SQL CASE STUDY

DATA IN MOTON
TINY SHOP SALES

DATA IN MOTION

PUJAITRI DAS

INTRODUCTION

- The challenge is from Data in Motion,LLC to analyse the Tiny Shop Sales (SQL case study 1).
- The dataset contains 4 tables customers,products,orders,order_items.
- Tool Used : MySQL.

#Customers Table

customer_id	first_name	last_name	email
1	John	Doe	johndoe@email.com
2	Jane	Smith	janesmith@email.com
3	Bob	Johnson	bobjohnson@email.com
4	Alice	Brown	alicebrown@email.com
5	Charlie	Davis	charliedavis@email.com
6	Eva	Fisher	evafisher@email.com
7	George	Harris	georgeharris@email.com
8	Ivy	Jones	ivyjones@email.com
9	Kevin	Miller	kevinmiller@email.com
10	Lily	Nelson	lilynelson@email.com
11	Oliver	Patterson	oliverpatterson@email.com
12	Quinn	Roberts	quinnroberts@email.com
13	Sophia	Thomas	sophiathomas@email.com

#Products Table

product_id	product_name	price
1	Product A	10
2	Product B	15
3	Product C	20
4	Product D	25
5	Product E	30
6	Product F	35
7	Product G	40
8	Product H	45
9	Product I	50
10	Product J	55
11	Product K	60
12	Product L	65
13	Product M	70

#Orders Table

order_id	customer_id	order_date
1	1	2023-05-01
2	2	2023-05-02
3	3	2023-05-03
4	1	2023-05-04
5	2	2023-05-05
6	3	2023-05-06
7	4	2023-05-07
8	5	2023-05-08
9	6	2023-05-09
10	7	2023-05-10
11	8	2023-05-11
12	9	2023-05-12
13	10	2023-05-13
14	11	2023-05-14
15	12	2023-05-15
16	13	2023-05-16

#Order_items Table

order_id	product_id	quantity
1	1	2
1	2	1
2	2	1
2	3	3
3	1	1
3	3	2
4	2	4
4	3	1
5	1	1
5	3	2
6	2	3
6	1	1
7	4	1
7	5	2
8	6	3
8	7	1
9	8	2
9	9	1
10	10	3
10	11	2
11	12	1
11	13	3
12	4	2
12	5	1
13	6	3
13	7	2
14	8	1
14	9	2
15	10	3
15	11	1
16	12	2
16	13	3

```sql
#1.Which product has the highest price? Only return a single row.
select product_id,product_name,price
from products
order by price desc
limit 1;
```

product_id	product_name	price
13	Product M	70

```sql
#2.Which customer has made the most orders?
with t as(
select customer_id,concat(first_name,' ' ,last_name) as FullName,
  count(*) as'No.of Orders',
dense_rank() over (order by count(*) desc) as `Rank`
from customers  join orders using(customer_id)
group by customer_id
)
select *
from t
where `Rank`=1;
```

customer_id	FullName	No.of Orders	Rank
1	John Doe	2	1
2	Jane Smith	2	1
3	Bob Johnson	2	1

```sql
#3.What's the total revenue per product?
select product_id,product_name, sum(price*quantity) as Revenue
from products inner join order_items using(product_id)
group by product_id;
```

product_id	product_name	Revenue
1	Product A	50
2	Product B	135
3	Product C	160
4	Product D	75
5	Product E	90
6	Product F	210
7	Product G	120
8	Product H	135
9	Product I	150
10	Product J	330
11	Product K	180
12	Product L	195
13	Product M	420

```sql
#4.Find the day with the highest revenue.
select sum(price*quantity) as Highest_Revenue, order_date
from products inner join order_items using(product_id)
inner join orders using(order_id)
group by order_id,order_date
order by Highest_Revenue desc
limit 1;
```

Highest_Revenue	order_date
340	2023-05-16

```sql
#5.Find the first order (by date) for each customer.
select customer_id, concat(first_name,' ' ,last_name) as FullName,
min(order_date) as `First order date`
from customers inner join orders using(customer_id)
group by customer_id;
```

customer_id	FullName	First order date
1	John Doe	2023-05-01
2	Jane Smith	2023-05-02
3	Bob Johnson	2023-05-03
4	Alice Brown	2023-05-07
5	Charlie Davis	2023-05-08
6	Eva Fisher	2023-05-09
7	George Harris	2023-05-10
8	Ivy Jones	2023-05-11
9	Kevin Miller	2023-05-12
10	Lily Nelson	2023-05-13
11	Oliver Patterson	2023-05-14
12	Quinn Roberts	2023-05-15
13	Sophia Thomas	2023-05-16

```sql
#6.Find the top 3 customers who have ordered the most distinct products.
with t as(
select customer_id,concat(first_name," ", last_name) as fullname,
  count(distinct product_id) as 'No.of Distinct products',
dense_rank() over (order by count(distinct product_id) desc) as `Rank`
from products inner join order_items using(product_id)
  inner join orders using(order_id) inner join customers using(customer_id)
group by customer_id
)
select *
from t
where `Rank`=1;
```

customer_id	fullname	No.of Distinct products	Rank
1	John Doe	3	1
2	Jane Smith	3	1
3	Bob Johnson	3	1

```sql
#7.Which product has been bought the least in terms of quantity?
with t as(
select product_name, sum(quantity),
dense_rank() over (order by sum(quantity) ) as `rank`
from products inner join order_items using(product_id)
  inner join orders using(order_id)
group by product_id
)
select *
from t
where `rank`=1;
```

product_name	sum(quantity)	rank
Product D	3	1
Product E	3	1
Product G	3	1
Product H	3	1
Product I	3	1
Product K	3	1
Product L	3	1

```sql
#8.What is the median order total?

SET @rowindex := -1;
 with T as(
 select order_id,sum(price*quantity) as total
 from order_items inner join products using(product_id)
 group by order_id
 )
 select
   round(avg(total),2) as Median
from
   (
   select @rowindex:=@rowindex + 1 AS rowindex,total
   from t
    order by total
    ) as d
where
d.rowindex in (floor(@rowindex / 2), ceil(@rowindex / 2));
```

Median
112.50

```sql
/*9.For each order, determine if it was ‘Expensive’ (total over 300),
‘Affordable’ (total over 100), or ‘Cheap’.*/
select order_id,sum(price*quantity) as Total,
case
when sum(price*quantity)>300 then "Expensive"
when sum(price*quantity) >100 then "Affordable"
else "Cheap"
end as Status
from orders inner join order_items using(order_id) inner join
products using(product_id)
group by order_id;
```

order_id	Total	Status
1	35	Cheap
2	75	Cheap
3	50	Cheap
4	80	Cheap
5	50	Cheap
6	55	Cheap
7	85	Cheap
8	145	Affordable
9	140	Affordable
10	285	Affordable
11	275	Affordable
12	80	Cheap
13	185	Affordable
14	145	Affordable
15	225	Affordable
16	340	Expensive

```sql
#10.Find customers who have ordered the product with the highest price.
with t as(
select customer_id,concat(first_name,' ' ,last_name) as FullName,price,
dense_rank() over (order by price desc) as `rank`
from customers inner join orders using(customer_id)
inner join order_items using(order_id) inner join  products using(product_id)
)
select *
from t
where `rank`=1;
```

customer_id	FullName	price	rank
8	Ivy Jones	70	1
13	Sophia Thomas	70	1

INSIGHTS

- Product M has the highest price.
- John Doe, Jane Smith and Bob Johnson had the most orders.
- Highest revenue was placed in 2023-05-16.
- John Doe, Jane Smith and Bob Johnson had ordered the most distinct products.
- Median order total was $112.50 .
- Ivy Jones and Sophia Thomas ordered the product with highest price.

THANK YOU