

DATABASE SYSTEMS

ASSIGNMENT 2

NOTES

- ❑ Students should read everything presented below carefully.
- ❑ This assignment 2 is worth **15%** of the overall grade.
- ❑ This assignment is on relational DBMS.
- ❑ Appropriate softwares can be used to support your implementation.
- ❑ **Plagiarism** must be avoided. Otherwise, **zero mark is given**.
- ❑ Each student must take part in all parts of assignment 2:
 - ❑ Part 1.1: Create table
 - ❑ Part 1.2: Insert data
 - ❑ Part 2: Store procedure, function, trigger
 - ❑ Part 3: Building application

PART 1: CREATE DATABASE (3.5 points)

I. Create table (2 points)

Write DDL statements to implement your database into the physical database. Students should identify data types, data length, and constraints based on the business description, EERD and relational database schema in Assignment 1.

Note: Your database must include following constraints:

- Primary key
- Auto increment primary key
 - Primary key with a given prefix and auto incremental values. Ex: EMP0001, EMP0002,...
- Foreign key constraint
- Unique constraint, Not null constraint, Default constraint
 - Row-based constraint relating to one or two columns. Ex: Dnumber must be in [1,20], check-in datetime must be earlier than check-out datetime

II. Insert (1.5 points)

Insert data for all tables in the database.

Note: The data in tables must be meaningful, and 3/4 tables have at least 4 rows.

NOTE: Each group needs to create a script for part 1. This script can be executed without any error to create a complete database.

PART 2: STORE PROCEDURE, FUNCTION, TRIGGER (3 points)

I. Function/ Store Procedure (2 points)

Write at least **2 functions** and **2 store procedures** to query data or perform some calculations/ operations. Please specify the input and output of these functions and store procedures in your report.

Note: Procedures and functions must be meaningful for your topic and all of the following must be used in at least 1 procedure/function:

- Query with *WHERE* and *ORDER BY* clauses from two or more tables
- Input parameter used in *WHERE* (or *HAVING*) clause
- *AGGREATE FUNCTION, GROUP BY, HAVING*
- *IF* or *FOR* statement
- Validation of input parameters

II. Trigger (1 point)

Write at least **2 triggers** to invoke operations automatically in response to *INSERT*, *UPDATE*, and *DELETE* statements on tables.

Note:

- At least 1 trigger to generate derived column values
- At least 1 trigger to enforce business rules that are meaningful for your topic. Ex: The salary of a department manager must be higher than the other employees who work for that department.

PART 3: BUILDING APPLICATIONS (3.5 points)

Build an application with the following requirements:

- Programming environment: optional (desktop application or web application).
- Programming language: optional.
- The application connects to the database created in Part 1 and Part 2.
- Display the data on the form and perform the requirement below. (Do not need to implement all the description in Assignment 1)
- Students need to prepare data, scripts to demo the application when reporting.

I. Create user (0.5 point)

Log in to the database with DBA privileges, create a user named *sManager* and assign all access rights to this user.

II. Implement features (3 points)

1. Log in, log out (enter the user name/password for *sManager* account to log in/out). (0.5 point)

2. Log in to the user *sManager* and do the following:

- a. View, Insert, Delete, Update information of an object **(1 point)**. Ex: View, Insert, Delete, Update detail information of an employee.

Note:

- At least one screen relating to two or more tables in DB
 - Students have to validate input data
 - Error message must be clearly
- b. Retrieve list of objects **(1 point)**. Ex: Retrieve all products of the company. Users can delete an object, view details of an object or create a new object from this screen (you can reuse screens in 2a). Besides, filter and sort features must be allowed.
- c. One feature that calls a store procedure/function of Part 2 **(0.5 point)**. Ex: Display monthly revenue of each month in a year.

III. Bonus (1 point)

- The user interface is attractive and friendly. **(0.5 point)**
- Building a data model (in MVC architecture) to communicate with databases. **(0.5 point)**

SUBMISSION & PRESENTATION

All teams have to **submit a hard copy report** and **present your work** to me on Sat 07th Dec 2024 (Week 49)

The team leader **submits your work (soft copy)** to LMS with only **one single zip file**. The zip file may contain resources as follows:

- Report (word or PDF) contains team member list
- SQL Script
- Source code
- Other supporting files (if any)

----- **GOOD LUCK!** -----