Questions

1. Retrieve the order ID, customer IDs and customer names and total amounts for orders that have a total amount greater than $1000.

```
SELECT a.order_id, a.customer_id, b.customer_name, a.total_amount FROM orders AS a
JOIN customers AS b
ON a.customer_id = b.customer_id
WHERE a.total_amount>1000
```

	order_id integer	customer_id integer	customer_name character varying (50)	total_amount numeric
1	3	3	Michael Johnson	2597
2	5	5	David Brown	2897

2. Retrieve the total quantity of each product sold.

```
WITH CT AS(
        SELECT a.product_id,b.product_name,b.description FROM orders_details As a
        JOIN products AS b
        ON a.product_id = b.product_id
)

SELECT product_id, product_name,
COUNT(product_id) AS count
FROM CT
GROUP BY product_name, product_id
ORDER BY product_id
```

	product_id integer	product_name character varying (50)	count bigint
1	1	iPhone X	3
2	2	Galaxy S9	1
3	3	iPad Pro	2
4	4	Pixel 4a	2
5	5	MacBook Air	1

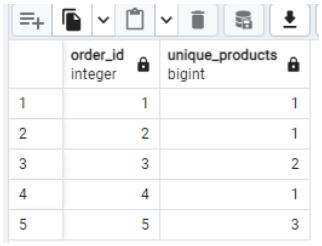
3. Retrieve the order details (order ID, product name, quantity) for orders with a quantity greater than the average quantity of all orders.

```
SELECT a.order_id,b.product_name,a.qty
FROM orders_details AS a
JOIN products AS b
ON a.product_id = b.product_id
WHERE qty > (SELECT AVG(qty) FROM orders_details)
```

	order_id integer	product_name character varying (50)	qty integer
1	3	iPad Pro	2

4. Retrieve the order IDs and the number of unique products included in each order.

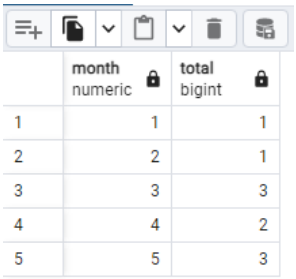
```
SELECT order_id, COUNT(DISTINCT product_id) As unique_products
FROM orders_details
GROUP BY order_id
```



	order_id integer	unique_products bigint
1	1	1
2	2	1
3	3	2
4	4	1
5	5	3

5. Retrieve the total number of products sold for each month in the year 2023. Display the month along with the total number of products.

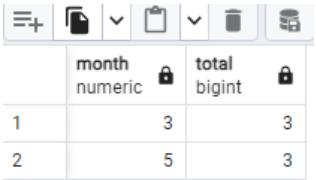
```
SELECT EXTRACT(MONTH FROM a.order_date)AS month,
SUM(b.qty) AS total FROM orders AS a
JOIN orders_details AS b
ON a.order_id = b.order_id
GROUP BY EXTRACT(MONTH FROM a.order_date)
ORDER BY month
```



	month numeric	total bigint
1	1	1
2	2	1
3	3	3
4	4	2
5	5	3

6. Retrieve the total number of products sold for each month in the year 2023 where the total number of products sold were greater than 2. Display the month along with the total number of products.

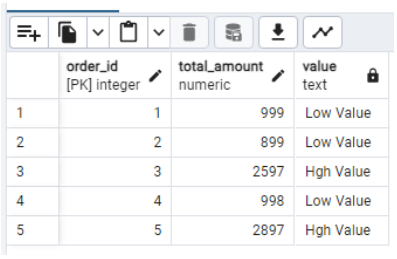
```
WITH ct AS(
        SELECT EXTRACT(MONTH FROM a.order_date)AS month,
        SUM(b.qty) AS total FROM orders AS a
        JOIN orders_details AS b
        ON a.order_id = b.order_id
        GROUP BY EXTRACT(MONTH FROM a.order_date)
        ORDER BY month
)
SELECT * FROM ct  WHERE total > 2
```



	month numeric	total bigint
1	3	3
2	5	3

7. Retrieve the order IDs and the order amount based on the following criteria:
    a. If the total amount > 1000 then ‘High Value’
    b. If it is less than or equal to 1000 then ‘Low Value’
    c. Output should be — order IDs, order amount and Value

```
SELECT  order_id, total_amount,
CASE
    WHEN total_amount>1000 THEN 'Hgh Value'
    ELSE 'Low Value'
END AS value
FROM orders
```

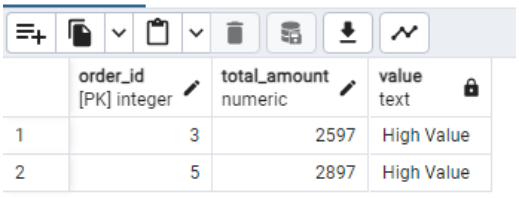


	order_id [PK] integer	total_amount numeric	value text
1	1	999	Low Value
2	2	899	Low Value
3	3	2597	Hgh Value
4	4	998	Low Value
5	5	2897	Hgh Value

8. Retrieve the order IDs and the order amount based on the following criteria:
    a. If the total_amount > 1000 then ‘High Value’
    b. If it is less than 1000 then ‘Low Value’
    c. If it is equal to 1000 then ‘Medium Value’
    Also, please only print the ‘High Value’ products. Output should be — order IDs, order amount and Value.

```
WITH ct AS(
        SELECT  order_id, total_amount,
CASE
        WHEN total_amount>1000 THEN 'High Value'
        WHEN total_amount = 1000 THEN 'Medium Value'
        ELSE 'Low Value'
END AS value
FROM orders
        )

SELECT * FROM ct WHERE value = 'High Value'
```



	order_id [PK] integer	total_amount numeric	value text
1	3	2597	High Value
2	5	2897	High Value