TINY SHOP SALES

DATA MOTION CASE STUDY 1

Functions Used:

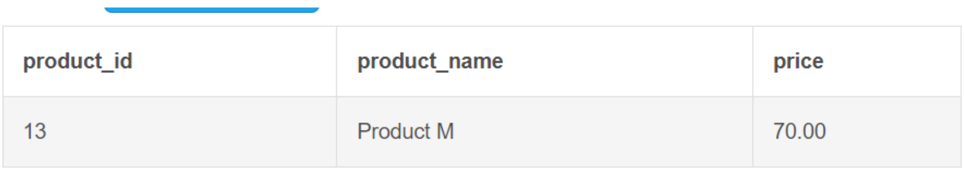
- CTE
- GROUP BY ()
- RANK()
- COUNT
- CASE
- JOIN
- IN
- WHERE

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

Solution:

```
select product_name from products
where price in (select max(price) from
products)
```

Output:



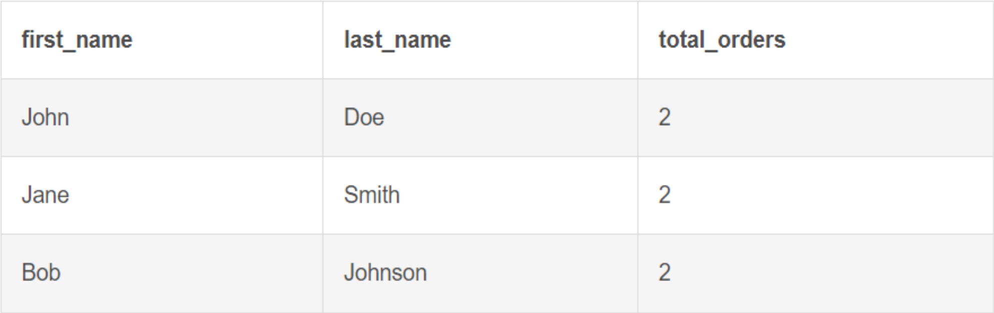
product_id	product_name	price
13	Product M	70.00

2. WHICH CUSTOMER HAS MADE THE MOST ORDERS?

Solution:

```
with order_details as(

   select c.customer_id, count(o.order_id) total_orders

   from orders o  join customers c on o.customer_id = c.customer_id

   group by c.customer_id

   order by total_orders desc

),

rank_data as(select first_name, last_name, total_orders, rank() over
    (order by o.total_orders desc) as rank

          from order_details o join customers c on o.customer_id =
    c.customer_id

)

select first_name, last_name, total_orders  from rank_data

where rank = 1
```

Output:



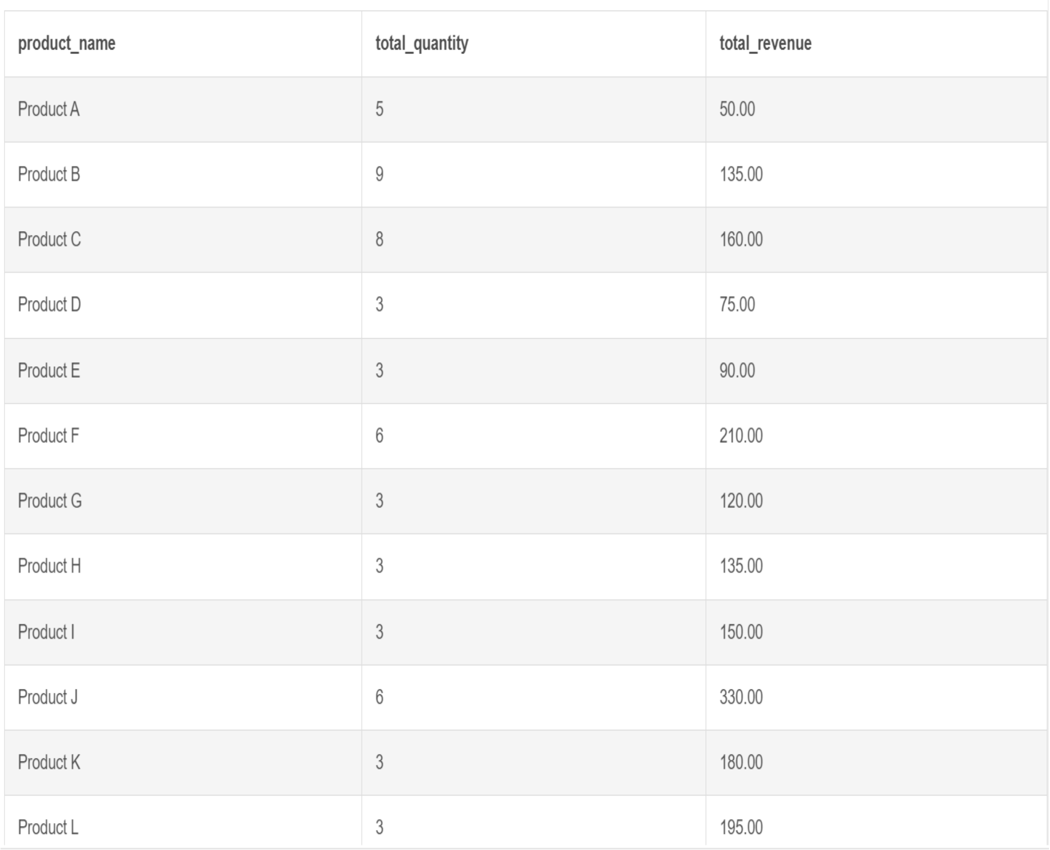
first_name	last_name	total_orders
John	Doe	2
Jane	Smith	2
Bob	Johnson	2

3. WHAT'S THE TOTAL REVENUE PER PRODUCT?

Solution:

```
select p.product_name, sum(o.quantity) as
  total_quantity,

    sum(o.quantity * p.price) as
  total_revenue

from order_items o

join products p

on o.product_id = p.product_id

group by p.product_name

order by p.product_name
```

Output:



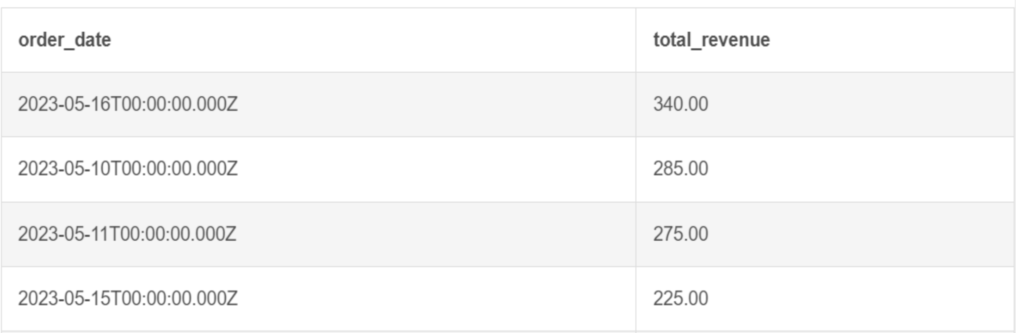
product_name	total_quantity	total_revenue
Product A	5	50.00
Product B	9	135.00
Product C	8	160.00
Product D	3	75.00
Product E	3	90.00
Product F	6	210.00
Product G	3	120.00
Product H	3	135.00
Product I	3	150.00
Product J	6	330.00
Product K	3	180.00
Product L	3	195.00

4. FIND THE DAY WITH THE HIGHEST REVENUE

Solution:

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

Output:



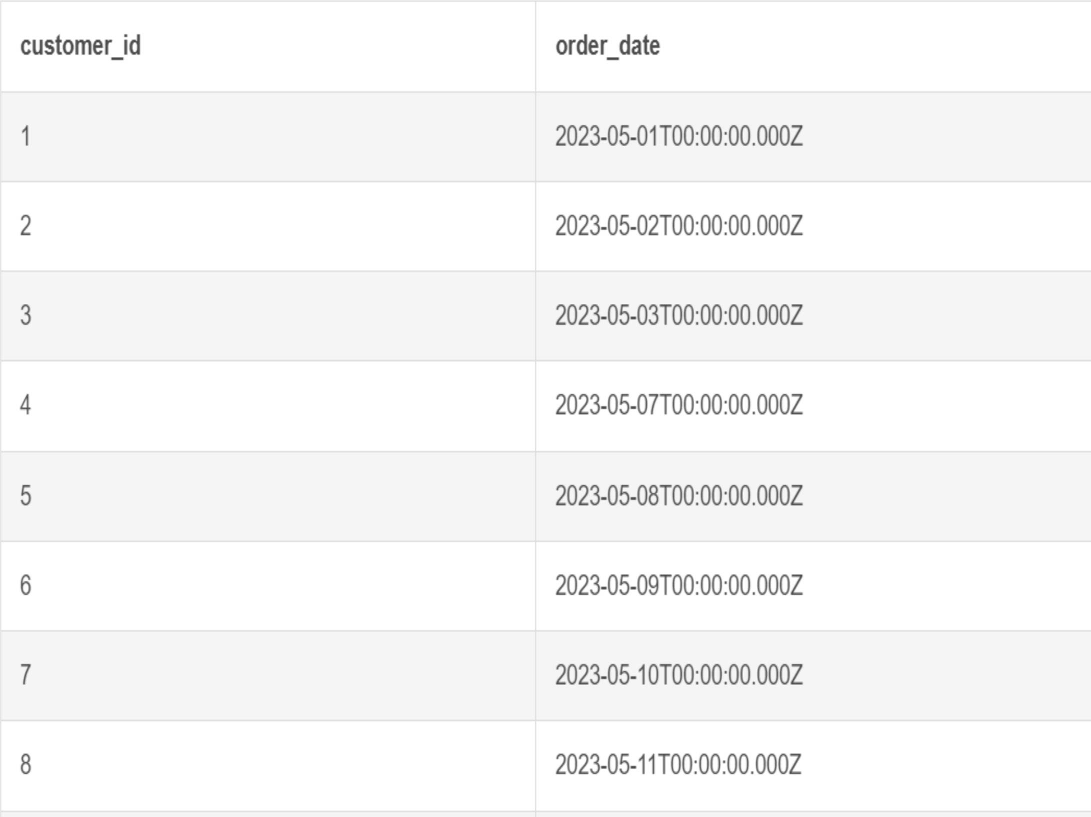
order_date	total_revenue
2023-05-16T00:00:00.000Z	340.00
2023-05-10T00:00:00.000Z	285.00
2023-05-11T00:00:00.000Z	275.00
2023-05-15T00:00:00.000Z	225.00

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

Solution:

```
select customer_id, order_date
from(
   select c.customer_id, o.order_id, o.order_date,
    rank() over (partition by c.customer_id order by
  o.order_date) as order_seq
  from orders o
  join customers c
  on o.customer_id = c.customer_id
  order by c.customer_id, o.order_date) as t
  where t.order_seq = 1;
```

Output:



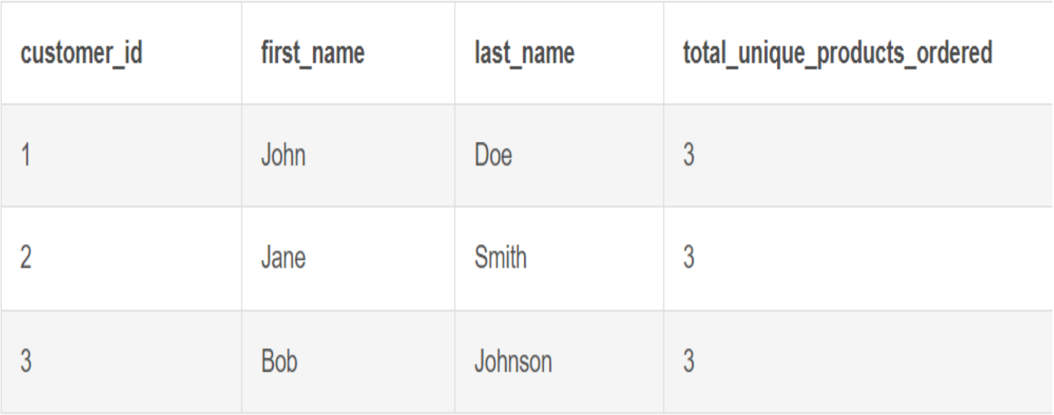
customer_id	order_date
1	2023-05-01T00:00:00.000Z
2	2023-05-02T00:00:00.000Z
3	2023-05-03T00:00:00.000Z
4	2023-05-07T00:00:00.000Z
5	2023-05-08T00:00:00.000Z
6	2023-05-09T00:00:00.000Z
7	2023-05-10T00:00:00.000Z
8	2023-05-11T00:00:00.000Z

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

Solution:

```
with prod_count as (
   select DISTINCT product_name, c.customer_id
  from order_items oi join products p on oi.product_id = p.product_id
  join orders o on oi.order_id = o.order_id
  join customers c on c.customer_id = o.customer_id
  order by c.customer_id
)
select c.customer_id, c.first_name, c.last_name,  count(c.customer_id)
      total_unique_products_ordered
from prod_count p join customers c on c.customer_id = p.customer_id
group by c.customer_id
limit 3
```

Output:



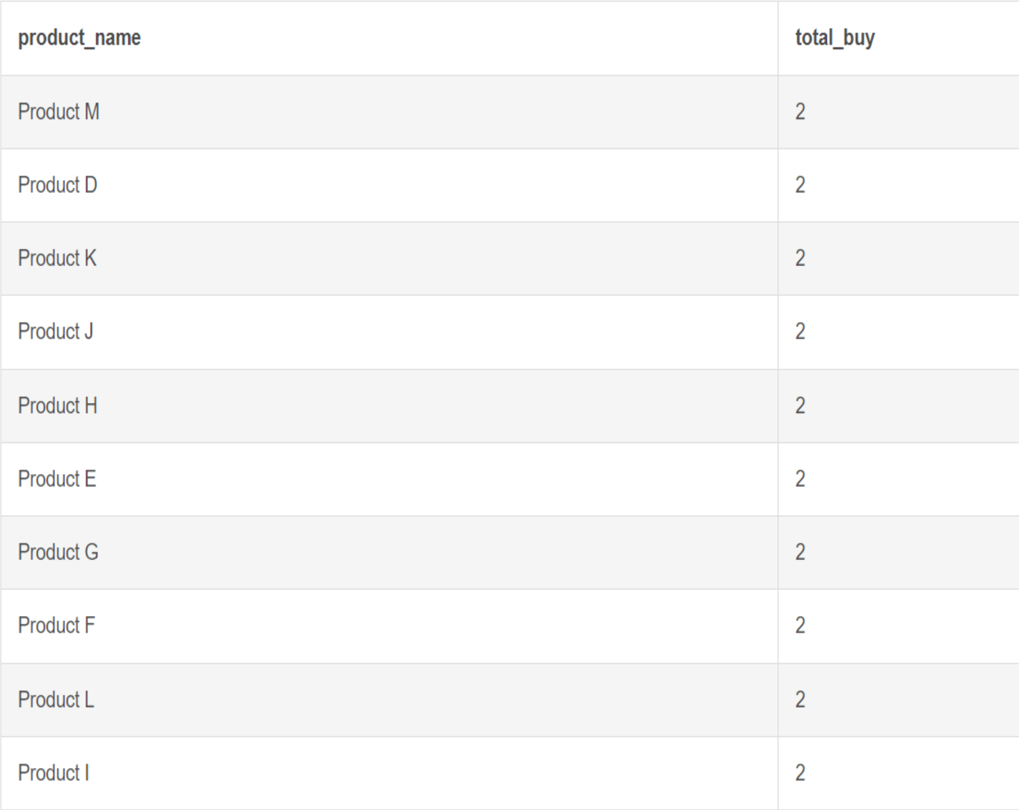
customer_id	first_name	last_name	total_unique_products_ordered
1	John	Doe	3
2	Jane	Smith	3
3	Bob	Johnson	3

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

Solution:

```
with rank_product_purchase as(
      select product_name, count(product_name) total_buy,
         rank () over (order by count(product_name)) rank
   from order_items oi join products p on oi.product_id =
    p.product_id
   join orders o on oi.order_id = o.order_id
   join customers c on c.customer_id = o.customer_id
   group by product_name
)
select product_name, total_buy
from rank_product_purchase
where rank = 1
```

Output:



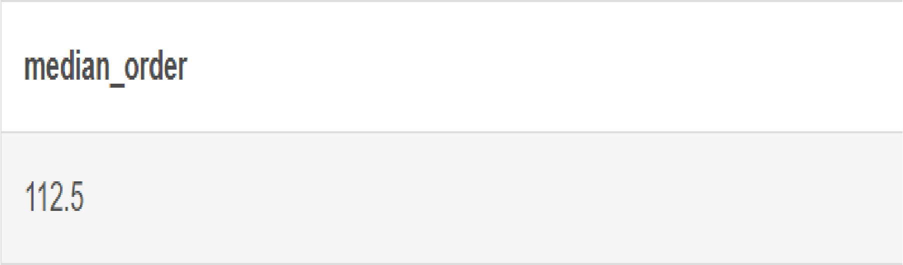
product_name	total_buy
Product M	2
Product D	2
Product K	2
Product J	2
Product H	2
Product E	2
Product G	2
Product F	2
Product L	2
Product I	2

8. WHAT IS THE MEDIAN ORDER TOTAL?

Solution:

```
with order_total as(
    select o.order_id, sum(o.quantity * p.price) total
    from order_items o
    join products p
    on o.product_id = p.product_id
    group by 1
    order by 2
)
select percentile_cont(0.5) within group (order by total) as
     median_order
from order_total
```

Output:



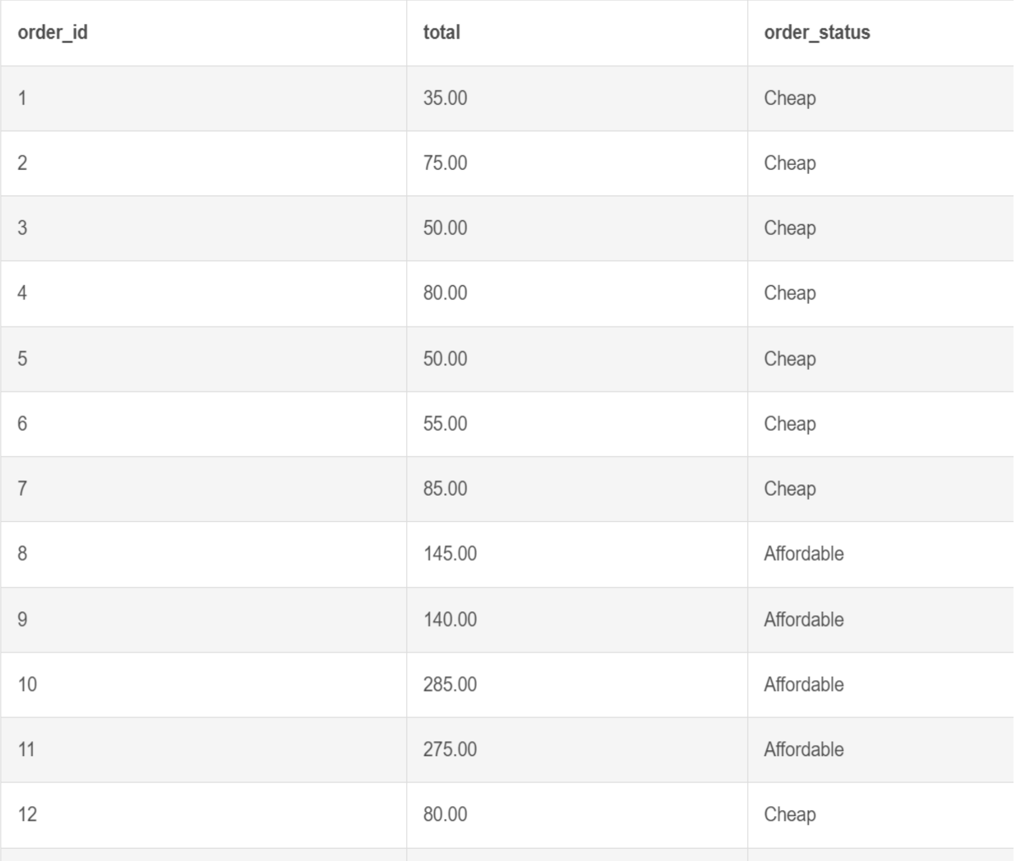
median_order
112.5

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

Solution:

```
with prod_count as (
      select DISTINCT product_name, c.customer_id
    from order_items oi join products p on oi.product_id = p.product_id
    join orders o on oi.order_id = o.order_id
    join customers c on c.customer_id = o.customer_id
    order by c.customer_id
)
select c.customer_id, c.first_name, c.last_name,  count(c.customer_id)
            total_unique_products_ordered
from prod_count p join customers c on c.customer_id = p.customer_id
group by c.customer_id
limit 3
```

Output:



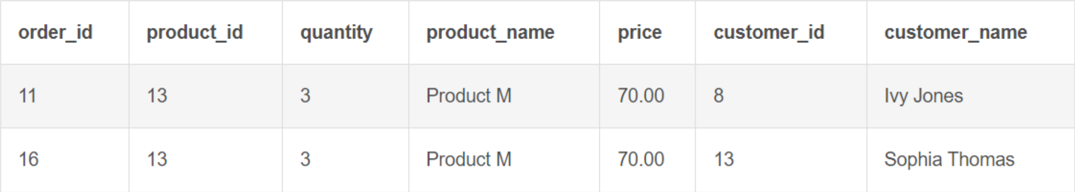
order_id	total	order_status
1	35.00	Cheap
2	75.00	Cheap
3	50.00	Cheap
4	80.00	Cheap
5	50.00	Cheap
6	55.00	Cheap
7	85.00	Cheap
8	145.00	Affordable
9	140.00	Affordable
10	285.00	Affordable
11	275.00	Affordable
12	80.00	Cheap

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

Solution:

```
select oi.order_id, p.product_id, oi.quantity, p.product_name,
  p.price,
   c.customer_id, c.first_name || ' ' || c.last_name
  customer_name

from order_items oi join products p

on oi.product_id = p.product_id

join orders o on oi.order_id = o.order_id

join customers c on c.customer_id = o.customer_id

where p.price = (select max(price) from products)

order by o.order_date
```

Output:



order_id	product_id	quantity	product_name	price	customer_id	customer_name
11	13	3	Product M	70.00	8	Ivy Jones
16	13	3	Product M	70.00	13	Sophia Thomas

TINY SHOP SALES

DATA MOTION CASE STUDY 1

THANK YOU