

TRƯỜNG ĐẠI HỌC TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
TRUNG TÂM PHÁT TRIỂN CÔNG NGHỆ THÔNG TIN

BÀI TẬP THỰC HÀNH 02

Xử Lý Thông Tin



Mã Môn: IE103
Tên Môn: Quản Lý Thông Tin
Lớp: LT.K2025.2.CNTT
Giảng Viên: Ths. Nguyễn Thành Luân
Thực Hiện: Nhóm 02

Tháng 12, 2025

MỤC LỤC

Trang Bìa (Bài Tập Thực Hành 02)	i
Mục Lục	ii
Danh Sách Bảng	ii
Danh Sách Hình Ảnh	iii
Danh Sách Mã Nguồn	iii
Danh Sách TODO	iii
Nhóm 02	iv
Đề Bài - Cơ Sở Dữ Liệu về Quản Lý Đề Tài	v
Môi Trường Triển Khai	v
Các Tài Nguyên	v
1. Phần 1	1
1.1. CREATE - Tạo Các Quan Hệ	1
1.2. INSERT - Thêm Dữ Liệu	4
2. Phần 2	8
2.1. A. STORED PROCEDURES VỚI THAM SỐ VÀO	8
2.2. B. STORED PROCEDURES VỚI THAM SỐ VÀO VÀ RA	11
2.3. C. TRIGGER	20
2.4. D. FUNCTION	31
2.5. E. CURSOR	35
3. Phụ Lục: Thiết Lập Môi Trường Lab	41
3.1. Thiết bị	41
3.2. Các công cụ	41
3.3. Quản lý vòng đời của SQL Server	41
3.4. Khởi tạo môi trường lab thủ công	41
3.5. Khởi tạo môi trường lab với ADS (SQL Projects)	41

DANH SÁCH BẢNG

Nhóm 02	iv
---------------	----

DANH SÁCH HÌNH ẢNH

DANH SÁCH MÃ NGUỒN

DANH SÁCH TODO

TODO 1 Hoàn Thành Notebook `IE103-BTH2.ipynb` cho lab này. v

NHÓM 02

Thành viên của nhóm và bảng nhiệm vụ cho Bài Tập Thực Hành 02.

No	MSSV	Họ và Tên	Phụ Trách
1	25410291	Đinh Xuân Sâm	<ul style="list-style-type: none">Tổng hợp & Biên tập.
2	25410319	Đặng Hữu Toàn	<ul style="list-style-type: none">Phần 2.C (Mục 2.3)
3	25410321	Nguyễn Điện Triết	<ul style="list-style-type: none">Phần 1.* (Mục 1.xx)Phần 2.A (Mục 2.1)
4	25410204	Trương Xuân Hậu	<ul style="list-style-type: none">Phần 2.E (Mục 2.5)
5	25410338	Lê Anh Vũ	
6	25410176	Trần Sơn Bình	
7	25410247	Lê Kim Long	<ul style="list-style-type: none">Phần 2.B (Mục 2.2)
8	25410337	La Anh Vũ	
9	25410209	Lê Ngọc Hiệp	<ul style="list-style-type: none">Phần 2.D (Mục 2.4)
10	25410271	Nguyễn Thị Ngọc Nhung	

Nhóm 02

ĐỀ BÀI - CƠ SỞ DỮ LIỆU VỀ QUẢN LÝ ĐỀ TÀI

Cho lược đồ quan hệ sau, hãy hoàn thành các yêu cầu ở Mục 1 (Phần 1) và Mục 2 (Phần 2).

1. SINHVIEN(MSSV, TENSV, SODT, LOP, DIACHI)
2. DETAI(MSDT, TENDT)
3. SV_DETAI(MSSV, MSDT)
4. GIAOVIEN(MSGV, TENGV, DIACHI, SODT, MSHH, NAMHH)
5. HOCVI(MSHV, TENHV)
6. CHUYENNGANH(MSCN, TENCN)
7. GV_HV_CN(MSGV, MSHV, MSCN, NAM)
8. HOCHAM(MSHH, TENHH)
9. GV_HDDT(MSGV, MSDT, DIEM)
10. GV_PBDT(MSGV, MSDT, DIEM)
11. GV_UVDT(MSGV, MSDT, DIEM)
12. HOIDONG(MSHD, PHONG, TGBD, NGAYHD, TINHTRANG, MSGV)
13. HOIDONG_GV(MSHD, MSGV)
14. HOIDONG_DT(MSHD, MSDT, QUYETDINH)

MÔI TRƯỜNG TRIỂN KHAI

- RDBMS: Microsoft SQL Server 2019 (Azure SQL Edge).
- Editor/IDE: Azure Database Studio (ADS).
- Extensions: SQL Database Projects (`microsoft.sql-database-projects`).

CÁC TÀI NGUYÊN

Các tài nguyên (tài liệu, code, vv...) của lab này được lưu tại:

- [LT.K2025.2-CNTT > IE103 > assignments > BTTH2](#)
 - ▶ `content`: Thư mục chứa các file nội dung của báo cáo.
 - ▶ `content/code`: Thư mục chứa các script `.sql` để thực thi.
 - ▶ `IE103-BTTH2`: Thư mục chứa file dự án `.sqlproj` cho SQL Projects.
 - ▶ `IE103-BTTH2.ipynb`: Notebook nội dung của lab này.

TODO: Hoàn Thành Notebook [`IE103-BTTH2.ipynb`](#) cho lab này.

PHẦN 1

1.1. CREATE - Tạo Các Quan Hệ

1.1.1. SINHVIEN

```
1 CREATE TABLE SINHVIEN (
2     MSSV CHAR(8) PRIMARY KEY,
3     TENV NVARCHAR(30) NOT NULL,
4     SODT VARCHAR(10) NOT NULL,
5     LOP CHAR(10) NOT NULL,
6     DIACHI NCHAR(50)
7 );
```

1.1.2. DETAI

```
1 CREATE TABLE DETAI (
2     MSDT CHAR(6) PRIMARY KEY,
3     TENDT NVARCHAR(30) NOT NULL
4 );
```

1.1.3. SV_DETAI

```
1 CREATE TABLE SV_DETAI (
2     MSSV CHAR(8),
3     MSDT CHAR(6),
4     PRIMARY KEY (MSSV, MSDT),
5     CONSTRAINT FK_SV_DETAI_SV FOREIGN KEY (MSSV) REFERENCES SINHVIEN(MSSV),
6     CONSTRAINT FK_SV_DETAI_DT FOREIGN KEY (MSDT) REFERENCES DETAI(MSDT)
7 );
```

1.1.4. HOCHAM

```
1 CREATE TABLE HOCHAM (
2     MSHH INT PRIMARY KEY,
3     TENHH NVARCHAR(20) NOT NULL
4 );
```

1.1.5. GIAOVIEN

```
1 CREATE TABLE GIAOVIEN (
2     MSGV INT PRIMARY KEY,
3     TENV NVARCHAR(30) NOT NULL,
4     DIACHI NVARCHAR(50) NOT NULL,
5     SODT VARCHAR(10) NOT NULL,
6     MSHH INT NOT NULL,
7     NAMHH SMALLDATETIME NOT NULL,
8     CONSTRAINT FK_GV_HH FOREIGN KEY (MSHH) REFERENCES HOCHAM(MSHH)
9 );
```

1.1.6. HOCVI

```
1 CREATE TABLE HOCVI (
2     MSHV INT PRIMARY KEY,
3     TENHV NVARCHAR(20) NOT NULL
4 );
```

1.1.7. CHUYENNGANH

```
1 CREATE TABLE CHUYENNGANH (
2     MSCN INT PRIMARY KEY,
3     TENCN NVARCHAR(30) NOT NULL
4 );
```

1.1.8. GV_HV_CN

```
1 CREATE TABLE GV_HV_CN (
2     MSGV INT,
3     MSHV INT,
4     MSCN INT,
5     NAM SMALLDATETIME NOT NULL,
6     PRIMARY KEY (MSGV, MSHV, MSCN),
7     CONSTRAINT FK_GV_HV_CN_GV FOREIGN KEY (MSGV) REFERENCES GIAOVIEN(MSGV),
8     CONSTRAINT FK_GV_HV_CN_HV FOREIGN KEY (MSHV) REFERENCES HOCVI(MSHV),
9     CONSTRAINT FK_GV_HV_CN_CN FOREIGN KEY (MSCN) REFERENCES CHUYENNGANH(MSCN)
10 );
```

1.1.9. GV_HDDT

```
1 CREATE TABLE GV_HDDT (
2     MSGV INT,
3     MSDT CHAR(6),
4     DIEM FLOAT NOT NULL,
5     PRIMARY KEY (MSGV, MSDT),
6     CONSTRAINT FK_GV_HDDT_GV FOREIGN KEY (MSGV) REFERENCES GIAOVIEN(MSGV),
7     CONSTRAINT FK_GV_HDDT_DT FOREIGN KEY (MSDT) REFERENCES DETAI(MSDT)
8 );
```

1.1.10. GV_PBDT

```
1 CREATE TABLE GV_PBDT (
2     MSGV INT,
3     MSDT CHAR(6),
4     DIEM FLOAT NOT NULL,
5     PRIMARY KEY (MSGV, MSDT),
6     CONSTRAINT FK_GV_PBDT_GV FOREIGN KEY (MSGV) REFERENCES GIAOVIEN(MSGV),
7     CONSTRAINT FK_GV_PBDT_DT FOREIGN KEY (MSDT) REFERENCES DETAI(MSDT)
8 );
```

1.1.11. GV_UVDT

```
1 CREATE TABLE GV_UVDT (
2     MSGV INT,
3     MSDT CHAR(6),
4     DIEM FLOAT NOT NULL,
5     PRIMARY KEY (MSGV, MSDT),
6     CONSTRAINT FK_GV_UVDT_GV FOREIGN KEY (MSGV) REFERENCES GIAOVIEN(MSGV),
7     CONSTRAINT FK_GV_UVDT_DT FOREIGN KEY (MSDT) REFERENCES DETAI(MSDT)
8 );
```

1.1.12. HOIDONG

```
1 CREATE TABLE HOIDONG (
2     MSHD INT PRIMARY KEY,
3     PHONG INT NOT NULL,
4     TGBD SMALLDATETIME NOT NULL,
5     NGAYHD SMALLDATETIME NOT NULL,
6     TINHTRANG NVARCHAR(30),
7     MSGV INT,
8     CONSTRAINT FK_HD_GV FOREIGN KEY (MSGV) REFERENCES GIAOVIEN(MSGV)
9 );
```

1.1.13. HOIDONG_GV

```
1 CREATE TABLE HOIDONG_GV (
2     MSHD INT,
3     MSGV INT,
4     PRIMARY KEY (MSHD, MSGV),
5     CONSTRAINT FK_HD_GV_HD FOREIGN KEY (MSHD) REFERENCES HOIDONG(MSHD),
6     CONSTRAINT FK_HD_GV_GV FOREIGN KEY (MSGV) REFERENCES GIAOVIEN(MSGV)
7 );
```

1.1.14. HOIDONG_DT

```
1 CREATE TABLE HOIDONG_DT (
2     MSHD INT,
3     MSDT CHAR(6),
4     QUYETDINH NCHAR(10),
5     CONSTRAINT PK_HD_DT PRIMARY KEY (MSHD, MSDT),
6     CONSTRAINT FK_HD_DT_HD FOREIGN KEY (MSHD) REFERENCES HOIDONG(MSHD),
7     CONSTRAINT FK_HD_DT_DT FOREIGN KEY (MSDT) REFERENCES DETAI(MSDT)
8 );
```

1.2. INSERT - Thêm Dữ Liệu

1.2.1. Table SINHVIEN

```
1 INSERT INTO SINHVIEN (MSSV, TENSV, SODT, LOP, DIACHI) VALUES
2 ('13520001', N'Nguyễn Văn An', '0986762255', 'SE103.U32', N'THỦ ĐỨC'),
3 ('13520002', N'Phan Tân Đạt', '0975672350', 'IE204.T21', N'QUẬN 1'),
4 ('13520003', N'Nguyễn Anh Hải', '0947578688', 'IE205.R12', N'QUẬN 9'),
5 ('13520004', N'Phạm Tài', '0956757869', 'IE202.A22', N'QUẬN 1'),
6 ('13520005', N'Lê Thúy Hằng', '0976668688', 'SE304.E22', N'THỦ ĐỨC'),
7 ('13520006', N'Üng Hồng Ân', '0957475898', 'IE208.F33', N'QUẬN 2');
```

1.2.2. Table DETAI

```
1 INSERT INTO DETAI (MSDT, TENDT) VALUES
2 ('97001', N'Quản lý thư viện'),
3 ('97002', N'Nhận dạng vân tay'),
4 ('97003', N'Bán đấu giá trên mạng'),
5 ('97004', N'Quản lý siêu thị'),
6 ('97005', N'Xử lý ảnh'),
7 ('97006', N'Hệ giải toán thông minh');
```

1.2.3. Table SV_DETAI

```
1 INSERT INTO SV_DETAI (MSSV, MSDT) VALUES
2 ('13520001', '97004'),
3 ('13520002', '97005'),
4 ('13520003', '97001'),
5 ('13520004', '97002'),
6 ('13520005', '97003'),
7 ('13520006', '97005');
```

1.2.4. Table HOCHAM

```
1 INSERT INTO HOCHAM (MSHH, TENHH) VALUES
2 (1, N'PHÓ GIÁO SƯ'),
3 (2, N'GIÁO SƯ');
```

1.2.5. Table GIAOVIEN

```
1 INSERT INTO GIAOVIEN (MSGV, TENGV, DIACHI, SODT, MSHH, NAMHH) VALUES
2 (201, N'Trần Trung', N'Bến Tre', '35353535', 1, '1996'),
3 (202, N'Nguyễn Văn An', N'Tiền Giang', '67868688', 1, '1996'),
4 (203, N'Trần Thu Trang', N'Cần Thơ', '74758687', 1, '1996'),
5 (204, N'Nguyễn Thị Loan', N'TP. HCM', '56575868', 2, '2005'),
6 (205, N'Chu Tiên', N'Hà Nội', '46466646', 2, '2005');
```

1.2.6. Table HOCVI

```
1 INSERT INTO HOCVI (MSHV, TENHV) VALUES
2 (1, N'Kỹ sư'),
3 (2, N'Cử nhân'),
4 (3, N'Thạc sĩ'),
5 (4, N'Tiến sĩ'),
6 (5, N'Tiến sĩ Khoa học');
```

1.2.7. Table CHUYENNGANH

```
1 INSERT INTO CHUYENNGANH (MSCN, TENCN) VALUES
2 (1, N'Công nghệ Web'),
3 (2, N'Mạng xã hội'),
4 (3, N'Quản lý CNTT'),
5 (4, N'GIS');
```

1.2.8. Table GV_HV_CN

```
1 INSERT INTO GV_HV_CN (MSGV, MSHV, MSCN, NAM) VALUES
2 (201, 1, 1, '2013'),
3 (201, 1, 2, '2013'),
4 (201, 2, 1, '2014'),
5 (202, 3, 2, '2013'),
6 (203, 2, 4, '2014'),
7 (204, 3, 2, '2014');
```

1.2.9. Table GV_HDDT

```
1 INSERT INTO GV_HDDT (MSGV, MSDT, DIEM) VALUES
2 (201, '97001', 8),
3 (202, '97002', 7),
4 (205, '97001', 9),
5 (204, '97004', 7),
6 (203, '97005', 9);
```

1.2.10. Table GV_PBDT

```
1 INSERT INTO GV_PBDT (MSGV, MSDT, DIEM) VALUES
2 (201, '97005', 8),
3 (202, '97001', 7),
4 (205, '97004', 9),
5 (204, '97003', 7),
6 (203, '97002', 9);
```

1.2.11. Table GV_UVDT

```
1 INSERT INTO GV_UVDT (MSGV, MSDT, DIEM) VALUES
2 (205, '97005', 8),
3 (202, '97005', 7),
4 (204, '97005', 9),
5 (203, '97001', 7),
6 (204, '97001', 9),
7 (205, '97001', 8),
8 (203, '97003', 7),
9 (201, '97003', 9),
10 (202, '97003', 7),
11 (201, '97004', 9),
12 (202, '97004', 8),
13 (203, '97004', 7),
14 (201, '97002', 9),
15 (204, '97002', 7),
16 (205, '97002', 9),
17 (201, '97006', 9),
18 (202, '97006', 7),
19 (204, '97006', 9);
```

1.2.12. Table HOIDONG

```
1 INSERT INTO HOIDONG (MSHD, PHONG, TGBD, NGAYHD, TINHTRANG, MSGV) VALUES
2 (1, 2, '07:00', '2014-11-29', N'Thật', 201),
3 (2, 102, '07:00', '2014-12-05', N'Thật', 202),
4 (3, 3, '08:00', '2014-12-06', N'Thật', 203);
```

1.2.13. Table HOIDONG_GV

```
1 INSERT INTO HOIDONG_GV (MSHD, MSGV) VALUES
2 (1, 201), (1, 202), (1, 203), (1, 204),
3 (2, 203), (2, 202), (2, 205), (2, 204),
4 (3, 201), (3, 202), (3, 203), (3, 204);
```

1.2.14. Table HOIDONG_DT

```
1 INSERT INTO HOIDONG_DT (MSHD, MSDT, QUYETDINH) VALUES
2 (1, '97001', N'Dược'),
3 (1, '97002', N'Dược'),
4 (2, '97001', N'Không'),
5 (2, '97004', N'Không'),
6 (1, '97005', N'Dược'),
7 (3, '97001', N'Không'),
8 (3, '97002', N'Dược');
```

PHẦN 2

2.1. A. STORED PROCEDURES VỚI THAM SỐ VÀO

2.1.1. Kiểm Tra MSHH Trong Table HOCHAM

- Tham số vào là MSGV , TENGV , SODT , DIACHI , MSHH , NAMHH .
- Trước khi insert dữ liệu cần kiểm tra MSHH đã tồn tại trong table HOCHAM chưa
 - Nếu chưa thì trả về giá trị 0 .

```
1 CREATE PROC SP_INS_GV_KT_MSHH
2     @MSGV INT,
3     @TENGV NVARCHAR(30),
4     @SODT VARCHAR(10),
5     @DIACHI NVARCHAR(50),
6     @MSHH INT,
7     @NAMHH SMALLDATETIME
8 AS
9 BEGIN
10    IF NOT EXISTS (SELECT MSHH FROM HOCHAM WHERE MSHH = @MSHH)
11        RETURN 0
12    INSERT INTO GIAOVIENT VALUES (@MSGV, @TENGV, @DIACHI, @SODT, @MSHH, @NAMHH)
13    RETURN 1
14 END;
```

2.1.2. Kiểm Tra MSGV Trong Table GIAOVIENT

- Tham số vào là MSGV , TENGV , SODT , DIACHI , MSHH , NAMHH .
- Trước khi insert dữ liệu cần kiểm tra MSGV trong table GIAOVIENT có trùng không
 - Nếu trùng thì trả về giá trị 0 .

```

1 CREATE PROC SP_INS_GV_KT_MSGV
2     @MSGV INT,
3     @TENGV NVARCHAR(30),
4     @SODT VARCHAR(10),
5     @DIACHI NVARCHAR(50),
6     @MSHH INT,
7     @NAMHH SMALLDATETIME
8 AS
9 BEGIN
10    IF EXISTS (
11        SELECT 1
12        FROM GIAOVIENT
13        WHERE MSGV = @MSGV
14    )
15    BEGIN
16        RETURN 0; -- MSGV đã tồn tại
17    END
18    INSERT INTO GIAOVIENT VALUES (@MSGV, @TENGV, @DIACHI, @SODT, @MSHH, @NAMHH)
19    RETURN 1
20 END;

```

2.1.3. Kiểm Tra MSGV và MSHH

- Giống Mục 2.1.1 và Mục 2.1.2, kiểm tra xem `MSGV` có trùng không? `MSHH` có tồn tại chưa?
 - ▶ Nếu `MSGV` trùng thì trả về `0`.
 - ▶ Nếu `MSHH` chưa tồn tại trả về `1`.
 - ▶ Ngược lại cho `INSERT` dữ liệu

```

1 CREATE PROC SP_INS_GV_KT_MSHH_MSGV
2     @MSGV INT,
3     @TENGV NVARCHAR(30),
4     @SODT VARCHAR(10),
5     @DIACHI NVARCHAR(50),
6     @MSHH INT,
7     @NAMHH SMALLDATETIME
8 AS
9 BEGIN
10    IF NOT EXISTS (SELECT MSHH FROM HOCHAM WHERE MSHH = @MSHH)
11    BEGIN
12        RETURN 1; -- Nếu MSHH chưa tồn tại trả về 1
13    END
14    IF EXISTS (SELECT 1 FROM GIAOVIENT WHERE MSGV = @MSGV)
15    BEGIN
16        RETURN 0; -- Nếu MSGV trùng thì trả về 0
17    END
18    INSERT INTO GIAOVIENT VALUES (@MSGV, @TENGV, @DIACHI, @SODT, @MSHH, @NAMHH)
19 END;

```

2.1.4. Thao Tác với MSDT và TENDT

- Đưa vào MSDT cũ, TENDT mới.
- Hãy cập nhật tên đề tài mới với mã đề tài cũ không đổi.
 - Nếu không tìm thấy, trả về 0.
 - Ngược lại cập nhật và trả về 1.

```
1 CREATE PROC SP_UPD_DETAI_KT_MSDT
2     @MSDT char(6),
3     @TENDT NVARCHAR(30)
4 AS
5 BEGIN
6     IF NOT EXISTS (SELECT MSDT FROM DETAI WHERE MSDT = @MSDT)
7     BEGIN
8         RETURN 0; -- Nếu MSHH chưa tồn tại trả về 0
9     END
10
11    UPDATE DETAI
12    SET TENDT = @TENDT
13    WHERE MSDT = @MSDT
14    RETURN 1
15 END;
```

2.1.5. Thao Tác với MSSV, TENSV, DIACHI

- Tham số đưa vào MSSV, TENSV mới, DIACHI mới.
- Hãy cập nhật sinh viên trên với MSSV không đổi.
 - Nếu không tìm thấy, trả về 0.
 - Ngược lại cập nhật và trả về 1.

```
1 CREATE PROC SP_UPD_SINHVIEN_KT_MSSV
2     @MSSV char(8),
3     @TENSV NVARCHAR(30),
4     @DIACHI nvarchar(50)
5 AS
6 BEGIN
7     IF NOT EXISTS (SELECT MSSV FROM SINHVIEN WHERE MSSV = @MSSV)
8     BEGIN
9         RETURN 0; -- Nếu không tìm thấy trả về 0,
10    END
11
12    UPDATE SINHVIEN
13    SET TENSV = @TENSV, DIACHI = @DIACHI
14    WHERE MSSV = @MSSV
15    RETURN 1
16 END;
```

2.2. B. STORED PROCEDURES VỚI THAM SỐ VÀO VÀ RA

2.2.1. Đưa vào `TENHV` trả ra GV

Trả ra:

- Số GV thỏa học vị
- Nếu không tìm thấy trả về 0

```
1 -- Tạo Stored Procedure
2 Create procedure sp_CountGVTheoHocVi
3     @TenHV NVARCHAR(20),
4     @SOGV INT OUTPUT
5 AS
6 BEGIN
7     SET @SOGV = 0;
8
9     DECLARE @MSHV_HOCVI INT;
10    SELECT @MSHV_HOCVI = MSHV
11    FROM HOCVI
12    WHERE TENHV = @TenHV;
13
14    IF @MSHV_HOCVI IS NOT NULL
15        BEGIN
16            SELECT @SOGV = COUNT(DISTINCT MSGV)
17            FROM GV_HV_CN
18            WHERE MSHV = @MSHV_HOCVI;
19        end;
20    end;
21 GO
```

2.2.1.1. Ví dụ 1: Học Vị Thạc Sĩ

- Học Vị = Thạc Sĩ

```
1 DECLARE @KETQUA INT;
2 EXEC sp_CountGVTheoHocVi N'Thạc sĩ', @KETQUA output ;
3 SELECT N'Số GV Thạc sĩ: ' AS KetQua, @KETQUA AS SoLuong;
```

KetQua	SoLuong
Số GV Thạc sĩ:	2

2.2.1.2. Ví dụ 2: Học Vị Tiến Sĩ

- Học vị = Tiến Sĩ (Không tồn tại)

```

1 DECLARE @KETQUA INT;
2 EXEC sp_CountGVTheoHocVi N'Tiến sĩ', @KETQUA output ;
3 SELECT N'Số GV Tiến sĩ: ' AS KetQua, @KETQUA AS SoLuong;

```

KetQua	SoLuong
Số GV Tiến sĩ:	0

2.2.2. Đưa vào MSDT trả về Điểm Trung Bình của Đề Tài

Đưa vào:

- MSDT

Cho biết:

- Điểm trung bình của đề tài.
 - Nếu không tìm thấy trả về 0.

```

1 CREATE PROCEDURE SP_DiemTBDeTai
2     @MSDT CHAR(6),
3     @DIEMTRUNG BINH FLOAT OUTPUT
4
5 AS
6 BEGIN
7     SET @DIEMTRUNG BINH = 0;
8
9     SELECT @DIEMTRUNG BINH = AVG(DIEM)
10    FROM (
11        SELECT DIEM FROM GV_HDDT WHERE MSDT = @MSDT
12        UNION ALL
13        SELECT DIEM FROM GV_PBDT WHERE MSDT = @MSDT
14        UNION ALL
15        SELECT DIEM FROM GV_UVDT WHERE MSDT = @MSDT
16    ) AS ALLDIEM;
17
18    IF @DIEMTRUNG BINH IS NULL
19        BEGIN
20            SET @DIEMTRUNG BINH = 0;
21        END
22    END
23 GO

```

2.2.2.1. Ví dụ 1: Đề tài 97001

```

1 DECLARE @KETQUA FLOAT;
2 EXEC SP_DiemTBDeTai '97001', @KETQUA output ;
3 SELECT 'Điểm TB Đề Tài 97001 ' AS KETQUA, @KETQUA AS DiemTB;

```

KETQUA

DiemTB

Diem TB De Tai 97001

8

2.2.2.2. Ví dụ 2: 97007

- Đề tài không có điểm: 97007

```

1 DECLARE @KETQUA FLOAT;
2 EXEC sp_DiemTBDeTai '97007', @KETQUA OUTPUT;
3 SELECT 'Diem TB De Tai 97007' AS KetQua, @KETQUA AS DiemTB;
```

KETQUA

DiemTB

Diem TB De Tai 97007

0

2.2.3. Đưa vào TENGV trả ra SDT của GV

Đưa vào TENGV trả ra:

- SDT của giáo viên đó.
- Nếu không tìm thấy trả về 0.
- Nếu trùng tên thì có báo lỗi không?
 - Tại sao?
- Làm sao để hiện thông báo có bao nhiêu giáo viên trùng tên và trả về các SDT?

Giải đáp:

- Nếu trùng tên thì sẽ không báo lỗi.
 - Nguyên nhân là nếu trùng tên thì đồng nghĩa với việc đang có nhiều records (bản ghi) có column là TENGV đang giống nhau.
 - Về mặt câu lệnh SQL, có thể như sau: sql SELECT * FROM GIAOVIENT WHERE TENGV = %NAME .
 - Kết quả trả về của câu lệnh SQL này có khả năng sẽ là 0, 1 hoặc nhiều records.
 - Nếu đề cập tới việc xảy ra lỗi ở đây thì có khả năng liên quan tới việc dữ liệu có khả năng bị sai lệch do sẽ không biết được sẽ lấy số điện thoại của record nào để trả về.
- Đáp ứng yêu cầu hiện thông báo có bao nhiêu giáo viên trùng tên và trả về các SDT, ta có thể làm như sau:

```

1 CREATE PROCEDURE sp_TimGVTheoTen @TENGV NVARCHAR(30),
2                                     @SOLUONGGV INT OUTPUT,
3                                     @DS_SDT VARCHAR(100) OUTPUT
4 AS
5 BEGIN
6     SET @DS_SDT = '';
7     -- DEM SO LUONG GIAO VIEN TRUNG TEN
8     SELECT @SOLUONGGV = COUNT(MSGV)
9     FROM GIAOVIENT
10    WHERE TENGV = @TENGV;
11 end
12 -- KIEM TRA SO LUONG GIAO VIEN VA TRA VE THONG TIN TUONG UNG
13 IF @SOLUONGGV > 0
14     BEGIN
15         SELECT @DS_SDT = STRING_AGG(SODT, ', ')
16         FROM GIAOVIENT
17         WHERE TENGV = @TENGV;
18     end
19 ELSE
20     BEGIN
21         SET @SOLUONGGV = 0;
22         SET @DS_SDT = '';
23     end;
24 GO

```

Dữ liệu bảng `GIAOVIENT` (Đã thêm một record Nguyễn Văn An, `MSGV`: 206, `SODT`: 12434343):

```

1 INSERT INTO GIAOVIENT (MSGV, TENGV, DIACHI, SODT, MSHH, NAMHH)
2 VALUES (206, 'Nguyễn Văn An', 'Cần Thơ', '12434343', 1, '2025');

```

Giá trị của bảng (không hiển thị cột `NAMHH` để tiết kiệm không gian):

MSGV	TENGV	DIACHI	SODT	MSHH
201	Trần Trung	Bến Tre	35353535	1
202	Nguyễn Văn An	Tiền Giang	67868688	1
203	Trần Thu Trang	Cần Thơ	74758687	1
204	Nguyễn Thị Loan	TP. HCM	56575868	2
205	Chu Tiến	Hà Nội	46466646	2
206	Nguyễn Văn An	Cần Thơ	12434343	1

2.2.3.1. Ví dụ 1: Có trùng tên

- Trường hợp trùng tên: Nguyễn Văn An

```

1 DECLARE @SoLuong INT, @DS_SDT NVARCHAR(100);
2 EXEC sp_TimGVTheoTen N'Nguyễn Văn An', @SoLuong OUTPUT, @DS_SDT OUTPUT;
3 SELECT N'Tong so giao vien trung ten: ' AS ThongBao, @SoLuong AS SoLuong, @DS_SDT AS 'Danh
sach so dien thoai';

```

- Kết quả: trả về 2 bản ghi có sự trùng lặp ở trường `TENGV`

ThongBao	SoLuong	Danh sach so dien thoai
Tong so giao vien trung ten:	2	67868688, 12434343

2.2.3.2. Ví dụ 2: Không trùng tên

- Trường hợp không trùng tên: Trần Trung

```

1 DECLARE @SoLuong INT, @DS_SDT NVARCHAR(100);
2 EXEC sp_TimGVTheoTen N'Trần Trung', @SoLuong OUTPUT, @DS_SDT OUTPUT;
3 SELECT N'Tong so giao vien trung ten: ' AS ThongBao, @SoLuong AS SoLuong, @DS_SDT AS 'Danh
sach so dien thoai';

```

- Kết quả: trả về duy nhất 1 bản ghi do không tìm thấy sự trùng lặp ở trường `TENGV`

ThongBao	SoLuong	Danh sach so dien thoai
Tong so giao vien trung ten:	1	35353535

2.2.3.3. Ví dụ 3: Tên không tồn tại

- Trường hợp không tìm thấy: Lê Kim Long

```

1 DECLARE @SoLuong INT, @DS_SDT NVARCHAR(100);
2 EXEC sp_TimGVTheoTen N'Lê Kim Long', @SoLuong OUTPUT, @DS_SDT OUTPUT;
3 SELECT N'Tong so giao vien trung ten: ' AS ThongBao, @SoLuong AS SoLuong, @DS_SDT AS 'Danh
sach so dien thoai';

```

- Kết quả: `0`, vì không tìm thấy `TENGV` phù hợp.

ThongBao	SoLuong	Danh sach so dien thoai
Tong so giao vien trung ten:	0	

2.2.4. Đưa vào MSHD cho biết Điểm Trung Bình

Cho biết:

- Điểm trung bình các đè tài của hội đồng đó.

```

1 CREATE PROCEDURE sp_DiemTBTheoHD @MSHD INT,
2                               @DIEM_TB FLOAT OUTPUT
3 AS
4 BEGIN
5     SET @DIEM_TB = 0;
6     SELECT @DIEM_TB = AVG(T2.DIEM) FROM HOIDONG_DT T1
7     JOIN (
8         SELECT MSDT, DIEM FROM GV_HDDT
9         UNION ALL
10        SELECT MSDT, DIEM FROM GV_PBDT
11        UNION ALL
12        SELECT MSDT, DIEM FROM GV_UVDT
13    ) AS T2 ON T1.MSDT = T2.MSDT
14    WHERE T1.MSHD = @MSHD
15    GROUP BY T1.MSDT;
16 end;
17 GO

```

2.2.4.1. Ví dụ: Hội đồng 1

```

1 DECLARE @MSHD INT = 1, @DiemTB FLOAT;
2 EXEC sp_DiemTBTheoHD @MSHD, @DiemTB output;
3 SELECT @MSHD as 'Ma So Hoi Dong', @DiemTB as 'Diem Trung Binh'

```

Ma So Hoi Dong	Diem Trung Binh
1	8.2

2.2.4.2. Ví dụ: Hội đồng 2

```

1 DECLARE @MSHD INT = 2, @DiemTB FLOAT;
2 EXEC sp_DiemTBTheoHD @MSHD, @DiemTB output;
3 SELECT @MSHD as 'Ma So Hoi Dong', @DiemTB as 'Diem Trung Binh'

```

Ma So Hoi Dong	Diem Trung Binh
2	8

2.2.5. Đưa vào TENGV

Đưa vào TENGV cho biết:

- Số đè tài hướng dẫn, số đè tài phản biện do giáo viên đó phụ trách.
- Nếu trùng tên thì có báo lỗi không hay hệ thống sẽ đếm tất cả các đè tài của những giáo viên trùng tên đó?

Giải đáp:

- Tương tự câu 3 ở trên, hệ thống sẽ không báo lỗi cú pháp SQL. Thay vào đó, hệ thống sẽ lấy được tất cả các records có cùng tên giáo viên đó.
- Xử lý Stored Procedure để trả về số lượng đề tài hướng dẫn và phản biện dựa theo tên giáo viên, ta có thể xử lý như sau (sử dụng THROW để thông báo lỗi):

```

1 CREATE PROCEDURE sp_ThongTinDeTaiGV @TENGV NVARCHAR(30),
2                                     @SoDT_HD INT OUTPUT,
3                                     @SoDT_PB INT OUTPUT
4 AS
5 BEGIN
6     SET @SoDT_HD = 0;
7     SET @SoDT_PB = 0;
8
9     DECLARE @SoLuong_GV_TrungTen INT;
10    SET @SoLuong_GV_TrungTen = 0;
11
12    -- Kiểm tra danh sách giáo viên xem có trùng tên không
13    SELECT @SoLuong_GV_TrungTen = Count(MSGV) FROM GIAOVIEN WHERE TENGV = @TENGV
14
15    -- Nếu là 1: không trùng tên.
16    -- Nếu là 2 trở lên: trùng tên → Lỗi
17    -- Nếu là 0: không tồn tại giáo viên có tên này
18    IF @SoLuong_GV_TrungTen = 1
19        BEGIN
20            -- Lấy MSGV từ khi chỉ có 1 record duy nhất
21            DECLARE @MSGV INT;
22            SELECT @MSGV = MSGV FROM GIAOVIEN WHERE TENGV = @TENGV
23
24            -- Đếm số lượng đề tài hướng dẫn và phản biện
25            SELECT @SoDT_HD = Count(MSDT) FROM GV_HDDT WHERE MSGV = @MSGV
26            SELECT @SoDT_PB = COUNT(MSDT) FROM GV_PBDT WHERE MSGV = @MSGV
27            end
28        ELSE IF @SoLuong_GV_TrungTen ≥ 2
29            BEGIN
30                THROW 51000, N'Trùng tên giáo viên', 2;
31            end
32        ELSE
33            BEGIN
34                THROW 51000, N'Không tìm thấy giáo viên', 1;
35            end
36        end
37    GO

```

2.2.5.1. Dữ liệu đang có

- GIAOVIEN

```

1  SELECT TOP (100) [MSGV]
2      ,[TENGV]
3      ,[DIACHI]
4      ,[SODT]
5      ,[MSHH]
6      --  ,[NAMHH]
7  FROM [IE103-BTTB2].[dbo].[GIAOVIEN]

```

MSGV	TENGV	DIACHI	SODT	MSHH
201	Trần Trung	Bến Tre	35353535	1
202	Nguyễn Văn An	Tiền Giang	67868688	1
203	Trần Thu Trang	Cần Thơ	74758687	1
204	Nguyễn Thị Loan	TP. HCM	56575868	2
205	Chu Tiên	Hà Nội	46466646	2
206	Nguyễn Văn An	Cần Thơ	12434343	1

- GV_HDDT

MSGV	MSDT	DIEM
201	97001	8
202	97002	7
203	97005	9
204	97004	7
205	97001	9

- GV_PBDT

MSGV	MSDT	DIEM
201	97005	8
202	97001	7
203	97002	9
204	97003	7
205	97004	9

2.2.5.2. Ví dụ 1: không tìm thấy giáo viên

- TENGV : Lê Kim Long

```

1 DECLARE @SoDT_HD INT, @SoDT_PB INT
2 EXEC sp_ThongTinDeTaiGV N'Lê Kim Long', @SoDT_HD OUTPUT ,@SoDT_PB OUTPUT ;
3 SELECT
4     @SoDT_HD AS N'Số lượng đề tài hướng dẫn',
5     @SoDT_PB AS N'Số lượng đề tài phản biện';

```

- Kết quả: thông báo “Không tìm thấy giáo viên”.

```

1 Msg 51000, Level 16, State 1, Procedure sp_ThongTinDeTaiGV, Line 34
2 Không tìm thấy giáo viên

```

2.2.5.3. Ví dụ 2: trùng tên giáo viên

- TENGV : Nguyễn Văn An

```

1 DECLARE @SoDT_HD INT, @SoDT_PB INT
2 EXEC sp_ThongTinDeTaiGV N'Nguyễn Văn An', @SoDT_HD OUTPUT ,@SoDT_PB OUTPUT ;
3 SELECT
4     @SoDT_HD AS N'Số lượng đề tài hướng dẫn',
5     @SoDT_PB AS N'Số lượng đề tài phản biện';

```

- Kết quả: thông báo “Trùng tên giáo viên”.

```

1 Msg 51000, Level 16, State 2, Procedure sp_ThongTinDeTaiGV, Line 30
2 Trùng tên giáo viên

```

2.2.5.4. Ví dụ 3: tên giáo viên hợp lệ

- TENGV : Trần Trung

```

1 DECLARE @SoDT_HD INT, @SoDT_PB INT
2 EXEC sp_ThongTinDeTaiGV N'Trần Trung', @SoDT_HD OUTPUT ,@SoDT_PB OUTPUT ;
3 SELECT
4     N'**Số lượng**' AS N'**Loại**',
5     @SoDT_HD AS N'Dề tài hướng dẫn',
6     @SoDT_PB AS N'Dề tài phản biện';

```

- Kết quả: trả về số lượng đề tài tương ứng hướng dẫn và phản biện.

Loại	Đề tài hướng dẫn	Đề tài phản biện
Số lượng	1	1

2.3. C. TRIGGER

2.3.1. Trigger khi Xóa Đè Tài

- Tạo Trigger thỏa mãn điều kiện khi xóa một đè tài (MSDT) sẽ xóa các thông tin liên quan.

Tham khảo: [INSTEAD OF](#).

2.3.1.1. Trigger

- Xóa trigger đã tạo trước đó nếu đây là lần chạy thứ 2 trở lên. Đảm bảo luôn có duy nhất 1 trigger được tạo và mới nhất.

```
1 IF OBJECT_ID('BTTH2_TRG_XoaDeTai', 'TR') IS NOT NULL  
2     DROP TRIGGER BTTH2_TRG_XoaDeTai;  
3 GO
```

- Tạo Trigger mới trên `DETAI`. DÙNG INSTEAD OF cho mục đích này.

```
1 CREATE TRIGGER BTTH2_TRG_XoaDeTai ON DETAI  
2 INSTEAD OF DELETE  
3 AS  
4 BEGIN  
5     -- Xóa dữ liệu tham chiếu ở các bảng con dựa trên MSDT  
6     DELETE FROM SV_DETAI  
7     WHERE MSDT IN (SELECT MSDT FROM DELETED);  
8  
9     DELETE FROM GV_HDDT  
10    WHERE MSDT IN (SELECT MSDT FROM DELETED);  
11  
12    DELETE FROM GV_PBDT  
13    WHERE MSDT IN (SELECT MSDT FROM DELETED);  
14  
15    DELETE FROM GV_UVDT  
16    WHERE MSDT IN (SELECT MSDT FROM DELETED);  
17  
18    DELETE FROM HOIDONG_DT  
19    WHERE MSDT IN (SELECT MSDT FROM DELETED);  
20  
21    -- Xóa chính đè tài đó trong bảng DETAI  
22    DELETE FROM DETAI  
23    WHERE MSDT IN (SELECT MSDT FROM DELETED);  
24  
25    PRINT N'Thực thi BTTH2_TRG_XoaDeTai: Đã xóa đè tài và các dữ liệu liên quan.';  
26 END;  
27 GO
```

2.3.1.2. Ví dụ

```
1 -- Ví dụ: Xóa dè tài '97001' (Quản lý thư viện)
2 DELETE FROM DETAI WHERE MSDT = '97001';
3
4 -- Sau khi chạy, bạn có thể select lại để kiểm tra xem nó còn tồn tại không
5 SELECT * FROM DETAI;
6 SELECT * FROM SV_DETAI;
```

- MSDT 97001 không còn tồn tại trong DETAI:

```
1 MSDT      TENDT
2 97002     Nhận dạng vân tay
3 97003     Bán đấu giá trên mạng
4 97004     Quản lý siêu thị
5 97005     Xử lý ảnh
6 97006     Hệ giải toán thông minh
```

- MSDT 97001 không còn tồn tại trong SV_DETAI:

```
1      MSSV      MSDT
2 13520001  97004
3 13520002  97005
4 13520004  97002
5 13520005  97003
6 13520006  97005
```

2.3.2. Trigger thay đổi MSGV

- Tạo Trigger thỏa mãn ràng buộc là khi đổi 1 mã số giáo viên (MSGV) thì sẽ thay đổi các thông tin liên quan.

2.3.2.1. Trigger

- Xóa trigger đã tạo trước đó nếu đây là lần chạy thứ 2 trở lên. Đảm bảo luôn có duy nhất 1 trigger được tạo và mới nhất.

```
1 IF OBJECT_ID('BTTH2_TRG_CapNhatMSGV', 'TR') IS NOT NULL
2     DROP TRIGGER BTTH2_TRG_CapNhatMSGV;
3 GO
```

- Tạo Trigger mới trên GIAOVIEN sử dụng cả INSERTED và DELETED để lưu trữ dữ liệu mới/cũ tương ứng.

```

1 CREATE TRIGGER BTTH2_TRG_CapNhatMSGV ON GIAOVIEN
2 INSTEAD OF UPDATE
3 AS
4 BEGIN
5     -- Khai báo biến để giữ giá trị cũ và mới
6     DECLARE @OldMSGV INT;
7     DECLARE @NewMSGV INT;
8
9     -- Lấy MSGV cũ từ bảng DELETED (dữ liệu trước khi update)
10    SELECT @OldMSGV = MSGV FROM DELETED;
11
12    -- Lấy MSGV mới từ bảng INSERTED (dữ liệu người dùng muốn update)
13    SELECT @NewMSGV = MSGV FROM INSERTED;
14
15    -- Kiểm tra: Nếu người dùng có sửa MSGV (Khóa chính)
16    IF (@OldMSGV <> @NewMSGV)
17        BEGIN
18            -- 1. Thêm dòng giáo viên mới với thông tin từ bảng INSERTED
19            INSERT INTO GIAOVIEN (MSGV, TENGV, DIACHI, SODT, MSHH, NAMHH)
20            SELECT MSGV, TENGV, DIACHI, SODT, MSHH, NAMHH
21            FROM INSERTED;
22
23            -- 2. Cập nhật các bảng con đang giữ MSGV cũ thành MSGV mới
24            UPDATE GV_HV_CN SET MSGV = @NewMSGV WHERE MSGV = @OldMSGV;
25            UPDATE GV_HDDT SET MSGV = @NewMSGV WHERE MSGV = @OldMSGV;
26            UPDATE GV_PBDT SET MSGV = @NewMSGV WHERE MSGV = @OldMSGV;
27            UPDATE GV_UVDT SET MSGV = @NewMSGV WHERE MSGV = @OldMSGV;
28            UPDATE HOIDONG SET MSGV = @NewMSGV WHERE MSGV = @OldMSGV;
29            UPDATE HOIDONG_GV SET MSGV = @NewMSGV WHERE MSGV = @OldMSGV;
30
31            -- 3. Xóa dòng giáo viên cũ (dựa trên bảng DELETED)
32            DELETE FROM GIAOVIEN WHERE MSGV = @OldMSGV;
33        END
34    ELSE
35        -- Trường hợp: Người dùng chỉ sửa tên, địa chỉ... (không sửa Khóa chính)
36        BEGIN
37            UPDATE GIAOVIEN
38            SET TENGV = I.TENGV,
39                DIACHI = I.DIACHI,
40                SODT = I.SODT,
41                MSHH = I.MSHH,
42                NAMHH = I.NAMHH
43            FROM GIAOVIEN G
44            JOIN INSERTED I ON G.MSGV = I.MSGV;
45        END
46    END;
47    GO

```

2.3.2.2. Ví dụ

- Kiểm tra trước khi đổi (xem bảng GV_HDDT của 202)

```
1 SELECT * FROM GIAOVIEN WHERE MSGV = 202;
2 SELECT * FROM GV_HDDT WHERE MSGV = 202;
3 -- Quên không copy kết quả ở lần chạy đầu
```

- Thực hiện đổi ID

```
1 UPDATE GIAOVIEN
2 SET MSGV = 2020
3 WHERE MSGV = 202;
```

- Kiểm tra lại: 202 sẽ mất, thay bằng 2020, dữ liệu con cũng đổi theo.

```
1 SELECT * FROM GIAOVIEN WHERE MSGV = 202;
2 -- Trả về 0 kết quả
```

- Kiểm tra xem 2020 đã xuất hiện và thông tin liên quan đã chuyển qua.

```
1 SELECT * FROM GIAOVIEN WHERE MSGV = 2020;
```

	MSGV	TENGV	MSHH
1	201	Trần Trung	1
2	203	Trần Thu Trang	1
3	204	Nguyễn Thị Loan	2
4	205	Chu Tiến	2
5	2020	Nguyễn Văn An	1

```
1 SELECT * FROM GV_HDDT WHERE MSGV = 2020;
```

	MSGV	MSDT	DIEM
1	203	97005	9
2	204	97004	7
3	2020	97002	7

2.3.3. Trigger Hội Đồng và Số Lượng Đề Tài

- Tạo Trigger thỏa mãn ràng buộc là một hội đồng không quá 10 đề tài.
- Dùng “Group by” có được không? Giải thích.

2.3.3.1. Trigger

- Xóa trigger đang tồn tại nếu có (ở các lần chạy thứ 2 trở đi).

```

1 IF OBJECT_ID('BTTH2_TRG_KiemTraSoLuongDeTai', 'TR') IS NOT NULL
2     DROP TRIGGER BTTH2_TRG_KiemTraSoLuongDeTai;
3 GO

```

- Tạo Trigger kiểm tra ràng buộc số lượng Đè Tài
 - Sử dụng `FOR (AFTER)`.
 - Kiểm tra xem có hội đồng nào vi phạm quy định (> 10 đè tài) hay không.
 - Logic: Chỉ cần kiểm tra các `MSHD` vừa bị tác động (có trong bảng `INSERTED`).
 - Nếu tìm thấy hội đồng nào có > 10 đè tài, hủy bỏ thao tác (`INSERT / UPDATE`) và báo lỗi.

```

1 CREATE TRIGGER BTTH2_TRG_KiemTraSoLuongDeTai
2 ON HOIDONG_DT
3 FOR INSERT, UPDATE
4 AS
5 BEGIN
6     IF EXISTS (
7         SELECT HOIDONG_DT.MSHD
8             FROM HOIDONG_DT
9                 JOIN INSERTED ON HOIDONG_DT.MSHD = INSERTED.MSHD
10                GROUP BY HOIDONG_DT.MSHD
11                   HAVING COUNT(HOIDONG_DT.MSHD) > 10
12     )
13     BEGIN
14         RAISERROR(N'Lỗi: Quá 10 đè tài trên Hội Đồng.', 16, 1);
15         ROLLBACK TRANSACTION;
16     END
17 END;
18 GO

```

2.3.3.2. Giải Thích

- CÓ thẻ dùng `GROUP BY`. Rất cần thiết cho chức năng đếm số lượng/lần xuất hiện giá trị của một trường cụ thể.
- VÌ:
 - Thông kê: Ràng buộc yêu cầu đếm số lượng đè tài trên mỗi hội đồng.
 - Trong mô hình quan hệ, bảng `HOIDONG_DT` chứa các dòng riêng lẻ (`MSHD`, `MSDT`).
 - Để biết một `MSHD` xuất hiện bao nhiêu lần (tương ứng với số lượng đè tài), ta gom nhóm các dòng có cùng `MSHD` lại với mệnh đề `GROUP BY`.
 - Kết hợp với `HAVING`: Sau khi `GROUP BY MSHD`, ta dùng hàm gộp `COUNT(MSDT)` để đếm và mệnh đề `HAVING` để lọc ra những nhóm vi phạm điều kiện (> 10).

2.3.3.3. Ví Dụ

- Thêm số lượng đè tài trước khi kiểm thử: với các đè tài bổ sung, và một hội đồng giả định.

```
1 INSERT INTO DETAI (MSDT, TENDT)
2 VALUES
3 ('T01', 'Test 1'),
4 ('T02', 'Test 2'),
5 ('T03', 'Test 3'),
6 ('T04', 'Test 4'),
7 ('T05', 'Test 5'),
8 ('T06', 'Test 6');
```

```
1 INSERT INTO HOIDONG (MSHD, PHONG, TGBD, NGAYHD, TINHTRANG, MSGV)
2 VALUES (99, 101, '07:00', '2024-01-01', N'Test', 201);
```

- Thêm 10 đè tài vào hội đồng giả định này, khiến số lượng đang ở = 10 .

```
1 INSERT INTO HOIDONG_DT (MSHD, MSDT, QUYETDINH) VALUES
2 (99, '97002', N'Được'),
3 (99, '97003', N'Được'),
4 (99, '97004', N'Được'),
5 (99, '97005', N'Được'),
6 (99, '97006', N'Được'),
7 (99, 'T01', N'Được'),
8 (99, 'T02', N'Được'),
9 (99, 'T03', N'Được'),
10 (99, 'T04', N'Được'),
11 (99, 'T05', N'Được');
```

	MSHD	MSDT	QUYETDINH
2	99	97002	Được
3	99	97003	Được
4	99	97004	Được
5	99	97005	Được
6	99	97006	Được
7	99	T01	Được
8	99	T02	Được
9	99	T03	Được
10	99	T04	Được
11	99	T05	Được

- Kiểm tra số lượng hiện tại

```
1 SELECT MSHD, COUNT(MSDT) as SoLuong FROM HOIDONG_DT WHERE MSHD = 99 GROUP BY MSHD;
```

```
1 MSHD      SoLuong  
2 99        10
```

- Kiểm thử: Thêm đề tài T06 vào hội đồng giả định này, khiến điều kiện trở thành > 10 .

```
1 INSERT INTO HOIDONG_DT (MSHD, MSDT, QUYETDINH) VALUES (99, 'T06', N'Dược');
```

- Gặp lỗi: Quá số lượng 10, không cho phép hoàn thành.

```
1 Msg 50000, Level 16, State 1, Procedure BTTH2_TRG_KiemTraSoLuongDeTai, Line 16  
2 Lỗi: Quá 10 đề tài trên Hội Đồng.
```

2.3.4. Trigger Đề Tài và Số Lượng Sinh Viên

- Tạo Trigger thỏa mãn ràng buộc là một đề tài không quá 2 sinh viên.
- Dùng “Group by” có được không? Giải thích.

2.3.4.1. Trigger

- Xóa trigger đang tồn tại nếu có (ở các lần chạy thứ 2 trở đi).

```
1 IF OBJECT_ID('BTTH2_TRG_KiemTraSoLuongSinhVien', 'TR') IS NOT NULL  
2   DROP TRIGGER BTTH2_TRG_KiemTraSoLuongSinhVien;  
3 GO
```

- Tạo Trigger kiểm tra ràng buộc số lượng Sinh Viên thực hiện Đề Tài.
 - Sử dụng `FOR INSERT / FOR UPDATE` (tương đương `AFTER`).
 - Kiểm tra xem có đề tài nào vừa bị tác động mà số lượng sinh viên tham gia vượt quá 2 hay không.

```

1 CREATE TRIGGER BTTH2_TRG_KiemTraSoLuongSinhVien
2 ON SV_DETAI
3 FOR INSERT, UPDATE
4 AS
5 BEGIN
6     IF EXISTS (
7         SELECT SV_DETAI.MSDT
8         FROM SV_DETAI
9         JOIN INSERTED ON SV_DETAI.MSDT = INSERTED.MSDT
10        GROUP BY SV_DETAI.MSDT
11        HAVING COUNT(SV_DETAI.MSSV) > 2
12    )
13    BEGIN
14        RAISERROR(N'Lỗi: Một đề tài không được quá 2 sinh viên thực hiện.', 16, 1);
15        ROLLBACK TRANSACTION;
16    END
17 END;
18 GO

```

2.3.4.2. Giải Thích

- CÓ thể dùng `GROUP BY`. Đây là mệnh đề bắt buộc để thực hiện yêu cầu này.
- VÌ:
 - ▶ Mục đích thống kê: Chúng ta cần đếm số lượng sinh viên (`MSSV`) thuộc về từng đề tài (`MSDT`) riêng biệt.
 - ▶ Cơ chế hoạt động: Trong bảng `SV_DETAI`, dữ liệu được lưu dưới dạng từng dòng chi tiết (Mỗi dòng là một sinh viên đăng ký một đề tài).
 - ▶ Kết hợp `HAVING`: Mệnh đề `GROUP BY SV_DETAI.MSDT` sẽ gom tất cả các dòng có cùng mã đề tài lại thành một nhóm. Sau đó, hàm `COUNT(SV_DETAI.MSSV)` sẽ đếm số thành viên trong nhóm đó. Cuối cùng, `HAVING` sẽ lọc ra nhóm nào có kết quả đếm `> 2` để Trigger xử lý báo lỗi.

2.3.4.3. Ví Dụ

- Chuẩn bị dữ liệu kiểm thử
 - ▶ Tạo một đề tài kiểm thử mới mã 99999.
 - ▶ Chúng ta đang có sẵn 3 sinh viên để thực hiện việc thêm vào đề tài này.

```

1 IF NOT EXISTS (SELECT * FROM DETAI WHERE MSDT = '99999')
2 INSERT INTO DETAI (MSDT, TENDT) VALUES ('99999', N'Dề tài kiểm thử Trigger SV');
3 GO

```

- Thêm 2 sinh viên vào đề tài (Hợp lệ)

```

1 -- Xóa dữ liệu cũ của đè tài 99999 trong bảng SV_DETAI nếu có đè test lại từ đầu
2 DELETE FROM SV_DETAI WHERE MSDT = '99999';
3 GO
4
5 -- Thêm sinh viên thứ 1
6 INSERT INTO SV_DETAI (MSSV, MSDT) VALUES ('13520001', '99999');
7 GO
8
9 -- Thêm sinh viên thứ 2
10 INSERT INTO SV_DETAI (MSSV, MSDT) VALUES ('13520002', '99999');
11 GO

```

- Kiểm tra số lượng hiện tại:

```

1 SELECT MSDT, COUNT(MSSV) AS SoLuongSV
2 FROM SV_DETAI
3 WHERE MSDT = '99999'
4 GROUP BY MSDT;
5 GO

```

```

1 MSDT      SoLuongSV
2 99999     2

```

- Thêm sinh viên thứ 3 (Vi phạm ràng buộc)

```

1 INSERT INTO SV_DETAI (MSSV, MSDT) VALUES ('13520003', '99999');
2 GO

```

- Hệ thống báo lỗi và hủy thao tác.

```

1 Msg 50000, Level 16, State 1, Procedure BTTH2_TRG_KiemTraSoLuongSinhVien, Line 15
2 Lỗi: Một đè tài không được quá 2 sinh viên thực hiện.
3 Msg 3609, Level 16, State 1, Line 46
4 The transaction ended in the trigger. The batch has been aborted.

```

2.3.5. Trigger Giáo Viên và Học Hàm

- **Yêu cầu:** Tạo Trigger thỏa mãn ràng buộc là một giáo viên muốn có học hàm Phó Giáo Sư (PGS) thì bắt buộc phải có học vị Tiến sĩ.

2.3.5.1. Trigger

- Xóa trigger cũ nếu tồn tại.

```

1 IF OBJECT_ID('BTTH2_TRG_KiemTraHocHamGiaoVien', 'TR') IS NOT NULL
2     DROP TRIGGER BTTH2_TRG_KiemTraHocHamGiaoVien;
3 GO

```

- Tạo Trigger trên bảng `GIAOVIEN`.
- **Sự kiện:** `INSERT`, `UPDATE` (Kiểm tra khi thêm mới hoặc khi sửa học hàm).
- **Logic:**
 - ▶ Lọc ra danh sách các Giáo viên trong bảng `INSERTED` đang được gán học hàm là ‘PHÓ GIÁO SƯ’.
 - ▶ Với mỗi giáo viên đó, kiểm tra trong bảng `GV_HV_CN` (Giáo viên - Học vị - Chuyên Ngành) xem họ có sở hữu học vị ‘Tiến sĩ’ (hoặc ‘Tiến sĩ Khoa học’) hay không.
 - ▶ Nếu không tìm thấy học vị Tiến sĩ, hệ thống sẽ báo lỗi và hủy thao tác.

```

1 CREATE TRIGGER BTTH2_TRG_KiemTraHocHamGiaoVien
2 ON GIAOVIEN
3 FOR INSERT, UPDATE
4 AS
5 BEGIN
6     IF EXISTS (
7         SELECT INSERTED.MSGV
8             FROM INSERTED
9                 JOIN HOCHAM ON INSERTED.MSHH = HOCHAM.MSHH
10                WHERE HOCHAM.TENHH = N'PHÓ GIÁO SƯ'
11                AND NOT EXISTS (
12                    SELECT 1
13                        FROM GV_HV_CN
14                            JOIN HOCVI ON GV_HV_CN.MSHV = HOCVI.MSHV
15                            WHERE GV_HV_CN.MSGV = INSERTED.MSGV
16                            AND (HOCVI.TENHV = N'Tiến sĩ' OR HOCVI.TENHV = N'Tiến sĩ Khoa học')
17                )
18            )
19            BEGIN
20                RAISERROR(N'Lỗi: Giáo viên muốn được phong Phó Giáo Sư phải có học vị Tiến sĩ.', 16, 1);
21                ROLLBACK TRANSACTION;
22            END
23        END;
24    GO

```

2.3.5.2. Ví Dụ

Để kiểm thử logic này một cách thực tế, chúng ta cần một trạng thái trung gian (Giáo viên chưa có học hàm) để thêm bằng cấp trước khi phong hàm.

- Chuẩn bị dữ liệu

- ▶ Thêm một loại học hàm tạm thời là “Chưa có” (Mã 0) vào bảng HOCHAM để có thể tạo giáo viên ban đầu.
- ▶ Tạo một giáo viên kiểm thử (Mã 901) với học hàm “Chưa có”.

```

1 IF NOT EXISTS (SELECT * FROM HOCHAM WHERE MSHH = 0)
2   INSERT INTO HOCHAM (MSHH, TENHH) VALUES (0, N'Chưa có');
3 GO
4
5 IF NOT EXISTS (SELECT * FROM GIAOVIEN WHERE MSGV = 901)
6   INSERT INTO GIAOVIEN (MSGV, TENGV, DIACHI, SODT, MSHH, NAMHH)
7     VALUES (901, N'Nguyễn Văn Test', N'TP.HCM', '0909090909', 0, '2024');
8 GO

```

• Gán học vị Cử nhân và cố gắng phong hàm PGS (Vi phạm)

- ▶ Giáo viên 901 chỉ mới có bằng Cử nhân.
- ▶ Thực hiện cập nhật MSHH lên mức ‘PHÓ GIÁO SƯ’ (Mã 1 - Theo dữ liệu bài 1).

```

1 INSERT INTO GV_HV_CN (MSGV, MSHV, MSCN, NAM) VALUES (901, 2, 1, '2010');
2 GO
3
4 -- Cố gắng cập nhật lên Phó Giáo Sư (Mã 1)
5 UPDATE GIAOVIEN SET MSHH = 1 WHERE MSGV = 901;
6 GO

```

• Hệ thống báo lỗi.

```

1 Msg 50000, Level 16, State 1, Procedure BTTH2_TRG_KiemTraHocHamGiaoVien, Line 23
2 Lỗi: Giáo viên muôn được phong Phó Giáo Sư phải có học vị Tiến sĩ.
3 Msg 3609, Level 16, State 1, Procedure BTTH2_TRG_CapNhatMSGV, Line 40
4 The transaction ended in the trigger. The batch has been aborted.

```

• Bổ sung học vị Tiến sĩ và phong hàm lại (Hợp lệ)

```

1 -- Bổ sung bằng Tiến sĩ (Mã 4) cho GV 901
2 INSERT INTO GV_HV_CN (MSGV, MSHV, MSCN, NAM) VALUES (901, 4, 1, '2015');
3 GO
4
5 -- Cập nhật lại lên Phó Giáo Sư (Mã 1)
6 UPDATE GIAOVIEN SET MSHH = 1 WHERE MSGV = 901;
7 GO

```

• Cập nhật thành công. Kiểm tra lại dữ liệu.

```

1 SELECT GV.MSGV, GV.TENGV, HH.TENHH
2 FROM GIAOVIEN GV
3 JOIN HOCHAM HH ON GV.MSHH = HH.MSHH
4 WHERE GV.MSGV = 901;

```

1	MSGV	TENGV	TENHH
2	901	Nguyễn Văn Test	PHÓ GIÁO SƯ

2.4. D. FUNCTION

2.4.1. Tính Điểm Trung Bình của Đề Tài từ MSDT

- Viết hàm tính điểm trung bình của một đề tài.
- Giá trị trả về là điểm trung bình ứng với mã số đề tài nhập vào.

```

1 IF OBJECT_ID('f_TinhDiemTB', 'FN') IS NOT NULL
2     DROP FUNCTION f_TinhDiemTB;
3 GO
4
5 CREATE FUNCTION f_TinhDiemTB (@MSDT char(6))
6 RETURNS float
7 AS
8 BEGIN
9     IF NOT EXISTS (SELECT 1 FROM DETAI WHERE MSDT = @MSDT)
10        RETURN 0;
11
12    DECLARE @DiemTB float;
13
14    SELECT @DiemTB = AVG(DIEM)
15    FROM (
16        SELECT DIEM FROM GV_HDDT WHERE MSDT = @MSDT
17        UNION ALL
18        SELECT DIEM FROM GV_PBDT WHERE MSDT = @MSDT
19        UNION ALL
20        SELECT DIEM FROM GV_UVDT WHERE MSDT = @MSDT
21    ) AS TongHopDiem;
22
23    RETURN ROUND(ISNULL(@DiemTB, 0), 3);
24 END;
25 GO

```

2.4.1.1. Ví dụ 1

- Tính điểm tất cả đề tài trong bảng DETAI.

```

1 SELECT
2     MSDT,
3     TENDT,
4     dbo.f_TinhDiemTB(MSDT) AS DiemTrungBinh
5 FROM DETAI;

```

MSDT	TENDT	DiemTrungBinh
97001	Quản lý thư viện	8
97002	Nhận dạng vân tay	8.2
97003	Bán đấu giá trên mạng	7.5
97004	Quản lý siêu thị	8
97005	Xử lý ảnh	8.2
97006	Hệ giải toán thông minh	8.333

2.4.1.2. Ví dụ 2

- Tính điểm của 1 đề tài nhập vào cụ thể.

```

1 SELECT
2     '97001' AS 'MSDT',
3     dbo.f_TinhDiemTB('97001') AS DiemTrungBinh;

```

MSDT	DiemTrungBinh
97001	8

2.4.1.3. Ví dụ 3

- MSDT không tồn tại.

```

1 SELECT
2     '99999' AS 'MSDT',
3     dbo.f_TinhDiemTB('99999') AS DiemTrungBinh;

```

MSDT	DiemTrungBinh
99999	0

2.4.2. Kết Quả của Đề Tài từ MSDT

- Trả về kết quả của đề tài theo MSDT nhập vào.
- Kết quả là DAT nếu như điểm trung bình từ 5 trở lên, và KHONGDAT nếu như điểm trung bình dưới 5.

```

1 IF OBJECT_ID('f_KetQuaDeTai', 'FN') IS NOT NULL
2     DROP FUNCTION f_KetQuaDeTai;
3 GO
4
5 CREATE FUNCTION f_KetQuaDeTai (@MSDT char(6))
6 RETURNS NVARCHAR(20)
7 AS
8 BEGIN
9     DECLARE @DiemTB float;
10    DECLARE @KetQua NVARCHAR(20);
11
12    SET @DiemTB = dbo.f_TinhDiemTB(@MSDT);
13
14    IF (@DiemTB ≥ 5)
15        SET @KetQua = N'DAT';
16    ELSE
17        SET @KetQua = N'KHONGDAT';
18
19    RETURN @KetQua;
20 END;
21 GO

```

2.4.2.1. Ví dụ 1

- Một đề tài cụ thể.

```

1 SELECT
2     '97001' AS 'MSDT',
3     dbo.f_KetQuaDeTai('97001') AS KetQuaDeTai;

```

MSDT	KetQuaDeTai
97001	DAT

2.4.2.2. Ví dụ 2

- Đề tài không tồn tại (được xem là KHÔNG ĐẠT).

```

1 SELECT
2     '99999' AS 'MSDT',
3     dbo.f_KetQuaDeTai('99999') AS KetQuaDeTai;

```

MSDT	KetQuaDeTai
99999	KHONGDAT

2.4.3. Thông Tin Sinh Viên từ `MSDT`

- Đưa vào `MSDT`, trả về mã số và họ tên của các sinh viên thực hiện đề tài.

```
1 IF OBJECT_ID('f_DanhSachSVThucHienDeTai', 'IF') IS NOT NULL
2     DROP FUNCTION f_DanhSachSVThucHienDeTai;
3 GO
4
5 CREATE FUNCTION f_DanhSachSVThucHienDeTai (@MSDT char(6))
6 RETURNS TABLE
7 AS
8 RETURN (
9     SELECT
10        SV.MSSV,
11        SV.TENSV
12    FROM SINHVIEN SV
13    JOIN SV_DETAI SD ON SV.MSSV = SD.MSSV
14    WHERE SD.MSDT = @MSDT
15 );
16 GO
```

2.4.3.1. Ví dụ 1

- `MSDT` cụ thể: 97001

```
1 SELECT * FROM dbo.f_DanhSachSVThucHienDeTai('97001');
```

MSSV	TENSV
13520003	Nguyễn Anh Hải

2.4.3.2. Ví dụ 2

- `MSDT` không tồn tại, có nghĩa không có sinh viên. Kết quả trống.

```
1 SELECT * FROM dbo.f_DanhSachSVThucHienDeTai('97006');
```

MSSV	TENSV

2.5. E. CURSOR

2.5.1. Tạo Bảng DETAI_DIEM

```
1 CREATE TABLE DETAI_DIEM (
2     MSDT char(6) PRIMARY KEY,
3     DIEMTB FLOAT,
4     CONSTRAINT FK_DETAI_DIEM_DETAI FOREIGN KEY (MSDT) REFERENCES DETAI(MSDT)
5 );
```

DETAI_DIEM đã được tạo:

database	schema	name	type
IE103-BTTH2	dbo	CHUYENNGANH	BASE TABLE
IE103-BTTH2	dbo	DETAI	BASE TABLE
IE103-BTTH2	dbo	GIAOVIEN	BASE TABLE
IE103-BTTH2	dbo	GV_HDDT	BASE TABLE
IE103-BTTH2	dbo	GV_HV_CN	BASE TABLE
IE103-BTTH2	dbo	GV_PBDT	BASE TABLE
IE103-BTTH2	dbo	GV_UVDT	BASE TABLE
IE103-BTTH2	dbo	HOCHAM	BASE TABLE
IE103-BTTH2	dbo	HOCVI	BASE TABLE
IE103-BTTH2	dbo	HOIDONG	BASE TABLE
IE103-BTTH2	dbo	HOIDONG_DT	BASE TABLE
IE103-BTTH2	dbo	HOIDONG_GV	BASE TABLE
IE103-BTTH2	dbo	SINHVIEN	BASE TABLE
IE103-BTTH2	dbo	SV_DETAI	BASE TABLE
IE103-BTTH2	dbo	DETAI_DIEM	BASE TABLE

2.5.2. Cursor Tính Điểm Trung Bình

- Viết Cursor tính điểm trung bình cho từng đề tài.
- Sau đó lưu kết quả vào bảng DETAI_DIEM.

Khởi tạo Cursor:

- Khai báo biến

```
1 DECLARE @MSDT CHAR(6);
2 DECLARE @DIEMTB FLOAT;
```

- Xóa dữ liệu cũ trong bảng kết quả (nếu có)

```
1 DELETE FROM DETAI_Diem;
```

- Khai báo Cursor

```
1 DECLARE CUR_TINH_Diem CURSOR FOR
2     SELECT MSDT
3         FROM DETAI;
4
5 -- Mở Cursor
6 OPEN CUR_TINH_Diem;
7
8 -- Lấy dòng đầu tiên
9 FETCH NEXT FROM CUR_TINH_Diem INTO @MSDT;
10
11 WHILE @@FETCH_STATUS = 0
12 BEGIN
13     -- Tính điểm trung bình cho đề tài
14     SELECT @DIEMTB = ROUND(AVG(DIEM), 2)
15     FROM (
16         SELECT DIEM FROM GV_HDDT WHERE MSDT = @MSDT
17         UNION ALL
18         SELECT DIEM FROM GV_PBDT WHERE MSDT = @MSDT
19         UNION ALL
20         SELECT DIEM FROM GV_UVDT WHERE MSDT = @MSDT
21     ) AS T;
22
23     -- Lưu kết quả vào DETAI_Diem
24     INSERT INTO DETAI_Diem(MSDT, DIEMTB)
25     VALUES (@MSDT, @DIEMTB);
26
27     -- Lấy dòng tiếp theo
28     FETCH NEXT FROM CUR_TINH_Diem INTO @MSDT;
29 END
30
31 -- Đóng và hủy Cursor
32 CLOSE CUR_TINH_Diem;
33 DEALLOCATE CUR_TINH_Diem;
```

- Kiểm tra bảng `DETAI_Diem` đã nhập được chưa.

```
1 SELECT * FROM DETAI_Diem;
```

MSDT	DIEMTB
97001	8
97002	8.2

97003	7.5
97004	8
97005	8.2
97006	8.33

2.5.3. Tạo Stored Procedure cho Tính Điểm Trung Bình

- Gom các bước xử lý của Cursor ở Mục Tính Điểm Trung Bình 2.5.2 vào một Stored Procedure.

```

1 CREATE OR ALTER PROCEDURE SP_TINH_DIEMTB_DETAI
2 AS
3 BEGIN
4     SET NOCOUNT ON;
5
6     DECLARE @MSDT CHAR(6);
7     DECLARE @DIEMTB FLOAT;
8
9     -- Làm sạch: Xóa dữ liệu cũ (tránh trùng khi chạy lại)
10    DELETE FROM DETAI_DIEM;
11
12    -- Cursor duyệt từng đè tài
13    DECLARE CUR_DETAI CURSOR FOR
14        SELECT MSDT
15        FROM DETAI;
16
17    OPEN CUR_DETAI;
18    FETCH NEXT FROM CUR_DETAI INTO @MSDT;
19
20    WHILE @@FETCH_STATUS = 0
21    BEGIN
22        -- Tính điểm trung bình của đè tài
23        SELECT @DIEMTB = ROUND(AVG(DIEM), 2)
24        FROM (
25            SELECT DIEM FROM GV_HDDT WHERE MSDT = @MSDT
26            UNION ALL
27            SELECT DIEM FROM GV_PBDT WHERE MSDT = @MSDT
28            UNION ALL
29            SELECT DIEM FROM GV_UVDT WHERE MSDT = @MSDT
30        ) AS T;
31
32        -- Lưu vào bảng DETAI_DIEM
33        INSERT INTO DETAI_DIEM(MSDT, DIEMTB)
34        VALUES (@MSDT, @DIEMTB);
35
36        FETCH NEXT FROM CUR_DETAI INTO @MSDT;
37    END
38
39    CLOSE CUR_DETAI;
40    DEALLOCATE CUR_DETAI;
41 END;

```

- Chạy Stored Procedure vừa tạo:

```
1 EXEC SP_TINH_DIEMTB_DETAI;
```

- Kiểm tra bảng `DETAI_DIEM`:

```
1 SELECT * FROM DETAI_DIEM;
```

MSDT	DIEMTB
97001	8
97002	8.2
97003	7.5
97004	8
97005	8.2
97006	8.33

2.5.4. Xếp Loại Kết Quả với XEPMLOAI

2.5.4.1. Yêu cầu

- Tạo thêm cột XEPMLOAI có kiểu là NVARCHAR(20) trong bảng DETAI_DIEM.
- Viết Cursor cập nhật kết quả xếp loại cho mỗi đề tài như sau:
 - “Xuất sắc”: điểm trung bình từ 9 đến 10.
 - “Giỏi”: điểm trung bình từ 8 đến 9.
 - “Khá”: điểm trung bình từ 7 đến 8.
 - “Trung bình khá”: điểm trung bình từ 6 đến 7.
 - “Trung bình”: điểm trung bình từ 5 đến 6.
 - “Yếu”: điểm trung bình từ 4 đến 5.
 - “Kém”: điểm trung bình dưới 4.

2.5.4.2. Thực hiện

- Thêm cột XEPMLOAI :

```
1 ALTER TABLE DETAI_DIEM  
2 ADD XEPMLOAI NVARCHAR(20);
```

- Khai báo biến

```
1 DECLARE @MSDT CHAR(6);  
2 DECLARE @DIEMTB FLOAT;  
3 DECLARE @XEPMLOAI NVARCHAR(20);
```

- Khai báo Cursor, lấy mã đề tài và điểm trung bình:

```
1 DECLARE CUR_XEPMLOAI CURSOR FOR  
2     SELECT MSDT, DIEMTB  
3     FROM DETAI_DIEM;
```

- Câu hình Cursor:

```

1 -- Mở Cursor
2 OPEN CUR_XEPOAI;
3 FETCH NEXT FROM CUR_XEPOAI INTO @MSDT, @DIEMTB;
4
5 WHILE @@FETCH_STATUS = 0
6 BEGIN
7     IF @DIEMTB ≥ 9 AND @DIEMTB ≤ 10
8         SET @XEPOAI = N'Xuất sắc';
9     ELSE IF @DIEMTB ≥ 8
10        SET @XEPOAI = N'Giỏi';
11    ELSE IF @DIEMTB ≥ 7
12        SET @XEPOAI = N'Khá';
13    ELSE IF @DIEMTB ≥ 6
14        SET @XEPOAI = N'Trung bình khá';
15    ELSE IF @DIEMTB ≥ 5
16        SET @XEPOAI = N'Trung bình';
17    ELSE IF @DIEMTB ≥ 4
18        SET @XEPOAI = N'Yếu';
19    ELSE
20        SET @XEPOAI = N'Kém';
21
22     UPDATE DETAI_DIEM
23     SET XEPOAI = @XEPOAI
24     WHERE MSDT = @MSDT;
25
26     FETCH NEXT FROM CUR_XEPOAI INTO @MSDT, @DIEMTB;
27 END
28 -- Đóng Cursor sau khi duyệt xong dữ liệu
29 CLOSE CUR_XEPOAI;
30 -- Hủy Cursor sau khi hoàn thành
31 DEALLOCATE CUR_XEPOAI;

```

- Kiểm tra bảng `DETAI_DIEM`:

```
1 SELECT * FROM DETAI_DIEM;
```

MSDT	DIEMTB	XEPOAI
97001	8	Giỏi
97002	8.2	Giỏi
97003	7.5	Khá
97004	8	Giỏi
97005	8.2	Giỏi
97006	8.33	Giỏi

PHỤ LỤC: THIẾT LẬP MÔI TRƯỜNG LAB

3.1. Thiết bị

- macOS, M4 Pro (arm64).

3.2. Các công cụ

- `make` (cài đặt sẵn)
- `brew.sh`: quản lý gói.
- Colima
- `docker`
- Azure Database Studio (ADS)
- SQL Projects (ADS Extension)

3.3. Quản lý vòng đời của SQL Server.

- Sử dụng `make` với `Makefile`.

```
1 # Khởi tạo hoặc/và chạy SQL Server
2 make up
3
4 # Tắt/Shutdown SQL Server
5 make down
6
7 # Xóa bỏ SQL Server
8 make clean
```

3.4. Khởi tạo môi trường lab thủ công

- Kết nối vào SQL Server: `localhost`, `sa`, `Password123!`
- Tạo DB với `00-CREATE-DB.sql` hoặc nhập truy vấn thủ công.
- Chạy các script `.sql` tương ứng lần lượt từ thư mục `content/code`.
 - ▶ `bai-01-1.sql`
 - ▶ `bai-01-2.sql`
 - ▶ ...

3.5. Khởi tạo môi trường lab với ADS (SQL Projects)

- Mở thư mục Lab với ADS.
- Mở file project `IE103-BTTH2.sqlproj` từ SQL Projects.
- Chuột phải vào tên của project (IE103-BTTH2) và chọn Publish.
- Tiếp tục theo trình thuật sĩ hướng dẫn.
 - ▶ Có thẻ Load Profile `IE103-BTTH2.publish.xml` nếu muốn.