SCHOOL MANAGEMENT SYSTEM DATABASE PROJECT

A. Idea

To design a database to maintain information about school staff and students satisfying the following properties:

Enrollment System

- Departments will have their names, buildings, budgets, and balances.
- Instructors will have their names, genders, departments, and other basic information.
- Students will have their names, roll numbers, genders, departments, and other basic information.
- A table containing all the course information, including course names, units of credit, study fees, teach fees, and the corresponding departments.
- A table containing all the classroom information, including corresponding buildings and room numbers.
- A table recording all section information, including courses, instructors, quarters, years, time, classrooms, and capacities. Notice that no two sections could be arrange in the same time and classroom, no two sections could be arranged with the same instructor and time, and an instructor could only teach courses in his/her department.
- A table recording the enrollment information, including students and sections they take, their grades of homework, midterms, projects, and so on. Notice that students could only take courses offered this year, they could only enroll when the capacity is not full, they cannot enroll in sections repeatedly, and there should not be time conflict of sections enrolled.
- A table recording the dynamic enrollment process, including the sections, their capacities, and numbers of spots left currently.

Financial System

- A table recording the tuition information, including students, their payment amounts, and time of payment.
- An audit table recording all the changes of tuition table.
- A table recording tuition status of students (deficit, equilibrium, or surplus). Tuition payable depends on the specific courses the student takes. Students will be warned and dropped if they haven't paid enough money.
- A table recording the salary payment information, including instructors and their paid salary amounts. An instructor's salary is calculated by the courses he/she teaches this quarter, their titles (Professor, Associate Professor or Assistant Professor), and their years of service.
- A table recording account balances of instructors. Current balance is calculated by the previous balance and the amount of money paid by the department or the amount withdrawn by the instructor him/herself.

B. SQL file execution order

01 Create table → 02 Load (Department to Classroom) → 03 Trigger 2 → 02 Load (Teach_I / DynEnroll) → 04 View 1 → 05 Trigger 3 → 06 Trigger 4 → 07 Store Procedure 1 → 02 Load (insert into Take_S) → 08 Store Procedure 2 → 09 Trigger 5 → 02 Load (update Take_S) → 02 Load (Tuition) → 10 Trigger 6 → 11 Store Procedure 4 → 12 Store Procedure 5 → 13 Trigger 1 → 14 Trigger 7 → 15 Store Procedure 9 → 02 Load (Payment) → 16-21 Reports (View 1,2 / SP 1-8 / KPI) → 22 Test

1. Create table:

- **Department**: departments information
- **Instructor**: instructors information
- **Student**: students information
- **Course**: courses information in 2018 and 2019
- **Classroom**: Classrooms information
- **Teach_I:** classes information for instructors (same course may have different classes)
- **DynEnroll**: dynamic class enrollment information in Year 2019
- **Take_s**: enrolled classes information for students (including grades)
- **Tuition**: tuition paid amount and time
- **TuitionAudit**: record/audit every tuition payment information
- **TuitionStatus**: tuition payment status information
- **Payment**: salary and allowances for instructors paid by the school (+), and miscellaneous expenses withdrawn by instructors (-)
- **Balance**: instructor's account balance information after each payment

2. Insert data into tables / Trigger / Store Procedure

- Load: **Department** / **Instructor** / **Student** / **Course** / **Classroom** table
- Trigger 2: **TeachCheck** - conflict of time and classroom / instructor and time / department (check before inserting into Teach_I table)
- Load: **Teach_I** / **DynEnroll** table
- View 1: **ListOfClasses2019**
- Trigger 3: **ChangeSpotsLeft** (take) / **ChangeSpotsLeft1** (drop) - update SpotsLeft column in **DynEnroll** table
- Trigger 4: **EnrollCheck** - course validation / capacity / repeated enrollment / time conflict
- Store Procedure 1: **Enrollment** – take (insert into **Take_S**) or drop (delete from **Take_S**) the class
- Load: **Take_S** table
- Store Procedure 2: **UpdateGrade_XXX** - update grades in **Take_S** table
- Trigger 5: **ShowUpadatedGrade** - show upadated grade
- Update: **Take_S** - execute **UpdateGrade** to update **Take_S** table

- Load: **Tuition** table
- Trigger 6 [**Logging / Auditing mechanism**]: **TuitionAudit_insert** / **_delete** / **_update** - insert into **TuitionAudit** table after every tuition transaction
- Store Procedure 4: **TuitionCheck1** / **TuitionCheckAll** - check all students' tuition status, execute TuitionCheck1 (insert or update tuition status information in **TuitionStatus** table), and return students in deficit status
- Store Procedure 5: **DropStudentsinDeficit** - delete enrolled students in deficit status from **Take_S** table
- Trigger 1: **CalculateBalance** - calculate instructor's account balance and insert into **Balance** table
- Trigger 7: **UpdateDepartmentBalance** - update balance of the department
- Store Procedure 9: **SalaryPayment1** / **SalaryPaymentAll** – pay salaries for instructors
- Load: **Payment** / (**Balance)** table

C. Reports and prompts for user inputs

For students:

- View 1 - **ListOfClasses2019**: a list of available classes in 2019 (including left spots)
- Store Procedure1 - **Enrollment**: show class information after enrollment or drop → exec Enrollment @StudentID, @TeachID, @select ('take' or 'drop')
- Store Procedure7 - **GradeView**: student's grades information on each enrolled class → exec GradeView @StudentID
- View 2 - **EnrolledCoursesInfo**: student's enrolled courses information

For instructors:

- Store Procedure 8 - **InstructorCoursesInfo**: instructor's courses information → exec InstructorCoursesInfo @InstructorID, @Years, @Quarters
- Store Procedure 3 - **RollBook**: students' grade review and roll call check (including the number of enrolled students) → exec RollBook @TeachID
- Store Procedure 2 - **UpdateGrade_XXX**: show updated grades → exec UpdateGrade_XX @TakeID, @XXX (grade)
- Store Procedure 6 - **InstructorPaymentReport**: show transactions and balance in instructor's account → exec InstructorPaymentReport @FirstName, @LastName

For the school:

- Store Procedure 4 - **TuitionCheckAll:** return students whose tuition (study fee) has not been paid yet → exec TuitionCheckAll
- Store Procedure 5 - **DropStudentsinDeficit**: On the tuition payment deadline, if the status is 'Deficit' → can't choose any classes (drop all enrolled classes) → only on deadline: exec DropStudentsinDeficit

- Store Procedure 9: **SalaryPaymentAll** – pay salaries for all instructors each quarter → exec SalaryPaymentAll @quarters, @years
- Key Performance Indicators: Top 5 highest average final exam score class / Top 5 popular courses in 2018 and 2019 / instructors' latest account info in the order of account balance (max → min)