

# HERZING COLLEGE

## Student Management

### Introduction:

This SQL script is designed for managing Herzing College Programs, encompassing the creation of a comprehensive database, tables, triggers, and procedures/functions. The goal is to provide users with a seamless and organized solution for handling student information, programs, classes, and associated financial transactions within the educational context.

### Objective

Create a single, executable .sql file that users can run to install the Student Management Database.

### Requirement

- Create ONE, ONLY **ONE**, .sql file for the user to run/install.
- Create Database: yourName\_herzing\_college(3pts)

### Database (30pts)

- Tables: Ensure that each table is equipped with an adequate set of properties (fields), and verify that the chosen data types are accurate.
  - **Programs**
    - Id
    - Program Name
    - Max Capacity
    - Duration
    - Cost
  - **Classes**
    - id
    - Classroom id
    - Programs id
    - Start date
  - **Students**
    - Id
    - First Name
    - Last Name
    - DOB
    - Address
    - Phone
    - Email
  - **Student Profile**
    - Id
    - Students id
    - Enrolment Date
    - Classes id
    - Status [*Active, Graduated, Suspended*]
  - **Student Invoice & Bills**
    - Id
    - Students id
    - Balance
    - Date
    - Status [0, 1]
  - **Classrooms**
    - id
    - Room Number
    - Capacity

- Create a database diagram displaying all the tables and their relationships. **(10pts)**
- **Table Inputs at installation: (12pts)**  
Populate each table with a minimum of **five** inputs unless otherwise specified.
- You must ensure the inclusion of constraints and foreign keys, and remember to specify the appropriate actions for the foreign key constraints on both deletion and updating. **(15pts)**
- **Programs**  
The following three programs are sufficient:
  - Programmer Analyst = 15 months, 30000\$, max 10 students/term
  - Networking = 24 months, 24000\$, max 8 students/term
  - Software Development = 12 months, 12000\$, max 5 students/term
  - No restriction on the minimum number of students for having a class
- **The use of Stored Procedures, Stored Functions, and Triggers is a total must for the following part (20pts)**
- **Classes**
  - **Add Classes** by getting the name of the program, classroom no, and starting date.
  - Classroom number should be based on the maximum capacity of the program.
  - New classes start on the 1st Monday of September or 2nd Monday of March. Create a procedure/function to get the program name and display the latest classes from the Classes table. Then another procedure/function gets the year you want to add a new program and returns the new program start date. Then you can use this date to add a new class for that program.
  - The starting date cannot be before today and before the starting date of the latest classes in the same program.
  - **Delete Classes.** Make sure you handle it correctly by appropriate action of the foreign key.
- **Students**
  - **To Add a new student,** get all the following information: First Name, Last Name, DOB, Address, Phone, Email, Program Name.
  - Then 2 separate records should be added to the Students table & Student Profile table
  - All the students you have added started Mar 2022 or Sept 2022.
  - New students will be added according to the next start day of the program and the maximum capacity of the classes unless don't add a new student.
  - **Edit the status of students,** get the ID of a student, and update their status. Graduation cannot be earlier than the end date of each class. Students with remaining balance cannot be graduated.
- **Student\_invoice**
  - A new balance equal to the tuition fee of the program should be added to the student\_invoice with the status 0 after adding a new student to the Students table.
  - By a function/procedure get a student id and display its balance and penalty that should be added to that student.

- If the tuition is paid **monthly**, consider some students haven't made all payments up to now & add a new invoice accordingly.
  - Add: used for penalties or fees to be added to the account (Status 0 for penalty or unpaid balance, status 1 for paid balance)
- Program
  - Add
  - Delete
- Classroom
  - Add
  - Delete
- Commenting/documentation/clean code **(10pts)**

***Extra stored procedures or functions such as displaying all active students, displaying all program, displaying all classes by getting program name and so on have extra marks.***

Good Luck!