**Tiny Shop Sales – SQL Case Study**

**--1) Which product has the highest price? Only return a single row.**

SELECT

\*

FROM products

WHERE price = (SELECT MAX(price) FROM products) ;

**--2) Which customer has made the most orders?**

WITH CUST\_MAX AS(

SELECT

CUSTOMER\_ID,

COUNT(CUSTOMER\_ID) AS CNT

FROM ORDERS

GROUP BY CUSTOMER\_ID

)

SELECT

CUSTOMER\_ID

FROM CUST\_MAX

WHERE CNT IN (SELECT MAX(CNT) FROM CUST\_MAX)

ORDER BY CUSTOMER\_ID;

**--3) What’s the total revenue per product?**

SELECT

O.PRODUCT\_ID,

SUM(O.QUANTITY \* P.PRICE) AS REVENUE

FROM ORDER\_ITEMS O

INNER JOIN PRODUCTS P

ON O.PRODUCT\_ID = P.PRODUCT\_ID

GROUP BY O.PRODUCT\_ID

ORDER BY O.PRODUCT\_ID;

**--4) Find the day with the highest revenue.**

SELECT

O.ORDER\_DATE,

SUM(I.QUANTITY\*P.PRICE) AS REVENUE

FROM ORDERS O

INNER JOIN ORDER\_ITEMS I

ON O.ORDER\_ID = I.ORDER\_ID

INNER JOIN PRODUCTS P

ON I.PRODUCT\_ID = P.PRODUCT\_ID

GROUP BY O.ORDER\_DATE

ORDER BY REVENUE DESC

LIMIT 1;

**--5) Find the first order (by date) for each customer.**

SELECT

CUSTOMER\_ID,

ORDER\_ID,

ORDER\_DATE

FROM ( SELECT

ORDER\_ID,

CUSTOMER\_ID,

ORDER\_DATE,

ROW\_NUMBER() OVER(PARTITION BY CUSTOMER\_ID ORDER BY ORDER\_DATE) AS RNUM

FROM ORDERS) O

WHERE RNUM =1;

**--6) Find the top 3 customers who have ordered the most distinct products**

SELECT

O.CUSTOMER\_ID,

COUNT(DISTINCT PRODUCT\_ID) AS DIST\_CNT

FROM ORDERS O

INNER JOIN ORDER\_ITEMS I

ON O.ORDER\_ID = I.ORDER\_ID

GROUP BY O.CUSTOMER\_ID

ORDER BY DIST\_CNT DESC, O.CUSTOMER\_ID

LIMIT 3;

**--7) Which product has been bought the least in terms of quantity?**

WITH PROD\_SUM AS (

SELECT

PRODUCT\_ID,

SUM(QUANTITY) AS TOTAL

FROM ORDER\_ITEMS

GROUP BY PRODUCT\_ID

)

SELECT

PRODUCT\_ID,

TOTAL

FROM PROD\_SUM

WHERE TOTAL = (SELECT MIN(TOTAL) FROM PROD\_SUM);

**--8) What is the median order total?**

/\* THE PERCENTILE\_CONT() WITH A VALUE OF 0.5 INDICATING A 50% OR MID-VALUE CAN BE USED TO GET THE MEDIAN VALUE \*/

WITH ALL\_ORDERS AS(

SELECT

I.ORDER\_ID,

SUM(I.QUANTITY \* P.PRICE) AS TOTAL\_COST

FROM ORDER\_ITEMS I

INNER JOIN PRODUCTS P

ON I.PRODUCT\_ID = P.PRODUCT\_ID

GROUP BY I.ORDER\_ID

)

SELECT

PERCENTILE\_CONT(0.5) WITHIN GROUP(ORDER BY TOTAL\_COST DESC) AS MEDIAN\_ORDER\_PRICE

FROM ALL\_ORDERS;

**--9) For each order, determine if it was ‘Expensive’ (total over 300), ‘Affordable’ (total over 100), or ‘Cheap’.**

WITH ALL\_ORDERS AS(

SELECT

I.ORDER\_ID,

SUM(I.QUANTITY \* P.PRICE) AS TOTAL\_COST

FROM ORDER\_ITEMS I

INNER JOIN PRODUCTS P

ON I.PRODUCT\_ID = P.PRODUCT\_ID

GROUP BY I.ORDER\_ID

)

SELECT

ORDER\_ID AS Order\_No,

TOTAL\_COST AS Total\_Cost,

CASE WHEN TOTAL\_COST>300 THEN 'Expensive'

WHEN TOTAL\_COST>100 AND TOTAL\_COST<=300 THEN 'Affordable'

ELSE 'Cheap'

END AS Order\_Type

FROM ALL\_ORDERS

ORDER BY Order\_No;

**--10) Find customers who have ordered the product with the highest price.**

/\* THE INNERMOST SUB-QUERY RETURNS THE PRODUCT WITH THE HIGHEST PRICE, THE NEXT LEVEL SUB-QUERY RETURNS THE ORDER\_ID WHICH CONTAINS THESE HIGHEST PRICED PRODUCTS AND THE OUTERMOST QUERY RETURNS THE CUSTOMERS WHO BOUGHT THESE PRODUCTS \*/

SELECT

C.CUSTOMER\_ID,

C.FIRST\_NAME,

C.LAST\_NAME

FROM ORDERS O

INNER JOIN CUSTOMERS C

ON O.CUSTOMER\_ID = C.CUSTOMER\_ID

WHERE ORDER\_ID IN (

SELECT

ORDER\_ID

FROM ORDER\_ITEMS

WHERE PRODUCT\_ID IN (

SELECT

PRODUCT\_ID

FROM PRODUCTS

ORDER BY PRICE DESC

LIMIT 1

)

)

ORDER BY C.CUSTOMER\_ID;