Assignment 2 – Database Systems

Requirement: Use a DBMS such as MS SQL Server or MySQL.

- Preliminary report submission: (See sample "Preliminary Report – Assignment 2") in the 5th practical session.
- Final report submission: (See sample "Final Report – Assignment 2") in the 6th practical session.
- Deadline to submit report files on LMS: Saturday, December 6, 2025 (end of Week 49)

1. Create Tables and Sample Data (3 points)

1.1 (2 points)
Write SQL code to implement **ALL** database tables designed in Assignment 1, including:
- Primary key, foreign key constraints
- Data constraints
- Semantic constraints (using CHECK or TRIGGER)

Note: Constraints that can be checked in table creation statements must not be checked using triggers.
1.2 (1 point)
Create **meaningful sample data** for all tables, at least 5 rows per table (you may input manually or write insert statements).

2. Write Triggers, Stored Procedures, and Functions (4 points)

2.1 (1 point)
Write stored procedures to insert, update, and delete data from **ONE** table.
Requirements:
- Validation must be performed to ensure that the data table constraints are satisfied. The system should display meaningful error messages that clearly indicate the specific error (instead of a generic message like "Data entry error!"). For example: check that an employee's age is greater than 18, the phone number and email formats are valid, and an employee's salary is less than the manager's salary, etc.
- For DELETE procedure, specify when deletion is allowed or disallowed, and why it's necessary, what is its purpose?

2.2 (1 point)
Write TRIGGERS:

2.2.1 Identify one business constraint that requires a trigger to enforce.
- Determine which DML operations could violate the constraint.
- Write trigger(s) to check the constraint.

Note: Constraints that can be checked within the table creation statement should not be validated using triggers.

2.2.2 Choose one derived attribute and write a trigger to calculate its value.
- Determine which DML operations change this attribute.
- Write the trigger(s) to compute it.

Note: If a trigger calculates the value of attribute **A** using a derived attribute **B**, then **B** must be computed first!
For example, if a trigger automatically updates the store's total revenue based on the total value of its orders, the total order value must be calculated first. **Do not assume** that the total order value is already available.

GENERAL REQUIREMENT: Prepare SQL statements and sample data to demonstrate the trigger testing **during the presentation**.
2.3 (1 point)
Write **two stored procedures** that only include queries to display data. Each takes input parameters corresponding to WHERE and/or HAVING clauses.
Include:
- One query that retrieves data from **two or more tables** and includes **WHERE** and **ORDER BY** clauses.
- One query that uses an **aggregate function**, along with **GROUP BY**, **HAVING**, **WHERE**, and **ORDER BY** clauses, and joins **two or more tables**.
- Include at least **one stored procedure** related to retrieving data from the tables used in query 2.1.
- Prepare the **SQL statements and sample data** to demonstrate the stored procedure execution **during the presentation.**

2.4 (1 point)
Write two functions satisfying the following:
- Contain IF and/or LOOP statements for calculations on stored data
- Use cursors
- Include a query to retrieve data for computation
- Have input parameters and validate them

3. Application Implementation (3 points)

Develop an application (web, mobile, or desktop) to demonstrate database connectivity.

3.1 (1 point) Implement a screen that supports Insert / Update / Delete for the table in section 2.1.
3.2 (1 point) 1 interface that displays a data list retrieved from the stored procedure in item 2.3 and related to the table in item 2.1. Allows updating and deleting data from the list. In addition, it includes functions such as search, sorting, input data validation, logical error

handling when updating and deleting data, appropriate and specific error messages, proper use of controls, and a clear, user-friendly interface.

Example: 1 interface displaying a list of products, including search, filter, and sorting functions, with features to create a new product (calling the interface from part a) and to select a row to delete or update product information.

3.3 (1 point) Create an interface illustrating at least one other procedure (from 2.3) or function (from 2.4). (Can reuse the same interface as 3.2 if using the same table.)

GENERAL NOTES:

- Every student in the group MUST write at least **one statement** in part 2 (trigger, function, or procedure). Anyone who does not contribute will receive **NO score for Assignment 2 (BTL2)**.

Regarding SQL statements (Part 2):

- The score for each component (trigger, function, and procedure) will be evaluated based on its **complexity**, **completeness**, and **relevance** to the application's business logic.

Regarding the application (Part 3):

- The function that displays the data list must retrieve data by **calling the stored procedure** with **input parameters** entered by the user in the **WHERE** or **HAVING** clauses (for example, through text boxes, combo boxes, or calendar pickers). This corresponds to the typical **search function** found in applications or websites.
- The **insert, update, and delete** operations must be performed by **calling the stored procedures** defined in item 2.1.
- The application **MUST ACTUALLY CONNECT** to the database created in part 1. If not, the **application part will receive no points**.

Point Deductions:

- Functions, procedures, or triggers that have **similar or repetitive content**.
  Example: Procedure 1 displays a list of employees by name, and Procedure 2 displays a list of employees by employee ID, etc.
- **Insufficient or meaningless data** prepared for the presentation.
- A member **does not understand** the purpose and content of each function or procedure (even if it was not their part), or **cannot perform the required operations** as instructed by the instructor. This deduction applies **to individuals or the whole group**.
- If any member **does not participate** in the project, the other members are **responsible for notifying** the instructor to avoid group-wide point deductions.