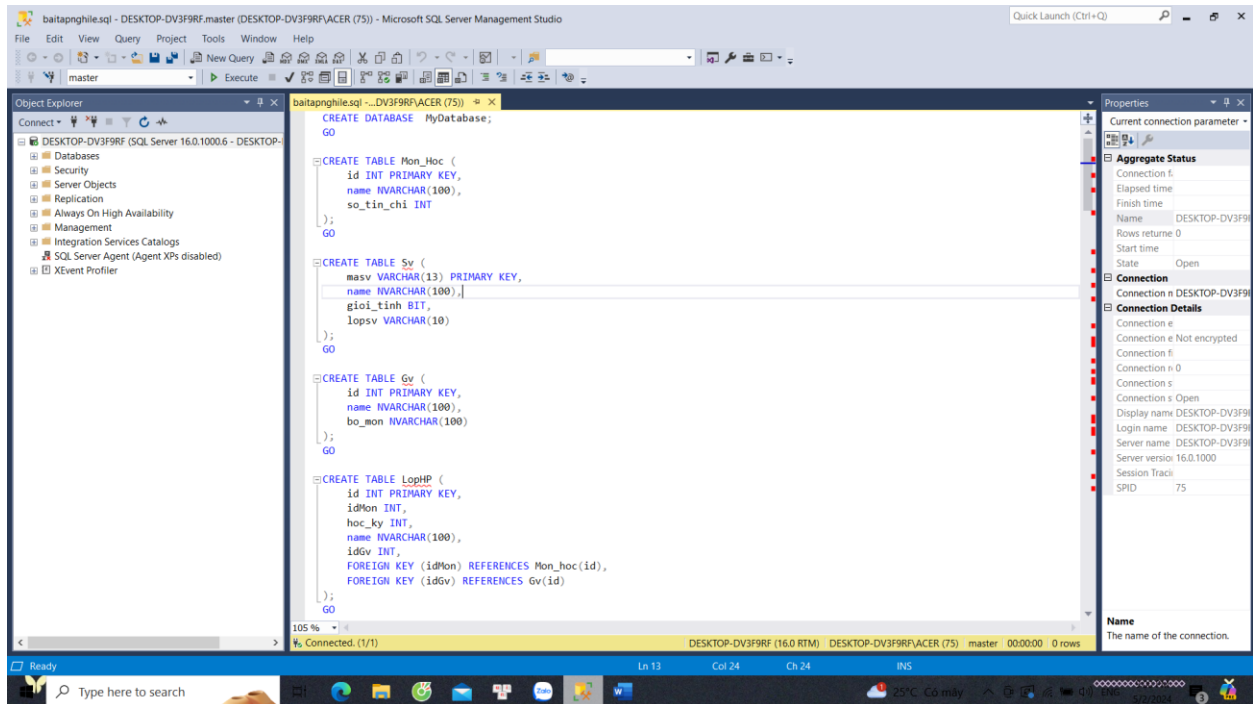
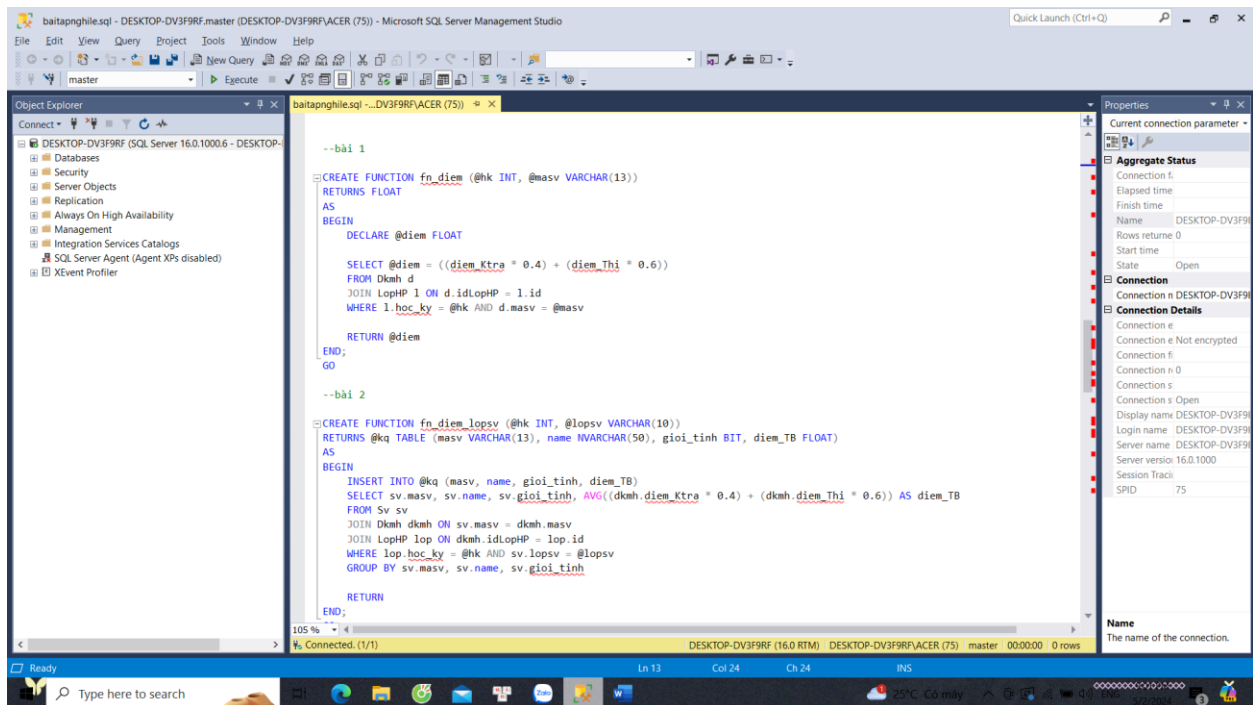


Giải thích code :



Tạo Database với table:



Bài 1:

Đoạn mã SQL trên tạo một hàm có tên là **fn_diem**, nhận hai tham số đầu vào là **@hk** (học kỳ) và **@masv** (mã số sinh viên), và trả về điểm trung bình học kỳ của một sinh viên cụ thể:

- **CREATE FUNCTION fn_diem (@hk INT, @masv VARCHAR(13)) RETURNS FLOAT AS:** Dòng này khai báo việc tạo một hàm có tên là **fn_diem** nhận hai tham số là **@hk** và **@masv**, và trả về một giá trị kiểu **FLOAT**.
- **BEGIN:** Bắt đầu phần thân của hàm.
- **DECLARE @diem FLOAT;** Khai báo một biến tạm thời **@diem** để lưu trữ kết quả tính toán.
- **SELECT @diem = AVG((d.diem_Ktra * 0.4) + (d.diem_Thi * 0.6)) FROM Dkmh d JOIN LopHP l ON d.idLopHP = l.id WHERE l.hoc_ky = @hk AND d.masv = @masv;** Truy vấn này tính toán điểm trung bình của sinh viên với mã số **@masv** trong học kỳ **@hk**. Nó kết hợp các điểm kiểm tra (**diem_Ktra**) và điểm thi (**diem_Thi**) với trọng số 40% và 60% tương ứng, sau đó tính trung bình của chúng.
- **RETURN @diem;** Trả về giá trị của biến **@diem**, tức là điểm trung bình đã tính toán.
- **END;** Kết thúc phần thân của hàm.

Bài 2:

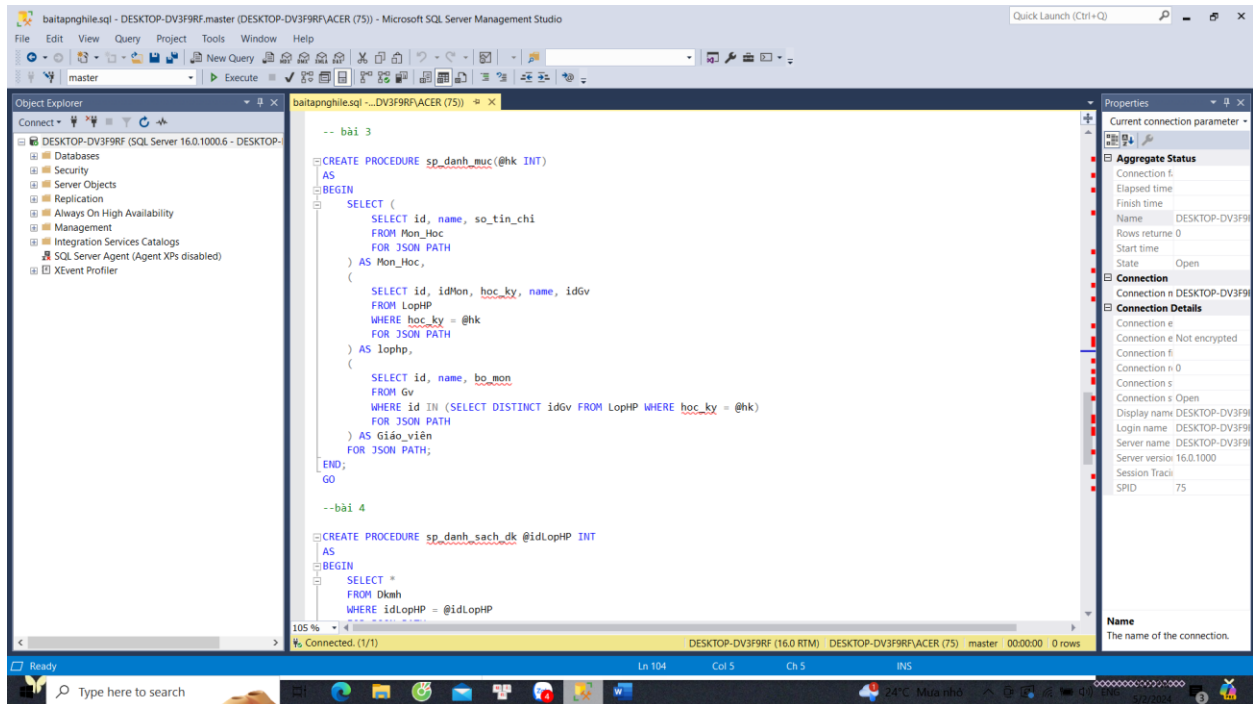
Đoạn mã SQL trên tạo một hàm có tên là **fn_diem_lopsv**, nhận hai tham số đầu vào là **@hk** (học kỳ) và **@lopszv** (tên lớp sinh viên), và trả về một bảng kết quả bao gồm các cột **masv** (mã số sinh viên), **name** (tên sinh viên), **gioi_tinh** (giới tính của sinh viên), và **diem_TB** (điểm trung bình học kỳ) cho tất cả sinh viên trong lớp đó.

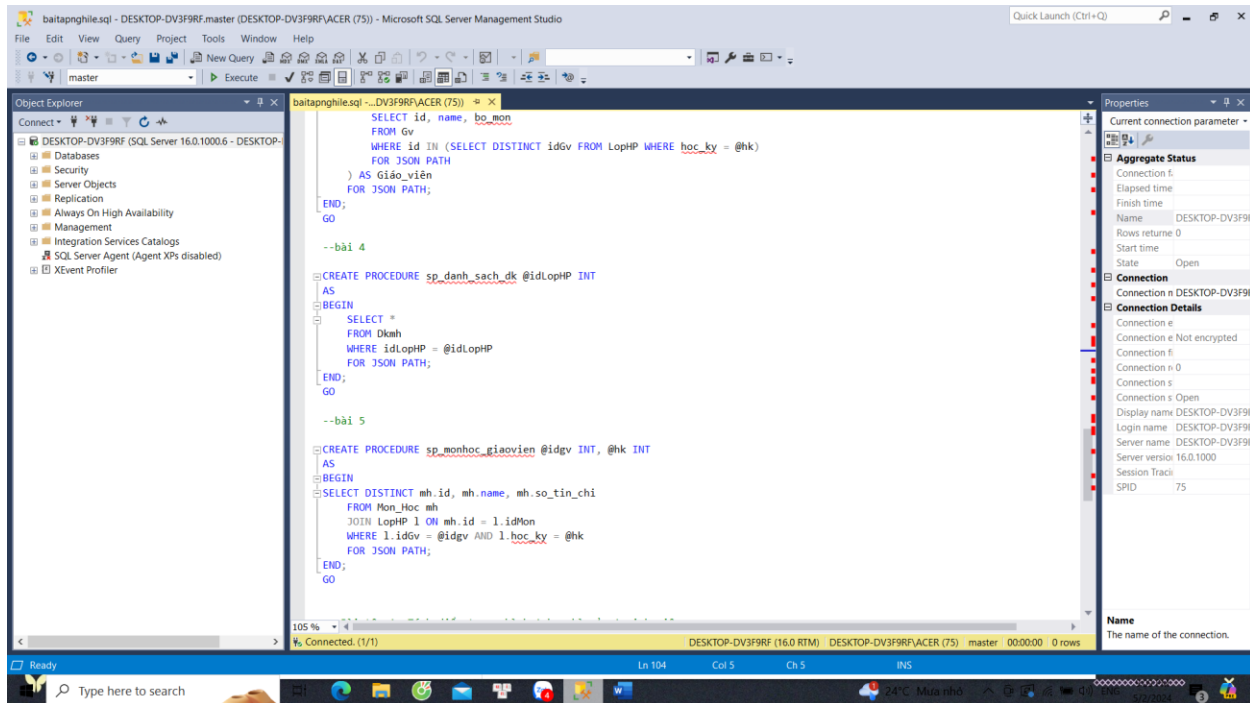
- **CREATE FUNCTION fn_diem_lopszv (@hk INT, @lopszv VARCHAR(10)) RETURNS @kq TABLE (masv VARCHAR(13), name NVARCHAR(50), gioi_tinh BIT, diem_TB FLOAT) AS:** Dòng này khai báo việc tạo một hàm có tên là **fn_diem_lopszv** nhận hai tham số là **@hk** và **@lopszv**, và trả về một bảng kết quả có các cột được chỉ định.
- **BEGIN:** Bắt đầu phần thân của hàm.
- **INSERT INTO @kq (masv, name, gioi_tinh, diem_TB) SELECT ...:** Dòng này chèn dữ liệu vào bảng kết quả **@kq**. Trong truy vấn **SELECT**, nó tính toán điểm trung bình

học kỳ (**diem_TB**) của tất cả sinh viên trong lớp **@lopsz** trong học kỳ **@hk**, và lấy thông tin về mã sinh viên (**masv**), tên (**name**), và giới tính (**gioi_tinh**) của họ. Kết quả được nhóm theo mã sinh viên, tên và giới tính.

- **RETURN**: Kết thúc hàm và trả về bảng kết quả đã được tính toán.

- **END**:: Kết thúc phần thân của hàm.





Bài 4:

Đoạn mã SQL trên tạo một thủ tục có tên là **sp_danh_sach_dk** nhận một tham số là **@idLopHP** (ID của lớp học phần), và trả về một danh sách các bản ghi đăng ký lớp học phần dưới dạng chuỗi JSON.

- **CREATE PROCEDURE sp_danh_sach_dk @idLopHP INT AS:** Dòng này khai báo việc tạo một thủ tục có tên là **sp_danh_sach_dk** nhận một tham số là **@idLopHP**.
- **BEGIN:** Bắt đầu phần thân của thủ tục.
- **SELECT * FROM Dkmh WHERE idLopHP = @idLopHP FOR JSON PATH;:** Dòng này thực hiện truy vấn để lấy danh sách các bản ghi từ bảng **Dkmh** mà có **idLopHP** trùng với giá trị của tham số **@idLopHP**. Kết quả của truy vấn được trả về dưới dạng một chuỗi JSON sử dụng cú pháp FOR JSON PATH.
- **END;:** Kết thúc phần thân của thủ tục.

Bài 5:

Đoạn mã SQL trên tạo một thủ tục có tên là **sp_monhoc_giaovien** nhận hai tham số là **@idgv** (ID của giáo viên) và **@hk** (học kỳ), và trả về một danh sách các môn học của giáo viên đó trong học kỳ nhất định dưới dạng chuỗi JSON.

- **CREATE PROCEDURE** sp_monhoc_giaovien @idgv INT, @hk INT AS: Dòng này khai báo việc tạo một thủ tục có tên là **sp_monhoc_giaovien** nhận hai tham số là **@idgv** và **@hk**.

- **BEGIN**: Bắt đầu phần thân của thủ tục.

- **SELECT DISTINCT mh.id, mh.name, mh.so_tin_chi FROM Mon_Hoc mh JOIN LopHP l ON mh.id = l.idMon WHERE l.idGv = @idgv AND l.hoc_ky = @hk FOR JSON PATH**;: Dòng này thực hiện truy vấn để lấy danh sách các môn học của giáo viên có idgv trùng với giá trị của tham số **@idgv** và trong học kỳ **@hk**. Kết quả của truy vấn được trả về dưới dạng một chuỗi JSON sử dụng cú pháp FOR JSON PATH.

- **END**;: Kết thúc phần thân của thủ tục.