VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

DATABASE SYSTEMS - CO2013

ASSIGNMENT 2

DATABASE MANAGEMENT SYSTEM

Author: BEng. Nguyen Minh Tam

HO CHI MINH CITY, 10/2025

ASSIGNMENT'S SPECIFICATION

Version 1.0

1 Assignment's Outcome

Upon completion of this assignment, students will be able to:

- Implement a relational database system using SQL.
- Define and enforce integrity constraints to maintain data consistency.
- Develop and execute stored procedures, functions, and triggers for automation and validation.
- Build a simple application that connects to and interacts with a relational database.
- Demonstrate integration between database programming and application logic.

2 Introduction

In this assignment, students will move from database design to database implementation. They will use SQL to create physical tables, enforce constraints, insert meaningful data, and develop database programs such as functions, stored procedures, and triggers. Finally, students will connect their database to a simple desktop or web-based application to demonstrate real interaction between an application layer and the database layer.

This assignment focuses on practical database programming skills that are essential in real-world database system development.

3 Assignment Overview

This assignment is worth **15%** of the overall course grade. All students must participate in all parts of the assignment. It consists of the following major parts:

- **Part 1: Create Database (3.5 points)**
- **Part 2: Store Procedures, Functions, Triggers (3 points)**
- **Part 3: Building Application (3.5 points)**

4 Part 1: Create Database (3.5 points)

4.1 Create Tables (2 points)

Students must write DDL statements to create all tables in the database based on the relational schema and business description from Assignment 1.

Notes:

- Identify appropriate data types, data lengths, and constraints.
- The database must include the following:
  - Primary Key and Auto Increment Primary Key.
  - Primary Key with given prefix and auto-incremental values (e.g., `EMP001, EMP002, ...`).
  - Foreign Key constraint.
  - Unique, NOT NULL, and DEFAULT constraints.
  - Row-based CHECK constraint related to one or two columns (e.g., `Dnumber` in [1, 20]; check-in datetime must be earlier than check-out datetime).

4.2 Insert Data (1.5 points)

Insert data into all tables created. The data should be **meaningful** and realistic.

Notes:

- At least $\frac{3}{4}$ of the tables must contain four or more rows.
- Prepare a SQL script that can be executed without errors to create and populate the complete database.

5 Part 2: Store Procedures, Functions, and Triggers (3 points)

5.1 Functions and Stored Procedures (2 points)

Students must implement at least **two functions** and **two stored procedures** that perform queries, calculations, or operations relevant to the business topic.

Notes:

- Each function/procedure must specify input and output parameters in the report.

- The set of procedures and functions must include examples of:
  - Queries with `WHERE` and `ORDER BY` clauses across two or more tables.
  - Input parameters used in `WHERE` or `HAVING` clauses.
  - Aggregate functions, `GROUP BY`, and `HAVING`.
  - Use of control statements such as `IF` or `FOR`.
  - Input parameter validation.

5.2 Triggers (1 point)

Write at least **two triggers** that automatically execute in response to `INSERT`, `UPDATE`, or `DELETE` statements.

Notes:

- At least one trigger must generate a derived column value.
- At least one trigger must enforce a meaningful business rule.
- *Example*: The salary of a department manager must be higher than that of any employee in the same department.

6 Part 3: Building Application (3.5 points)

Students will build an application that connects to the database from Part 1 and Part 2.

6.1 General Requirements

- Programming environment: optional (desktop, mobile or web application).
- Programming language: optional.
- The application must demonstrate data connection, display, and manipulation functionalities.
- **Students must use MySQL Server or MySQL as their DBMS.**

6.2 Create User (0.5 point)

Log in as DBA and create a user named `sManager`. Grant all access rights to this user.

6.3 Implement Features (3 points)

- **Login/Logout (0.5 point)** Allow users to log in and log out using the `sManager` account.
- **CRUD Operations (1 point)** View, insert, delete, and update data for an object (e.g., Employee, Product, etc.). *At least one screen must involve two or more tables.* Validate all input data and provide clear error messages.
- **Data Retrieval (1 point)** Retrieve a list of objects (e.g., list of products). Enable delete, detail view, or creation of new records from this screen. Implement filtering and sorting features.
- **Call Function/Procedure (0.5 point)** Demonstrate one feature that calls a stored procedure or function (e.g., monthly revenue calculation).

6.4 Bonus (1 point)

- Attractive and user-friendly interface (0.5 point)
- MVC architecture implemented for database communication (0.5 point)

7 Submission & Presentation

All teams are required to submit a report and present their work. The presentation will take place during Week 50 (exact date to be announced later).

Submission Guidelines:

- The team leader must submit one single `.zip` file to LMS.
- The file should contain:
  - **Report** (PDF) – including team member list and explanations as well as link to video demonstration.
  - **SQL Script** – used to create and populate the database.
  - **Source Code** – of the developed application.
  - **Supporting Files** (if any).

8 Regulations

- Students must read the requirements carefully before starting.
- Each student must participate in all parts of the assignment.
- The assignment is worth **15% of the total course grade.**

- Plagiarism will result in a **zero mark.**
- Use of appropriate software tools (DBMS and programming IDEs) is encouraged.

——————— END ———————