SQL ASSIGNMENT 3

Dataset: Sales Information

You have been given a dataset containing information about sales transactions. The dataset includes the following columns:

• order_id (integer): Unique identifier for each order.

• customer_id (integer): Unique identifier for each customer.

• product_id (integer): Unique identifier for each product.

• product_name (string): Name of the product.

• quantity (integer): The quantity of the product sold.

• unit_price (decimal): The unit price of the product.

• order_date (date): The date when the order was placed

Table Structure:

Create a table named sales with the following structure:

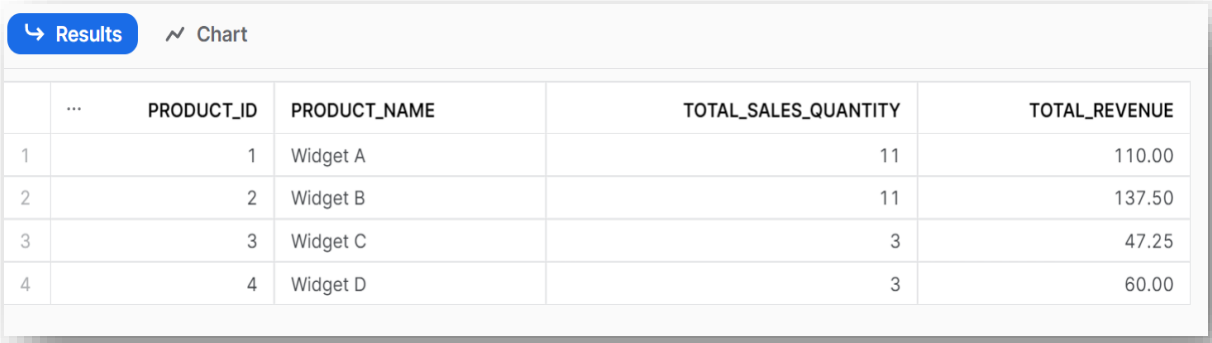
```
CREATE TABLE VK_sales (
 order_id INT PRIMARY KEY,
 customer_id INT,
 product_id INT,
 product_name VARCHAR(50),
 quantity INT,
 unit_price DECIMAL(10, 2),
 order_date DATE
);
```

```
INSERT INTO VK_sales  (order_id, customer_id, product_id, product_name, quantity, unit_price, order_date)
VALUES
 (1, 101, 1, 'Widget A', 5, 10.00, '2023-01-15'),
 (2, 102, 2, 'Widget B', 2, 12.50, '2023-01-16'),
 (3, 103, 1, 'Widget A', 3, 10.00, '2023-01-16'),
 (4, 104, 3, 'Widget C', 1, 15.75, '2023-01-17'),
 (5, 105, 2, 'Widget B', 4, 12.50, '2023-01-17'),
 (6, 106, 1, 'Widget A', 2, 10.00, '2023-01-18'),
 (7, 107, 4, 'Widget D', 3, 20.00, '2023-01-18'),
 (8, 108, 2, 'Widget B', 5, 12.50, '2023-01-19'),
 (9, 109, 1, 'Widget A', 1, 10.00, '2023-01-19'),
 (10, 101, 3, 'Widget C', 2, 15.75, '2023-01-20');


SELECT * FROM DEMO_DATABASE.PUBLIC.VK_SALES ;
```

1. Retrieve the total sales quantity and revenue for each product.

```
   SELECT
  PRODUCT_ID,PRODUCT_NAME,
  SUM(QUANTITY) AS total_sales_quantity,
  SUM(UNIT_PRICE*QUANTITY) AS total_revenue
FROM
  DEMO_DATABASE.PUBLIC.VK_SALES
GROUP BY
  PRODUCT_ID,PRODUCT_NAME;
```

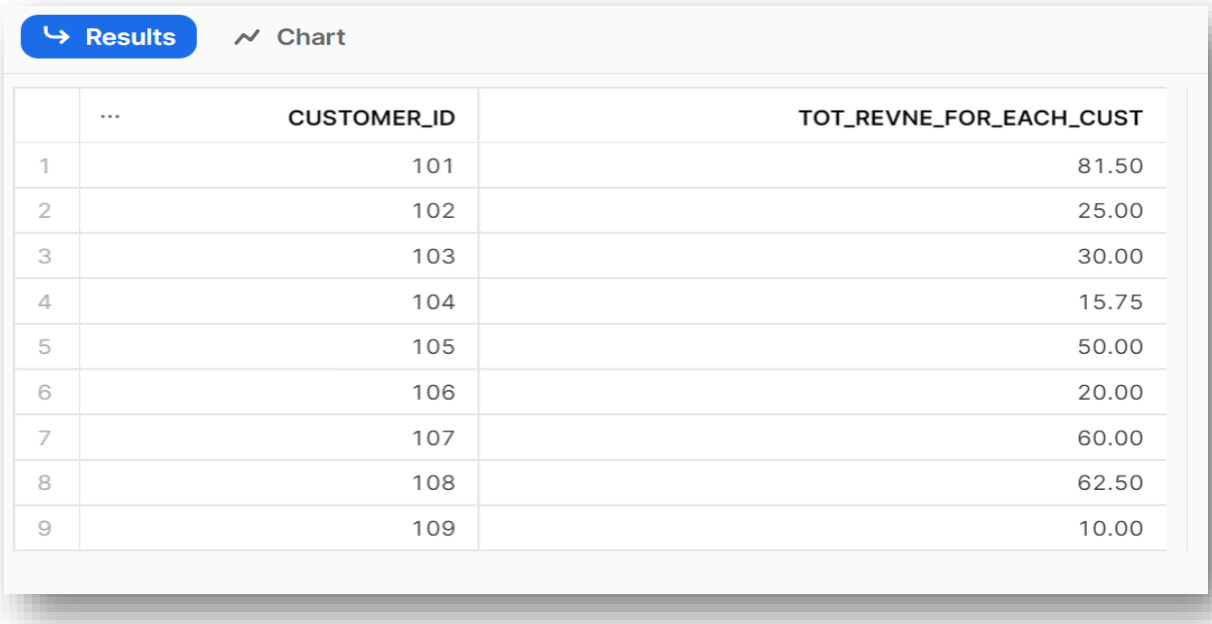


Results Chart

	...	PRODUCT_ID	PRODUCT_NAME	TOTAL_SALES_QUANTITY	TOTAL_REVENUE
1		1	Widget A	11	110.00
2		2	Widget B	11	137.50
3		3	Widget C	3	47.25
4		4	Widget D	3	60.00

2. Find the total revenue for each customer.

```
SELECT CUSTOMER_ID,
SUM(UNIT_PRICE*QUANTITY) AS TOT_REVNE_FOR_EACH_CUST
FROM
  DEMO_DATABASE.PUBLIC.VK_SALES
GROUP BY
 CUSTOMER_ID
 ORDER BY 1;
```

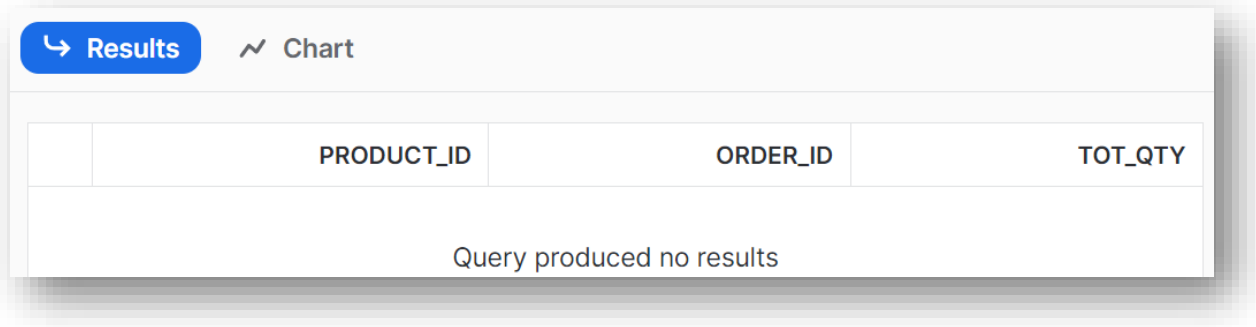


Results Chart

	...	CUSTOMER_ID	TOT_REVNE_FOR_EACH_CUST
1		101	81.50
2		102	25.00
3		103	30.00
4		104	15.75
5		105	50.00
6		106	20.00
7		107	60.00
8		108	62.50
9		109	10.00

3. Get the products with more than 10 units sold in a single order.

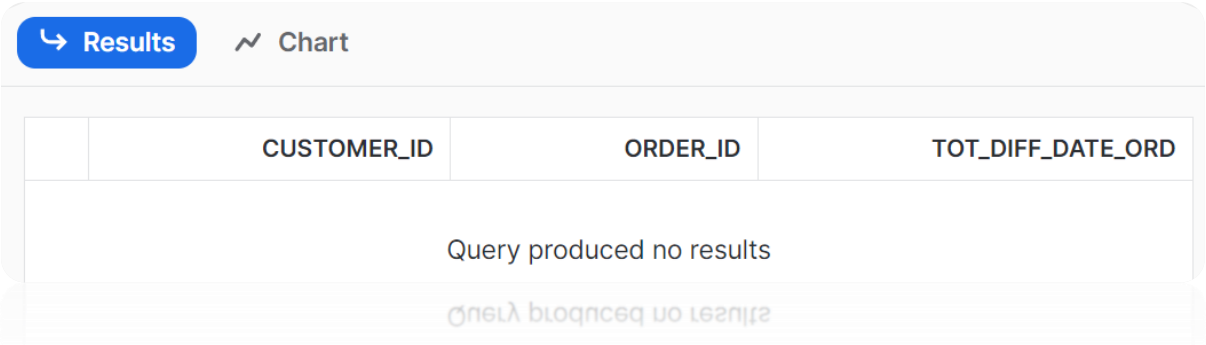
```
SELECT PRODUCT_ID,ORDER_ID,
SUM(QUANTITY) AS TOT_QTY
FROM DEMO_DATABASE.PUBLIC.VK_SALES
GROUP BY PRODUCT_ID,ORDER_ID
HAVING SUM(QUANTITY)> 10;
```



---there is no product products with more than 10 units sold in a single order.

4.List the customers who have placed orders on at least three different dates.

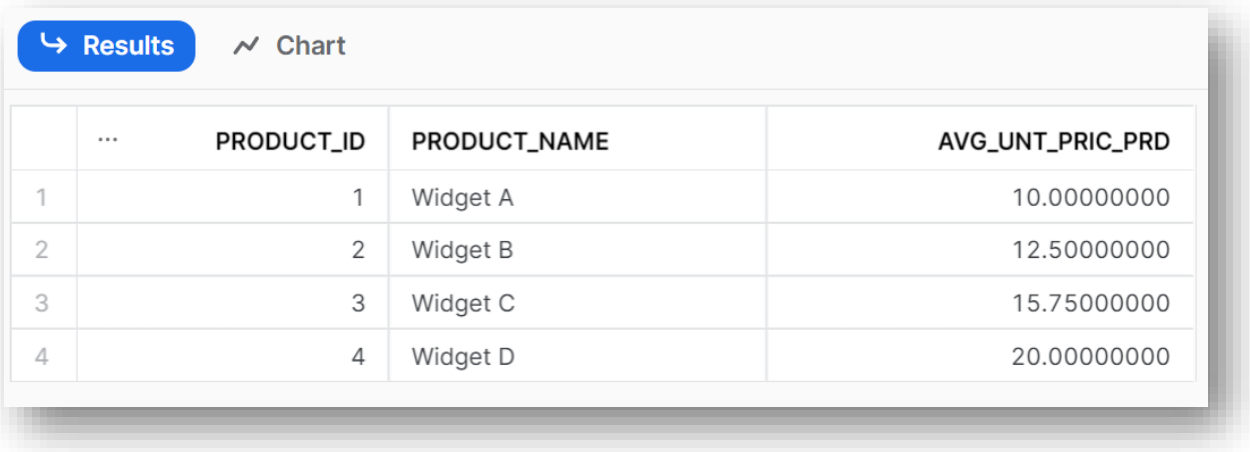
```
SELECT CUSTOMER_ID,ORDER_ID,
COUNT(DISTINCT ORDER_DATE) AS TOT_DIFF_DATE_ORD
 FROM
 DEMO_DATABASE.PUBLIC.VK_SALES
 GROUP BY
 CUSTOMER_ID,ORDER_ID
 HAVING COUNT(DISTINCT ORDER_DATE) >= 3;
```



---**there is no such records**

5. Calculate the average unit price of products

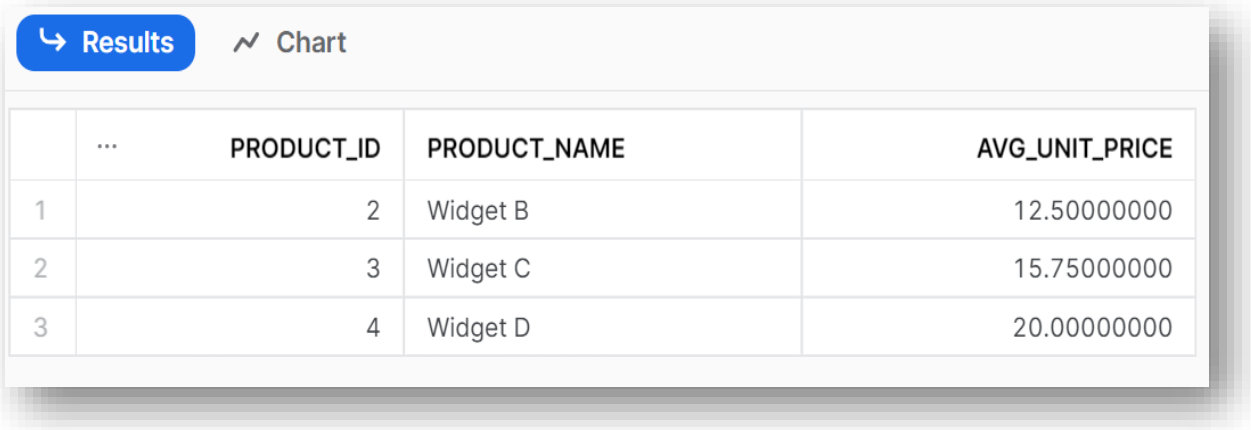
```
SELECT PRODUCT_ID,PRODUCT_NAME,
AVG(UNIT_PRICE) AS AVG_UNT_PRIC_PRD
FROM
DEMO_DATABASE.PUBLIC.VK_SALES
GROUP BY
PRODUCT_ID,PRODUCT_NAME;
```



	PRODUCT_ID	PRODUCT_NAME	AVG_UNT_PRIC_PRD
1	1	Widget A	10.00000000
2	2	Widget B	12.50000000
3	3	Widget C	15.75000000
4	4	Widget D	20.00000000

6. Find the products with an average unit price greater than $12.00.

```
SELECT
  PRODUCT_ID,
  PRODUCT_NAME,
  AVG(UNIT_PRICE) AS AVG_UNIT_PRICE
FROM
  DEMO_DATABASE.PUBLIC.VK_SALES
GROUP BY
  PRODUCT_ID, PRODUCT_NAME
HAVING
  AVG(UNIT_PRICE) > 12.00;
```

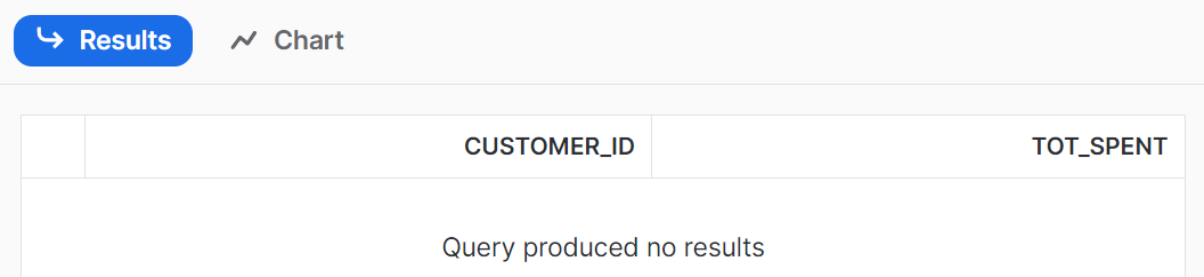


Results | Chart

	... PRODUCT_ID	PRODUCT_NAME	AVG_UNIT_PRICE
1	2	Widget B	12.50000000
2	3	Widget C	15.75000000
3	4	Widget D	20.00000000

7. Retrieve the customers who have spent more than $100.00 in total.

```
SELECT CUSTOMER_ID,
SUM(UNIT_PRICE*QUANTITY) AS TOT_SPENT
FROM
  DEMO_DATABASE.PUBLIC.VK_SALES
 GROUP BY
  CUSTOMER_ID
HAVING
  TOT_SPENT > 100.00;
```

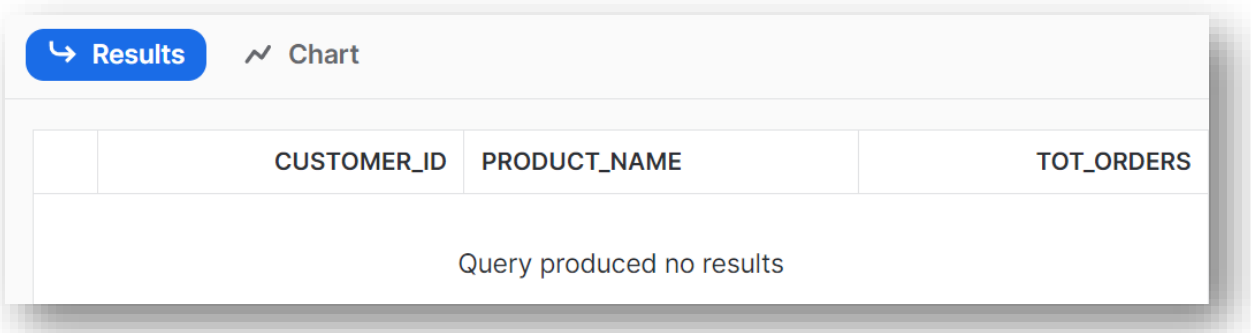


---so there is no customers who have spent more than $100.00 in total.

8. List the customers who have purchased 'Widget B' and 'Widget A' in the same order.

```
SELECT CUSTOMER_ID,PRODUCT_NAME,
COUNT(DISTINCT ORDER_ID) AS TOT_ORDERS
FROM
DEMO_DATABASE.PUBLIC.VK_SALES
WHERE
    PRODUCT_NAME IN ('Widget A', 'Widget B')
GROUP BY
```

```
CUSTOMER_ID,PRODUCT_NAME
  HAVING
    COUNT(DISTINCT ORDER_ID) = 2;
```



---so there is no records

----------THE END----------