# 

**Lewis University  
CPSC 50900: Database Systems   
Spring 2024 Term Project**

**Student name: Sai Naren Reddy Chittireddy**

**INSTRUCTOR: Raymond Klump**

**project title: E-commerce database**

**Github link:**

Table of Contents

[A. Title Page 1](#_gjdgxs)

[B. Initial Proposal 2](#_30j0zll)

[C. Data Sources 3](#_1fob9te)

[D. Alternative Ways to Store the Data 3](#_2et92p0)

[E. Relational Database Design Process 4](#_tyjcwt)

[F. Relational Database Design 4](#_1t3h5sf)

[G. Data Definition Language (DDL) Scripts 5](#_4d34og8)

[H. Data Manipulation Language Scripts 5](#_2s8eyo1)

[I. Indexes 6](#_17dp8vu)

[J. Views 6](#_3rdcrjn)

[K. Triggers 6](#_26in1rg)

[L. Transactions 7](#_lnxbz9)

[M. Database Security 7](#_35nkun2)

[N. Locking and Concurrent Access 7](#_1ksv4uv)

[O. Backing Up Your Database 7](#_44sinio)

[P. Programming 8](#_2jxsxqh)

[Q. Suggested Future Work 8](#_z337ya)

[R. Activity Log 9](#_3j2qqm3)

# Schedule of Milestones

Here is a schedule that shows when each milestone is due and what sections comprise it.

| Deadline | Sections for which you must demonstrate significant progress |
| --- | --- |
| January 29 at 11:59pm | a. Title page  b. Initial proposal  c. Data sources  d. Alternative ways to store the data  r. Activity Log – at least six entries covering the first two weeks |
| February 12  at 11:59pm | e. Relational database design process  f. Relational database design  g. Data definition language scripts  h. Data manipulation language scripts  r. Activity Log – at least six entries covering the past two weeks |
| February 26 at 11:59pm | i. Indexes  j. Views  l. Transactions  m. Security  n. Locking  o. Backup  r. Activity Log – at least six entries covering the past two weeks |

# B. Initial Proposal

Through our e-commerce database I will store different product data like accessories, bags, beauty, housewares, jewelry, kid’s items, shoes, men’s apparel, and women apparel. This data is important for online shopping, comparison, authorizing efficient product searches, and purchase. Here sources include manufactures, user-generated content, and suppliers. The users include administrators, analysts, and customers aligned from individual shoppers to business exploits in our platform for retail operations. I aim to create a robust e-commerce application for seamless browsing, various product management, serving customer requirements, and purchasing. Our application provides a user friendly interface and ensures to manage the data over multiple product categories.

# C. Data Sources

The dataset discusses NewChic.com's product catalog and the popularity of items from August 2020. It defines what people love and how they rate items. A large dataset is accessible through the Telegram bot for updates and custom queries. This is suitable for researchers to see top products, trends, and understand user preferences. Here data are collected from CSV and JSON files where it compromised product details and popularity metrics. Each of the CSV files includes product attributes like ID, name, category, description, and price. On the other hand JSON files include popularity metrics like purchase, rating, and views.

I collect this data straight from NewChic.com, setting product details and popularity as of early August 2020. The data covers user interactions and ratings, facilitating research on product trends, user tastes, and segmentation. At the same time it provides business opportunities such as optimizing stock and identifying top-selling categories. Perfect for searching what people like and making understanding business decisions.

i collect data from [*https://data.world/jfreex/products-catalog-from-newchiccom/*](https://data.world/jfreex/products-catalog-from-newchiccom/)

# D. Alternative Ways to Store the Data

NoSQL Database with JSON Encoding: Here data is stored using a NoSQL Database like a document-oriented model MongoDB. Each of the product categories is documented and organized with JSON. It allows to accommodate diverse data structures, scalable storage, and flexibility. There are some advantages to using this data storage like easy to grow, support for semi-structured data, and quick development cycles. However, it becomes complex when there is limited support for complex transactions and schema flexibility.

Graph Database for Product Relationships: A graph database Neo4j store data by representing products as nodes and their relationships. This model efficiently represents complex interconnections in product popularity and user tastes. Advantages include fast query performance for relationship-based queries. However, a learning curve and potential inefficiency for simple, tabular data storage compared to traditional databases.

# E. Relational Database Design Process

*Description: Consider the list of fields you identified in part c. Identify functional dependencies that exist among them. For each functional dependency, identify the determinants and the fields they determine. This becomes the basis for identifying your entity sets, which then become your tables. Give each entity set or table you identify in this way a unique and clear name, making sure that the names you use are singular nouns. Then list the relationships that exist among the various entity sets. For each relationship, identify its connectivity (one-to-one, one-to-many, many-to-many) and participation (optional or mandatory). Finally, make sure that none of the attributes you've assigned to each entity set are multi-valued. If they are, take the steps needed to break them down.*

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

* *8 points for identifying all the functional dependencies, including determinants and the columns whose values they determine.*
* *2 points for naming the entity sets that make up your data with clear, easy-to-understand names.*
* *6 points for identify the relationships among the entity sets and identifying connectivity and participation for each.*
* *2 points for breaking down multi-valued attributes.*

*Total points possible: 18*

ENTER YOUR RELATIONAL DATABASE DESIGN DESCRIPTION HERE. INCLUDE SOURCE CODE AND SCREEN SHOTS.

# F. Relational Database Design

*Description: This is where you will complete your database design. For each of the entity sets you identified in the preceding section, analyze them to make sure they pass 2nd, 3rd, 4th, and Boyce-Codd Normal Form. If they do not, introduce additional entity sets or key changes to make sure that they do. Then, add foreign keys to connect entity sets that are related. For many-to-many relationships, introduce bridge entity sets to convert them into two one-to-many relationships. Also, consider whether you should introduce surrogate keys to create a more efficient primary key for some of your entity sets. Finally, diagram your design in Vertabelo. Make sure your ER diagram correctly shows all entity sets, their primary and foreign keys, the data types for each attribute, and the connectivity and participation characteristics of each entity set. Your final Vertabelo design should be something you could actually implement in a relational database management system.*

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

* *4 points for the normalization analysis of your entity sets.*
* *3 points for introducing bridge entity sets.*
* *3 points for choosing foreign keys and perhaps more efficient surrogate keys*
* *10 points for correctly depicting your physical database model in Vertabello*

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

*Total points possible: 20*

ENTER YOUR RELATIONAL DATABASE DESIGN HERE

# G. Data Definition Language (DDL) Scripts

*Description: Use Vertabello to generate a script of SQL commands that build the database and its table structures. Write scripts or build Excel spreadsheets that take your data files and generate scripts of SQL insert statements from them. Use the MySQL source command to run the various scripts needed to build and populate the database in MySQL. Include the source code and / or Excel spreadsheets you use to manipulate and populate the data. Make sure all your tables have at least three records in them and that you've linked the tables through their foreign keys.*

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

* *Database and table creation statements from Vertabelo saved as an sql script file: 3 points*
* *Scripts you write or Excel spreadsheets you create to generate SQL commands for populating the tables, uploaded to GitHub: 8 points*
* *Descriptions of the scripts and Excel spreadsheets you wrote along with code excerpts included in the Word document: 5 points*
* *Screenshots of your successful attempts to use the MySQL source command to populate each table with at least three records: 4 points*

*Total points possible: 20*

ENTER YOUR DDL WORK HERE

# H. Data Manipulation Language 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:*

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

*Total points possible: 6*

ENTER YOUR WORK WITH VIEWS HERE

# K. Triggers

*Description: Add a trigger to a table so that data will be updated when a certain event occurs*

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

* *2 points for including the SQL for the trigger in your Word document*
* *2 points for clearly explaining the purpose of the trigger*
* *2 points for a screenshot and explanation that shows the trigger in action.*

*Total points possible: 6*

ENTER YOUR WORK WITH TRIGGERS 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:*

* *3 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: 6*

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.*
* *2 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 I 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*

MAKE AT LEAST THREE ENTRIES PER WEEK. CLEARLY IDENTIFY WHAT EACH PERSON ON YOUR TEAM ACCOMPLISHED. YOU MUST SHARE THE RESPONSIBILITY OF COMPLETING THE PROJECT.

**Week1**

Day1:

Discussed project requirement and responsibility.

Research e-commerce data structures.

Research potential technology for database implementation.

Day3:

In meeting, share findings and decide on database schema.

Drafted the initial proposal document.

Explored security considerations for the database.

Day5

Work on the data fields for each product category.

Research best practices for user authentication.

**Week 2**

Day1:

Reviewed and finalized the database schema.

Explore the data migration strategies.

Day3:

Began database design and develop initial tables.

Investigate all potential API integrations.

Day5:

Discussed progress and addressed roadblocks.

Explore options for hosting databases.

Drafted project title.