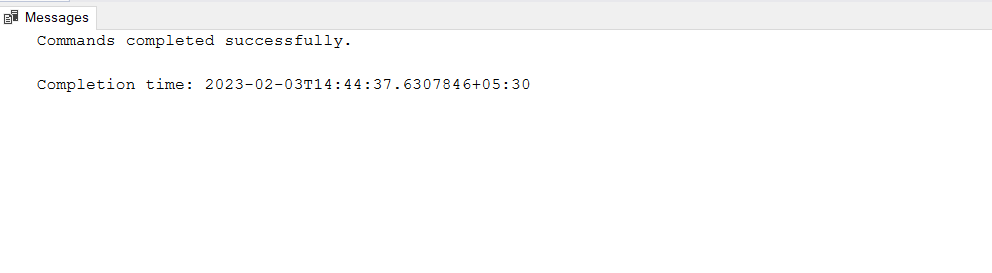
**Assignment-1**

-- Create Database

CREATE DATABASE Northwind



-- Create Table

CREATE TABLE Products(

ProductID int not null,

ProductName varchar(50) not null,

SupplierID int not null,

CategoryID int not null,

QuantityPerUnit varchar(50) not null,

UnitPrice decimal(10,4) not null,

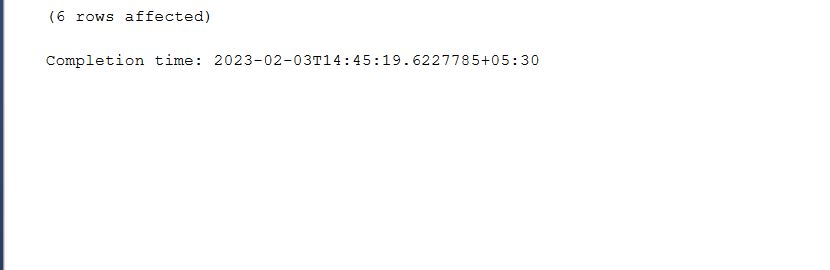
UnitsInStock smallint not null,

UnitsOnOrder smallint not null,

ReorderLevel smallint not null,

Discountinued bit not null,

)



-- Insert Data into table

INSERT INTO Products VALUES (1,'Chang',1,1,'24 -12 oz bottles', 19.0000, 20,45, 25, 'False'),

(2,'Chai',1,1,'21 boxes x 20 bags',18.0000,20,0,10,'False'),

(3,'Genen Shouyu',6,2,'24 - 250 ml bottles',15.5000,20,0,5,'False'),

(4,'Aniseed Syrup',1,2,'12 - 550 ml bottles',10.0000,20,70,25,'False'),

(5,'Northwoods Cranberry Sauce',3,2,'12 - 12 oz jars',40.0000,20,0,0,'False'),

(6,'Chef Antons Gumbo Mix',2,2,'36 boxes',21.3500,20,0,0,'True'),

(7,'Grandmas Boysenberry Spread',3,2,'12 - 8 oz jars',25.0000,20,0,25,'False'),

(8,'Chef Antons Cajun Seasoning',2,2,'48 - 6 oz jars',22.0000,20,0,0,'False'),

(9,'Queso Cabrales',5,4,'1 kg pkg.',21.0000,20,30,30,'False'),

(10,'Queso Manchego La Pastora',5,4,'10 - 500 g pkgs.',38.0000,20,0,0,'False'),

(11,'Mishi Kobe Niku',4,6,'18 - 500 g pkgs.',97.0000,20,0,0,'True'),

(12,'Tofu',6,7,'40 - 100 g pkgs. ',25.2500,20,0,0,'False'),

(13,'Uncle Bobs Organic Dried Pears',3,7,'12 - 1 lb pkgs.', 30.0000, 20, 0, 10, 'False'),

(14,'Konbu',6,8,'2 kg box',6.0000,20,0,5,'False'),

(15,'Ikura',4,8,'12 - 200 ml jars',31.0000,20,0,0,'False'),

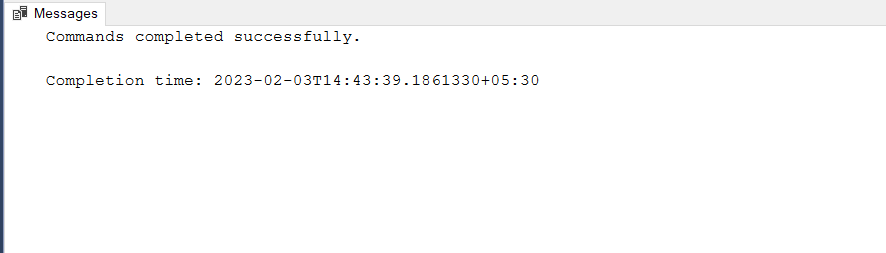
(16,'Geitost',3,9,'500 g',2.5000,20,0,0,'False'),

(17,'Rd Kaviar',5,9,'24 -200 g jars',15.0000,20,0,0,'False'),

(18,'Ipoh coffee',6,9,'16 - 500 g tins',46.0000,21,17,7,'False'),

(19,'Rssle Sauerkraut',3,9,'25 - 825 g cans',45.6000,23,0,0,'False'),

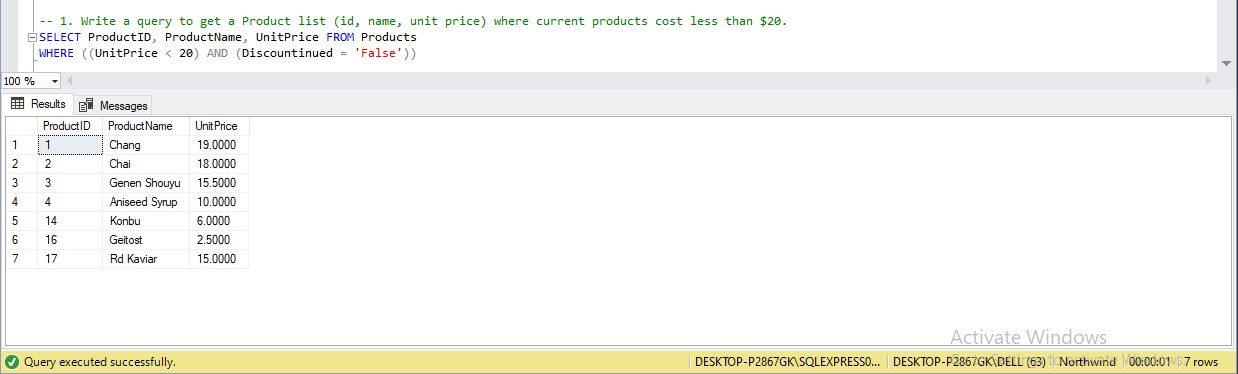
(20,'Manjimup Dried Apples',5,9,'50 - 300 g pkgs.',53.0000,27,0,0,'False')



-- 1. Write a query to get a Product list (id, name, unit price) where current products cost less than $20.

SELECT ProductID, ProductName, UnitPrice FROM Products

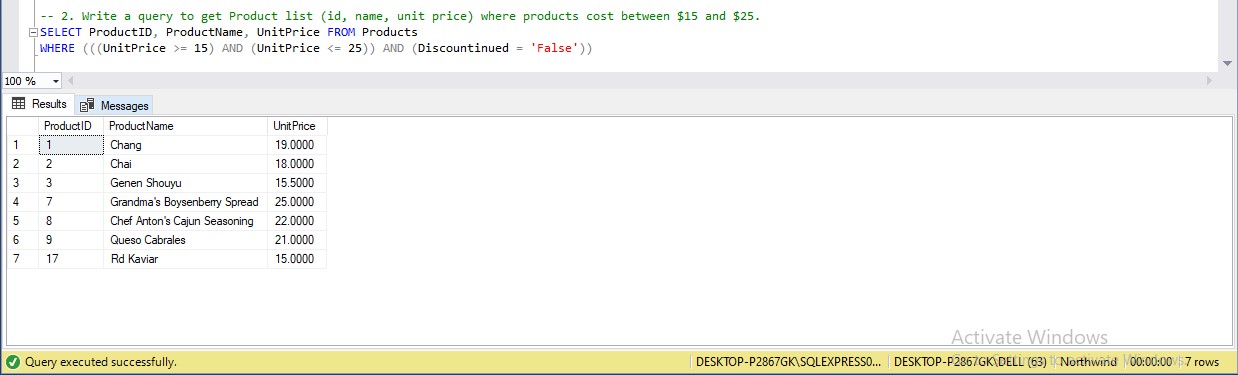
WHERE ((UnitPrice < 20) AND (Discountinued = 'False'))



-- 2. Write a query to get Product list (id, name, unit price) where products cost between $15 and $25.

SELECT ProductID, ProductName, UnitPrice FROM Products

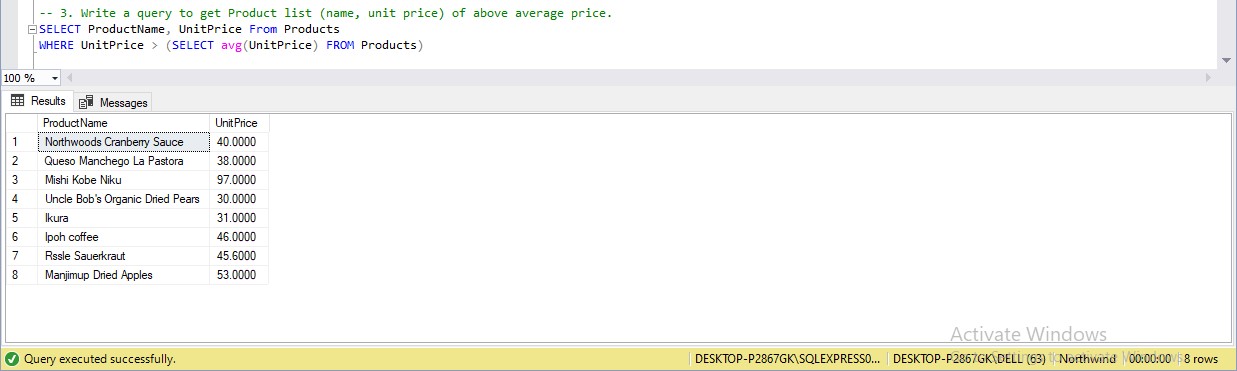
WHERE (((UnitPrice >= 15) AND (UnitPrice <= 25)) AND (Discountinued = 'False'))



-- 3. Write a query to get Product list (name, unit price) of above average price.

SELECT ProductName, UnitPrice From Products

WHERE UnitPrice > (SELECT avg(UnitPrice) FROM Products)



-- 4. Write a query to get Product list (name, unit price) of ten most expensive products.

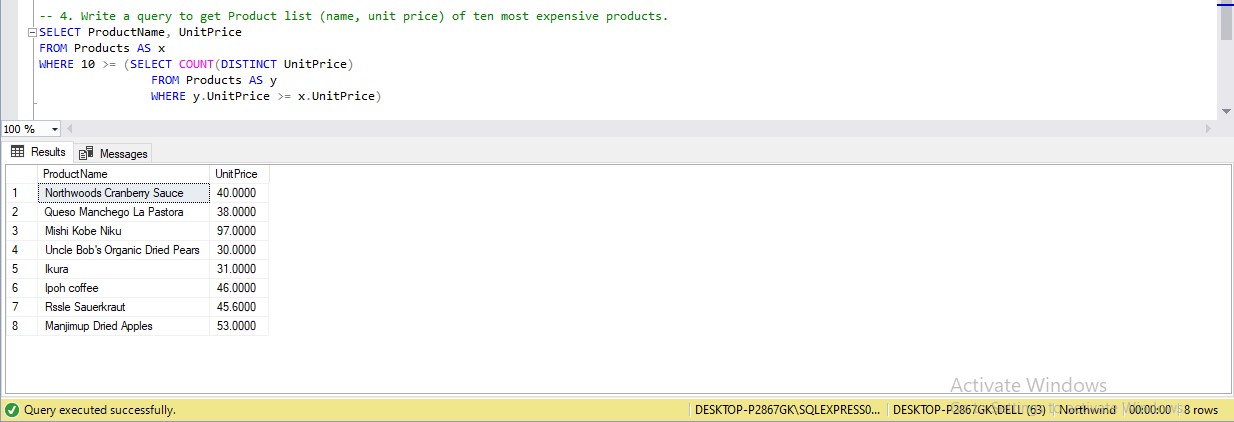
SELECT ProductName, UnitPrice

FROM Products AS x

WHERE 10 >= (SELECT COUNT(DISTINCT UnitPrice)

FROM Products AS y

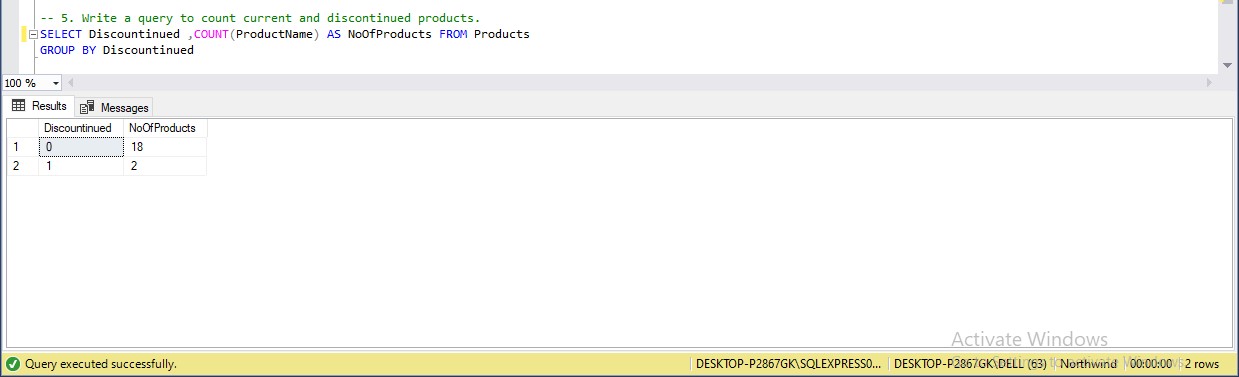
WHERE y.UnitPrice >= x.UnitPrice)



-- 5. Write a query to count current and discontinued products.

SELECT Discountinued ,COUNT(ProductName) AS NoOfProducts FROM Products

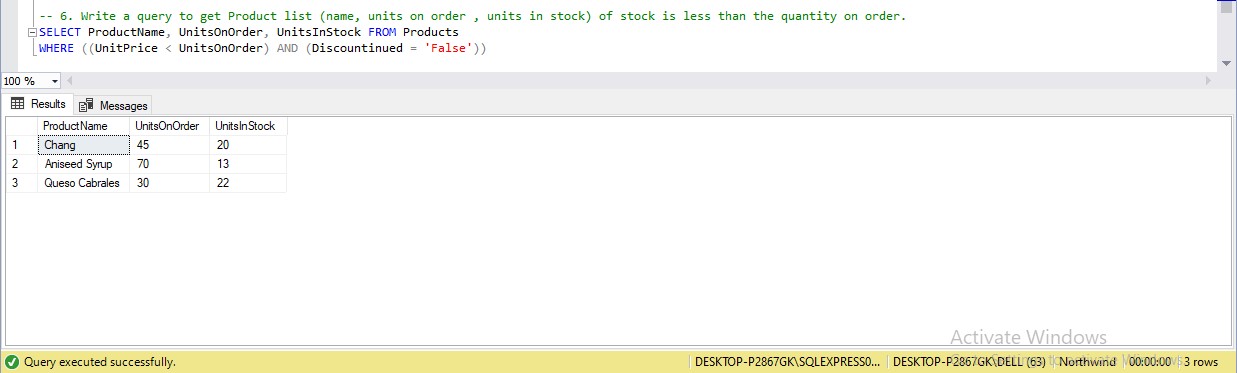
GROUP BY Discountinued



-- 6. Write a query to get Product list (name, units on order , units in stock) of stock is less than the quantity on order.

SELECT ProductName, UnitsOnOrder, UnitsInStock FROM Products

WHERE ((UnitPrice < UnitsOnOrder) AND (Discountinued = 'False'))

****