EduCenterDB – GROUP 5 DBI202 IA1807

## Introduction:

* 1. Project objectives:
     1. Design and implement a comprehensive management database for an English teaching center, supporting the management of student information, teachers, courses, classes, scores, and payments. Manage flexible class schedule information and detailed reports on academic and financial status .
  2. Project scope:
     1. Management Objects:
* Students: Personal information, student account, class schedule, scores, payment
* Teacher: Personal information, teacher account, teaching assignment.
* Courses: Information about courses, levels, language of instruction, course materials.
* Classes: Class information, teachers in charge, timetable.
* Payment: Manage student payments.
* Score: Manage students' scores through exams.
* Level: Manage information about course level.
* Course Materials: Manage documents related to courses
  + 1. Target Users:
* Students: Register for courses, track class schedules, check scores and manage personal accounts.
* Teachers: Manage teaching schedules, student scores and teaching materials.

## Database Model

* 1. **Database Architecture**
     1. **student** and **student**\_**account** tables: Manage personal information and student accounts. Trigger checks and updates account information when there are changes.
     2. **teacher** and **teacher**\_**account** tables: Manage personal information and teacher accounts.
     3. table **course** and **course**\_**material**, **level**: Manage courses, levels and related materials.
     4. **class** and **class**\_**weekday** tables: Manage class schedules and classes.
     5. **exam** and **grade** table: Manage exam information and scores.
     6. **payment** table: Manage payments and payment status.
  2. **Relationships**
     1. **student and class to class\_student**:
* **Description:** student có mối quan hệ nhiều-nhiều với class thông qua bảng trung gian class\_student. Mỗi học sinh có thể tham gia nhiều lớp học, và mỗi lớp học có thể có nhiều học sinh**.**
* **Foreign key:**
  + class\_student.class\_id -> class.id
  + class\_student.student\_id -> student.id
    1. **course and course\_material**:
* **Description:** course có mối quan hệ một-nhiều với course\_material, nghĩa là mỗi khóa học có thể có nhiều tài liệu, nhưng mỗi tài liệu chỉ thuộc về một khóa học**.**
* **Foreign key:** course\_material.course\_id -> course.id
  + 1. **teacher and class:**
* **Description:** teacher có mối quan hệ một-nhiều với class, nghĩa là mỗi giáo viên có thể dạy nhiều lớp, nhưng mỗi lớp chỉ có một giáo viên**.**
* **Foreign key:** class.teacher\_id -> teacher.id
  + 1. **exam and class**:
* **Description:** exam có mối quan hệ một-nhiều với class, nghĩa là mỗi lớp có thể có nhiều kỳ thi, nhưng mỗi kỳ thi chỉ thuộc về một lớp**.**
* **Foreign key:** exam.class\_id -> class.id
  + 1. **payment and student:**
* **Description:** payment có mối quan hệ nhiều-một với student, nghĩa là mỗi học sinh có thể có nhiều khoản thanh toán, nhưng mỗi khoản thanh toán chỉ thuộc về một học sinh).
* **Foreign key:** payment.student\_id -> student.id

## DataSheet Details

* 1. student
* id: Mã số học viên (Primary Key).
* first\_name: Tên.
* last\_name: Họ.
* date\_birth: Ngày sinh.
* gender: Giới tính.
* email: Địa chỉ email.
* phone: Số điện thoại.
* street, city, state, zip: Địa chỉ.
  1. teacher
* id: Mã số giáo viên (Primary Key).
* first\_name: Tên.
* last\_name: Họ.
* description: Mô tả.
* date\_birth: Ngày sinh.
* gender: Giới tính.
* email: Địa chỉ email.
* phone: Số điện thoại.
* street, city, state, zip: Địa chỉ.
  1. class
* id: Mã lớp học (Primary Key).
* start\_date: Ngày bắt đầu.
* end\_date: Ngày kết thúc.
* teacher\_id: Mã giáo viên phụ trách lớp (Foreign Key tham chiếu bảng teacher).
* course\_id: Mã khóa học liên quan (Foreign Key tham chiếu bảng course)
  1. course
* id: Mã khóa học (Primary Key).
* description: Mô tả khóa học.
* level\_id: Mã cấp độ khóa học (Foreign Key tham chiếu bảng level).
  1. Payment
* id: Mã thanh toán (Primary Key).
* payment\_date: Ngày thanh toán.
* amount: Số tiền thanh toán.
* status: Trạng thái thanh toán.
* student\_id: Mã học viên (Foreign Key tham chiếu bảng student)
  1. grade
* id: Mã điểm số (Primary Key).
* value: Giá trị điểm số.
* student\_id: Mã học viên (Foreign Key tham chiếu bảng student).
* exam\_id: Mã bài thi (Foreign Key tham chiếu bảng exam
  1. Exam
* id: Mã bài thi (Primary Key).
* date: Ngày thi.
* description: Mô tả bài thi.
* class\_id: Mã lớp học liên quan (Foreign Key tham chiếu bảng class).
  1. class\_student
* class\_id: Mã lớp học (Primary Key kết hợp).
* student\_id: Mã học viên (Primary Key kết hợp)
  1. course\_material
* id: Mã tài liệu (Primary Key).
* course\_id: Mã khóa học liên quan (Foreign Key tham chiếu bảng course)
* description: Mô tả tài liệu
* material\_type: Loại tài liệu
* material\_url: Đường dẫn tới tài liệu.
* date\_add: Ngày thêm tài liệu
  1. Level
* id: Mã cấp độ (Primary Key).
* name: Tên cấp độ.
  1. teacher\_account
* teacher\_id: Mã giáo viên (Primary Key và Foreign Key tham chiếu bảng teacher).
* username: Tên đăng nhập.
* password: Mật khẩu.
* is\_active: Trạng thái hoạt động (BIT)
  1. student\_account
* student\_id: Mã học viên (Primary Key và Foreign Key tham chiếu bảng student).
* username: Tên đăng nhập.
* password: Mật khẩu.
* balance: Số dư tài khoản.
* created\_date: Ngày tạo tài khoản
* class\_weekday:
* class\_id: Mã lớp học (Foreign Key tham chiếu bảng class).
* weekday\_id: Mã ngày trong tuần

## Tasks - Plan

* 1. **List of basic and advanced tasks according to the table:**
* **Sơn**: Manage queries and data: course, course\_material, level
* **Bằng**: Managing queries and data: class, class\_weekday.
* **Vân:** Manage queries and data: student, student\_account, class\_student.
* **Dương** Manage queries and data: grade, exam, payment.
* **Đạt**: Manage queries, data teacher, teacher\_account.

## Advanced Components

* 1. **Transactions**
     1. Transactions to maintain data integrity.
  2. **Indexes**
     1. INDEX testcourse ON dbo.course(description)
* The purpose of creating such an index is to increase query speed when you perform conditional SELECT statements with the description column. Index helps databases organize data more effectively for faster access when needed.
  1. **Views**
     1. View ClassWithTeacherName
* Uses:
  + Easy to query: Organize data in a clearer and more structured way, making information management and access easy
  + Easy query: Easier to query class information and teacher names without rewriting complex JOIN statements.
    1. View Grade và Payment
* Purpose (Grade) : Query all average scores of all students
* Purpose (Payment) : Query all transactions of each specific student, payment status of each transaction
  1. **Functions**
     1. Function get\_student\_info
* Uses:
  + The get\_student\_info function is created to retrieve detailed information of a student based on the student\_id provided. The main purpose of this function is to simplify querying student information from the student table without having to rewrite the SQL statement every time you need to retrieve data for a specific student. This makes SQL code easier to read and reduces errors when executing complex queries.
* Using:
  + SELECT \* FROM get\_student\_info(1);
  1. **Stored Procedures**
     1. Procedure AddClassSchedule
* Uses :
  + **Check data**: Make sure the class exists, the weekday ID is valid, and the class schedule does not exist before adding new data.
  + **Increase consistency**: Avoid potential errors caused by manually adding class schedules.
  + **Add class schedule**: Helps add class schedule to the class\_weekday table safely
* Using : For example, add the class schedule of class C001 on Monday
  + EXEC AddClassSchedule @class\_id = 'C001', @weekday\_id = 'Mon';
    1. Procedure `AddTeacher
  + To insert a new record into the ` teacher ` table
  + Gets detailed parameters about the teacher and inserts this information into the table ` teacher
  1. **Triggers**
     1. **Trigger UpdateTeacherAccount**
* Uses :
  + Trigger UpdateTeacherAccount is designed to automatically update teacher account information in the teacher\_account table when there is a change in the teacher table.
  + Specifically, it checks to see if the teacher's email or phone number has changed, and if so, it updates the username field in the teacher\_account table to correspond to the new email.
* **Benefits of using UpdateTeacherAccount trigger**
  + Automation: Trigger helps automate updating teacher account information, minimizing errors caused by manual operations.
  + Ensure data integrity: Trigger ensures that teacher account information in the teacher\_account table is always synchronized with the teacher table.
  + Increased performance: Triggers run on the database server, increasing performance compared to checking and updating from the application side.
* Using

UPDATE teacher

SET email = 'new.email@example.com', phone = '0987654321'

WHERE id = 1;

* 1. **Cursor**
     1. Browse Student List and Print Out Students Who Don't Have Email
* Uses :
  + Check and Update Data: Ensure that all students have valid email information.
  + Special Data Processing: Perform special tasks for students lacking email, such as sending notifications or updating data.
  + Reporting and Statistics: Create a report on the number of students who do not have an email to have a plan to update information.

## Conclusion

* The education center management system has been built with the functions of managing student information, teachers, courses, classes, exams, scores, and payments.
* The system uses database tables linked together through foreign keys to ensure data integrity.
* Advanced components such as transactions, indexes, views, functions, stored procedures, triggers, and cursors are used to optimize and automate tasks.
* **Result:** 
  + Students and Student Accounts: 100 students and 100 accounts respectively.
  + Teachers and Teacher Accounts: 10 teachers and 10 accounts.
  + Classes: A total of 10 classes.
  + Course: Includes 12 courses, such as Listening 1-3, Reading 1-3, Writing 1-3, Speaking 1-3.
  + Documents: A total of 36 documents, including textbooks, workbooks and reference books for each level.
  + Payment: Successfully performed 100 payment transactions (one transaction per student).
  + Score: A total of 500 points for 5 exams taken within 5 months of study for 10 classes.