**SQL EXERCISE – ADVANCED CONCEPTS**

**Exercise 1: Ranking and Window Functions**

**Code :-**

-- Create Database

CREATE DATABASE GourmetRankingDB;

GO

USE GourmetRankingDB;

GO

-- Create Table

CREATE TABLE Dishes (

DishID INT IDENTITY(1,1) PRIMARY KEY,

DishName NVARCHAR(100) NOT NULL,

Category NVARCHAR(50) NOT NULL,

Price DECIMAL(10,2) NOT NULL

);

GO

-- Insert Sample Data

INSERT INTO Dishes (DishName, Category, Price) VALUES

('Chocolate Lava Cake', 'Dessert', 5.99),

('Tiramisu', 'Dessert', 6.49),

('Macaron', 'Dessert', 3.99),

('Cheesecake', 'Dessert', 6.49),

('Grilled Salmon', 'Main Course', 12.99),

('Pasta Alfredo', 'Main Course', 10.49),

('Veggie Burger', 'Main Course', 9.99),

('Iced Latte', 'Beverage', 3.49),

('Mango Lassi', 'Beverage', 3.99),

('Hot Chocolate', 'Beverage', 3.49),

('Garlic Bread', 'Appetizer', 4.49),

('Stuffed Mushrooms', 'Appetizer', 6.99),

('Spring Rolls', 'Appetizer', 5.99),

('Nachos', 'Appetizer', 6.99);

GO

-- Query 1: ROW\_NUMBER() - Unique rank within each category

SELECT

DishName,

Category,

Price,

ROW\_NUMBER() OVER (

PARTITION BY Category

ORDER BY Price DESC

) AS RowNum

FROM Dishes

ORDER BY Category, RowNum;

GO

-- Query 2: RANK() - Allows gaps in ranking when ties occur

SELECT

DishName,

Category,

Price,

RANK() OVER (

PARTITION BY Category

ORDER BY Price DESC

) AS PriceRank

FROM Dishes

ORDER BY Category, PriceRank;

GO

-- Query 3: DENSE\_RANK() - No gaps in ranking for ties

SELECT

DishName,

Category,

Price,

DENSE\_RANK() OVER (

PARTITION BY Category

ORDER BY Price DESC

) AS DensePriceRank

FROM Dishes

ORDER BY Category, DensePriceRank;

GO

-- Query 4: Top 3 most expensive dishes per category using DENSE\_RANK()

WITH TopDishes AS (

SELECT

DishName,

Category,

Price,

DENSE\_RANK() OVER (

PARTITION BY Category

ORDER BY Price DESC

) AS PriceTier

FROM Dishes

)

SELECT

DishName,

Category,

Price,

PriceTier

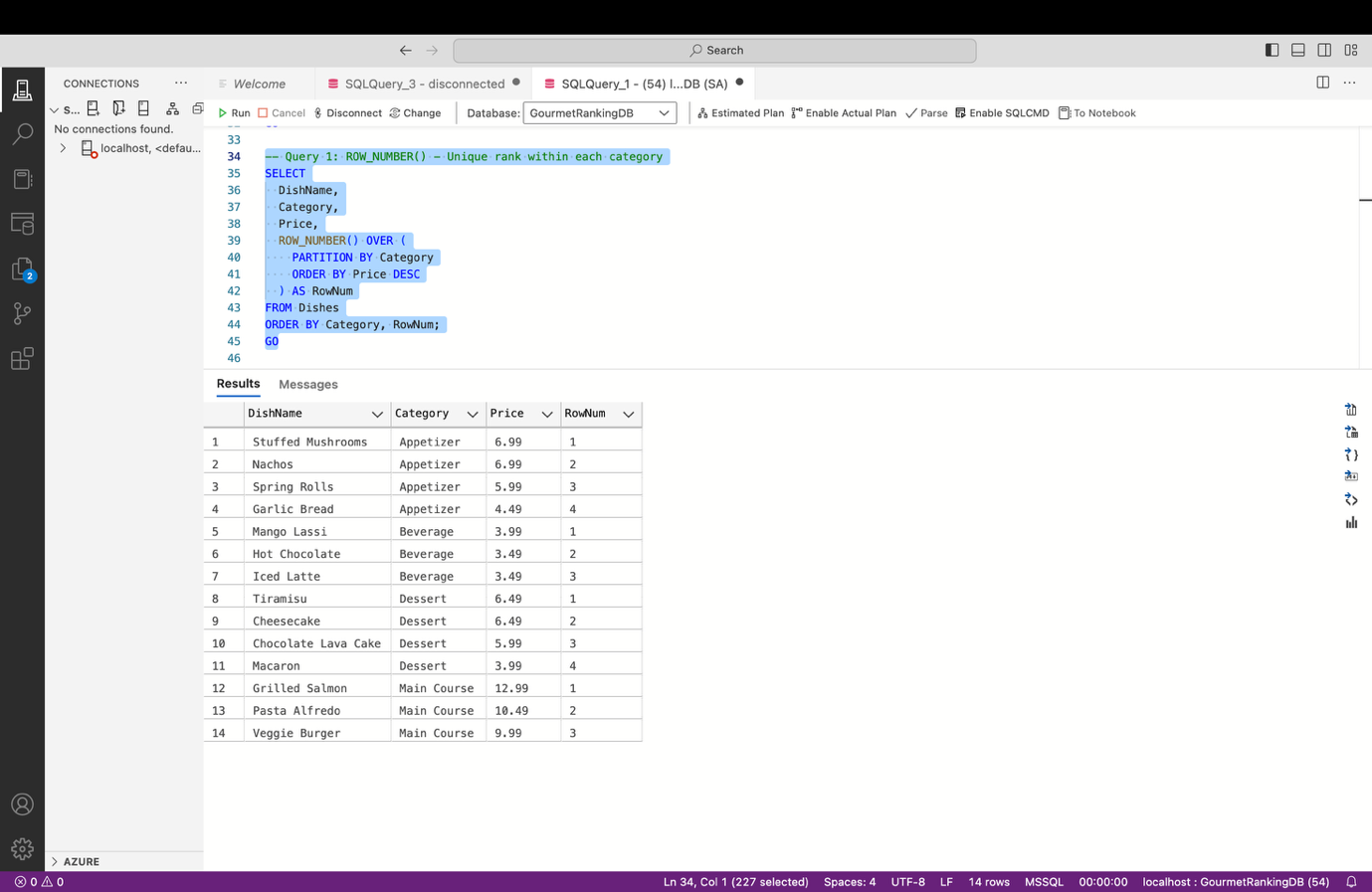
FROM TopDishes

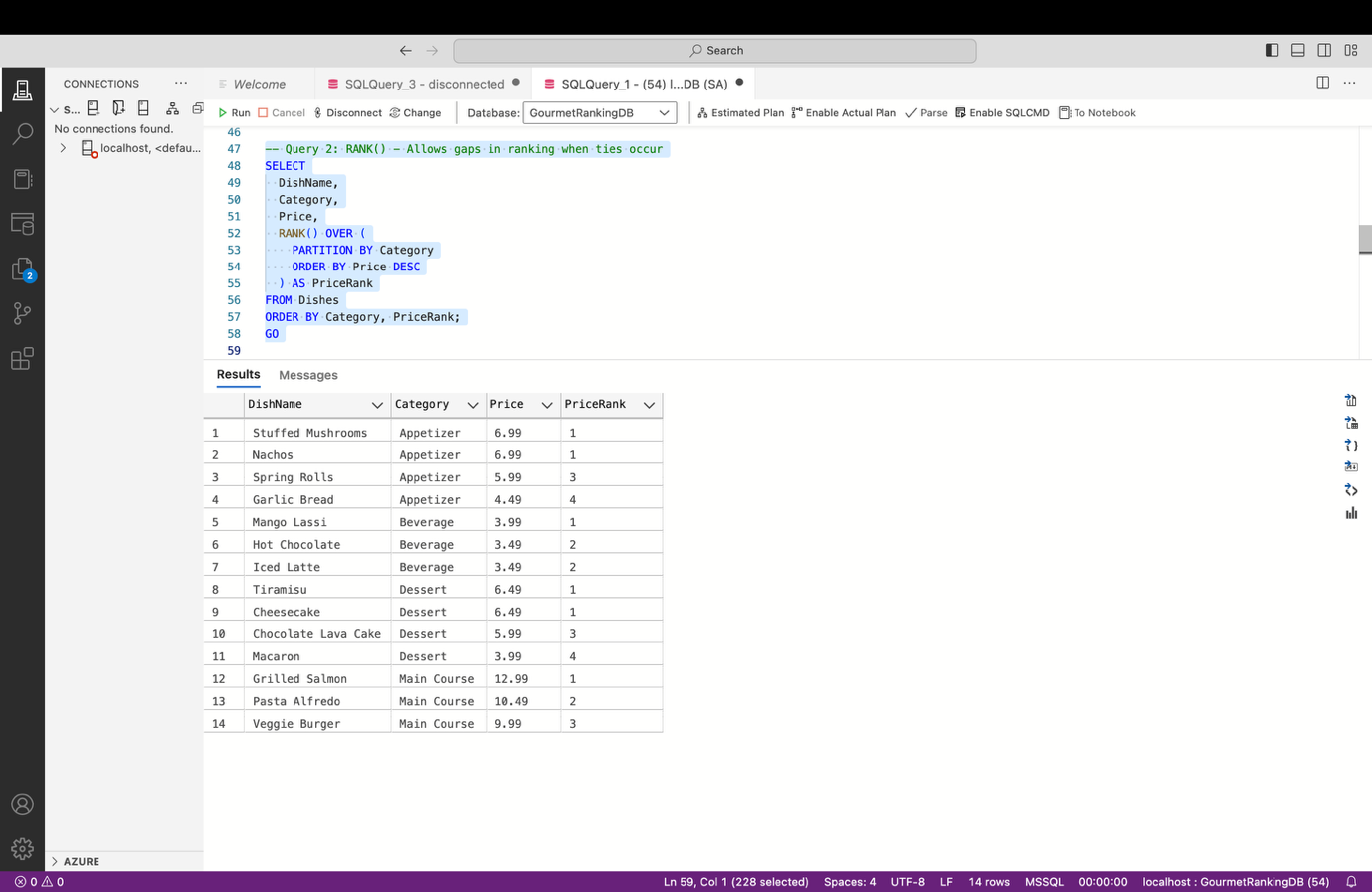
WHERE PriceTier <= 3

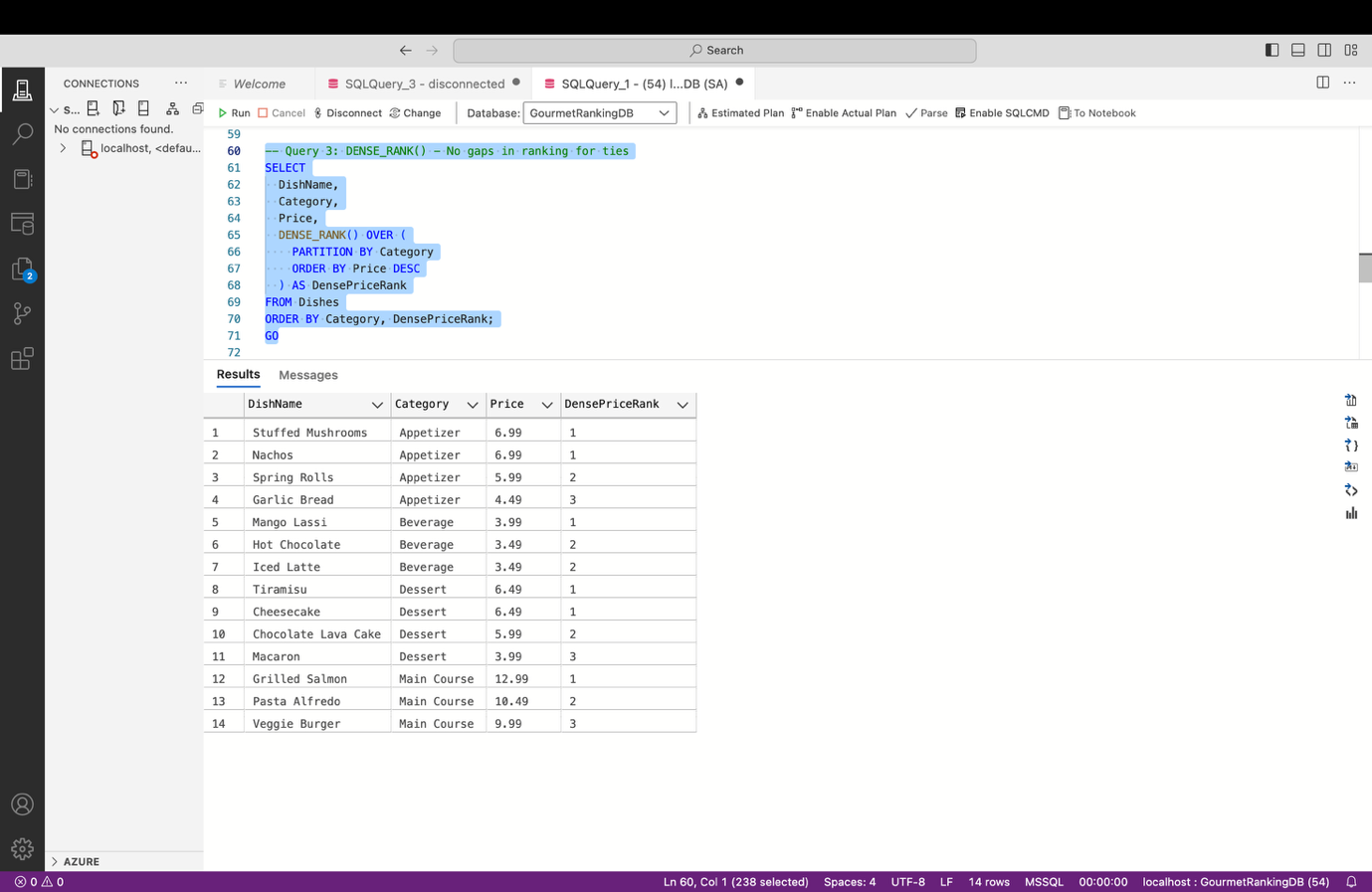
ORDER BY Category, PriceTier;

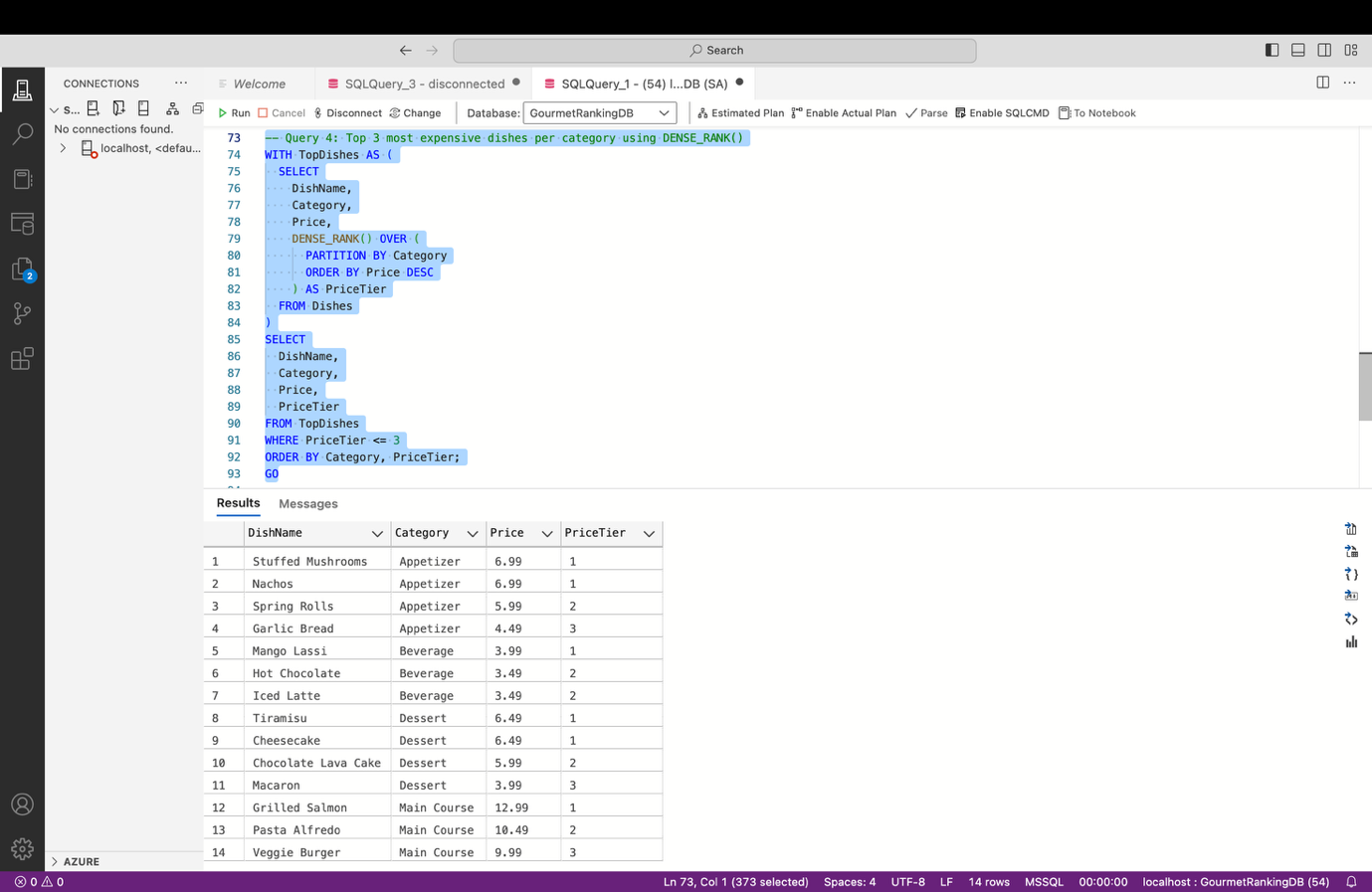
GO

**OUTPUT :-**

****

****

****

****