

## Final Project (35 Points)

**Due: Thursday 11th December, 11:59 pm**

The final project is the highlight of this course. Since software development is usually a team effort, you can work with **three or four classmates** on this project. Everyone in the group is expected to contribute equally to the design and implementation.

### **Idea: Creating a Business Database from Scratch**

**Project overview:** A business owner approached you for help in creating a new database from scratch to streamline their company's operations. The existing business data is stored in an Excel file, which can be downloaded from Canvas. The dataset contains 31 columns and 1,000 rows. You need to design a robust and customized database solution tailored to their specific needs. For this project, you will use MySQL (MySQL Workbench) as the Database Management System (DBMS).

**Final output:** You must submit a complete **report** that clearly documents all phases of the project via **Canvas**. Please follow the guidelines below:

#### **a) Report Requirements**

1. **Explanation of Each Phase:** Provide a detailed description of what was done in each phase of the project.
2. **Screenshots:** Include a screenshot of the relevant output for **each phase**.
3. **Team Contribution Table:** Add a **table** in your report that clearly shows **which parts** of the project were completed by **you and your teammate**.

**Note:** You will be **graded based on this table**, so make sure it's clear and accurate.

#### **b) Files to Upload (Alongside the Report)**

1. **SQL Output Files:** Save the output of each phase in **.sql format**.
2. **Exported Code Outputs:** Export any outputs generated by your code (e.g., query results) as **separate files**. Use **.csv format**, if applicable.
3. **File Naming:** Name each exported file **clearly and appropriately**. Example file names: db\_view.sql — SQL script file, db\_view.csv — Exported result of the query.
4. **info.txt File:** Create an info.txt file that contains clear instructions on how to run your project from scratch. This should include: software requirements (e.g., MySQL version), setup steps (e.g., how to import the database), how to run the queries or scripts and any dependencies or configurations needed.

**Hint and specifications:** After analyzing the data structure and identifying the necessary actions to assist the business, you should have outlined the process into the following steps:

- **Phase1: Database Creation & Data Loading (5 points):** Before creating the database, the data in the Excel file should be thoroughly reviewed, cleaned, and then saved in CSV format. Then, the next step is to create a database and import the data into it. This can be done in two ways:
  1. Using the Table Import Wizard in MySQL Workbench (or another SQL tool with a similar import feature) to import the data.
  2. Using the LOAD DATA INFILE command in MySQL Shell to load the data into a table.To use the second method, start by cleaning the data in the Excel file and saving it as a CSV file. The cleaning process includes:
  1. Ensuring date formats are in “yyyy-mm-dd.”
  2. Verifying that the correct number format is used, with no currency symbols, commas, or other symbols.
- **Phase2: Normalization and Denormalization (5 points):** After examining the imported data, you'll notice that it is denormalized. This leads to the next step: Data Normalization. Normalization involves organizing the data according to specific attributes within a larger data model. The primary goals of database normalization are to remove redundant data, reduce the risk of data modification errors, and simplify the querying process. The SQL technique you can use to normalize your data is the CTAS (CREATE TABLE AS SELECT) method.
- **Phase3: Database Diagram Design (ER) (5 points):** The next step is to create an ER (Entity-Relationship) Diagram. ER diagrams utilize symbols to represent entities, attributes, and relationships, effectively visualizing how the entities in the database are interconnected. You can create your entity relationship diagram any way you'd like but allow me to suggest a few!
  - If you're the pencil-and-paper type, you can draw your diagram, take a picture, and upload it as a separate file.
  - If you're digitally inclined, you can use any software that helps you draw the types of shapes you'd like to draw to create and export diagrams.
  - Or you can use Reverse Engineer feature of MySQL to create the ER Diagram.

- **Phase4: Table Alterations (5 points):** After creating the ER Diagram and defining the table constraints, you need to modify the tables to align with the constraints outlined in the diagram. This section will focus on table alterations.
- **Phase5: Views, Triggers, and Stored Procedures (5 points):** At this stage, the database for the business should be set up, and the business owner has three requests:
  - Create a view that shows the quantity sold and revenue generated by each employee.
  - Implement a trigger on the products table that automatically reduces the stock quantity after an order is placed.
  - Develop a procedure to check for products that need to be restocked and generate a list of those requiring replenishment.
- **Phase6: User Management and Privileges (5 points):** The business owner has requested the creation of two users—a Database Administrator and a Data Analyst—with specific privileges on the database. You need to create two users and grant them appropriate access to the database. The first user, "AlexSmith" (or any other name), is a DBA and should be given full database administrator privileges. The second user, "JamieLee" (or any other name), is an Analyst and should be assigned read-only access. Designing a database also includes managing users and their permissions, which can be done using the SQL commands GRANT and REVOKE.
- **Phase7: Database Backup (5 points):** The main goal of backing up a database is to create a duplicate of its data and structure at a specific moment in time. This process involves taking a snapshot of the entire database or selected parts of it and securely storing the copy. Some of the locations you can use to backup your database are:
  - On-premises Servers
  - External Hard Drives or USB Drives
  - Network Attached Storage (NAS)
  - Cloud Storage Services: Storing backups in cloud-based storage solutions like Amazon S3, Google Cloud Storage, Microsoft Azure Blob Storage, or other similar services. Cloud storage offers scalability, accessibility, and disaster recovery benefits.
  - and so many other locations.

This database can be backed up to a hard drive using the MySQL local instance Data Export feature.

**Good luck with your project!**