

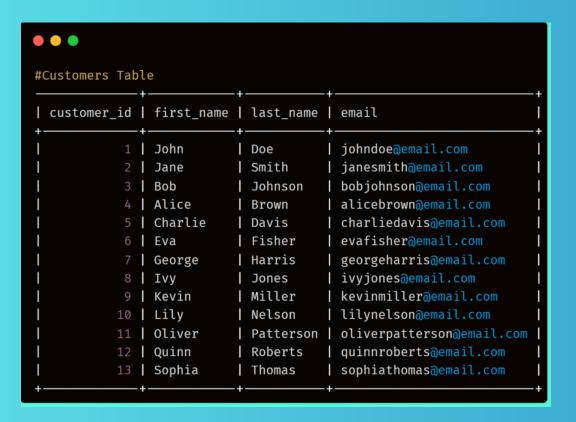


PUJAITRI DAS

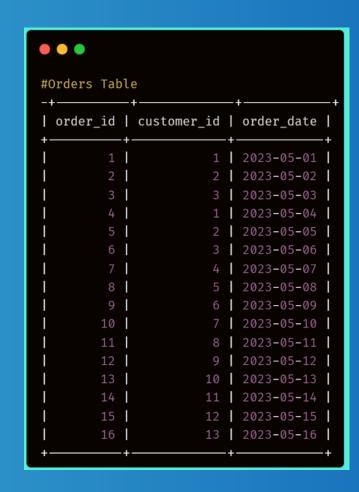


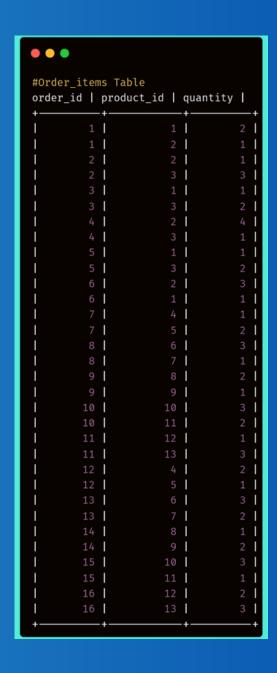
INTRODUCTION

- 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.

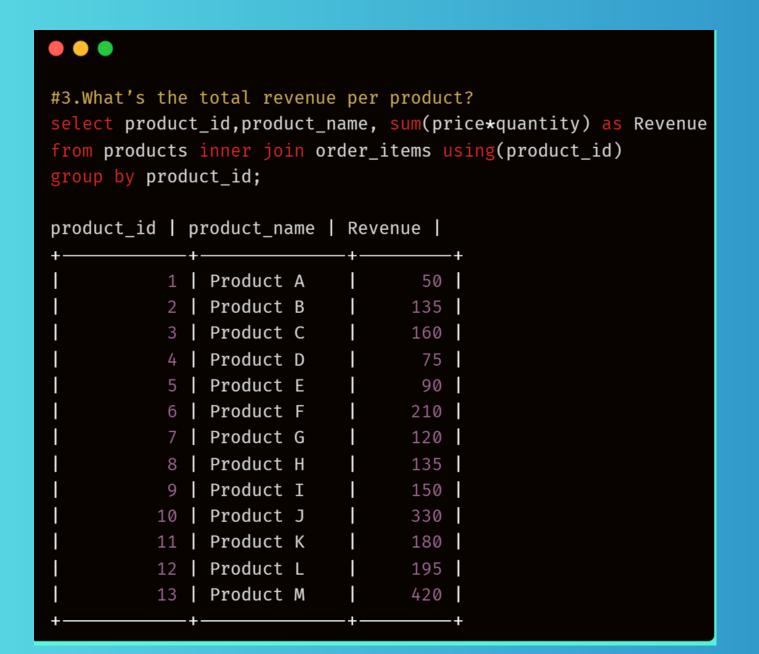








```
#2.Which customer has made the most orders?
with t as(
select customer_id,concat(first_name,' ' ,last_name) as FullName,
 count(*) as'No.of Orders',
dense_rank() over (order by count(*) desc) as `Rank`
from customers join orders using(customer_id)
group by customer_id
select *
from t
where `Rank`=1;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



```
#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;
                               | First order date |
| customer_id | FullName
           1 | John Doe
                                2023-05-01
           2 | Jane Smith
                               2023-05-02
                               2023-05-03
           3 | Bob Johnson
           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
```

```
#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
           2 | Jane Smith |
                                                         1
            3 | Bob Johnson |
```

```
#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
                              1
                         3 | 1 |
| Product E
 Product G
                         3 1
Product H
                         3 | 1 |
Product I
                         3 1
 Product K
                         3 1
                         3 | 1 |
Product L
```

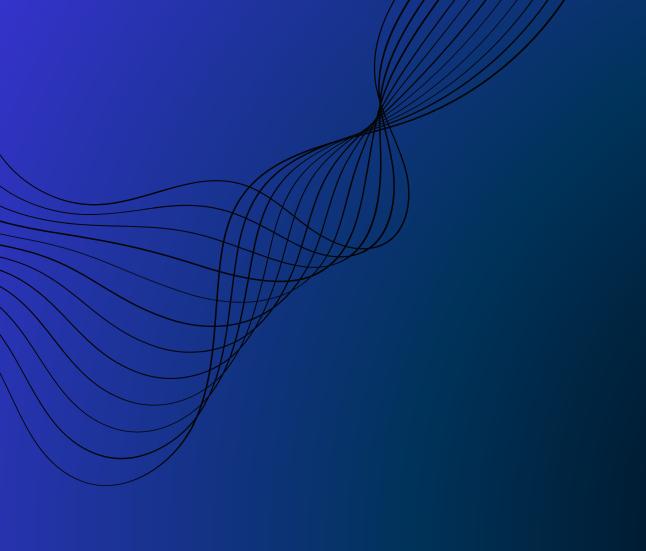
```
#8.What is the median order total?
SET @rowindex := -1;
with T as(
select order_id,sum(price*quantity) as total
from order_items inner join products using(product_id)
 group by order_id
select
  round(avg(total),2) as Median
from
  select @rowindex:=@rowindex + 1 AS rowindex,total
  from t
   order by total
   ) as d
where
d.rowindex in (floor(@rowindex / 2), ceil(@rowindex / 2));
| Median |
112.50
```

```
/*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,
when sum(price*quantity)>300 then "Expensive"
when sum(price*quantity) >100 then "Affordable"
else "Cheap"
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
             75 | Cheap
              50 | Cheap
              80 | Cheap
              50 | Cheap
              55 | Cheap
             85 | Cheap
             145 | Affordable |
             140 | Affordable |
             285 | Affordable |
             275 | Affordable |
       11
       12
              80 | Cheap
             185 | Affordable |
             145 | Affordable |
       15 | 225 | Affordable |
       16 | 340 | Expensive |
```

```
#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
           13 | Sophia Thomas |
                                  70
```

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

