

Western  Engineering

Electrical and Computer Engineering

# SE 3309 – Database Management Systems

## Assignment 04 Web Application

**Issued On:** November 22 (Wednesday), 2023

**Deadline:** December 6 (Wednesday), 2023

# Table of Contents

1. Overview:.....	1
2. Assignment Requirements.....	1
3. Evaluation Criteria.....	1
4. Deliverables.....	3
5. Conclusion.....	3

---

## 1. Overview:

---

In the final phase of your database project, you will be building a Web Interface to the database application you proposed and outlined at the beginning of the project (Assignment 1). While we do not expect every aspect of your design to be implemented, your application should resemble what you have proposed and contain its core functionality. Remember that each of the application functionality must rely on a database query.

---

## 2. Assignment Requirements

---

**Your program should consist of:**

1. **At least five alternative options** to the user. For each option there must be a way to validate it through the Web app. For example, if your option is to add a user, there must be a way from the Web app to view that the user has been added. If a function is used to validate the result of another option, it cannot be counted toward the five mandatory alternatives.
2. For each functionality, the user will navigate or click on the desired option:
  - a. The system will prompt the user for appropriate input values as required.
  - b. The system will access the database to perform the appropriate queries and/or modifications.
  - c. The application will display the results or an appropriate acknowledgment to the user.

---

## 3. Evaluation Criteria

---

This assignment will be marked based on the:

**1. Functionalities provided by the application:**

The functionalities provided must include **both modifications** and **queries**. Each provided functionality must be **unique**. If any are too similar they will be marked as one. Some of your SQL statements **must include joins**. All but one query **must exhibit some interesting features of SQL**: for example, queries over more than one relation, subqueries, EXISTS or GROUP BY.

For example, if your application was about car rentals you might offer options such as:

- a. Look up the price of a given car model **at the highest rated outlet**.
- b. Find the outlet with the **lowest price** for a given car model.
- c. Given a customer, find all the other customers that rent cars and have **at least one outlet in common**.
- d. Add a new car model to the car relation.
- e. Increase all the prices at a given outlet by a given amount.

## 2. The interface:

We are not expecting anything fancy in the way of interface; however the interface should be intuitive, functional and stable. The handling of SQL errors can be quite simple, but they must not impact the operation of the application.

## 3. Other technical consideration relevant to a simple database application.

Issues falling into this category are varied, but are based upon common sense and basic database design, as taught in class. Code should have a logical structure so that it can be easily determined what modules perform what functions.

Below are some examples of what would be considered common sense:

- Your functions should be logical and useful in some situation.
- Data used in the demo make sense in the context of the application.
- The tables presented are different from each other.
- Primary and Foreign Keys are present and appropriate.
- Table fields only accept the proper values or data types.
- Unique fields aren't updateable and don't allow duplicate values.

### Marking (out of 10):

The design of the web application, including but not limited to quality of the user interface and its usability.	2 points
The presence of 5 basic functionalities. This includes functionalities with limited complexity. If any are too similar they will be marked as one.	2.5 points
The presence of interesting features and demonstration of the overall complexity of the functionalities.	2.5 points
The validation of the user inputs and the appropriateness of the displayed results.	1.5 points
The suitability of the functionalities for the specific Web app, the cohesiveness and completeness of the application, and of the database design.	1.5 points

**All groups are required to prepare for a demo to show the 5+ functionalities implemented in the database application. It is your responsibility to showcase the complexity of your application.**

---

## 4. Deliverables

---

### HOW TO HAND THE ASSIGNMENT IN

- (Some provide queries in doc and some only in GitHub. Can we have one place to submit? ) Upload a single document on OWL containing the queries and the corresponding screenshots showing the response of the editor for each of your queries, AND
- Upload all your code to a GitHub classroom repository:
  - ✓ A GitHub classroom has been created for this class.
  - ✓ Make sure you are logged into GitHub, and then click on the link.
  - ✓ The invitation for assignment 4 is: <https://classroom.github.com/a/Joc2KsYq>
  - ✓ The link can also be found in the assignment description on OWL.
  - ✓ When you click on the link, you should see a list of uwo userid, select your userid from the list. If you are the first member to accept the assignment, you will have to create your group, (MUST add to select their group and how to create a group as they joined random) otherwise you can join the existing group. The group name used when creating a group should be based on the group number given on OWL.
  - ✓ Inside the assignment3 repository you will see SRC and DUMP folders;
  - ✓ In the src folder, you must upload the source code.
  - ✓ In the DUMP folder, you must upload the dump file of your database.

---

## 5. Conclusion

---

In this final phase of our project, we have successfully transitioned from theoretical planning to practical implementation, culminating in the creation of a dynamic and functional web interface for our database application. This stage was not just about realizing the designs proposed in the initial phase but also about ensuring that the core functionalities are efficiently integrated on precise database queries. The interface we developed, though simple, is intuitive and robust, emphasizing functionality and user experience. Each of the five unique functionalities implemented showcases our ability to effectively utilize complex SQL features like joins, subqueries, and group functions. The successful demonstration of these functionalities underscores the cohesiveness of our application and the soundness of our database design. This assignment has not only been a testament to our technical proficiency but also a reflection of our understanding of practical application needs, preparing us for future challenges in the field of database management and web application development.