

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



HỆ CƠ SỞ DỮ LIỆU (CO2014)

Bài tập lớn 2 - L01 - Học kì 241

"Hệ Thống Gia Sư Dạy Kèm Tại Nhà"

Instructor(s): GV. Trần Thị Quế Nguyệt

Student(s):	Dào Duy Quý	2212864	(Lớp L01 - Nhóm 7)
	Lê Bùi Trung Dũng	2210573	(Lớp L01 - Nhóm 7)
	Phạm Phương Duy	2210532	(Lớp L01 - Nhóm 7)
	Trần Thị Ngọc Huyền	2211311	(Lớp L01 - Nhóm 7)
	Trương Nguyễn Minh Nhiên	2212452	(Lớp L01 - Nhóm 7)



Contents

Thành viên và phân chia công việc	3
1 Giới thiệu đề tài	4
2 Tạo bảng và dữ liệu mẫu	5
2.1 Các câu lệnh tạo bảng	5
2.2 Ràng buộc dữ liệu trên các bảng	17
2.3 Thêm dữ liệu	18
3 Hiện thực ứng dụng	27
3.1 Thủ tục INSERT / UPDATE / DELETE dữ liệu vào 1 bảng dữ liệu	27
3.2 Trigger	55
3.2.1 Trigger tính toán điểm đánh giá (rate) gia sư	55
3.2.2 Trigger tính toán điểm đóng góp của gia sư	58
3.3 Thủ tục	61
3.3.1 Thủ tục 1: Viết 1 câu truy vấn có mệnh đề WHERE và ORDER BY	61
3.3.2 Thủ tục 2	68
3.4 Hàm	78
3.4.1 Hàm tính doanh thu thực tế, kỳ vọng và giảm giá	78
3.4.2 Hàm tính số lượng đơn đăng ký của gia sư	86
4 Ứng dụng kết nối cơ sở dữ liệu	89
4.1 Tổng quan công nghệ	89
4.1.1 Công nghệ sử dụng	89
4.1.2 Tổ chức thư mục code	89
4.1.3 Khai báo kết nối với BDMS	89
4.2 Các màn hình thử nghiệm	89
5 Phần làm thêm	95
5.1 Phần tạo bảng và dữ liệu mẫu: Bổ sung ràng buộc ngữ nghĩa địa chỉ của lớp học phải thuộc những địa chỉ của học sinh trong lớp học đó	95
5.2 Hiện thực ràng buộc ngữ nghĩa năm của bảng cấp mà gia sư thêm vào hệ thống phải nằm trong khoảng 2 năm gần đây, hiện thực bằng thủ tục adding_new_certificate	95
5.3 Phần tạo bảng và dữ liệu mẫu: Bổ sung ràng buộc ngữ nghĩa rằng tiền đặt cọc của lớp học không được lớn hơn lương nhằm đảm bảo quyền lợi cho gia sư	102
5.4 Phần hiện thực ứng dụng	102
5.4.1 Phần nhóm làm thêm: Thủ tục INSERT/ UPDATE vào bảng dữ liệu	102
6 Phụ lục	106
ĐÁNH GIÁ CỦA GIÁO VIÊN	108



Thành viên và phân chia công việc

No.	Fullscreen	StudentID	Problems	Contribute
1	Dào Duy Quý	2212864	2.1 Tạo bảng - ràng buộc 4.2 Hiện thực frontend và backend	100%
2	Lê Bùi Trung Dũng	2210573	3.1 Viết thủ tục update/insert/delete 3.3.2 Viết thủ tục 2 và 4.2 Hiện thực backend	100%
3	Phạm Phương Duy	2210532	2.2 Ràng buộc dữ liệu và 3.4.2 Viết hàm thống kê 4.2 Hiện thực frontend	100%
4	Trần Thị Ngọc Huyền	2211311	3.1 Viết thủ tục update/insert/delete 3.2 Trigger và 3.4.1 Viết hàm doanh thu	100%
5	Trương Nguyễn Minh Nhiên	2212452	2.3 Thêm dữ liệu và 4.2 Thiết kế giao diện 3.3.1 Viết thủ tục với mệnh đề WHERE/ORDERBY	100%

Phân chia công việc theo tuần

No.	Week	Mission	Members
1	Nửa đầu tuần 46	Tìm hiểu đề tài	Nhóm
2	Nửa sau tuần 46 - 47	Tạo bảng và ràng buộc ngữ nghĩa, thêm dữ liệu	Quý, Nhiên, Duy
3	Tuần 48	- Set up backend - Set up frontend - Thiết kế giao diện với Figma - Requirement + báo cáo - Chính sửa thay đổi so với BTL1	Dũng Quý Nhiên Huyền Duy
4	Tuần 49	- Code frontend - Code backend + thủ tục 3.1 - Viết thủ tục với mệnh đề 3.3 - Viết thủ tục 3.1 + trigger	Quý, Duy Dũng Nhiên Huyền
5	Tuần 50	- Code frontend + backend - Code backend + thủ tục với mệnh đề 3.3 - Viết hàm 3.4 - Hoàn thiện báo cáo	Quý Dũng Huyền, Duy Nhóm



1 Giới thiệu đề tài

Hệ thống "Gia sư Dạy kèm tại nhà" được thiết kế nhằm quản lý tài khoản người dùng, các hoạt động và thông tin liên quan đến gia sư, học viên cũng như lớp học. Mục tiêu của hệ thống là tối ưu hóa quá trình kết nối giữa gia sư và học viên thông qua nền tảng trực tiếp, tạo điều kiện thuận lợi cho quá trình học tập và giảng dạy.

Người dùng trong hệ thống này được chia thành hai nhóm chính: Nhân viên và Học viên. Nhân viên bao gồm hai nhóm: Gia sư và Quản trị viên. Người dùng khi tham gia hệ thống bắt buộc phải cung cấp thông tin cá nhân tùy vào loại tài khoản và tải lên hệ thống minh chứng nhằm đảm bảo uy tín, minh bạch hệ thống.

Học viên sau khi đăng nhập vào website có thể đăng ký học tập theo hai cách: Yêu cầu tư vấn và Đăng ký gia sư.

Học viên gửi Yêu cầu tư vấn bằng cách cung cấp các thông tin: Hình thức dạy (trực tuyến, trực tiếp,...), loại giảng dạy (các khối cơ sở, trung học cơ sở, ôn đại học,...), môn học, địa chỉ học, yêu cầu khác. Dựa vào thông tin này, Quản trị viên sẽ liên hệ học viên qua số điện thoại và tư vấn. Nếu đạt được sự thống nhất, Quản trị viên tạo Lớp.

Những Lớp này ở trạng thái chưa được giao và được đăng công khai trên website. Lớp phải đủ các thông tin môn, hình thức dạy, loại giảng dạy, địa chỉ, phí hoa hồng, phí cọc, lương, thời gian học và yêu cầu cụ thể. Gia sư dựa vào thông tin này để đăng ký nhận lớp, Quản trị viên sau khi nhận được đăng ký nhận lớp của Gia sư tiến hành kiểm tra thông tin chuyên môn gia sư. Nếu đủ yêu cầu, đơn đăng ký nhận lớp được duyệt, gia sư đóng phí cọc để giữ lớp. Sau khi dạy từ 3 - 4 buổi, phí giữ lớp sẽ được hoàn trả. Gia sư đóng phí hoa hồng cho website khi nhận tháng lương đầu tiên, phí hoa hồng thường từ 10% - 35%.

Đối với Hình thức Đăng ký gia sư, Học viên cũng cung cấp các thông tin tương tự Yêu cầu tư vấn, tuy nhiên có chỉ định gia sư cụ thể. Lúc này, Quản trị viên đóng vai trò trung gian giữa hai bên, nếu hai bên nhất trí sẽ mở lại cho Gia sư và Học viên. Gia sư cũng đóng các loại phí như trên. Hóa đơn thanh toán cần lưu lại trong hệ thống nhằm quản lý doanh thu.

Hệ thống "Gia sư Day kèm tại nhà" là một hệ thống quản lý toàn diện, cung cấp các công cụ hữu ích nhằm hỗ trợ gia sư và học viên trong quá trình giảng dạy và học tập. Tuy nhiên, trong bài tập lớn này, nhóm tập trung hiện thực nghiệp vụ **Quản lý lớp học của Quản trị viên (Admin)** nhằm phù hợp với yêu cầu đề tài.



2 Tạo bảng và dữ liệu mẫu

2.1 Các câu lệnh tạo bảng

1. Bảng address:

```
CREATE TABLE [dbo].[address](
    [addr_id] [bigint] IDENTITY(1,1) NOT NULL,
    [house_number] [int] NULL,
    [str_name] [varchar](255) NULL,
    [user_id] [bigint] NOT NULL,
    [dist_city_id] [bigint] NULL,
    [pro_id] [bigint] NULL,
    [ward_id] [bigint] NULL,
PRIMARY KEY CLUSTERED
(
    [addr_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 1: Lệnh tạo bảng address

2. Bảng Administrator:

```
CREATE TABLE [dbo].[administrator](
    [admin_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [admin_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 2: Lệnh tạo bảng Administrator

3. Bảng belongs_to:

```
CREATE TABLE [dbo].[belongs_to](
    [bill_id] [bigint] NOT NULL,
    [class_id] [bigint] NOT NULL,
    [tutor_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [bill_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 3: Lệnh tạo bảng belongs_to



4. Bảng bill:

```
CREATE TABLE [dbo].[bill](
    [bill_id] [bigint] IDENTITY(1,1) NOT NULL,
    [bill_money] [bigint] NULL,
    [bill_photo_url] [varbinary](max) NULL,
    [bill_status] [varchar](255) NULL,
    [bill_type] [varchar](255) NULL,
    [admin_id] [bigint] NULL,
    [vou_id] [bigint] NULL,
PRIMARY KEY CLUSTERED
(
    [bill_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

Listing 4: Lệnh tạo bảng bill

5. Bảng can_teach_at:

```
CREATE TABLE [dbo].[can_teach_at](
    [tutor_id] [bigint] NOT NULL,
    [dist_city_id] [bigint] NOT NULL,
    [pro_id] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 5: Lệnh tạo bảng can_teach_at

6. Bảng can_teach_style:

```
CREATE TABLE [dbo].[can_teach_style](
    [tutor_id] [bigint] NOT NULL,
    [ts_id] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 6: Lệnh tạo bảng can_teach_style

7. Bảng can_teach:

```
CREATE TABLE [dbo].[can_teach](
    [tutor_id] [bigint] NOT NULL,
    [subject_id] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 7: Lệnh tạo bảng can_teach

8. Bảng can_teach_type:

```
CREATE TABLE [dbo].[can_teach_type](
    [tutor_id] [bigint] NOT NULL,
    [class_type_id] [bigint] NOT NULL
) ON [PRIMARY]
```



Listing 8: Lệnh tạo bảng can_teach_type

9. Bảng certificate:

```
CREATE TABLE [dbo].[certificate](
    [cert_id] [bigint] NOT NULL,
    [cert_year] [int] NULL,
    [cert_grade] [varchar](255) NULL,
    [tutor_id] [bigint] NOT NULL,
    [ct_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [cert_id] ASC,
    [tutor_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 9: Lệnh tạo bảng certificate

10. Bảng certificate_type:

```
CREATE TABLE [dbo].[certificate_type](
    [ct_id] [bigint] IDENTITY(1,1) NOT NULL,
    [ct_name] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
    [ct_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 10: Lệnh tạo bảng certificate_type

11. Bảng class:

```
CREATE TABLE [dbo].[class](
    [class_id] [bigint] IDENTITY(1,1) NOT NULL,
    [class_deposit] [bigint] NULL,
    [class_status] [varchar](255) NULL,
    [commission_fee] [bigint] NULL,
    [requirements] [varchar](255) NULL,
    [date_start] [datetime2](7) NULL,
    [salary] [bigint] NULL,
    [addr_id] [bigint] NOT NULL,
    [student_id] [bigint] NOT NULL,
    [ts_id] [bigint] NOT NULL,
    [tutor_id] [bigint] NULL,
PRIMARY KEY CLUSTERED
(
```



```
[class_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 11: Lệnh tạo bảng class

12. Bảng class_type:

```
CREATE TABLE [dbo].[class_type](
    [class_type_id] [bigint] IDENTITY(1,1) NOT NULL,
    [class_type_name] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
    [class_type_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 12: Lệnh tạo bảng class_type

13. Bảng consultation_req:

```
CREATE TABLE [dbo].[consultation_req](
    [cq_id] [bigint] IDENTITY(1,1) NOT NULL,
    [cq_status] [varchar](255) NULL,
    [requirement] [varchar](255) NULL,
    [addr_id] [bigint] NOT NULL,
    [student_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [cq_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 13: Lệnh tạo bảng consultation_req

14. Bảng wants_subject:

```
CREATE TABLE [dbo].[wants_subject](
    [cq_id] [bigint] NOT NULL,
    [subject_id] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 14: Lệnh tạo bảng wants_subject

15. Bảng wants_type:



```
CREATE TABLE [dbo].[wants_type](
    [cq_id] [bigint] NOT NULL,
    [class_type_id] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 15: Lệnh tạo bảng wants_type

16. Bảng degree:

```
CREATE TABLE [dbo].[degree](
    [degr_id] [bigint] NOT NULL,
    [degr_major_name] [varchar](255) NULL,
    [degr_type] [varchar](255) NULL,
    [degr_year] [int] NULL,
    [tutor_id] [bigint] NOT NULL,
    [edi_code] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [degr_id] ASC,
    [tutor_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 16: Lệnh tạo bảng degree

17. Bảng educational_institution:

```
CREATE TABLE [dbo].[educational_institution](
    [edi_code] [bigint] IDENTITY(1,1) NOT NULL,
    [edi_name] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
    [edi_code] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 17: Lệnh tạo bảng educational_institution

18. Bảng district_city:

```
CREATE TABLE [dbo].[district_city](
    [dist_city_id] [bigint] NOT NULL,
    [name] [varchar](255) NULL,
    [pro_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [dist_city_id] ASC,
    [pro_id] ASC
```



```
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 18: Lệnh tạo bảng district_city

19. Bảng has:

```
CREATE TABLE [dbo].[has](
    [tutor_id] [bigint] NOT NULL,
    [edi_code] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 19: Lệnh tạo bảng has

20. Bảng has_class_type:

```
CREATE TABLE [dbo].[has_class_type](
    [class_id] [bigint] NOT NULL,
    [class_type_id] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 20: Lệnh tạo bảng has_class_type

21. Bảng has_subject:

```
CREATE TABLE [dbo].[has_subject](
    [class_id] [bigint] NOT NULL,
    [subject_id] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 21: Lệnh tạo bảng has_subject

22. Bảng is_held_on:

```
CREATE TABLE [dbo].[is_held_on](
    [class_id] [bigint] NOT NULL,
    [week_id] [bigint] NOT NULL,
    [slot_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [class_id] ASC,
    [slot_id] ASC,
    [week_id] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 22: Lệnh tạo bảng is_held_on

23. Bảng province:



```
CREATE TABLE [dbo].[province](
    [pro_id] [bigint] IDENTITY(1,1) NOT NULL,
    [pro_name] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
    [pro_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
    OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 23: Lệnh tạo bảng province

24. Bảng qualification:

```
CREATE TABLE [dbo].[qualification](
    [qualification] [varchar](255) NOT NULL,
    [tutor_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [qualification] ASC,
    [tutor_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
    OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 24: Lệnh tạo bảng qualification

25. Bảng staff:

```
CREATE TABLE [dbo].[staff](
    [date_of_birth] [datetime2](7) NULL,
    [national_id] [varchar](255) NULL,
    [place_of_origin] [varchar](255) NULL,
    [profile_photo_url] [varbinary](max) NULL,
    [staff_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [staff_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
    OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

Listing 25: Lệnh tạo bảng staff

26. Bảng student:

```
CREATE TABLE [dbo].[student](
    [stu_grade] [int] NULL,
    [stu_school] [varchar](255) NULL,
    [student_id] [bigint] NOT NULL,
```



```
PRIMARY KEY CLUSTERED
(
    [student_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 26: Lệnh tạo bảng student

27. Bảng subject:

```
CREATE TABLE [dbo].[subject](
    [subject_id] [bigint] IDENTITY(1,1) NOT NULL,
    [subject_name] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
    [subject_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 27: Lệnh tạo bảng subject

28. Bảng s_wants_subject:

```
CREATE TABLE [dbo].[s_wants_subject](
    [ta_id] [bigint] NOT NULL,
    [subject_id] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 28: Lệnh tạo bảng s_wants_subject

29. Bảng s_wants_type:

```
CREATE TABLE [dbo].[s_wants_type](
    [ta_id] [bigint] NOT NULL,
    [class_type_id] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 29: Lệnh tạo bảng s_wants_type

30. Bảng teaching_application:

```
CREATE TABLE [dbo].[teaching_application](
    [teaching_appl_id] [bigint] IDENTITY(1,1) NOT NULL,
    [application_status] [varchar](255) NULL,
    [date_of_creation] [datetime2](7) NULL,
    [class_id] [bigint] NOT NULL,
    [tutor_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
```



```
[teaching_appl_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 30: Lệnh tạo bảng teaching_application

31. Bảng teaching_style:

```
CREATE TABLE [dbo].[teaching_style](
    [ts_id] [bigint] IDENTITY(1,1) NOT NULL,
    [ts_name] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
    [ts_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 31: Lệnh tạo bảng teaching_style

32. Bảng time_slot:

```
CREATE TABLE [dbo].[time_slot](
    [slot_id] [bigint] IDENTITY(1,1) NOT NULL,
    [end_time] [datetime2](7) NULL,
    [start_time] [datetime2](7) NULL,
PRIMARY KEY CLUSTERED
(
    [slot_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 32: Lệnh tạo bảng time_slot

33. Bảng tutor:

```
CREATE TABLE [dbo].[tutor](
    [bio] [varchar](255) NULL,
    [date_joined] [datetime2](7) NULL,
    [inviting_code] [varchar](255) NULL,
    [n_of_invitations] [int] NULL,
    [rate] [int] NULL,
    [tutor_id] [bigint] NOT NULL,
    [invited_code] [bigint] NULL,
PRIMARY KEY CLUSTERED
(
    [tutor_id] ASC
```



```
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
CONSTRAINT [UK_g5accvpotj6irddk8dn58ef47] UNIQUE NONCLUSTERED
(
    [inviting_code] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 33: Lệnh tạo bảng tutor

34. Bảng tutor_application:

```
CREATE TABLE [dbo].[tutor_application](
    [ta_id] [bigint] IDENTITY(1,1) NOT NULL,
    [requirement] [varchar](255) NULL,
    [ta_status] [varchar](255) NULL,
    [addr_id] [bigint] NOT NULL,
    [student_id] [bigint] NOT NULL,
    [ts_id] [bigint] NOT NULL,
    [tutor_id] [bigint] NULL,
PRIMARY KEY CLUSTERED
(
    [ta_id] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 34: Lệnh tạo bảng tutor_application

35. Bảng tutor_review:

```
CREATE TABLE [dbo].[tutor_review](
    [tutor_review_id] [bigint] IDENTITY(1,1) NOT NULL,
    [comment] [varchar](255) NULL,
    [rate] [int] NULL,
    [time_stamp] [datetime2](7) NULL,
    [class_id] [bigint] NOT NULL,
    [student_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [tutor_review_id] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 35: Lệnh tạo bảng tutor_review

36. Bảng user_contact:



```
CREATE TABLE [dbo].[user_contact](
    [contact_email] [varchar](255) NOT NULL,
    [contact_phone_number] [varchar](255) NOT NULL,
    [social_media_link] [varchar](255) NOT NULL,
    [user_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [contact_email] ASC,
    [contact_phone_number] ASC,
    [social_media_link] ASC,
    [user_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 36: Lệnh tạo bảng user_contact

37. Bảng user:

```
CREATE TABLE [dbo].[user](
    [user_id] [bigint] IDENTITY(1,1) NOT NULL,
    [full_name] [varchar](255) NULL,
    [password] [varchar](255) NOT NULL,
    [phone_number] [varchar](255) NOT NULL,
    [user_sex] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
    [user_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
CONSTRAINT [UK_9q63snka3mdh91as4io72espi] UNIQUE NONCLUSTERED
(
    [phone_number] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 37: Lệnh tạo bảng user

38. Bảng voucher:

```
CREATE TABLE [dbo].[voucher](
    [vou_id] [bigint] IDENTITY(1,1) NOT NULL,
    [vou_discount] [NUMERIC](10,2) NULL,
    [vou_status] [varchar](255) NULL,
    [tutor_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [vou_id] ASC
```



```
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 38: Lệnh tạo bảng voucher

39. Bảng wants_style:

```
CREATE TABLE [dbo].[wants_style](
    [cq_id] [bigint] NOT NULL,
    [ts_id] [bigint] NOT NULL
) ON [PRIMARY]
```

Listing 39: Lệnh tạo bảng wants_style

40. Bảng ward:

```
CREATE TABLE [dbo].[ward](
    [ward_id] [bigint] NOT NULL,
    [ward_name] [varchar](255) NULL,
    [dist_city_id] [bigint] NOT NULL,
    [pro_id] [bigint] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [dist_city_id] ASC,
    [pro_id] ASC,
    [ward_id] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 40: Lệnh tạo bảng ward

41. Bảng week_day:

```
CREATE TABLE [dbo].[week_day](
    [week_id] [bigint] IDENTITY(1,1) NOT NULL,
    [name] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
    [week_id] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
      = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
      OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Listing 41: Lệnh tạo bảng week_day



2.2 Ràng buộc dữ liệu trên các bảng

- Ràng buộc về giới tính người dùng

```
ALTER TABLE [dbo].[user]
ADD CONSTRAINT chk_user_sex
CHECK ([user_sex] IN ('Male', 'Female', 'Other'));
```

Listing 42: Kiểm tra giới tính hợp lệ

- kiểm tra nhân viên có đủ 18 tuổi trở lên không

```
ALTER TABLE [dbo].[staff]
ADD CONSTRAINT chk_age_limit
CHECK (DATEDIFF(YEAR, date_of_birth, GETDATE()) BETWEEN 18 AND
65);
```

Listing 43: Kiểm tra tuổi nhân viên

- kiểm tra rating trong phạm vi từ 1 đến 5

```
ALTER TABLE dbo.tutor_review
ADD CONSTRAINT chk_rate_range
CHECK (rate >= 1 AND rate <= 5);
```

Listing 44: Kiểm tra rating

- kiểm tra trạng thái thanh toán

```
ALTER TABLE [dbo].[bill]
ADD CONSTRAINT chk_bill_status
CHECK (bill_status IN ('Xac nhan', 'Chua xac nhan'));

SET IDENTITY_INSERT [dbo].[user] ON;
```

Listing 45: Kiểm tra trạng thái thanh toán



2.3 Thêm dữ liệu

Dữ liệu mẫu được lưu trữ trong 41 tệp .XLXS trong folder `Dataset` và dữ liệu được chèn từ Dataset vào DBMS bằng file `insertDatabase(Ass2_CO2013).py`.

Kết quả dữ liệu của 41 bảng trong CSDL:

1. Bảng address:

	addr_id	house_number	str_name	user_id	dist_city_id	pro_id	ward_id
1	1	190	duong Le Thanh Ton	1	71000	1	71009
2	2	215	duong Nguyen Van Troi	2	72200	1	72206
3	3	216	duong Vo Van Gan	3	71300	1	71306
4	4	278	duong Phan Anh	4	72000	1	72014
5	5	66	duong Co Bac	5	71000	1	71011
6	6	8	duong Hoa Binh	6	72000	1	72014
7	7	53	duong Lam Van Ben	7	72900	1	72911
8	8	94	duong so 50	8	71900	1	71915
9	9	15	duong Ky Dong	9	72400	1	72414
10	10	91	duong Chu Van An	10	72300	1	72312
11	11	24	duong Tran Binh	11	73100	1	73107
12	12	278	duong Long Thuan	12	71200	1	71214
13	13	25	duong Le Duan	13	71000	1	71006
14	14	177	duong Phan Xich Long	14	72200	1	72210
15	15	2	duong Le Van Mien	15	71300	1	71107
16	16	3	duong Vo Thi Thua	16	71500	1	71516
17	17	55	duong A	17	71100	1	71112
18	18	35	duong Lanh Binh Tha...	18	72600	1	72619
19	19	728	duong Nguyen Duy	19	73000	1	73011
20	20	24	duong so 6	20	72500	1	72510

Hình 1: Bảng address.

2. Bảng administrator và province và s_wants_type:

	admin_id	pro_id	pro_name	ta_id	class_type_id
1	8		Tp. Ho Chi Minh	1	4
2	9			2	8
3	10	1		3	11
4	11			4	2
5	12				

Hình 2: Bảng administrator và province và s_wants_type.



3. Bảng can_teach_style và bill:

		bill_id	bill_money	bill_photo_url	bill_status	bill_type	admin_id	vou_id
	tutor_id	ts_id						
1	13	1	200000	NULL	Xac nhan	Dong coc	10	NULL
2	14	1	200000	NULL	Xac nhan	Hoan coc	8	NULL
3	14	2	500000	NULL	Xac nhan	Phi hoa hong	9	NULL
4	15	1	200000	NULL	Xac nhan	Dong coc	12	NULL
5	16	2	6	270000	Xac nhan	Hoan coc	11	NULL
6	17	1	7	200000	Xac nhan	Phi hoa hong	8	1
7	17	2	8	200000	Xac nhan	Dong coc	9	NULL
8	18	1	9	8	Xac nhan	Hoan coc	10	NULL
9	19	1	9	400000	Xac nhan	Phi hoa hong	12	NULL
10	20	1	15	360000	Chua xa...	Phi hoa hong	12	5
			16	300000	Xac nhan	Dong coc	10	NULL
			17	300000	Chua xa...	Hoan coc	11	NULL
			18	300000	Chua xa...	Dong coc	10	NULL
			19	200000	Chua xa...	Dong coc	8	NULL
			20	400000	Chua xa...	Dong coc	8	NULL

Hình 3: Bảng can_teach_style và bill.

4. Bảng class:

	class_id	class_deposit	class_status	commission_fee	requirements	date_start	salary	addr_id	student_id	ts_id	tutor_id
1	1	200000	Da giao	500000	NULL	2022-10-07 ...	2000000	2	1	1	13
2	2	200000	Da giao	300000	NULL	2022-08-20 ...	3000000	3	2	1	14
3	3	200000	Da giao	400000	NULL	2022-09-05 ...	2000000	4	3	1	15
4	4	100000	Da giao	100000	NULL	2023-05-30 ...	1000000	5	4	2	16
5	5	300000	Da giao	400000	NULL	2023-06-19 ...	4000000	6	5	1	17
6	6	300000	Da giao	300000	NULL	2024-07-05 ...	3000000	7	6	1	18
7	7	300000	Da giao	400000	NULL	2024-06-08 ...	3000000	8	7	1	19
8	8	300000	Chua giao	300000	NULL	NULL	3000000	5	4	2	16
9	9	200000	Chua giao	200000	NULL	NULL	2000000	8	7	2	19
10	10	400000	Chua giao	500000	NULL	NULL	4000000	4	3	1	15

Hình 4: Bảng class.

5. Bảng consultant_req:

	cq_id	cq_status	requirement	addr_id	student_id
1	1	Da xu ly	lop 4: van, khxh, toan	1	1
2	2	Da xu ly	lop 8: toan, tieng anh, khtn	2	2
3	3	Da xu ly	luyen thi dai hoc: khxh, tieng anh va on luyen t...	4	4
4	4	Da xu ly	lop 11: tieng anh, van, khtn	5	5
5	5	Chua xu ly	lop 2: khxh, toan, van	6	6

Hình 5: Bảng consultant_req.



6. Bảng can_teach và can_teach_at và belongs_to:

	tutor_id	subject_id		tutor_id	dist_city_id	pro_id		bill_id	class_id	tutor_id
1	13	1		1	13	71000	1	1	1	13
2	13	3		2	13	72700	1	2	2	13
3	13	2		3	13	72800	1	3	3	13
4	14	2		4	14	72400	1	4	4	14
5	14	5		5	14	72200	1	5	5	14
6	14	4		6	14	71400	1	6	6	14
7	15	3		7	15	72300	1	7	7	15
8	15	2		8	15	71200	1	8	8	15
9	15	6		9	15	71300	1	9	9	15
10	16	3		10	16	71500	1	10	10	16
11	16	6		11	16	71900	1	11	11	16
12	16	4		12	16	72000	1	12	12	16
13	17	4		13	17	71100	1	13	13	17
14	17	1		14	17	72800	1	14	14	17
15	17	5		15	17	71000	1	15	15	17
16	18	3		16	18	72600	1	16	16	18
17	18	2		17	18	73100	1	17	17	18
18	18	1		18	18	72000	1	18	18	19
19	19	5		19	19	72800	1	19	19	19
20	19	4		20	19	73000	1	20	20	15
21	19	6		21	19	72900	1			
22	20	1		22	20	72500	1			
23	20	5		23	20	72100	1			
24	20	6		24	20	72400	1			

Hình 6: Bảng can_teach và can_teach_at và belongs_to.

7. Bảng certificate và certificate_type:

	cert_id	cert_year	cert_grade	tutor_id	ct_id		ct_id	ct_name
1	1245783	2021	800	14	2		1	IELTS
2	2937461	2021	900	16	4		2	TOEIC
3	4061835	2021	850	19	5		3	TOEFL
4	5379218	2024	835	20	5		4	MOS
5	5684317	2020	8	16	1		5	IC3
6	7129483	2022	915	17	2			
7	8473925	2020	900	15	4			
8	8596742	2023	550	19	3			
9	9356712	2022	7	14	1			

Hình 7: Bảng certificate và certificate_type.



8. Bảng can_teach_type:

	tutor_id	class_type_id
1	13	1
2	13	2
3	13	3
4	13	4
5	13	5
6	14	6
7	14	7
8	14	8
9	14	9
10	15	10
11	15	11
12	15	12
13	15	14
14	16	13
15	16	14
16	17	11
17	17	12
18	17	13
19	18	1
20	18	2
21	18	3
22	19	8
23	19	9
24	19	10
25	19	14
26	20	4
27	20	5
28	20	6
29	20	7
30	20	14

Hình 8: Bảng can_teach_type.

9. Bảng degree:

	degr_id	degr_major_name	degr_type	degr_year	tutor_id	edi_code
1	4424	Ky thuat co dien tu	Ky su	2020	17	1
2	12806	Marketing	Cu nhan	2020	15	3
3	18053	Giao duc Tieu hoc	Cu nhan	2021	13	4
4	75102	Su pham Toan	Cu nhan	2019	14	4
5	107224	Cong nghe thong tin	Cu nhan	2022	20	6
6	145248	Ngon ngu Anh	Cu nhan	2022	16	7
7	202173	Kinh te doi ngoai	Cu nhan	2018	16	2
8	1677907	Su pham cong nghe	Thac si	2019	19	5

Hình 9: Bảng degree.

10. Bảng has và educational_institution:

	tutor_id	edi_code		edi_code	edi_name
1	13	4	1	1	truong Dai hoc Bach khoa Ha Noi
2	14	4	2	2	truong Dai hoc Ngoai thuong
3	15	3	3	3	truong Dai hoc Hoa sen
4	16	2	4	4	truong Dai hoc Su pham Thanh pho Ho Chi Minh
5	16	7	5	5	truong Dai hoc Su pham Ky thuat TP.HCM
6	17	1	6	6	truong Dai hoc Cong nghe TP. Ho Chi Minh
7	19	5	7	7	truong Dai hoc Khoa hoc Xa hoi va Nhan van
8	20	6			

Hình 10: Bảng has và educational_institution.



11. Bảng class_type và district_city:

	class_type_id	class_type_name	dist_city_id	name	pro_id
1	1	Lop 1	71000	quan 1	1
2	2	Lop 2	71100	quan 2	1
3	3	Lop 3	71200	quan 9	1
4	4	Lop 4	71300	TP. Thu Duc	1
5	5	Lop 5	71400	quan Go Vap	1
6	6	Lop 6	71500	quan 12	1
7	7	Lop 7	71900	quan Binh Tan	1
8	8	Lop 8	72000	quan Tan Phu	1
9	9	Lop 9	72100	quan Tan Binh	1
10	10	Lop 10	72200	quan Phu Nhuan	1
11	11	Lop 11	72300	quan Binh Thanh	1
12	12	Lop 12	72400	quan 3	1
13	13	Luyen thi Dai hoc	72500	quan 10	1
14	14	On luyen Tin hoc	72600	quan 11	1
			72700	quan 5	1
			72800	quan 4	1
			72900	quan 7	1
			73000	quan 8	1
			73100	quan 6	1

Hình 11: Bảng class_type và district_city.

12. Bảng has_class_type và is_held_on:

	class_id	class_type_id	class_id	week_id	slot_id
1	1	4	1	4	3
2	2	8	2	2	7
3	3	10	3	6	1
4	4	14	4	1	5
5	5	11	5	5	2
6	6	2	6	7	6
7	7	9	7	3	4
8	8	13	8	2	6
9	9	8	9	6	7
10	10	12	10	5	1

Hình 12: Bảng has_class_type và is_held_on.



13. Bảng staff:

	date_of_birth	national_id	place_of_origin	profile_photo_url	staff_id
1	1988-05-20 ...	079221762419	TP. Ho Chi Minh	NULL	8
2	1990-11-19 ...	079003881543	Can Tho	NULL	9
3	1995-01-11 ...	079393792382	Hai Phong	NULL	10
4	1997-07-09 ...	079092196280	TP. Ho Chi Minh	NULL	11
5	1985-08-31 ...	079579529749	Da Nang	NULL	12
6	1999-07-29 ...	079618930726	Lam Dong	NULL	13
7	1997-07-07 ...	079465472370	Binh Duong	NULL	14
8	1998-06-12 ...	079930963421	Long An	NULL	15
9	1996-02-22 ...	079170315644	Ha Noi	NULL	16
10	1997-10-20 ...	079012367970	Ha Noi	NULL	17
11	2003-07-08 ...	079996259335	Ben Tre	NULL	18
12	1994-05-06 ...	079295242993	An Giang	NULL	19
13	2000-06-01 ...	079534653344	Quang Nam	NULL	20

Hình 13: Bảng staff.

14. Bảng student và subject:

	stu_grade	stu_school	student_id			subject_id	subject_name
1	4	truong Tieu hoc Quoc te A Chau	1			1	Ngu Van
2	8	truong THCS Ngo Tat To	2			2	Toan
3	10	truong THPT Nguyen Huu Huan	3			3	KHXH
4	12	truong THPT Tran Phu	4			4	Tieng Anh
5	11	truong THCS THPT Tran Dai Nghia	5			5	KHTN
6	2	truong THPT Le Trong Tan	6			6	Tin Hoc
7	9	truong THPT Le Thanh Ton	7				

Hình 14: Bảng student và subject.

15. Bảng teaching_application và qualification:

	teaching_appl_id	application_status	date_of_crea...	class_id	tutor_id	qualification	tutor_id
1	1	chap nhan	2022-06-15 ...	3	15	Giang vien	13
2	2	chap nhan	2023-05-27 ...	7	19	Giang vien	14
3	3	tu choi	2023-06-05 ...	9	19	Sinh vien da tot nghiep	15
4	4	chap nhan	2023-06-15 ...	9	19	Sinh vien da tot nghiep	16
5	5	chap nhan	2024-09-07 ...	10	15	Sinh vien da tot nghiep	17
						Sinh vien da tot nghiep	20
						Sinh vien Dai hoc	18
						Thac si	19

Hình 15: Bảng teaching_application và qualification.



16. Bảng wants_style và tutor_review:

	cq_id	ts_id	tutor_review_id	comment	rate	time_stamp	class_id	student_id
1	1	1	1	Giang hay, de hieu, luon dung gio.	5	2022-10-11 ...	1	1
2	2	1	2	Dat muc van dung, chua hieu VDC.	4	2022-12-20 ...	2	2
3	3	2	3	Hieu nhung hon chan, giang hon cham.	3	2023-09-30 ...	4	4
4	4	1	4	Rat hop, hy vong duoc hoc giao vien nay tiep tuc.	5	2023-10-19 ...	5	5
5	5	1	5	Nhiet tinh, than thien, cham soan tai lieu.	5	2024-05-11 ...	6	6

Hình 16: Bảng wants_style và tutor_review.

17. Bảng wants_subject và s_wants_subject và voucher:

cq_id	subject_id	ta_id subject_id			vou_id	vou_discount	vou_status	tutor_id
		1	2	3				
1	1	1			1	0.10	da su dung	13
2	1	3			2	0.05	chua su dung	14
3	1	2			3	0.15	chua su dung	15
4	2	2			4	0.20	chua su dung	16
5	2	4			5	0.10	da su dung	18
6	2	5			6	0.15	chua su dung	17
7	3	3			7	0.05	chua su dung	19
8	3	4			8	0.20	chua su dung	20
9	3	6						
10	4	4						
11	4	1						
12	4	5						
13	5	3						
14	5	2						
15	5	1						

Hình 17: Bảng wants_subject và s_wants_subject và voucher.

18. Bảng time_slot và week_day và wants_type:

	slot_id	end_time	start_time	week_id	name	cq_id	class_type_id
1	1	09:00:00....	07:00:00....	1	Thu hai	1	4
2	2	11:00:00....	09:00:00....	2	Thu ba	2	8
3	3	13:00:00....	11:00:00....	3	Thu tu	3	13
4	4	15:00:00....	13:00:00....	4	Thu nam	4	14
5	5	17:00:00....	15:00:00....	5	Thu sau	5	11
6	6	19:00:00....	17:00:00....	6	Thu bay	6	2
7	7	21:00:00....	19:00:00....	7	Chu nhat		

Hình 18: Bảng time_slot và week_day và wants_type.



19. Bảng tutor_application và teaching_style:

	ta_id	requirement	ta_status	addr_id	student_id	ts_id	tutor_id	ts_id	ts_name
1	1	Tot nghiep dai hoc chuyen nganh lien quan.	da mo lop	2	1	1	13	1	Truc tiep
2	2	It nhiet 1-2 nam kinh nghiem day kem hoac giang d...	da mo lop	3	2	1	14	2	Truc tuyen
3	3	Tot nghiep dai hoc chuyen nganh lien quan.	da huy bo	6	5	1	15		
4	4	Lich trinh linh hoạt, có thể làm việc vào cuối tuần ho...	da mo lop	6	5	1	17		
5	5	Thanh thao các công cụ và nền tảng dạy học trực t...	chua xu ly	7	6	1	18		

Hình 19: Bảng tutor_application và teaching_style.

20. Bảng has_subject và ward:

	class_id	subject_id		ward_id	ward_name	dist_city_id	pro_id
1	1	1	1	71006	phuong Ben Nghe	71000	1
2	1	3	2	71009	phuong Ben Thanh	71000	1
3	1	2	3	71011	phuong Cau Ong Lanh	71000	1
4	2	2	4	71112	phuong Binh Khanh	71100	1
5	2	5	5	71214	phuong Long Phuoc	71200	1
6	3	3	6	71107	phuong Thao Dien	71300	1
7	3	2	7	71306	phuong Binh Tho	71300	1
8	4	6	8	71516	phuong An Phu Dong	71500	1
9	5	4	9	71915	phuong Tan Tao	71900	1
10	5	1	10	72014	phuong Hiep Tan	72000	1
11	5	5	11	72206	phuong 11	72200	1
12	6	3	12	72210	phuong 2	72200	1
13	6	2	13	72312	phuong 12	72300	1
14	6	1	14	72414	phuong 9	72400	1
15	7	5	15	72510	phuong 12	72500	1
16	7	4	16	72619	phuong 13	72600	1
17	8	3	17	72911	phuong Tan Kieng	72900	1
18	8	4	18	73011	phuong 12	73000	1
19	9	4	19	73107	phuong 2	73100	1
20	10	2					
21	10	3					

Hình 20: Bảng has_subject và ward.

21. Bảng tutor:

	bio	date_joined	inviting_code	n_of_invitations	rate	tutor_id	invited_code
1	Gia su voi 5 nam kinh nghiem, giup hoc sinh phat huy toi da kha n...	2022-02-04 ...	A3B2C1	1	5	13	NULL
2	Giang day tan tam voi tung hoc sinh, cam ket mang cao kien thuc ...	2022-03-08 ...	D7E8F5	0	4	14	NULL
3	Gia su giup hoc vien tu tin giao tiep va cai thien diem so nhanh ch...	2022-06-10 ...	X1Y4Z9	0	NULL	15	NULL
4	Voi niem dam me day hoc, toi cam ket mang den nhung buoi hoc t...	2022-07-08 ...	Q2W7R6	0	3	16	NULL
5	Voi phong cach day hoc sang tao va de hieu,toi giup hoc sinh vuo...	2023-02-20 ...	M9N3P8	0	5	17	NULL
6	Toi la gia su luon tao moi truong hoc tap thoai mai,giup hoc sinh t...	2023-03-17 ...	V5B1K4	0	5	18	A3B2C1
7	Gia su giao kinh nghiem, san sang dong hanh cung hoc sinh tren ...	2023-05-04 ...	T2A9L3	0	NULL	19	NULL
8	Voi phuong phap day hoc linh hoạt,toi giup hoc sinh cai thien die...	2024-09-20 ...	J8H7W2	0	NULL	20	NULL

Hình 21: Bảng tutor.



22. Bảng user:

	user_id	full_name	password	phone_number	user_sex
1	1	Tran Thanh	TranThanh123	0905123457	Nam
2	2	Nguyen Hoang Anh	HoangAnh456	0937456789	Nam
3	3	Nguyen Phong	NguyenPhong789	0968321654	Nam
4	4	Pham Minh Thu	MinhThu2024	0912987654	Nu
5	5	Le Khanh Linh	KhanhLinh098	0987654321	Nu
6	6	Dang Bao Tram	BaoTram567	0923123789	Nu
7	7	Thai Duong	ThaiDuong1234	0945678912	Nam
8	8	Vu Quynh Anh	QuynhAnh9876	0978234567	Nu
9	9	Tran Kim Ngan	KimNgan321	0907876543	Nu
10	10	Nguyen Trung Khoa	TrungKhoa999	0936543210	Nam
11	11	Nguyen Hoang Hieu	HieuHoang555	0962345678	Nam
12	12	Le Ngoc Anh	NgocAnh4321	0918765432	Nu
13	13	Nguyen Viet Anh	VietAnh777	0981234567	Nam
14	14	Pham Linh Nguyen	LinhNguyen234	0928654321	Nu
15	15	Tran Khanh	KhangTran666	0949876543	Nam
16	16	Nguyen Minh Hoang	MinhHoangabc1	0973210987	Nam
17	17	Nguyen Duy	DuyNguyen789	0909543678	Nam
18	18	Tran Tuan Anh	TuanAnh5678	0935678123	Nam
19	19	Vo Thu Trang	ThuTrang9876	0964432109	Nu
20	20	Nguyen Hai Anh	HaiAnh999	0917890123	Nu

Hình 22: Bảng user.

23. Bảng user_contact:

	contact_email	contact_phone_number	social_media_link	user_id
1	baotram567@gmail.com	0923123789	https://facebook.com/baotram567	6
2	duynguyen9999@gmail.com	0909543678	https://facebook.com/duynguyen9999	17
3	haianh9999@gmail.com	0917890123	https://facebook.com/haianh9999	20
4	hieuhoang555@gmail.com	0962345678	https://facebook.com/hieuhoang555	11
5	hoanganh456@gmail.com	0937456789	https://facebook.com/hoanganh456	2
6	khangtran666@gmail.com	0949876543	https://facebook.com/khangtran666	15
7	khanhlinh098@gmail.com	0987654321	https://facebook.com/khanhlinh098	5
8	kimnganabc@gmail.com	0907876543	https://facebook.com/kimnganabc	9
9	linhnguyen4321@gmail.com	0928654321	https://facebook.com/linhnguyen4321	14
10	minhhoangabc@gmail.com	0973210987	https://facebook.com/minhhoangabc	16
11	minhthu2024@gmail.com	0912987654	https://facebook.com/minhthu2024	4
12	ngocanh321@gmail.com	0918765432	https://facebook.com/ngocanh321	12
13	nguyenphong789@gmail.c...	0968321654	https://facebook.com/nguyenphong...	3
14	quynhanh9876@gmail.com	0978234567	https://facebook.com/quynhanh9876	8
15	thaidueung1234@gmail.com	0945678912	https://facebook.com/thaidueung1234	7
16	thanhanh123@gmail.com	0905123457	https://facebook.com/thanhanh123	1
17	thutrang123456@gmail.com	0964432109	https://facebook.com/thutrang123456	19
18	trungkhoa999@gmail.com	0936543210	https://facebook.com/trungkhoa999	10
19	tuananh56789@gmail.com	0935678123	https://facebook.com/tuananh56789	18
20	vietanh7777@gmail.com	0981234567	https://facebook.com/vietanh7777	13

Hình 23: Bảng user_contact.



3 Hiện thực ứng dụng

3.1 Thủ tục INSERT / UPDATE / DELETE dữ liệu vào 1 bảng dữ liệu

Bảng dữ liệu: dbo.class

- Thủ tục INSERT

- Thủ tục `dbo.insert_class` được dùng để insert các thông tin cơ bản của một class
 - Thủ tục được sử dụng trong trường hợp một admin muốn thêm một lớp mới trong hệ thống khi học viên có mong muốn được mở lớp. Thủ tục sẽ được kết hợp với những thủ tục khác trong database trong use-case tạo mới ra một lớp học trong hệ thống. `dbo.insert_class` chỉ thực hiện việc insert trên `dbo.class` nhằm đảm bảo tính chất `separation of concerns` và `modularity`, đảm bảo tầng `application` và `procedure` không bị trùng chức năng. Trong use-case tạo mới một class, `dbo.insert_class` sẽ được gọi cùng với một loạt cách thủ tục khác để có thể tạo mới được một lớp học cùng với những thông tin liên quan đến lớp học đó nhưng nằm ở các bảng khác. Lúc này tầng `application` sẽ "orchestrates" các thủ tục nhằm thực thi một use-case nào đó, đảm bảo được tính chất `reusability` của các procedures.

- Thủ tục bao gồm các thao tác chính sau:

- * Kiểm tra các điều kiện đầu vào trước khi thực hiện câu lệnh `INSERT`: các điều kiện như địa chỉ của lớp học phải có, lớp học phải luôn gắn với học viên, các quy định nghiệp vụ như `class_deposit` phải nhỏ hơn `salary` (tiền cọc phải nhỏ hơn lương, đảm bảo quyền lợi của gia sư), vì một học viên có nhiều địa chỉ, vì vậy địa chỉ của lớp học phải là một trong những địa chỉ của học viên (câu lệnh `IF`)
 - * Sau khi đã kiểm tra các điều kiện, thủ tục thực hiện câu lệnh `INSERT` các dữ liệu vào bảng `dbo.class`
 - * Khi kiểm tra các điều kiện đã nêu, nếu điều kiện bị sai, thủ tục dùng `THROW` và `CATCH` để thông báo cho DBA khi giao tiếp trực tiếp với database hoặc tầng ứng dụng (`application`) thực hiện catch exception và thông báo và xử lý lỗi cho người dùng.

- Input:

- * `@class_deposit` (BIGINT): Số tiền cọc của một lớp học
 - * `@class_status` (VARCHAR(255), default = 'Chưa giao'): Trạng thái hiện tại của lớp
 - * `@commission_fee` (BIGINT): Tiền hoa hồng dành cho hệ thống
 - * `@requirements` (VARCHAR(255)): Các yêu cầu đặc biệt của lớp học dành cho gia sư
 - * `@date_start` (DATETIME2(7)): Ngày bắt đầu của lớp học
 - * `@salary` (BIGINT): Lương dành cho giảng viên
 - * `@addr_id` (BIGINT): ID của địa chỉ mà lớp học được tổ chức (chọn ra từ các địa chỉ của học viên)
 - * `@student_id` (BIGINT): ID của học viên tham gia vào lớp học
 - * `@ts_id` (BIGINT): ID của cách mà lớp học được tổ chức
 - * `@tutor_id` (BIGINT, default = NULL): ID của gia sư gắn với lớp học

- Output:



* `@inserted_class_id` (BIGINT, output): ID trả về sau khi insert lớp học thành công

– Câu lệnh tạo thủ tục:

```
-- Create a new stored procedure called 'insert_class' in schema
'dbo'
-- Drop the stored procedure if it already exists
IF EXISTS (
SELECT *
    FROM INFORMATION_SCHEMA.ROUTINES
WHERE SPECIFIC_SCHEMA = N'dbo'
    AND SPECIFIC_NAME = N'insert_class'
    AND ROUTINE_TYPE = N'PROCEDURE'
)
DROP PROCEDURE dbo.insert_class
GO

CREATE PROCEDURE dbo.insert_class
    @class_deposit      BIGINT,
    @class_status        VARCHAR(255) = 'Chưa giao',
    @commission_fee     BIGINT,
    @requirements       VARCHAR(255),
    @date_start         DATETIME2(7),
    @salary              BIGINT,
    @addr_id             BIGINT,
    @student_id          BIGINT,
    @ts_id               BIGINT,
    @tutor_id             BIGINT = NULL,
    @inserted_class_id   BIGINT OUTPUT
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY

        -- Validation for addr_id
        IF @addr_id IS NULL
            THROW 50001, 'An address is needed', 1;

        -- Validation for student_id
        IF @student_id IS NULL
            THROW 50007, 'Class must have a student', 1;

        -- Validation for simple business logic
        IF @class_deposit > @salary
            THROW 50002, 'The deposit must be less than the
                           salary', 1;

        -- Validation for class_style
        IF @ts_id IS NULL
            THROW 50008, 'Class must have a class style', 1;

        -- Validation that the address of the class
```



```
-- must belong to the student in the class
-- [a student can have multiple address]
IF
(
    @addr_id NOT IN
    (
        -- All addresses of student
        SELECT a.addr_id
        FROM dbo.address a
        WHERE a.user_id = @student_id
    )
)
    THROW 50009, 'Student does not have the address', 1;

INSERT INTO dbo.class
(
    class_deposit, class_status, commission_fee,
    requirements, date_start, salary,
    addr_id, student_id, ts_id, tutor_id
)
-- return the class_id
OUTPUT inserted.class_id

VALUES
(
    @class_deposit, @class_status, @commission_fee,
    @requirements, @date_start, @salary,
    @addr_id, @student_id, @ts_id, @tutor_id
);

END TRY

BEGIN CATCH

    -- For debugging as SA
    SELECT ERROR_LINE() AS [error line],
    ERROR_MESSAGE() AS [message],
    ERROR_NUMBER() AS [number],
    ERROR_PROCEDURE() AS [procedure],
    ERROR_SEVERITY() AS [severity],
    ERROR_STATE() AS [state];

    -- For application
    THROW;
END CATCH
END
```

Listing 46: Câu lệnh tạo thủ tục

- Các thủ tục kèm theo mà tầng ứng dụng sẽ sử dụng trong use-case tạo mới một lớp học hoặc được **reuse** trong use-case update thông tin của một lớp học (có thể update các thông tin được lưu trên những bản kèm theo, ngoại trừ bảng dbo.class)



- * Thêm hoặc cập nhật (reusability) các môn học vào trong lớp

```
CREATE TYPE SUBJECT_ID_LIST AS TABLE
(
    subject_id BIGINT NOT NULL
);

-- Create a new stored procedure called 'insert_has_subject'
-- in schema 'dbo'
-- Drop the stored procedure if it already exists
IF EXISTS
(
    SELECT *
        FROM INFORMATION_SCHEMA.ROUTINES
    WHERE SPECIFIC_SCHEMA = N'dbo'
        AND SPECIFIC_NAME = N'insert_has_subject'
        AND ROUTINE_TYPE = N'PROCEDURE'
)
DROP PROCEDURE dbo.insert_has_subject
GO

-- Create the stored procedure in the specified schema
CREATE PROCEDURE dbo.insert_has_subject
    @class_id BIGINT,
    @subject_id_list SUBJECT_ID_LIST READONLY
AS
BEGIN

    SET NOCOUNT ON;

    BEGIN TRY

        -- can be updated for the update use-case
        -- checking for class_id
        IF NOT EXISTS
        (
            SELECT 1
            FROM dbo.class c
            WHERE c.class_id = @class_id
        )
        THROW 50003, 'Class has been altered or deleted', 1;

        -- checking for the subject_id
        IF EXISTS
        (
            SELECT 1
            FROM @subject_id_list l
            WHERE l.subject_id NOT IN
            (
                SELECT s.subject_id
                FROM dbo.subject s
            )
        )
        INSERT INTO dbo.subject_class(class_id, subject_id)
        SELECT @class_id, l.subject_id
        FROM @subject_id_list l
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;
        ELSE
            IF @@TRANCOUNT = 0
                BEGIN TRANSACTION;
            END;
        RAISERROR (@@TRANCOUNT, 16, 1);
    END CATCH
END;
```



```
)  
)  
THROW 50004, 'One or more subjects do not exist',  
      1;  
  
-- deleting any links  
-- for updating  
DELETE FROM dbo.has_subject  
WHERE class_id = @class_id;  
  
INSERT INTO dbo.has_subject(class_id, subject_id)  
SELECT @class_id, l.subject_id  
FROM @subject_id_list l;  
  
END TRY  
  
BEGIN CATCH  
    -- For debugging as SA  
    SELECT ERROR_LINE() AS [error line],  
    ERROR_MESSAGE() AS [message],  
    ERROR_NUMBER() AS [number],  
    ERROR_PROCEDURE() AS [procedure],  
    ERROR_SEVERITY() AS [severity],  
    ERROR_STATE() AS [state];  
  
    -- For application to show error  
    THROW;  
END CATCH  
END  
GO
```

- * Thêm hoặc cập nhật (reusability) các loại lớp liên quan đến lớp

```
-- TYPE for the input  
CREATE TYPE CLASS_TYPE_ID_LIST AS TABLE  
(  
    class_type_id BIGINT NOT NULL  
);  
GO  
  
  
-- Create a new stored procedure called 'insert_has_class_type'  
-- in schema 'dbo'  
-- Drop the stored procedure if it already exists  
IF EXISTS (  
    SELECT *  
    FROM INFORMATION_SCHEMA.ROUTINES  
    WHERE SPECIFIC_SCHEMA = N'dbo'  
        AND SPECIFIC_NAME = N'insert_has_class_type'  
        AND ROUTINE_TYPE = N'PROCEDURE'  
)  
DROP PROCEDURE dbo.insert_has_class_type  
GO
```



```
-- Create the stored procedure in the specified schema
CREATE PROCEDURE dbo.insert_has_class_type
    @class_id BIGINT,
    @class_type_id_list CLASS_TYPE_ID_LIST READONLY
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        -- can be updated for the update use-case
        -- checking for class_id
        IF NOT EXISTS
        (
            SELECT 1
            FROM dbo.class c
            WHERE c.class_id = @class_id
        )
        THROW 50003, 'Class has been altered or deleted', 1;

        -- checking for class_id_type
        IF EXISTS
        (
            SELECT 1
            FROM @class_type_id_list l
            WHERE l.class_type_id NOT IN
            (
                SELECT ct.class_type_id
                FROM dbo.class_type ct
            )
        )
        THROW 50005, 'One or more class types do not exist', 1;

        -- delete from update
        DELETE FROM dbo.has_class_type
        WHERE class_id = @class_id;

        INSERT INTO dbo.has_class_type(class_id, class_type_id)
        )
        SELECT @class_id, l.class_type_id
        FROM @class_type_id_list l;
    END TRY

    BEGIN CATCH
        -- For debugging as SA
        SELECT ERROR_LINE() AS [error line],
        ERROR_MESSAGE() AS [message],
        ERROR_NUMBER() AS [number],
        ERROR_PROCEDURE() AS [procedure],
        ERROR_SEVERITY() AS [severity],
        ERROR_STATE() AS [state];
    END CATCH

```



```
-- For application to show error
THROW;
END CATCH

END
GO
```

- * Thêm hoặc cập nhật (reusability) các ngày và giờ vào trong lớp học

```
CREATE TYPE TIME_LIST AS TABLE
(
    week_id BIGINT NOT NULL ,
    slot_id BIGINT NOT NULL
);
GO

-- Create a new stored procedure called 'insert_is_held_on' in
-- schema 'dbo'
-- Drop the stored procedure if it already exists
IF EXISTS (
    SELECT *
        FROM INFORMATION_SCHEMA.ROUTINES
    WHERE SPECIFIC_SCHEMA = N'dbo'
        AND SPECIFIC_NAME = N'insert_is_held_on'
        AND ROUTINE_TYPE = N'PROCEDURE'
)
DROP PROCEDURE dbo.insert_is_held_on
GO

-- Create the stored procedure in the specified schema
CREATE PROCEDURE dbo.insert_is_held_on
    @class_id BIGINT,
    @time_list TIME_LIST READONLY
-- add more stored procedure parameters here
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        -- can be updated for the update use-case
        -- checking for class_id
        IF NOT EXISTS
            (
                SELECT 1
                FROM dbo.class c
                WHERE c.class_id = @class_id
            )
        THROW 50003, 'Class has been altered or deleted',
        1;

        -- checking for slot_id
```



```
IF EXISTS
(
    SELECT 1
    FROM @time_list l
    WHERE l.week_id NOT IN
    (
        SELECT w.week_id
        FROM dbo.week_day w
    )
)
OR
EXISTS
(
    SELECT 1
    FROM @time_list l
    WHERE l.slot_id NOT IN
    (
        SELECT t.slot_id
        FROM dbo.time_slot t
    )
)
THROW 50006, 'Weekdays or times are not allowed to
choose', 1;

-- delete from update
DELETE FROM dbo.is_held_on
WHERE class_id = @class_id;

INSERT INTO dbo.is_held_on
SELECT @class_id, t.week_id, t.slot_id
FROM @time_list t;

END TRY

BEGIN CATCH
    -- For debugging as SA
    SELECT ERROR_LINE() AS [error line],
    ERROR_MESSAGE() AS [message],
    ERROR_NUMBER() AS [number],
    ERROR_PROCEDURE() AS [procedure],
    ERROR_SEVERITY() AS [severity],
    ERROR_STATE() AS [state];

    -- For application to show error
    THROW;
END CATCH
END
GO
```

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc tạo thủ tục
- Các câu lệnh thực thi thủ tục trường hợp báo lỗi:
 - * **Thiếu địa chỉ** (addr_id không được cung cấp): Thủ tục sẽ báo lỗi nếu không có



The screenshot shows the SQL Server Management Studio interface. In the Object Explorer on the left, under the 'Programmability' node, 'Stored Procs.' is selected. Inside it, there is a single item named 'dbo.insert_class'. The central pane displays the script for creating this stored procedure. The code includes logic to drop the procedure if it exists, select specific schema information, and then create the procedure itself with parameters for class deposit, status, commission fee, requirements, date start, salary, address ID, student ID, time ID, tutor ID, and inserted class ID. The 'Messages' pane at the bottom shows successful execution messages from the database.

```
IF EXISTS (
    SELECT *
    FROM INFORMATION_SCHEMA.ROUTINES
    WHERE SPECIFIC_SCHEMA = N'dbo'
        AND SPECIFIC_NAME = N'insert_class'
        AND ROUTINE_TYPE = N'PROCEDURE'
)
DROP PROCEDURE dbo.insert_class
GO

CREATE PROCEDURE dbo.insert_class
(
    @class_deposit BIGINT,
    @class_status VARCHAR(255) = 'Chưa giao',
    @commission_fee BIGINT,
    @requirements VARCHAR(255),
    @date_start DATETIME,
    @salary BIGINT,
    @addr_id BIGINT,
    @student_id BIGINT,
    @ts_id BIGINT,
    @tutor_id BIGINT = NULL,
    @inserted_class_id BIGINT OUTPUT
)
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        BEGIN TRANSACTION;
        -- Insert logic here
        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;
        -- Error handling logic here
    END CATCH;
END;
```

Hình 24: Kết quả tạo thủ tục

addr_id, vì địa chỉ của lớp học là bắt buộc.

```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
@class_deposit = 500000,
@class_status = 'Chưa giao',
@commission_fee = 100000,
@requirements = 'Gia su phai la gia su nam, la sinh vien
truong Dai hoc Bach Khoa',
@date_start = '2024-07-12',
@salary = 800000,
@addr_id = NULL,
@student_id = 2,
@ts_id = 1,
@tutor_id = NULL,
@inserted_class_id = @class_id;
GO
```

- * Thiếu thông tin học viên (student_id không được cung cấp): Thủ tục yêu cầu phải có student_id để xác định học viên liên quan đến lớp học.

```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
@class_deposit = 500000,
@class_status = 'Chưa giao',
@commission_fee = 100000,
```



```
@requirements = 'Gia su phai la gia su nam, la sinh vien
    truong Dai hoc Bach Khoa',
@date_start = '2024-07-12',
@salary = 800000,
@addr_id = 1,
@student_id = NULL,
@ts_id = 1,
@tutor_id = NULL,
@inserted_class_id = @class_id;
GO
```

- * **Tiền đặt cọc lớn hơn lương (class_deposit > salary):** Thủ tục sẽ báo lỗi nếu tiền đặt cọc lớn hơn mức lương đã thỏa thuận.

```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
@class_deposit = 500000,
@class_status = 'Chưa giao',
@commission_fee = 100000,
@requirements = 'Gia su phai la gia su nam, la sinh vien
    truong Dai hoc Bach Khoa',
@date_start = '2024-07-12',
@salary = 400000,
@addr_id = 1,
@student_id = 1,
@ts_id = 1,
@tutor_id = NULL,
@inserted_class_id = @class_id;
GO
```

- * **Lớp học không có phong cách dạy (style không được cung cấp):** Lỗi xảy ra nếu phong cách dạy của lớp không được xác định.

```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
@class_deposit = 500000,
@class_status = 'Chưa giao',
@commission_fee = 100000,
@requirements = 'Gia su phai la gia su nam, la sinh vien
    truong Dai hoc Bach Khoa',
@date_start = '2024-07-12',
@salary = 800000,
@addr_id = 1,
@student_id = 1,
@ts_id = NULL,
@tutor_id = NULL,
@inserted_class_id = @class_id;
GO
```

- * **Địa chỉ không hợp lệ (addr_id không thuộc danh sách địa chỉ của học viên):** Thủ tục sẽ báo lỗi nếu địa chỉ được chèn không nằm trong danh sách địa chỉ của học viên liên quan.



```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
@class_deposit = 500000,
@class_status = 'Chưa giao',
@commission_fee = 100000,
@requirements = 'Gia sư phải là gia sư nam, là sinh viên trường Đại học Bách Khoa',
@date_start = '2024-07-12',
@salary = 800000,
@addr_id = 4 ,
@student_id = 1,
@ts_id = 2,
@tutor_id = NULL,
@inserted_class_id = @class_id;
GO
```

Results Messages

```
9:47:50 PM Started executing query at line 128
Msg 50001, Level 16, State 1, Procedure dbo.insert_class, Line 23
An address is needed
Total execution time: 00:00:00.076
```

Hình 25: Kết quả khi địa chỉ lớp học không được cung cấp

```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
@class_deposit = 500000,
@class_status = 'Chưa giao',
@commission_fee = 100000,
@requirements = 'Gia sư phải là gia sư nam, là sinh viên
    trường Đại học Bách Khoa',
@date_start = '2024-07-12',
@salary = 800000,
@addr_id = 4 ,
@student_id = 1,
@ts_id = 2,
@tutor_id = NULL,
@inserted_class_id = @class_id;
GO
```

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc kiểm tra thủ tục:
- Các câu lệnh thực thi thủ tục trường hợp đúng:
 - * Trường hợp đúng 1:

```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
@class_deposit = 500000,
@class_status = 'Chưa giao',
@commission_fee = 100000,
```



```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
@class_deposit = 500000,
@class_status = 'Chưa giao',
@commission_fee = 100000,
@requirements = 'Giá su phai là giá su nam, là sinh viên trường Đại học Bách Khoa',
@date_start = '2024-07-12',
@salary = 800000,
@addr_id = 1,
@ts_id = 1,
@tutor_id = NULL,
@inserted_class_id = @class_id;
```

Results Messages

```
9:50:55 PM Started executing_query_at Line 128
Msg 50007, Level 16, State 1, Procedure dbo.insert_class, Line 27
Class must have a student
Total execution time: 00:00:00.026
```

Hình 26: Kết quả khi lớp học không có học viên

```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
@class_deposit = 500000,
@class_status = 'Chưa giao',
@commission_fee = 100000,
@requirements = 'Giá su phai là giá su nam, là sinh viên trường Đại học Bách Khoa',
@date_start = '2024-07-12',
@salary = 400000,
@addr_id = 1,
@ts_id = 1,
@tutor_id = NULL,
@inserted_class_id = @class_id;
```

Results Messages

```
9:54:07 PM Started executing_query_at Line 128
Msg 50002, Level 16, State 1, Procedure dbo.insert_class, Line 31
The deposit must be less than the salary
Total execution time: 00:00:00.032
```

Hình 27: Kết quả khi tiền đặt cọc lớn hơn lương



The screenshot shows the SQL Server Management Studio interface. In the center pane, there is a code editor with the following SQL script:

```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
    @class_deposit = 500000,
    @class_status = 'Chưa giao',
    @commission_fee = 100000,
    @date_start = '2024-07-12',
    @salary = 800000,
    @student_id = 1,
    @tutor_id = NULL,
    @inserted_class_id = @class_id;
```

Below the code editor is a results grid titled "Results". It contains one row with the following data:

error line	message	number	procedure	severity	state
1	Class must have a class style	50008	dbo.insert_class	16	1

The status bar at the bottom of the screen shows the following information: Ln 142, Col 3, Spaces: 4, UTF-8, CRLF, 1 rows, Choose SQL Language, 00:00:00, localhost: Ass2_CO2013, 9:56 PM, ENG, 12/8/2024.

Hình 28: Kết quả khi lớp học không có cách dạy

The screenshot shows the SQL Server Management Studio interface. In the center pane, there is a code editor with the same SQL script as in Figure 28:

```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
    @class_deposit = 500000,
    @class_status = 'Chưa giao',
    @commission_fee = 100000,
    @date_start = '2024-07-12',
    @salary = 800000,
    @student_id = 1,
    @tutor_id = NULL,
    @inserted_class_id = @class_id;
```

Below the code editor is a results grid titled "Results". It contains one row with the following data:

error line	message	number	procedure	severity	state
10:00:20 PM	Started executing query at Line 128 Msg 50009, Level 16, State 1, Procedure dbo.insert_class, Line 50 Student does not have the address Total execution time: 00:00:00.045				

The status bar at the bottom of the screen shows the following information: Ln 140, Col 18, Spaces: 4, UTF-8, CRLF, 1 rows, Choose SQL Language, 00:00:00, localhost: Ass2_CO2013, 10:00 PM, ENG, 12/8/2024.

Hình 29: Kết quả khi địa chỉ lớp học không phải là địa chỉ của học viên trong lớp học



```
@requirements = 'Gia su phai la gia su nam, la sinh vien  
truong Dai hoc Bach Khoa',  
@date_start = '2024-07-12',  
@salary = 800000,  
@addr_id = 1 ,  
@student_id = 1,  
@ts_id = 2,  
@tutor_id = NULL ,  
@inserted_class_id = @class_id;  
GO
```

* Trường hợp đúng 2:

```
-- example to execute the stored procedure we just created  
DECLARE @class_id BIGINT  
EXECUTE dbo.insert_class  
@class_deposit = 100000,  
@class_status = 'Chưa giao',  
@commission_fee = 100000,  
@requirements = 'Gia su phai la hoc sinh DHQG-HCM',  
@date_start = '2024-07-12',  
@salary = 1000000,  
@addr_id = 3 ,  
@student_id = 3,  
@ts_id = 1,  
@tutor_id = NULL ,  
@inserted_class_id = @class_id;  
GO
```

* Trường hợp đúng 3:

```
-- example to execute the stored procedure we just created  
DECLARE @class_id BIGINT  
EXECUTE dbo.insert_class  
@class_deposit = 100000,  
@class_status = 'Chưa giao',  
@commission_fee = 100000,  
@requirements = 'Gia su phai la cuu hoc sinh truong chuyen',  
@date_start = '2024-07-12',  
@salary = 1000000,  
@addr_id = 5 ,  
@student_id = 5,  
@ts_id = 1,  
@tutor_id = NULL ,  
@inserted_class_id = @class_id;  
GO
```

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc thực hiện kiểm tra thủ tục

- Thủ tục UPDATE

- Mô tả thủ tục: thủ tục update_class được sử dụng để update các thông tin trong class. Thủ tục được sử dụng trong use-case chỉnh sửa các thông tin liên quan đến một lớp học trong hệ thống (bởi admin của hệ thống)



The screenshot shows the SQL Server Management Studio interface. The left sidebar displays the database structure under 'SERVERS'. The main pane shows a script window with the following SQL code:

```
128 -- example to execute the stored procedure we just created
129 DECLARE @class_id BIGINT
130 EXECUTE dbo.insert_class
131 @class_deposit = 100000,
132 @class_status = 'Chưa giao',
133 @commission_fee = 100000,
134 @requirements = 'Gia sư phải là giáo viên nam, là sinh viên trường Đại học Bách Khoa',
135 @date_start = '2024-07-12',
136 @salary = 800000,
137 @student_id = 1,
138 @tutor_id = 1,
139 @ts_id = 2,
140 @tutor_id = NULL,
141 @inserted_class_id = @Class_id;
142 GO
```

The results pane shows a single row of data:

class_id
10018

Hình 30: Thủ tục thực hiện thành công

The screenshot shows the SQL Server Management Studio interface. The left sidebar displays the database structure under 'SERVERS'. The main pane shows a script window with the following SQL code:

```
125 WHERE class_id > 10;
```

The results pane shows a single row of data:

class_id
10019

Hình 31: Thủ tục thực hiện thành công



```
-- example to execute the stored procedure we just created
DECLARE @class_id BIGINT
EXECUTE dbo.insert_class
@class_deposit = 100000,
@class_status = 'Chưa giao',
@commission_fee = 100000,
@requirements = 'Gia sư phải là sinh viên trường chuyên',
@date_start = '2024-07-12',
@salary = 1000000,
@gaddr_id = 5 ,
@student_id = 5,
@ts_id = 1,
@tutor_id = NULL,
@Inserted_class_id = @class_id;
```

class_id
10020

Hình 32: Thủ tục thực hiện thành công

class_id	class_deposit	class_status	commission_fee	requirements	date_start	salary	addr
1	200000	Đã giao	NULL	NULL	2022-10-07 00:00:00.0000000	2000000	2
2	2	NULL	Da giao	300000	2022-08-20 00:00:00.0000000	3000000	3
3	3	200000	Da giao	NULL	2022-05-30 00:00:00.0000000	2000000	4
4	4	NULL	Da giao	100000	2023-05-30 00:00:00.0000000	1000000	5
5	5	NULL	Da giao	400000	2023-06-19 00:00:00.0000000	4000000	6
6	6	NULL	Da giao	300000	2024-07-05 00:00:00.0000000	3000000	7
7	7	300000	Da giao	NULL	2024-06-08 00:00:00.0000000	3000000	8
8	8	NULL	Chưa giao	100000	NULL	1000000	9
9	9	200000	Chưa giao	NULL	NULL	2000000	10
10	10	400000	Chưa giao	NULL	NULL	4000000	11
11	10018	500000	Chưa giao	1000000	Gia sư phải là sinh viên nam, là sinh viên trường Đại học Bách Khoa	2024-07-12 00:00:00.0000000	8000000
12	10019	100000	Chưa giao	100000	Gia sư phải là học sinh DHCN-HCM	2024-07-12 00:00:00.0000000	1000000
13	10020	100000	Chưa giao	100000	Gia sư phải là sinh viên trường Chuyên	2024-07-12 00:00:00.0000000	1000000

Hình 33: Lớp học sau khi đã thực hiện 3 thủ tục trên



– Thủ tục bao gồm các thao tác chính sau đây:

- * Vì đây là thủ tục thực hiện update, đầu tiên thủ tục sẽ bắt đầu một transaction
- * Thủ tục sẽ tiến hành gọi câu lệnh UPDATE để tiến hành update bảng **class** với sự hỗ trợ của hàm ISNULL() với ý nghĩa rằng khi các parameter được đưa vào, nếu có giá trị là NULL thì sẽ sử dụng lại giá trị cũ, do đó cho phép người dùng thay đổi **class** "partially". Ngoài ra, các điều kiện nghiệp vụ đã được kiểm tra bởi các CHECK constraint và các trigger, do đó, việc validate dữ liệu là không cần thiết trong trường hợp này
- * Nếu với exception nào xảy ra trong khi thực thi thủ tục, hay các điều kiện ứng với nghiệp vụ không thoái mãn do CHECK constraint hoặc là TRIGGER câu lệnh ROLLBACK TRANSACTION trong phần BEGIN CATCH sẽ tiến hành roll back lại những gì đã thực hiện, ngăn chặn việc update các thông tin không đúng với yêu cầu nghiệp vụ

– Input:

- * @class_id (BIGINT):
 - . **Description:** ID của class cần thực hiện update
 - . **Default Value:** (None)
 - . **Required / Optional:** Required
- * @class_deposit (BIGINT, default = NULL):
 - . **Description:** tiền đặt cọc mới
 - . **Default Value:** NULL
 - . **Required / Optional:** Optional
- * @class_status (VARCHAR(255), default = NULL):
 - . **Description:** trạng thái của lớp học
 - . **Default Value:** NULL
 - . **Required / Optional:** Optional
- * @commission_fee (BIGINT, default = NULL):
 - . **Description:** tiền hoa hồng dành cho hệ thống
 - . **Default Value:** NULL
 - . **Required / Optional:** Optional
- * @requirements (VARCHAR(255), default = NULL):
 - . **Description:** các yêu cầu thêm gắn liền với lớp học
 - . **Default Value:** NULL
 - . **Required / Optional:** Optional
- * @date_start (DATETIME2(7), default = NULL):
 - . **Description:** ngày bắt đầu của lớp học
 - . **Default Value:** NULL
 - . **Required / Optional:** Optional
- * @salary (BIGINT, default = NULL):
 - . **Description:** lương của lớp học
 - . **Default Value:** NULL
 - . **Required / Optional:** Optional
- * @addr_id (BIGINT, default = NULL):



- **Description:** ID của address mà lớp học được tổ chức
- **Default Value:** NULL
- **Required / Optional:** Optional
- * @student_id (BIGINT, default = NULL):
 - **Description:** ID của học viên gắn với lớp học
 - **Default Value:** NULL
 - **Required / Optional:** Optional
- * @ts_id (BIGINT, default = NULL):
 - **Description:** ID của style gắn với lớp học
 - **Default Value:** NULL
 - **Required / Optional:** Optional
- * @tutor_id (BIGINT, default = NULL):
 - **Description:** ID của gia sư gắn với lớp học
 - **Default Value:** NULL
 - **Required / Optional:** Optional
- Output: NONE
- Câu lệnh tạo thủ tục:

```
-- Create a new stored procedure called 'update_class' in schema
'dbo'
-- Drop the stored procedure if it already exists
IF EXISTS (
SELECT *
    FROM INFORMATION_SCHEMA.ROUTINES
WHERE SPECIFIC_SCHEMA = N'dbo'
    AND SPECIFIC_NAME = N'update_class'
    AND ROUTINE_TYPE = N'PROCEDURE'
)
DROP PROCEDURE dbo.update_class
GO

-- Create the stored procedure in the specified schema
CREATE PROCEDURE update_class
    @class_id      BIGINT,
    @class_deposit  BIGINT = NULL,           -- Default NULL
        allows optional update
    @class_status   VARCHAR(255) = NULL,
    @commission_fee BIGINT = NULL,
    @requirements   VARCHAR(255) = NULL,
    @date_start     DATETIME2(7) = NULL,
    @salary         BIGINT = NULL,
    @addr_id        BIGINT = NULL,           -- This is NOT NULL,
        so optional update logic is important
    @student_id     BIGINT = NULL,           -- This is NOT NULL,
        so optional update logic is important
    @ts_id          BIGINT = NULL,           -- This is NOT NULL,
        so optional update logic is important
    @tutor_id        BIGINT = NULL
AS
```



```
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;
        -- Update the class
        -- ISNULL() is used for optional update
        UPDATE dbo.class
        SET
            class_deposit = ISNULL(@class_deposit,
            class_deposit),
            class_status = ISNULL(@class_status,
            class_status),
            commission_fee = ISNULL(@commission_fee,
            commission_fee),
            requirements = ISNULL(@requirements,
            requirements),
            date_start = ISNULL(@date_start, date_start),
            salary = ISNULL(@salary, salary),
            addr_id = ISNULL(@addr_id, addr_id),
            student_id = ISNULL(@student_id, student_id),
            ts_id = ISNULL(@ts_id, ts_id),
            tutor_id = ISNULL(@tutor_id, tutor_id)
        WHERE class_id = @class_id;

        -- If no rows were affected, class_id might not exist
        IF @@ROWCOUNT = 0
        BEGIN
            ROLLBACK TRANSACTION;
            RETURN;
        END

        -- Commit the transaction
        COMMIT TRANSACTION;
    END TRY

    BEGIN CATCH
        -- Roll back the transaction if any error occurs
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;

        -- For debugging as SA
        SELECT ERROR_LINE() AS [error line],
        ERROR_MESSAGE() AS [message],
        ERROR_NUMBER() AS [number],
        ERROR_PROCEDURE() AS [procedure],
        ERROR_SEVERITY() AS [severity],
        ERROR_STATE() AS [state];

        -- For application
        THROW;
    END CATCH

```



```
CREATE PROCEDURE update_class
    @class_id BIGINT,
    @class_deposit BIGINT = NULL,          -- Default NULL allows optional update
    @class_status VARCHAR(255) = NULL,
    @commission_fee INT = NULL,
    @requirements VARCHAR(255) = NULL,
    @date_start DATETIME2(7) = NULL,
    @salary BIGINT = NULL,
    @addr_id BIGINT = NULL,                -- This is NOT NULL, so optional update logic is important
    @student_id BIGINT = NULL,             -- This is NOT NULL, so optional update logic is important
    @ts_id BIGINT = NULL,                  -- This is NOT NULL, so optional update logic is important
    @tutor_id BIGINT = NULL

AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        BEGIN TRANSACTION;
        -- Update the class
        -- ISNULL() is used for optional update
        UPDATE dbo.class
    END TRY
    BEGIN CATCH
        -- Catch any errors
    END CATCH
END;
```

Hình 34: Tạo thủ tục thành công

```
END CATCH
END ;
GO
```

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc tạo thủ tục
- Các câu lệnh thực thi thủ tục trường hợp báo lỗi:

* Trường hợp update class nhưng địa chỉ của lớp học lại không thuộc là một trong các địa chỉ của học viên, do đó câu lệnh sẽ bị lỗi tại trigger, procedure sẽ thực hiện catch lỗi và hiển thị với DBA hoặc tiến hành THROW đến tầng ứng dụng

```
-- example to execute the stored procedure we just created
EXEC update_class
    @class_id = 10018,
    @class_deposit = 500000,
    @class_status = 'Da giao',
    @commission_fee = 250,
    @requirements = 'Advanced Math Required',
    @date_start = '2024-12-15',
    @salary = 1000000,
    @addr_id = 5,
    @student_id = 4,
    @ts_id = 1,
    @tutor_id = 13;
GO
```

* Trường hợp update class nhưng lại vi phạm ràng buộc rằng tiền đặt cọc của giáo sư không được lớn hơn tiền lương của giáo sư



```
-- example to execute the stored procedure we just created
EXEC update_class
    @class_id = 10018,
    @class_deposit = 500000,
    @class_status = 'Da giao',
    @commission_fee = 250,
    @requirements = 'Advanced Math Required',
    @date_start = '2024-12-15',
    @salary = 100000,
    @addr_id = 5,
    @student_id = 4,
    @ts_id = 1,
    @tutor_id = 13;
GO

-- example to execute the stored procedure we just created
EXEC update_class
    @class_id = 10019,
    @class_status = 'Da giao',
    @commission_fee = 250,
    @requirements = 'Advanced Math Required',
    @date_start = '2024-12-15',
    @salary = 100000,
    @addr_id = 5,
    @student_id = 4,
    @ts_id = 1,
    @tutor_id = 13;
SELECT *
FROM dbo.class
WHERE class_id = 10019;

EXECUTE update_class
    @class_id = 10019,
```

error line	message
1	28

Ln 91, Col 1 (392 selected) Spaces: 4 UTF-8 CRLF SQL 1 rows Choose SQL Language 00:00:00 localhost: As2_CO2013 7:26 AM 12/10/2024

Hình 35: Học viên không có địa chỉ nào ứng với địa chỉ mới được cập nhật

```
-- example to execute the stored procedure we just created
EXEC update_class
    @class_id = 10018,
    @class_deposit = 500000,
    @class_status = 'Da giao',
    @commission_fee = 250,
    @requirements = 'Advanced Math Required',
    @date_start = '2024-12-15',
    @salary = 100000,
    @addr_id = 5,
    @student_id = 4,
    @ts_id = 1,
    @tutor_id = 13;
GO
```

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc kiểm tra thủ tục:
- Các câu lệnh thực thi thủ tục trường hợp đúng:

* Trường hợp đúng 1: Chính sửa lại trạng thái của class

```
-- example to execute the stored procedure we just created
EXEC update_class
    @class_id = 10018,
    @class_status = 'Da giao'
GO
```

* Trường hợp đúng 2: Chính sửa lại yêu cầu lớp học, tiền cọc khi vào lớp học, ngày bắt đầu của lớp học



```
File Edit View Help
CONNECTIONS ... Welcome
D:\coding-space>sql-project> update_class.sql - disconnected
update_class.sql - (51) o...finding
SQLQuery_2 - (59) o...finding
SQLQuery_1 - (54) d...finding
trigger_class_address ...
Run Cancel Change Database: Ass2_CO2013
Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook
GO
SELECT *
FROM dbo.class;
GO
-- example to execute the stored procedure we just created
EXEC update_class
@class_id = 10019,
@class_deposit = 200000,
@class_status = 'gioao',
@requirements = 'Advanced Math Required',
@date_start = '2024-12-15',
@salary = 100000,
@addr_id = 5,
@tutor_id = 4,
@cls_id = 1,
@tutor_id = 13;
GO
SELECT *
FROM dbo.class;
Results Messages
error line | message
number | procedure | severity | state |
1 | 24 | The UPDATE statement conflicted with the CHECK constraint... | 547 | update_class | 16 | 0 |
LN 91, Col 1 (88 selected) Spaces: 4 UTF-8 CRLF SQL 1 rows Choose SQL Language 00:00:00 localhost: Ass2_CO2013 12/10/2024 7:29 AM AZURE
```

Hình 36: Tiền đặt cọc đã lớn hơn tiền lương của giáo sư

```
EXECUTE update_class
@class_id = 10019,
@requirements = 'Gia su phai la gia su nu',
@class_deposit = 200000,
@date_start = '2024-12-10'
GO
```

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc thực hiện kiểm tra thủ tục
 - Thủ tục DELETE
 - Mô tả thủ tục: Thủ tục này được sử dụng trong use-case admin mong muốn xóa một lớp học nào đó
 - Thủ tục bao gồm các thao tác chính như:
 - * Kiểm tra xem `class_id` đầu vào của thủ tục có tồn tại trong bảng `class` hay không, nếu không thì sẽ dùng `THROW` thông báo cho DBA hoặc tầng application để thông báo ra lỗi. Các bảng khác nếu có foreign key đến `class` đã được xử lý bằng `ON DELETE CASCADE` do đó trong procedure không cần xử lý gì thêm
 - * Thủ tục sẽ bắt đầu một `TRANSACTION`, nếu có lỗi gì xảy ra trong quá trình xóa, thì sẽ tiến hành `ROLLBACK` và thông báo lỗi trong phần `BEGIN CATCH` và `END CATCH`
 - Input:
 - * `@class_id` (`BIGINT`):
 - **Description:** ID của class cần thực hiện delete
 - **Default Value:** (None)



The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the 'Tables' node under 'As2_CO2013' is expanded, showing various tables like 'dbo.address', 'dbo.administrator', etc. In the center pane, the 'update_class.sql' script is being run. The results show a single row updated in the 'dbo.class' table:

class_id	class_deposit	class_status	commission_fee	requirements	date_start	salary	addr_id	student_id	ts_id
1	10018	500000	Da giao	250	Advanced Math Required	2024-12-15 00:00:00.0000000	1000000	4	4

Hình 37: Chính sửa trạng thái của lớp học thành công

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the 'Tables' node under 'As2_CO2013' is expanded, showing various tables like 'dbo.address', 'dbo.administrator', etc. In the center pane, the 'update_class.sql' script is being run. The results show a single row updated in the 'dbo.class' table:

class_id	class_deposit	class_status	commission_fee	requirements	date_start	salary	addr_id	student_id	ts_id
1	10019	200000	Da giao	100000	Gia su phai la gia su nu	2024-12-10 00:00:00.0000000	1000000	3	3

Hình 38: Chính sửa lại yêu cầu, tiền cọc và ngày bắt đầu của lớp học



The screenshot shows the SSMS interface with the following details:

- Connections:** Shows multiple connections including 'NORA-NEKO\MMYNEWS...', 'smart_printing_system', 'online_tutor-finding', and 'As2_CO2013' (selected).
- Query Window:** Contains the following SQL code:

```
77    -- For application
78    THROW;
79    END CATCH
80
81    GO
82
83    GO
84
85    SELECT *
86    FROM dbo.class c
87    WHERE c.class_id > 10;
88
89
90
91
92    -- example to execute the stored procedure we just created
93    EXEC update_class
94    @class_id = 10018,
95    @class_status = 'Da giao'.
96    GO
97
98    SELECT *
99    FROM dbo.class
100   WHERE class_id = 10018;
101
```
- Results Grid:** Displays a table of class data with columns: class_id, class_deposit, class_status, commission_fee, requirements, date_start, salary, addr_id, student.
- Status Bar:** Shows 'Ln 85, Col 1 (58 selected)', 'Spaces: 4', 'UTF-8', 'CRLF', '3 rows', 'Choose SQL Language', '00:00:00', 'localhost: As2_CO2013', and the current date and time '12/10/2024 7:41 AM'.

Hình 39: Các lớp học sau khi đã update với những câu lệnh ở trên

• Required / Optional: Required

- Output: NONE
- Câu lệnh tạo thủ tục:

```
IF EXISTS (
SELECT *
    FROM INFORMATION_SCHEMA.ROUTINES
WHERE SPECIFIC_SCHEMA = N'dbo'
    AND SPECIFIC_NAME = N'delete_class'
    AND ROUTINE_TYPE = N'PROCEDURE'
)
DROP PROCEDURE dbo.delete_class
GO

-- Create the stored procedure in the specified schema
CREATE PROCEDURE [dbo].[delete_class]
    @class_id BIGINT
AS
BEGIN
    BEGIN TRY

        -- Start transaction to ensure atomicity
        BEGIN TRANSACTION;

        -- Check if the class exists
        IF NOT EXISTS
        (
```



```
        SELECT 1
        FROM [dbo].[class]
        WHERE [class_id] = @class_id
    )
    -- If the class does not exist, return an error
    THROW 50003, 'Class has been altered or deleted', 1;

    -- Delete the class from the table
    DELETE FROM [dbo].[class]
    WHERE [class_id] = @class_id;

    -- Commit the transaction
    COMMIT TRANSACTION;

    -- PRINT 'Class with ID ' + CAST(@class_id AS VARCHAR) +
    --       ' has been successfully deleted.';

END TRY

BEGIN CATCH

    -- Rollback the transaction if an error occurs
    IF @@TRANCOUNT > 0
        ROLLBACK TRANSACTION;

    -- For debugging as SA
    SELECT ERROR_LINE() AS [error line],
    ERROR_MESSAGE() AS [message],
    ERROR_NUMBER() AS [number],
    ERROR_PROCEDURE() AS [procedure],
    ERROR_SEVERITY() AS [severity],
    ERROR_STATE() AS [state];

    -- For application
    THROW;

END CATCH
END;
GO
```

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc tạo thủ tục
- Các câu lệnh thực thi thủ tục trường hợp báo lỗi:
 - * Trường hợp xóa một lớp không tồn tại/lớp đã xóa

```
-- example to execute the stored procedure we just created
EXECUTE dbo.delete_class
@class_id = 15
GO
```

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc kiểm tra thủ tục
- Các câu lệnh thực thi thủ tục trường hợp đúng:
 - * Trường hợp thực thi đúng 1:



The screenshot shows the SQL Server Management Studio interface. In the center pane, a script window displays the creation of a stored procedure:

```
1 USE Ass2_CO2013;
2 GO
3 -- Create a new stored procedure called 'delete_class' in schema 'dbo'
4 -- Drop the stored procedure if it already exists
5 IF EXISTS (
6     SELECT *
7     FROM INFORMATION_SCHEMA.ROUTINES
8     WHERE ROUTINE_NAME = N'dbo.delete_class'
9     AND SPECIFIC_NAME = N'delete_class'
10    AND ROUTINE_TYPE = N'PROCEDURE'
11)
12 DROP PROCEDURE dbo.delete_class
13 GO
14
15
16 -- Create the stored procedure in the specified schema
17 CREATE PROCEDURE [dbo].[delete_class]
18 @class_id BIGINT
19 AS
20 BEGIN
```

In the bottom right corner of the interface, the status bar shows "1001 AM 12/10/2024".

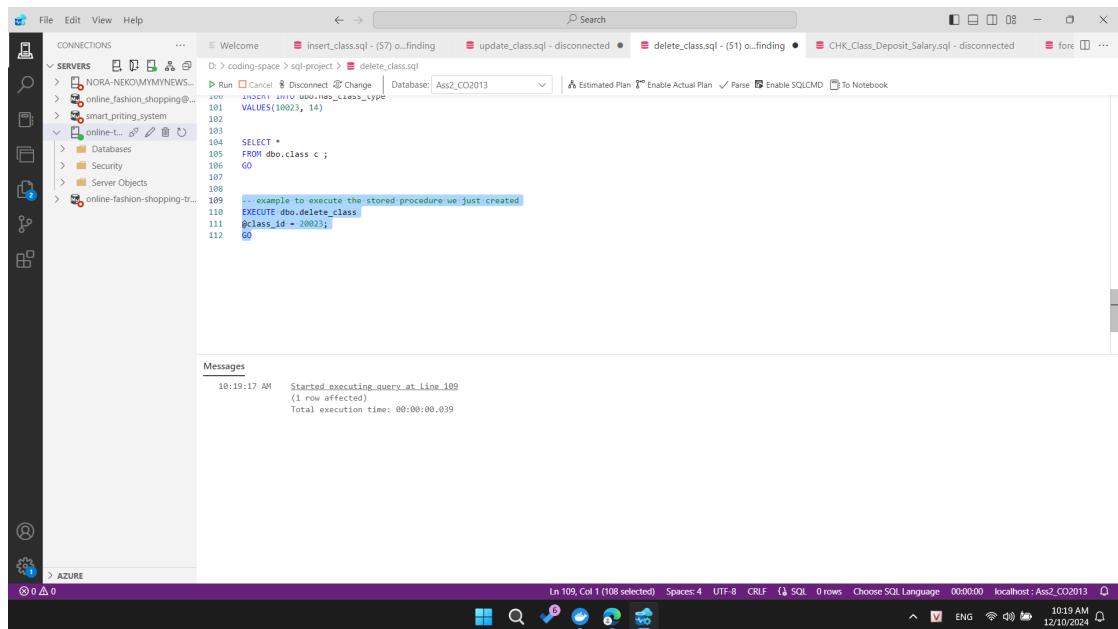
Hình 40: Tạo thủ tục để xóa một lớp thành công

The screenshot shows the SQL Server Management Studio interface. In the center pane, a script window displays the execution of the stored procedure delete_class with an invalid class_id:

```
102
103
104 SELECT *
105     FROM dbo.class c
106     WHERE c.class_id = 10023;
107
108
109 -- example to execute the stored procedure we just created
110 EXECUTE dbo.delete_class
111 @class_id = 15;
112 GO
```

In the bottom right corner of the interface, the status bar shows "1003 AM 12/10/2024".

Hình 41: Thủ tục thực thi bị lỗi do class_id không tồn tại trong bảng



Hình 42: Xóa row class với class_id = 20023 thành công

```
-- example to execute the stored procedure we just created
EXECUTE dbo.delete_class
@class_id = 20023;
GO
```

* Trường hợp thực thi đúng 2:

```
-- example to execute the stored procedure we just created
EXECUTE dbo.delete_class
@class_id = 20024;
GO
```

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc thực hiện kiểm tra thủ tục



The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the 'D:\coding-space>sql-project' folder is expanded, showing 'insert_class.sql', 'update_class.sql', 'delete_class.sql', and 'CHK_Class_Deposit_Salary.sql'. The 'delete_class.sql' file is open in the script editor, containing the following code:

```
101     VALUES(10923, 14)
102
103
104     SELECT *
105     FROM dbo.class c ;
106
107
108
109     -- example to execute the stored procedure we just created
110     EXECUTE dbo.delete_class
111     @class_id = 20024;
112
113
```

In the Messages pane, the output of the query is shown:

```
10:20:48 AM Started executing query at Line 109
(1 row affected)
Total execution time: 00:00:00.013
```

The status bar at the bottom indicates the session is connected to 'localhost: Ass2_CO2013'.

Hình 43: Xóa row class với class_id = 20024 thành công

The screenshot shows the SQL Server Management Studio interface. The 'class' table is selected in the Results pane. The results show 12 rows of data, corresponding to the 12 rows that were inserted earlier. The columns are: class_id, class_deposit, class_status, commission_fee, requirements, date_start, salary, addr_id, student_id, and ts_id.

	class_id	class_deposit	class_status	commission_fee	requirements	date_start	salary	addr_id	student_id	ts_id
3	3	2000000	Da giao	NULL	NULL	2022-05-09 00:00:00.0000000	2000000	4	3	1
4	4	NULL	Da giao	1000000	NULL	2023-05-30 00:00:00.0000000	1000000	5	4	2
5	5	NULL	Da giao	4000000	NULL	2023-06-19 00:00:00.0000000	4000000	6	5	1
6	6	NULL	Da giao	3000000	NULL	2024-07-05 00:00:00.0000000	3000000	7	6	1
7	7	3000000	Da giao	NULL	NULL	2024-06-08 00:00:00.0000000	3000000	8	7	1
8	8	NULL	Chua giao	1000000	NULL	NULL	1000000	5	4	2
9	9	2000000	Chua giao	NULL	NULL	NULL	2000000	8	7	2
10	10	4000000	Chua giao	NULL	NULL	NULL	4000000	4	3	3
11	100019	10000000	Da giao	1000000	Gia su phai la gia su nu	2024-12-10 00:00:00.0000000	1000000	3	3	1
12	100020	1000000	Chua giao	1000000	Learn at HCMUT for 2 ye...	2024-07-12 00:00:00.0000000	1000000	5	5	1

Hình 44: Bảng class sau khi đã xóa các lớp với các class_id ở trên



3.2 Trigger

3.2.1 Trigger tính toán điểm đánh giá (rate) gia sư

1. Yêu cầu: Tính toán lại điểm đánh giá gia sư khi có thêm điểm đánh giá mới, khi phiếu đánh giá cũ bị xóa hay thay đổi.
2. SQL:

```
CREATE TRIGGER Calculate_Tutor_Rate
ON [Ass2_CO2013].[dbo].[tutor_review]
AFTER INSERT, DELETE, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    -- Store tutor\_id
    DECLARE @AffectedTutors TABLE (tutor_id BIGINT);

    INSERT INTO @AffectedTutors (tutor_id)
    SELECT DISTINCT c.tutor_id
    FROM inserted i
    JOIN [Ass2_CO2013].[dbo].[class] c ON i.class_id = c.class_id;

    INSERT INTO @AffectedTutors (tutor_id)
    SELECT DISTINCT c.tutor_id
    FROM deleted d
    JOIN [Ass2_CO2013].[dbo].[class] c ON d.class_id = c.class_id;

    DELETE FROM @AffectedTutors WHERE tutor_id IS NULL;

    UPDATE t
    SET rate = (
        SELECT AVG(r.rate)
        FROM [Ass2_CO2013].[dbo].[tutor_review] r
        JOIN [Ass2_CO2013].[dbo].[class] c ON r.class_id = c.class_id
        WHERE c.tutor_id = t.tutor_id
    )
    FROM [Ass2_CO2013].[dbo].[tutor] t
    WHERE t.tutor_id IN (SELECT tutor_id FROM @AffectedTutors);
END;
```

Listing 47: Trigger trên bảng tutor_review

3. Test case: Thực hiện test với student, class, tutor có id lần lượt là 1,1,13.

	tutor_review_id	comment	rate	time_stamp	class_id	student_id
1	1	Giang hay, dễ hiểu, luôn đúng giờ.	5	2022-10-11 00:00:00.0000000	1	1
2	2	Dạy mục đích rõ ràng, chưa hiểu VDC.	4	2022-12-20 00:00:00.0000000	2	2
3	3	Hieu nhung hoi chan, giang hoi cham.	3	2023-09-30 00:00:00.0000000	4	4
4	4	Rất hợp lý, hy vọng được học giáo viên này tiếp tục.	5	2023-10-19 00:00:00.0000000	5	5
5	5	Nhiệt tình, thân thiện, chăm soan tài liệu.	5	2024-05-11 00:00:00.0000000	6	6

Hình 45: Bảng tutor_review trước khi thực hiện insert/update/delete dữ liệu



bio	date_joined	inviting_code	n_of_invitations	rate	tutor_id	invited_code
Gia su voi 5 nam kinh nghiệm, giúp học sinh phát huy...	2022-04-02 00:00:00.0000000	A3B2C1	1	5	13	NULL
Giang dạy tận tâm với tung học sinh, cam kết nang ca...	2022-08-03 00:00:00.0000000	D7E8F5	0	4	14	NULL
Gia su giup hoc vien tu tin giao tiep va cai thien diem ...	2022-10-06 00:00:00.0000000	X1Y4Z9	0	3	15	NULL
Voi niem dam me day hoc,toi cam ket mang den nhu...	2022-08-07 00:00:00.0000000	Q2W7R6	0	5	16	NULL
Voi phong cach day hoc sang tao va de hieu,toi giup ...	2023-02-20 00:00:00.0000000	M9N3P8	0	5	17	NULL
Toi la gia su luon tao moi truong hoc tap thoai mai, gi...	2023-03-17 00:00:00.0000000	V5B1K4	0	4	18	13
Gia su giao kinh nghiệm, san sang dong hanh cung ...	2023-04-05 00:00:00.0000000	T2A9L3	0	5	19	NULL
Voi phuong pháp day hoc linh hoạt,toi giup hoc sinh ...	2024-09-20 00:00:00.0000000	J8H7W2	0	5	20	NULL

Hình 46: Bảng tutor trước khi thực hiện insert/update/delete dữ liệu

- Trường hợp 1: Insert dữ liệu.
 - Rate trước đó của gia sư: 5.
 - Thực hiện insert thêm giá trị rate với rate = 3.
 - Dự kiến điểm rate của gia sư: $(5 + 3)/2 = 4$.

```
INSERT INTO [Ass2_C02013].[dbo].[tutor_review]
(comment, rate, time_stamp, class_id, student_id)
VALUES ('Day de hieu', 3, '2024-12-11', 1, 1);
```

Listing 48: Câu lệnh Insert dữ liệu vào bảng tutor_review

tutor_review_id	comment	rate	time_stamp	class_id	student_id
1	Giang hay, de hieu, luon dung gio.	5	2022-10-11 00:00:00.0000000	1	1
2	Dat muc van dung, chua hieu VDC.	4	2022-12-20 00:00:00.0000000	2	2
3	Hieu nhung hoi chan, giang hoi cham.	3	2023-09-30 00:00:00.0000000	4	4
4	Rat hop, hy vong duoc hoc giao vien nay tiep tục.	5	2023-10-19 00:00:00.0000000	5	5
5	Nhiet tinh, than thien, cham soan tai lieu.	5	2024-05-11 00:00:00.0000000	6	6
8	Day de hieu	3	2024-12-11 00:00:00.0000000	1	1

Hình 47: Bảng tutor_review sau khi thực hiện insert dữ liệu

bio	date_joined	inviting_code	n_of_invitations	rate	tutor_id	invited_code
Gia su voi 5 nam kinh nghiệm, giúp học sinh phát huy...	2022-04-02 00:00:00.0000000	A3B2C1	1	4	13	NULL
Giang dạy tận tâm với tung học sinh, cam kết nang ca...	2022-08-03 00:00:00.0000000	D7E8F5	0	4	14	NULL
Gia su giup hoc vien tu tin giao tiep va cai thien diem ...	2022-10-06 00:00:00.0000000	X1Y4Z9	0	3	15	NULL
Voi niem dam me day hoc,toi cam ket mang den nhu...	2022-08-07 00:00:00.0000000	Q2W7R6	0	5	16	NULL
Voi phong cach day hoc sang tao va de hieu,toi giup ...	2023-02-20 00:00:00.0000000	M9N3P8	0	5	17	NULL
Toi la gia su luon tao moi truong hoc tap thoai mai, gi...	2023-03-17 00:00:00.0000000	V5B1K4	0	4	18	13
Gia su giao kinh nghiệm, san sang dong hanh cung ...	2023-04-05 00:00:00.0000000	T2A9L3	0	5	19	NULL
Voi phuong pháp day hoc linh hoạt,toi giup hoc sinh ...	2024-09-20 00:00:00.0000000	J8H7W2	0	5	20	NULL

Hình 48: Bảng tutor sau khi thực hiện insert dữ liệu

- Trường hợp 2: Update dữ liệu.
 - Rate trước đó của gia sư: 4.
 - Thực hiện Update thêm giá trị rate với rate = 1.
 - Dự kiến điểm rate của gia sư: $(5 + 1)/2 = 3$.

```
UPDATE [Ass2_C02013].[dbo].[tutor_review]
SET rate = 1 WHERE tutor_review_id=8;
```



Listing 49: Câu lệnh Update dữ liệu vào bảng tutor_review

	tutor_review_id	comment	rate	time_stamp	class_id	student_id
1	1	Giang hay, de hieu, luon dung gio.	5	2022-10-11 00:00:00.0000000	1	1
2	2	Dat muc van dung, chua hieu VDC.	4	2022-12-20 00:00:00.0000000	2	2
3	3	Hieu nhung hoi chan, giang hoi cham.	3	2023-09-30 00:00:00.0000000	4	4
4	4	Rat hop, hy vong duoc hoc giao vien nay tiep tuc.	5	2023-10-19 00:00:00.0000000	5	5
5	5	Nhiet tinh, than thien, cham soan tai lieu.	5	2024-05-11 00:00:00.0000000	6	6
6	8	Day de hieu	1	2024-12-11 00:00:00.0000000	1	1

Hình 49: Bảng tutor_review sau khi thực hiện Update dữ liệu

	bio	date_joined	inviting_code	n_of_invitations	rate	tutor_id	invited_code
1	Gia su voi 5 nam kinh nghiem, giup hoc sinh phat huy...	2022-04-02 00:00:00.0000000	A3B2C1	1	3	13	NULL
2	Giang day tan tam voi tung hoc sinh, cam ket nang ca...	2022-08-03 00:00:00.0000000	D7E8F5	0	4	14	NULL
3	Gia su giup hoc vien tu tin giao tiep va cai thien diem ...	2022-10-06 00:00:00.0000000	X1Y4Z9	0	3	15	NULL
4	Voi niem dam me day hoc,toi cam ket mang den nhu...	2022-08-07 00:00:00.0000000	Q2W7R6	0	5	16	NULL
5	Voi phong cach day hoc sang tao va de hieu,toi giup ...	2023-02-20 00:00:00.0000000	M9N3P8	0	5	17	NULL
6	Toi la gia su luon tao moi truong hoc tap thoai mai,gi...	2023-03-17 00:00:00.0000000	V5B1K4	0	4	18	13
7	Gia su giau kinh nghiem, san sang dong hanh cung ...	2023-04-05 00:00:00.0000000	T2A9L3	0	5	19	NULL
8	Voi phuong phap day hoc linh hoạt,toi giup hoc sinh ...	2024-09-20 00:00:00.0000000	J8H7W2	0	5	20	NULL

Hình 50: Bảng tutor sau khi thực hiện Update dữ liệu

- Trường hợp 3: Delete dữ liệu.
 - Rate trước đó của gia sư: 3.
 - Thực hiện Delete tutor_review có id =8.
 - Dự kiến điểm rate của gia sư: $(5)/1 = 5$.

```
DELETE FROM [Ass2_C02013].[dbo].[tutor_review] WHERE
tutor_review_id=8;
```

Listing 50: Câu lệnh Delete dữ liệu vào bảng tutor_review

	tutor_review_id	comment	rate	time_stamp	class_id	student_id
1	1	Giang hay, de hieu, luon dung gio.	5	2022-10-11 00:00:00.0000000	1	1
2	2	Dat muc van dung, chua hieu VDC.	4	2022-12-20 00:00:00.0000000	2	2
3	3	Hieu nhung hoi chan, giang hoi cham.	3	2023-09-30 00:00:00.0000000	4	4
4	4	Rat hop, hy vong duoc hoc giao vien nay tiep tuc.	5	2023-10-19 00:00:00.0000000	5	5
5	5	Nhiet tinh, than thien, cham soan tai lieu.	5	2024-05-11 00:00:00.0000000	6	6

Hình 51: Bảng tutor_review sau khi thực hiện Delete dữ liệu



	bio	date_joined	inviting_code	n_of_invitations	rate	tutor_id	invited_code
1	Gia su voi 5 nam kinh nghiệm, giúp học sinh phát huy...	2022-04-02 00:00:00.0000000	A3B2C1	1	5	13	NULL
2	Giang dạy tận tâm với tung học sinh, cam kết nâng cao...	2022-08-03 00:00:00.0000000	D7E8F5	0	4	14	NULL
3	Gia sư giúp học viên tự tin giao tiếp và cải thiện điểm...	2022-10-06 00:00:00.0000000	X1Y4Z9	0	3	15	NULL
4	Với niềm đam mê dạy học, tôi cam kết mang đến những...	2022-08-07 00:00:00.0000000	Q2W7R6	0	5	16	NULL
5	Với phong cách dạy học sang tạo và dễ hiểu, tôi giúp...	2023-02-20 00:00:00.0000000	M9N3P8	0	5	17	NULL
6	Toi là gia sư luôn tạo môi trường học tập thoải mái, gi...	2023-03-17 00:00:00.0000000	V5B1K4	0	4	18	13
7	Gia sư giàu kinh nghiệm, sẵn sàng đồng hành cung...	2023-04-05 00:00:00.0000000	T2A9L3	0	5	19	NULL
8	Với phương pháp dạy học linh hoạt, tôi giúp học sinh ...	2024-09-20 00:00:00.0000000	J8H7W2	0	5	20	NULL

Hình 52: Bảng tutor sau khi thực hiện Delete dữ liệu

3.2.2 Trigger tính toán điểm đóng góp của gia sư

- Yêu cầu: Tính toán lại điểm đóng góp của gia sư khi có gia sư khác được mời vào thành công.
- SQL:

```
CREATE TRIGGER trg_ManageInvitations
ON [dbo].[tutor]
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM [dbo].[tutor] AS t
        WHERE t.inviting_code = t.invited_code
        AND t.inviting_code IS NOT NULL
    )
    BEGIN
        THROW 50041, 'A tutor cannot invite themselves.', 1;
    END;

    IF EXISTS (
        SELECT 1
        FROM inserted AS i
        WHERE i.invited_code IS NOT NULL
        AND NOT EXISTS (
            SELECT 1
            FROM [dbo].[tutor] AS t
            WHERE t.inviting_code = i.invited_code
        )
    )
    BEGIN
        THROW 50042, 'The invited_code must exist in the system.', 1;
    END;

    UPDATE t
    SET t.n_of_invitations = ISNULL(
        SELECT COUNT(*)
        FROM [dbo].[tutor] AS t2
        WHERE t2.invited_code = t.inviting_code
    )
END;
```



```
        AND t2.inviting_code != t2.invited_code;
), 0)
FROM [dbo].[tutor] AS t;

END;
```

Listing 51: Trigger trên bảng tutor_review

3. Test case: Thực hiện test với tutor có id là 14.

	bio	date_joined	inviting_code	n_of_invitations	rate	tutor_id	invited_code
1	Gia su voi 5 nam kinh nghiem, giup hoc sinh phat huy...	2022-02-04 00:00:00.0000000	A3B2C1	1	5	13	NULL
2	Giang day tan tam voi tung hoc sinh, cam ket nang ca...	2022-03-08 00:00:00.0000000	D7E8F5	0	4	14	NULL
3	Gia su giup hoc vien tu tin giao tiep va cai thien diem ...	2022-06-10 00:00:00.0000000	X1Y4Z9	0	NULL	15	NULL
4	Voi niem dam me day hoc,toi cam ket mang den nhu...	2022-07-08 00:00:00.0000000	Q2W7R6	0	3	16	NULL
5	Voi phong cach day hoc sang tao va de hieu,toi giup ...	2023-02-20 00:00:00.0000000	M9N3P8	0	5	17	NULL
6	Toi la gia su luon tao moi truong hoc tap thoai mai, gi...	2023-03-17 00:00:00.0000000	V5B1K4	0	5	18	A3B2C1
7	Gia su giao kinh nghiem, san sang dong hanh cung ...	2023-05-04 00:00:00.0000000	T2A9L3	0	NULL	19	NULL
8	Voi phuong phap day hoc linh hoat,toi giup hoc sinh ...	2024-09-20 00:00:00.0000000	J8H7W2	0	NULL	20	NULL

Hình 53: Bảng tutor trước khi thực hiện thay đổi dữ liệu

- Trường hợp 1: Update mã mời của gia sư.
 - n_of_invitations trước đó của gia sư 13: 1.
 - Thực hiện update mã mời của gia sư 14 là mã của gia sư có id 13.
 - Dự kiến điểm n_of_invitations của gia sư 13: 2.

```
UPDATE [Ass20].[dbo].[tutor]
SET invited_code = 'A3B2C1'
WHERE tutor_id = 14;
```

Listing 52: Câu lệnh Update mã mời gia sư id 14

	bio	date_joined	inviting_code	n_of_invitations	rate	tutor_id	invited_code
1	Gia su voi 5 nam kinh nghiem, giup hoc sinh phat huy...	2022-02-04 00:00:00.0000000	A3B2C1	2	5	13	NULL
2	Giang day tan tam voi tung hoc sinh, cam ket nang ca...	2022-03-08 00:00:00.0000000	D7E8F5	0	4	14	A3B2C1
3	Gia su giup hoc vien tu tin giao tiep va cai thien diem ...	2022-06-10 00:00:00.0000000	X1Y4Z9	0	NULL	15	NULL
4	Voi niem dam me day hoc,toi cam ket mang den nhu...	2022-07-08 00:00:00.0000000	Q2W7R6	0	3	16	NULL
5	Voi phong cach day hoc sang tao va de hieu,toi giup ...	2023-02-20 00:00:00.0000000	M9N3P8	0	5	17	NULL
6	Toi la gia su luon tao moi truong hoc tap thoai mai, gi...	2023-03-17 00:00:00.0000000	V5B1K4	0	5	18	A3B2C1
7	Gia su giao kinh nghiem, san sang dong hanh cung ...	2023-05-04 00:00:00.0000000	T2A9L3	0	NULL	19	NULL
8	Voi phuong phap day hoc linh hoat,toi giup hoc sinh ...	2024-09-20 00:00:00.0000000	J8H7W2	0	NULL	20	NULL

Hình 54: Bảng tutor sau khi thực hiện update dữ liệu

- Trường hợp 2: Update mã mời về null.
 - n_of_invitations trước đó của gia sư 13: 2.
 - Thực hiện Update thêm giá trị mã mời gia sư 14 = NULL.
 - Dự kiến điểm n_of_invitations của gia sư 13: 1.

```
ALTER TABLE [Ass20].[dbo].[tutor]
ALTER COLUMN invited_code VARCHAR(255) NULL;

UPDATE [Ass20].[dbo].[tutor]
```



```
SET invited_code = NULL
WHERE tutor_id = 14;
```

Listing 53: Câu lệnh Update dữ liệu vào bảng tutor

	bio	date_joined	inviting_code	n_of_invitations	rate	tutor_id	invited_code
1	Gia su voi 5 nam kinh nghiem, giup hoc sinh phat huy...	2022-02-04 00:00:00.0000000	A3B2C1	1	5	13	NULL
2	Giang day tan tam voi tung hoc sinh, cam ket nang ca...	2022-03-08 00:00:00.0000000	D7E8F5	0	4	14	NULL
3	Gia su giup hoc vien tu tin giao tiep va cai thien diem ...	2022-06-10 00:00:00.0000000	X1Y4Z9	0	NULL	15	NULL
4	Voi niem dam me day hoc,toi cam ket mang den nhu...	2022-07-08 00:00:00.0000000	Q2W7R6	0	3	16	NULL
5	Voi phong cach day hoc sang tao va de hieu,toi giup ...	2023-02-20 00:00:00.0000000	M9N3P8	0	5	17	NULL
6	Toi la gia su luon tao moi truong hoc tap thoai mai, gi...	2023-03-17 00:00:00.0000000	V5B1K4	0	5	18	A3B2C1
7	Gia su giao kinh nghiem, san sang dong hanh cung ...	2023-05-04 00:00:00.0000000	T2A9L3	0	NULL	19	NULL
8	Voi phuong phap day hoc linh hoạt,toi giup hoc sinh ...	2024-09-20 00:00:00.0000000	J8H7W2	0	NULL	20	NULL

Hình 55: Bảng tutor sau khi thực hiện Update dữ liệu

- Trường hợp 3: Trường hợp lỗi gia sư tự mời mình hoặc mã mời không tồn tại.
 - Thực hiện Update mã mời bằng "ABC".
 - Dự kiến : Báo lỗi không tồn tại mã mời hoặc vi phạm khóa ngoại.

```
UPDATE [Ass20].[dbo].[tutor]
SET invited_code = 'ABC'
WHERE tutor_id = 14;
UPDATE [Ass20].[dbo].[tutor]
SET invited_code = 'D7E8F5'
WHERE tutor_id = 14;
```

Listing 54: Trường hợp câu lệnh gây lỗi

```
! Messages
Msg 547, Level 16, State 0, Line 14
The UPDATE statement conflicted with the FOREIGN KEY SAME TABLE constraint "FKKn6esf2ou7po2yjmvkfvwk9swp". The conflict occurred in database "Ass20", table "dbo.tutor", column 'inviting_code'.
The statement has been terminated.

Completion time: 2024-12-13T20:30:44.5654862+07:00
```

Hình 56: Lỗi khi insert giá trị mã mời không tồn tại (conflicted with the Foreign key)

```
Msg 50041, Level 16, State 1, Procedure trg_ManageInvitations, Line 16 [Batch Start Line 9]
A tutor cannot invite themselves.

Completion time: 2024-12-13T20:31:23.3317667+07:00
```

Hình 57: Lỗi khi insert giá trị mã mời chính gia sư đó (tutor can not invite themselves)



3.3 Thủ tục

3.3.1 Thủ tục 1: Viết 1 câu truy vấn có mệnh đề WHERE và ORDER BY

1. Mô tả thủ tục:

- **Mục đích thủ tục:** Dùng để hiển thị thông tin cơ bản của các lớp học trong hệ thống. Các thông tin được truy xuất (phần Output).
- **Sử dụng trong trường hợp - usecase:** hệ thống quản lý lớp học của administrator cần tra cứu thông tin lớp học theo các tiêu chí lọc nhất định như: Khối lớp, Kiểu dạy, Khu vực, Trạng thái, Môn học, Mã lớp học, SDT học viên, Ngày bắt đầu.
- **Thao tác chính:**
 - + *Bước 1:* Truy vấn, kết hợp các bảng dữ liệu để lấy thông tin đầy đủ về lớp học: `class`, `address`, `district_city`, `teaching_style`, `has_subject`, `subject`, `user`, `class_type`, `has_class_type`, `student`.
 - + *Bước 2:* Lọc các dữ liệu theo điều kiện dựa trên các sự lựa chọn.
 - + *Bước 3:* Danh sách lớp học được sắp xếp theo mã lớp học (`class_id`).
 - + *Bước 4:* Trả về danh sách các lớp học với các thông tin đã liệt kê trên.

2. Input:

- `class_type_name VARCHAR(255)`: Tên khối lớp.
- `ts_name VARCHAR(255)`: Tên kiểu dạy.
- `name VARCHAR(255)`: Tên khu vực (quận/ thành phố).
- `class_status VARCHAR(255)`: Trạng thái.
- `subject_name VARCHAR(255)`: Tên môn học.
- `phone_number VARCHAR(255)`: SDT học viên.
- `date_start_from DATETIME2(7)`: Ngày bắt đầu tìm kiếm.
- `date_start_to DATETIME2(7)`: Ngày kết thúc tìm kiếm.
- `sort_order NVARCHAR(4)`: Thứ tự mã lớp học tăng/ giảm dần.

3. Output:

- `class_id`: Mã lớp học.
- `date_start`: Ngày bắt đầu.
- `class_status`: Trạng thái (Chưa giao, Đã giao).
- `name`: Khu vực (quận/thành phố).
- `ts_name`: Kiểu dạy (Trực tiếp, Trực tuyến).
- `subject_name`: Môn học (Ngữ Văn, Tiếng Anh, Toán, KHXH, KHTN, Tin học).
- `class_type_name`: Khối lớp (Lớp 1, Lớp 2,..., Luyện thi Đại học, Ôn luyện Đại học).
- `phone_number`: SDT (học viên).

4. Câu lệnh tạo thủ tục:



```
CREATE OR ALTER PROCEDURE FindClassByFilter
    @class_type_name VARCHAR(255) = NULL,
    @ts_name VARCHAR(255) = NULL,
    @name VARCHAR(255) = NULL,
    @class_status VARCHAR(255) = NULL,
    @subject_name VARCHAR(255) = NULL,
    @phone_number VARCHAR(255) = NULL,
    @date_start_from DATETIME2(7) = NULL, -- Ngay bat dau tim kiem.
    @date_start_to DATETIME2(7) = NULL, -- Ngay ket thuc tim kiem.
    -- Sap xep tang/ giam dan ('ASC' hoac 'DESC') theo ma lop hoc.
    @sort_order NVARCHAR(4) = 'ASC'

AS
BEGIN
    -- 1. Truy van kiem tra su ton tai cua du lieu.
    IF NOT EXISTS (
        SELECT 1
        FROM
            class c -- Bang @class.

            -- "Khu vuc": Bang @address va Bang @district_city.
            INNER JOIN
                address a ON c.addr_id = a.addr_id
            INNER JOIN
                district_city d ON a.dist_city_id = d.dist_city_id

            -- "Kieu day": Bang @teaching_style.
            INNER JOIN
                teaching_style ts ON c.ts_id = ts.ts_id

            -- "Mon hoc": Bang @has_subject va Bang @subject.
            INNER JOIN
                has_subject hs ON c.class_id = hs.class_id
            INNER JOIN
                subject s ON hs.subject_id = s.subject_id

            -- "Khoi lop": Bang @has_class_type va Bang @class_type.
            INNER JOIN
                has_class_type hct ON c.class_id= hct.class_id
            INNER JOIN
                class_type ct ON hct.class_type_id = ct.class_type_id

            -- "SDT (Hoc vien)": Bang @student va Bang @user.
            INNER JOIN
                student st ON c.student_id = st.student_id
            INNER JOIN
                [user] u ON st.student_id = u.user_id

        WHERE
            -- Loc "Khoi lop" theo class_type_name tu Bang
            -- @class_type.
            (@class_type_name IS NULL OR ct.class_type_name =
            @class_type_name) AND
            -- Loc "Kieu day" theo ts_name tu Bang @teaching_style.
```



```
(@ts_name IS NULL OR ts.ts_name = @ts_name) AND
-- Loc "Khu vuc" theo name tu Bang @district_city.
(@name IS NULL OR d.name = @name) AND
-- Loc "Trang thai" theo class_status tu Bang @class.
(@class_status IS NULL OR c.class_status = @class_status)
    AND
-- Loc "Mon hoc" theo subject_name tu Bang @subject.
(@subject_name IS NULL OR s.subject_name = @subject_name)
    AND
-- Tim "SDT (Hoc vien)" theo phone_number tu Bang @user.
(@phone_number IS NULL OR u.phone_number = @phone_number)
    AND
-- Loc "Ngay bat dau" trong mot khoang thoi gian theo
    date_start tu Bang @class.
(@date_start_from IS NULL OR c.date_start >=
    @date_start_from) AND
(@date_start_to IS NULL OR c.date_start <=
    @date_start_to)
)

BEGIN
    -- Thong bao loi neu khong tim thay du lieu phu hop.
    RAISERROR('Khong tim thay du lieu phu hop yeu cau.', 16, 1);
    RETURN;
END;
-- 2. Truy van thong tin khi du lieu ton tai.
DECLARE @sql_query NVARCHAR(MAX);

SET @sql_query = N'
SELECT
    -- Tu Bang @class: ma lop hoc, ngay bat dau, trang thai.
    c.class_id,
    c.date_start,
    c.class_status,
    d.name, -- Ten QUAN/ THANH PHO tu Bang @district_city.
    ts.ts_name, -- Ten KIEU DAY tu Bang @teaching_style.
    s.subject_name, -- Ten MON HOC tu Bang @subject.
    ct.class_type_name, -- Ten KHOI LOP tu Bang @class_type.
    u.phone_number -- SDT (Hoc vien) tu Bang @user.
FROM
    class c -- Bang @class.

    -- "Khu vuc": Bang @address va Bang @district_city.
    INNER JOIN
        address a ON c.addr_id = a.addr_id
    INNER JOIN
        district_city d ON a.dist_city_id = d.dist_city_id

    -- "Kieu day": Bang @teaching_style.
    INNER JOIN
        teaching_style ts ON c.ts_id = ts.ts_id

    -- "Mon hoc": Bang @has_subject va Bang @subject.
'
```



```
    INNER JOIN
        has_subject hs ON c.class_id = hs.class_id
    INNER JOIN
        subject s ON hs.subject_id = s.subject_id

    -- "Khoi lop": Bang @has_class_type va Bang @class_type.
    INNER JOIN
        has_class_type hct ON c.class_id= hct.class_id
    INNER JOIN
        class_type ct ON hct.class_type_id = ct.class_type_id

    -- "SDT (Hoc vien)": Bang @student va Bang @user.
    INNER JOIN
        student st ON c.student_id = st.student_id
    INNER JOIN
        [user] u ON st.student_id = u.user_id

WHERE
    -- Loc "Khoi lop" theo class_type_name tu Bang @class_type.
    (@class_type_name IS NULL OR ct.class_type_name =
     @class_type_name) AND
    -- Loc "Kieu day" theo ts_name tu Bang @teaching_style.
    (@ts_name IS NULL OR ts.ts_name = @ts_name) AND
    -- Loc "Khu vuc" theo name tu Bang @distric_city.
    (@name IS NULL OR d.name = @name) AND
    -- Loc "Trang thai" theo class_status tu Bang @class.
    (@class_status IS NULL OR c.class_status = @class_status) AND
    -- Loc "Mon hoc" theo subject_name tu Bang @subject.
    (@subject_name IS NULL OR s.subject_name = @subject_name) AND
    -- Tim "SDT (Hoc vien)" theo phone_number tu Bang @user.
    (@phone_number IS NULL OR u.phone_number = @phone_number) AND
    -- Loc "Ngay bat dau" trong mot khoang thoi gian theo
    date_start tu Bang @class.
    (@date_start_from IS NULL OR c.date_start >= @date_start_from
     ) AND
    (@date_start_to IS NULL OR c.date_start <= @date_start_to)

ORDER BY
    -- Sap xep thu tu tang/ giam dan (ASC/ DESC) theo ma lop hoc.
    c.class_id ' + @sort_order + ';'

-- Thuc thi truy van
EXEC sp_executesql @sql_query,
N'@class_type_name VARCHAR(255), @ts_name VARCHAR(255), @name
VARCHAR(255), @class_status VARCHAR(255), @subject_name
VARCHAR(255), @phone_number VARCHAR(255),
@date_start_from DATETIME2(7), @date_start_to DATETIME2
(7),
@class_type_name, @ts_name, @name, @class_status,
@subject_name, @phone_number, @date_start_from,
@date_start_to;
END;
```

Listing 55: Câu lệnh tạo thủ tục 1.



5. Kết quả các màn hình hiển thị từ DBMS minh họa cho việc tạo thủ tục:

The screenshot shows the SQL Server Management Studio interface. A new stored procedure is being created with the following code:

```
CREATE OR ALTER PROCEDURE FindClassByFilter
    @class_type_name VARCHAR(255) = NULL,
    @ts_name VARCHAR(255) = NULL,
    @name VARCHAR(255) = NULL,
    @class_status VARCHAR(255) = NULL,
    @subject_name VARCHAR(255) = NULL,
    @phone_number VARCHAR(255) = NULL,
    @date_start_from DATETIME2(7) = NULL, -- Ngay bat dau tim kiem.
    @date_start_to DATETIME2(7) = NULL, -- Ngay ket thuc tim kiem.
    -- Sap xep tang/ giam dan ('ASC' hoac 'DESC') theo ma lop hoc.
    @sort_order NVARCHAR(4) = 'ASC'
AS
BEGIN
    -- Truy van kiem tra su ton tai cua du lieu.
    IF NOT EXISTS (
        SELECT 1
        FROM
            class c -- Bang @class.
    )
    BEGIN
        -- Xuat loi trinh loi.
        RAISERROR('Khong co du lieu', 16, 1);
        RETURN;
    END
    ELSE
        -- Thuc hien truy van.
        SELECT *
        FROM
            class c -- Bang @class.
        WHERE
            (@class_type_name IS NULL OR class_type_name = @class_type_name)
            AND (@ts_name IS NULL OR ts_name = @ts_name)
            AND (@name IS NULL OR name = @name)
            AND (@class_status IS NULL OR class_status = @class_status)
            AND (@subject_name IS NULL OR subject_name = @subject_name)
            AND (@phone_number IS NULL OR phone_number = @phone_number)
            AND (@date_start_from IS NULL OR date_start_from >= @date_start_from)
            AND (@date_start_to IS NULL OR date_start_to <= @date_start_to)
            AND (@sort_order = 'ASC' ORDER BY ma_lop_hoc ASC)
            AND (@sort_order = 'DESC' ORDER BY ma_lop_hoc DESC);
END
```

The execution message at the bottom of the screen says "Query executed successfully."

Hình 58: Ảnh minh họa kết quả hiển thị từ DBMS việc tạo thủ tục.

6. Nhóm câu lệnh thực thi thủ tục mẫu:

```
-- exec1. Loc nhung lop hoc theo "Khoi lop: Lop 4".
EXEC FindClassByFilter
    @class_type_name = 'Lop 4';

-- exec2. Loc nhung lop hoc theo "Khu vuc: Quan Phu Nhuan".
EXEC FindClassByFilter
    @name = 'Tp. Thu Duc';

-- exec3. Loc nhung lop hoc theo "Kieu day: Truc tiep".
EXEC FindClassByFilter
    @ts_name = 'Truc tuyen';

-- exec4. Loc nhung lop hoc theo "Trang thai: Da giao".
EXEC FindClassByFilter
    @class_status = 'Chua giao';

-- exec5. Loc nhung lop hoc theo "Mon hoc: Tin hoc".
EXEC FindClassByFilter
    @subject_name = 'Tin hoc';

-- exec6. Sap xep lop hoc theo thu tu giam dan.
EXEC FindClassByFilter
    @sort_order = 'DESC';

-- exec7. Loc nhung lop hoc theo "SDT: 0905123457".
EXEC FindClassByFilter
    @phone_number = '0905123457';

-- exec8. Loc nhung lop hoc trong khoang thoi gian tu ngay
--       "07-10-2022" den "07-10-2023".
EXEC FindClassByFilter
```



```
@date_start_from = '2022-10-07',
@date_start_to = '2023-10-07';

-- exec9. Lọc những lớp học có "Kiểu dạy: Trực tiếp" với "Trạng thái:
-- Da giao" và có "Môn học: KHTN".
EXEC FindClassByFilter
    @ts_name = 'Trực tiếp',
    @class_status = 'Da giao',
    @subject_name = 'KHTN';

-- exec10. Lọc những lớp học theo "Khối lớp: Lớp 1", "Kiểu dạy: Trực
-- tuyen", "Trạng thái: Chưa giao" và "Môn học: Tiếng Anh".
EXEC FindClassByFilter
    @class_type_name = 'Lớp 1',
    @ts_name = 'Trực tuyen',
    @class_status = 'Chưa giao',
    @subject_name = 'Tiếng Anh';

-- exec11. Lọc những lớp học có các giá trị sau: "Khối lớp: Lớp 4", "
-- Kiểu dạy: Trực tiếp", "Khu vực: Quan Phu Nhuan", "Trạng thái: Da
-- giao", "Môn học: Toán", "SDT: 0905 123 456", trong khoảng thời
-- gian từ ngày "2022-01-01" đến "2024-01-01" và sắp xếp theo mã lớp
-- giảm dần.
EXEC FindClassByFilter
    @class_type_name = 'Lớp 4',
    @ts_name = 'Trực tiếp',
    @name = 'Quan Phu Nhuan',
    @class_status = 'Da giao',
    @subject_name = 'Toán',
    @phone_number = '0905123457',
    @date_start_from = '2022-01-01',
    @date_start_to = '2024-01-01',
    @sort_order = 'DESC';
```

Listing 56: Nhóm câu lệnh thực thi thủ tục mẫu.

7. Kết quả các màn hình hiển thị từ DBMS minh họa cho việc kiểm tra thủ tục:

	class_id	date_start	class_status	name	ts_name	subject_name	class_type_name	phone_number
1	1	2022-10-07 ...	Da giao	quan Phu Nhuan	Trực tiếp	Ngu Van	Lớp 4	0905123457
2	1	2022-10-07 ...	Da giao	quan Phu Nhuan	Trực tiếp	KHXH	Lớp 4	0905123457
3	1	2022-10-07 ...	Da giao	quan Phu Nhuan	Trực tiếp	Toán	Lớp 4	0905123457

Hình 59: Kết quả lọc những lớp có "Khối lớp: Lớp 4".

	class_id	date_start	class_status	name	ts_name	subject_name	class_type_name	phone_number
1	2	2022-08-20 ...	Da giao	TP. Thủ Đức	Trực tiếp	Toán	Lớp 8	0937456789
2	2	2022-08-20 ...	Da giao	TP. Thủ Đức	Trực tiếp	KHTN	Lớp 8	0937456789

Hình 60: Kết quả lọc những lớp theo "Khu vực: Tp. Thủ Đức".



	class_id	date_start	class_status	name	ts_name	subject_name	class_type_name	phone_number
1	4	2023-05-30...	Da giao	quan 1	Truc tuyen	Tin Hoc	On luyen Tin hoc	0912987654
2	8	NULL	Chua giao	quan 1	Truc tuyen	KHXH	Luyen thi Dai hoc	0912987654
3	8	NULL	Chua giao	quan 1	Truc tuyen	Tieng Anh	Luyen thi Dai hoc	0912987654
4	9	NULL	Chua giao	quan Binh Tan	Truc tuyen	Tieng Anh	Lop 8	0945678912

Hình 61: Kết quả lọc những lớp theo "Kiểu dạy: Trực tuyến".

	class_id	date_start	class_status	name	ts_name	subject_name	class_type_name	phone_number
1	8	NULL	Chua giao	quan 1	Truc tuyen	KHXH	Luyen thi Dai hoc	0912987654
2	8	NULL	Chua giao	quan 1	Truc tuyen	Tieng Anh	Luyen thi Dai hoc	0912987654
3	9	NULL	Chua giao	quan Binh Tan	Truc tuyen	Tieng Anh	Lop 8	0945678912
4	10	NULL	Chua giao	quan Tan Phu	Truc tiep	Toan	Lop 12	0968321654
5	10	NULL	Chua giao	quan Tan Phu	Truc tiep	KHXH	Lop 12	0968321654

Hình 62: Kết quả lọc những lớp theo "Trạng thái: Chưa giao".

	class_id	date_start	class_status	name	ts_name	subject_name	class_type_name	phone_number
1	4	2023-05-30...	Da giao	quan 1	Truc tuyen	Tin Hoc	On luyen Tin hoc	0912987654

Hình 63: Kết quả lọc những lớp theo "Môn học: Tin học".

	class_id	date_start	class_status	name	ts_name	subject_name	class_type_name	phone_number
1	10	NULL	Chua giao	quan Tan Phu	Truc tiep	Toan	Lop 12	0968321654
2	10	NULL	Chua giao	quan Tan Phu	Truc tiep	KHXH	Lop 12	0968321654
3	9	NULL	Chua giao	quan Binh Tan	Truc tuyen	Tieng Anh	Lop 8	0945678912
4	8	NULL	Chua giao	quan 1	Truc tuyen	KHXH	Luyen thi Dai hoc	0912987654
5	8	NULL	Chua giao	quan 1	Truc tuyen	Tieng Anh	Luyen thi Dai hoc	0912987654
6	7	2024-06-08...	Da giao	quan Binh Tan	Truc tiep	KHTN	Lop 9	0945678912
7	7	2024-06-08...	Da giao	quan Binh Tan	Truc tiep	Tieng Anh	Lop 9	0945678912
8	6	2024-07-05...	Da giao	quan 7	Truc tiep	KHXH	Lop 2	0923123789
9	6	2024-07-05...	Da giao	quan 7	Truc tiep	Toan	Lop 2	0923123789
10	6	2024-07-05...	Da giao	quan 7	Truc tiep	Ngu Van	Lop 2	0923123789
11	5	2023-06-19...	Da giao	quan Tan Phu	Truc tiep	Tieng Anh	Lop 11	0987654321
12	5	2023-06-19...	Da giao	quan Tan Phu	Truc tiep	Ngu Van	Lop 11	0987654321
13	5	2023-06-19...	Da giao	quan Tan Phu	Truc tiep	KHTN	Lop 11	0987654321
14	4	2023-05-30...	Da giao	quan 1	Truc tuyen	Tin Hoc	On luyen Tin hoc	0912987654
15	3	2022-09-05...	Da giao	quan Tan Phu	Truc tiep	KHXH	Lop 10	0968321654
16	3	2022-09-05...	Da giao	quan Tan Phu	Truc tiep	Toan	Lop 10	0968321654
17	2	2022-08-20...	Da giao	TP. Thu Duc	Truc tiep	Toan	Lop 8	0937456789
18	2	2022-08-20...	Da giao	TP. Thu Duc	Truc tiep	KHTN	Lop 8	0937456789
19	1	2022-10-07...	Da giao	quan Phu Nh...	Truc tiep	Ngu Van	Lop 4	0905123457
20	1	2022-10-07...	Da giao	quan Phu Nh...	Truc tiep	KHXH	Lop 4	0905123457
21	1	2022-10-07...	Da giao	quan Phu Nh...	Truc tiep	Toan	Lop 4	0905123457

Hình 64: Kết quả lọc những lớp có thứ tự giảm dần theo mã lớp học.

	class_id	date_start	class_status	name	ts_name	subject_name	class_type_name	phone_number
1	1	2022-10-07...	Da giao	quan Phu Nhuan	Truc tiep	Ngu Van	Lop 4	0905123457
2	1	2022-10-07...	Da giao	quan Phu Nhuan	Truc tiep	KHXH	Lop 4	0905123457
3	1	2022-10-07...	Da giao	quan Phu Nhuan	Truc tiep	Toan	Lop 4	0905123457

Hình 65: Kết quả lọc những lớp theo "SDT: 0905123457".



	class_id	date_start	class_status	name	ts_name	subject_name	class_type_name	phone_number
1	1	2022-10-07...	Da giao	quan Phu Nhuan	Truc tiep	Ngu Van	Lop 4	0905123457
2	1	2022-10-07...	Da giao	quan Phu Nhuan	Truc tiep	KHXH	Lop 4	0905123457
3	1	2022-10-07...	Da giao	quan Phu Nhuan	Truc tiep	Toan	Lop 4	0905123457
4	4	2023-05-30...	Da giao	quan 1	Truc tuyen	Tin Hoc	On luyen Tin hoc	0912987654
5	5	2023-06-19...	Da giao	quan Tan Phu	Truc tiep	Tieng Anh	Lop 11	0987654321
6	5	2023-06-19...	Da giao	quan Tan Phu	Truc tiep	Ngu Van	Lop 11	0987654321
7	5	2023-06-19...	Da giao	quan Tan Phu	Truc tiep	KHTN	Lop 11	0987654321

Hình 66: Kết quả lọc những lớp có ngày bắt đầu từ "2022-10-07" đến "2023-10-07".

	class_id	date_start	class_status	name	ts_name	subject_name	class_type_name	phone_number
1	2	2022-08-20...	Da giao	TP. Thu Duc	Truc tiep	KHTN	Lop 8	0937456789
2	5	2023-06-19...	Da giao	quan Tan Phu	Truc tiep	KHTN	Lop 11	0987654321
3	7	2024-06-08...	Da giao	quan Binh Tan	Truc tiep	KHTN	Lop 9	0945678912

Hình 67: Kết quả lọc những lớp có "Kiểu dạy: Trực tiếp" với "Trạng thái: Da giao" và có "Môn học: KHTN".

The screenshot shows a database query results window. At the top, there is a message area with the following text:
Msg 50000, Level 16, State 1, Procedure FindClassByFilter, Line 70 [Batch Start Line 39]
Khong tim thay du lieu phu hop yeu cau.
Completion time: 2024-12-10T20:10:10.6786607+07:00

Below the message area, there is a progress bar indicating 100% completion. At the bottom, a status bar displays:
Query completed with errors. ZMRM (16.0 RTM) | ZMRM\DELL (52) | Ass2-2 | 00:00:00 | 0 rows.

Hình 68: Kết quả lọc những lớp theo "Khối lớp: Lớp 1" với "Kiểu dạy: Trực tuyến", "Trạng thái: Chưa giao" và có "Môn học: Tiếng Anh".

	class_id	date_start	class_status	name	ts_name	subject_name	class_type_name	phone_number
1	1	2022-10-07 ...	Da giao	quan Phu Nhuan	Truc tiep	Toan	Lop 4	0905123457

Hình 69: Kết quả lọc những lớp có các giá trị sau: "Khối lớp: Lớp 4" với "Kiểu dạy: Trực tiếp", "Khu vực: Quận Phú Nhuận", "Trạng thái: Da giao", "Môn học: Toán", SDT: 0905123457, ngày bắt đầu từ "2022-01-01" đến "2024-01-01" và sắp xếp theo thứ tự giảm dần.

3.3.2 Thủ tục 2

Thủ tục get_tutor_class_summary

- Mô tả thủ tục: Thủ tục get_tutor_class_summary được sử dụng trong use-case khi Admin muốn biết thông tin bao gồm giáo sư có tổng số lớp học và tổng số doanh thu từ các lớp mà họ đã dạy. Doanh thu của giáo sư có thể được tính dựa trên số lượng lớp học họ giảng dạy và lương của giáo sư gắn với các lớp đó. Bên cạnh đó, admin còn có thể lọc các giáo sư dựa trên min_class_num và min_class_money, nghĩa là admin có thể lọc ra được giáo sư có tổng số lớp học lớn hơn một số nào đó, hay tổng doanh thu lớn hơn một số nào đó, điều này có thể cho admin biết được giáo sư nào đang là ngôi sao của hệ thống, admin cũng có thể dựa vào dữ liệu đầu ra để có thể tạo ra Bảng vàng trong hệ thống nhằm tăng tính cạnh tranh giữa các giáo sư với nhau



- Thủ tục `get_tutor_class_summary` gồm các thao tác chính sau:
 - **Tính Toán Vị Trí Bắt Đầu Cho Phân Trang:** Thủ tục tính toán giá trị `@offset`, vị trí bắt đầu của trang kết quả dựa trên tham số `@page_number` và `@page_size`:
 - * `@offset` được tính bằng công thức: $(@page_number - 1) * @page_size$.
 - **Lọc Lớp Dạy Theo Tham Số Đầu Vào:** Thủ tục lọc các giảng viên dựa trên các tham số `@min_class_num` và `@min_class_money`, nếu các tham số này được cung cấp (không phải NULL). Các giảng viên có số lớp dạy và thu nhập thấp hơn các giá trị này sẽ bị loại bỏ.
 - **Tính Tổng Số Lớp Và Thu Nhập:** Dùng các bảng phụ để tính tổng số lớp mà giảng viên đã dạy và tổng thu nhập (lương lớp trừ đi phí hoa hồng), sau đó nhóm theo `tutor_id`.
 - **Kết Hợp Thông Tin Từ Các Bảng Khác:** Sau khi tính toán tổng số lớp và thu nhập, thủ tục kết hợp thông tin với bảng `tutor`, `staff`, và `oft_user` để lấy các thông tin chi tiết về giảng viên, bao gồm họ tên, số điện thoại, giới tính, ngày sinh, và tiểu sử.
 - **Phân Trang Kết Quả:** Thủ tục sử dụng `OFFSET` và `FETCH NEXT` để phân trang kết quả trả về, dựa trên giá trị của `@offset` và `@page_size`.
 - **Sắp Xếp Kết Quả:** Kết quả trả về được sắp xếp theo `class_count` giảm dần, tức là giảng viên có số lớp dạy nhiều nhất sẽ được liệt kê trước.
 - **Xử Lý Lỗi:** Nếu xảy ra lỗi trong quá trình thực thi, thủ tục sẽ chuyển sang phần xử lý lỗi trong khối `CATCH`. Các thông tin về lỗi bao gồm:
 - * `error_line`: Số dòng lỗi xảy ra trong thủ tục.
 - * `error_message`: Thông điệp chi tiết về lỗi.
 - * `error_number`: Mã lỗi.
 - * `error_procedure`: Tên thủ tục gây ra lỗi.
 - * `error_severity`: Mức độ nghiêm trọng của lỗi.
 - * `error_state`: Trạng thái lỗi.
- Thủ tục sẽ ném lại lỗi bằng cách sử dụng `THROW` để ứng dụng có thể xử lý hoặc hiển thị thông báo lỗi.
- Input:
 - `@min_class_num` (INT, tùy chọn): Số lớp tối thiểu mà giảng viên phải dạy. Mặc định là NULL. Nếu được cung cấp, sẽ lọc giảng viên có số lớp dạy lớn hơn hoặc bằng số này.
 - `@min_class_money` (BIGINT, tùy chọn): Thu nhập tối thiểu mà giảng viên phải có. Mặc định là NULL. Nếu được cung cấp, sẽ lọc giảng viên có tổng thu nhập (lương lớp trừ đi phí hoa hồng) lớn hơn hoặc bằng giá trị này.
 - `@page_number` (INT, tùy chọn): Số trang cho phân trang. Mặc định là 1.
 - `@page_size` (INT, tùy chọn): Số bản ghi mỗi trang cho phân trang. Mặc định là 10.
 - Output: `NONE`
 - Thủ tục sẽ đưa ra một result set bao gồm các cột sau:
 - **`tutor_id`** (INT): Mã định danh duy nhất của giảng viên.



- **tutor_name** (VARCHAR): Họ và tên giảng viên.
- **phone_number** (VARCHAR): Số điện thoại của giảng viên.
- **sex** (VARCHAR): Giới tính của giảng viên.
- **date_of_birth** (DATE): Ngày sinh của giảng viên.
- **biography** (TEXT): Tiểu sử hoặc thông tin bổ sung về giảng viên.
- **number_of_class** (INT): Tổng số lớp mà giảng viên đã dạy.
- **earned_money** (BIGINT): Tổng thu nhập của giảng viên (tính là lương lớp trừ đi phí hoa hồng).

- Câu lệnh tạo thủ tục:

```
-- Create a new stored procedure called 'get_tutor_class_summary' in
-- schema 'dbo'
-- Drop the stored procedure if it already exists
IF EXISTS (
SELECT *
    FROM INFORMATION_SCHEMA.ROUTINES
WHERE SPECIFIC_SCHEMA = N'dbo'
    AND SPECIFIC_NAME = N'get_tutor_class_summary'
    AND ROUTINE_TYPE = N'PROCEDURE'
)
DROP PROCEDURE dbo.get_tutor_class_summary
GO

-- Create the stored procedure in the specified schema
CREATE PROCEDURE dbo.get_tutor_class_summary
    @min_class_num INT = NULL,
    @min_class_money BIGINT = NULL,
    @page_number INT = 1,
    @page_size INT = 10

AS
BEGIN
    BEGIN TRY
        -- Calculate the starting row for pagination
        DECLARE @offset INT = (@page_number - 1) * @page_size;

        -- SELECTION
        SELECT
            -- basic infor of tutor
            id AS [tutor_id],
            u.full_name AS [tutor_name],
            u.phone_number AS [phone_number],
            u.user_sex AS [sex],
            s.date_of_birth AS [date_of_birth],
            t.bio AS [biography],
            -- useful infor summary
            class_count AS [number_of_class],
            money AS [earned_money]
```



```
FROM
(
    SELECT
        c.tutor_id                               AS [id],
        COUNT(c.class_id)                         AS [class_count],
        SUM(c.salary - c.commission_fee)          AS [money]

    FROM
        dbo.class c
    WHERE
        c.class_status = 'Da giao'
    GROUP BY c.tutor_id
    -- filter if NOT NULL
    HAVING
        ( COALESCE(@min_class_num, 0) <= COUNT(c.class_id) )
    AND
        ( COALESCE(@min_class_money, 0) <= SUM(c.salary - c.
            commission_fee))
) AS temp
-- resolve tutor
JOIN
-- resolve tutor
dbo.tutor t
ON temp.id = t.tutor_id
-- resolve staff
JOIN
dbo.staff s
ON t.tutor_id = s.staff_id
-- resolve user
JOIN
dbo.oft_user u
ON u.user_id = s.staff_id

ORDER BY [number_of_class] DESC
OFFSET @offset ROWS
FETCH NEXT @page_size ROWS ONLY

END TRY

BEGIN CATCH

    -- For debugging as SA
    SELECT ERROR_LINE() AS [error line],
    ERROR_MESSAGE() AS [message],
    ERROR_NUMBER() AS [number],
    ERROR_PROCEDURE() AS [procedure],
    ERROR_SEVERITY() AS [severity],
    ERROR_STATE() AS [state];

    -- For application to show error
    THROW;

END CATCH
```



class_id	class_deposit	class_status	commission_fee	requirements	date_start
21	40043	500000	Da giao	150000	Math major
22	40044	700000	Da giao	130000	Biology major
23	40045	600000	Da giao	110000	Chemistry major
24	40047	300000	Da giao	80000	History major
25	40048	450000	Da giao	100000	English major
26	40051	350000	Da giao	120000	Physics major
27	40053	450000	Da giao	110000	Economics major
28	40054	550000	Da giao	100000	Computer Science major
29	40056	600000	Da giao	140000	Engineering major
30	40057	450000	Da giao	90000	Literature major
31	40058	450000	Da giao	90000	Student at HCMUT
32	40059	500000	Da giao	90000	Student at HCMUS
33	40060	950000	Da giao	90000	Student at HCMUS
34	40061	850000	Da giao	90000	Student at HCMUS
35	40063	980000	Da giao	90000	Student at HCMUS
36	40064	950000	Da giao	90000	Student at HCMUS with a strong ...
37	40065	800000	Da giao	90000	Student at HCMUS, majoring in C...
38	40066	950000	Da giao	90000	Student at HCMUS with a strong ...
39	40067	800000	Da giao	90000	Student at HCMUS, majoring in C...

Hình 70: Dữ liệu mẫu thực thi procedure

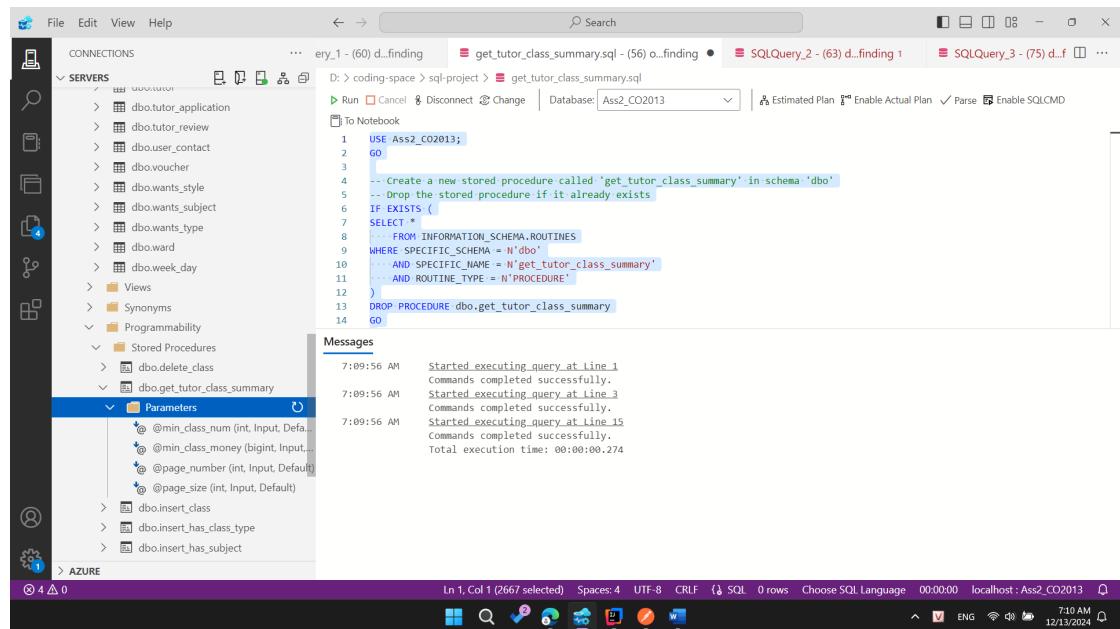
END
GO

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc tạo thủ tục
- Nhóm câu lệnh thực thi thủ tục mẫu:

```
-- example to execute the stored procedure we just created
-- DEFAULT VALUE
EXECUTE dbo.get_tutor_class_summary
-- @min_class_num = 5,
-- @min_class_money = 1000000,
-- @page_number = 1,
-- @page_size = 10
GO

-- example to execute the stored procedure we just created
-- PAGE 1, SIZE = 5
EXECUTE dbo.get_tutor_class_summary
-- @min_class_num = 5,
-- @min_class_money = 1000000,
@page_number = 1,
@page_size = 5
GO

-- example to execute the stored procedure we just created
```



Hình 71: Tạo thủ tục thành công

```
-- PAGE 2, SIZE = 5
EXECUTE dbo.get_tutor_class_summary
-- @min_class_num = 5,
-- @min_class_money = 1000000,
@page_number = 2,
@page_size = 5
GO

-- example to execute the stored procedure we just created
-- TUTOR WHO TEACHES OVER 5 CLASSES
EXECUTE dbo.get_tutor_class_summary
@min_class_num = 3,
-- @min_class_money = 1000000,
@page_number = 1,
@page_size = 5
GO

-- example to execute the stored procedure we just created
-- CHOOSE TUTOR WHO HAS THE MONEY OVER 2000000
EXECUTE dbo.get_tutor_class_summary
-- @min_class_num = 5,
@min_class_money = 2000000,
@page_number = 1,
@page_size = 5
GO
```



The screenshot shows the SSMS interface. In the Object Explorer (left), under 'get_tutor_class_summary', there is a 'Parameters' node. The 'date_of_birth' parameter is highlighted. In the Results pane (right), the output of the stored procedure is displayed as a table:

date_of_birth	biography	number_of_class	earned_money
1996-02-22 00:00:00.0000000	Voi niem dam me day hoc,toi cam ket mang den nhung buoi...	8	7650000
1997-10-20 00:00:00.0000000	Voi phong cach day hoc sang tao va de hieu,toi giup hoc...	8	10450000
1994-05-06 00:00:00.0000000	Gia su giao kinh nghiem,san sang dong hanh cung hoc sinh...	5	3720000
2003-07-08 00:00:00.0000000	Toi la gia su luon tau moi truong hoc tap thao mai,giup...	4	5659000
1998-06-12 00:00:00.0000000	Gia su giup hoc vien tu tin giao tiep voi cai thienn diem...	3	1930000
2000-06-01 00:00:00.0000000	Voi phuong phap day hoc linh hoạt,toi giup hoc sinh cai...	3	3110000
1997-07-07 00:00:00.0000000	Giang day tan tam voi tung hoc sinh,cam ket nang cao kie...	2	3650000

Hình 72: Kết quả thực thi thủ tục với các giá trị default

```
-- example to execute the stored procedure we just created
-- CHOOSE TUTOR WHO HAS TEACHES OVER 5 CLASSES AND HAS THE MONEY OVER
-- 4000000
EXECUTE dbo.get_tutor_class_summary
@min_class_num = 5,
@min_class_money = 4000000,
@page_number = 1,
@page_size = 5
GO
```

- Kết quả các màn hình hiển thị từ DBMS minh họa cho việc kiểm tra thủ tục



The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, under the 'Servers' node, there is a 'Parameters' folder which contains four parameters: @min_class_num, @min_class_money, @page_number, and @page_size. In the main pane, a query window displays the following code:

```
-- example to execute the stored procedure we just created
-- PAGE 1, SIZE = 5
EXECUTE dbo.get_tutor_class_summary
    --@min_class_num = 5,
    --@min_class_money = 1000000,
    @page_number = 1,
    @page_size = 5
GO
```

The results pane shows a table with the following data:

date_of_birth	biography	number_of_class	earned_money
1996-02-22 00:00:00.000000	Voi niem dam me day hoc,toi cam ket mang den nhung buoi...	8	7650000
1997-10-20 00:00:00.000000	Voi phong cach day hoc sanga tao va de hieu,toi giup hoc...	8	10450000
1994-05-08 00:00:00.000000	Gia su giao kinh nghiem,san sang dong hanh cung hoc sinh...	5	3720000
2003-07-08 00:00:00.000000	Toi la gia su luon tao moi truong hoc tap thao mai,giup...	4	5650000
1998-06-12 00:00:00.000000	Gia su giup hoc vien tu tin giao tiep va cai thienc diem...	3	1930000

Hình 73: Kết quả của trang 1, size của một trang là 5

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, under the 'Servers' node, there is a 'Parameters' folder which contains four parameters: @min_class_num, @min_class_money, @page_number, and @page_size. In the main pane, a query window displays the following code:

```
-- example to execute the stored procedure we just created
-- PAGE 1, SIZE = 5
EXECUTE dbo.get_tutor_class_summary
    --@min_class_num = 5,
    --@min_class_money = 1000000,
    @page_number = 2,
    @page_size = 5
GO
```

The results pane shows a table with the following data:

date_of_birth	biography	number_of_class	earned_money
2000-06-01 00:00:00.000000	Voi phuong phap day hoc linh hoat,toi giup hoc sinh cai...	3	3110000
1997-07-07 00:00:00.000000	Giang day can tam voi tung hoc sinh,cam ket nang cao kie...	2	3650000

Hình 74: Kết quả của trang 2, size của một trang là 5



The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, under the 'Servers' node, there is a connection to 'Ass2_CO2013'. In the 'Stored Procedures' folder, there is a 'dbo.get_tutor_class_summary' procedure. The 'Parameters' node contains four parameters: '@min_class_num', '@min_class_money', '@page_number', and '@page_size'. In the 'SQL' tab, the following code is written:

```
-- example to execute the stored procedure we just created
-- TUTOR WHO TEACHES OVER 5 CLASSES
EXECUTE dbo.get_tutor_class_summary
    @min_class_num = 3,
    --@min_class_money = 1000000,
    @page_number = 1,
    @page_size = 5
GO
```

The 'Results' tab displays the output of the stored procedure:

date_of_birth	biography	number_of_class	earned_money
1996-02-22 00:00:00.000000	Voi niem dam me day hoc, toi cam ket mang den nhung buoi ...	8	7650000
1997-10-20 00:00:00.000000	Voi phong cach day hoc sang tao va de hieu, toi giup hoc ...	8	10450000
1994-05-06 00:00:00.000000	Gia su giao kinh nghiem, san sang dong hanh cung hoc sinh...	5	3720000
2003-07-08 00:00:00.000000	Toi la gia su luon tau moi truong hoc tap thoi mai, giup...	4	5659000
2000-06-01 00:00:00.000000	Voi phuong phap day hoc linh hoạt, toi giup hoc sinh cai ...	3	3110000

Hình 75: Kết quả khi thêm bộ lọc ràng chỉ chọn những giáo sư dạy trên 3 lớp học

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, under the 'Servers' node, there is a connection to 'Ass2_CO2013'. In the 'Stored Procedures' folder, there is a 'dbo.get_tutor_class_summary' procedure. The 'Parameters' node contains four parameters: '@min_class_num', '@min_class_money', '@page_number', and '@page_size'. In the 'SQL' tab, the following code is written:

```
-- example to execute the stored procedure we just created
-- CHOOSE TUTOR WHO HAS THE MONEY OVER 2000000
EXECUTE dbo.get_tutor_class_summary
    @min_class_num = 5,
    --@min_class_money = 2000000,
    @page_number = 1,
    @page_size = 5
GO
```

The 'Results' tab displays the output of the stored procedure:

date_of_birth	biography	number_of_class	earned_money
1996-02-22 00:00:00.000000	Voi niem dam me day hoc, toi cam ket mang den nhung buoi ...	8	7650000
1997-10-20 00:00:00.000000	Voi phong cach day hoc sang tao va de hieu, toi giup hoc ...	8	10450000
1994-05-06 00:00:00.000000	Gia su giao kinh nghiem, san sang dong hanh cung hoc sinh...	5	3720000
2003-07-08 00:00:00.000000	Toi la gia su luon tau moi truong hoc tap thoi mai, giup...	4	5659000
2000-06-01 00:00:00.000000	Voi phuong phap day hoc linh hoạt, toi giup hoc sinh cai ...	3	3110000

Hình 76: Kết quả khi thêm bộ lọc ràng chỉ chọn những giáo sư có tổng thu nhập trên 2000000



The screenshot shows the SSMS interface with the following details:

- Connections:** ery_1 - (60) d...finding, get_tutor_class_summary.sql - (56) o...finding, SQLQuery_2 - (63) d...finding, SQLQuery_3 - (75) d...finding.
- Servers:** A tree view of database objects under 'Servers' including 'dbo.tutor_application', 'dbo.tutor_review', 'dbo.user_contact', 'dbo.voucher', 'dbo.wants_style', 'dbo.wants_subject', 'dbo.wants_type', 'dbo.ward', 'dbo.week_day', 'Views', 'Synonyms', 'Programmability', 'Stored Procedures', 'dbo.delete_class', 'dbo.get_tutor_class_summary', and 'Parameters' (with parameters: @min_class_num, @min_class_money, @page_number, @page_size).
- Script pane:** Shows the T-SQL code for the stored procedure 'dbo.get_tutor_class_summary'. The code includes comments and variable declarations for 'min_class_num', 'min_class_money', 'page_number', and 'page_size'.
- Results pane:** Displays the output of the stored procedure. The table has columns: date_of_birth, biography, number_of_class, and earned_money. Two rows are shown:

date_of_birth	biography	number_of_class	earned_money
1996-02-22 00:00:00.0000000	Voi niem dam me day hoc,toi cam ket mang den nhung buoi ...	8	7650000
1997-10-20 00:00:00.0000000	Voi phong cach day hoc sang tao va de hieu,toi giup hoc ...	8	10450000
- Status bar:** Shows 'Ln 183, Col 1 (184 selected)', 'Spaces: 4', 'UTF-8', 'CRLF', 'SQL', '2 rows', 'Choose SQL Language', '00:00:00', 'localhost : Ass2_CO2013', '7:22 AM', 'ENG', '12/13/2024'.

Hình 77: Kết quả khi thêm bộ lọc kết hợp, chỉ chọn những giáo sư có tổng số lớp dạy trên 5 và tổng thu nhập trên 4000000



3.4 Hàm

3.4.1 Hàm tính doanh thu thực tế, kỳ vọng và giảm giá

1. Yêu cầu:

- Tính toán doanh thu thực tế: tổng phí hoa hồng đã nhận được ở các lớp.
- Tính toán doanh thu kỳ vọng/ mong muốn: tổng phí hoa hồng các lớp.
- Tính toán phần giảm giá: tổng hiệu hoa hồng các lớp trừ đi hoa hồng nhận được (*commission_fee - bill_money*).

2. Input:

- Kiểu tính toán @Types: Nhận hai giá trị là ALL hoặc Subject (tính tất cả các lớp hoặc tính theo môn học).
- Giá trị tính toán @InputValues: Đây là tên môn học tương ứng khi muốn tính toán doanh thu, @InputValues có thể mang giá trị bất kì khi so sánh là ALL và phải là các môn có trong cơ sở dữ liệu nếu kiểu là Subject.

3. Kết quả: Kết quả hàm trả về chứa trong bảng sau:

```
RETURNS @ResultProfit TABLE (
    Current_Revenue      DECIMAL(18, 2) DEFAULT 0,
    Expected_Revenue     DECIMAL(18, 2) DEFAULT 0,
    Discount_Revenue    DECIMAL(18, 2) DEFAULT 0,
    Error_Message        NVARCHAR(255) NULL -- Error
)
```

Listing 57: Bảng kết quả hàm tính doanh thu

- Trong đó Current_Revenue là doanh thu hiện có, Expected_Revenue là doanh thu kỳ vọng, Discount_Revenue là tổng giảm giá và Error_Message là trường hợp lỗi.

4. SQL:

```
CREATE FUNCTION Calculate_Revenue_All()
RETURNS @ResultProfit TABLE (
    Current_Revenue      DECIMAL(18, 2) NULL ,
    Expected_Revenue     DECIMAL(18, 2) NULL ,
    Discount_Revenue    DECIMAL(18, 2) NULL
)
AS
BEGIN

    DECLARE @Current_Revenue DECIMAL(18, 2) = 0,
            @Expected_Revenue DECIMAL(18, 2) = 0,
            @Discount_Revenue DECIMAL(18, 2) = 0;

    DECLARE @TempTable TABLE (
        ClassID INT,
        Commission_Fee DECIMAL(18, 2),
        BillMoney DECIMAL(18, 2),
        BillStatus NVARCHAR(50),
        BillType NVARCHAR(50)
    );

```



```
INSERT INTO @TempTable (ClassID, Commission_Fee, BillMoney,
BillStatus, BillType)
SELECT
    C.class_id,
    C.commission_fee,
    B.bill_money,
    B.bill_status,
    B.bill_type
FROM [Ass2_C02013].[dbo].class C
LEFT JOIN [Ass2_C02013].[dbo].belongs_to BE ON C.class_id = BE.
    class_id
LEFT JOIN [Ass2_C02013].[dbo].bill B ON BE.bill_id = B.bill_id;

DECLARE temp_cursor CURSOR FOR
SELECT ClassID, Commission_Fee, BillMoney, BillStatus, BillType
FROM @TempTable;
-- Biển l_ud_lieu_trong_mi_l_n_duyt
DECLARE @ClassID INT,
        @Commission_Fee DECIMAL(18, 2),
        @BillMoney DECIMAL(18, 2),
        @BillStatus NVARCHAR(50),
        @BillType NVARCHAR(50);

OPEN temp_cursor;

FETCH NEXT FROM temp_cursor INTO @ClassID, @Commission_Fee,
    @BillMoney, @BillStatus, @BillType;

WHILE @@FETCH_STATUS = 0
BEGIN

    IF (@Commission_Fee >0) SET @Expected_Revenue =
        @Expected_Revenue + @Commission_Fee;

    IF @BillStatus = 'Xác nhận' AND @BillType = 'Phi hoa hồng'
    BEGIN
        SET @Current_Revenue += @BillMoney;

        IF @BillMoney < @Commission_Fee
            SET @Discount_Revenue += (@Commission_Fee -
                @BillMoney);
    END;

    FETCH NEXT FROM temp_cursor INTO @ClassID, @Commission_Fee,
        @BillMoney, @BillStatus, @BillType;
END;

CLOSE temp_cursor;
DEALLOCATE temp_cursor;
```



```
INSERT INTO @ResultProfit (Current_Revenue, Expected_Revenue,
                           Discount_Revenue)
VALUES (@Current_Revenue, @Expected_Revenue, @Discount_Revenue);
RETURN;
END;
CREATE FUNCTION Calculate_Revenue_Subject(@SubjectID BIGINT)
RETURNS @ResultProfit TABLE (
    Current_Revenue      DECIMAL(18, 2) NULL,
    Expected_Revenue     DECIMAL(18, 2) NULL,
    Discount_Revenue     DECIMAL(18, 2) NULL,
    Error_Message        NVARCHAR(255) NULL -- C t th ng b o l i
)
AS
BEGIN

    DECLARE @Current_Revenue DECIMAL(18, 2) = 0,
            @Expected_Revenue DECIMAL(18, 2) = 0,
            @Discount_Revenue DECIMAL(18, 2) = 0;

    DECLARE @TempTable TABLE (
        ClassID INT,
        Commission_Fee DECIMAL(18, 2),
        BillMoney DECIMAL(18, 2),
        BillStatus NVARCHAR(50),
        BillType NVARCHAR(50)
    );

    INSERT INTO @TempTable (ClassID, Commission_Fee, BillMoney,
                           BillStatus, BillType)
    SELECT
        C.class_id,
        C.commission_fee,
        B.bill_money,
        B.bill_status,
        B.bill_type
    FROM [Ass2_C02013].[dbo].class C
    LEFT JOIN [Ass2_C02013].[dbo].has_subject H ON C.class_id = H.
        class_id
    LEFT JOIN [Ass2_C02013].[dbo].belongs_to BE ON C.class_id = BE.
        class_id
    LEFT JOIN [Ass2_C02013].[dbo].bill B ON BE.bill_id = B.bill_id
    WHERE H.subject_id = @SubjectID;

    IF NOT EXISTS (SELECT 1 FROM @TempTable)
    BEGIN
        INSERT INTO @ResultProfit (Current_Revenue, Expected_Revenue,
                                   Discount_Revenue, Error_Message)
        VALUES (NULL, NULL, NULL, 'Subject does not exist');
        RETURN;
    END

    DECLARE temp_cursor CURSOR FOR
        SELECT ClassID, Commission_Fee, BillMoney, BillStatus, BillType
```



```
FROM @TempTable;

DECLARE @ClassID INT,
        @Commission_Fee DECIMAL(18, 2),
        @BillMoney DECIMAL(18, 2),
        @BillStatus NVARCHAR(50),
        @BillType NVARCHAR(50);

OPEN temp_cursor;

FETCH NEXT FROM temp_cursor INTO @ClassID, @Commission_Fee,
        @BillMoney, @BillStatus, @BillType;

WHILE @@FETCH_STATUS = 0
BEGIN
    IF (@Commission_Fee > 0) SET @Expected_Revenue =
        @Expected_Revenue + @Commission_Fee;

    IF @BillStatus = 'Xác nhận' AND @BillType = 'Phi hóa đơn'
    BEGIN
        SET @Current_Revenue += @BillMoney;

        IF @BillMoney < @Commission_Fee
            SET @Discount_Revenue += (@Commission_Fee -
                @BillMoney);
    END;

    FETCH NEXT FROM temp_cursor INTO @ClassID, @Commission_Fee,
        @BillMoney, @BillStatus, @BillType;
END;

CLOSE temp_cursor;
DEALLOCATE temp_cursor;

INSERT INTO @ResultProfit (Current_Revenue, Expected_Revenue,
    Discount_Revenue)
VALUES (@Current_Revenue, @Expected_Revenue, @Discount_Revenue);

RETURN;
END;
CREATE FUNCTION Calculate_Revenue(@Types VARCHAR(255), @InputValues
    VARCHAR(255))
RETURNS @ResultProfit TABLE (
    Current_Revenue      DECIMAL(18, 2) DEFAULT 0,
    Expected_Revenue     DECIMAL(18, 2) DEFAULT 0,
    Discount_Revenue     DECIMAL(18, 2) DEFAULT 0,
    Error_Message        NVARCHAR(255) NULL -- C t th ng b o l i
)
AS
BEGIN
    IF (@Types IS NULL OR @Types = '')
    BEGIN
        INSERT INTO @ResultProfit (Current_Revenue, Expected_Revenue,
```



```
        Discount_Revenue , Error_Message)
    VALUES (NULL , NULL , NULL , 'Invalid type');
    RETURN;
END;

IF (@InputValues IS NULL OR @InputValues = '') AND @Types != 'ALL'
,
BEGIN
    INSERT INTO @ResultProfit (Current_Revenue , Expected_Revenue ,
        Discount_Revenue , Error_Message)
    VALUES (NULL , NULL , NULL , 'Invalid input values');
    RETURN;
END;

IF (@Types = 'ALL')
BEGIN
    INSERT INTO @ResultProfit
    SELECT Current_Revenue , Expected_Revenue , Discount_Revenue ,
        NULL
    FROM [Ass2_C02013].[dbo].Calculate_Revenue_All();
    RETURN;
END;

IF (@Types = 'Subject')
BEGIN
    DECLARE @SubjectID BIGINT;

    SELECT @SubjectID = subject_id
    FROM [Ass2_C02013].[dbo].subject
    WHERE @InputValues = subject_name;

    IF @SubjectID IS NULL
    BEGIN

        INSERT INTO @ResultProfit (Current_Revenue ,
            Expected_Revenue , Discount_Revenue , Error_Message)
        VALUES (NULL , NULL , NULL , 'Subject does not exist');
        RETURN;
    END;

    INSERT INTO @ResultProfit
    SELECT Current_Revenue , Expected_Revenue , Discount_Revenue ,
        NULL
    FROM [Ass2_C02013].[dbo].Calculate_Revenue_Subject(@SubjectID
        );
    RETURN;
END;

INSERT INTO @ResultProfit (Current_Revenue , Expected_Revenue ,
    Discount_Revenue , Error_Message)
VALUES (NULL , NULL , NULL , 'Invalid parameters');
RETURN;
END;
```



Listing 58: Các câu lệnh hàm tính doanh thu

5. Chạy thủ tục:

```
SELECT
    C.class_id,
    C.commission_fee,
    B.bill_money,
    B.bill_status,
    B.bill_type
FROM [Ass2_C02013].[dbo].class C
LEFT JOIN [Ass2_C02013].[dbo].belongs_to BE ON C.class_id = BE.
    class_id
LEFT JOIN [Ass2_C02013].[dbo].bill B ON BE.bill_id = B.bill_id;
SELECT
    C.class_id,
    C.commission_fee,
    B.bill_money,
    B.bill_status,
    B.bill_type
FROM [Ass2_C02013].[dbo].class C
LEFT JOIN [Ass2_C02013].[dbo].has_subject H ON C.class_id = H.
    class_id
LEFT JOIN [Ass2_C02013].[dbo].belongs_to BE ON C.class_id = BE.
    class_id
LEFT JOIN [Ass2_C02013].[dbo].bill B ON BE.bill_id = B.bill_id
WHERE H.subject_id = 2; --Toan
```

Listing 59: Left join các bảng class, belong_to, bill để quan sát tốt hơn



	class_id	commission_fee	bill_money	bill_status	bill_type
1	1	500000	200000	Xac nhan	Dong coc
2	1	500000	200000	Xac nhan	Hoan coc
3	1	500000	500000	Xac nhan	Phi hoa hong
4	2	300000	200000	Xac nhan	Dong coc
5	2	300000	200000	Xac nhan	Hoan coc
6	2	300000	270000	Xac nhan	Phi hoa hong
7	3	400000	200000	Xac nhan	Dong coc
8	3	400000	200000	Xac nhan	Hoan coc
9	3	400000	400000	Xac nhan	Phi hoa hong
10	4	100000	100000	Xac nhan	Dong coc
11	4	100000	100000	Xac nhan	Hoan coc
12	4	100000	100000	Xac nhan	Phi hoa hong
13	5	400000	300000	Xac nhan	Dong coc
14	5	400000	300000	Xac nhan	Hoan coc
15	5	400000	360000	Chua xac nhan	Phi hoa hong
16	6	300000	300000	Xac nhan	Dong coc
17	6	300000	300000	Chua xac nhan	Hoan coc
18	7	400000	300000	Chua xac nhan	Dong coc
19	8	300000	NULL	NULL	NULL
20	9	200000	200000	Chua xac nhan	Dong coc
21	10	500000	400000	Chua xac nhan	Dong coc

Hình 78: Bảng đối chiếu giá trị phí hoa hồng của từng lớp

	class_id	commission_fee	bill_money	bill_status	bill_type
1	1	500000	200000	Xac nhan	Dong coc
2	1	500000	200000	Xac nhan	Hoan coc
3	1	500000	500000	Xac nhan	Phi hoa hong
4	2	300000	200000	Xac nhan	Dong coc
5	2	300000	200000	Xac nhan	Hoan coc
6	2	300000	270000	Xac nhan	Phi hoa hong
7	3	400000	200000	Xac nhan	Dong coc
8	3	400000	200000	Xac nhan	Hoan coc
9	3	400000	400000	Xac nhan	Phi hoa hong
10	6	300000	300000	Xac nhan	Dong coc
11	6	300000	300000	Chua xac nhan	Hoan coc
12	10	500000	400000	Chua xac nhan	Dong coc

Hình 79: Bảng đối chiếu giá trị phí hoa hồng theo môn toán id=2



- Tính toán doanh thu tất cả các lớp, kết quả trả về đúng:

```
SELECT * FROM [Ass2_C02013].[dbo].Calculate_Revenue('ALL', '')  
;  
----- OR -----  
SELECT * FROM [Ass2_C02013].[dbo].Calculate_Revenue('ALL', 'Toan');
```

Listing 60: Lệnh tính doanh thu tất cả các lớp

	Current_Revenue	Expected_Revenue	Discount_Revenue	Error_Message
1	1270000.00	3100000.00	30000.00	NULL

Hình 80: Kết quả tính doanh thu

- Tính toán doanh thu theo lớp:

```
SELECT * FROM [Ass2_C02013].[dbo].Calculate_Revenue('Subject', 'Toan');
```

Listing 61: Lệnh tính doanh thu tất cả các lớp

	Current_Revenue	Expected_Revenue	Discount_Revenue	Error_Message
1	1170000.00	4700000.00	30000.00	NULL

Hình 81: Kết quả tính doanh thu

- Trường hợp lỗi:

```
SELECT * FROM [Ass2_C02013].[dbo].Calculate_Revenue(' ', '');  
SELECT * FROM [Ass2_C02013].[dbo].Calculate_Revenue('Subject', '');  
SELECT * FROM [Ass2_C02013].[dbo].Calculate_Revenue('Subject', 'Van van');
```

Listing 62: Lệnh tính doanh thu tất cả các lớp

	Current_Revenue	Expected_Revenue	Discount_Revenue	Error_Message
1	NULL	NULL	NULL	Invalid type

Hình 82: Trường hợp lỗi không có tham số



	Current_Revenue	Expected_Revenue	Discount_Revenue	Error_Message
1	NULL	NULL	NULL	Subject does not exist

Hình 83: Trường hợp môn học không hợp lệ

3.4.2 Hàm tính số lượng đơn đăng ký của gia sư

1. Yêu cầu:

- Tóm tắt thông tin về số lượng đơn xin dạy học (teaching_application) của một gia sư cụ thể dựa trên trạng thái đơn (chấp nhận, từ chối).

2. Input:

- @tutor_id (kiểu INT): ID của gia sư. Đây là giá trị bắt buộc phải hợp lệ (khác NULL và lớn hơn 0).

3. Kết quả: Kết quả hàm trả về chứa trong bảng sau:

```
TABLE
(
    tutor_id INT,
    accepted_count INT,
    denied_count INT,
    error_message NVARCHAR(255)
)
```

Listing 63: Bảng kết quả hàm tính số đơn đăng ký gia sư

Trong đó:

- tutor_id là ID của gia sư
- accepted_count là số lượng đơn "Chấp nhận"
- denied_count là số lượng đơn "Từ chối"
- error_message là thông điệp lỗi (nếu có)

4. SQL:

```
CREATE FUNCTION dbo.fn_TutorApplicationSummary
(
    @tutor_id INT
)
RETURNS @ResultTable TABLE
(
    tutor_id INT,
    accepted_count INT,
    denied_count INT,
    error_message NVARCHAR(255)
)
AS
BEGIN
    IF @tutor_id IS NULL OR @tutor_id <= 0
        BEGIN
```



```
        INSERT INTO @ResultTable (tutor_id, accepted_count,
                           denied_count, error_message)
        VALUES (NULL, NULL, NULL, 'Invalid tutor_id');
        RETURN;
    END;

DECLARE @accepted_count INT = 0;
DECLARE @denied_count INT = 0;

SELECT @accepted_count = COUNT(*)
FROM dbo.teaching_application
WHERE tutor_id = @tutor_id
    AND application_status IN ('chap nhan');

SELECT @denied_count = COUNT(*)
FROM dbo.teaching_application
WHERE tutor_id = @tutor_id
    AND application_status IN ('tu choi');

DECLARE @i INT = 0;
WHILE @i < 1
BEGIN
    INSERT INTO @ResultTable (tutor_id, accepted_count,
                           denied_count, error_message)
    VALUES (@tutor_id, @accepted_count, @denied_count, NULL);

    SET @i = @i + 1;
END;

RETURN;
END;
```

Listing 64: Các câu lệnh hàm tính đơn đăng ký gia sư

	teaching_appl_id	application_status	date_of_creation	class_id	tutor_id
1	1	chap nhan	2022-06-15 00:00:00.0000000	3	15
2	2	chap nhan	2023-05-27 00:00:00.0000000	7	19
3	3	tu choi	2023-05-06 00:00:00.0000000	9	19
4	4	chap nhan	2023-06-15 00:00:00.0000000	9	19
5	5	chap nhan	2024-07-09 00:00:00.0000000	10	15

Hình 84: Bảng đối chiếu đơn đăng ký của từng gia sư

- Hiển thị số lượng đơn đăng ký của gia sư có trong bảng

```
SELECT * FROM dbo.fn_TutorApplicationSummary(19);
```

Listing 65: Câu lệnh tính toán



	tutor_id	accepted_count	denied_count	error_message
1	19	2	1	NULL

Hình 85: Kết quả tính đơn đăng ký của gia sư có tutor_id = 19

- Trường hợp lỗi:

```
SELECT * FROM dbo.fn_TutorApplicationSummary(99);
```

Listing 66: Tính số đơn đăng ký của gia sư không có trong bảng đối chiếu

	tutor_id	accepted_count	denied_count	error_message
1	99	0	0	NULL

Hình 86: Trường hợp lỗi tutor_id không hợp lệ



4 Ứng dụng kết nối cơ sở dữ liệu

4.1 Tổng quan công nghệ

4.1.1 Công nghệ sử dụng

1. Môi trường phát triển: Web application.
2. Ngôn ngữ lập trình:
 - Frontend: Javascript (thư viện ReactJs).
 - Backend: Java, python.
3. Hệ cơ sở dữ liệu: SQL server.

4.1.2 Tổ chức thư mục code

4.1.3 Khai báo kết nối với BDMS

```
spring.application.name=online-tutor-finding

# DATABASE CONFIGURATION
spring.jpa.hibernate.ddl-auto=validate
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format-sql=true

# MS SQL SERVER
spring.datasource.url=jdbc:sqlserver://localhost:1433;databaseName=hcsdl;
    encrypt=true;trustServerCertificate=true
spring.datasource.username=sa
spring.datasource.password=123456789
```

Listing 67: Khai báo kết nối DBMS trong file application.properties

4.2 Các màn hình thử nghiệm

- Màn hình 1: Màn hình hiển thị ra danh sách các lớp học (các use-case liên quan đến quá trình quản lý lớp học của người dùng admin) Các chức năng có thể thực hiện được trên màn hình là
 - Admin có thể xem được một danh sách các lớp học (chỉ chứa các thông tin cơ bản và không chi tiết), danh sách các lớp học này được phân trang động để dễ dàng hiển thị thông tin dữ liệu mà admin muốn
 - Bên trên các thông tin cơ bản của các lớp học, sẽ có một thanh đóng vai trò là bộ lọc dữ liệu cho phép admin lọc ra các lớp học theo các tiêu chí khác nhau
 - Màn hình còn hỗ trợ admin có thể xem được các thông tin chi tiết liên quan đến 1 lớp học duy nhất, tức là admin sẽ có toàn quyền chỉnh sửa, thậm chí xóa lớp học này
 - Màn hình còn cho phép admin có quyền được tạo mới một lớp học, admin có thể nhập các thông tin cơ bản của lớp học, các môn học của lớp học đó, ... Thêm giáo sư và học sinh vào lớp bằng số điện thoại (dùng để đăng ký tài khoản), ...
 - Ảnh chụp các màn hình liên quan đến hiển thị danh sách lớp



Hình 87: Bảng các API được sử dụng trong lúc hiện thực tầng ứng dụng

- Mô tả đoạn code gọi API và render dữ liệu từ response của API hoặc sự kiện
- Mô tả API hoặc các sự kiện được gọi trong màn hình
- Màn hình 2: là màn hình cho phép admin xem danh sách các đơn đăng ký gia sư (Tutor Application)
 - Ánh chụp các màn hình liên quan đến hiển thị danh sách lớp
 - Mô tả đoạn code gọi API và render dữ liệu từ response của API hoặc sự kiện
 - Mô tả API hoặc các sự kiện được gọi trong màn hình
- Màn hình 3: Trang login
 - Chức năng: Admin nhập đúng mật khẩu và số điện thoại đăng ký để đăng nhập vào hệ thống.
 - Ánh chụp giao diện và minh họa hiển thị các thông báo lỗi và thành công.
 - Mô tả đoạn code API và render dữ liệu từ response:
 - **Chức năng chính:**
 - * Xử lý đăng nhập: Gửi yêu cầu đến API endpoint để xác thực thông tin đăng nhập (số điện thoại và mật khẩu).
 - * Hiển thị thông báo: Sử dụng Snackbar để thông báo kết quả (thành công, tài khoản không tồn tại, sai mật khẩu hoặc lỗi kết nối).
 - * Hiển thị hiệu ứng tải: Backdrop hiển thị vòng tròn loading trong thời gian xử lý yêu cầu.
 - * Chuyển hướng: Sau khi đăng nhập thành công, admin được điều hướng đến trang thông tin quản lý.



Hình 88: Giao diện đăng nhập

– **Luồng hoạt động:**

- * Người dùng nhập thông tin đăng nhập vào form.
- * Nhấn nút "Đăng nhập" để gửi yêu cầu xác thực tới server.
- * Hiển thị hiệu ứng loading (Backdrop) trong lúc xử lý.
- * Xử lý kết quả:
 - * Thành công: Hiển thị thông báo và chuyển hướng admin đến trang thông tin.
 - * Thất bại: Hiển thị thông báo lỗi phù hợp như tài khoản không tồn tại hoặc sai mật khẩu.
 - * Lỗi kết nối: Hiển thị thông báo lỗi mạng.
 - * Kết thúc: Đóng hiệu ứng loading.

- Màn hình 4: Trang home-pages hiển thị thông tin cá nhân của admin
 - + Chức năng:
 - Hiển thị thông tin cá nhân admin.
 - Thanh công cụ cho phép: Đăng xuất, xem các đơn yêu cầu tư vấn, xem các đơn đăng ký gia sư và danh sách các lớp học. + Ảnh chụp giao diện điện và minh họa hiển thị các thông báo lỗi và thành công.



The screenshot shows a list of five consultation requests (Mã đơn yêu cầu tư vấn) with the following details:

- Mã đơn yêu cầu tư vấn: 1**
Trạng thái: Đã xử lý
Học viên: Trần Thành
Môn học: Ngu Văn, KH Xa Hoi, Toán
Địa chỉ: 190, đường Lê Thanh Ton, phường Ben Thanh, quan 1, Tp. Hồ Chí Minh
Kiểu dạy: Trực tiếp (offline)
Khối lớp: 4
SĐT: 0905 12 3 4
- Mã đơn yêu cầu tư vấn: 2**
Trạng thái: Đã xử lý
Học viên: Nguyễn Hoàng Anh
Môn học: Toán, Tiếng Anh, KH Tu Nhien
Địa chỉ: 215, đường Nguyễn Văn Troi, phường 11, Phường Phú Nhuận, TP. Hồ Chí Minh
Kiểu dạy: Trực tiếp (offline)
Khối lớp: 8
SĐT: 0937 45 6 7
- Mã đơn yêu cầu tư vấn: 3**
Trạng thái: Đã xử lý
Học viên: Phạm Minh Thủ
Môn học: KH Xa Hoi, Tiếng Anh, Tin Học
Địa chỉ: 278, đường Phan Anh, phường Hiệp Tân, quan Tân Phú, Tp. Hồ Chí Minh
Kiểu dạy: Trực tuyến (online)
Khối lớp: 12
SĐT: 0912 98 7 6
- Mã đơn yêu cầu tư vấn: 4**
Trạng thái: Đã xử lý
Học viên: Lê Khanh Linh
Môn học: Tiếng Anh, Ngu Văn, KH Tu Nhien
Địa chỉ: 66, đường Cố Bá Cát, phường Cau Ông Lanh, quan 1, Tp. Hồ Chí Minh
Kiểu dạy: Trực tiếp (offline)
Khối lớp: 11
SĐT: 0987 65 4 3
- Mã đơn yêu cầu tư vấn: 5**
Trạng thái: Chưa xử lý
Học viên: Đặng Bảo Trâm
Môn học: KH Xa Hoi, Toán, Ngu Văn
Địa chỉ: 8, đường Hòa Bình, phường Hiệp Tân, quan Tân Phú, Tp. Hồ Chí Minh
Kiểu dạy: Trực tiếp (offline)
Khối lớp: 2
SĐT: 0923 12 3 7

Hình 90: Giao diện cập nhật trạng thái đơn yêu cầu tư vấn

The screenshot shows a user profile with a placeholder icon and a table of personal information:

Họ và tên	Vũ Quynh Anh
Giới tính	Nữ
Ngày sinh	20/05/1988
Quê quán	TP. Hồ Chí Minh
CCCD/ CMND	079221762419
Số điện thoại	0978 23 4 5
Email	quynhanh9876@gmail.com
Mạng xã hội	https://facebook.com/quynhanh9876

A button at the bottom left says "Đăng nhập thành công" (Login successful).

Hình 89: Giao diện homepages

- + Mô tả đoạn code API và render dữ liệu từ response.
- Màn hình 5: Trang danh sách đơn yêu cầu tư vấn + Chức năng:
 - Hiển thị các đơn yêu cầu tư vấn của học viên.
 - Cho phép: Cập nhật trạng thái đơn. + Ảnh chụp giao diện và minh họa hiển thị các thông báo lỗi và thành công. + Mô tả đoạn code API và render dữ liệu từ response.
- Màn hình 6: Trang danh sách đơn đăng ký gia sư + Chức năng:
 - Hiển thị các đơn đăng ký gia sư của học viên.
 - Cho phép: Cập nhật trạng thái đơn/ tự động thêm lớp học mới khi chuyển trạng thái



Mã đơn đăng ký gia sư 1

Trạng thái: Chưa xử lý

Môn học: Ngữ Văn, KH Xa Hoi, Toán

Học viên: Trần Thành

Gia sư: Nguyễn Việt

Khối lớp: 4

Anh

Địa chỉ: 215, đường Nguyễn Văn Trỗi, chung cư 11, quan Phu Nhuan, Tp. Hồ Chí Minh

SDT: 0905 12 3 4

Yêu cầu: Tốt nghiệp đại học chuyên ngành liên quan.

Kiểu dạy: Trực tiếp (offline)

Mã đơn đăng ký gia sư 2

Trạng thái: Đang xử lý

Môn học: Toán, Tiếng Anh, KH Tu Nien

Học viên: Nguyễn Hoàng Anh

Gia sư: Phạm Linh Nguyên

Khối lớp: 8

Địa chỉ: 216, đường Vo Van Gan, chung cư Bình Thạnh, TP. Thủ Đức, Tp. Hồ Chí Minh

SDT: 0937 45 6 7

Yêu cầu: Ít nhất 1-2 năm kinh nghiệm dạy kèm hoặc giảng dạy.

Kiểu dạy: Trực tiếp (offline)

Mã đơn đăng ký gia sư 3

Trạng thái: Đã hủy bỏ

Môn học: Tiếng Anh, Ngữ Văn, KH Tu Nien

Học viên: Lê Khanh Linh

Gia sư: Nguyễn Duy

Khối lớp: 11

Địa chỉ: 8, đường Hoa Bình, phường Hiệp Tân, quan Tân Phú, Tp. Hồ Chí Minh

SDT: 0987 65 4 3

Yêu cầu: Lịch trình linh hoạt, có thể làm việc vào cuối tuần hoặc buổi tối.

Kiểu dạy: Trực tiếp (offline)

Mã đơn đăng ký gia sư 4

Trạng thái: Đã mở lớp

Môn học: KH Xa Hoi, Toán, Ngữ Văn

Học viên: Đặng Bảo Trâm

Gia sư: Trần Tuấn Anh

Khối lớp: 2

Địa chỉ: 53, đường Lam Văn Bền, chung cư Tân Kiểng, quan 7, Tp. Hồ Chí Minh

SDT: 0923 12 3 7

Yêu cầu: Thành thạo các công cụ và nền tảng dạy học trực tuyến.

Kiểu dạy: Trực tuyến (offline)

Hình 91: Giao diện cập nhật đơn đăng ký gia sư

đơn sang 'Da mo lop'. + Ảnh chụp giao diện điện tử minh họa hiển thị các thông báo lỗi và thành công. + Mô tả đoạn code API và render dữ liệu từ response.

- Màn hình 77: Trang hiển thị bill của lớp học + Chức năng:
 - Hiển thị thông tin hóa đơn của lớp.
 - Cho phép: Cập nhật trạng thái đơn. + Ảnh chụp giao diện và minh họa hiển thị các thông báo lỗi và thành công. + Mô tả đoạn code API và render dữ liệu từ response.



Hệ thống Gia sư
Dạy kèm tại nhà

Danh sách đơn
d đăng ký, gia sư

Danh sách đơn
yêu cầu tư vấn

Danh sách lớp học

Đăng xuất

Mã hóa đơn:1

Đóng tiền cọc

Trang thái	Đã hoàn	Loại hóa đơn	Đóng tiền cọc
Tiền đặt cọc	Chưa duyệt	Mã voucher	
Gia sư	Đã hoàn	Mã lớp học	1
Admin	Chưa hoàn		
Ảnh minh chứng	Đã duyệt	Ảnh minh chứng	

Hệ thống Gia sư Dạy kèm tại nhà / TMS
Email : tutor.manager@gmail.com
Call us : 091 234 5678

Hệ thống Gia sư
Dạy kèm tại nhà

Hình 92: Giao diện cập nhật trạng thái bill



5 Phần làm thêm

5.1 Phân tạo bảng và dữ liệu mẫu: Bổ sung ràng buộc ngữ nghĩa địa chỉ của lớp học phải thuộc những địa chỉ của học sinh trong lớp học đó

```
CREATE OR ALTER TRIGGER trigger_class_address_belongs_student
ON [dbo].[class]
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE
        @student_id BIGINT,
        @addr_id BIGINT;
    SELECT
        @student_id = student_id,          -- NOT NULL
        @addr_id     = addr_id           -- NOT NULL
    FROM inserted;
    IF
    (
        @addr_id NOT IN
        (
            SELECT a.addr_id
            FROM dbo.address a
            WHERE a.user_id = @student_id
        )
    )
    BEGIN
        ROLLBACK TRANSACTION;
        THROW 50009, 'Student does not have the address', 1;
    END
END;
```

5.2 Hiện thực ràng buộc ngữ nghĩa năm của bảng cấp mà gia sư thêm vào hệ thống phải nằm trong khoảng 2 năm gần đây, hiện thực bằng thủ tục adding_new_certificate

- Thủ tục được sử dụng trong use-case khi một gia sư nào đó có mong muốn thêm một bảng cấp mà mình đạt được vào trong hệ thống gia sư nhằm làm tăng thêm background của mình, tạo sự thu hút khi có một background học tập nổi bật
- Thủ tục cũng chính là hiện thực của một ràng buộc ngữ nghĩa của hệ thống, nhằm để đảm bảo tính xác thực, cũng như thời hạn của bảng cấp, gia sư chỉ được cho phép thêm vào loại bảng cấp nếu bảng cấp đó được cấp trong vòng hai năm trở lại so với năm hiện tại, điều này cũng nhằm đảm bảo gia sư không sử dụng bằng cấp quá cũ để quảng bá cho mình
- Các thao tác chính của thủ tục: Thủ tục `dbo.adding_new_certificate` thực hiện các thao tác chính sau đây:
 - Kiểm tra năm của chứng chỉ:



- * Kiểm tra nếu @cert_year không đủ 2 năm trước năm hiện tại.
* Nếu không đạt, ném lỗi với mã lỗi 50011.
- **Kiểm tra sự tồn tại của gia sư:**
 - * Kiểm tra xem @tutor_id có tồn tại trong bảng tutor hay không.
* Nếu không tồn tại, ném lỗi với mã lỗi 50012.
- **Kiểm tra sự tồn tại của loại chứng chỉ:**
 - * Kiểm tra xem @ct_id có tồn tại trong bảng certificate_type hay không.
* Nếu không tồn tại, ném lỗi với mã lỗi 50013.
- **Kiểm tra trùng lặp chứng chỉ:**
 - * Kiểm tra xem kết hợp @cert_id và @tutor_id đã tồn tại trong bảng certificate hay chưa.
* Nếu đã tồn tại, ném lỗi với mã lỗi 50014.
- **Chèn dữ liệu vào bảng certificate:**
 - * Chèn các giá trị (@cert_id, @cert_year, @cert_grade, @tutor_id, @ct_id) vào bảng certificate.
- **Xử lý lỗi và ném lỗi (CATCH):**
 - * Bắt tất cả các lỗi có thể xảy ra trong quá trình thực thi.
* Ghi chi tiết lỗi ra bảng điều khiển và ném lại lỗi cho ứng dụng phía trên.
- Input:
 - **@cert_id (BIGINT):**
 - * Mã định danh duy nhất của chứng chỉ (*Primary Key*) cho gia sư.
 - * Được sử dụng để đảm bảo không có sự trùng lặp chứng chỉ cho cùng một gia sư.
 - **@cert_year (INT):**
 - * Năm cấp chứng chỉ.
 - * Yêu cầu: Năm chứng chỉ phải **ít nhất là 2 năm về trước** so với năm hiện tại.
 - **@cert_grade (VARCHAR(255)):**
 - * Điểm hoặc xếp hạng của chứng chỉ.
 - * Có thể là mô tả, ví dụ: "Excellent" hoặc "Grade A".
 - **@tutor_id (BIGINT):**
 - * Mã định danh của gia sư, tham chiếu đến bảng tutor.
 - * Được sử dụng để liên kết chứng chỉ với gia sư cụ thể.
 - **@ct_id (BIGINT):**
 - * Mã định danh của loại chứng chỉ, tham chiếu đến bảng certificate_type.
 - * Giúp xác định loại chứng chỉ mà gia sư đã đạt được.
- Output: Thủ tục dbo.adding_new_certificate không trả về kết quả trực tiếp, nhưng có thể trả về thông báo hoặc thông tin lỗi. Các kết quả đầu ra có thể bao gồm:
 - **Thông báo thành công** (bên trong PRINT):
 - * Ví dụ: "Certificate successfully added for tutor ID 101".
 - **Thông báo lỗi** (được ném bởi THROW):



- * **Lỗi 50011:** "The certificate year must be at least 2 years from the current year."
- * **Lỗi 50012:** "The specified tutor does not exist."
- * **Lỗi 50013:** "The specified certificate type is not supported."
- * **Lỗi 50014:** "The certificate might already exist."

– **Thông tin lỗi** (bên trong CATCH):

- * Cung cấp thông tin chi tiết về lỗi:
 - . Dòng lỗi: ERROR_LINE()
 - . Thông báo lỗi: ERROR_MESSAGE()
 - . Mã lỗi: ERROR_NUMBER()
 - . Tên thủ tục: ERROR_PROCEDURE()
 - . Mức độ nghiêm trọng: ERROR_SEVERITY()
 - . Trạng thái lỗi: ERROR_STATE()

• Câu lệnh tạo thủ tục:

```
-- Create a new stored procedure called 'adding_new_certificate' in
-- schema 'dbo'
-- Drop the stored procedure if it already exists
IF EXISTS (
SELECT *
    FROM INFORMATION_SCHEMA.ROUTINES
WHERE SPECIFIC_SCHEMA = N'dbo'
    AND SPECIFIC_NAME = N'adding_new_certificate'
    AND ROUTINE_TYPE = N'PROCEDURE'
)
DROP PROCEDURE dbo.adding_new_certificate
GO

-- Create the stored procedure in the specified schema
-- will be used in the use-case adding a new certificate
-- in the system

-- as the front-end can query the list
-- of available CERTIFICATE_TYPE, thus we don't have
-- to worry about that
CREATE PROCEDURE dbo.adding_new_certificate
    @cert_id      BIGINT,
    @cert_year    INT,
    @cert_grade   VARCHAR(255),
    @tutor_id     BIGINT,
    @ct_id        BIGINT
-- add more stored procedure parameters here
AS
BEGIN
    BEGIN TRY

        -- Check if the year is at least 2 years in the past from the
        -- current year
        IF (@cert_year < (YEAR(GETDATE()) - 2))
            THROW 50011, 'The certificate year must be at least 2
            years from the current year.', 1;
```



```
-- Check if the tutor exists
IF NOT EXISTS (SELECT 1 FROM [dbo].[tutor] WHERE tutor_id = @tutor_id)
    THROW 50012, 'The specified tutor does not exist.', 1;

-- Check if the certificate type exists
IF NOT EXISTS (SELECT 1 FROM [dbo].[certificate_type] WHERE ct_id = @ct_id)
    THROW 50013, 'The specified certificate type is not supported.', 1;

-- Check if the certificate already exists (prevent duplicate cert_id for the same tutor)
IF EXISTS (SELECT 1 FROM [dbo].[certificate] WHERE cert_id = @cert_id AND tutor_id = @tutor_id)
    THROW 50014, 'The certificate might already exist.', 1;

-- Insert the new certificate
INSERT INTO [dbo].[certificate] (cert_id, cert_year, cert_grade, tutor_id, ct_id)
VALUES (@cert_id, @cert_year, @cert_grade, @tutor_id, @ct_id);

-- FOR DBA interaction
PRINT 'Certificate successfully added for tutor ID ' + CAST(@tutor_id AS VARCHAR(50));
END TRY

BEGIN CATCH
    -- For debugging as SA
    SELECT ERROR_LINE() AS [error line],
    ERROR_MESSAGE() AS [message],
    ERROR_NUMBER() AS [number],
    ERROR_PROCEDURE() AS [procedure],
    ERROR_SEVERITY() AS [severity],
    ERROR_STATE() AS [state];

    -- For application
    THROW;
END CATCH
END
GO
```

- Màn hình tạo thủ tục thành công
- Các câu lệnh mà khi chạy thủ tục chạy ra lỗi

```
-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 1451284,
@cert_year = 2020,
@cert_grade = '8.0',
@tutor_id = 13,
```



```
-- to worry about that
CREATE PROCEDURE dbo.adding_new_certificate
    @cert_id      BIGINT,
    @cert_year    INT,
    @cert_grade   VARCHAR(255),
    @tutor_id     BIGINT,
    @ct_id        BIGINT
    -- add more stored procedure parameters here.
AS
BEGIN
    BEGIN TRY
        -- Check if the year is at least 2 years in the past from the current year.
        IF (@cert_year < (YEAR(GETDATE()) - 2))
            THROW 50011, 'The certificate year must be at least 2 years from the current year.', 1;
        -- Check if the tutor exists
        IF NOT EXISTS (SELECT 1 FROM [dbo].[tutor] WHERE tutor_id = @tutor_id)
            THROW 50012, 'The specified tutor does not exist.', 1;
        -- check if the certificate type exists
    END TRY
    BEGIN CATCH
        -- Handle errors
    END CATCH
END
```

Messages

```
9:50:25 PM Started executing_query_at_line 1
Commands completed successfully.
9:50:25 PM Started executing_query_at_line 20
Commands completed successfully.
Total execution time: 00:00:00.428
```

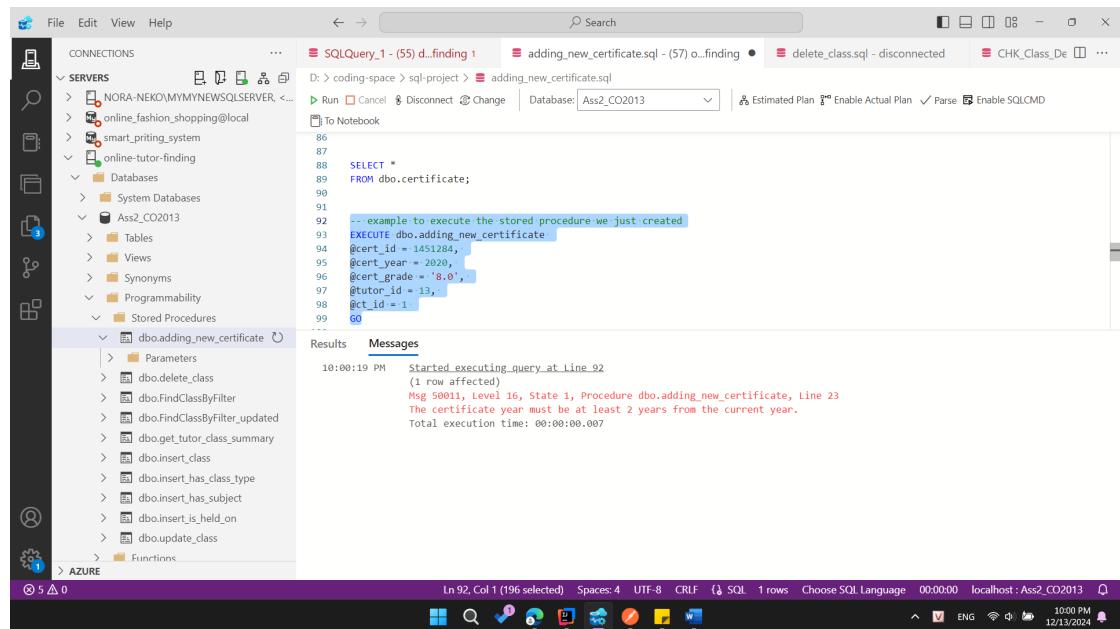
Hình 93: Câu lệnh tạo thủ tục thành công

```
@ct_id = 1
GO

-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 1452234,
@cert_year = 2024,
@cert_grade = '8.0',
@tutor_id = 31,
@ct_id = 1
GO

-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 1452234,
@cert_year = 2024,
@cert_grade = '8.0',
@tutor_id = 19,
@ct_id = 6
GO

-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 1452234,
@cert_year = 2024,
```



Hình 94: Nấm bằng cấp đã vi phạm ràng buộc ngữ nghĩa

```
@cert_grade = '8.0',
@tutor_id = 19,
@ct_id = 1
GO
```

- Màn hình kiểm tra quá trình chạy thủ tục
- Các câu lệnh mà thủ tục thực hiện chính xác

```
-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 1453234,
@cert_year = 2023,
@cert_grade = '900',
@tutor_id = 14,
@ct_id = 2
GO

-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 145454,
@cert_year = 2022,
@cert_grade = '450',
@tutor_id = 14,
@ct_id = 3
GO
```



The screenshot shows the SSMS interface with the following details:

- Servers:** NORA-NEKO\MMYNEWSQLSERVER, <...>
- Databases:** Ass2_CO2013
- Stored Procedures:** dbo.adding_new_certificate
- Code:** A script to execute the stored procedure. Line 102 contains the command `EXECUTE dbo.adding_new_certificate @cert_id = 1452234, @cert_year = 2024, @cert_grade = '8.0', @tutor_id = 31, @ct_id = 1`. Line 112 contains the comment `-- example to execute the stored procedure we just created`.
- Results:** Shows the execution log. It starts at 10:01:23 PM with the message "Started executing_query at Line 102". It then shows "(1 row affected)" and the error "Msg 50012, Level 16, State 1, Procedure dbo.adding_new_certificate, Line 27 The specified tutor does not exist." The total execution time is 00:00:00.018.
- Messages:** Shows the same log entries.
- Status Bar:** Displays "In 102, Col 1 (198 selected)", "Spaces: 4", "UTF-8", "CRLF", "SQL", "1 rows", "Choose SQL Language", "00:00:00", "localhost: Ass2_CO2013", and the date/time "12/13/2024 1001 PM".

Hình 95: Gia sư không có trong hệ thống

The screenshot shows the SSMS interface with the following details:

- Servers:** NORA-NEKO\MMYNEWSQLSERVER, <...>
- Databases:** Ass2_CO2013
- Stored Procedures:** dbo.adding_new_certificate
- Code:** A script to execute the stored procedure. Line 112 contains the command `EXECUTE dbo.adding_new_certificate @cert_id = 1452234, @cert_year = 2024, @cert_grade = '8.0', @tutor_id = 19, @ct_id = 6`. Line 122 contains the comment `-- example to execute the stored procedure we just created`.
- Results:** Shows the execution log. It starts at 10:01:59 PM with the message "Started executing_query at Line 112". It then shows "(1 row affected)" and the error "Msg 50013, Level 16, State 1, Procedure dbo.adding_new_certificate, Line 31 The specified certificate type is not supported." The total execution time is 00:00:00.009.
- Messages:** Shows the same log entries.
- Status Bar:** Displays "In 112, Col 1 (195 selected)", "Spaces: 4", "UTF-8", "CRLF", "SQL", "1 rows", "Choose SQL Language", "00:00:00", "localhost: Ass2_CO2013", and the date/time "12/13/2024 1002 PM".

Hình 96: Hệ thống gia sư hiện tại chưa hỗ trợ việc cung cấp thông tin liên quan đến loại bằng cấp này



```
-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 1452234,
@cert_year = 2024,
@cert_grade = '8.0',
@tutor_id = 19,
@ct_id = 1
GO

-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 1452234,
```

Results Messages

10:03:08 PM Started executing_query_at Line 122
(1 row affected)
Msg 50014, Level 16, State 1, Procedure dbo.adding_new_certificate, Line 35
The certificate might already exist.
Total execution time: 00:00:00.018

Ln 122, Col 1 (197 selected) Spaces: 4 UTF-8 CRLF { SQL 1 rows Choose SQL Language 00:00:00 localhost : Ass2_CO2013 10:03 PM 12/13/2024

Hình 97: Gia sư đã từng cung cấp các thông tin liên quan đến bằng cấp này

- Màn hình kiểm tra quá trình chạy các thủ tục

5.3 Phần tạo bảng và dữ liệu mẫu: Bổ sung ràng buộc ngữ nghĩa rằng tiền đặt cọc của lớp học không được lớn hơn lương nhằm đảm bảo quyền lợi cho gia sư

```
ALTER TABLE [dbo].[class]
ADD CONSTRAINT [CHK_Class_Deposit_Salary]
CHECK ([class_deposit] <= [salary]);
```

5.4 Phân hiện thực ứng dụng

5.4.1 Phần nhóm làm thêm: Thủ tục INSERT/ UPDATE vào bảng dữ liệu

- Yêu cầu: Admin cần xử lý và cập nhật Đơn đăng ký gia sư, Admin có thể tạo lớp mới dựa trên các thông tin có trong đơn.
- Mô tả: Khi trạng thái đơn có thay đổi và chuyển thành 'Da mo lop' hệ thống tự động tạo lớp mới dựa trên những thông tin có sẵn.
- SQL:

```
-- Thủ tục tạo lớp
CREATE PROCEDURE Create_Class_From_TA
@requirement VARCHAR(255),
@addr_id BIGINT,
```



The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays a tree structure of servers, databases, tables, and stored procedures. In the center, the Query Editor window contains a script for executing a stored procedure named 'dbo.adding_new_certificate'. The script includes parameters for certificate ID, year, grade, tutor ID, and class ID. The 'Messages' pane at the bottom shows the execution results, indicating that one row was affected and a new certificate was successfully added for tutor ID 14. The total execution time is listed as 00:00:00.031.

```
-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 1453234,
@cert_year = 2023,
@cert_grade = '900',
@tutor_id = 14,
@ct_id = 2
GO
-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 145454,
@cert_year = 2022,
@cert_grade = '450',
@tutor_id = 14,
@ct_id = 3
GO
```

Hình 98: Gia sư đã thêm bằng cấp thành công

The screenshot shows the SQL Server Management Studio interface. The left pane displays the database structure. The central Query Editor window runs the same stored procedure script as in Figure 98. The 'Messages' pane shows the execution completed successfully with one row affected, adding a new certificate for tutor ID 14. The execution time is 00:00:00.019.

```
-- example to execute the stored procedure we just created
EXECUTE dbo.adding_new_certificate
@cert_id = 145454,
@cert_year = 2022,
@cert_grade = '450',
@tutor_id = 14,
@ct_id = 3
GO
```

Hình 99: Gia sư đã thêm bằng cấp thành công



	user_id	full_name	cert_id	cert_year	cert_grade	ct_name
1	14	Pham Linh Nguyen	145454	2022	450	TOEFL
2	14	Pham Linh Nguyen	1451134	2023	900	IC3
3	14	Pham Linh Nguyen	1453234	2023	900	TOEIC
4	14	Pham Linh Nguyen	1453784	2022	450	TOEFL
5	14	Pham Linh Nguyen	9356712	2022	7	IELTS
6	14	Pham Linh Nguyen	14543354	2022	450	TOEIC
7	16	Nguyen Minh Hoang	14543354	2022	9.0	IELTS
8	17	Nguyen Duy	7129483	2022	915	TOEIC
9	19	Vo Thu Trang	8596742	2023	550	TOEFL
10	19	Vo Thu Trang	1452234	2024	8.0	IELTS
11	20	Nguyen Hai Anh	5379218	2024	835	IC3

Hình 100: Bảng trạng thái thông tin các loại bằng cấp sau khi thực thi thủ tục

```
@student_id BIGINT,
@ts_id BIGINT,
@tutor_id BIGINT,
@Tutor_App_Id BIGINT
AS
BEGIN
    DECLARE @NewClassID INT;
    -- Insert du lieu vao bang class
    INSERT INTO [Ass2_CO2013].[dbo].[class]
    (requirements,addr_id, student_id, ts_id, tutor_id)
    VALUES (@requirement, @addr_id, @student_id, @ts_id,@tutor_id
    );
    -- Lay id class
    SET @NewClassID = SCOPE_IDENTITY();
    -- Insert du lieu vao bang type c a class
    INSERT INTO [Ass2_CO2013].[dbo].[has_class_type] (class_id,
    class_type_id)
    SELECT @NewClassID AS class_id, class_type_id
    FROM [Ass2_CO2013].[dbo].[s_wants_type]
    WHERE ta_id = @Tutor_App_Id;
    -- Insert du lieu vao bang subject c a class
    INSERT INTO [Ass2_CO2013].[dbo].[has_subject] ( class_id,
    subject_id)
    SELECT @NewClassID AS class_id, subject_id
    FROM [Ass2_CO2013].[dbo].[s_wants_subject]
    WHERE ta_id = @Tutor_App_Id;
END;
```

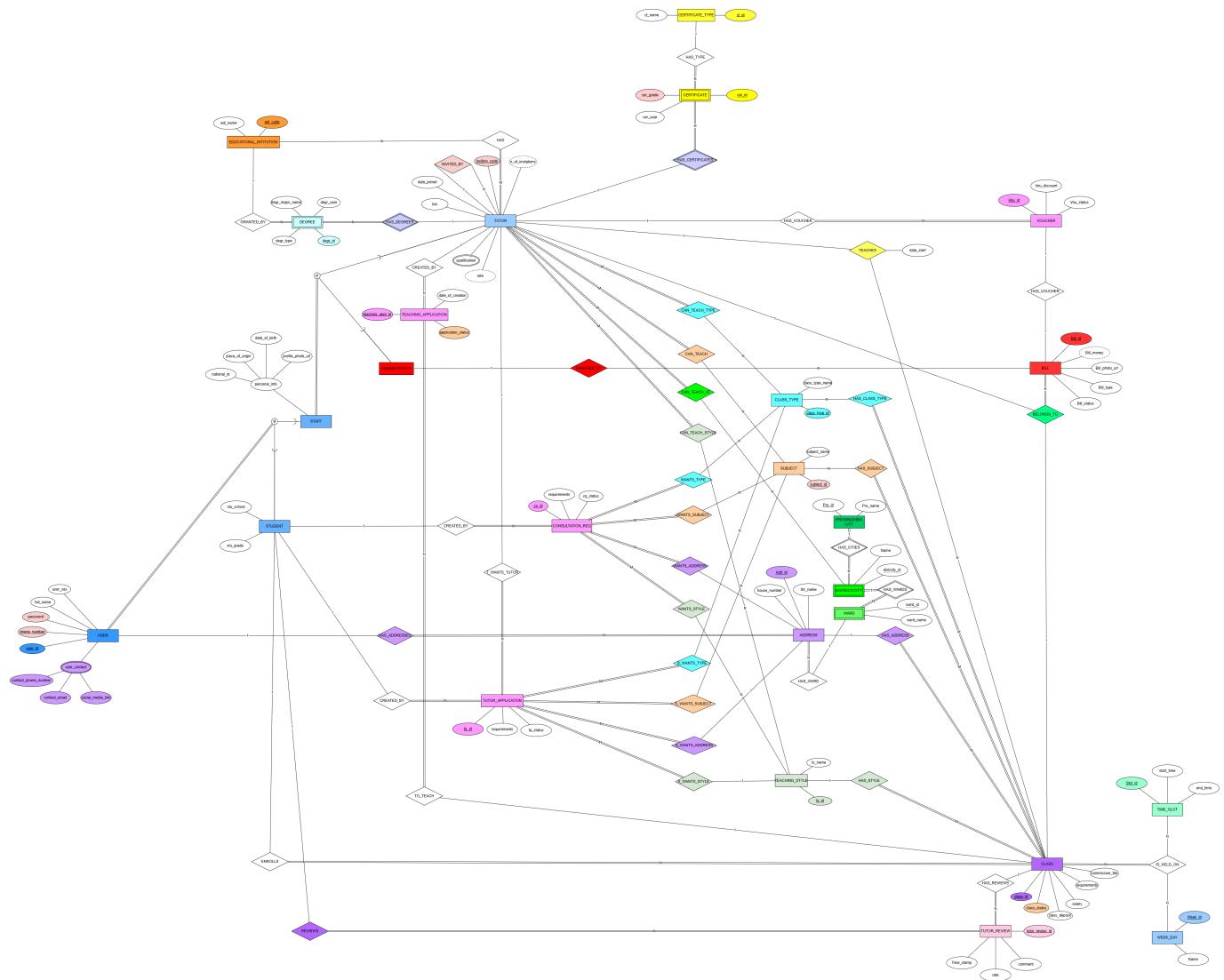


```
-- Thu tuc cap nhat trang thai don
CREATE PROCEDURE Update_Application_Status
    @Tutor_App_Status VARCHAR(255),
    @Tutor_App_Id BIGINT
AS
BEGIN
    BEGIN TRY
        DECLARE @state VARCHAR(255);
        -- Lay gia tri status
        SELECT @state = ta_status
        FROM [Ass2_CO2013].[dbo].[tutor_application]
        WHERE ta_id = @Tutor_App_Id;
        -- Neu trang thai thay doi thanh 'Da mo lop'
        IF (@state != 'Da mo lop' AND @Tutor_App_Status = 'Da mo lop')
        BEGIN
            DECLARE
                @requirement VARCHAR(255), @addr_id BIGINT, @student_id
                BIGINT, @ts_id BIGINT, @tutor_id BIGINT;
            -- Lay thong tin tu don dang ki tutor\_\application
            SELECT
                @requirement = requirement,
                @addr_id = addr_id,
                @student_id = student_id,
                @ts_id = ts_id,
                @tutor_id = tutor_id
            FROM [Ass2_CO2013].[dbo].[tutor_application]
            WHERE ta_id = @Tutor_App_Id;
            -- Tao lop moi
            EXECUTE [Ass2_CO2013].[dbo].Create_Class_From_TA
                @requirement, @addr_id, @student_id, @ts_id,
                @tutor_id, @Tutor_App_Id;
        END;
        IF (@state != @Tutor_App_Status )
        BEGIN
            UPDATE [Ass2_CO2013].[dbo].[tutor_application]
            SET ta_status = @Tutor_App_Status
            WHERE ta_id = @Tutor_App_Id;
            PRINT 'Updated successfully!';
        END;
        ELSE PRINT 'There is no change!';
    END TRY
    BEGIN CATCH
        PRINT 'Error when update: ' + ERROR_MESSAGE();
    END CATCH;
END;
```

4. Test case (thay đổi trạng thái và id đơn để xem thay đổi):

```
EXECUTE [Ass2_CO2013].[dbo].Update_Application_Status "Da mo lop", 3;
```

6 Phụ lục



Hình 101: Lược đồ ánh xạ cơ sở dữ liệu

Link báo cáo bài tập lớn 1: [Báo cáo BTL 1](#).

Chỉnh sửa phần ràng buộc dữ liệu

- Bỏ một số thuộc tính cũ không còn sử dụng trong hệ thống (đổi tên thuộc tính **user_sex** thành **user_sex**, bỏ thuộc tính ; bỏ phần mô tả thuộc tính **session_duration** và **session_amount**; Thuộc tính liên quan đến vị trí như **zip_code**)
- Bổ sung ràng buộc cho thuộc tính **rate** của **TUTOR REVIEW**



- Thêm thuộc tính `application_status` chỉ nhận giá trị là "chap nhan" hoặc "tu choi"; thuộc tính `Vou_status` nhận các giá trị "Da su dung" hoặc "Chua su dung", thuộc tính `teaching_style` chỉ nhận 2 giá trị là "Truc tiep" và "Truc tuyen".
- Thuộc tính `bill` có sự đổi tên cho các trạng thái:
 - Trạng thái "Da hoan" sửa lại thành "Xac nhan"
 - Trạng thái "Chua hoan" sửa thành "Chua xac nhan"
- Trạng thái thuộc tính `tutor_application` được đổi từ "da xu ly" thành "da mo lop".

Chỉnh sửa phần ràng buộc nghiệp vụ

- Chỉnh sửa lại phần ràng buộc của phần "Quản lý lịch học" và "Quản lý thanh toán" cho phù hợp với hệ thống.

Chỉnh sửa phần ràng buộc ngữ nghĩa

- Chỉnh sửa phần ràng buộc độ tuổi (gia sư dạy học phải trên 18 tuổi)
- Sửa lại ràng buộc thời gian về số giờ tối đa gia sư được dạy trong một tuần.
- Bổ sung thêm ràng buộc liên quan đến mã khuyến mãi Voucher khi đóng phí hoa hồng.
- Bổ sung ràng buộc ngữ nghĩa rằng địa chỉ của một lớp học phải là một trong những địa chỉ của học sinh trong lớp học đó (học sinh phải chọn từ các địa chỉ của mình)
- Bổ sung ràng buộc ngữ nghĩa rằng tiền đặt cọc không được lớp hơn mức lương đề xuất của gia sư
- Bổ sung ràng buộc ngữ nghĩa rằng các loại bằng cấp mà gia sư thêm vào hệ thống phải có thời gian (năm nhận bằng) nằm trong khoảng 2 năm tính từ thời điểm hiện tại, tức là gia sư không thể thêm cấp được cấp từ quá 2 năm trước vào hệ thống, những loại bằng cấp đó không có giá trị, ràng buộc ngữ nghĩa được hiện thực bằng cách sử dụng stored procedure (thay vì trigger) vì đơn giản và có performance tốt hơn



Trường Đại Học Bách Khoa Tp. Hồ Chí Minh

Khoa Khoa Học Vật Kỹ Thuật Máy Tính

ĐÁNH GIÁ CỦA GIÁO VIÊN

STT	Họ và tên	MSSV	Đóng góp	Nhận xét	ĐT(-)	ĐC(+)	Tổng
1	Đào Duy Quý	2212864	100%				
2	Lê Bùi Trung Dũng	2210573	100%				
3	Phạm Phương Duy	2210532	100%				
4	Trần Thị Ngọc Huyền	2211311	100%				
5	Trương Nguyễn Minh Nhiên	2212452	100%				