Oracle SQL Sessions Assignment #3: Final-Project - Orders and items Database

1. Top 3 customers based on the amount spent.
Ans:

```
SELECT CUSTOMER_ID, CUST_LAST_NAME, CUST_FIRST_NAME,
SUM(ORDER_TOTAL) AS TOTAL_AMOUNT_SPENT,
RANK() OVER (ORDER BY SUM(ORDER_TOTAL) DESC) AS RANK
FROM DEMO_CUSTOMERS C
INNER JOIN DEMO_ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID
GROUP BY CUSTOMER_ID, CUST_LAST_NAME, CUST_FIRST_NAME
HAVING RANK <= 3;
```

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

```
SELECT CUSTOMER_ID, CUST_LAST_NAME, CUST_FIRST_NAME, COUNT(ORDER_ID) AS
TOTAL_ORDERS,RANK() OVER (ORDER BY COUNT(ORDER_ID) DESC) AS RANK
FROM DEMO_CUSTOMERS C
INNER JOIN DEMO_ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID
GROUP BY CUSTOMER_ID, CUST_LAST_NAME, CUST_FIRST_NAME
HAVING RANK <= 3;
```

3. Top 3 products based on quantity sold.
Ans:

```
SELECT PRODUCT_ID, PRODUCT_NAME, CATEGORY, SUM(QUANTITY) AS
TOTAL_QUANTITY_SOLD,RANK() OVER (ORDER BY SUM(QUANTITY) DESC) AS RANK
FROM DEMO_PRODUCTS P
INNER JOIN DEMO_ORDER_ITEMS OI ON P.PRODUCT_ID = OI.PRODUCT_ID
GROUP BY PRODUCT_ID, PRODUCT_NAME, CATEGORY
HAVING RANK <= 3;
```

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

```
SELECT C.CUST_LAST_NAME, P.PRODUCT_NAME, SUM(OI.QUANTITY) AS
TOTAL_QUANTITY_BOUGHT,RANK() OVER (PARTITION BY C.CUST_LAST_NAME ORDER
BY SUM(OI.QUANTITY) 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
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, RANK;
```

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

Ans:

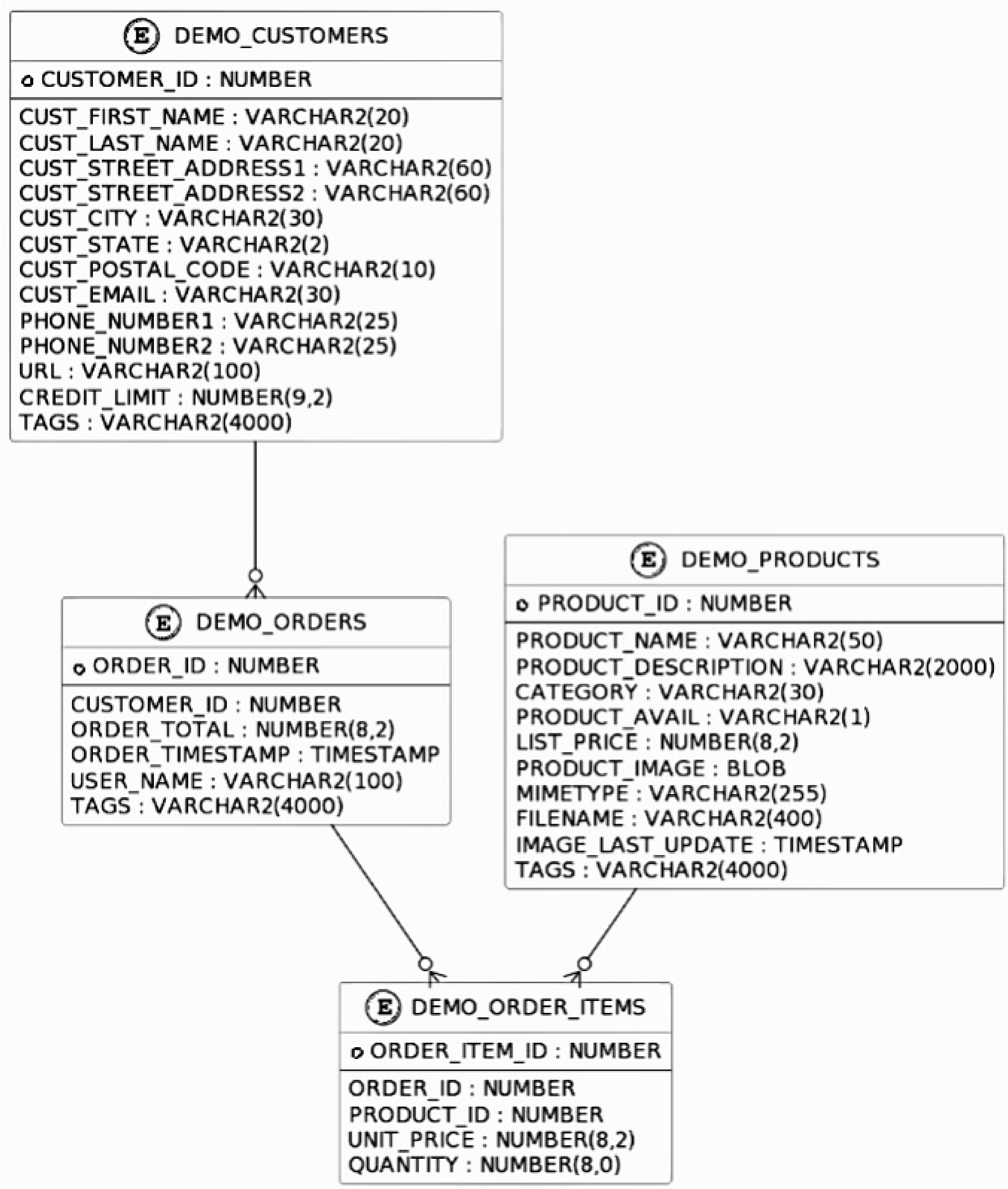
```
SELECT *
FROM (
   SELECT C.CUST_LAST_NAME AS LNAME,
        P.PRODUCT_NAME AS PNAME,
        SUM(OI.QUANTITY) AS TOT_QUANTITY
   FROM DEMO_CUSTOMERS C
   INNER JOIN DEMO_ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID
   INNER JOIN DEMO_ORDER_ITEMS OI ON O.ORDER_ID = OI.ORDER_ID
   INNER JOIN DEMO_PRODUCTS P ON OI.PRODUCT_ID = P.PRODUCT_ID
   GROUP BY C.CUST_LAST_NAME, P.PRODUCT_NAME
)
PIVOT (
   SUM(TOT_QUANTITY) FOR PNAME IN (
      'Business Shirt' AS Business_Shirt,
      'Trousers' AS Trousers,
      'Jacket' AS Jacket,
      'Skirt' AS Skirt,
      'Ladies Shoes' AS Ladies_Shoes,
      'Bag' AS Bag,
      'Mens Shoes' AS Mens_Shoes,
      'Wallet' AS Wallet,
      'Belt' AS Belt,
      'Blouse' AS Blouse
   )
);
```

6. Which year had most orders?

Ans:

```
SELECT EXTRACT(YEAR FROM ORDER_TIMESTAMP) AS ORDER_YEAR,
COUNT(ORDER_ID) AS TOTAL_ORDERS,
RANK() OVER (ORDER BY COUNT(ORDER_ID) DESC) AS RANK
FROM DEMO_ORDERS
GROUP BY EXTRACT(YEAR FROM ORDER_TIMESTAMP)
HAVING RANK = 1;
```

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.
Ans:



8. Which product category was most sold?
Ans:

```
SELECT CATEGORY, SUM(QUANTITY) AS TOTAL_QUANTITY_SOLD,
RANK() OVER (ORDER BY SUM(QUANTITY) DESC) AS RANK
FROM DEMO_PRODUCTS P
INNER JOIN DEMO_ORDER_ITEMS OI ON P.PRODUCT_ID = OI.PRODUCT_ID
GROUP BY CATEGORY
HAVING RANK = 1;
```

9. Which product category took the second position in terms of quantity sold?
Ans:

```
SELECT CATEGORY, SUM(QUANTITY) AS TOTAL_QUANTITY_SOLD,RANK() OVER (ORDER
BY SUM(QUANTITY) DESC) AS CATEGORY_RANK
FROM DEMO_PRODUCTS P
INNER JOIN DEMO_ORDER_ITEMS OI ON P.PRODUCT_ID = OI.PRODUCT_ID
GROUP BY CATEGORY
HAVING CATEGORY_RANK = 2;
```

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

```
SELECT C.CUST_FIRST_NAME, C.CUST_LAST_NAME, P.PRODUCT_NAME,
SUM(I.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 I ON O.ORDER_ID = I.ORDER_ID
INNER JOIN DEMO_PRODUCTS P ON I.PRODUCT_ID = P.PRODUCT_ID
GROUP BY ROLLUP(C.CUST_FIRST_NAME, C.CUST_LAST_NAME, P.PRODUCT_NAME)
ORDER BY C.CUST_FIRST_NAME, C.CUST_LAST_NAME, P.PRODUCT_NAME;
```