

BTVN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

- Họ và tên: Phạm Quang Trường.
- MSV: K215480106129.
- Lớp: K57KMT

Bài làm:

- Cấu trúc và dữ liệu đầu vào của database:

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The central pane shows the SQL script for creating a database named 'TRUOT' and several tables: 'Mon_Hoc', 'SV', 'GV', 'Lop_HP', and 'DKMH'. The script includes primary keys, foreign keys, and constraints. The right pane shows the 'Properties' window for the 'NEKOTRANG (sa)' connection, displaying connection details such as connection name, state, and server version. The bottom status bar indicates the query was executed successfully.

```
CREATE DATABASE TRUOT

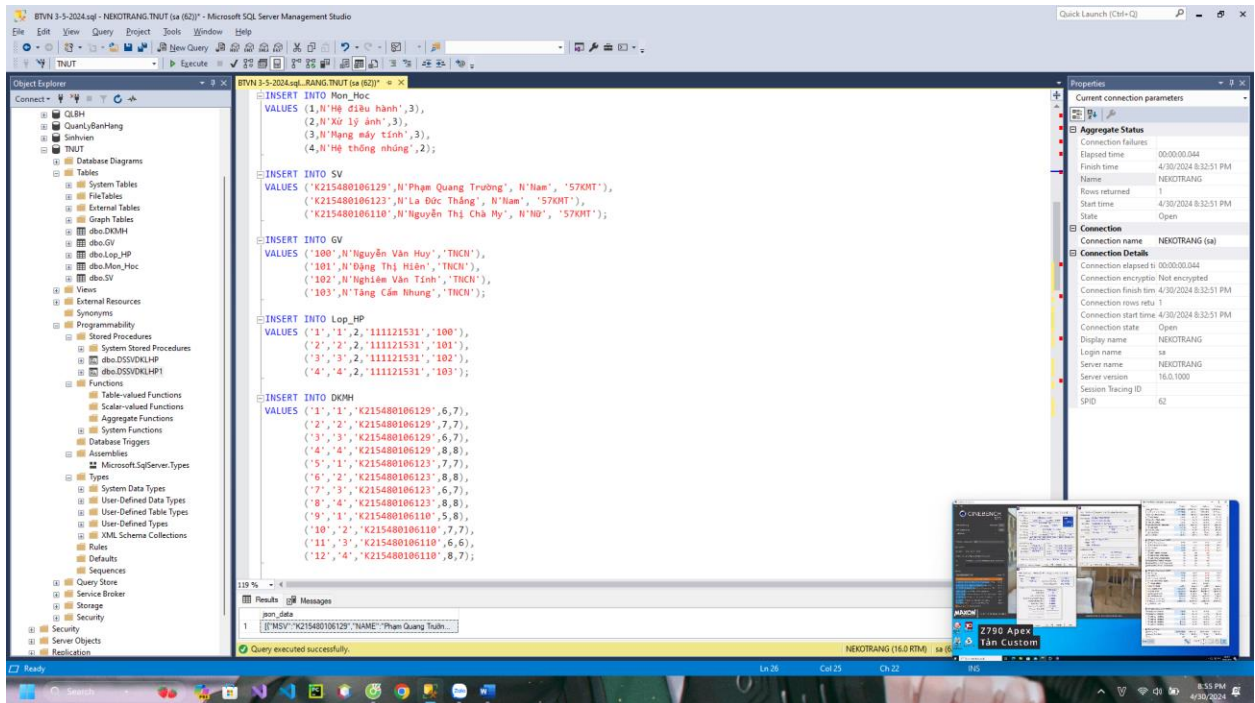
CREATE TABLE Mon_Hoc (
    ID_MH VARCHAR(50) PRIMARY KEY,
    Ten_MH NVARCHAR(50) not null,
    STC INT not null CHECK (STC > 0)
);

CREATE TABLE SV (
    Ma_SV VARCHAR(50) PRIMARY KEY,
    Ten_SV NVARCHAR(50) not null,
    Sex NVARCHAR(50) not null,
    Lop_SV NVARCHAR(50) not null
);

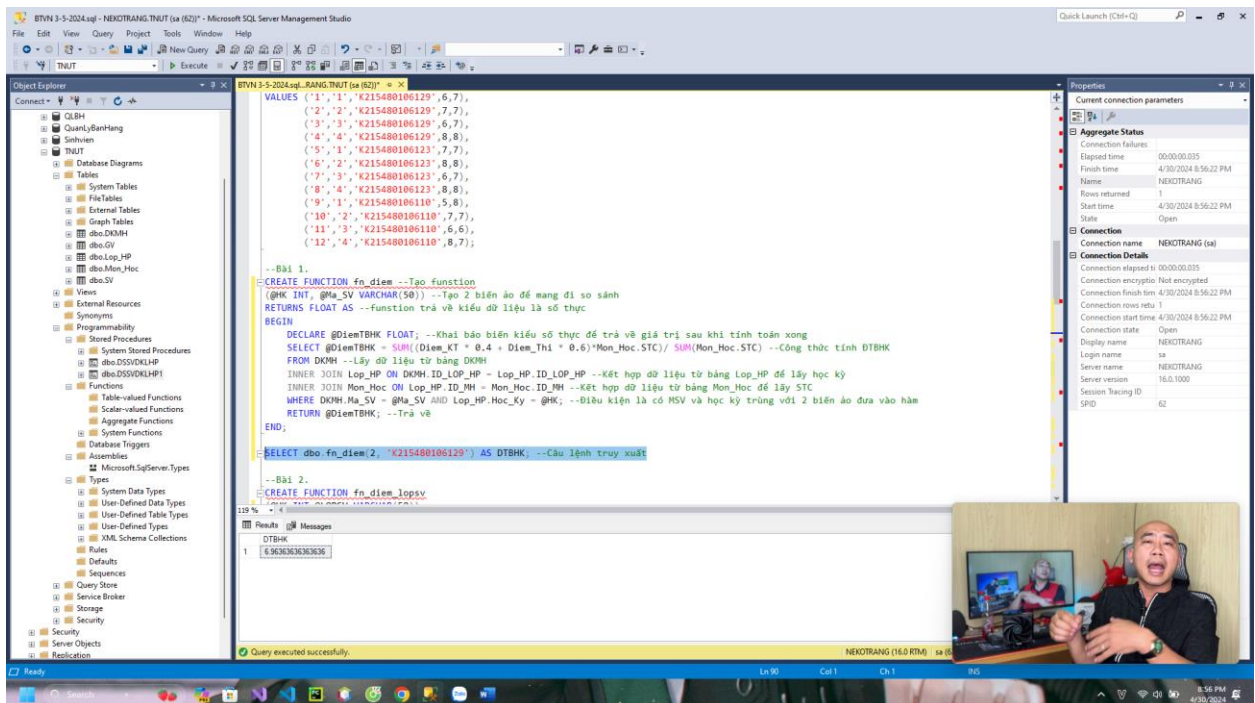
CREATE TABLE GV (
    ID_GV VARCHAR(50) PRIMARY KEY,
    Ten_GV NVARCHAR(50) not null,
    Bo_Mon NVARCHAR(50) not null
);

CREATE TABLE Lop_HP (
    ID_Lop_HP VARCHAR(50) PRIMARY KEY,
    ID_MH VARCHAR(50),
    CONSTRAINT fk_ID_MH FOREIGN KEY (ID_MH) REFERENCES Mon_Hoc(ID_MH),
    Hoc_Ky INT not null,
    Ten_Lop_HP NVARCHAR(50) not null,
    ID_GV VARCHAR(50) FOREIGN KEY (ID_GV) REFERENCES GV(ID_GV)
);

CREATE TABLE DKMH (
    ID VARCHAR(50) PRIMARY KEY,
    ID_Lop_HP VARCHAR(50) FOREIGN KEY (ID_Lop_HP) REFERENCES Lop_HP(ID_Lop_HP),
    Ma_SV VARCHAR(50) FOREIGN KEY (Ma_SV) REFERENCES SV(Ma_SV),
    Diem_KT FLOAT NOT NULL CHECK (Diem_KT >= 0 AND Diem_KT <= 10),
    Diem_Thi FLOAT NOT NULL CHECK (Diem_Thi >= 0 AND Diem_Thi <= 10)
);
```



- Bài 1:



Ý chính:

- Đưa 2 biến đầu vào @HK và @Ma_SV để dùng vào WHERE để tìm kiếm dữ liệu giống. Dùng dữ liệu của bảng DKMH kết hợp với

- Tính tổng của (Điểm KT*0,4 + Điểm thi * 0.6)* số tín chỉ. Cứ gộp dữ liệu phù hợp với điều kiện where thì cộng cả cụm trên vào với nhau. Sau đó sẽ chia cho tổng tín chỉ để ra điểm trung bình học kì.

The screenshot displays the Microsoft SQL Server Enterprise Edition interface. The main window shows a T-SQL script being executed. The script defines a function to calculate a score based on various database metrics and then uses it to calculate a score for a specific database instance.

Script Content:

```
FROM DDMH --Lấy dữ liệu từ bảng DDMH
INNER JOIN Lop_HP ON DDMH.ID_LOP_HP = Lop_HP.ID_LOP_HP --Kết hợp dữ liệu từ bảng Lop_HP để lấy học kỳ
INNER JOIN Mon_Hoc ON Lop_HP.ID_MH = Mon_Hoc.ID_MH --Kết hợp dữ liệu từ bảng Mon_Hoc để lấy STC
WHERE DDMH.Ma_SV = @Ma_SV AND Lop_HP.Hoc_Ky = @HK; --Điều kiện là có MSV và học kỳ trùng với 2 biến ảo đưa vào hàm
RETURN @DienTrungBinh; --Trả về
END;

--BÀI 2.
CREATE FUNCTION fn_diem_lopsv
(@HK INT, @LOPSV VARCHAR(50))
RETURNS TABLE AS --Hàm trả về
RETURN
(
    SELECT
        SV.Ten_SV,
        SV.Sex,
        SV.Ma_SV,
        SUM((DDMH.Diem_KT * 0.4 + DDMH.Diem_Thi * 0.6) * Mon_Hoc.STC) / SUM(Mon_Hoc.STC) AS DiemTrungBinh --Công thức tính DTBK
    FROM SV
    INNER JOIN DDMH ON SV.Ma_SV = DDMH.Ma_SV --Kết hợp dữ liệu từ bảng DDMH để lấy điểm
    INNER JOIN Lop_HP ON SV.Ma_SV = Lop_HP.ID_LOP_HP --Kết hợp dữ liệu từ bảng Lop_HP để lấy học kỳ
    INNER JOIN Mon_Hoc ON Lop_HP.ID_MH = Mon_Hoc.ID_MH --Kết hợp dữ liệu từ bảng Mon_Hoc để lấy số tín chỉ
    WHERE SV.Ten_SV = @LOPSV AND Lop_HP.Hoc_Ky = @HK --Điều kiện
    GROUP BY SV.Ten_SV, SV.Sex, SV.Ma_SV --Nhóm dữ liệu và để dùng các hàm tính toán
);

--BÀI 4.
SELECT * FROM fn_diem_lopsv(2, 'SYKHT');
```

Results:

| Ten_SV | Sex | Ma_SV | DiemTrungBinh |
|--------------------|-----|---------------|------------------|
| Nguyễn Thị Châu Mỹ | Nữ | K215480106110 | 6.74545454545454 |
| Là Đức Thắng | Nam | K215480106123 | 7.34545454545454 |
| Phạm Quang Trường | Nam | K215480106129 | 6.96363636363636 |

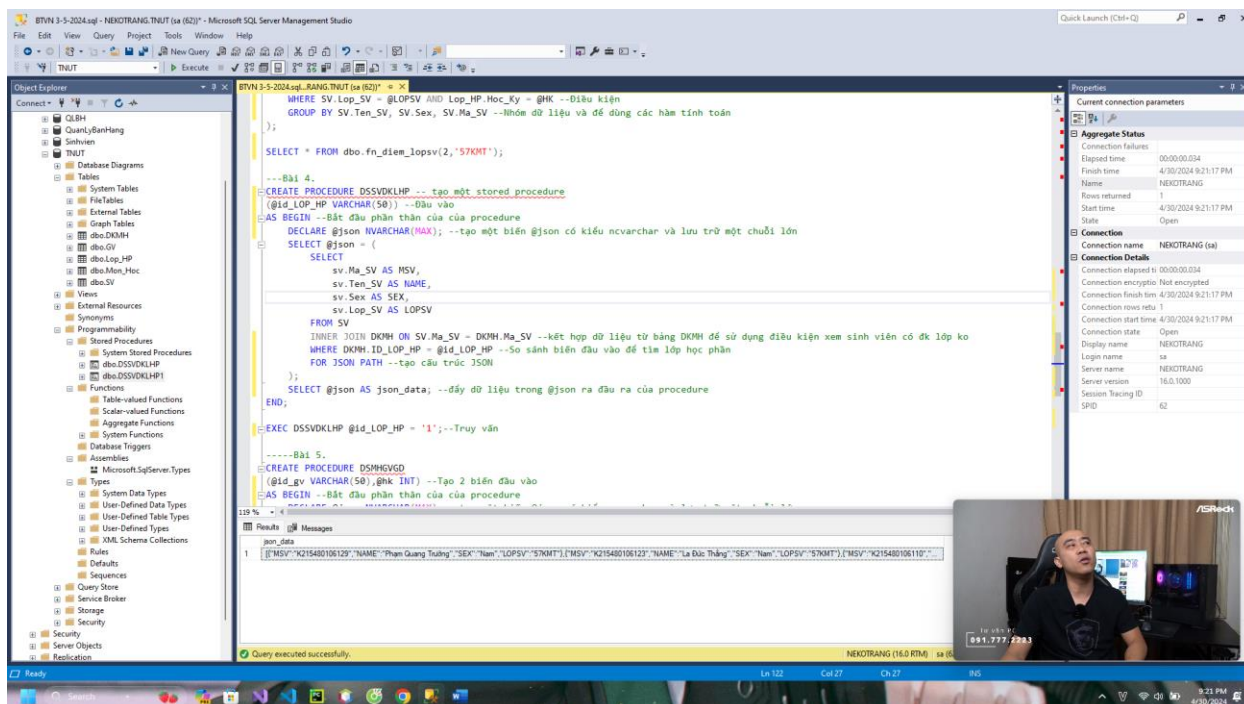
Properties Panel:

- Aggregate Status:** Connection failures: 00:00:00.039, Elapsed time: 4/30/2024 9:06:19 PM, Name: NEXOTRANG, Rows returned: 3, Start time: 4/30/2024 9:06:19 PM, State: Open.
- Connection:** Connection name: NEXOTRANG (sa).
- Connection Details:** Connection elapsed time: 00:00:00.039, Connection encrypted: Not encrypted, Connection finish time: 4/30/2024 9:06:19 PM, Connection rowset info: 3, Connection start time: 4/30/2024 9:06:19 PM, Connection state: Open, Display name: NEXOTRANG, Login name: sa, Server name: NEXOTRANG, Server version: 16.0.1000, Session Tracking ID: SPID: 62.

Taskbar: The taskbar at the bottom shows the Windows taskbar with various application icons and the system clock displaying 9:08 PM on 4/30/2024.

- Sử dụng công thức giống ở bài 1 và in ra danh sách sinh viên học trong lớp học phần đó. Gần giống với bài 1 nhưng giờ in ra thêm tên, giới tính và mã sinh viên.
- Dùng 2 biến đầu vào dùng để so sánh xem sinh viên nào phù hợp với điều kiện để in ra.

Bài 4:

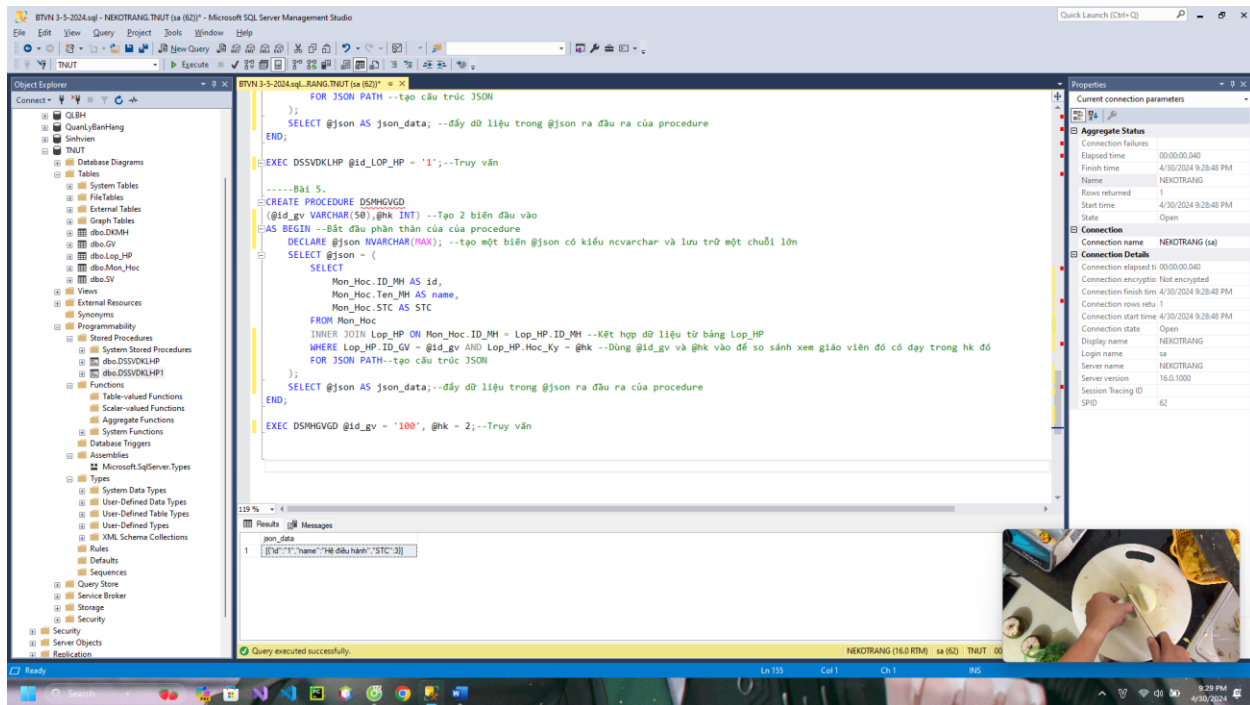


Ý chính:

1. Tạo stored procedure có tên là DSSVDKLHP (viết tắt của "Danh sách sinh viên đăng ký lớp học phần").
2. Tham số đầu vào: @id_LOP_HP đại diện cho ID của lớp học phần để truy vấn thông tin sinh viên.
3. Phần thân:
 - Một biến @json kiểu NVARCHAR(MAX) được khai báo để lưu trữ kết quả truy vấn dưới dạng chuỗi JSON.
 - Một truy vấn SELECT được thực hiện để chọn ra các trường dữ liệu của sinh viên từ bảng SV và kết hợp nó với bảng DKMH thông qua điều kiện rằng Ma_SV trong cả hai bảng phải khớp nhau. Điều kiện này xác định xem sinh viên có đăng ký vào lớp học phần có ID là @id_LOP_HP không.
 - Kết quả của truy vấn được chuyển đổi thành cấu trúc JSON bằng cú pháp FOR JSON PATH.
 - Kết quả JSON này được gán cho biến @json

4. Kết quả trả về: Kết quả JSON được lưu trữ trong biến @json và được trả về thông qua một câu lệnh SELECT cuối cùng với tên cột là json_data.

Bài 5:



Ý chính:

- Tham số đầu vào: @id_gv, @hk.
- SELECT được thực hiện để chọn ra các trường dữ liệu của môn học từ bảng Mon_Hoc và kết hợp nó với bảng Lop_HP thông qua điều kiện rằng ID_MH trong cả hai bảng phải khớp nhau.
- WHERE được sử dụng để lọc ra các bản ghi trong Lop_HP mà có ID_GV trùng khớp với giá trị của tham số @id_gv và Hoc_Ky trùng khớp với giá trị của tham số @hk.
- Những phần còn lại tương đương với bài 4.