Database System (CS-262)

Assignment

Session 2021-2025

Submitted By

Syed Hashir Husnain 2021-CS-01

Submitted To

Mr. Nauman Babar

Department of Computer Science

University of Engineering and Technology
Lahore, Pakistan

Contents

Chapter 1

Assignment Report

1.1 Abstract

The purpose of this Assignment is to practice GROUP BY and HAVING Clause concepts in Database Systems. We had given the queries related to some data. The problem is we have to filter out some data based on some conditions. DBMS helps us to do this. Just we have to write queries(means tell it) and it will generate our required output. We practiced some aggregate functions available in T-Sql and aliased the tables and columns for our simplicity. Main idea behind this assignment is to practice these queries.

1.2 Technlogy Stack

Table 1.1: Details of technology used in completion this assignment

Technology	**Version**
MS SQL Server	2022 Developer
SQL Server Management Studio	19.0.20179.0+4bc80247
Microsoft .NET Framework	4.0.30319.42000

1.3 Database Schema Used

- Northwind Schema

1.4 Learning Objectives

- Aggregate Functions
- GROUP BY Clause
- HAVING Clause
- Aliasing Method(Tables and Columns)

1.5 Lab Tasks

1.5.1 Problem 01

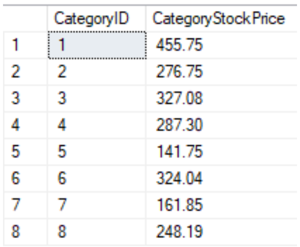
Perform all the group function on Northwind Schema

Problem Statement

List the total price of each category present in a stock.

Query

```
SELECT CategoryID, SUM(UnitPrice) AS CategoryStockPrice
FROM Products
GROUP BY CategoryID
```



	CategoryID	CategoryStockPrice
1	1	455.75
2	2	276.75
3	3	327.08
4	4	287.30
5	5	141.75
6	6	324.04
7	7	161.85
8	8	248.19

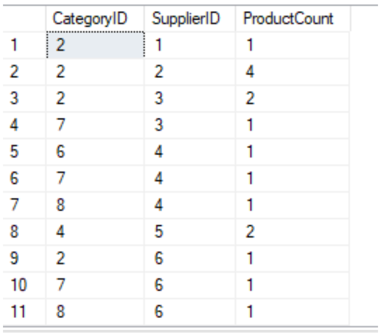
Figure 1.1: Result generated: Total Price of each Category

Problem Statement

List the no. of products supplied by supplier in each category.

Query

```
SELECT CategoryID, SupplierID, COUNT(*) AS ProductCount
FROM Products
WHERE UnitsOnOrder IS NOT NULL
GROUP BY CategoryID, SupplierID
```



	CategoryID	SupplierID	ProductCount
1	2	1	1
2	2	2	4
3	2	3	2
4	7	3	1
5	6	4	1
6	7	4	1
7	8	4	1
8	4	5	2
9	2	6	1
10	7	6	1
11	8	6	1

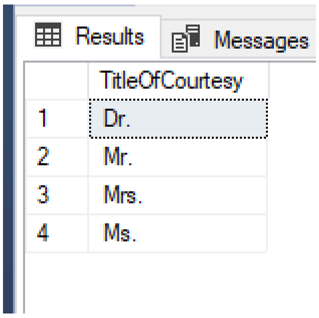
Figure 1.2: Result generated: Count of Products

Problem Statement

Group By can also be used as DISTINCT.

Query

SELECT TitleOfCourtesy
FROM Employees
GROUP BY TitleOfCourtesy



	TitleOfCourtesy
1	Dr.
2	Mr.
3	Mrs.
4	Ms.

Figure 1.3: Result generated: Alternate to DISTINCT

1.5.2 Problem 02

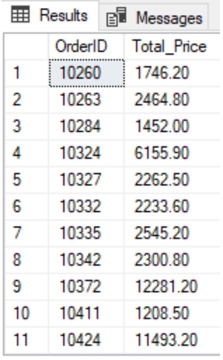
Perform all the group function using HAVING clause on Northwind Schema

Problem Statement

Select all Order IDs and their total amount where discount greater than 5 percent is applied.

Query

SELECT OrderID, SUM(UnitPrice ∗ Quantity) TotalPrice
FROM Order Details
GROUP BY OrderID
HAVING SUM(Discount) > 0.5



	OrderID	Total_Price
1	10260	1746.20
2	10263	2464.80
3	10284	1452.00
4	10324	6155.90
5	10327	2262.50
6	10332	2233.60
7	10335	2545.20
8	10342	2300.80
9	10372	12281.20
10	10411	1208.50
11	10424	11493.20

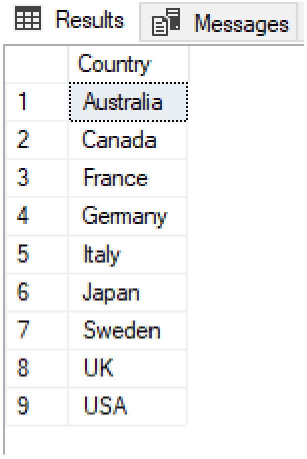
Figure 1.4: Result generated: Total Price of each order

Problem Statement

Select Countries which have more than one suppliers.

Query

```
SELECT Country
FROM Suppliers
GROUP BY Country
HAVING COUNT(ContactName)>= 2
```



	Country
1	Australia
2	Canada
3	France
4	Germany
5	Italy
6	Japan
7	Sweden
8	UK
9	USA

Figure 1.5: Result generated: Multiple Suppliers

1.5.3 Problem 03

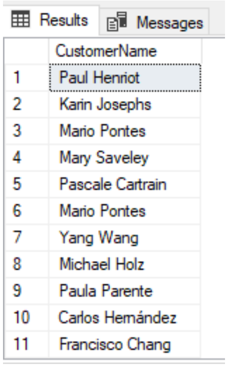
Apply aliasing syntax on arbitrary column on Northwind Schema.

Problem Statement

List All the customers who have ordered something.

Query

```
SELECT C.ContactName AS CustomerName
FROM Customers AS C
JOIN Orders AS O
ON O.CustomerID = C.CustomerID
```



	CustomerName
1	Paul Henriot
2	Karin Josephs
3	Mario Pontes
4	Mary Saveley
5	Pascale Cartrain
6	Mario Pontes
7	Yang Wang
8	Michael Holz
9	Paula Parente
10	Carlos Hernández
11	Francisco Chang

Figure 1.6: Result generated: Customer Names From Orders List

1.6 Home Tasks

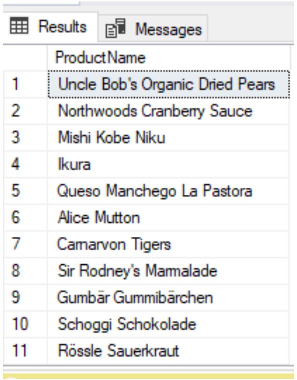
1.6.1 Problem 01

Problem Statement

List name of all the products whose price is above average.

Query

SELECT ProductName
FROM Products
WHERE UnitPrice > (SELECT AVG(UnitPrice) FROM Products)



	ProductName
1	Uncle Bob's Organic Dried Pears
2	Northwoods Cranberry Sauce
3	Mishi Kobe Niku
4	Ikura
5	Queso Manchego La Pastora
6	Alice Mutton
7	Carnarvon Tigers
8	Sir Rodney's Marmalade
9	Gumbär Gummibärchen
10	Schoggi Schokolade
11	Rössle Sauerkraut

Figure 1.7: Result generated: Price greater than average

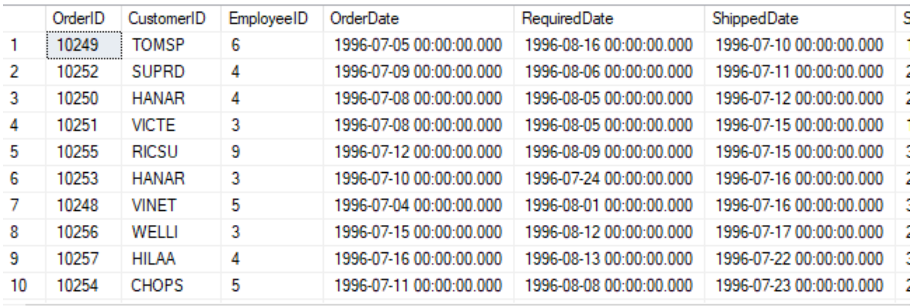
1.6.2 Problem 02

Problem Statement

Write a query to generate report showing date wise orders shipped

Query

SELECT *
FROM Orders
WHERE ShippedDate is NOT NULL
ORDER BY ShippedDate



	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	S
1	10249	TOMSP	6	1996-07-05 00:00:00.000	1996-08-16 00:00:00.000	1996-07-10 00:00:00.000	1
2	10252	SUPRD	4	1996-07-09 00:00:00.000	1996-08-06 00:00:00.000	1996-07-11 00:00:00.000	2
3	10250	HANAR	4	1996-07-08 00:00:00.000	1996-08-05 00:00:00.000	1996-07-12 00:00:00.000	2
4	10251	VICTE	3	1996-07-08 00:00:00.000	1996-08-05 00:00:00.000	1996-07-15 00:00:00.000	1
5	10255	RICSU	9	1996-07-12 00:00:00.000	1996-08-09 00:00:00.000	1996-07-15 00:00:00.000	3
6	10253	HANAR	3	1996-07-10 00:00:00.000	1996-07-24 00:00:00.000	1996-07-16 00:00:00.000	2
7	10248	VINET	5	1996-07-04 00:00:00.000	1996-08-01 00:00:00.000	1996-07-16 00:00:00.000	3
8	10256	WELLI	3	1996-07-15 00:00:00.000	1996-08-12 00:00:00.000	1996-07-17 00:00:00.000	2
9	10257	HILAA	4	1996-07-16 00:00:00.000	1996-08-13 00:00:00.000	1996-07-22 00:00:00.000	3
10	10254	CHOPS	5	1996-07-11 00:00:00.000	1996-08-08 00:00:00.000	1996-07-23 00:00:00.000	2

Figure 1.8: Result generated: Date Wise Orders Shipped

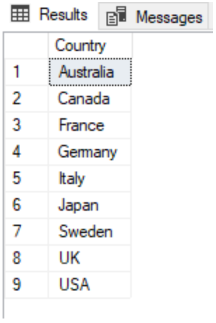
1.6.3 Problem 03

Problem Statement

List name of all countries from where two or more suppliers belong to.

Query

```
SELECT Country
FROM Suppliers
GROUP BY Country
HAVING COUNT(ContactName) >= 2
```



	Country
1	Australia
2	Canada
3	France
4	Germany
5	Italy
6	Japan
7	Sweden
8	UK
9	USA

Figure 1.9: Result generated: Date Wise Orders Shipped

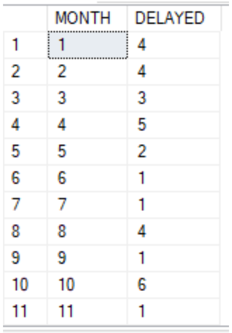
1.6.4 Problem 04

Problem Statement

Write a query to generate report showing month wise orders delayed shipped.

Query

```
SELECT MONTH(ShippedDate) AS MONTH,
COUNT(OrderID) AS DELAYED
FROM Orders
WHERE RequiredDate ¡ ShippedDate
GROUP BY MONTH(ShippedDate)
```



	MONTH	DELAYED
1	1	4
2	2	4
3	3	3
4	4	5
5	5	2
6	6	1
7	7	1
8	8	4
9	9	1
10	10	6
11	11	1

Figure 1.10: Result generated: Month Wise Delayed Orders

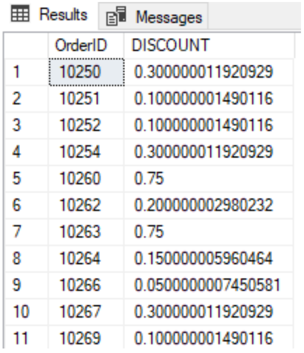
1.6.5 Problem 05

Problem Statement

Report all the orders which have been discounted. Your result should show the total discount against each order.

Query

```
SELECT OrderID, SUM(Discount) AS DISCOUNT
FROM Order Details
GROUP BY OrderID
HAVING SUM(Discount) <> 0
```



	OrderID	DISCOUNT
1	10250	0.300000011920929
2	10251	0.100000001490116
3	10252	0.100000001490116
4	10254	0.300000011920929
5	10260	0.75
6	10262	0.200000002980232
7	10263	0.75
8	10264	0.150000005960464
9	10266	0.0500000007450581
10	10267	0.300000011920929
11	10269	0.100000001490116

Figure 1.11: Result generated: Discounted Orders

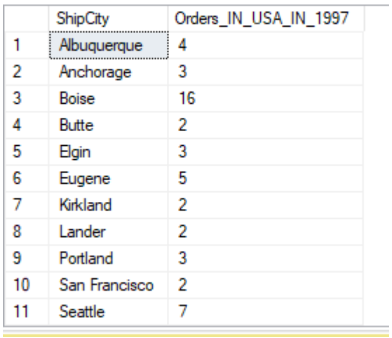
1.6.6 Problem 06

Problem Statement

Write a query to list the number of orders which were shipped in the cities of USA in 1997. Show the number of order against each city.

Query

```
SELECT ShipCity, COUNT(OrderID) Orders_IN_USA_IN_1997
FROM Orders
WHERE ShipCountry = ’USA’ AND YEAR(ShippedDate) =
1997
GROUP BY ShipCity
```



	ShipCity	Orders_IN_USA_IN_1997
1	Albuquerque	4
2	Anchorage	3
3	Boise	16
4	Butte	2
5	Elgin	3
6	Eugene	5
7	Kirkland	2
8	Lander	2
9	Portland	3
10	San Francisco	2
11	Seattle	7

Figure 1.12: Result generated: Orders Shipped in Cities of USA

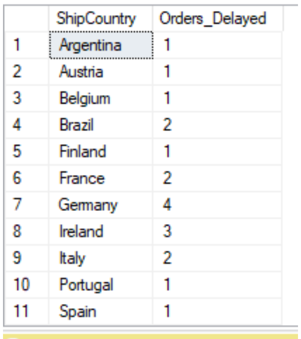
1.6.7 Problem 07

Problem Statement

Write a query to generate report showing country wise orders delayed shipped.

Query

```
SELECT ShipCountry, COUNT(OrderID) AS Orders_Delayed
FROM Orders
WHERE RequiredDate < ShippedDate
GROUP BY ShipCountry
```



	ShipCountry	Orders_Delayed
1	Argentina	1
2	Austria	1
3	Belgium	1
4	Brazil	2
5	Finland	1
6	France	2
7	Germany	4
8	Ireland	3
9	Italy	2
10	Portugal	1
11	Spain	1

Figure 1.13: Result generated: Orders Delayed Country Wise

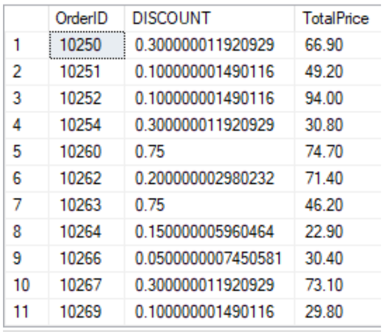
1.6.8 Problem 08

Problem Statement

Report all the orders which have been discounted with total price of order. Your result should show the total discount against each order.

Query

```
SELECT OrderID, SUM(Discount) AS DISCOUNT, SUM(((100-Discount
)/100)*UnitPrice) AS TotalPrice
FROM Order Details
GROUP BY OrderID
HAVING SUM(Discount) <> 0
```



	OrderID	DISCOUNT	TotalPrice
1	10250	0.300000011920929	66.90
2	10251	0.100000001490116	49.20
3	10252	0.100000001490116	94.00
4	10254	0.300000011920929	30.80
5	10260	0.75	74.70
6	10262	0.200000002980232	71.40
7	10263	0.75	46.20
8	10264	0.150000005960464	22.90
9	10266	0.0500000007450581	30.40
10	10267	0.300000011920929	73.10
11	10269	0.100000001490116	29.80

Figure 1.14: Result generated: Discounted Orders with Total Price

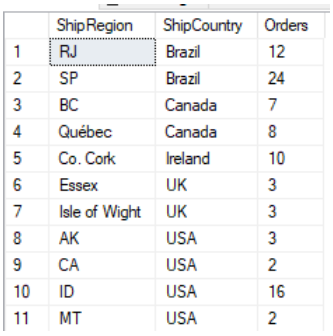
1.6.9 Problem 09

Problem Statement

Write a query to list the number of orders which were shipped in the cities of each region in 1997. Show the number of order against each city.

Query

```
SELECT ShipRegion, ShipCountry, COUNT(OrderID) AS Orders
FROM Orders
WHERE ShipRegion IS NOT NULL AND YEAR(ShippedDate) = 1997
GROUP BY ShipRegion, ShipCountry
```



	ShipRegion	ShipCountry	Orders
1	RJ	Brazil	12
2	SP	Brazil	24
3	BC	Canada	7
4	Québec	Canada	8
5	Co. Cork	Ireland	10
6	Essex	UK	3
7	Isle of Wight	UK	3
8	AK	USA	3
9	CA	USA	2
10	ID	USA	16
11	MT	USA	2

Figure 1.15: Result generated: Orders Shipped in each region