

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

BỘ MÔN: CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

**ĐỀ TÀI: QUẢN LÝ THÔNG TIN ĐỊA ĐIỂM DU
LỊCH**

Giảng viên hướng dẫn	: ThS. ĐỖ DUY CỚP
Họ tên sinh viên	: Vi Tuấn Đạt
MSSV	: K215480106088
Lớp	: K57KMT.01

Thái Nguyên 2024

PHIẾU GIAO ĐỀ TÀI MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

Sinh viên: Vi Tuấn Đạt

MSSV: K215480106088

Lớp: K57KMT

Ngành: Kỹ thuật máy tính

Giáo viên hướng dẫn: ThS. Đỗ Duy Cốp

1. Tên đề tài: **“Quản lý thông tin địa điểm du lịch”**.
2. Yêu cầu của phần mềm, ứng dụng:
 - Tạo cơ sở dữ liệu cho bài toán gồm các bảng và thuộc tính bằng việc phân tích
3. Các sản phẩm, kết quả:
 - Bản báo cáo, và cơ sở dữ liệu.

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

ThS. Đỗ Duy Cốp

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày...tháng.....năm.....

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

Github cá nhân:

https://github.com/TuanDat23/BTL_HQTCSDL_QuanLyThongTinDiaDiemDuLi
ch

QR:



Thông tin cá nhân

- Họ và Tên: Vi Tuấn Đạt
- MSSV: K215480106088
- Lớp: K57KMT
- Môn học: Hệ quản trị cơ sở dữ liệu

ĐỀ TÀI: Quản lý thông tin địa điểm du lịch

Mô tả bài toán quản lý: Quản lý thông tin các địa điểm du lịch, các nhà hàng, khách sạn xung quanh và đánh giá của mọi người. Từ kết quả đánh giá giúp người dùng biết được chất lượng của khu du lịch.

Chương 1. Chức Năng Cơ Bản

1. Quản lý

1.1. Quản Lý Địa Điểm Du Lịch

- Thêm địa điểm du lịch: Cho phép thêm mới một địa điểm du lịch vào cơ sở dữ liệu
- Sửa thông tin địa điểm du lịch: Cho phép chỉnh sửa thông tin của một địa điểm du lịch đã tồn tại
- Xóa địa điểm du lịch: Cho phép xóa một địa điểm du lịch khỏi cơ sở dữ liệu
- Xem danh sách địa điểm du lịch: Hiển thị danh sách tất cả các địa điểm du lịch

1.2. Quản Lý Khách Sạn

- Thêm khách sạn: Cho phép thêm mới một khách sạn vào cơ sở dữ liệu
- Sửa thông tin khách sạn: Cho phép chỉnh sửa thông tin của một khách sạn đã tồn tại
- Xóa khách sạn: Cho phép xóa một khách sạn khỏi cơ sở dữ liệu
- Xem danh sách khách sạn: Hiển thị danh sách tất cả các khách sạn

1.3. Quản Lý Nhà Hàng

- Thêm nhà hàng: Cho phép thêm mới một nhà hàng vào cơ sở dữ liệu
- Sửa thông tin nhà hàng: Cho phép chỉnh sửa thông tin của một nhà hàng đã tồn tại
- Xóa nhà hàng: Cho phép xóa một nhà hàng khỏi cơ sở dữ liệu
- Xem danh sách nhà hàng: Hiển thị danh sách tất cả các nhà hàng

1.4. Quản Lý Đánh Giá

- Thêm đánh giá: Cho phép thêm mới một đánh giá từ khách hàng về địa điểm du lịch
- Sửa thông tin đánh giá: Cho phép chỉnh sửa thông tin của một đánh giá đã tồn tại
- Xóa đánh giá: Cho phép xóa một đánh giá khỏi cơ sở dữ liệu
- Xem danh sách đánh giá: Hiển thị danh sách tất cả các đánh giá

1.5. Quản Lý Khách Hàng

- Thêm khách hàng: Cho phép thêm mới một khách hàng vào cơ sở dữ liệu
- Sửa thông tin khách hàng: Cho phép chỉnh sửa thông tin của một khách hàng đã tồn tại
- Xóa khách hàng: Cho phép xóa một khách hàng khỏi cơ sở dữ liệu.
- Xem danh sách khách hàng: Hiện thị danh sách tất cả các khách hàng

2. Chức Năng cơ bản

- Tìm kiếm địa điểm du lịch theo tên hoặc loại hình du lịch
- Tìm kiếm khách sạn hoặc nhà hàng theo địa điểm du lịch
- Xem chi tiết đánh giá của một địa điểm du lịch cụ thể
- Xem chi tiết thông tin khách hàng

3. Chức Năng Nâng Cao

Cập nhật tổng số sao và số lượng đánh giá: Tự động cập nhật tổng số sao và số lượng đánh giá của một địa điểm du lịch khi có đánh giá mới được thêm vào hoặc khi một đánh giá bị xóa hoặc chỉnh sửa.

Duyệt qua các địa điểm du lịch: Tạo thủ tục lưu trữ để duyệt qua các địa điểm du lịch và thực hiện các tính toán tổng hợp như số lượng khách sạn hoặc nhà hàng tại mỗi địa điểm.

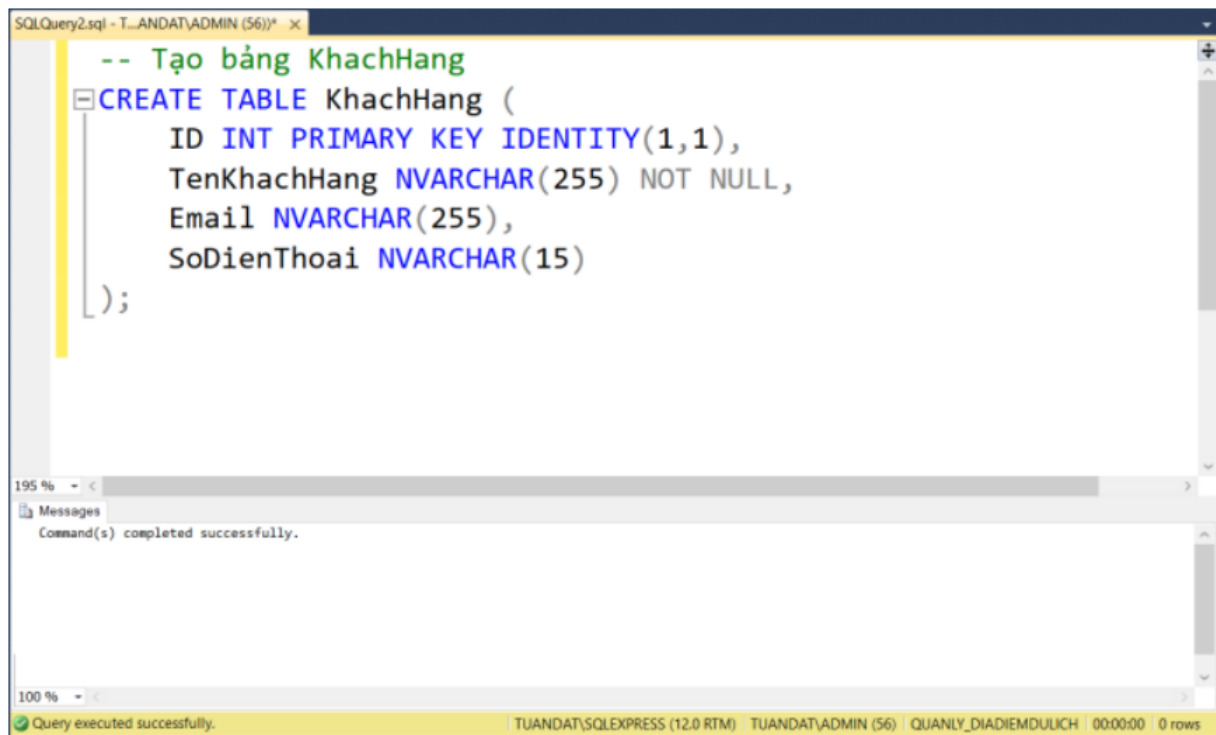
Báo cáo địa điểm du lịch phổ biến: Tạo báo cáo hiện thị các địa điểm du lịch phổ biến dựa trên số lượng đánh giá và điểm số trung bình.

Chương 2. Thiết kế chương trình trong SQL

1. Tạo các bảng

Bảng KháchHang (Phải tạo đầu tiên để các bảng khác tham chiếu tới)

- ID🔑: Khóa chính được sử dụng để xác định mỗi khách hàng một cách duy nhất, tự tăng.
- TenKhachHang: Tên khách hàng được đặt là NOT NULL để đảm bảo mỗi khách hàng được lưu trữ đều có thông tin tên. Điều này cực kỳ quan trọng để có thể phân biệt và quản lý các khách hàng.
- Email: Email của khách hàng.
- SoDienThoai: Số điện thoại của khách hàng.



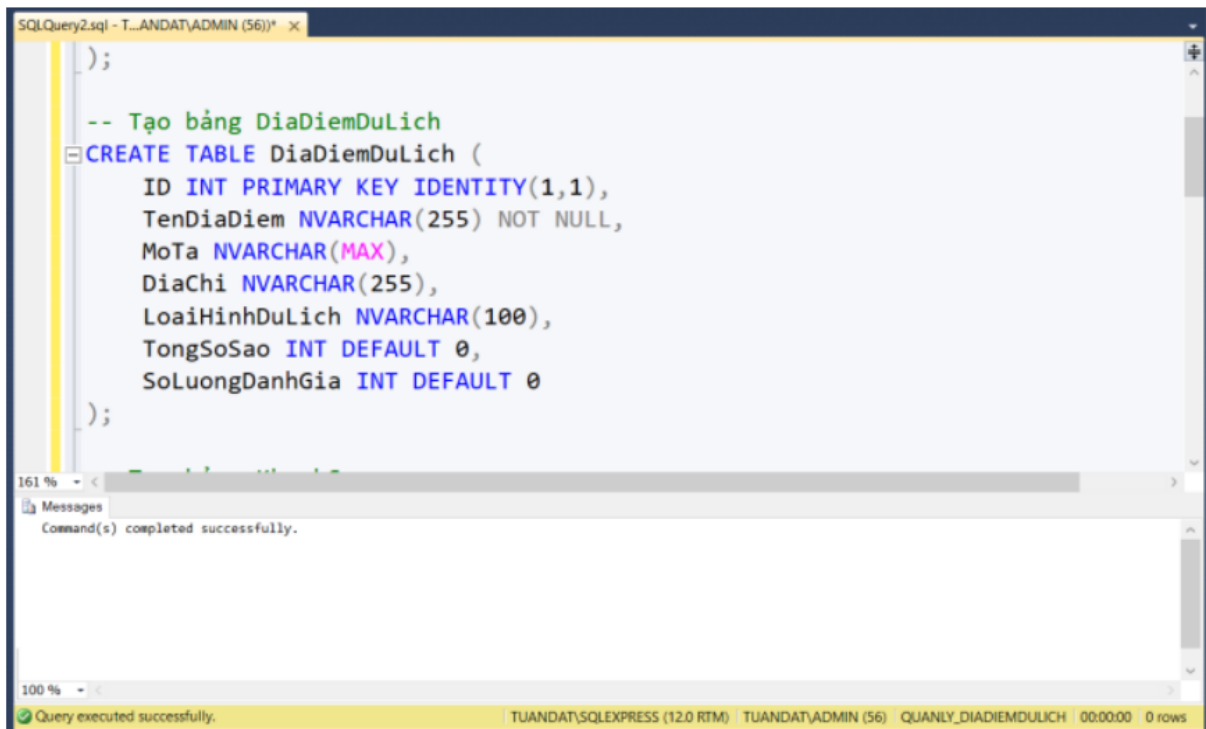
```
-- Tạo bảng KháchHang
CREATE TABLE KháchHang (
    ID INT PRIMARY KEY IDENTITY(1,1),
    TenKhachHang NVARCHAR(255) NOT NULL,
    Email NVARCHAR(255),
    SoDienThoai NVARCHAR(15)
);
```

Messages
Command(s) completed successfully.

Query executed successfully. TUANDAT\SQLEXPRESS (12.0 RTM) TUANDAT\ADMIN (56) QUANLY_DIADIEMDULICH 00:00:00 0 rows

Bảng DiaDiemDuLich

- ID: Khóa chính, tự tăng.
- TenDiaDiem: Tên địa điểm du lịch là thông tin cần thiết và được đánh dấu là NOT NULL để đảm bảo tính chính xác và đầy đủ của dữ liệu.
- MoTa: Mô tả về địa điểm.
- DiaChi: Địa chỉ địa điểm.
- LoaiHinhDuLich: Loại hình du lịch (nghỉ dưỡng, tham quan, ...).



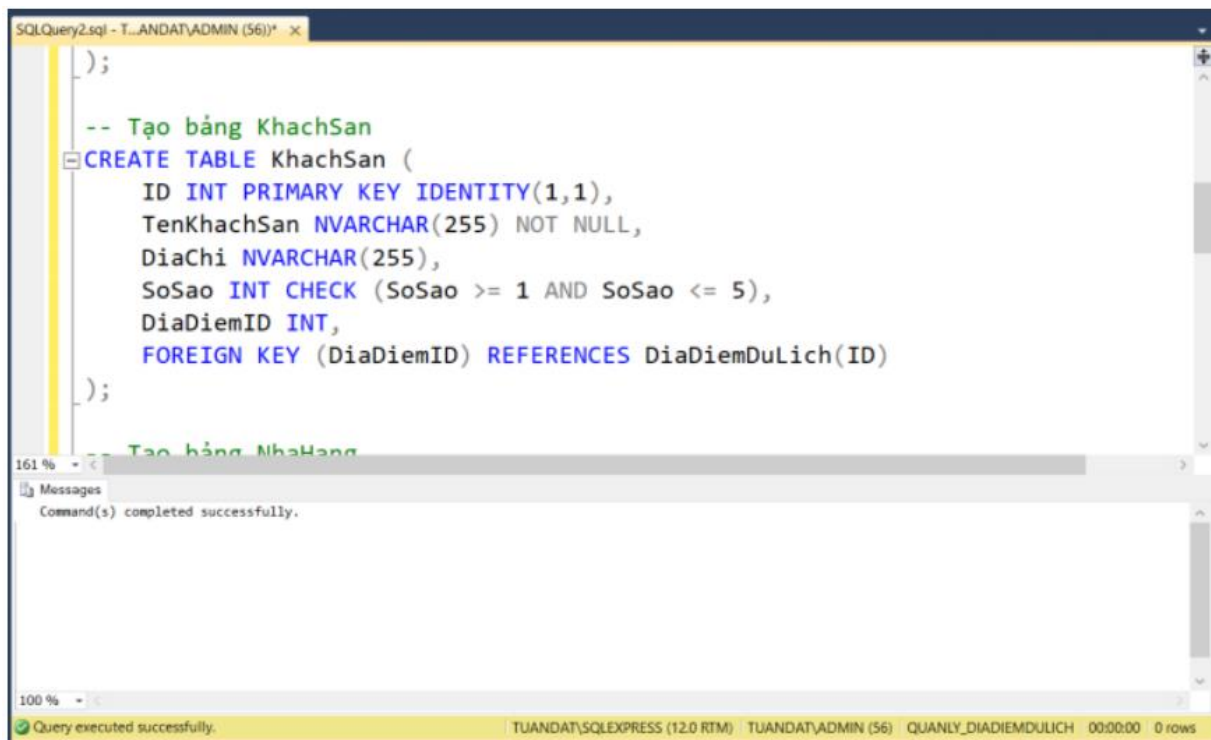
The screenshot shows a SQL Server query window with the following T-SQL code:

```
-- Tạo bảng DiaDiemDuLich
CREATE TABLE DiaDiemDuLich (
    ID INT PRIMARY KEY IDENTITY(1,1),
    TenDiaDiem NVARCHAR(255) NOT NULL,
    MoTa NVARCHAR(MAX),
    DiaChi NVARCHAR(255),
    LoaiHinhDuLich NVARCHAR(100),
    TongSoSao INT DEFAULT 0,
    SoLuongDanhGia INT DEFAULT 0
);
```

Below the query window, the Messages pane displays: "Command(s) completed successfully." The status bar at the bottom indicates: "Query executed successfully. TUANDAT\SQLEXPRESS (12.0 RTM) TUANDAT\ADMIN (56) QUANLY_DIADIEMDULICH 00:00:00 0 rows".

Bảng KháchSan

- ID: Khóa chính, tự tăng.
- TenKhachSan: Tên khách sạn được đánh dấu là NOT NULL để đảm bảo tính duy nhất của dữ liệu.
- DiaChi: Địa chỉ khách sạn.
- SoSao: Số sao của khách sạn.
- DiaDiemID: Là khóa ngoại (FOREIGN KEY) tham chiếu đến ID của bảng DiaDiemDuLich. Điều này thiết lập mối quan hệ giữa các bảng và cho phép truy cập thông tin về địa điểm du lịch liên quan tới từng khách sạn và nhà hàng.

The screenshot shows a SQL Server Enterprise Manager window titled 'SQLQuery2.sql - T...ANDAT\ADMIN (56)'. The main pane displays a SQL script for creating a table named 'KhachSan'. The script includes a comment '-- Tạo bảng KhachSan' followed by a 'CREATE TABLE' statement. The table has columns: 'ID' (INT, PRIMARY KEY, IDENTITY(1,1)), 'TenKhachSan' (NVARCHAR(255), NOT NULL), 'DiaChi' (NVARCHAR(255)), 'SoSao' (INT, CHECK constraint: SoSao >= 1 AND SoSao <= 5), and 'DiaDiemID' (INT, FOREIGN KEY referencing DiaDiemDuLich(ID)). The script ends with a semicolon. Below the script, a 'Messages' pane shows the message 'Command(s) completed successfully.' The status bar at the bottom indicates 'Query executed successfully.' and provides details about the server (TUANDAT\SQLEXPRESS (12.0 RTM)), user (TUANDAT\ADMIN (56)), database (QUANLY_DIADIEMDULICH), and execution time (00:00:00) with 0 rows affected.

```
-- Tạo bảng KhachSan
CREATE TABLE KhachSan (
    ID INT PRIMARY KEY IDENTITY(1,1),
    TenKhachSan NVARCHAR(255) NOT NULL,
    DiaChi NVARCHAR(255),
    SoSao INT CHECK (SoSao >= 1 AND SoSao <= 5),
    DiaDiemID INT,
    FOREIGN KEY (DiaDiemID) REFERENCES DiaDiemDuLich(ID)
);

-- Tạo bảng NhaHang
```

Messages
Command(s) completed successfully.

100 %
Query executed successfully. TUANDAT\SQLEXPRESS (12.0 RTM) TUANDAT\ADMIN (56) QUANLY_DIADIEMDULICH 00:00:00 0 rows

Bảng NhaHang

- ID: Khóa chính, tự tăng.
- TenNhaHang: Tên nhà hàng được đánh dấu là NOT NULL để đảm bảo tính duy nhất của dữ liệu.
- DiaChi: Địa chỉ nhà hàng.
- LoaiHinhAmThuc: Loại hình ẩm thực (Âu, Á, Hải sản, ...).
- DiaDiemID: Là khóa ngoại (FOREIGN KEY) tham chiếu đến ID của bảng DiaDiemDuLich. Điều này thiết lập mối quan hệ giữa các bảng và cho phép truy cập thông tin về địa điểm du lịch liên quan tới từng khách sạn và nhà hàng.

```
FOREIGN KEY (DiaDiemID) REFERENCES DiaDiemDuLich(ID)
);

-- Tạo bảng NhaHang
CREATE TABLE NhaHang (
    ID INT PRIMARY KEY IDENTITY(1,1),
    TenNhaHang NVARCHAR(255) NOT NULL,
    DiaChi NVARCHAR(255),
    LoaiHinhAmThuc NVARCHAR(100),
    DiaDiemID INT,
    FOREIGN KEY (DiaDiemID) REFERENCES DiaDiemDuLich(ID)
);
```

Messages
Command(s) completed successfully.

100 %
Query executed successfully. TUANDAT\SQLEXPRESS (12.0 RTM) TUANDAT\ADMIN (56) QUANLY_DIADIEMDULICH 00:00:00 0 rows

Bảng DanhGia

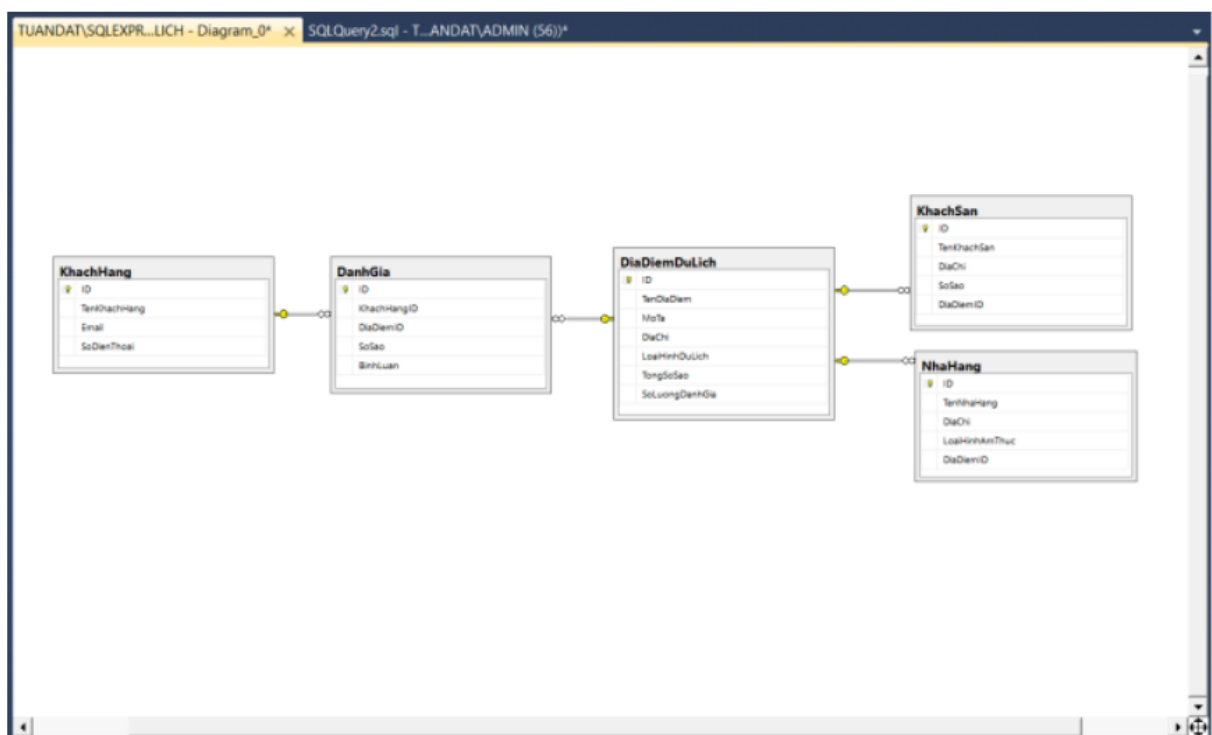
- ID: Khóa chính, tự tăng.
- KháchHangID: Là khóa ngoại (FOREIGN KEY) tham chiếu đến ID của bảng KháchHang. Điều này thiết lập mối quan hệ giữa bảng DanhGia.
- DiaDiemID: Là khóa ngoại (FOREIGN KEY) tham chiếu đến ID của bảng DiaDiemDuLich. Điều này thiết lập mối quan hệ giữa các bảng và cho phép truy cập thông tin về địa điểm du lịch liên quan tới từng khách sạn và nhà hàng.
- SoSao: Số sao đánh giá.
- BinhLuan: Bình luận của khách hàng.

```
SQLQuery2.sql - T...ANDAT\ADMIN (56)) * x
-- Tạo bảng DanhGia
CREATE TABLE DanhGia (
    ID INT PRIMARY KEY IDENTITY(1,1),
    KháchHangID INT,
    DiaDiemID INT,
    SoSao INT CHECK (SoSao >= 1 AND SoSao <= 5),
    BinhLuan NVARCHAR(MAX),
    FOREIGN KEY (KháchHangID) REFERENCES KháchHang(ID),
    FOREIGN KEY (DiaDiemID) REFERENCES DiaDiemDuLich(ID)
);

161 %
Messages
Command(s) completed successfully.

100 %
Query executed successfully. TUANDAT\SQLEXPRESS (12.0 RTM) TUANDAT\ADMIN (56) QUANLY_DIADIEMDULICH 00:00:00 0 rows
```

Sơ đồ thực thể liên kết



2. Thêm dữ liệu vào các bảng

-- Thêm dữ liệu vào bảng DiaDiemDuLich

```
INSERT INTO DiaDiemDuLich (TenDiaDiem, MoTa, DiaChi, LoaiHinhDuLich) VALUES
```

(N'Đà Lạt', N'Thành phố ngàn hoa', N'Lâm Đồng', N'Nghỉ dưỡng'),

(N'Hạ Long', N'Vịnh biển đẹp nhất', N'Quảng Ninh', N'Tham quan');
-- Thêm dữ liệu vào bảng KháchSan

INSERT INTO KháchSan (TenKhachSan, DiaChi, SoSao, DiaDiemID)
VALUES

(N'Khách sạn Đà Lạt Palace', N'Số 12, Đường Trần Phú', 5, 1),

(N'Khách sạn Mường Thanh', N'Số 1, Đường Hạ Long', 4, 2);
-- Thêm dữ liệu vào bảng NhaHang

INSERT INTO NhaHang (TenNhaHang, DiaChi, LoaiHinhAmThuc,
DiaDiemID) VALUES

(N'Nhà hàng Hoa Đà Lạt', N'Số 15, Đường Phan Bội Châu', N'Ấu', 1),

(N'Nhà hàng Biển Vàng', N'Số 3, Đường Bãi Cháy', N'Hải sản', 2);
-- Thêm dữ liệu vào bảng KháchHang

INSERT INTO KháchHang (TenKhachHang, Email, SoDienThoai) VALUES

(N'Nguyễn Văn A', N'vana@gmail.com', N'0901234567'),

(N'Trần Thị B', N'thib@gmail.com', N'0912345678');
-- Thêm dữ liệu vào bảng DanhGia

INSERT INTO DanhGia (KhachHangID, DiaDiemID, SoSao, BinhLuan)
VALUES

(1, 1, 5, N'Rất đẹp và thơ mộng!'),

(2, 2, 4, N'Cảnh đẹp nhưng dịch vụ chưa tốt.');

Chương 3. Thiết lập chức năng

1. Chức năng cơ bản

1.1. Chức năng tìm kiếm thông tin

* Lấy danh sách các địa điểm du lịch:

```
SELECT * FROM DiaDiemDuLich;
```

* Lấy danh sách các khách sạn:

```
SELECT * FROM KhachSan WHERE DiaDiemID = (id của địa điểm cần tìm);
```

* Lấy danh sách các nhà hàng

```
SELECT * FROM NhaHang WHERE DiaDiemID = (id của địa điểm cần tìm);
```

* Lấy danh sách đánh giá cho 1 địa điểm:

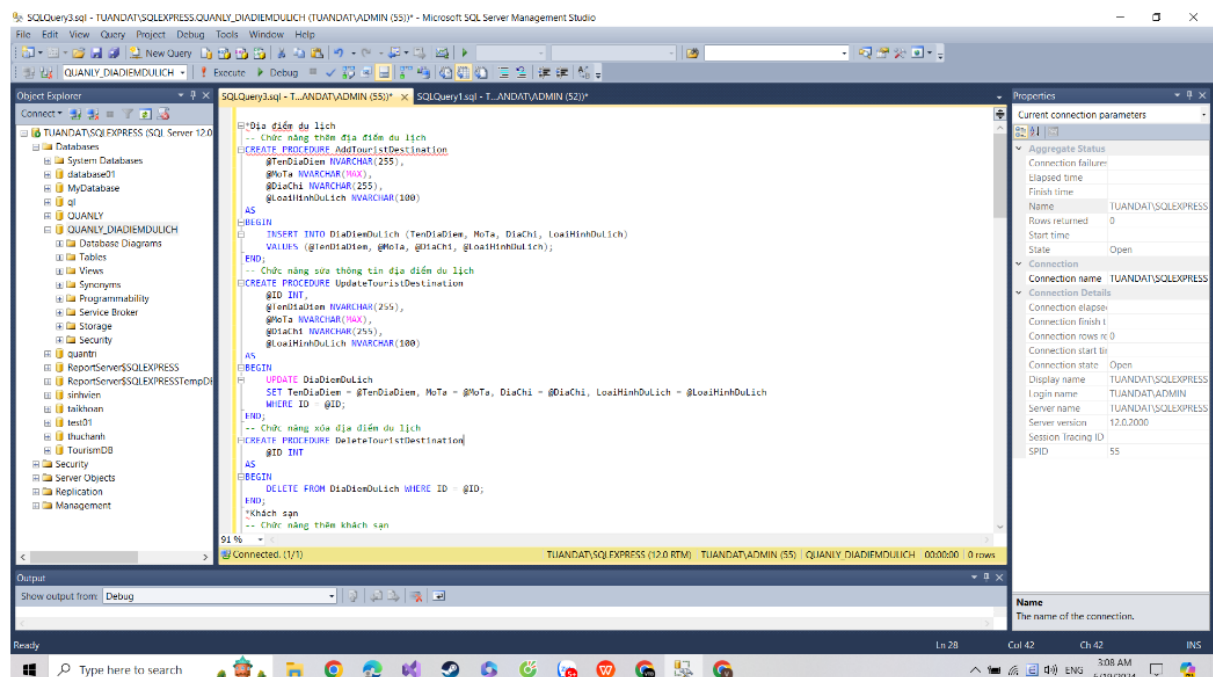
```
SELECT * FROM DanhGia WHERE DiaDiemID = (id của địa điểm cần tìm);
```

* Lấy thông tin chi tiết của 1 khách hàng:

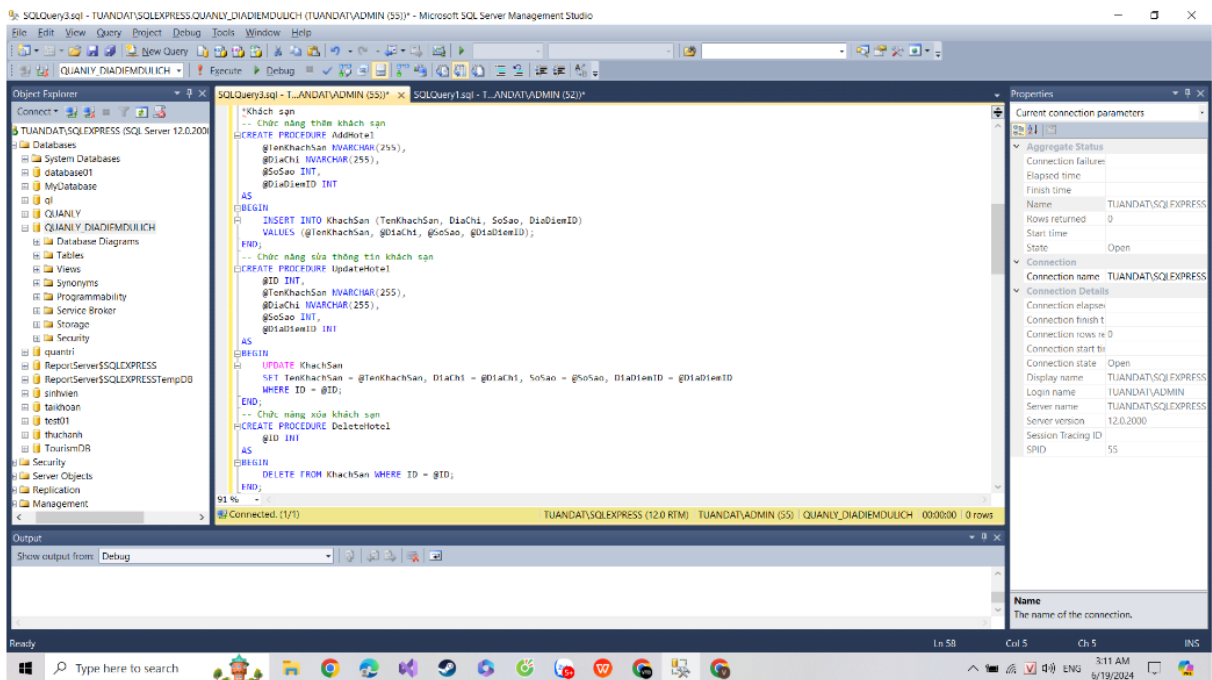
```
SELECT * FROM KhachHang WHERE ID = (id của khách hàng);
```

1.2. Chức năng thêm, xóa, sửa

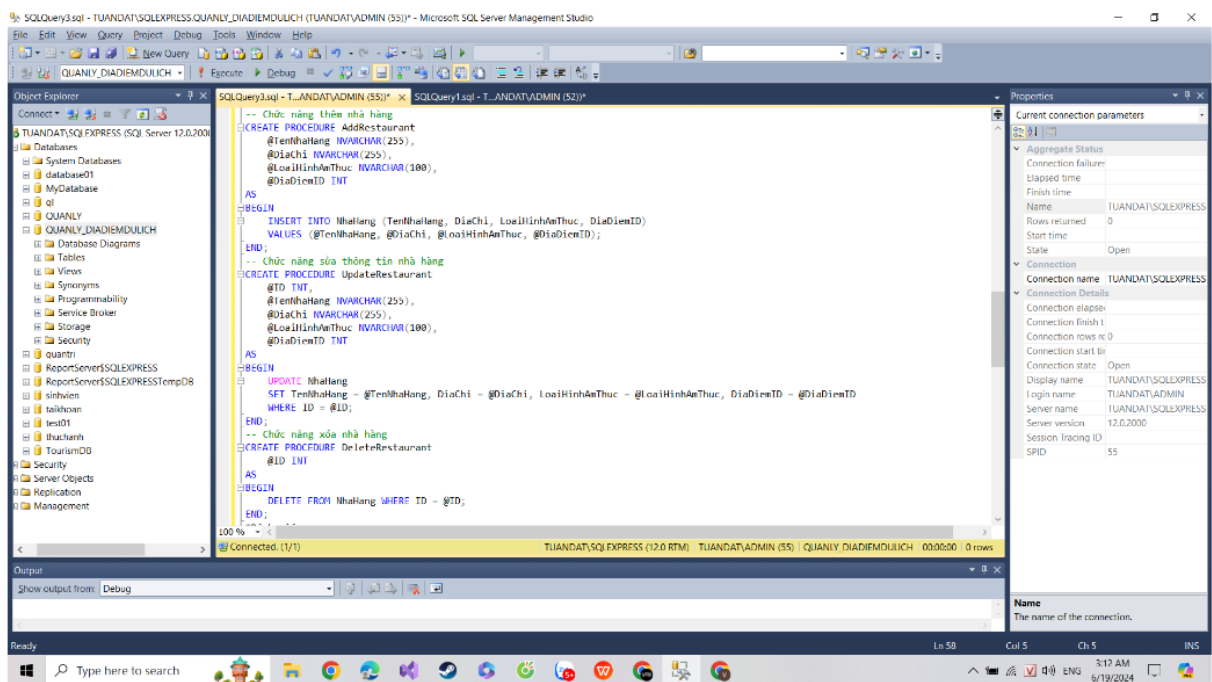
- Địa điểm du lịch



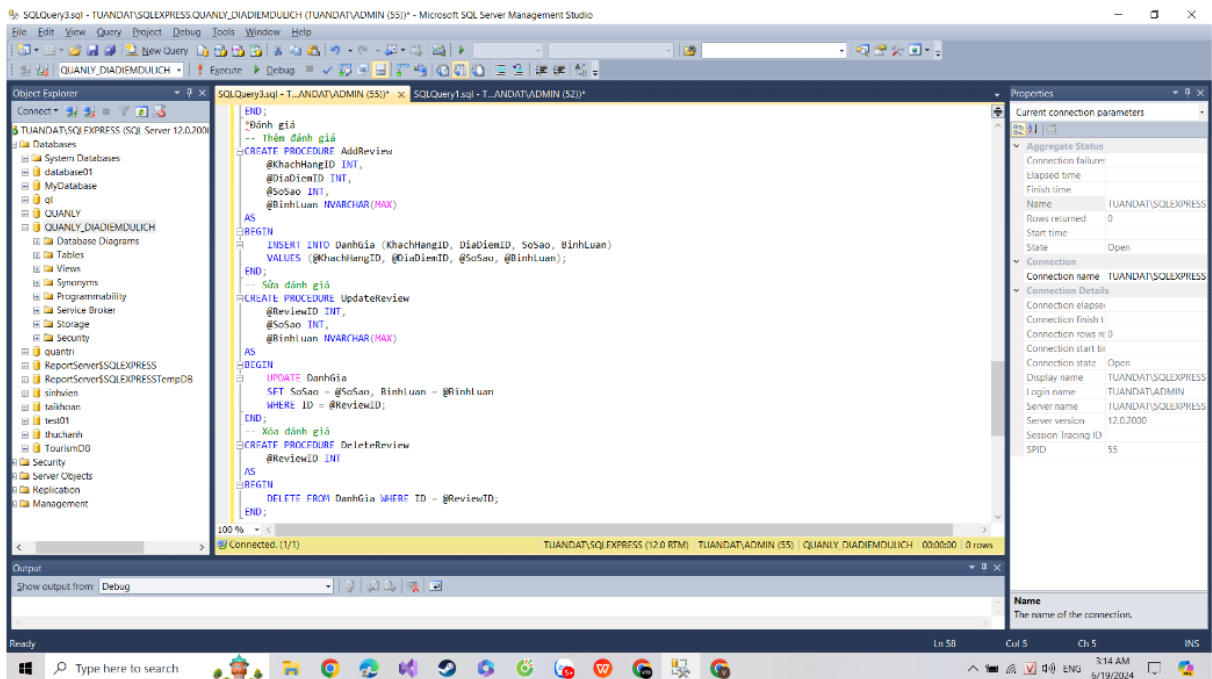
- Khách sạn



- Nhà hàng



- Đánh giá



2.Chức năng nâng cao

- Tự động cập nhật tổng số sao và số lượng đánh giá của một địa điểm du lịch khi có đánh giá mới

CREATE TRIGGER trg_UpdateReviewStats

ON DanhGia

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

-- Xử lý cho trường hợp thêm hoặc cập nhật đánh

IF EXISTS (SELECT * FROM inserted)

BEGIN

DECLARE @DiaDiemID INT;

SELECT @DiaDiemID = DiaDiemID FROM inserted;

DECLARE @TotalStars INT;

```

        SELECT @TotalStars = SUM(SoSao) FROM DanhGia WHERE
DiaDiemID = @DiaDiemID;

        DECLARE @ReviewCount INT;

        SELECT @ReviewCount = COUNT(*) FROM DanhGia WHERE
DiaDiemID = @DiaDiemID;

        UPDATE DiaDiemDuLich

        SET TongSoSao = @TotalStars, SoLuongDanhGia = @ReviewCount

        WHERE ID = @DiaDiemID;

END;

-- Xử lý cho trường hợp xóa đánh giá

IF EXISTS (SELECT * FROM deleted)

BEGIN

        DECLARE @DiaDiemIDDel INT;

        SELECT @DiaDiemIDDel = DiaDiemID FROM deleted;

        DECLARE @TotalStarsDel INT;

        SELECT @TotalStarsDel = SUM(SoSao) FROM DanhGia WHERE
DiaDiemID = @DiaDiemIDDel;

        DECLARE @ReviewCountDel INT;

        SELECT @ReviewCountDel = COUNT(*) FROM DanhGia WHERE
DiaDiemID = @DiaDiemIDDel;

        UPDATE DiaDiemDuLich

        SET TongSoSao = @TotalStarsDel, SoLuongDanhGia =
@ReviewCountDel

        WHERE ID = @DiaDiemIDDel;

END;

```


END;

- Duyệt qua các địa điểm du lịch và in ra tên địa điểm và số lượng khách sạn tại đó

CREATE PROCEDURE ListHotelsPerLocation

AS

BEGIN

DECLARE @locationID INT;

DECLARE @locationName NVARCHAR(255);

DECLARE location_cursor CURSOR FOR

SELECT ID, TenDiaDiem FROM DiaDiemDuLich;

OPEN location_cursor;

FETCH NEXT FROM location_cursor INTO @locationID, @locationName;

-- Duyệt qua các địa điểm du lịch

WHILE @@FETCH_STATUS = 0

BEGIN

-- In ra tên địa điểm và số lượng khách sạn tại đó

PRINT 'Địa điểm: ' + @locationName + ' - Số lượng khách sạn: ' +
CAST((SELECT COUNT(*) FROM KhachSan WHERE DiaDiemID =
@locationID) AS NVARCHAR(10));

FETCH NEXT FROM location_cursor INTO @locationID,
@locationName;

END;

CLOSE location_cursor;

DEALLOCATE location_cursor;

END;

- Tìm kiếm thông tin

-- Tạo một VIEW mới có tên 'View_HotelInfo' hiển thị khách sạn giảm dần của sao đánh giá

CREATE VIEW View_HotelInfo AS

SELECT

ks.ID, -- ID của khách sạn

ks.TenKhachSan, -- Tên của khách sạn

ks.DiaChi, -- Địa chỉ của khách sạn

ks.DiaDiemID, -- ID của địa điểm du lịch liên kết với khách sạn

dd.TenDiaDiem, -- Tên của địa điểm du lịch

ISNULL(AVG(dg.SoSao), 0) AS TrungBinhSoSao -- Tính trung bình số sao của các đánh giá, nếu không có đánh giá nào thì trả về 0

FROM

KhachSan ks -- Bảng KhachSan

LEFT JOIN

DiaDiemDuLich dd ON ks.DiaDiemID = dd.ID -- Kết nối bảng DiaDiemDuLich để lấy tên địa điểm du lịch

LEFT JOIN

DanhGia dg ON ks.ID = dg.DiaDiemID -- Kết nối bảng DanhGia để lấy các đánh giá liên quan đến khách sạn

GROUP BY

ks.ID, ks.TenKhachSan, ks.DiaChi, ks.DiaDiemID, dd.TenDiaDiem; -- Nhóm theo các cột cần thiết để tính trung bình số sao

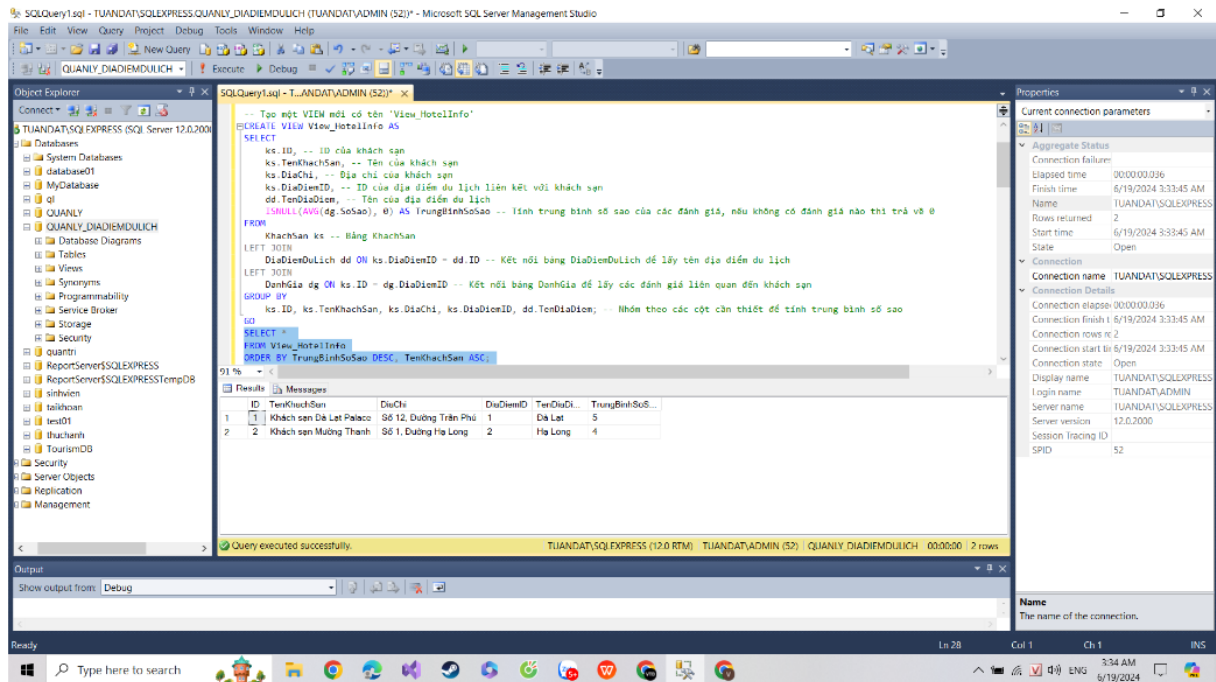
GO

SELECT *

FROM View_HotelInfo

ORDER BY TrungBinhSoSao DESC, TenKhachSan ASC;

- Kết quả



-- Tạo VIEW mới để hiển thị thông tin địa điểm du lịch theo thứ tự giảm dần của sao đánh giá

CREATE VIEW View_DiaDiemDuLich AS

SELECT

dd.ID,

dd.TenDiaDiem,

dd.MoTa,

dd.DiaChi,

dd.LoaiHinhDuLich,

ISNULL(AVG(dg.SoSao), 0) AS TrungBinhSoSao,

COUNT(dg.ID) AS TongSoDanhGia

FROM

DiaDiemDuLich dd

LEFT JOIN

DanhGia dg ON dd.ID = dg.DiaDiemID

GROUP BY

dd.ID, dd.TenDiaDiem, dd.MoTa, dd.DiaChi, dd.LoaiHinhDuLich;

GO

-- Sử dụng VIEW để lấy thông tin và sắp xếp

SELECT *

FROM View_DiaDiemDuLich

ORDER BY TrungBinhSoSao DESC, TenDiaDiem ASC;

- Kết quả

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The central pane displays a SQL query in the 'SQLQuery1.sql' file. The query is as follows:

```
-- Tạo VIEW mới để hiển thị thông tin địa điểm du lịch
CREATE VIEW View_DiaDiemDuLich AS
SELECT
    dd.ID,
    dd.TenDiaDiem,
    dd.MoTa,
    dd.DiaChi,
    dd.LoaiHinhDuLich,
    dg.LoaiHinhDuLich,
    ISNULL(AVG(dg.SoSao), 0) AS TrungBinhSoSao,
    COUNT(dg.ID) AS TongSoDanhGia
FROM
    DiaDiemDuLich dd
LEFT JOIN
    DanhGia dg ON dd.ID = dg.DiaDiemID
GROUP BY
    dd.ID, dd.TenDiaDiem, dd.MoTa, dd.DiaChi, dd.LoaiHinhDuLich;
GO

-- Sử dụng VIEW để lấy thông tin và sắp xếp
SELECT *
FROM View_DiaDiemDuLich
ORDER BY TrungBinhSoSao DESC, TenDiaDiem ASC;
```

The 'Results' pane at the bottom shows the output of the query, which consists of two rows:

ID	TenDiaDiem	MoTa	DiaChi	LoaiHinhDuLich	TrungBinhSoSao	TongSoDanhGia
1	Đà Lạt	Thành phố ngàn hoa	Lâm Đồng	Nghỉ dưỡng	5	1
2	Hạ Long	Vịnh biển đẹp nhất	Quảng Ninh	Thăm quan	4	1

The 'Properties' pane on the right shows the connection details for 'TUANDAT\SQLEXPRESS'.

-- Tạo VIEW mới có tên View_NhaHang để hiển thị thông tin nhà hàng và trung bình số sao

CREATE VIEW View_NhaHang AS

SELECT

nh.ID, -- ID của nhà hàng

nh.TenNhaHang, -- Tên của nhà hàng

nh.DiaChi, -- Địa chỉ của nhà hàng

nh.LoaiHinhAmThuc, -- Loại hình ẩm thực của nhà hàng

dd.TenDiaDiem, -- Tên địa điểm của nhà hàng

ISNULL(AVG(dg.SoSao), 0) AS TrungBinhSoSao, -- Trung bình số sao của đánh giá, nếu không có đánh giá thì trả về 0

COUNT(dg.ID) AS TongSoDanhGia -- Tổng số lượng đánh giá của nhà hàng

FROM

NhaHang nh -- Từ bảng NhaHang

LEFT JOIN

DiaDiemDuLich dd ON nh.DiaDiemID = dd.ID -- LEFT JOIN với bảng DiaDiemDuLich để lấy tên địa điểm liên quan

LEFT JOIN

DanhGia dg ON nh.ID = dg.DiaDiemID -- LEFT JOIN với bảng DanhGia để lấy đánh giá liên quan

GROUP BY

nh.ID, nh.TenNhaHang, nh.DiaChi, nh.LoaiHinhAmThuc, dd.TenDiaDiem;

GO

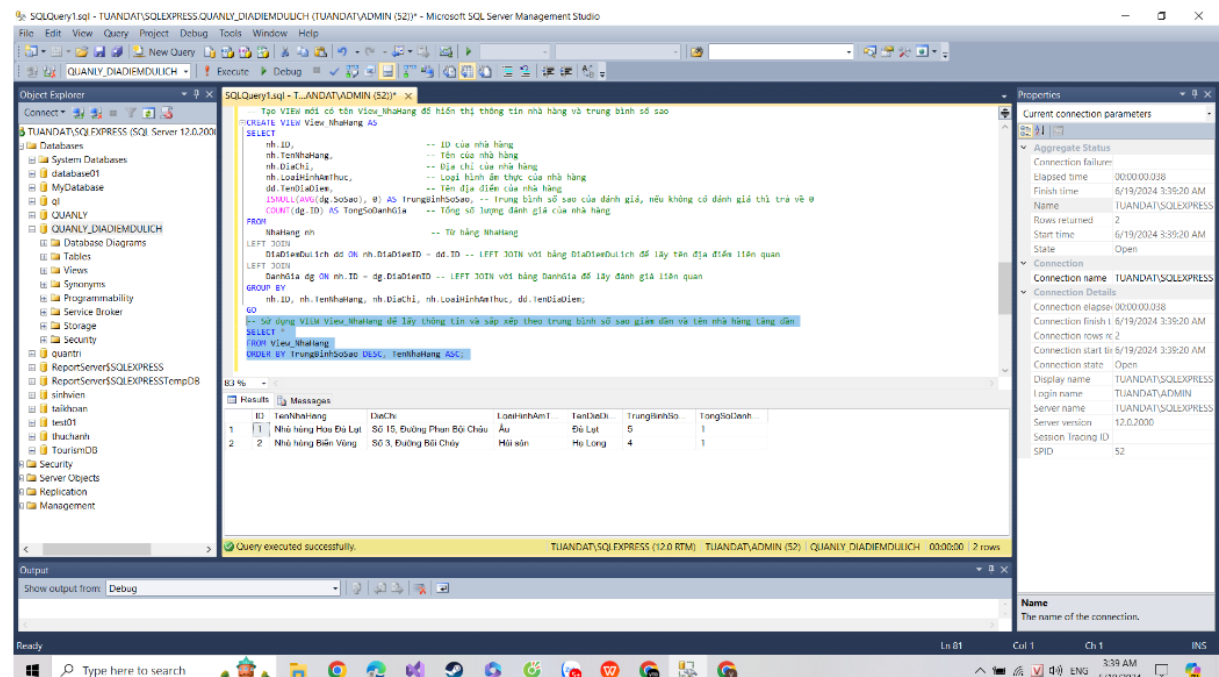
-- Sử dụng VIEW View_NhaHang để lấy thông tin và sắp xếp theo trung bình số sao giảm dần và tên nhà hàng tăng dần

SELECT *

FROM View_NhaHang

ORDER BY TrungBinhSoSao DESC, TenNhaHang ASC;

- Kết quả



- Code cả bài

```
-- Tạo bảng KháchHang
CREATE TABLE KháchHang (
    ID INT PRIMARY KEY IDENTITY(1,1),
    TenKhachHang NVARCHAR(255) NOT NULL,
    Email NVARCHAR(255),
    SoDienThoai NVARCHAR(15)
);
```

```
-- Tạo bảng DiaDiemDuLich
CREATE TABLE DiaDiemDuLich (
    ID INT PRIMARY KEY IDENTITY(1,1),
    TenDiaDiem NVARCHAR(255) NOT NULL,
    MoTa NVARCHAR(MAX),
    DiaChi NVARCHAR(255),
    LoaiHinhDuLich NVARCHAR(100),
    TongSoSao INT DEFAULT 0,
    SoLuongDanhGia INT DEFAULT 0
);
```

```
-- Tạo bảng KháchSan
CREATE TABLE KháchSan (
    ID INT PRIMARY KEY IDENTITY(1,1),
```

```

TenKhachSan NVARCHAR(255) NOT NULL,
DiaChi NVARCHAR(255),
SoSao INT CHECK (SoSao >= 1 AND SoSao <= 5),
DiaDiemID INT,
FOREIGN KEY (DiaDiemID) REFERENCES DiaDiemDuLich(ID)
);

-- Tạo bảng NhaHang
CREATE TABLE NhaHang (
    ID INT PRIMARY KEY IDENTITY(1,1),
    TenNhaHang NVARCHAR(255) NOT NULL,
    DiaChi NVARCHAR(255),
    LoaiHinhAmThuc NVARCHAR(100),
    DiaDiemID INT,
    FOREIGN KEY (DiaDiemID) REFERENCES DiaDiemDuLich(ID)
);

-- Tạo bảng DanhGia
CREATE TABLE DanhGia (
    ID INT PRIMARY KEY IDENTITY(1,1),
    KhachHangID INT,
    DiaDiemID INT,
    SoSao INT CHECK (SoSao >= 1 AND SoSao <= 5),
    BinhLuan NVARCHAR(MAX),
    FOREIGN KEY (KhachHangID) REFERENCES KhachHang(ID),
    FOREIGN KEY (DiaDiemID) REFERENCES DiaDiemDuLich(ID)
);

-- Thêm dữ liệu vào bảng DiaDiemDuLich
INSERT INTO DiaDiemDuLich (TenDiaDiem, MoTa, DiaChi,
LoaiHinhDuLich) VALUES
(N'Đà Lạt', N'Thành phố ngàn hoa', N'Lâm Đồng', N'Nghỉ dưỡng'),
(N'Hạ Long', N'Vịnh biển đẹp nhất', N'Quảng Ninh', N'Tham quan');

-- Thêm dữ liệu vào bảng KhachSan
INSERT INTO KhachSan (TenKhachSan, DiaChi, SoSao, DiaDiemID)
VALUES
(N'Khách sạn Đà Lạt Palace', N'Số 12, Đường Trần Phú', 5, 1),
(N'Khách sạn Mường Thanh', N'Số 1, Đường Hạ Long', 4, 2);

-- Thêm dữ liệu vào bảng NhaHang
INSERT INTO NhaHang (TenNhaHang, DiaChi, LoaiHinhAmThuc,
DiaDiemID) VALUES
(N'Nhà hàng Hoa Đà Lạt', N'Số 15, Đường Phan Bội Châu', N'Âu', 1),
(N'Nhà hàng Biển Vàng', N'Số 3, Đường Bãi Cháy', N'Hải sản', 2);

```

```

-- Thêm dữ liệu vào bảng KháchHang
INSERT INTO KháchHang (TenKhachHang, Email, SoDienThoai) VALUES
(N'Nguyễn Văn A', N'vana@gmail.com', N'0901234567'),
(N'Trần Thị B', N'thib@gmail.com', N'0912345678');

-- Thêm dữ liệu vào bảng DanhGia
INSERT INTO DanhGia (KhachHangID, DiaDiemID, SoSao, BinhLuan)
VALUES
(1, 1, 5, N'Rất đẹp và thơ mộng!'),
(2, 2, 4, N'Cảnh đẹp nhưng dịch vụ chưa tốt.');
```

-- Chức năng thêm địa điểm du lịch

```

CREATE PROCEDURE AddTouristDestination
    @TenDiaDiem NVARCHAR(255),
    @MoTa NVARCHAR(MAX),
    @DiaChi NVARCHAR(255),
    @LoaiHinhDuLich NVARCHAR(100)
AS
BEGIN
    INSERT INTO DiaDiemDuLich (TenDiaDiem, MoTa, DiaChi,
    LoaiHinhDuLich)
    VALUES (@TenDiaDiem, @MoTa, @DiaChi, @LoaiHinhDuLich);
END;
```

-- Chức năng sửa thông tin địa điểm du lịch

```

CREATE PROCEDURE UpdateTouristDestination
    @ID INT,
    @TenDiaDiem NVARCHAR(255),
    @MoTa NVARCHAR(MAX),
    @DiaChi NVARCHAR(255),
    @LoaiHinhDuLich NVARCHAR(100)
AS
BEGIN
    UPDATE DiaDiemDuLich
    SET TenDiaDiem = @TenDiaDiem, MoTa = @MoTa, DiaChi = @DiaChi,
    LoaiHinhDuLich = @LoaiHinhDuLich
    WHERE ID = @ID;
END;
```

-- Chức năng xóa địa điểm du lịch

```

CREATE PROCEDURE DeleteTouristDestination
    @ID INT
AS
BEGIN
    DELETE FROM DiaDiemDuLich WHERE ID = @ID;
END;
```

-- Kiểm tra nếu VIEW có tên 'View_HotelInfo' đã tồn tại trong cơ sở dữ liệu

```

IF OBJECT_ID('View_HotelInfo', 'V') IS NOT NULL
```



```

-- Xóa VIEW 'View_HotelInfo' nếu nó đã tồn tại
DROP VIEW View_HotelInfo;
GO

-- Tạo một VIEW mới có tên 'View_HotelInfo'
CREATE VIEW View_HotelInfo AS
SELECT
    ks.ID, -- ID của khách sạn
    ks.TenKhachSan, -- Tên của khách sạn
    ks.DiaChi, -- Địa chỉ của khách sạn
    ks.DiaDiemID, -- ID của địa điểm du lịch liên kết với khách sạn
    dd.TenDiaDiem, -- Tên của địa điểm du lịch
    ISNULL(AVG(dg.SoSao), 0) AS TrungBinhSoSao -- Tính trung bình số sao
    của các đánh giá, nếu không có đánh giá nào thì trả về 0
FROM
    KháchSan ks -- Bảng KháchSan
LEFT JOIN
    DiaDiemDuLich dd ON ks.DiaDiemID = dd.ID -- Kết nối bảng
    DiaDiemDuLich để lấy tên địa điểm du lịch
LEFT JOIN
    DanhGia dg ON ks.ID = dg.DiaDiemID -- Kết nối bảng DanhGia để lấy các
    đánh giá liên quan đến khách sạn
GROUP BY
    ks.ID, ks.TenKhachSan, ks.DiaChi, ks.DiaDiemID, dd.TenDiaDiem; -- Nhóm
    theo các cột cần thiết để tính trung bình số sao
GO
SELECT *
FROM View_HotelInfo
ORDER BY TrungBinhSoSao DESC, TenKhachSan ASC;

-- Kiểm tra xem VIEW có tồn tại không và xóa nếu có
IF OBJECT_ID('View_DiaDiemDuLich', 'V') IS NOT NULL
    DROP VIEW View_DiaDiemDuLich;
GO

-- Tạo VIEW mới để hiển thị thông tin địa điểm du lịch
CREATE VIEW View_DiaDiemDuLich AS
SELECT
    dd.ID,
    dd.TenDiaDiem,
    dd.MoTa,
    dd.DiaChi,
    dd.LoaiHinhDuLich,
    ISNULL(AVG(dg.SoSao), 0) AS TrungBinhSoSao,
    COUNT(dg.ID) AS TongSoDanhGia

```

```

FROM
    DiaDiemDuLich dd
LEFT JOIN
    DanhGia dg ON dd.ID = dg.DiaDiemID
GROUP BY
    dd.ID, dd.TenDiaDiem, dd.MoTa, dd.DiaChi, dd.LoaiHinhDuLich;
GO

-- Sử dụng VIEW để lấy thông tin và sắp xếp
SELECT *
FROM View_DiaDiemDuLich
ORDER BY TrungBinhSoSao DESC, TenDiaDiem ASC;

-- Kiểm tra xem VIEW có tên View_NhaHang đã tồn tại chưa
IF OBJECT_ID('View_NhaHang', 'V') IS NOT NULL
    -- Nếu tồn tại, thực hiện lệnh DROP VIEW để xóa VIEW này đi
    DROP VIEW View_NhaHang;
GO

-- Tạo VIEW mới có tên View_NhaHang để hiển thị thông tin nhà hàng và trung
    bình số sao
CREATE VIEW View_NhaHang AS
SELECT
    nh.ID,                -- ID của nhà hàng
    nh.TenNhaHang,        -- Tên của nhà hàng
    nh.DiaChi,            -- Địa chỉ của nhà hàng
    nh.LoaiHinhAmThuc,    -- Loại hình ẩm thực của nhà hàng
    dd.TenDiaDiem,        -- Tên địa điểm của nhà hàng
    ISNULL(AVG(dg.SoSao), 0) AS TrungBinhSoSao, -- Trung bình số sao của
    đánh giá, nếu không có đánh giá thì trả về 0
    COUNT(dg.ID) AS TongSoDanhGia -- Tổng số lượng đánh giá của nhà
    hàng
FROM
    NhaHang nh                -- Từ bảng NhaHang
LEFT JOIN
    DiaDiemDuLich dd ON nh.DiaDiemID = dd.ID -- LEFT JOIN với bảng
    DiaDiemDuLich để lấy tên địa điểm liên quan
LEFT JOIN
    DanhGia dg ON nh.ID = dg.DiaDiemID -- LEFT JOIN với bảng DanhGia để
    lấy đánh giá liên quan
GROUP BY
    nh.ID, nh.TenNhaHang, nh.DiaChi, nh.LoaiHinhAmThuc, dd.TenDiaDiem;
GO

-- Sử dụng VIEW View_NhaHang để lấy thông tin và sắp xếp theo trung bình số
    sao giảm dần và tên nhà hàng tăng dần
SELECT *

```

```
FROM View_NhaHang  
ORDER BY TrungBinhSoSao DESC, TenNhaHang ASC;
```