

Assignment 1 : Simple SQL Query with a single table with where clause

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))
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL query:

```
--tab1 products-->
select * from Products

<--1. Write a query to get a Product list (id, name, unit price) where current products costless than $20.-->
select ProductID, ProductName, UnitPrice from Products where ((UnitPrice < 20))
```

The query results are displayed in a table with 39 rows. The columns are ProductID, ProductName, and UnitPrice. The results show products with unit prices less than 20.

ProductID	ProductName	UnitPrice
1	Chai	18.00
2	Chang	19.00
3	Aniseed Syrup	10.00
4	Korbu	6.00
5	Garden of Eatin'	15.50
6	Pavlova	17.45
7	Teatime Chocolate Biscuits	9.20
8	Sir Rodney's Scones	10.00
9	Tutti Frutti	9.00
10	Guaraná Fantástica	4.50
11	NuNuCa Nuß-Nougat-Creme	14.00
12	Gorgonzola Telino	12.50
13	Glenlivet	2.50
14	Stagbush Ale	14.00
15	Steeleye Stout	18.00
16	Inland Sil	19.00
17	Chateau d'Aux	18.00
18	Boston Crab Meat	18.40
19	Jack's New England Clam Chowder	9.85
20	Singaporean Hokkien Fried Mee	14.00
21	Gula Malacca	19.45
22	Poppinge old	9.50
23	Rongzhi	12.00
24	Zaansse koeken	9.50
25	Chocolade	12.75
26	Valkonnen suklaa	16.25
27	Filo filu	7.00
28	Tourtiere	7.45
29	Ravioli Angelo	19.50
30	Escaroto de Bourgoone	13.25

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))
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
--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))
```

The query has been executed successfully, and the results are displayed in the Results pane. The results show 25 rows of product data, including ProductID, ProductName, and UnitPrice.

ProductID	ProductName	UnitPrice
1	Chai	18.00
2	Chang	19.00
3	Chef Anton's Cajun Seasoning	22.00
4	Chef Anton's Gumbo Mix	21.35
5	Grandma's Boysenberry Spread	25.00
6	Queso Cabrales	21.00
7	Tofu	23.25
8	Genen Shouyu	15.50
9	Pavlova	17.45
10	Oustaf's Knickerbrød	21.00
11	Steelnye Sloud	18.00
12	Inlagd Sill	19.00
13	Chartreuse verte	18.00
14	Boston Crab Meat	18.40
15	Gula Malacca	19.45
16	Maxilaku	20.00
17	Valkonnen suklaa	16.25
18	Pâté chinois	24.00
19	Ravioli Angelo	19.50
20	Louisiana Fiery Hot Pepper Sauce	21.05
21	Louisiana Hot Spiced Okra	17.00
22	Outback Lager	15.00
23	Pilsenerbeer	21.50
24	Röd Kaviar	15.00
25	Lakkalikööri	18.00

The status bar at the bottom indicates that the query was executed successfully, returning 25 rows.

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 )
```

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The 'Object Explorer' on the left shows the 'Northwind' database structure, including tables like 'Products'. The 'SQL Query2.sql' window in the center contains the following query:

```
--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 )
```

The 'Results' pane on the right shows the output of the query, listing 30 products with their names and unit prices. The status bar at the bottom indicates 'Query executed successfully.' and '52 rows'.

productname	unitprice
Chai	18.00
Chang	19.00
Aniseed Syrup	10.00
Chef Anton's Cajun Seasoning	22.00
Chef Anton's Gumbo Mix	21.35
Grandma's Boysenberry Spread	25.00
Queso Cabrales	21.00
Konbu	6.00
Tokai	23.25
Garden of Eatin'	15.50
Pavlova	17.45
Treatime Chocolate Biscuits	9.20
Sir Rodney's Scones	10.00
Gustaf's Knäckebröd	21.00
Turkey	9.00
Garden of Eatin'	4.50
NuNuCa Nuß-Nougat-Creme	14.00
Nord-Ost Majesthering	25.89
Gorgonzola Telino	12.50
Gelost	2.50
Sasquatch Ale	14.00
Stearns Stout	18.00
Islagdt Sill	19.00
Gravad lax	26.00
Chartreuse verte	18.00
Boston Crab Meat	18.40
Jack's New England Clam Chowder	9.65
Singaporean Hokkien Fried Mee	14.00
Gula Malacca	19.45
Roadside id	9.50

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

```
select top 10 ProductName, UnitPrice from Products  
order by UnitPrice desc;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
--4. Write a query to get Product list (name, unit price) of ten most expensive products -->  
select top 10 ProductName, UnitPrice from Products order by UnitPrice desc;
```

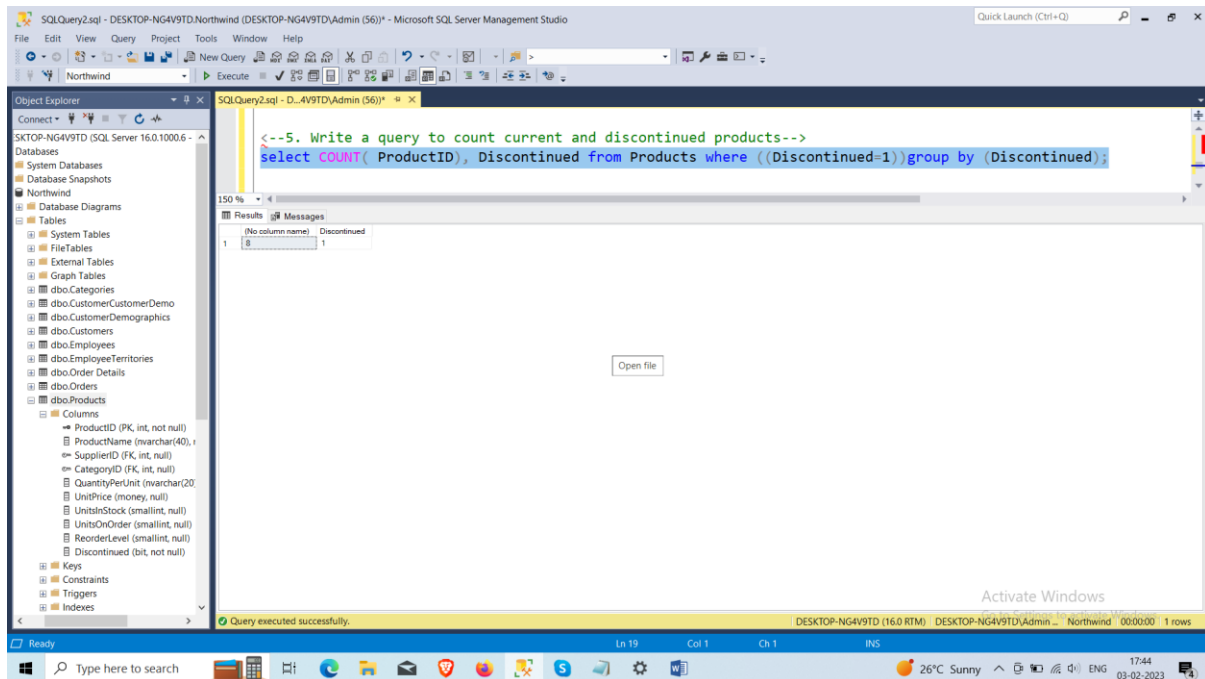
The query has been executed successfully, and the results are displayed in the Results pane. The results show the top 10 most expensive products from the Products table, ordered by UnitPrice in descending order.

	ProductName	UnitPrice
1	Côte de Blaye	263.50
2	Thüringer Rostbratwurst	123.79
3	Mala Kebab Nöku	97.00
4	Sir Rodney's Marmalade	81.00
5	Camaroon Tigers	62.50
6	Raclette Courdault	55.00
7	Manjirup Dried Apples	53.00
8	Tarte au sucre	49.30
9	Ispah Coffee	48.00
10	Rosie Sauerkraut	45.60

The status bar at the bottom indicates that the query was executed successfully and returned 10 rows.

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

```
select COUNT( ProductID), Discontinued from Products  
where ((Discontinued=1))group by (Discontinued);
```



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 ( ((UnitsInStock)<UnitsOnOrder));
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
--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 ( ((UnitsInStock)<UnitsOnOrder));
```

The query results are displayed in a table with the following data:

ProductID	ProductName	UnitsOnOrder	UnitsInStock
1	Chang	40	17
2	Aniseed Syrup	70	13
3	Queso Cabrales	30	22
4	Sir Rodney's Scones	40	3
5	Gorgonzola Telino	70	0
6	Mascarpone Fabioli	40	9
7	Graved lax	50	11
8	Rogede sild	70	5
9	Chocolade	70	15
10	Maailaku	60	10
11	Wimmers gute Semmelknödel	80	22
12	Louisiana Hot Spiced Orea	100	4
13	Scottish Longbreads	10	6
14	Longlife Tofu	20	4

The status bar at the bottom indicates that the query was executed successfully, returning 14 rows.