

-- sales order management

USE salesorderdb;

-- 1. Which category has maximum products

```
SELECT CATEGORY_CODE , PRODUCT_COUNT
FROM (
    SELECT C.CATEGORY_CODE , COUNT(P.PRODUCT_CODE) AS PRODUCT_COUNT,
    RANK() OVER(ORDER BY COUNT(P.PRODUCT_CODE) DESC) AS Rnk
    FROM CATEGORY C
    JOIN PRODUCT P ON C.CATEGORY_CODE = P.CATEGORY_CODE
    GROUP BY CATEGORY_CODE
) AS ranked_products
WHERE rnk = 1;
```

-- 2. Which category has minimum products?

```
SELECT CATEGORY_CODE , PRODUCT_COUNT
FROM (
    SELECT C.CATEGORY_CODE , COUNT(P.PRODUCT_CODE) AS PRODUCT_COUNT,
    RANK() OVER(ORDER BY COUNT(P.PRODUCT_CODE) ASC) AS Rnk
    FROM CATEGORY C
    JOIN PRODUCT P ON C.CATEGORY_CODE = P.CATEGORY_CODE
    GROUP BY CATEGORY_CODE
) AS ranked_products
WHERE rnk = 1;
```

-- 3. Which category has no products?

-- one way

```
SELECT C.CATEGORY_CODE
FROM CATEGORY C
LEFT JOIN PRODUCT P ON C.CATEGORY_CODE = P.CATEGORY_CODE
WHERE P.PRODUCT_CODE IS NULL;
```

-- Another way

```
SELECT CATEGORY_CODE
```

```
FROM CATEGORY
WHERE CATEGORY_CODE NOT IN (
    SELECT DISTINCT CATEGORY_CODE
    FROM PRODUCT
);
```

-- 4. Which is the costliest product?

```
SELECT PRODUCT_CODE , PROD_DESC, PRICE
FROM PRODUCT
WHERE PRICE = (SELECT MAX(PRICE) FROM PRODUCT);
```

-- 5. Which category has costliest product?

```
SELECT CATEGORY_CODE , PRICE
FROM (
    SELECT C.CATEGORY_CODE , P.PRICE ,
    RANK() OVER(ORDER BY (P.PRICE) DESC) AS rnk
    FROM CATEGORY C
    JOIN PRODUCT P ON C.CATEGORY_CODE = P.CATEGORY_CODE
)AS rnked_price
WHERE rnk = 1;
```

-- 6. Which category has lot of products (with respect to quantity on hand)?

```
SELECT CATEGORY_CODE, TOTAL_QTY
FROM (
    SELECT CATEGORY_CODE,
    SUM(QTY_ON_HAND) AS TOTAL_QTY,
    RANK() OVER (ORDER BY SUM(QTY_ON_HAND) DESC) AS rnk
    FROM PRODUCT
    GROUP BY CATEGORY_CODE
) ranked
WHERE rnk = 1;
```

-- 7. Category wise display the costliest products?

```
SELECT CATEGORY_CODE, PRODUCT_CODE, PROD_DESC, PRICE
```

```
FROM (  
    SELECT CATEGORY_CODE, PRODUCT_CODE, PROD_DESC, PRICE,  
           RANK() OVER (PARTITION BY CATEGORY_CODE ORDER BY PRICE DESC) AS rnk  
    FROM PRODUCT  
) ranked  
WHERE rnk = 1;
```

-- 8. Category wise display the product whose quantity on hand is minimum?

```
SELECT CATEGORY_CODE, PRODUCT_CODE, PROD_DESC, QTY_ON_HAND  
FROM (  
    SELECT CATEGORY_CODE, PRODUCT_CODE, PROD_DESC, QTY_ON_HAND,  
           RANK() OVER (PARTITION BY CATEGORY_CODE ORDER BY QTY_ON_HAND  
ASC) AS rnk  
    FROM PRODUCT  
) ranked  
WHERE rnk = 1;
```

-- 9. Which order has maximum products?

```
SELECT ORDER_CODE, TOTAL_PRODUCTS  
FROM (  
    SELECT ORDER_CODE, COUNT(DISTINCT PRODUCT_CODE) AS TOTAL_PRODUCTS,  
           RANK() OVER (ORDER BY COUNT(DISTINCT PRODUCT_CODE) DESC) AS rnk  
    FROM ORDER_DETAIL  
    GROUP BY ORDER_CODE  
) ranked  
WHERE rnk = 1;
```

-- 10. Which is the frequently ordered product?

```
SELECT PRODUCT_CODE, TOTAL_ORDERS  
FROM (  
    SELECT PRODUCT_CODE, COUNT(ORDER_CODE) AS TOTAL_ORDERS,  
           RANK() OVER (ORDER BY COUNT(ORDER_CODE) DESC) AS rnk
```

```
FROM ORDER_DETAIL
GROUP BY PRODUCT_CODE
) ranked
WHERE rnk = 1;
```

-- 11. Which product is least ordered product?

```
SELECT PRODUCT_CODE, TOTAL_ORDERS
FROM (
    SELECT PRODUCT_CODE, COUNT(ORDER_CODE) AS TOTAL_ORDERS,
           RANK() OVER (ORDER BY COUNT(ORDER_CODE) ASC) AS rnk
    FROM ORDER_DETAIL
    GROUP BY PRODUCT_CODE
) ranked
WHERE rnk = 1;
```

-- 12. What product is not at all ordered?

```
SELECT PRODUCT_CODE, PROD_DESC
FROM PRODUCT
WHERE PRODUCT_CODE NOT IN (
    SELECT DISTINCT PRODUCT_CODE
    FROM ORDER_DETAIL
);
```

-- 13. Which is the costliest order? (Calculate the bill)

```
SELECT ORDER_CODE , TOTAL_COST
FROM (
    SELECT ORDER_CODE , SUM(P.PRICE * O.QTY_ORDERED) AS TOTAL_COST ,
           RANK() OVER(ORDER BY SUM(P.PRICE * O.QTY_ORDERED) DESC) AS rnk
    FROM ORDER_DETAIL O
    JOIN PRODUCT P ON O.PRODUCT_CODE = P.PRODUCT_CODE
    GROUP BY ORDER_CODE
) ranked
```

WHERE rnk = 1;

-- 14. In which date the costliest order made?

```
SELECT ORDER_CODE , ORDER_DATE , TOTAL_COST
FROM ( SELECT O.ORDER_CODE , OM.ORDER_DATE , SUM(P.PRICE * O.QTY_ORDERED) AS
TOTAL_COST ,
        RANK() OVER(ORDER BY SUM(P.PRICE * O.QTY_ORDERED) DESC) AS rnk
FROM ORDER_DETAIL O
        JOIN PRODUCT P ON O.PRODUCT_CODE = P.PRODUCT_CODE
        JOIN ORDER_MASTER OM ON O.ORDER_CODE = OM.ORDER_CODE
GROUP BY O.ORDER_CODE , OM.ORDER_DATE
) Rnked
```

WHERE rnk = 1;

-- 15. Which customer made the costliest order?

```
SELECT CUSTOMER_CODE , CUSTOMER_NAME , TOTAL_COST
FROM (
        SELECT OM.CUSTOMER_CODE , C.CUSTOMER_NAME , SUM(P.PRICE * O.QTY_ORDERED) AS
TOTAL_COST ,
        RANK() OVER(ORDER BY SUM(P.PRICE * O.QTY_ORDERED) DESC) AS rnk
FROM ORDER_DETAIL O
        JOIN PRODUCT P ON O.PRODUCT_CODE = P.PRODUCT_CODE
        JOIN ORDER_MASTER OM ON O.ORDER_CODE = OM.ORDER_CODE
        JOIN CUSTOMER C ON OM.CUSTOMER_CODE = C.CUSTOMER_CODE
GROUP BY OM.CUSTOMER_CODE , C.CUSTOMER_NAME
) rnked
```

WHERE rnk = 1;

-- 16. Which customer made the costliest order today?

```
SELECT CUSTOMER_CODE, CUSTOMER_NAME, TOTAL_COST
FROM (
        SELECT OM.CUSTOMER_CODE, C.CUSTOMER_NAME,
        SUM(P.PRICE * OD.QTY_ORDERED) AS TOTAL_COST,
        RANK() OVER (ORDER BY SUM(P.PRICE * OD.QTY_ORDERED) DESC) AS rnk
FROM ORDER_DETAIL OD
```

```

JOIN PRODUCT P ON OD.PRODUCT_CODE = P.PRODUCT_CODE
JOIN ORDER_MASTER OM ON OD.ORDER_CODE = OM.ORDER_CODE
JOIN CUSTOMER C ON OM.CUSTOMER_CODE = C.CUSTOMER_CODE
WHERE OM.ORDER_DATE = CURDATE()
GROUP BY OM.CUSTOMER_CODE, C.CUSTOMER_NAME

```

) ranked

WHERE rnk = 1;

-- 17. Generate the report like

Customer name	Cust_addr	Order code	Order date	No_of_products_ordered	Total Bill
---------------	-----------	------------	------------	------------------------	------------

SELECT

```

C.CUSTOMER_NAME,
OM.ORDER_CODE,
OM.ORDER_DATE,
COUNT(OD.PRODUCT_CODE) AS NO_OF_PRODUCTS_ORDERED,
SUM(P.PRICE * OD.QTY_ORDERED) AS TOTAL_BILL

```

FROM CUSTOMER C

JOIN ORDER_MASTER OM ON C.CUSTOMER_CODE = OM.CUSTOMER_CODE

JOIN ORDER_DETAIL OD ON OM.ORDER_CODE = OD.ORDER_CODE

JOIN PRODUCT P ON OD.PRODUCT_CODE = P.PRODUCT_CODE

GROUP BY C.CUSTOMER_NAME, OM.ORDER_CODE, OM.ORDER_DATE;

-- 18. Write query the customers who are eligible for home delivery (if tot_bill >= 5000)?

SELECT

```

C.CUSTOMER_NAME,
OM.ORDER_CODE,
OM.ORDER_DATE,
SUM(P.PRICE * OD.QTY_ORDERED) AS TOTAL_BILL

```

FROM CUSTOMER C

JOIN ORDER_MASTER OM ON C.CUSTOMER_CODE = OM.CUSTOMER_CODE

JOIN ORDER_DETAIL OD ON OM.ORDER_CODE = OD.ORDER_CODE

JOIN PRODUCT P ON OD.PRODUCT_CODE = P.PRODUCT_CODE

GROUP BY C.CUSTOMER_NAME, OM.ORDER_CODE, OM.ORDER_DATE

```
HAVING TOTAL_BILL >= 5000;
```

-- 19. Which country has maximum branches?

```
SELECT COUNTRY_NAME, TOTAL_BRANCHES
FROM (
    SELECT CN.COUNTRY_NAME, COUNT(B.BRANCH_CODE) AS TOTAL_BRANCHES,
           RANK() OVER (ORDER BY COUNT(B.BRANCH_CODE) DESC) AS rnk
    FROM COUNTRY CN
    JOIN BRANCH B ON CN.COUNTRY_CODE = B.COUNTRY_CODE
    GROUP BY CN.COUNTRY_NAME
) ranked
WHERE rnk = 1;
```

-- 20. Which country has minimum branches?

```
SELECT COUNTRY_NAME, TOTAL_BRANCHES
FROM (
    SELECT CN.COUNTRY_NAME, COUNT(B.BRANCH_CODE) AS TOTAL_BRANCHES,
           RANK() OVER (ORDER BY COUNT(B.BRANCH_CODE) ASC) AS rnk
    FROM COUNTRY CN
    LEFT JOIN BRANCH B ON CN.COUNTRY_CODE = B.COUNTRY_CODE
    GROUP BY CN.COUNTRY_NAME
) ranked
WHERE rnk = 1;
```

-- 21. Which country has no branches?

```
SELECT COUNTRY_CODE ,
       COUNTRY_NAME
FROM COUNTRY
WHERE COUNTRY_CODE NOT IN ( SELECT DISTINCT COUNTRY_CODE FROM BRANCH);
```

-- 22. Which branch has more sales representatives?

```
SELECT BRANCH_NAME, TOTAL_SALESREPS
FROM (
    SELECT B.BRANCH_NAME, COUNT(SR.SALESREP_ID) AS TOTAL_SALESREPS,
           RANK() OVER (ORDER BY COUNT(SR.SALESREP_ID) DESC) AS rnk
    FROM BRANCH B
    LEFT JOIN SALESREP SR ON B.BRANCH_CODE = SR.BRANCH_CODE
    GROUP BY B.BRANCH_NAME
) ranked
WHERE rnk = 1;
```

-- 23. Which branch has less sales representatives?

```
SELECT BRANCH_NAME, TOTAL_SALESREPS
FROM (
    SELECT B.BRANCH_NAME, COUNT(SR.SALESREP_ID) AS TOTAL_SALESREPS,
           RANK() OVER (ORDER BY COUNT(SR.SALESREP_ID) ASC) AS rnk
    FROM BRANCH B
    LEFT JOIN SALESREP SR ON B.BRANCH_CODE = SR.BRANCH_CODE
    GROUP BY B.BRANCH_NAME
) ranked
WHERE rnk = 1;
```


-- 24. Which branch has no sales representatives?

```
SELECT B.BRANCH_NAME
FROM BRANCH B
LEFT JOIN SALESREP SR ON B.BRANCH_CODE = SR.BRANCH_CODE
WHERE SR.SALESREP_ID IS NULL;
```

-- 25. Who is the active sales rep?

```
SELECT DISTINCT SR.SALESREP_ID, SR.SALESREP_NAME
FROM SALESREP SR
JOIN ORDER_MASTER OM ON SR.SALESREP_ID = OM.SALESREP_ID;
```

-- 26. Display all the manager's names?

```
SELECT DISTINCT S.SALESREP_ID , S.SALESREP_NAME AS Manager_name
FROM SALESREP S
JOIN SALESREP M ON S.SALESREP_ID = M.MGR;
```

-- 27. Display the sales rep that who has joined before their manager?

```
SELECT E.SALESREP_ID , E.SALESREP_NAME , E.SALESREP_DOJ ,
       M.SALESREP_NAME AS Manager_name , M.SALESREP_DOJ AS MANAGER_DOJ
FROM SALESREP E
JOIN SALESREP M ON E.MGR = M.SALESREP_ID
WHERE E.SALESREP_DOJ < M.SALESREP_DOJ ;
```

-- 28. Display the sales rep that who earns more than their manager?

```
SELECT E.SALESREP_ID , E.SALESREP_NAME , E.SALARY ,
       M.SALESREP_NAME AS mANAGER_NAME , M.SALESREP_DOJ AS MANAGER_DOJ
FROM SALESREP E
JOIN SALESREP M ON E.MGR = M.SALESREP_ID
WHERE E.SALARY > M.SALARY ;
```

-- 29. Display the manager who earns more than average salary of the sales rep working under them.

```
SELECT M.SALESREP_ID , M.SALESREP_NAME , M.SALARY , AVG(E.SALARY) AS AVG_TEAM_SALARY
FROM SALESREP M
```

```
JOIN SALESREP E ON M.SALESREP_ID = E.MGR
GROUP BY M.SALESREP_ID , M.SALESREP_NAME , M.SALARY
HAVING M.SALARY > AVG(E.SALARY);
```

-- 30. Display the sales rep details that who started their work immediately?

```
SELECT SALESREP_ID, SALESREP_NAME, SALESREP_DOJ
FROM (
    SELECT *, RANK() OVER(ORDER BY SALESREP_DOJ ASC) AS rnk
    FROM SALESREP
) AS RankedReps
WHERE rnk = 1;
```

-- 31. Display the sales rep that who is in bench for a long time?

```
SELECT SALESREP_ID , SALESREP_NAME , SALESREP_DOJ
FROM ( SELECT S.SALESREP_ID , S.SALESREP_NAME , S.SALESREP_DOJ ,
    RANK() OVER(ORDER BY (SALESREP_DOJ) ASC) AS rnk
    FROM SALESREP S
    LEFT JOIN ORDER_MASTER OM ON S.SALESREP_ID = OM.SALESREP_ID
    WHERE OM.ORDER_CODE IS NULL
)rnked
WHERE rnk = 1;
```

-- 32. Display the details of in-active sales representatives.

```
SELECT S.SALESREP_ID , S.SALESREP_NAME , S.SALESREP_DOJ
FROM SALESREP S
LEFT JOIN ORDER_MASTER OM ON S.SALESREP_ID = OM.SALESREP_ID
WHERE OM.ORDER_CODE IS NULL ;
```

```
SELECT *
FROM SALESREP
WHERE SALESREP_ID NOT IN (
    SELECT DISTINCT SALESREP_ID FROM ORDER_MASTER);
```

-- 33. Generate the report like

Sales_rep_id	Sales_rep_name	salary	Total_orde_value	Commission(5% of bill)
--------------	----------------	--------	------------------	------------------------

```
SELECT S.SALESREP_ID , S.SALESREP_NAME , S.SALARY ,  
       COALESCE( SUM(P.PRICE * O.QTY_ORDERED) ,0)AS Total_order_value ,  
       COALESCE( SUM(P.PRICE * O.QTY_ORDERED) , 0) * 0.05 AS Commision  
FROM SALESREP S  
  
LEFT JOIN ORDER_MASTER OM ON S.SALESREP_ID = OM.SALESREP_ID  
LEFT JOIN ORDER_DETAIL O ON OM.ORDER_CODE = O.ORDER_CODE  
LEFT JOIN PRODUCT P ON O.PRODUCT_CODE = P.PRODUCT_CODE  
GROUP BY S.SALESREP_ID , S.SALESREP_NAME , S.SALARY;
```

-- 34. Generate the report like

Sales_rep_name	Order_code	Bill_value
----------------	------------	------------

```
SELECT S.SALESREP_NAME , O.ORDER_CODE ,  
       SUM(P.PRICE * O.QTY_ORDERED) AS BILL_VALUE  
FROM SALESREP S  
  
JOIN ORDER_MASTER OM ON S.SALESREP_ID = OM.SALESREP_ID  
JOIN ORDER_DETAIL O ON OM.ORDER_CODE = O.ORDER_CODE  
JOIN PRODUCT P ON O.PRODUCT_CODE = P.PRODUCT_CODE  
GROUP BY S.SALESREP_NAME , O.ORDER_CODE;
```

-- 35. Generate the report like

Sales_rep_name	Order_code	No_of_products(in order)
----------------	------------	--------------------------

```
SELECT  
       S.SALESREP_NAME,  
       OM.ORDER_CODE,  
       SUM(OD.QTY_ORDERED) AS NO_OF_PRODUCTS  
FROM ORDER_MASTER OM  
  
JOIN SALESREP S ON OM.SALESREP_ID = S.SALESREP_ID  
JOIN ORDER_DETAIL OD ON OM.ORDER_CODE = OD.ORDER_CODE  
GROUP BY S.SALESREP_NAME, OM.ORDER_CODE;
```

-- 36.

Sales_rep_name	Order_code	Bill_value	No_of_products(in order)
----------------	------------	------------	--------------------------

-- A. For the orders placed on the year 2010

```
SELECT
    S.SALESREP_NAME,
    OM.ORDER_CODE,
    SUM(P.PRICE * OD.QTY_ORDERED) AS BILL_VALUE,
    SUM(OD.QTY_ORDERED) AS NO_OF_PRODUCTS
FROM ORDER_MASTER OM
JOIN SALESREP S ON OM.SALESREP_ID = S.SALESREP_ID
JOIN ORDER_DETAIL OD ON OM.ORDER_CODE = OD.ORDER_CODE
JOIN PRODUCT P ON OD.PRODUCT_CODE = P.PRODUCT_CODE
WHERE YEAR(OM.ORDER_DATE) = 2010
GROUP BY S.SALESREP_NAME, OM.ORDER_CODE;
```

-- B. For the sales rep who has placed not more than 10 orders

```
SELECT
    S.SALESREP_NAME,
    OM.ORDER_CODE,
    SUM(P.PRICE * OD.QTY_ORDERED) AS BILL_VALUE,
    SUM(OD.QTY_ORDERED) AS NO_OF_PRODUCTS
FROM ORDER_MASTER OM
JOIN SALESREP S ON OM.SALESREP_ID = S.SALESREP_ID
JOIN ORDER_DETAIL OD ON OM.ORDER_CODE = OD.ORDER_CODE
JOIN PRODUCT P ON OD.PRODUCT_CODE = P.PRODUCT_CODE
GROUP BY S.SALESREP_NAME, OM.ORDER_CODE
HAVING COUNT(OM.ORDER_CODE) <= 10;
```

-- C. For the sales rep whose salary >45000

```
SELECT
    S.SALESREP_NAME,
    OM.ORDER_CODE,
    SUM(P.PRICE * OD.QTY_ORDERED) AS BILL_VALUE,
    SUM(OD.QTY_ORDERED) AS NO_OF_PRODUCTS
FROM ORDER_MASTER OM
JOIN SALESREP S ON OM.SALESREP_ID = S.SALESREP_ID
```

```
JOIN ORDER_DETAIL OD ON OM.ORDER_CODE = OD.ORDER_CODE
JOIN PRODUCT P ON OD.PRODUCT_CODE = P.PRODUCT_CODE
WHERE S.SALARY > 45000
GROUP BY S.SALESREP_NAME, OM.ORDER_CODE;
```

-- 37. Who is the senior most sales rep?

```
SELECT SALESREP_ID , SALESREP_NAME , SALESREP_DOJ
FROM (SELECT *, RANK() OVER(ORDER BY (SALESREP_DOJ) ASC) AS rnk
      FROM SALESREP
      ) rnked
WHERE rnk = 1;
```

-- 38. Find the no of orders processed by junior most sales rep

```
SELECT SALESREP_ID, SALESREP_NAME, NO_OF_ORDERS
FROM (
      SELECT S.SALESREP_ID, S.SALESREP_NAME, S.SALESREP_DOJ,
             COUNT(O.ORDER_CODE) AS NO_OF_ORDERS,
             RANK() OVER (ORDER BY S.SALESREP_DOJ DESC) AS rnk
      FROM SALESREP S
      LEFT JOIN ORDER_MASTER O ON S.SALESREP_ID = O.SALESREP_ID
      GROUP BY S.SALESREP_ID, S.SALESREP_NAME, S.SALESREP_DOJ
      ) AS ranked_salesreps
WHERE rnk = 1;
```

-- 39. Find the no of orders processed by each month in the year 2010?

```
SELECT COUNT(ORDER_CODE) AS NO_OF_ORDERS ,
       MONTH(ORDER_DATE) AS MONTHLY_PROCESSING
FROM ORDER_MASTER
WHERE YEAR(ORDER_DATE) = 2010
GROUP BY MONTH(ORDER_DATE) ;
```

-- 40. Which region is having in-active sales rep (using joins?)

```
SELECT S.SALESREP_ID , S.SALESREP_NAME , C.REGION
FROM COUNTRY C

JOIN BRANCH B ON C.COUNTRY_CODE = B.COUNTRY_CODE
```

```
JOIN SALESREP S ON S.BRANCH_CODE = B.BRANCH_CODE
LEFT JOIN ORDER_MASTER OM ON S.SALESREP_ID = OM.SALESREP_ID
WHERE OM.ORDER_CODE IS NULL ;
```

-- 41. Which region is having junior most sales rep?

```
SELECT REGION , SALESREP_ID , SALESREP_NAME , SALESREP_DOJ
FROM (SELECT C.REGION , S.SALESREP_ID , S.SALESREP_NAME , S.SALESREP_DOJ ,
        RANK() OVER(ORDER BY(S.SALESREP_DOJ) DESC) AS rnk
FROM SALESREP S
JOIN BRANCH B ON S.BRANCH_CODE = B.BRANCH_CODE
JOIN COUNTRY C ON B.COUNTRY_CODE = C.COUNTRY_CODE
) rnk
WHERE rnk = 1 ;
```

-- 42. Find the no of orders whose bill is less than 500?

```
SELECT O.ORDER_CODE , SUM(P.PRICE * O.QTY_ORDERED) AS BILL ,
COUNT(O.ORDER_CODE) AS NO_OF_ORDERS
FROM ORDER_DETAIL O
JOIN ORDER_MASTER OM ON O.ORDER_CODE = OM.ORDER_CODE
JOIN PRODUCT P ON O.PRODUCT_CODE = P.PRODUCT_CODE
GROUP BY O.ORDER_CODE
HAVING SUM(P.PRICE * O.QTY_ORDERED) < 500;
```

-- 43. What is the frequently ordered product by the most active customer?

-- Get the most frequently ordered product by the most active customer using RANK()

-- Display the most frequently ordered product by the most active customer

```
SELECT PRODUCT_CODE, ORDER_COUNT
FROM (
    SELECT OD.PRODUCT_CODE,
        COUNT(OD.ORDER_CODE) AS ORDER_COUNT,
        RANK() OVER (ORDER BY COUNT(OD.ORDER_CODE) DESC) AS rnk
FROM ORDER_DETAIL OD
JOIN ORDER_MASTER OM ON OD.ORDER_CODE = OM.ORDER_CODE
WHERE OM.CUSTOMER_CODE = (
```

```

-- Subquery to get the most active customer using RANK()
SELECT CUSTOMER_CODE
FROM (
    SELECT CUSTOMER_CODE,
           RANK() OVER (ORDER BY COUNT(ORDER_CODE) DESC) AS rnk
    FROM ORDER_MASTER
    GROUP BY CUSTOMER_CODE
) ranked_customers
WHERE rnk = 1
)
GROUP BY OD.PRODUCT_CODE
) ranked_products
WHERE rnk = 1;

```

-- 44. What are all the products are not ordered by active customer?

-- Display products not ordered by the most active customer

```

SELECT P.PRODUCT_CODE
FROM PRODUCT P
WHERE P.PRODUCT_CODE NOT IN (
    SELECT OD.PRODUCT_CODE
    FROM ORDER_DETAIL OD
    JOIN ORDER_MASTER OM ON OD.ORDER_CODE = OM.ORDER_CODE
    WHERE OM.CUSTOMER_CODE = (
        -- Get the most active customer using RANK()
        SELECT CUSTOMER_CODE
        FROM (
            SELECT CUSTOMER_CODE,
                   RANK() OVER (ORDER BY COUNT(ORDER_CODE) DESC) AS rnk
            FROM ORDER_MASTER
            GROUP BY CUSTOMER_CODE
        ) ranked_customers
    )
)

```

```
WHERE rnk = 1
)
);
```

-- 45. Display the orders which can't be processed immediately?

-- (Hint: where order quantity>qoh)

```
SELECT OD.ORDER_CODE , P.PRODUCT_CODE , OD.QTY_ORDERED , P.QTY_ON_HAND
FROM ORDER_DETAIL OD
JOIN PRODUCT P ON OD.PRODUCT_CODE = P.PRODUCT_CODE
WHERE OD.QTY_ORDERED > P.QTY_ON_HAND;
```

-- Complete Order Report showing customer, order, and product details

-- 46. Complete order report

Cust_name	Cust_address	Order_code	Product_id	Product_name	Order_date
-----------	--------------	------------	------------	--------------	------------

```
SELECT C.CUSTOMER_NAME,
       C.CUSTOMER_ADDRESS,
       OM.ORDER_CODE,
       P.PRODUCT_CODE,
       P.PRODUCT_NAME,
       OM.ORDER_DATE
FROM CUSTOMER C
JOIN ORDER_MASTER OM ON C.CUSTOMER_CODE = OM.CUSTOMER_CODE
JOIN ORDER_DETAIL OD ON OM.ORDER_CODE = OD.ORDER_CODE
JOIN PRODUCT P ON OD.PRODUCT_CODE = P.PRODUCT_CODE;
```

-- 48. Department wise who is the maximum salary earner?


```

SELECT BRANCH_CODE , SALESREP_ID , SALESREP_NAME , SALARY
FROM (
SELECT B.BRANCH_CODE , S.SALESREP_ID , S.SALESREP_NAME , S.SALARY ,
      RANK() OVER(PARTITION BY BRANCH_CODE ORDER BY (S.SALARY)DESC) AS rnk
FROM BRANCH B
JOIN SALESREP S ON B.BRANCH_CODE = S.BRANCH_CODE
GROUP BY B.BRANCH_CODE , S.SALESREP_ID , S.SALESREP_NAME
) AS rnked
WHERE rnk = 1;

```

-- 49. Find the Top-10 active customers based on no. of orders?

```

SELECT CUSTOMER_CODE , ORDER_COUNT
FROM (SELECT OM.CUSTOMER_CODE , COUNT(OD.ORDER_CODE) AS ORDER_COUNT ,
      RANK() OVER(ORDER BY COUNT(OD.ORDER_CODE) DESC) AS rnk
FROM ORDER_DETAIL OD
JOIN ORDER_MASTER OM ON OD.ORDER_CODE = OM.ORDER_CODE
GROUP BY OM.CUSTOMER_CODE
) AS rnked
WHERE rnk <= 10 ;

```

-- 50. Find the Top-5 orders based on bill value.

```

SELECT ORDER_CODE , BILL_VALUE
FROM ( SELECT OD.ORDER_CODE , SUM(P.PRICE * OD.QTY_ORDERED) AS BILL_VALUE,
      RANK() OVER(ORDER BY SUM(P.PRICE * OD.QTY_ORDERED) DESC) AS rnk
FROM ORDER_DETAIL OD
JOIN PRODUCT P ON OD.PRODUCT_CODE = P.PRODUCT_CODE
GROUP BY OD.ORDER_CODE
) rnked
WHERE rnk <= 5;

```

-- 51. Display the sales rep that who processed the orders ORD11 and ORD15.

```

SELECT SALESREP_ID , SALESREP_NAME
FROM SALESREP

WHERE SALESREP_ID IN (
      SELECT SALESREP_ID

```

```
FROM ORDER_MASTER
WHERE ORDER_CODE IN ('ORD11', 'ORD15')
GROUP BY SALESREP_ID
HAVING COUNT(DISTINCT ORDER_CODE) = 2);
```

-- 52. Generate a report like 'worker works for manager'

```
SELECT
    CONCAT(sr.SALESREP_NAME, ' works for ', m.SALESREP_NAME) AS relationship
FROM SALESREP sr
JOIN SALESREP m ON sr.MGR = m.SALESREP_ID;
```

```
SELECT * FROM SALESREP;
```