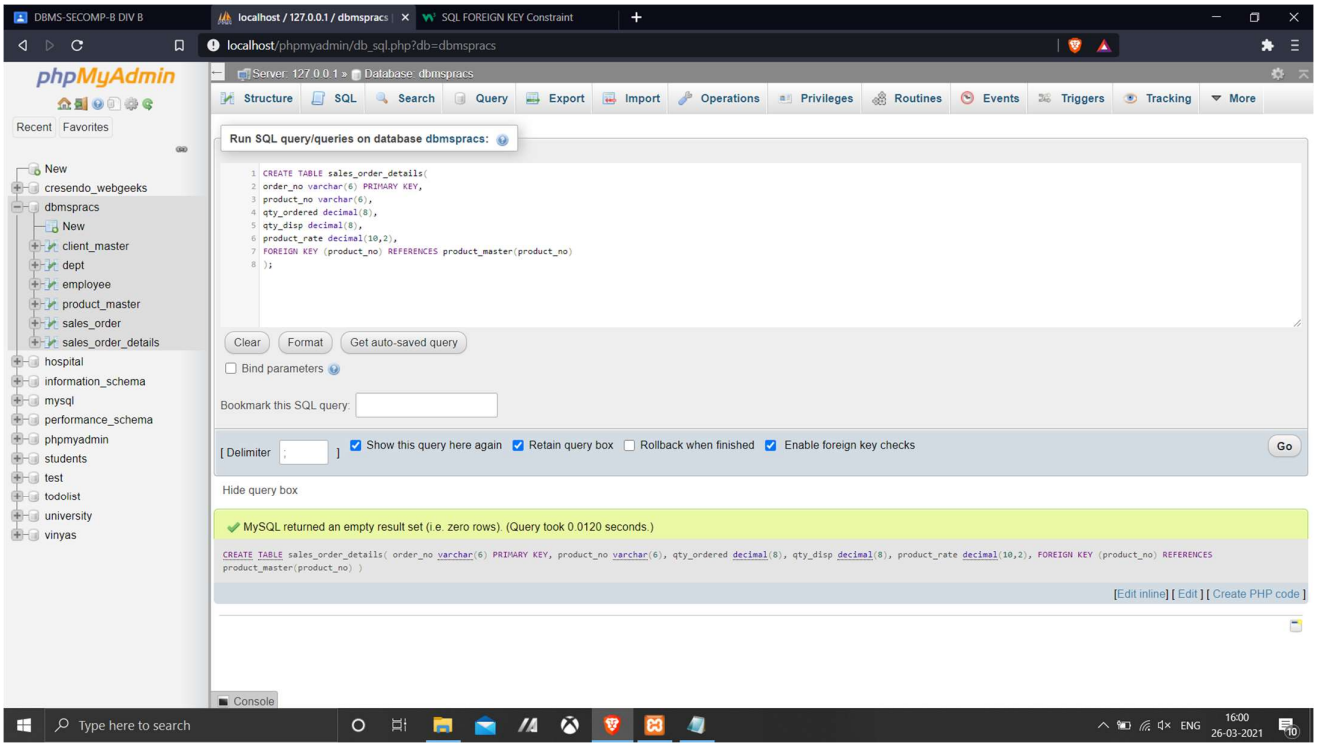
Name: Brendan Lucas, Div: SE COMP B, Roll No: 8953

DBMS Practical Implementation, Lab 6.

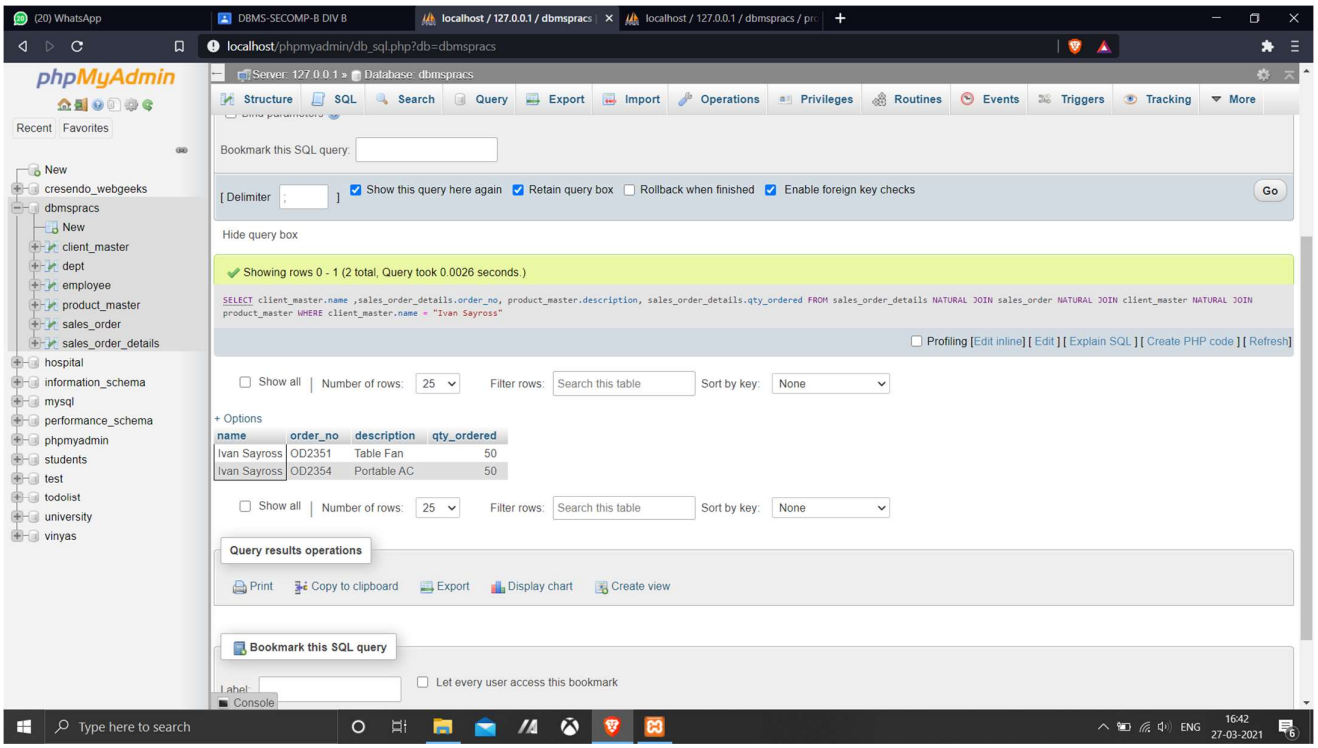
1. To Create Table sales_order_details:

```
CREATE TABLE sales_order_details(
order_no varchar(6) PRIMARY KEY,
product_no varchar(6),
FOREIGN KEY (product_no) REFERENCES product_master(product_no),
qty_ordered decimal(8),
qty_disp decimal(8),
product_rate decimal(10,2)
);
```



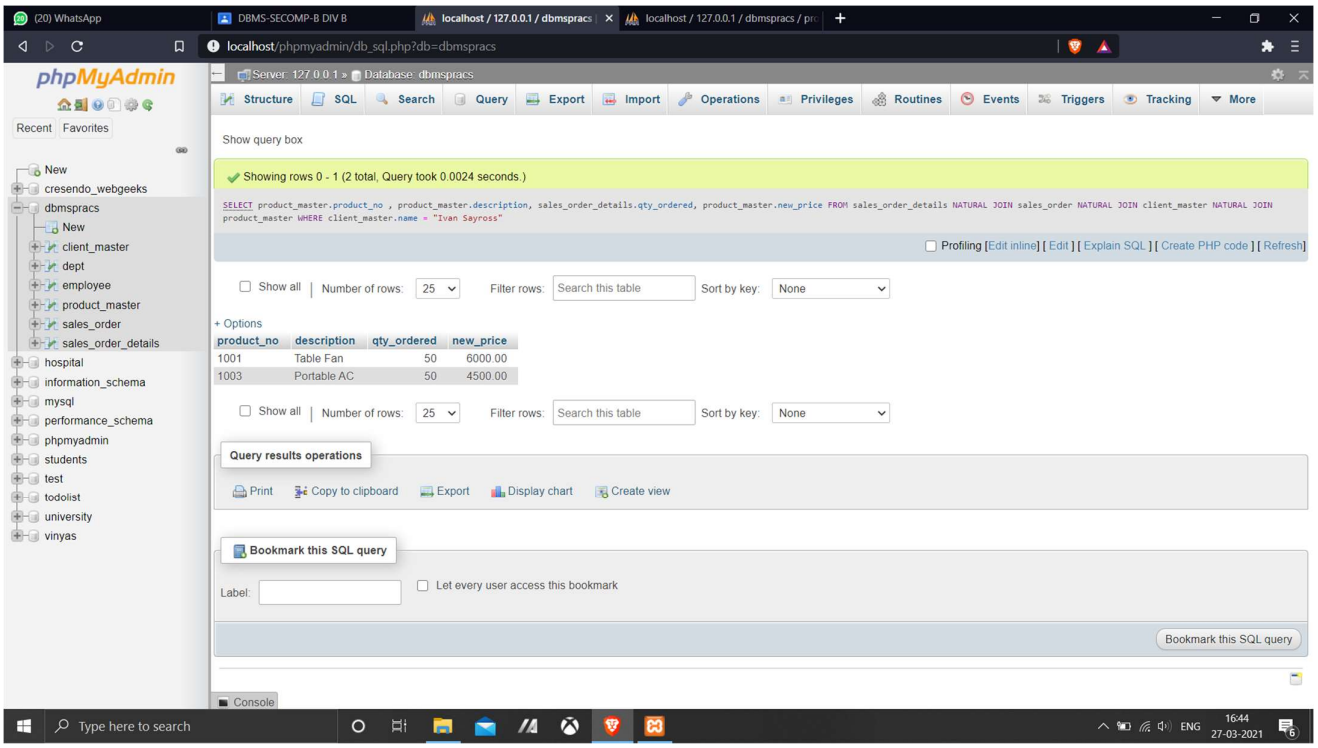
2. Find the products which has been sold to client ‘Ivan Sayross’

```
SELECT client_master.name ,sales_order_details.order_no,
product_master.description, sales_order_details.qty_ordered
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN
client_master NATURAL JOIN product_master
WHERE client_master.name = "Ivan Sayross"
```



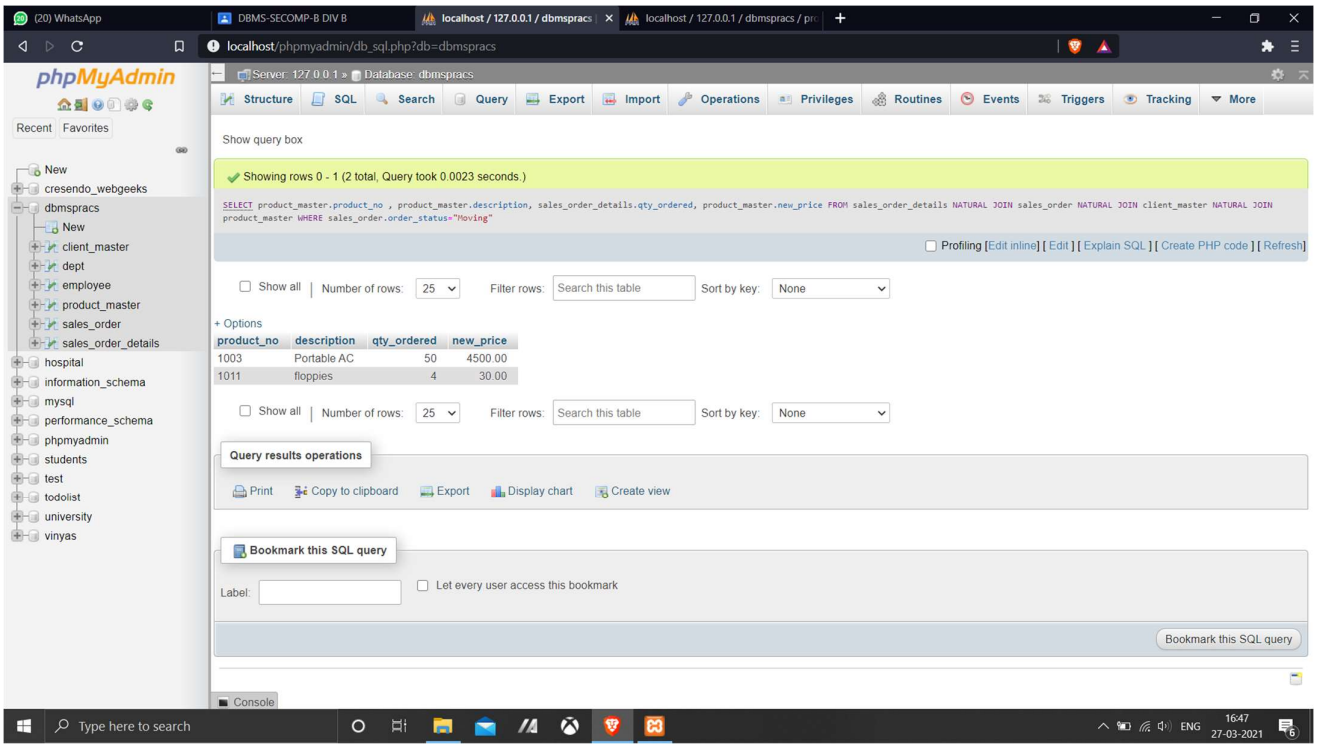
3. Find out product and their quantities that is to be delivered.

```
SELECT product_master.product_no , product_master.description,
sales_order_details.qty_ordered, product_master.new_price
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN
client_master NATURAL JOIN product_master
WHERE client_master.name = "Ivan Sayross"
```



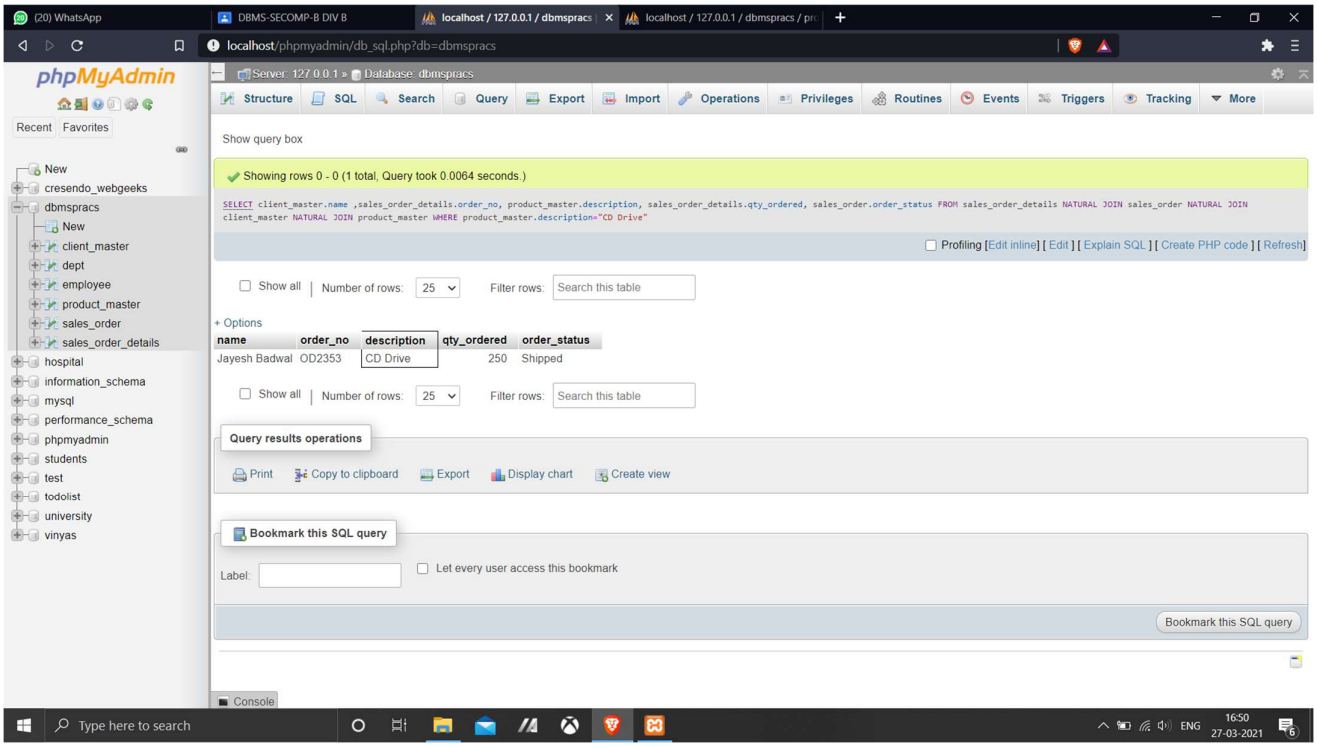
4. Find out the product number and description of Moving products.

```
SELECT product_master.product_no , product_master.description
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN
client_master NATURAL JOIN product_master
WHERE sales_order.order_status="Moving";
```



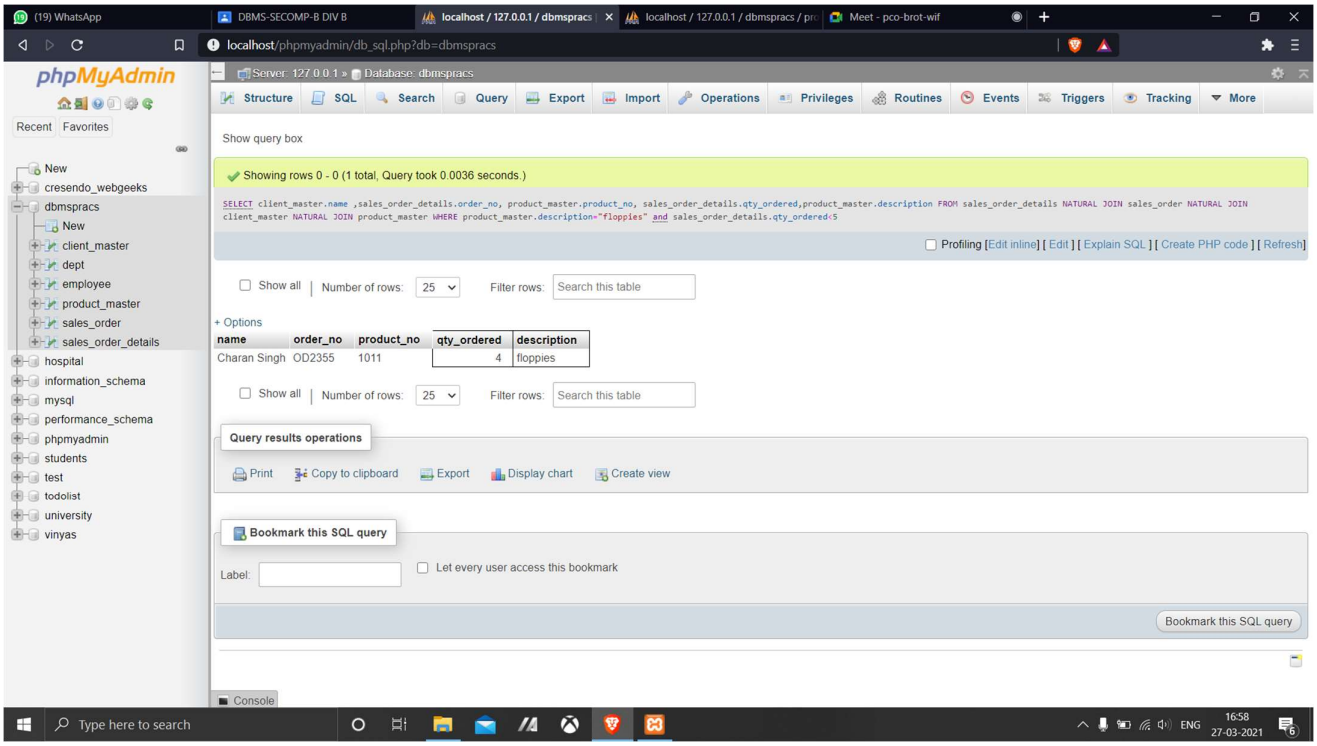
5. Find out the names of clients who have purchased ‘CD Drive’.

```
SELECT client_master.name ,sales_order_details.order_no,
product_master.description, sales_order_details.qty_ordered,
sales_order.order_status
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN
client_master NATURAL JOIN product_master
WHERE product_master.description="CD Drive";
```



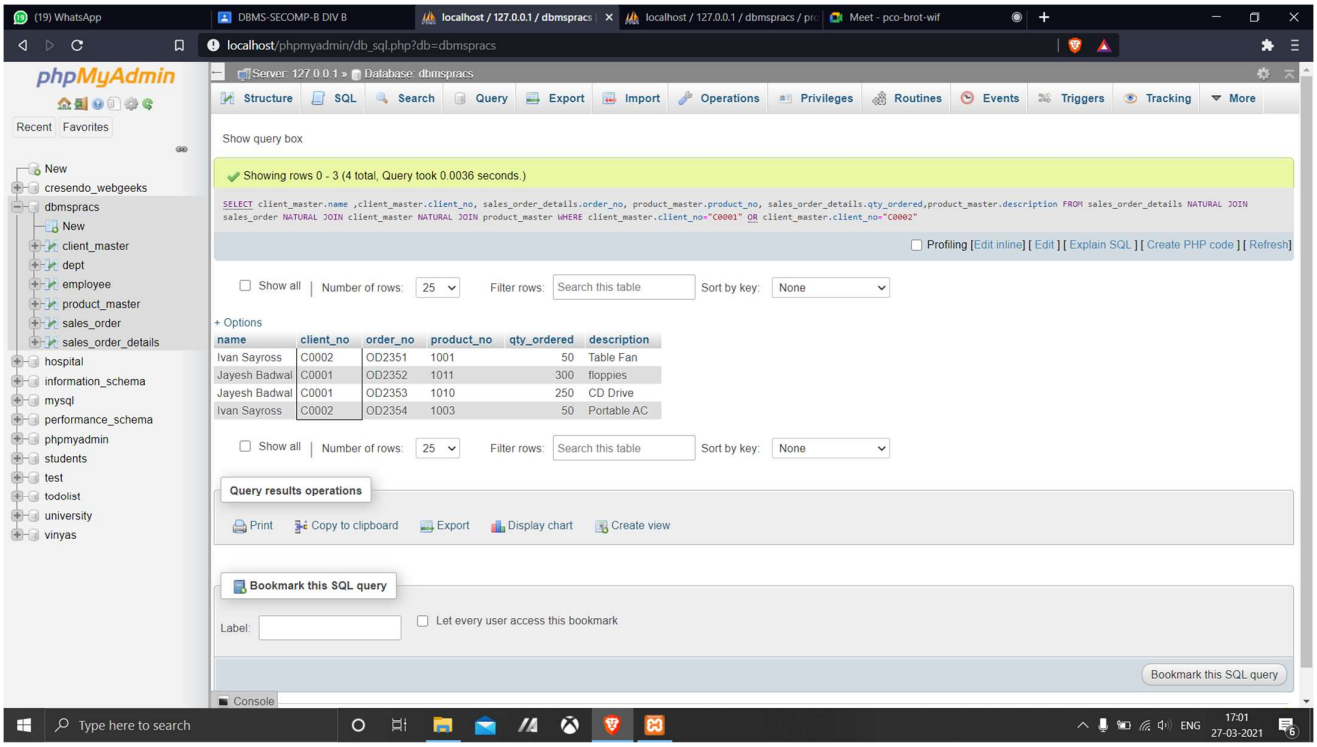
6. List the product_no and order_no of customers having quantity ordered less than 5 form sales_order_details table for the product ‘floppies’.

```
SELECT client_master.name ,sales_order_details.order_no,
product_master.product_no,
sales_order_details.qty_ordered,product_master.description
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN
client_master NATURAL JOIN product_master
WHERE product_master.description="floppies" and
sales_order_details.qty_ordered<5;
```



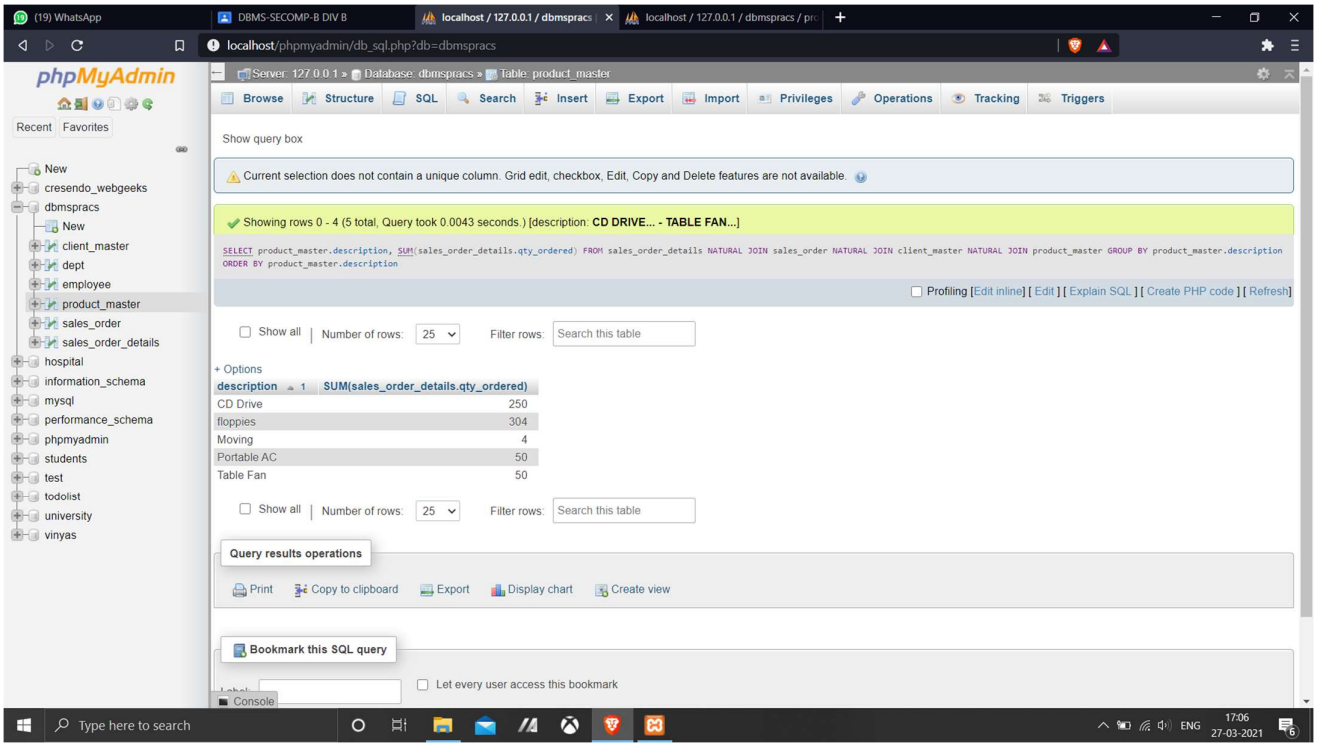
7. Find the products and their quantities for the orders placed by client_no ‘C0001’ and ‘C0002’.

```
SELECT client_master.name ,client_master.client_no,
sales_order_details.order_no, product_master.product_no,
sales_order_details.qty_ordered,product_master.description
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN
client_master NATURAL JOIN product_master
WHERE client_master.client_no="C0001" OR
client_master.client_no="C0002";
```



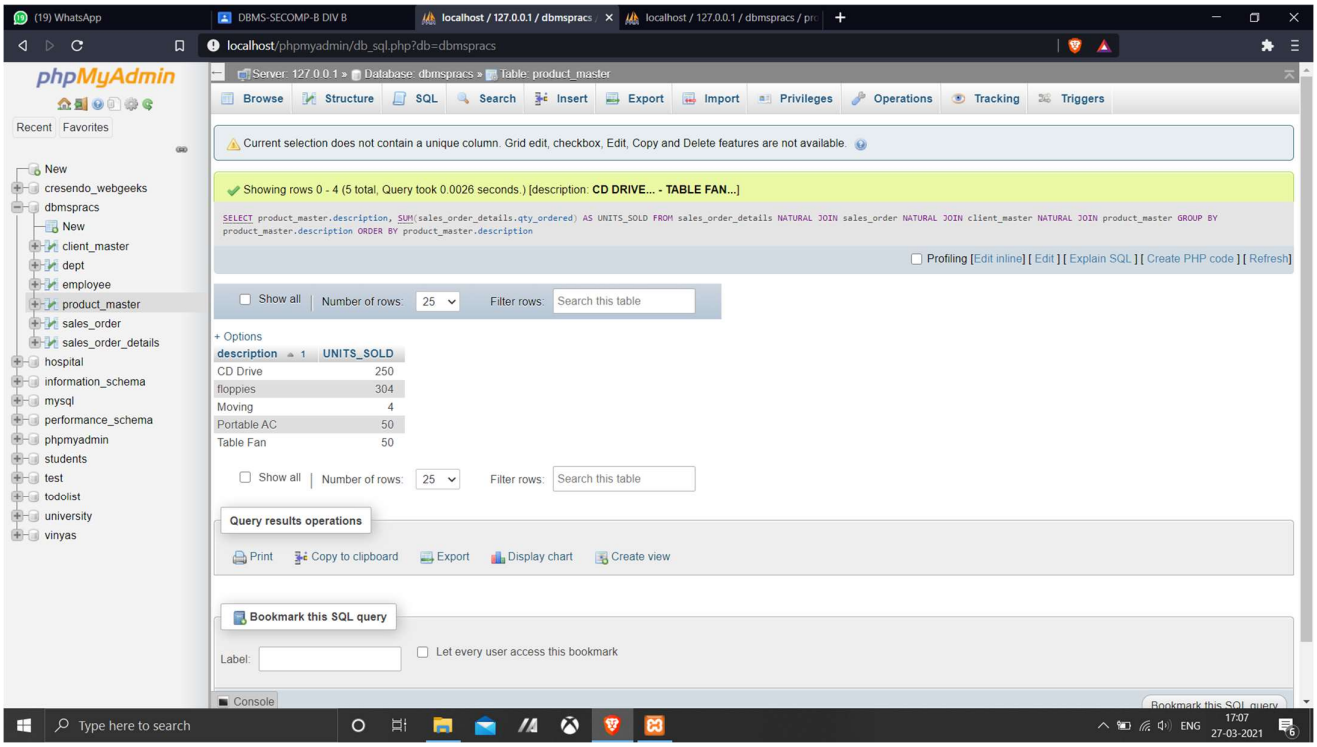
8. Find the description and total quantity sold for each products.

```
SELECT product_master.description, SUM(sales_order_details.qty_ordered) AS
UNITS SOLD
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN
client_master NATURAL JOIN product_master
GROUP BY product_master.description
ORDER BY product_master.description
```



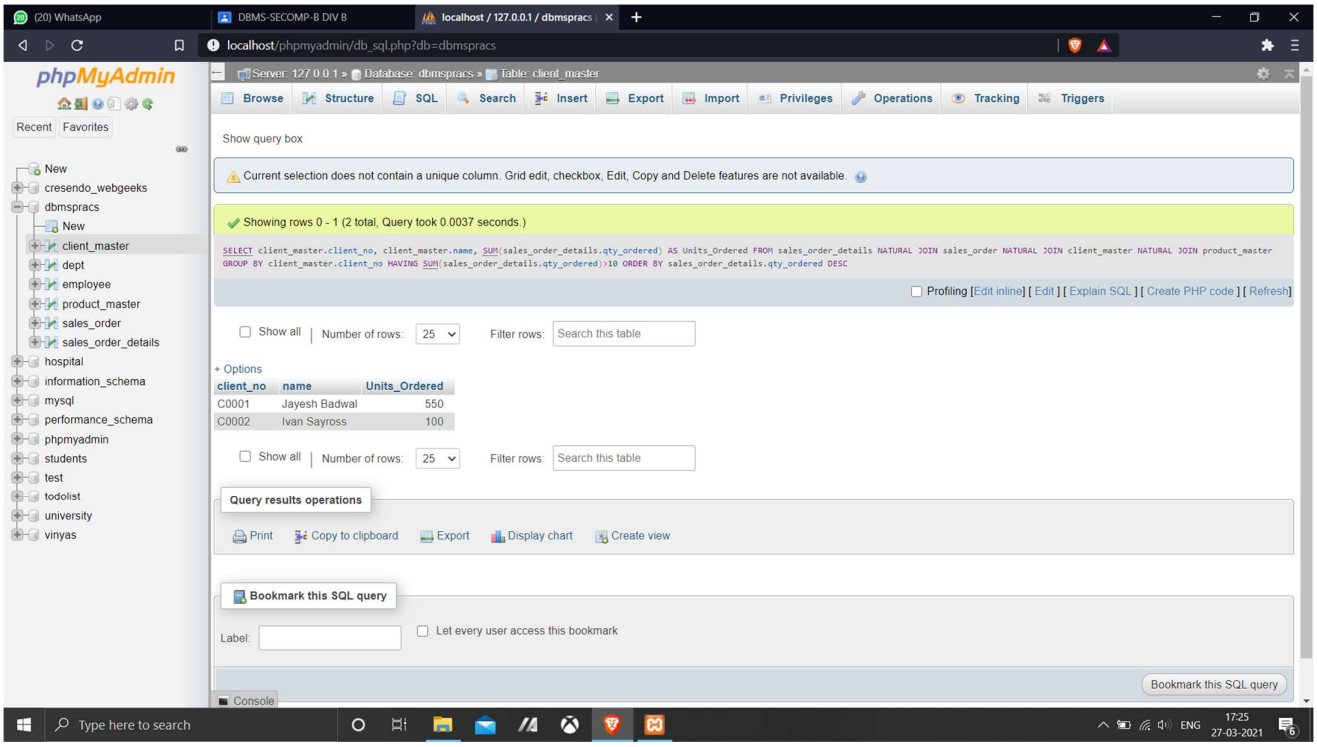
9. Find the value of each product sold.

```
SELECT product_master.description,SUM(sales_order_details.qty_ordered) AS UNITS_SOLD
,SUM(sales_order_details.qty_ordered)*sales_order_details.product_rate AS Value_Recieved
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN client_master NATURAL JOIN product_master
GROUP BY product_master.description
ORDER BY product_master.description
```



10. Find out the name of customers who have given the order of more than 10 qty.

```
SELECT client_master.client_no, client_master.name,
SUM(sales_order_details.qty_ordered) AS Units_Ordered
FROM sales_order_details NATURAL JOIN sales_order NATURAL JOIN
client_master NATURAL JOIN product_master
GROUP BY client_master.client_no
HAVING SUM(sales_order_details.qty_ordered)>10
ORDER BY sales_order_details.qty_ordered DESC
```



Page No.
Date

Postlab EXP 6.

(1) Difference between inner join and outer join

As (i) Inner Join

(a) It returns the combined tuple between two or more tables.

(b) When any attributes are not common then it will ~~to~~ return nothing.

(c) If tuples are more. Then INNER JOIN works faster than OUTER JOIN.

(d) It is used where we want detailed information about any specific attribute.

(e) Used clause INNER JOIN and JOIN.

(ii) Outer Join.

(a) It will return the combined tuple ~~for~~ a specified table even join condition will fail

(b) It does not depend upon the common attributes. If the attributes is blank then here also replaced. NULL

(c) Generally the OUTER JOIN is slower than INNER JOIN. But except for some special cases.

(d) It is used when we want complete information

(e) Used clause LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN etc.

12) EQUI JOIN & Non Equi Join examples.

(i) EQUI Join creates Join by using Join with ON and then providing the names of the column with their relative tables to check equal using equal sign (=).

Example:-

```
SELECT student.name, record.class, record.city
From student, record
WHERE student.city = record.city;
```

(ii) Non EQUI JOIN performs a JOIN using comparison operator other than equal (=) sign like !, <, >=, <= with condition

Example:-

```
SELECT student.name, record.id, record.city
From student, record
WHERE Student.id < Record.id;
```

Exercise:

Choose the correct `JOIN` clause to select all records from the two tables where there is a match in both tables.

```
SELECT *
FROM Orders
Inner Join Customer
ON Orders.CustomerID=
Customers.CustomerID;
```

Submit Answer ❯ Show Answer

Exercise:

Choose the correct `JOIN` clause to select all the records from the `Customers` table plus all the matches in the `Orders` table.

```
SELECT *
FROM Orders
Right Join Customers
ON Orders.CustomerID=
Customers.CustomerID;
```

Submit Answer ❯ Show Answer

Exercise:

List the number of customers in each country.

```
SELECT Count(CustomerID),
Country
FROM Customers
Group by country;
```

Submit Answer ❯ Show Answer

Exercise:

List the number of customers in each country, ordered by the country with the most customers first.

```
SELECT Count(CustomerID),
Country
FROM Customers
Group by country
ORDER BY
COUNT(CustomerID) DESC;
```

Submit Answer ❯ Show Answer

Exercise:

Insert the missing parts in the `JOIN` clause to join the two tables `Orders` and `Customers`, using the `CustomerID` field in both tables as the relationship between the two tables.

Correct!

Next ❯

Next Exercise ❯ Show Answer