

SMIT CLOUD DATA ENGINEERING BATCH – 3

SQL SERVER QUIZ # 1

QUERIES AND OUTPUT

NAME: HAFIZ MUHAMMAD WAQAS

ROLL NO: 465476

Q1. List top 5 customers by total order amount.

Retrieve the top 5 customers who have spent the most across all sales orders. Show CustomerID, CustomerName, and TotalSpent.

SQL QUERY:

select

TOP 5

c.CustomerID,

c.Name,

sum(s.TotalAmount) as TotalSales

from [dbo].[Customer] c

inner join SalesOrder s

on s.CustomerID = c.CustomerID

group by c.CustomerID, c.Name

order by TotalSales desc;

OUTPUT:

	CustomerID	Name	TotalSales
1	98	Armando	232783.99
2	12	Dwight	175719.20
3	84	Courtney	156259.77
4	90	Chelsae	149290.57
5	27	Lawton	141343.95

Q2. Find the number of products supplied by each supplier.

Display SupplierID, SupplierName, and ProductCount. Only include suppliers that have more than 10 products.

SQL QUERY:

```
select s.SupplierID, s.Name, count(p.ProductID) as ProductsCount
from [dbo].[Supplier] s
inner join [dbo].[PurchaseOrder] po
on s.SupplierID = po.SupplierID
inner join [dbo].[PurchaseOrderDetail] pod
on po.OrderID = pod.OrderID
inner join [dbo].[Product] p
on pod.ProductID = p.ProductID
group by s.SupplierID, s.Name
having count(p.ProductID) > 10;
```

OUTPUT:

Results		Messages	
	SupplierID	Name	ProductsCount
1	1	Schamberger Inc	44
2	2	Bogisich-Block	51
3	3	Abshire Inc	49
4	4	Schmidt LLC	49
5	5	Mann, Will and Fritsch	45
6	6	Abernathy-Waelchi	46
7	7	Kassulke LLC	62
8	8	Lang-Rodriguez	59
9	9	Sawayn-Adams	43
10	10	Stokes, Funk and Herman	52

Q3. Identify products that have been ordered but never returned.

Show ProductID, ProductName, and total order quantity.

SQL QUERY:

```
select p.ProductID, p.Name, sum(sod.Quantity) as TotalOrderQuantity
```

```

from [dbo].[Product] p
left join [dbo].[ReturnDetail] r
on p.ProductID = r.ProductID
inner join [dbo].[SalesOrderDetail] sod
on p.ProductID = sod.ProductID
WHERE r.ProductID is null
group by p.ProductID, p.Name;

```

OUTPUT:

	ProductID	Name	TotalOrderQuantity
1	3	Sertraline Hydrochloride	20
2	15	Flecainide Acetate	55

Q4. For each category, find the most expensive product.

Display CategoryID, CategoryName, ProductName, and Price. Use a subquery to get the max price per category.

SQL QUERY:

```

select c.CategoryID, c.Name, p.Name, p.Price
from [dbo].[Category] c
inner join [dbo].[Product] p
on c.CategoryID = p.CategoryID
where p.Price = (select Max(p1.Price) from [dbo].[Product] p1
where p.CategoryID = p1.CategoryID
)order by p.Price desc;

```

OUTPUT:

	Results	Messages			
	CategoryID	Name	Name	Price	
1	7	Fire Protection	Oral Analgesic	2980.75	
2	16	Elevator	Sertraline Hydrochloride	2896.17	
3	17	Termite Control	SohMed Cold Relief	2813.71	
4	13	NULL	COMETRIQ	2778.36	
5	14	Glass & Glazing	Losartan Potassium	2749.20	
6	12	Asphalt Paving	Acetaminophen PM	2677.44	
7	9	Termite Control	Secura Moisturizing Cleanser	2627.59	
8	5	Soft Flooring and Base	Isopropyl Alcohol Surgical Scrub	2554.94	
9	18	Wall Protection	Natural Cherry Honey Herb Throat Drops	2502.47	
10	4	Painting & Vinyl Wall Covering	Focalin	2455.81	
11	3	Framing (Wood)	ILLUMINATING DAILY MOISTURIZER	2437.99	
12	10	Exterior Signage	Flecainide Acetate	2325.87	
13	19	Epoxy Flooring	Entex T	2012.38	
14	15	Hard Tile & Stone	Quetiapine Fumarate	1831.84	
15	2	Landscaping & Irrigation	Betapace AF	1543.64	
16	6	Electrical and Fire Alarm	Fluoxetine	726.23	
17	20	Hard Tile & Stone	Butalbital, Acetaminophen and Caffeine	85.95	

Q5. List all sales orders with customer name, product name, category, and supplier.

For each sales order, display:

OrderID, CustomerName, ProductName, CategoryName, SupplierName, and Quantity.

SQL QUERY:

```
select so.OrderID, cus.Name CustomerName, p.Name ProductName, c.Name
CategoryName, s.Name SupplierName, sod.Quantity
```

```
from [dbo].[Supplier] s
```

```
inner join [dbo].[PurchaseOrder] po
```

```
on s.SupplierID = po.SupplierID
```

```
inner join [dbo].[PurchaseOrderDetail] pod
```

```
on po.OrderID = pod.OrderID
```

```
inner join [dbo].[Product] p
```

```
on pod.ProductID = p.ProductID
```

```
inner join [dbo].[Category] c
```

```
on p.CategoryID = c.CategoryID
```

```
inner join [dbo].[SalesOrderDetail] sod
```

```
on p.ProductID = sod.ProductID
```

```
inner join [dbo].[SalesOrder] so
on sod.OrderID = so.OrderID
inner join [dbo].[Customer] cus
on so.CustomerID = cus.CustomerID;
```

--- OR ----

```
SELECT
    so.OrderID,
    cus.Name AS CustomerName,
    p.Name AS ProductName,
    c.Name AS CategoryName,
    m.Name AS SupplierName,
    sod.Quantity
FROM dbo.SalesOrder so
INNER JOIN dbo.Customer cus
    ON so.CustomerID = cus.CustomerID
INNER JOIN dbo.SalesOrderDetail sod
    ON so.OrderID = sod.OrderID
INNER JOIN dbo.Product p
    ON sod.ProductID = p.ProductID
INNER JOIN dbo.Category c
    ON p.CategoryID = c.CategoryID
INNER JOIN dbo.Manufacturer m
    ON p.ManufacturerID = m.ManufacturerID;
```

OUTPUT:

OrderID	CustomerName	ProductName	CategoryName	SupplierName	Quantity
1	Armando	Football	Painting & Vinyl Wall Covering	Tender	12
2	Armando	Acetylsalicylic	NULL	Copilot	16
3	Burns	Mallomint Hydrochloride	Temple Control	Mia	21
4	Bary	Molal Complex	Temple Control	Muron	26
5	Bright	KIDS CHOICE	Soft Flooring and Base	Tender	30
6	Bright	Artisan Foam Block	Fire Protection	Platform	26
7	Chelise	SSD Cream	Exterior Signage	Mia	36
8	Chase	Mallomint Hydrochloride	Temple Control	Mia	16
9	Core	Deacetylsalicylic	Landscaping & Irrigation	Copilot	26
10	Core	Sevusa Mordanting Casser	Temple Control	Bloggie	22
11	Subay	Glaxosolone Hydrochloride	Fire Protection	Copilot	22
12	Tramain	Exotic T	Exotic Flooring	Muron	33
13	Kel	Foldal One	Exterior Signage	Tato	15
14	Nela	Austrianophilus PM	Asphalt Paving	Topdrive	21
15	Tetta	Phenothiazine Hydrochloride	Framing (Wood)	Mia	13
16	Murre	Relapace AJ	Landscaping & Irrigation	Tato	24
17	Tetta	Maximum Strength	Framing (Wood)	Paracode	21
18	Lawton	Isopropyl Alcohol Surgical E.	Soft Flooring and Base	Platform	32
19	Louilla	PASPALUM NOTATUM PD.	Glass & Glazing	Topdrive	14
20	Okara	COMETRIQ	NULL	Copilot	16
21	Manate	Fluoxetine	Electrical and Fire Alarm	Copilot	31
22	Lawton	Nature's Cherry Honey Herb	Wall Protection	Topdrive	29
23	Francisco	Orlistatrinol	Hard Tile & Stone	Tato	14
24	Manate	PASPALUM NOTATUM PD.	Glass & Glazing	Topdrive	17
25	Naki	KIDS CHOICE	Soft Flooring and Base	Tender	30
26	Troika	Extra Strength Naprox	Glass & Glazing	Tender	16
27	Saene	Fluoxetine	Electrical and Fire Alarm	Copilot	12
28	Troika	Strong Multivitamin Flavored	Temple Control	Blankos	18

Q6. Find all shipments with details of warehouse, manager, and products shipped.

Display:

ShipmentID, WarehouseName, ManagerName, ProductName, QuantityShipped, and TrackingNumber.

SQL QUERY:

```

select
    s.ShipmentID,
    l.Name Warehouse_Name,
    e.Name Manager_Name,
    p.Name Product_Name,
    sd.Quantity,
    s.TrackingNumber
from [dbo].[Shipment] s
inner join [dbo].[Warehouse] w
on s.WarehouseID = w.WarehouseID
inner join [dbo].[Location] l
on w.LocationID = l.LocationID
inner join [dbo].[Employee] e
on w.ManagerID = e.ManagerID
inner join [dbo].[ShipmentDetail] sd
on s.ShipmentID = sd.ShipmentID

```

inner join [dbo].[Product] p
on sd.ProductID = p.ProductID

OUTPUT:

ShipmentID	Warehouse Name	Manager Name	Product Name	Quantity	TrackingNumber
1	Mr.	Basilio	PASIPALLUM NGISTUM POLLEN	8	NULL
2	Mr.	Dalena	SHREDDO ADVANCED HYDRO-LIQUID COMPACT (PREPUL)	1	NULL
3	Mr.	Aurora	SHREDDO ADVANCED HYDRO-LIQUID COMPACT (PREPUL)	1	NULL
4	Mr.	Kapone	SHREDDO ADVANCED HYDRO-LIQUID COMPACT (PREPUL)	1	NULL
5	Mr.	Basilio	Toy Hot Sack and Large Anise	8	NULL
6	Mr.	Dalena	Natural Chewy Honey Herb Throat Drops	20	NULL
7	Mr.	Aurora	Natural Chewy Honey Herb Throat Drops	20	NULL
8	Mr.	Kapone	Natural Chewy Honey Herb Throat Drops	20	NULL
9	Mr.	Dalena	Maximum Strength	15	NULL
10	Mr.	Aurora	Maximum Strength	15	NULL
11	Mr.	Kapone	Maximum Strength	15	NULL
12	Mr.	Basilio	Quilapone Fluoride	21	NULL
13	Mr.	Nelle	Antibacterial Flamingo Herb Sorbent	21	NULL
14	Mr.	Basilio	PASIPALLUM NGISTUM POLLEN	1	NULL
15	Mr.	Dalena	Ginsawson Hydrochloride	30	NULL
16	Mr.	Aurora	Ginsawson Hydrochloride	30	NULL
17	Mr.	Kapone	Ginsawson Hydrochloride	30	NULL
18	Mr.	Basilio	RELIGIA RAL POLLEN	8	NULL
19	Mr.	Nelle	Wollipson Hydrochloride	28	NULL
20	Mr.	Dalena	Midi Complete	25	NULL
21	Mr.	Aurora	Midi Complete	25	NULL
22	Mr.	Kapone	Midi Complete	25	NULL
23	Mr.	Dalena	Antibac Foam Wash	27	NULL

Q7. Find the top 3 highest-value orders per customer using RANK(). Display CustomerID, CustomerName, OrderID, and TotalAmount.

SQL QUERY:

with cte as (

select

c.CustomerID,

c.Name,

so.OrderID,

so.TotalAmount,

RANK() OVER (PARTITION BY so.CustomerID ORDER BY so.TotalAmount DESC) rnk

from

[dbo].[Customer] c

inner join [dbo].[SalesOrder] so

on c.CustomerID = so.CustomerID

)

select *

from

cte

where rnk <= 3

OUTPUT:



	CustomerID	Name	OrderID	TotalAmount	rnk
1	1	Osborne	97	20649.27	1
2	2	Navajo	66	93376.84	1
3	3	Navajo	83	40081.83	2
4	4	Osborne	9	29706.66	1
5	8	Murphy	18	32848.15	1
6	7	Liam	91	73182.12	1
7	11	Dora	45	51943.81	1
8	12	Dwight	5	51621.22	1
9	12	Dwight	52	99504.03	2
10	12	Dwight	6	24563.86	3
11	13	Juliana	48	49724.66	1
12	18	Rayna	27	22066.23	1
13	19	Renee	84	70566.26	1
14	19	Renee	74	61123.66	2
15	21	Lucas	96	50207.46	1
16	21	Lucas	97	44102.74	2
17	24	Eddy	82	30551.84	1
18	25	Leif	41	58133.77	1
19	25	Leif	47	32660.66	2
20	29	Talbot	66	62369.66	1
21	29	Talbot	80	20783.81	2
22	27	Lauron	71	83375.16	1
23	27	Lauron	22	21801.84	2

Q8. For each product, show its sales history with the previous and next sales quantities (based on order date). Display ProductID, ProductName, OrderID, OrderDate, Quantity, PrevQuantity, and NextQuantity.

SQL QUERY:

select

p.ProductID,

p.Name,

sod.OrderID,

so.OrderDate,

sod.Quantity,

LAG(sod.Quantity) OVER (PARTITION BY p.ProductID ORDER BY so.OrderDate)
PrevQuantity,

LEAD(sod.Quantity) OVER (PARTITION BY p.ProductID ORDER BY so.OrderDate)
NextQuantity

from

[dbo].[SalesOrderDetail] sod

inner join [dbo].[Product] p

on sod.ProductID = p.ProductID

inner join [dbo].[SalesOrder] so

on sod.OrderID = so.OrderID;

OUTPUT:

ProductID	Name	OrderID	OrderDate	Quantity	PrevQuantity	NewQuantity
1	Buataolol: Acetaminophen and Caffeine	79	2003-04-21 04:11:26	19	NULL	17
2	Buataolol: Acetaminophen and Caffeine	81	2016-06-18 03:27:40	17	79	15
3	Buataolol: Acetaminophen and Caffeine	88	2017-03-11 14:02:27	13	17	NULL
4	IBUPROFEN ADVANCED HYDRO LIQUID COMPACT (REFILL)	80	2009-12-26 15:50:29	25	NULL	NULL
5	Betaxolol Hydrochloride	76	2009-10-24 02:50:28	20	NULL	NULL
6	Sol-Med Cold Relief	72	2003-11-14 11:53:28	24	NULL	14
7	Sol-Med Cold Relief	44	2004-04-04 09:26:01	14	24	11
8	Sol-Med Cold Relief	55	2010-04-04 07:46:19	11	14	24
9	Sol-Med Cold Relief	70	2021-08-20 09:18:44	24	11	NULL
10	Propofolone Hydrochloride	16	2009-05-26 09:37:58	13	NULL	21
11	Propofolone Hydrochloride	80	2017-08-06 16:44:19	23	73	NULL
12	Ondansetron	21	2002-02-08 08:44:18	14	NULL	24
13	Ondansetron	88	2011-05-18 11:07:42	24	14	37
14	Ondansetron	23	2013-01-20 02:56:24	37	24	18
15	Ondansetron	50	2020-10-08 05:32:26	25	37	NULL
16	TopCase Salicyl Colones	43	2007-12-19 07:06:49	15	NULL	29
17	TopCase Salicyl Colones	94	1998-03-01 03:56:21	29	15	NULL
18	Maximum Strength	17	2019-07-29 20:13:49	21	NULL	NULL
19	Betrix Masking Creams	58	2003-01-18 13:11:31	34	NULL	18
20	Betrix Masking Creams	28	2014-10-07 17:30:02	18	34	22
21	Betrix Masking Creams	19	2020-05-22 22:34:14	22	18	NULL
22	SALICOLA KALI POLLEN	29	2000-10-01 11:47:48	37	NULL	NULL
23	Extra Strength Mopex	26	2004-10-18 10:46:47	15	NULL	18

Q9. Create a view named vw_CustomerOrderSummary that shows for each customer:

CustomerID, CustomerName, TotalOrders, TotalAmountSpent, and LastOrderDate.

SQL QUERY:

create view vw_CustomerOrderSummary

as

select

c.CustomerID,

c.Name,

COUNT(so.OrderID) TotalOrders,

SUM(so.TotalAmount) TotalAmount,

MAX(so.OrderDate) LastOrderDate

from

[dbo].[Customer] c

inner join [dbo].[SalesOrder] so

on c.CustomerID = so.CustomerID

group by

c.CustomerID,

c.Name;

select * from vw_CustomerOrderSummary;

OUTPUT:

Results		Messages			
	CustomerID	Name	TotalOrders	TotalRevenue	LastOrderDate
1	2	Deane	1	29546.27	2008-12-11 11:30:20
2	3	Neve	2	89452.79	2010-11-30 13:33:08
3	4	Fuscoe	1	39706.68	2010-12-22 18:57:18
4	6	Morley	1	12848.18	2008-07-08 01:10:56
5	7	Lynn	1	79192.12	2012-03-18 18:36:10
6	11	Ray	1	51043.81	2007-12-19 07:00:49
7	12	Deight	2	176719.28	2009-07-21 05:01:11
8	13	Lillem	1	46724.08	2012-12-21 02:36:15
9	18	Stevens	1	22866.23	2012-11-18 00:14:26
10	19	Kerr	2	137718.85	2007-06-18 06:30:48
11	21	Lorrie	2	94112.23	2009-08-12 14:14:23
12	24	Eddy	1	82561.84	2002-03-20 23:00:58
13	25	Laf	2	90784.27	2009-02-17 04:02:48
14	26	Talbot	2	82132.37	2009-06-08 01:37:32
15	27	Lewton	3	181343.95	2010-02-12 21:50:03
16	28	Swain	3	140188.88	2010-06-28 11:15:00
17	29	Adams	1	82048.84	2009-10-05 05:33:26
18	31	Marcia	2	111524.89	2012-06-05 06:52:53
19	35	Quinn	1	37648.58	2006-03-18 00:05:43
20	36	Walt	1	32551.88	2012-05-03 11:38:48
21	47	Farrar	2	84618.21	2005-11-09 18:07:14
22	38	Jordan	1	39029.91	2012-02-27 21:22:18
23	39	Sherry	1	41216.61	2008-09-01 03:00:21

Query executed successfully.

DESKTOP-H96K00L (76.0 RTM) | DESKTOP-H96K00L\USER (62) | NEXT_SQLA_CUTP : 00:00:00 : 63 mins

Q10. Write a stored procedure `sp_GetSupplierSales` that takes a `SupplierID` as input and returns the total sales amount for all products supplied by that supplier.

SQL QUERY:

```

alter procedure sp_total_sales
    @SupplierID int
as
begin
select
    ISNULL(SUM(pod.TotalAmount), 0) TotalSalesAmount
from
    [dbo].[Supplier] sp
inner join [dbo].[PurchaseOrder] po
on sp.SupplierID = po.SupplierID
inner join [dbo].[PurchaseOrderDetail] pod
on po.OrderID = pod.OrderID
where sp.SupplierID = @SupplierID
end;

exec sp_total_sales 2;

```

OUTPUT:

Results Messages

	CustomerID	Name	TotalOrders	TotalAmount	LastOrderDate
1	2	Diana	1	25549.27	2006-12-11 11:36:28
2	3	Nadia	2	66452.77	2010-11-30 13:33:08
3	4	Eudora	1	28705.50	2019-12-22 19:57:18
4	6	Murray	1	12846.15	2008-07-06 01:18:58
5	7	Lera	1	73182.12	2012-03-18 18:38:18
6	11	Dora	1	91043.81	2007-12-19 07:08:48
7	12	Shantel	3	119719.20	2020-07-21 06:01:11
8	13	Liliana	1	44724.08	2010-12-21 02:35:19
9	16	Deyan	1	22395.23	2022-11-18 00:14:29
10	19	Rania	2	137718.89	2007-06-18 08:30:46
11	21	Lucas	2	94310.23	2005-08-12 14:14:23
12	24	Felix	1	80531.84	2002-03-20 23:06:58
13	26	Leif	1	80784.27	2006-03-17 04:03:45
14	28	Talbot	2	83132.37	2006-08-06 01:37:32
15	27	Leifun	3	141343.96	2019-02-12 21:46:03
16	29	Norma	3	149189.68	2019-06-24 11:15:56
17	29	Alana	1	30248.94	2020-10-08 06:33:28
18	31	Marcia	2	111324.65	2017-06-06 06:52:53
19	35	Quinn	1	57843.33	2006-03-16 09:08:43
20	36	Walt	1	55811.80	2017-05-05 11:38:48
21	37	Frederica	2	84816.31	2009-11-09 16:07:14
22	38	Jocelyn	1	39029.81	2012-02-27 21:22:19
23	39	Benny	1	41318.61	2008-03-01 03:58:21

Query executed successfully.

DESKTOP-H56K00L (16.0 KTM)DESKTOP-H56K00L\USER (62)NEXT_COLA_OUTP00000063 rows

Query executed successfully.

DESKTOP-H56K9DL (16.0 RTM) | DESKTOP-H56K9DL\USER [62] | NEXT_CSXA_OUT | 000000 | 63 rows