

SQL CASE STUDY

DATA IN MOTION TINY SHOP SALES



DATA IN MOTION



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

```
SELECT PRODUCT_NAME,  
       PRICE  
FROM PRODUCTS  
ORDER BY PRICE DESC  
LIMIT 1;
```

2. Which customer has made the most orders?

```
WITH ORDER_COUNT (CUSTOMER_ID, NUM_OF_ORDERS) AS  
    (SELECT CUSTOMER_ID,  
            COUNT(ORDER_ID)  
            FROM ORDERS  
            GROUP BY CUSTOMER_ID)  
SELECT CONCAT(C.FIRST_NAME, ' ', C.LAST_NAME) AS customer_name,  
       C.CUSTOMER_ID,  
       OC.NUM_OF_ORDERS  
FROM CUSTOMERS C  
JOIN ORDER_COUNT OC ON C.CUSTOMER_ID = OC.CUSTOMER_ID  
WHERE OC.NUM_OF_ORDERS =  
      (SELECT MAX(NUM_OF_ORDERS)  
       FROM ORDER_COUNT);
```



3. What's the total revenue per product?

```
WITH QUANTITY_SOLD AS
  (SELECT PRODUCT_ID,
         SUM(QUANTITY) AS TOTAL_QUANTITIES
   FROM ORDER_ITEMS
   GROUP BY PRODUCT_ID)
SELECT P.PRODUCT_ID,
       P.PRODUCT_NAME,
       P.PRICE * QS.TOTAL_QUANTITIES AS REVENUE
FROM PRODUCTS P
JOIN QUANTITY_SOLD QS ON P.PRODUCT_ID = QS.PRODUCT_ID;
```



4. Find the day with the highest revenue.

```
SELECT INFO.ORDER_DATE,  
       SUM(ITEMS.QUANTITY * P.PRICE) AS TOTAL_REVENUE  
FROM ORDERS INFO  
JOIN ORDER_ITEMS ITEMS ON INFO.ORDER_ID = ITEMS.ORDER_ID  
JOIN PRODUCTS P ON ITEMS.PRODUCT_ID = P.PRODUCT_ID  
GROUP BY INFO.ORDER_DATE  
ORDER BY TOTAL_REVENUE DESC  
LIMIT 1;
```

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

```
SELECT CONCAT(C.FIRST_NAME, ' ', C.LAST_NAME) AS customer_name,  
       O.CUSTOMER_ID,  
       MIN(O.ORDER_DATE) AS FIRST_ORDER_DATE  
FROM ORDERS O  
JOIN CUSTOMERS C ON O.CUSTOMER_ID = C.CUSTOMER_ID  
GROUP BY customer_name, O.CUSTOMER_ID  
ORDER BY customer;
```



6. Find the top 3 customers who have ordered the most distinct products

```
SELECT O.CUSTOMER_ID,  
       CONCAT(C.FIRST_NAME, " ", C.LAST_NAME) AS CUSTOMER_NAME,  
       COUNT(DISTINCT OI.PRODUCT_ID) AS DISTINCT_PRODUCTS_BOUGHT  
FROM ORDERS O  
JOIN ORDER_ITEMS OI ON OI.ORDER_ID = O.ORDER_ID  
JOIN CUSTOMERS C ON C.CUSTOMER_ID = O.CUSTOMER_ID  
GROUP BY O.CUSTOMER_ID  
LIMIT 3;
```

7. Which product has been bought the least in terms of quantity?

```
WITH LEAST_QUANTITY AS  
    (SELECT PRODUCT_ID,  
             SUM(QUANTITY) AS QUANTITIES_SOLD  
     FROM ORDER_ITEMS  
     GROUP BY PRODUCT_ID  
     ORDER BY PRODUCT_ID)  
SELECT PRODUCT_ID,  
       QUANTITIES_SOLD  
FROM LEAST_QUANTITY  
WHERE QUANTITIES_SOLD =  
      (SELECT MIN(QUANTITIES_SOLD)  
       FROM LEAST_QUANTITY);
```



8. What is the median order total?

```
WITH ORDER_TOTAL (ORDER_ID, TOTAL_REVENUE) AS
    (SELECT O.ORDER_ID,
        SUM(OI.QUANTITY * P.PRICE) AS TOTAL_REVENUE
     FROM ORDERS O
     JOIN ORDER_ITEMS OI ON O.ORDER_ID = OI.ORDER_ID
     JOIN PRODUCTS P ON P.PRODUCT_ID = OI.PRODUCT_ID
     GROUP BY O.ORDER_ID)
SELECT PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY TOTAL_REVENUE) AS MEDIAN_ORDER_TOTAL
FROM ORDER_TOTAL;
```

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

```
SELECT OI.ORDER_ID,
    CASE
        WHEN SUM(OI.QUANTITY * P.PRICE) > 300 THEN 'Expensive'
        WHEN SUM(OI.QUANTITY * P.PRICE) > 100 THEN 'Affordable'
        ELSE 'Cheap'
    END AS ORDER_CATEGORY
FROM ORDER_ITEMS OI
JOIN PRODUCTS P ON OI.PRODUCT_ID = P.PRODUCT_ID
GROUP BY OI.ORDER_ID
ORDER BY OI.ORDER_ID
```



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

```
SELECT C.CUSTOMER_ID,  
       CONCAT(C.FIRST_NAME, ' ', C.LAST_NAME) AS CUSTOMER_NAME,  
       P.PRICE AS PROD_PRICE  
FROM CUSTOMERS C  
JOIN ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID  
JOIN ORDER_ITEMS OI ON O.ORDER_ID = OI.ORDER_ID  
JOIN PRODUCTS P ON P.PRODUCT_ID = OI.PRODUCT_ID  
WHERE P.PRICE =  
       (SELECT MAX(PRICE)  
        FROM PRODUCTS)
```

