Table Structure: Create a table named sales with the following structure:

CREATE TABLE 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 Data: Insert the following sample data into the sales table:
INSERT INTO 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');

--1. Retrieve the total sales quantity and revenue for each product.
select product_id, sum(quantity) as total_sales_quantity, sum(quantity*unit_price) as revenue
from sales
group by product_id

```
190
191   --1. Retrieve the total sales quantity and revenue for each product.
192   select product_id, sum(quantity) as total_sales_quantity, sum(quantity*unit_price) as revenue
193   from sales
194   group by product_id
195
```

Results Chart

	PRODUCT_ID	TOTAL_SALES_QUANTITY	REVENUE
1	1	11	110.00
2	2	11	137.50
3	3	3	47.25
4	4	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 customer_id

```
196  --2. Find the total revenue for each customer.
197  select customer_id, sum(quantity*unit_price) as total_revenue
198  from sales
199  group by customer_id
200
```

Results | Chart

	CUSTOMER_ID	TOTAL_REVENUE
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	109	10.00
9	108	62.50

--3. Get the products with more than 10 units sold in a single order.
select order_id, product_id from sales where quantity>10

Result - There are no products with 10 units sold in single order

```
200
201  --3. Get the products with more than 10 units sold in a single order.
202  select order_id, product_id from sales where quantity>10
203  Result - There are no products with 10 units sold in single order
```

Results | Chart

ORDER_ID	PRODUCT_ID

Query produced no results

--4. List the customers who have placed orders on at least three different dates.
select customer_id
from sales
group by customer_id
having count(distinct order_date)>3

```
204
205  --4. List the customers who have placed orders on at least three different dates.
206  select customer_id
207  from sales
208  group by customer_id
209  having count(distinct order_date)>3
```

Results | Chart

CUSTOMER_ID

Query produced no results

--5. Calculate the average unit price of products.
select product_name, avg(unit_price) as average_unit_price from sales
group by product_name

```
211  --5. Calculate the average unit price of products.
212  select product_name, avg(unit_price) as average_unit_price from sales
213  group by product_name
214
```

Results | Chart

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

--6. Find the products with an average unit price greater than $12.00.
select product_name, round(avg(unit_price),2) as average_unit_price from sales
group by product_name
having avg(unit_price)>12.00

```
214
215  --6. Find the products with an average unit price greater than $12.00.
216  select product_name, round(avg(unit_price),2) as average_unit_price from sales
217  group by product_name
218  having avg(unit_price)>12.00
219
```

Results | Chart

	PRODUCT_NAME	AVERAGE_UNIT_PRICE
1	Widget B	12.50
2	Widget C	15.75
3	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_revenue
from sales
group by customer_id
having sum(quantity*unit_price)>100

Result - No customer have spent more than $100

```
219
220  --7. Retrieve the customers who have spent more than $100.00 in total.
221  select customer_id, sum(quantity*unit_price) as total_revenue
222  from sales
223  group by customer_id
224  having sum(quantity*unit_price)>100
225
226  Result - No customer have spent more than $100
```

Results | Chart

	CUSTOMER_ID	TOTAL_REVENUE

Query produced no results

--8. List the customers who have purchased 'Widget B' and 'Widget A' in the same order
select customer_id
from sales
where product_name in ('Widget B','Widget A')
group by customer_id
having count(distinct product_name)>2

Result - No customer have purchased the product in same order