NAME: SHAIK MOHAMMED SAMEER ALI

Write SQL queries for the below mentioned requirements to extract:

1. Top 3 customers based on the amount spent.

A. SELECT c.customer\_id, c.cust\_first\_name, c.cust\_last\_name, SUM(oi.quantity \* oi.unit\_price) AS total\_spent,

DENSE\_RANK() OVER (ORDER BY SUM(oi.quantity \* oi.unit\_price) DESC) AS rank

FROM demo\_customers c

INNER JOIN demo\_orders o ON c.customer\_id = o.customer\_id

INNER JOIN demo\_order\_items oi ON o.order\_id = oi.order\_id

GROUP BY c.customer\_id, c.cust\_first\_name, c.cust\_last\_name

order by total\_spent desc

fetch first 3 row only;

2. Top 3 customers based on the number of orders.

A.

SELECT c.customer\_id, c.cust\_first\_name, c.cust\_last\_name, COUNT(\*) AS num\_orders,

rank() OVER (ORDER BY COUNT(\*) DESC) AS rank

FROM demo\_customers c

INNER JOIN demo\_orders o ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_id, c.cust\_first\_name, c.cust\_last\_name

ORDER BY num\_orders DESC

fetch first 3 row only;

3. Top 3 products based on quantity sold.

A.

SELECT p.product\_id, p.product\_name, SUM(oi.quantity) AS total\_sold, dense\_rank() over (order by SUM(oi.quantity) desc) as rank

FROM demo\_products p

INNER JOIN demo\_order\_items oi ON p.product\_id = oi.product\_id

GROUP BY p.product\_id, p.product\_name

ORDER BY total\_sold DESC

fetch first 3 row only;

4. Write a query to capture the customer's last name, product names (he bought) and total quantity of each of them.

A.

SELECT c.cust\_last\_name, p.product\_name, SUM(oi.quantity) AS total\_quantity

FROM demo\_customers c

INNER JOIN demo\_orders o ON c.customer\_id = o.customer\_id

INNER JOIN demo\_order\_items oi ON o.order\_id = oi.order\_id

INNER JOIN demo\_products p ON oi.product\_id = p.product\_id

GROUP BY c.cust\_last\_name, p.product\_name

ORDER BY c.cust\_last\_name, p.product\_name ;

5. Transform the rows to columns for the result you get from Query # 4 above.

A. SELECT \*

FROM (

SELECT c.cust\_last\_name, p.product\_name, SUM(oi.quantity) AS total\_quantity

FROM demo\_customers c

INNER JOIN demo\_orders o ON c.customer\_id = o.customer\_id

INNER JOIN demo\_order\_items oi ON o.order\_id = oi.order\_id

INNER JOIN demo\_products p ON oi.product\_id = p.product\_id

GROUP BY c.cust\_last\_name, p.product\_name

ORDER BY c.cust\_last\_name, p.product\_name

)

PIVOT(

max(total\_quantity) FOR product\_name IN (

'Business\_Shirt' AS Business\_Shirt,

'Trousers' AS Trousers,

'Jacket' AS Jacket,

'Blouse' AS Blouse,

'Skirt' AS Skirt,

'Ladies\_Shoes' AS Ladies\_Shoes,

'Belt' AS Belt,

'Bag' AS Bag,

'Mens\_Shoes' AS Mens\_Shoes,

'Wallet' AS Wallet ))

ORDER BY 1;

6. Which year had most orders?

A.

SELECT to\_char(order\_timestamp,'yyyy') AS order\_year, COUNT(\*) AS num\_orders

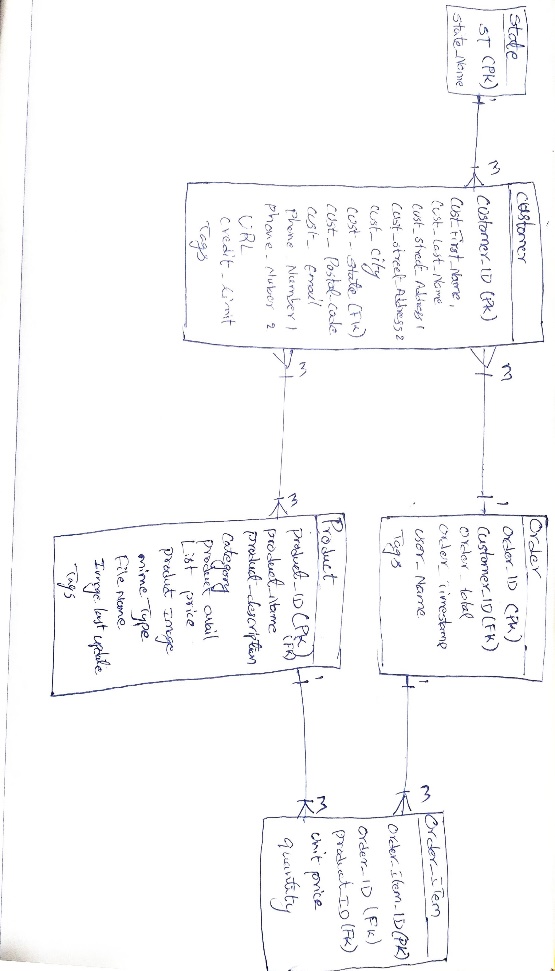
FROM demo\_orders

GROUP BY to\_char(order\_timestamp,'yyyy')

ORDER BY num\_orders DESC

fetch first 1 row only;

7. Draw an ER (entity-relationship) model for this database depicting tables, PK and FK (i.e., relationships between them) on a plain paper using pen/pencil. Share the screenshot (ensure picture is clearly visible) and share.

A.

8. Which product category was most sold?

A. SELECT p.category, SUM(oi.quantity \* oi.unit\_price) AS total\_revenue

FROM demo\_products p

INNER JOIN demo\_order\_items oi ON p.product\_id = oi.product\_id

GROUP BY p.category

ORDER BY total\_revenue DESC

fetch first 1 row only;

9. Which product category took the second position in terms of quantity sold?  
A. SELECT p.category, SUM(oi.quantity) AS total\_sold,

RANK() OVER (ORDER BY SUM(oi.quantity) DESC) AS rank

FROM demo\_products p

INNER JOIN demo\_order\_items oi ON p.product\_id = oi.product\_id

GROUP BY p.category

ORDER BY total\_sold DESC

fetch next 2 row only;

10. Write a query to help rollup total quantity on customer and product (name)

A.SELECT c.cust\_last\_name, p.product\_name, SUM(oi.quantity) AS total\_quantity

FROM demo\_customers c

INNER JOIN demo\_orders o ON c.customer\_id = o.customer\_id

INNER JOIN demo\_order\_items oi ON o.order\_id = oi.order\_id

INNER JOIN demo\_products p ON oi.product\_id = p.product\_id

GROUP BY ROLLUP (c.cust\_last\_name, p.product\_name)

ORDER BY c.cust\_last\_name, p.product\_name;