORK WITH STORED PROGRAMS HERE

# L. Transactions

*Description: Demonstrate that you know how to define and use a transaction. Why are transactions important for ensuring ACID behavior?*

*Rubric: Your work will be graded as follows:*

* *5 points for clearly explaining the importance of transactions to ensuring ACID behavior*
* *3 points for including a screenshot and accompanying explanation of a MySQL transaction.*

*Total points possible: 8*

ENTER YOUR WORK WITH TRANSACTIONS HERE

# M. Database Security

*Description: Identify the different kinds of users who will use your database. Write GRANT statements to define the privileges for these different kinds of users.*

*Rubric: Your work will be graded as follows:*

* *4 points for clearly identifying and describing the various kinds of users who will use the databases and identifying and justifying what privileges each should have.*
* *4 points for writing GRANT statements that assign privileges to these different kinds of users.*
* *4 points for demonstrating with screenshots that your GRANT statements do distinguish among different kinds of users in regard to what they can do with the database.*

*Total points possible: 12*

ENTER YOUR WORK WITH DATABASE SECURITY HERE

# N. Locking and Concurrent Access

*Description: Explain the purpose of locking tables and show how to do that to prevent inconsistencies that may arise in your data when concurrent transactions take place.*

*Rubric: Your work will be graded as follows:*

* *3 points for clearly explaining an example that shows why you should lock tables to prevent inconsistencies.*
* *3 points for providing a screenshot and accompanying explanation of locking tables.*

*Total points possible: 5*

ENTER YOUR WORK WITH LOCKING AND CONCURRENT ACCESS HERE

# O. Backing Up Your Database

*Description: How you will back up your database. What commands will you issue? How frequently will the commands run? How can they be automated? Where will the backups be stored?*

*Rubric: Your work will be graded as follows:*

* *6 points for clearly explaining and justifying your database backup strategy, including the frequency with which you will back up the database, how you will automate backups, where you will store them, and how you will secure them. You will earn three points for addressing each factor (frequency, location, automation, and security)*
* *2 points for providing a screenshot of the command you would issue to back up the database and for including a portion of the resulting file.*

*Total points possible: 8*

ENTER YOUR WORK ON DATABASE BACKUPS HERE

# P. Programming

*Description: Write a Python, Java, or PHP program that generates a report that contains a subset of the data from your database. Include the code for your Python program in your Word document, and also post the program to your GitHub repository.*

*Rubric: Your work will be graded as follows:*

* *10 points for writing a Python script (and including its code in the Word doc) that will pull data from a database and store it to a text file and present it to the screen. Your code must have comments in it that explain how it works. You will be awarded 3 points for successfully connecting to the database, 3 points for successfully querying it, and 4 points for presenting the data to the screen and to a file. Internal comments count for 2 points.*
* *2 points for posting the code to GitHub*
* *6 points for showing a screenshot of your running the script and showing the results it produces on the screen.*

*Total points possible: 18*

ENTER YOUR PYTHON, PHP, or JAVA DATABASE PROGRAMMING WORK HERE

# Q. Suggested Future Work

*Description: Describe the limitations of your current database and explain how you or someone else could improve the design to address these shortcomings. Also describe how you might take advantage of leverage cloud services to increase the performance and availability of your database. Finally, explain the advantages and disadvantages of storing your data in a NoSQL format instead.*

*Rubric: Your work will be graded as follows:*

* *3 points for clearly describing the limitations of your databases*
* *3 points for explaining how you would address these shortcomings*
* *3 points for explaining how you might migrate the database to the cloud and describing what advantages you might gain from doing that.*
* *3 points for explaining the advantages and disadvantages of storing your data in a document-based NoSQL format instead.*

*Total points possible: 12*

ENTER YOUR SUGGESTED FUTURE WORK IDEAS HERE

# R. Activity Log

*Description: As an appendix, the team will keep a frequently updated diary or log of their activity. What did you or your team study in this class each day? What did you learn? What did you accomplish or build or design? You don't have to enter something every day, but there should be at least three entries each week. Since we have eight weeks, that means you should make 3 posts to the Activity Log each week, for a total of at least 24 posts. Each post will be worth 1 point.*

*If you are working as part of a team, make sure you clearly identify which team member worked on which tasks. The Activity Log should help me figure out how each team member contributed to the project. If I cannot discern who worked on what aspects of the project from the activity log, no points will be awarded for it.*

*Total points possible: 24*

*The* ***Activity Log*** *will serve as a detailed record of the team’s progress throughout the project, ensuring accountability and tracking milestones. Each entry will document the* ***date, task completed, and the team member(s) responsible****. Below is an example of how activities will be logged over the course of the project.*

***Date Activity Description***

***Week 1***

Day 1 Defined the project scope and finalized data sources.

Day 3 Researched relational database structures for grocery store data.

Reviewed the Kaggle supermarket sales dataset and analyzed key fields.

Week 4  Identified eight core database tables: Products, Orders, Customers, Employees, Payments, Inventory, Suppliers, and Categories.

Week 6 Planned database schema and relationships among tables.

**Week 2**

Day 1 *Gathered the supermarket sales dataset and cleaned the data.*

Day 2 *Checked for missing values and removed inconsistencies in the dataset.*

Day 3 Transformed the dataset into a relational format.

Day 4 Created CSV and JSON versions of the dataset for different storage approaches.

Day 5 Uploaded cleaned data files to GitHub for project tracking.

Day 6 Started preliminary database implementation in MySQL.

Day 7 Conducted initial tests for data integrity and relationships.