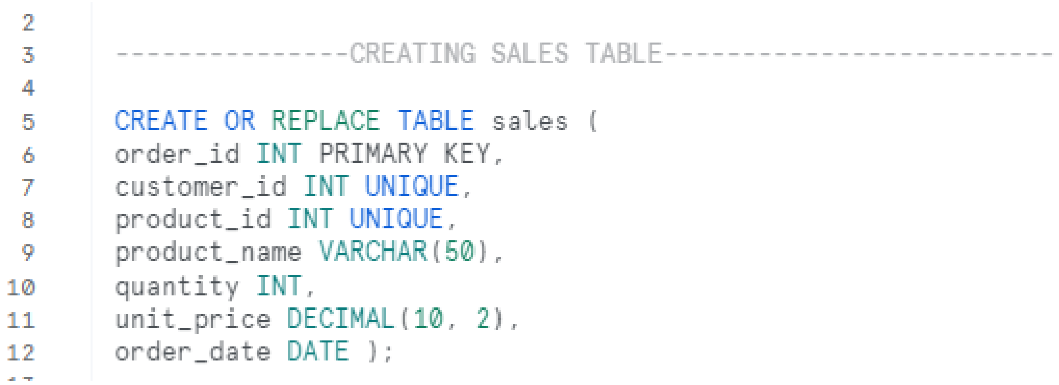
SQL Assignment-3 : SQL GROUP BY And Aggregation

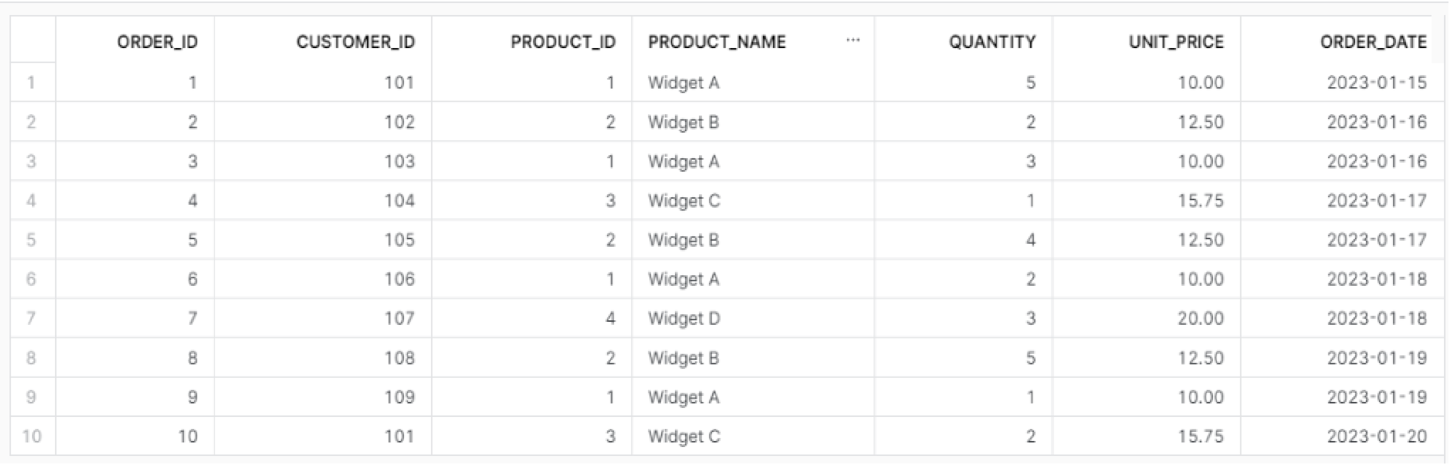
Creating a table named sales with the following structure:-



```

---------------CREATING SALES TABLE------------------------

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

Inserting the data in the table Sales:-

Results:-After data insertion Table be Like-



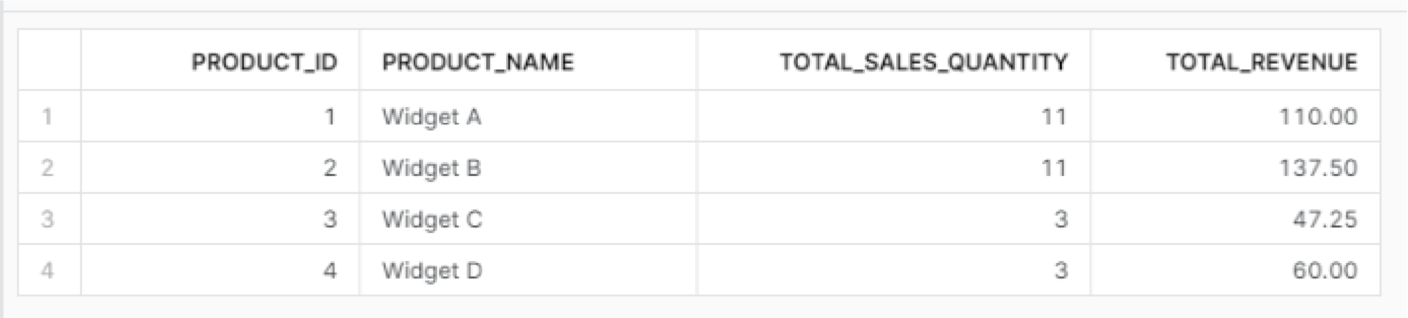
	ORDER_ID	CUSTOMER_ID	PRODUCT_ID	PRODUCT_NAME	QUANTITY	UNIT_PRICE	ORDER_DATE
1	1	101	1	Widget A	5	10.00	2023-01-15
2	2	102	2	Widget B	2	12.50	2023-01-16
3	3	103	1	Widget A	3	10.00	2023-01-16
4	4	104	3	Widget C	1	15.75	2023-01-17
5	5	105	2	Widget B	4	12.50	2023-01-17
6	6	106	1	Widget A	2	10.00	2023-01-18
7	7	107	4	Widget D	3	20.00	2023-01-18
8	8	108	2	Widget B	5	12.50	2023-01-19
9	9	109	1	Widget A	1	10.00	2023-01-19
10	10	101	3	Widget C	2	15.75	2023-01-20

Below are the question answer with their results of the SQL Queries.:-

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

```
SELECT PRODUCT_ID,PRODUCT_NAME ,

SUM(QUANTITY) AS TOTAL_SALES_QUANTITY,

SUM(QUANTITY*UNIT_PRICE) AS TOTAL_REVENUE

FROM SALES

GROUP BY 1,2;
```

Results:-

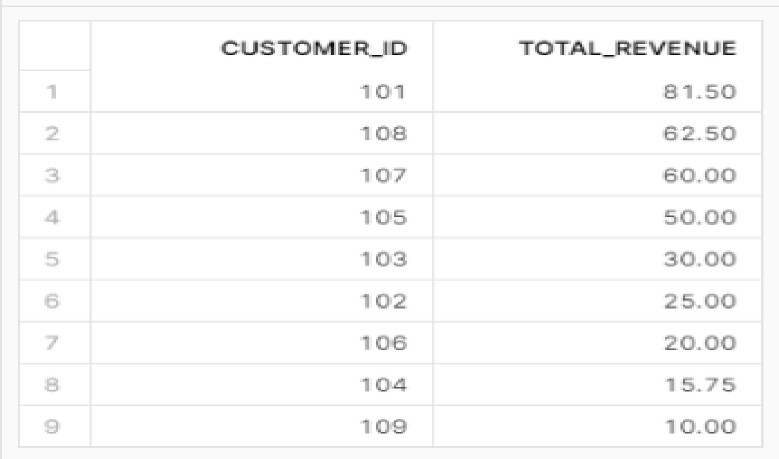


	PRODUCT_ID	PRODUCT_NAME	TOTAL_SALES_QUANTITY	TOTAL_REVENUE
1	1	Widget A	11	110.00
2	2	Widget B	11	137.50
3	3	Widget C	3	47.25
4	4	Widget D	3	60.00

2. Find the total revenue for each customer.

```
SELECT CUSTOMER_ID,

SUM( QUANTITY*UNIT_PRICE) AS TOTAL_REVENUE

FROM SALES

GROUP BY 1

ORDER BY 2 DESC;
```

Results:-

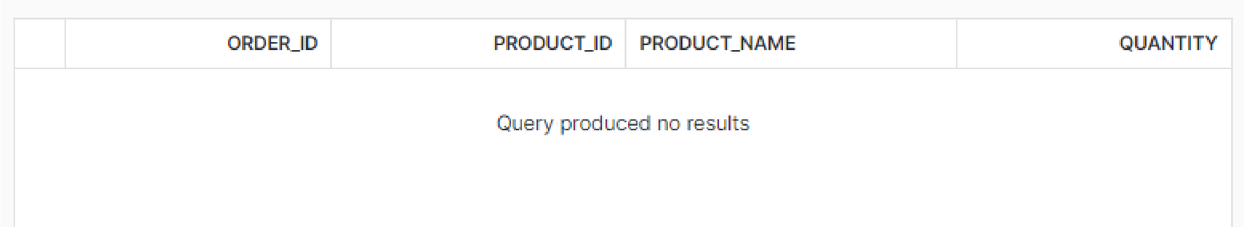


	CUSTOMER_ID	TOTAL_REVENUE
1	101	81.50
2	108	62.50
3	107	60.00
4	105	50.00
5	103	30.00
6	102	25.00
7	106	20.00
8	104	15.75
9	109	10.00

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

```
SELECT DISTINCT ORDER_ID, PRODUCT_ID ,
PRODUCT_NAME,QUANTITY

FROM SALES

WHERE QUANTITY > 10;
```

Results:- **AS there is no Product whoes 10 units sold in single order so no Query is produced as result.**



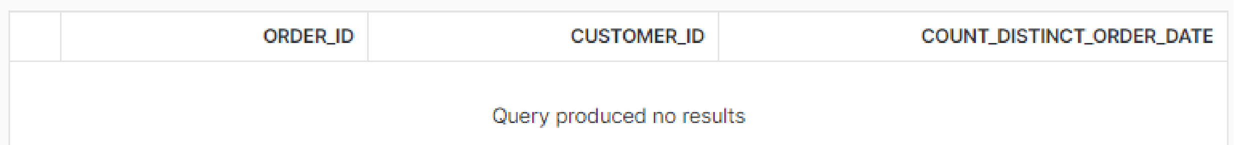
	ORDER_ID	PRODUCT_ID	PRODUCT_NAME	QUANTITY

Query produced no results

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

```
SELECT ORDER_ID,CUSTOMER_ID,COUNT(DISTINCT ORDER_DATE) AS COUNT_DISTINCT_ORDER_DATE

FROM SALES

GROUP BY 1,2

HAVING COUNT_DISTINCT_ORDER_DATE > 3;
```

Results:- **As there is no customers who have placed orders on at least three different dates ,So query is produced as results**.



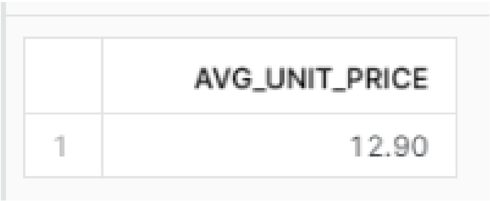
	ORDER_ID	CUSTOMER_ID	COUNT_DISTINCT_ORDER_DATE

Query produced no results

5. Calculate the average unit price of products.

```
SELECT ROUND(AVG(UNIT_PRICE),2) AS AVG_UNIT_PRICE

FROM SALES;
```

Results:-

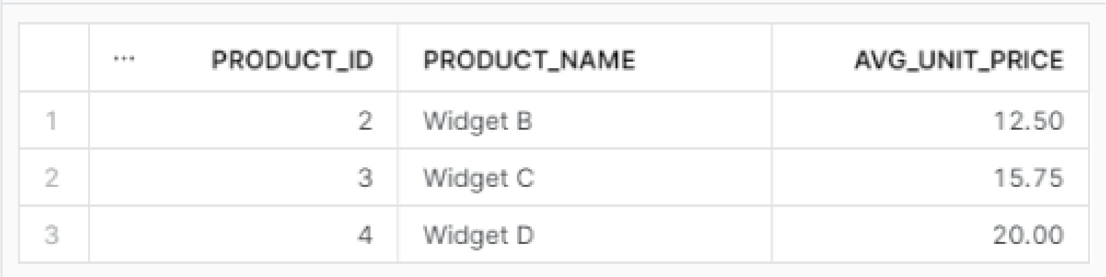


	AVG_UNIT_PRICE
1	12.90

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

```
SELECT PRODUCT_ID, PRODUCT_NAME,ROUND(AVG(UNIT_PRICE),2) AS AVG_UNIT_PRICE

FROM SALES

GROUP BY 1,2

HAVING AVG_UNIT_PRICE > 12;
```

Results:-

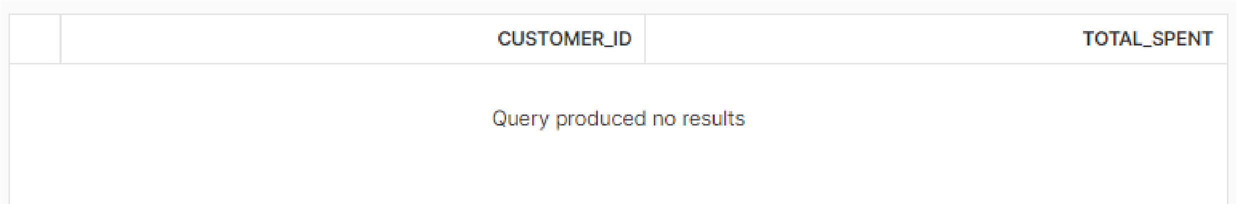


	··· PRODUCT_ID	PRODUCT_NAME	AVG_UNIT_PRICE
1	2	Widget B	12.50
2	3	Widget C	15.75
3	4	Widget D	20.00

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

```
SELECT CUSTOMER_ID,SUM(QUANTITY*UNIT_PRICE) AS TOTAL_SPENT

FROM SALES

GROUP BY 1

HAVING TOTAL_SPENT > 100;
```

Results:- As there is no customers who have spent more than $100 in total ,So no query is produced as a result .



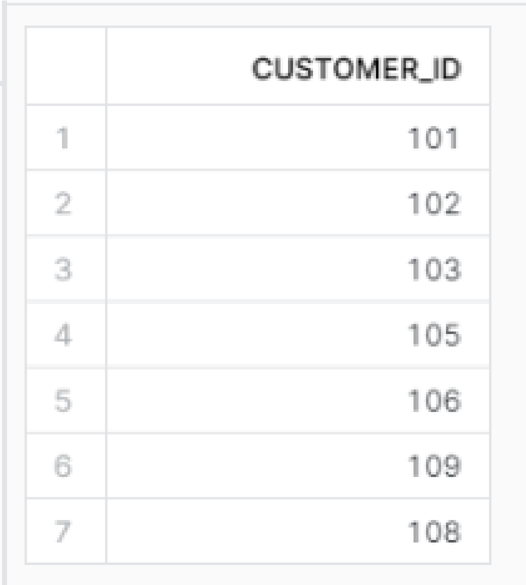
	CUSTOMER_ID	TOTAL_SPENT

Query produced no results

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

```
SELECT CUSTOMER_ID FROM
(SELECT DISTINCT ORDER_ID, CUSTOMER_ID,PRODUCT_NAME
FROM SALES
WHERE PRODUCT_NAME IN ('Widget A','Widget B'));
```

Results:-



	CUSTOMER_ID
1	101
2	102
3	103
4	105
5	106
6	109
7	108