# A. Medical device startup company operations database

Lewis University  
CPSC 50900: Database Systems   
Spring 2024 Term Project

Medical device startup company operations database.

Sharath Kumar Reddy Sashani, SharathKumarReddyS@lewisu.edu

Second Person, Second Person Email Address

Third Person, Third Person Email Address

Work products stored in the Github repository

https://github.com/sharath561/Medical-device-database-system.git

Table of Contents

[A. Title Page 1](#_Toc152230519)

[B. Initial Proposal 2](#_Toc139688952)

[C. Data Sources 3](#_Toc102302373)

[D. Alternative Ways to Store the Data 3](#_Toc253987590)

[E. Relational Database Design Process 4](#_Toc47888474)

[F. Relational Database Design 4](#_Toc1794510899)

[G. Data Definition Language (DDL) Scripts 5](#_Toc9532520)

[H. Data Manipulation Language Scripts 5](#_Toc2014337146)

[I. Indexes 6](#_Toc364229772)

[J. Views 6](#_Toc423795366)

[K. Triggers 6](#_Toc937775206)

[L. Transactions 7](#_Toc1336454402)

[M. Database Security 7](#_Toc1164301272)

[N. Locking and Concurrent Access 7](#_Toc1578907519)

[O. Backing Up Your Database 7](#_Toc1968323262)

[P. Programming 8](#_Toc723961217)

[Q. Suggested Future Work 8](#_Toc1323642763)

[R. Activity Log 9](#_Toc159721872)

# 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

*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.*

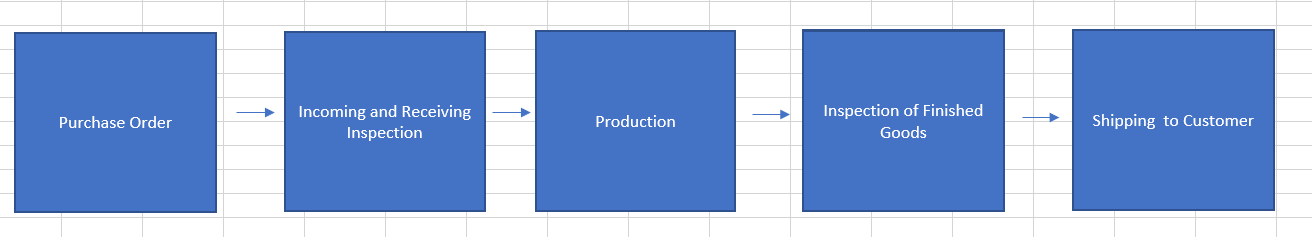
*As you consider various ideas for your project, keep in mind that your database is going to have to store data for at least 8 different types of things. Each of these different “types of things” will become a table in the database you design and build. So, the idea can’t be so narrow that you can’t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

*Total points possible: 20*

You will describe the data you aim to store. What data will be storing?

The purpose of this project is to implement a database system for a medical startup company, which will store the purchase order, vendor list , inventory list , and the analyzer part list , the cartridge part list, the quality list Etc. We are taking a hypothetical medical startup company which will create an analyzer and a cartridge. This cartridge will take the sample of the patient and will be inserted into analyzer and then this analyzer will process the cartridge and let the nurse know if the patient has covid or not. So, this project will focus mostly on the operations, Finance and the quality aspects of the company, the research and development and HR part is out of the scope for this project as all the R&D has been done and the final product which needs to be commercialized has been forwarded it to finance, quality and operations. The workflow of this project will start from a person who will place a purchase order, based on the lead time of the inventory. Once the purchase order has been placed and the item has been received it will go through the qualities and inspection to make sure the parts are conforming as per the quality and the FDA regulations. Parallelly we will also be tracking the work order of the amount of analyzer and cartridge that needs to be built. Once we have received the number of analyzer and cartridge to be built, that information will be sent to the operations and operations is going to start pulling the parts from the inventory. This database needs to be smart enough that the moment any new inventory comes in and the part out for making the devices is pulled, it needs to constantly update the values. Once all the analyzer and cartridge has been built and has been shipped the work order needs to be closed and the process needs to be repeated all over again.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purchase Orders | Quality | Production | Inventory Management | Shipping |
| Purchase Orders | Incoming/Receiving | Work Orders In/Out | Inventory Control Cycle/R&D/Scrap | Shipping |
|  | Finished Goods |  |  |  |
|  | MRB |  |  |  |



Why are you interested in this data?

We are trying to implement a whole database system for the medical startup company from scratch to final deliverable and push it to the production level. We will get the understanding of both database systems as well as the business operation side of running a company.

Why is it important?

It is important that we learn these technical and non-technical skills so that in the future when I'm employed, I can confidently say that the skills I acquired through this course and given any roles and responsibilities thrown at me, we can confidently finish the task or project. As we are also working in a team, we will learn a valuable skill of collaboration.

Where will the data come from?

As this is a hypothetical startup company so we will be creating the data from the scratch and also with the help of Google we can also look through some of the data which is available for other medical company or we can write a python script where it will automatically generate the data.

Who will use this data?

As we are implementing the databases system for the company, any person belonging to the following department like finance, operations and quality and also who is involved with any of the steps like placing a purchase order, performing the quality or inspection on the parts, performing the quality on the finished goods, creating the work orders, Interacting to the vendors to get a quote, the operation people who is going to look into the inventory and realize how much inventory we have on hand and if we are running low on inventory then they can approach the concerned person to place the order for those particular parts.

What kind of application do you plan to build with it?

To create a database system where all the records of the concern departments like (finance, operations and quality) of a company will be stored. There will be a front end of database system where people can either query the database and update the values or insert the value into the database. This application is work in progress as we are working through the semester, this part will be updated accordingly.

# 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.*

*Total points possible: 15*

Identifying what each file contains by indicating where it came from,

Currently we have total number of 6 excel files. so the files are purchase order, production workbook, quality , inventory control , shipping. So all these five workbook will be connected to the final workbook called Inventory management system and these will be connected via formula called get pivot table and anytime we will be updating any one of these CSV files and then if we refresh the final Inventory management system file all the values will be refreshed and updated.

explaining in detail how it is structured, and

So each of these five CSV files will serve a purpose of either quality, shipping production purchase order and inventory control and all of the values will be corresponding to the functions of each operations . And as all of these values are either numeric or floating are characteristics .

Describing how you will reorganize the data into a relational database.

We can create different tables based on different CSV data and all the columns can be used as an attribute and the corresponding data type can be extracted from the column name, also by creating the primary key, and foreign key we can build the entity-relationship model logically, and further implement physically.

Post your data files to your GitHub repository, and provide samples of the data in your Word doc.

Purchase order

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Name** | **Vendor** | **NanoDx PO #** | **NanoDx PO Issue Date** | **Item #** | **Quantity Ordered** | **Price per part** | **Total Price of Parts** | **Total PO Price** |
| 1/29/2024 | xyz | (Approved) Abcam | PO-0001 | 1/29/2024 | AS-00001 | 1 | $1.00 | $1.00 | $1.00 |

Quality

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Name** | **Item #** | **Quantity Passed Inspection** | **Quantity Issued to MRB** |
| 1/29/2024 | xyz | AS-00006 | 100 | 0 |

Inventory Control

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Receipt Date** | **Purchase Order Number** | **Vendor/Supplier** | **Vendor/Supplier Part Number** | **Description** | **Item #** | **Lot Number** | **Expiration Date** | **Total Item Quantity Received** | **Receiving Status** |
| xyz | 1/30/2024 | PO-0001 | Abcam | x-007 | This is sample | AS-00006 | 123456 | 1/1/2030 | 100 | Ready for QC |

Production

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Open Work Order Date** | **Closed Work Order Date** | **Name** | **Work Order #** | **Description** | **Item #** | **Quantity Being Built** | **Status** |
| 1/1/2024 | 1/30/2024 | First Name | WO-0001 | Packaged 20X Box Cartridge TBI | AS-00082 | 100 | WIP |

Shipping

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Name** | **Customer** | **Sales Order #** | **Packing List #** | **Item #** | **Quantity Shipped** |
| 1/30/2024 | First Name | University Medical Center of El Paso (AKA UMC of El Paso) | | 123456 | AS-00045 | 100 |

Inventory Management System

Analyzer Part List

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Vendor Part #** | **Item #** | **Description** | **Vendor** | **Location** | **Unit of Measure** | **Available Quantity** | **Total Inventory** | **Quantity Waiting Inspection** | **Quantity On Receiving Hold** | **Quantity on Production Hold** | **Quantity On Order** | **MRB Quantity** | **RMA Quantity** | **Scrapped Quantity** | **R&D Issued Quantity** | **Total Price of Parts** |
| VP-1111112 | AS-00006 | Wire Harness, PCB to POGO |  | 2 | each | 18 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $ 529.74 |
| VP-1111113 | AS-00016 | Analog Front End - AFE, Analyzer |  | 2 | each | 23 | 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $ 6,877.00 |
| VP-1111114 | AS-00018 | Test Card, PCB |  | 2 | each | 15 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $ 187.50 |

Cartridge Part List

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Item #** | **Description** | **Total Inventory** | **Quantity Waiting Inspection** | **Quantity On Receiving Hold** | **Quantity on Production Quality Hold** | **Quantity On Order** | **MRB Quantity** | **RMA Quantity** | **Scrapped Quantity** | **R&D Issued Quantity** |
| AS-00069 | Anti-GFAP, EDC activated | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| AS-00074 | Panel Subassembly, PMOS Cartridge | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| AS-00078 | Midcard Fluidics, PMOS Cartridge | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Part Quantity

|  |  |  |  |
| --- | --- | --- | --- |
| **Material** | **Part Number** | **Qty Per Build** | **WIP/Finished Goods** |
| Wire Harness, PCB to POGO | AS-00006 | 1 | 0 |
| PCBA, AFE | AS-00016 | 1 | 0 |
| PCB, ETC | AS-00018 | 2 | 0 |
| PCB-A, Processor Board | AS-00019 | 1 | 0 |

# 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*

*Alternative 1: NoSQL Database with JSON Encoding*

*Description: NoSQL databases, particularly those employing JSON (JavaScript Object Notation) encoding, have gained popularity for their flexibility and scalability compared to traditional relational databases. Here's how data could be stored using a NoSQL database with JSON encoding:*

*Implementation:*

1. ***Data Model****: In a NoSQL database with JSON encoding, data is typically stored in a document-oriented manner. Each record is represented as a JSON document, which can contain nested structures and arrays.*
2. ***Schema Flexibility****: Unlike relational databases, NoSQL databases don't enforce a rigid schema. JSON documents can vary in structure, allowing for dynamic changes without altering the entire database schema.*
3. ***Querying****: NoSQL databases usually offer query capabilities through specialized query languages or APIs. Queries can target specific fields within JSON documents, but complex joins and transactions may be limited compared to relational databases.*
4. ***Scalability****: NoSQL databases excel at horizontal scalability. They can distribute data across multiple nodes, making them suitable for handling large volumes of data and high traffic loads.*

*Advantages:*

* ***Schema Flexibility****: Changes to data structures can be accommodated easily without database schema modifications.*
* ***Scalability****: NoSQL databases are designed for distributed environments, allowing for seamless scaling as data grows.*
* ***Performance****: With simpler data models and horizontal scalability, NoSQL databases can offer better performance for certain types of applications, especially those requiring high throughput and low latency.*

*Disadvantages:*

* ***Lack of ACID Transactions****: NoSQL databases often sacrifice ACID (Atomicity, Consistency, Isolation, Durability) properties for scalability and performance. This can lead to eventual consistency and data integrity challenges.*
* ***Limited Querying Capabilities****: Querying in NoSQL databases may not be as powerful or flexible as SQL in relational databases, especially for complex queries involving multiple data entities.*
* ***Learning Curve****: Developers accustomed to relational databases may face a learning curve when transitioning to NoSQL databases and JSON-based data modeling.*

*Alternative 2: Hierarchical Database Model*

*Description: The hierarchical database model organizes data in a tree-like structure with parent-child relationships. Each record contains information about its parent record, forming a hierarchy. Though less common today, hierarchical databases were widely used in early database systems.*

*Implementation:*

1. ***Tree Structure****: Data in a hierarchical database is organized in a tree structure, where each record has one parent and zero or more children.*
2. ***Parent-Child Relationships****: Records are linked through parent-child relationships. Retrieving data typically involves navigating the tree structure from parent to child or vice versa.*
3. ***Indexed Access****: Access to data is often indexed for efficient retrieval. However, traversing the hierarchy can become complex, especially for deeply nested data.*
4. ***Data Integrity****: Hierarchical databases enforce referential integrity by maintaining strict parent-child relationships.*

*Advantages:*

* ***Efficient for Certain Data Structures****: For data with natural hierarchical relationships, such as organizational structures or file systems, hierarchical databases can offer efficient storage and retrieval.*
* ***Data Integrity****: Hierarchical databases enforce strict parent-child relationships, reducing the risk of orphaned records or inconsistent data.*
* ***Indexing for Speed****: Indexed access can make data retrieval relatively fast, especially when accessing records by key.*

*Disadvantages:*

* ***Lack of Flexibility****: Hierarchical databases are rigid in structure, making it challenging to accommodate changes in data organization or relationships.*
* ***Limited Querying Capabilities****: Querying hierarchical databases often involves navigating the tree structure, which can be cumbersome for complex queries.*
* ***Scalability Issues****: Hierarchical databases may struggle to scale with growing data and evolving application requirements, especially compared to modern NoSQL or relational databases.*

*In conclusion, both NoSQL databases with JSON encoding and the hierarchical database model offer distinct approaches to storing data outside the realm of traditional relational databases. The choice between them depends on factors such as data structure, scalability requirements, and the complexity of querying and maintaining the database.*

# 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 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*

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.