SQL CASE STUDY

TINY SHOP SALES

Hashmeet Kaur

Data in Motion

CONTENTS

INTRODUCTION

This challenge is introduced by Data In Motion, to analyze and derive insights of Tiny Shop Sales data, with the help of SQL Queries and get exposed to following areas :

- Basic Aggregations
- CASE WHEN Statements
- Window Functions
- Joins
- Date Time Functions
- CTEs

The dataset link : SQL Case Study 1: Tiny Shop Sales – Data in Motion (d-i-motion.com)

DATASETS

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....
12	Quinn	Roberts	quinnroberts@email.com
13	Sophia	Thomas	sophiathomas@email.com

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

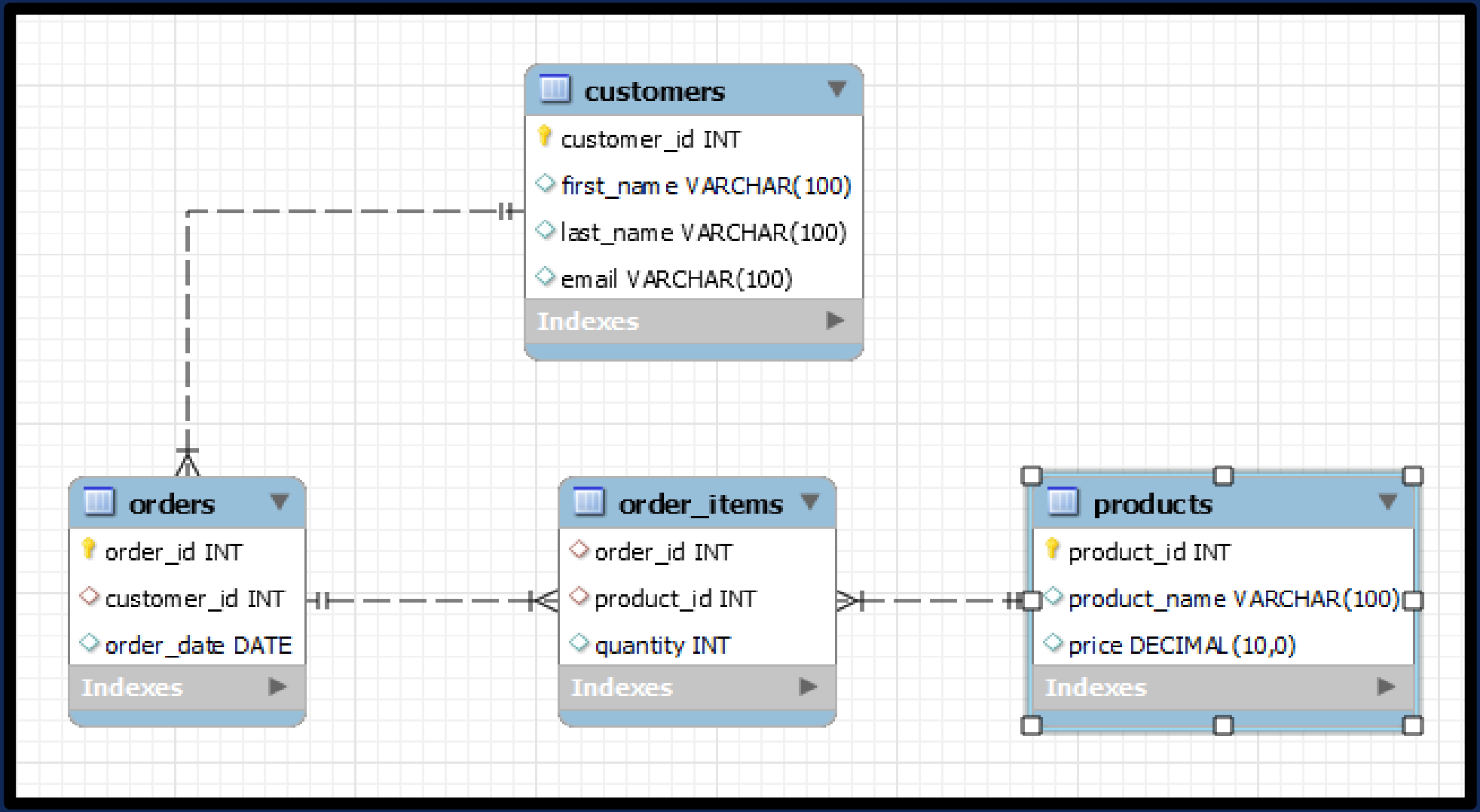
ORDER ITEMS TABLE

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

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

ER DIAGRAM


customers
customer_id INT
first_name VARCHAR(100)
last_name VARCHAR(100)
email VARCHAR(100)
Indexes
orders
order_id INT
customer_id INT
order_date DATE
Indexes
order_items
order_id INT
product_id INT
quantity INT
Indexes
products
product_id INT
product_name VARCHAR(100)
price DECIMAL(10,0)
Indexes

FINDING INSIGHTS

- Which product has the highest price? Only return a single row.
- Which customer has made the most orders?
- What's the total revenue per product?
- Find the day with the highest revenue.
- Find the first order (by date) for each customer.
- Find the top 3 customers who have ordered the most distinct products
- Find the top 3 customers who have ordered the most distinct products
- What is the median order total?
- For each order, determine if it was 'Expensive' (total over 300), 'Affordable' (total over 100), or 'Cheap'.
- Find customers who have ordered the product with the highest price.

1) WHICH PRODUCT HAS THE HIGHEST PRICE? ONLY RETURN A SINGLE ROW.

```
#1) Which product has the highest price? Only return a single row.

    Select * from products order by price desc limit 1;

    #OR

    Select product_id, product_name, price from products where price = (Select Max(price) from products) ;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

product_id	product_name	price
13	Product M	70
NULL	NULL	NULL

2) WHICH CUSTOMER HAS MADE THE MOST ORDERS?

```
#2) Which customer has made the most orders?

    with get_detail as ( select c.*,  count(*) as Orders
    from customers c
    inner join orders o
    on c.customer_id = o.customer_id
    group by 1 )

   ,  max_order_details as ( select max(Orders) as Max_Order from get_detail )
    select a.*
    from get_detail a inner join max_order_details b
    on a.Orders = b.Max_order
    order by a.customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	email	Orders
1	John	Doe	johndoe@email.com	2
2	Jane	Smith	janesmith@email.com	2
3	Bob	Johnson	bobjohnson@email.com	2

3) WHAT'S THE TOTAL REVENUE PER PRODUCT?

```
#3) What's the total revenue per product?

        With db as ( select product_id, sum(quantity) as quantity
        from order_items
        group by product_id
)
        select p.product_id, p.product_name, p.price, d.quantity, (d.quantity * p.price ) AS PRICE
        from db d
        inner join products p
        on d.product_id = p.product_id
        order by 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	product_name	price	quantity	PRICE
1	Product A	10	5	50
2	Product B	15	9	135
3	Product C	20	8	160
4	Product D	25	3	75
5	Product E	30	3	90
6	Product F	35	6	210
7	Product G	40	3	120
8	Product H	45	3	135
9	Product I	50	3	150
10	Product J	55	6	330
11	Product K	60	3	180
12	Product L	65	3	195
13	Product M	70	6	420

4) FIND THE DAY WITH THE HIGHEST REVENUE.

```
#4) Find the day with the highest revenue.

select o.order_date, sum(oi.quantity * p.price) as revenue
from orders o
join order_items oi
on o.order_id = oi.order_id
join products p
on p.product_id = oi.product_id
group by o.order_date
order by revenue desc
limit 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch

order_date	revenue
2023-05-16	340

5) FIND THE FIRST ORDER (BY DATE) FOR EACH CUSTOMER.

```
#5) Find the first order (by date) for each customer.

    with db as ( select c.*, o.order_date,
    Dense_rank () OVER ( partition by customer_id order by order_date) as Rank_order
    from customers c
     inner join orders o
     on c.customer_id = o.customer_id
     order by 1  )
  select d.customer_id, d.first_name, d.last_name, d.email, d.order_date
  from db d
  where Rank_order = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	email	order_date
1	John	Doe	johndoe@email.com	2023-05-01
2	Jane	Smith	janesmith@email.com	2023-05-02
3	Bob	Johnson	bobjohnson@email.com	2023-05-03
4	Alice	Brown	alicebrown@email.com	2023-05-07
5	Charlie	Davis	charliedavis@email.com	2023-05-08
6	Eva	Fisher	evafisher@email.com	2023-05-09
7	George	Harris	georgeharris@email....	2023-05-10
8	Ivy	Jones	ivyjones@email.com	2023-05-11
9	Kevin	Miller	kevinmiller@email.com	2023-05-12
10	Lily	Nelson	lilynelson@email.com	2023-05-13
11	Oliver	Patterson	oliverpatterson@em...	2023-05-14
12	Quinn	Roberts	quinnroberts@email....	2023-05-15
13	Sophia	Thomas	sophiathomas@email...	2023-05-16

6) FIND THE TOP 3 CUSTOMERS WHO HAVE ORDERED THE MOST DISTINCT PRODUCTS

```
#6) Find the top 3 customers who have ordered the most distinct products

select c.customer_id, c.first_name, c.last_name,  count(distinct(oi.product_id)) as Orders
from customers c
inner join orders o
inner join order_items oi
on c.customer_id = o.customer_id
and o.order_id = oi.order_id
group by c.customer_id
order by Orders desc limit 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

customer_id	first_name	last_name	Orders
1	John	Doe	3
2	Jane	Smith	3
3	Bob	Johnson	3

7) WHICH PRODUCT HAS BEEN THE LEAST IN TERMS OF QUANTITY?

```
#7) Which product has been bought the least in terms of quantity?

    with db as (select p.product_id,p.product_name, sum(quantity) as quantity_sold
    from products p
    inner join order_items o
    on p.product_id = o.product_id
    group by product_id
    order by quantity_sold )

    , db2 as ( select min(quantity_sold) as min_qs from db)

    select product_id,product_name, min_qs
    from db a
    inner join db2 b
    on a.quantity_sold = b.min_qs;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	product_name	min_qs
4	Product D	3
5	Product E	3
7	Product G	3
8	Product H	3
9	Product I	3
11	Product K	3
12	Product L	3

8) WHAT IS THE MEDIAN ORDER TOTAL?

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

with db as (select o.order_id, sum(p.price* o.quantity) as revenue
from order_items o inner join products p on o.product_id = p.product_id
group by 1)

, ranked_data as(SELECT revenue,
    ROW_NUMBER() OVER (ORDER BY revenue) AS row_num,
    COUNT(*) OVER () AS total_rows
 FROM db)

SELECT AVG(revenue) AS Median
FROM ranked_data
WHERE row_num IN ((total_rows + 1) / 2, (total_rows + 2) / 2);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Median
140.0000

9) FOR EACH ORDER, DETERMINE IF IT WAS ‘EXPENSIVE’ (TOTAL OVER 300), ‘AFFORDABLE’ (TOTAL OVER 100), OR ‘CHEAP’.

```
#9) For each order, determine if it was ‘Expensive’ (total over 300), ‘Affordable’ (total over 100), or ‘Cheap’.

with db as (select o.order_id, sum(p.price* o.quantity) as revenue
from order_items o inner join products p on o.product_id = p.product_id group by 1 order by 2)

select d.*, Case
when revenue > 300 then "Expensive"
when revenue > 100 then "Affordable"
Else "Cheap" end as Type_of_order from db d;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

10) FIND CUSTOMERS WHO HAVE ORDERED THE PRODUCT WITH THE HIGHEST PRICE.

```
#10) Find customers who have ordered the product with the highest price

with db as (
select a.customer_id, b.order_id, d.product_id, d.product_name,  sum(c.quantity* d.price) as Total, d.price,
dense_rank () over (order by sum(c.quantity* d.price) desc) as Rankk
from customers a inner join orders b inner join order_items c inner join products d
on a.customer_id = b.customer_id
and b.order_id = c.order_id
and c.product_id = d.product_id
group by a.customer_id, b.order_id,
    d.product_id, d.product_name)

    select d.customer_id,concat(c.first_name, " ", c.last_name) as Name, d.order_id, d.product_id, d.product_name,d.price,d.Total, d.Rankk
    from db d
    inner join customers c
    on d.customer_id = c.customer_id
    order by d.Rankk limit 2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	Name	order_id	product_id	product_name	price	Total	Rankk
8	Ivy Jones	11	13	Product M	70	210	1
13	Sophia Thomas	16	13	Product M	70	210	1

SUMMARY

- Product M shows the highest price item.
- John Doe, Jane Smith, and Bob Jhonson are the customers who have made the maximum orders overall, and distinct orders.
- Highest Revenue made in the shop was on 16 May 2023.
- The median order total with overall sales is $140 that help us to find out the transactions made by the shop.
- Ivy Jones, and Sophia Thomas are the customers who have ordered products with highest price.

THANK YOU

Hashmeet Kaur

hashmeetk.hk@gmail.com

Hashmeet (HK) Kaur | LinkedIn

hashmeet22 (HASHMEET KAUR) (github.com)