EASTERN INTERNATIONAL UNIVERSITY

**SCHOOL OF COMPUTING**

**AND INFORMATION TECHNOLOGY**

□□□

# Practice Assignment 5

**Practice Assignment – Quarter 4, 2022-2023**

**Course Name:** Database

**Course Code:** CSE 301

**Student's Full Name: Phan Ngọc Hạnh Nhi**

**Student ID: 2131209002**

***Instruction*:**

** Students are allowed to write their answers (like SQL queries, Screen shot of outputs, etc.) in word file (Answer sheet) provided by instructor. After finishing the assignment, students must convert the word file (Answer sheet) into a PDF file. Finally, students upload the file in Moodle.*

1. Create the following tables in a new database 'Assignment3':

Clients(**Client_Number**, Client_Name, Address, City, Pincode, Province, Amount_Paid, Amount_Due)

Product(**Product_Number**, Product_Name, Quantity_On_Hand, Quantity_Sell, Sell_Price, Cost_Price)

Salesman (**Salesman_Number**, Salesman _Name, Address, City, Pincode, Province, Salary, Sales_Target, Target_Achieve, Phone)

Salesorder(**Order_Number**, Order_Date**, Client_Number, Salesman_Number**, Delivery_Status, Delivery_Date, Order_Status)

Salesorderdetails(**Order_Number, Product_Number**, Order_Quantity)

**Insert more values data below:**

**Salesman**

('S007','Quang','Chanh My','Da Lat',700032,'Lam Dong',25000,90,95,'0900853487')

('S008','Hoa','Hoa Phu','Thu Dau Mot',700051,'Binh Duong',13500,50,75,'0998213659')

**Salesorder**

('O20015','2022-05-12','C108','S007','On Way', '2022-05-15','Successful')

('O20016','2022-05-16','C109','S008','Ready to Ship',null,'In Process')

**Salesorderdetails**

('O20015','P1008',15),

('O20015','P1007',10),

('O20016','P1007',20);

('O20016','P1003',5);

1. *Using Joining table to combine rows from more tables. (NATURAL JOIN, INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN, SELF JOIN)*

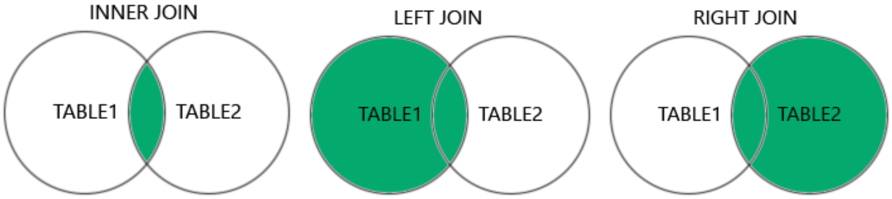
*Supported Types of Joins in MySQL*

*INNER JOIN: Returns records that have matching values in both tables*

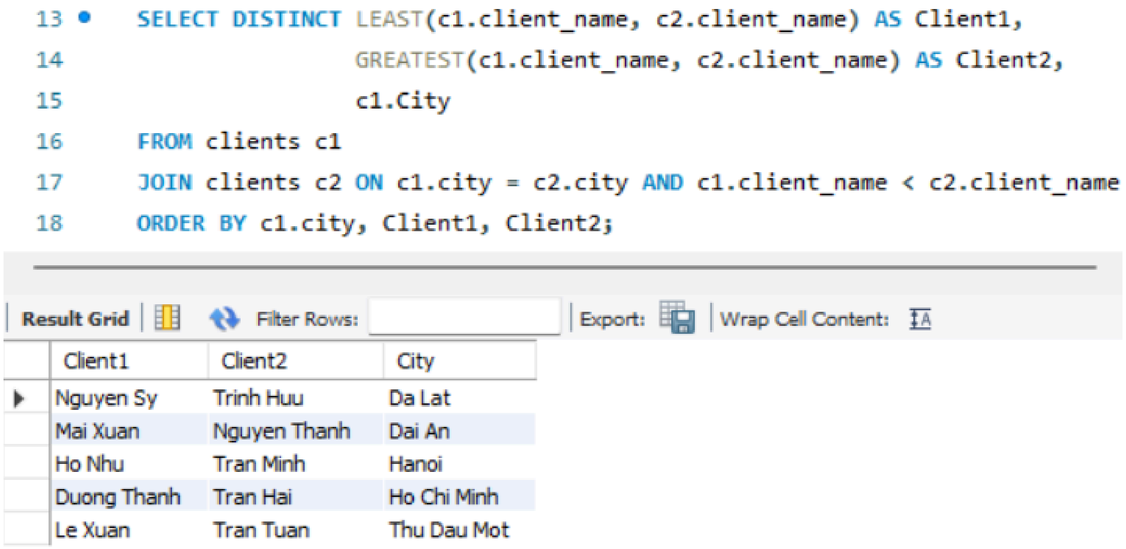
*LEFT JOIN: Returns all records from the left table, and the matched records from the right table*

*RIGHT JOIN: Returns all records from the right table, and the matched records from the left table*

*CROSS JOIN: Returns all records from both tables*



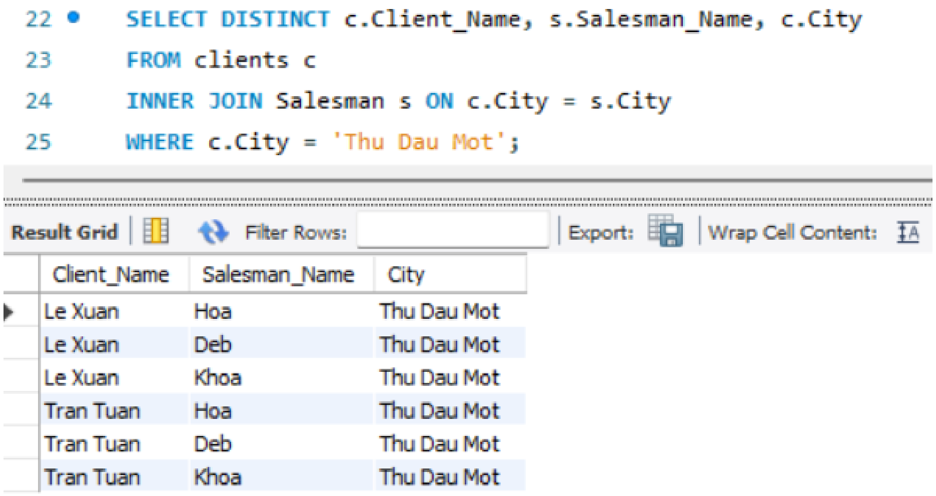
1. Display the clients (name) who lives in same city.



```
SELECT DISTINCT LEAST(c1.client_name, c2.client_name) AS Client1,
                GREATEST(c1.client_name, c2.client_name) AS Client2,
                c1.City
FROM clients c1
JOIN clients c2 ON c1.city = c2.city AND c1.client_name < c2.client_name
ORDER BY c1.city, Client1, Client2;
```

| Client1 | Client2 | City |
|---|---|---|
| Nguyen Sy | Trinh Huu | Da Lat |
| Mai Xuan | Nguyen Thanh | Dai An |
| Ho Nhu | Tran Minh | Hanoi |
| Duong Thanh | Tran Hai | Ho Chi Minh |
| Le Xuan | Tran Tuan | Thu Dau Mot |

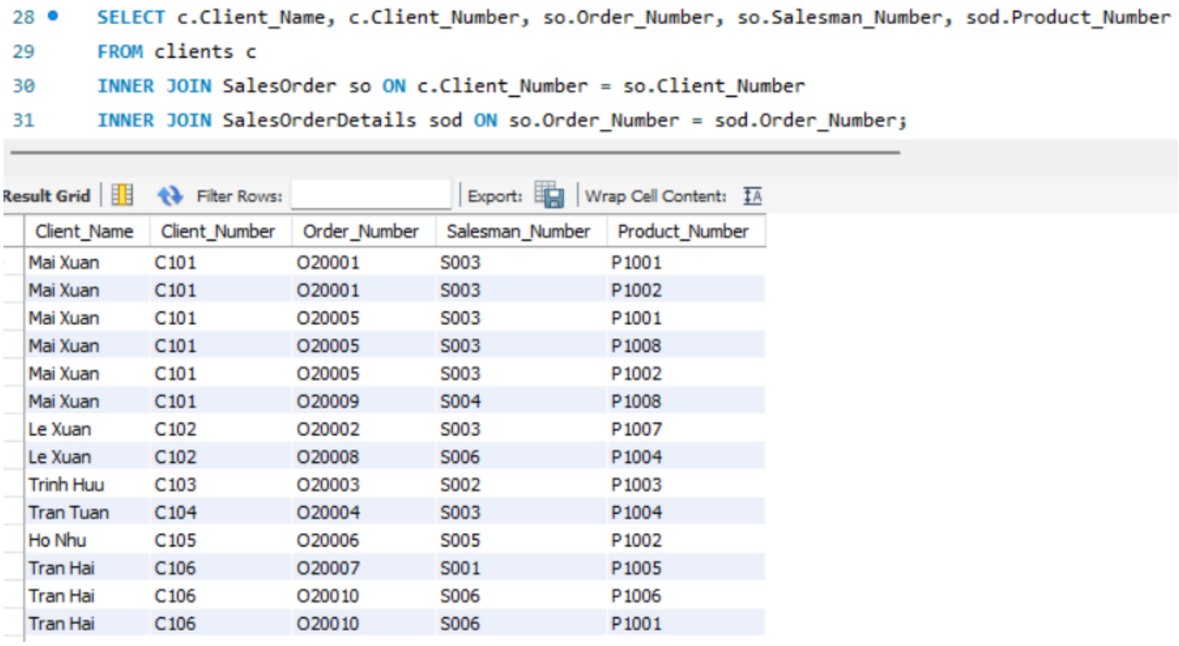
2. Display city, the client names and salesman names who live in “Thu Dau Mot” city.



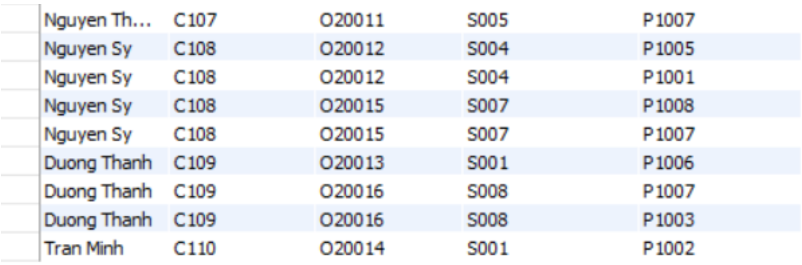
```
SELECT DISTINCT c.Client_Name, s.Salesman_Name, c.City
FROM clients c
INNER JOIN Salesman s ON c.City = s.City
WHERE c.City = 'Thu Dau Mot';
```

| Client_Name | Salesman_Name | City |
|---|---|---|
| Le Xuan | Hoa | Thu Dau Mot |
| Le Xuan | Deb | Thu Dau Mot |
| Le Xuan | Khoa | Thu Dau Mot |
| Tran Tuan | Hoa | Thu Dau Mot |
| Tran Tuan | Deb | Thu Dau Mot |
| Tran Tuan | Khoa | Thu Dau Mot |

3. Display client name, client number, order number, salesman number, and product number for each order.

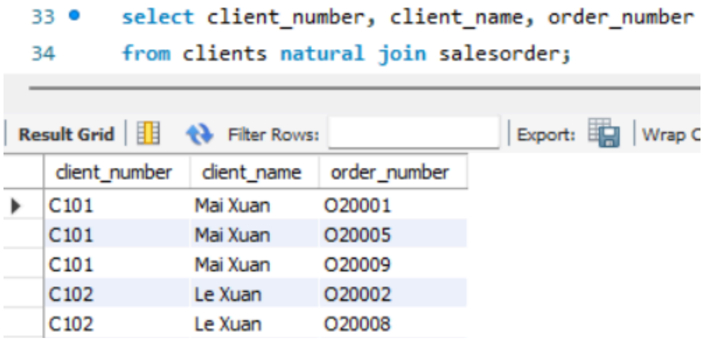


```sql
SELECT c.Client_Name, c.Client_Number, so.Order_Number, so.Salesman_Number, sod.Product_Number
FROM clients c
INNER JOIN SalesOrder so ON c.Client_Number = so.Client_Number
INNER JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number;
```

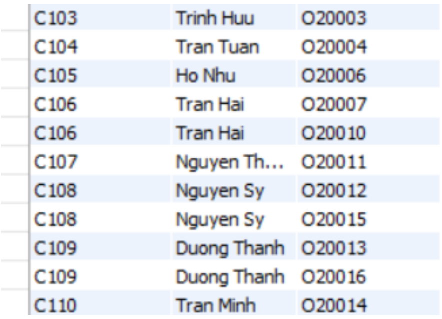


| Client_Name | Client_Number | Order_Number | Salesman_Number | Product_Number |
|---|---|---|---|---|
| Mai Xuan | C101 | O20001 | S003 | P1001 |
| Mai Xuan | C101 | O20001 | S003 | P1002 |
| Mai Xuan | C101 | O20005 | S003 | P1001 |
| Mai Xuan | C101 | O20005 | S003 | P1008 |
| Mai Xuan | C101 | O20005 | S003 | P1002 |
| Mai Xuan | C101 | O20009 | S004 | P1008 |
| Le Xuan | C102 | O20002 | S003 | P1007 |
| Le Xuan | C102 | O20008 | S006 | P1004 |
| Trinh Huu | C103 | O20003 | S002 | P1003 |
| Tran Tuan | C104 | O20004 | S003 | P1004 |
| Ho Nhu | C105 | O20006 | S005 | P1002 |
| Tran Hai | C106 | O20007 | S001 | P1005 |
| Tran Hai | C106 | O20010 | S006 | P1006 |
| Tran Hai | C106 | O20010 | S006 | P1001 |
| Nguyen Th... | C107 | O20011 | S005 | P1007 |
| Nguyen Sy | C108 | O20012 | S004 | P1005 |
| Nguyen Sy | C108 | O20012 | S004 | P1001 |
| Nguyen Sy | C108 | O20015 | S007 | P1008 |
| Nguyen Sy | C108 | O20015 | S007 | P1007 |
| Duong Thanh | C109 | O20013 | S001 | P1006 |
| Duong Thanh | C109 | O20016 | S008 | P1007 |
| Duong Thanh | C109 | O20016 | S008 | P1003 |
| Tran Minh | C110 | O20014 | S001 | P1002 |

4. Find each order (client_number, client_name, order_number) placed by each client.

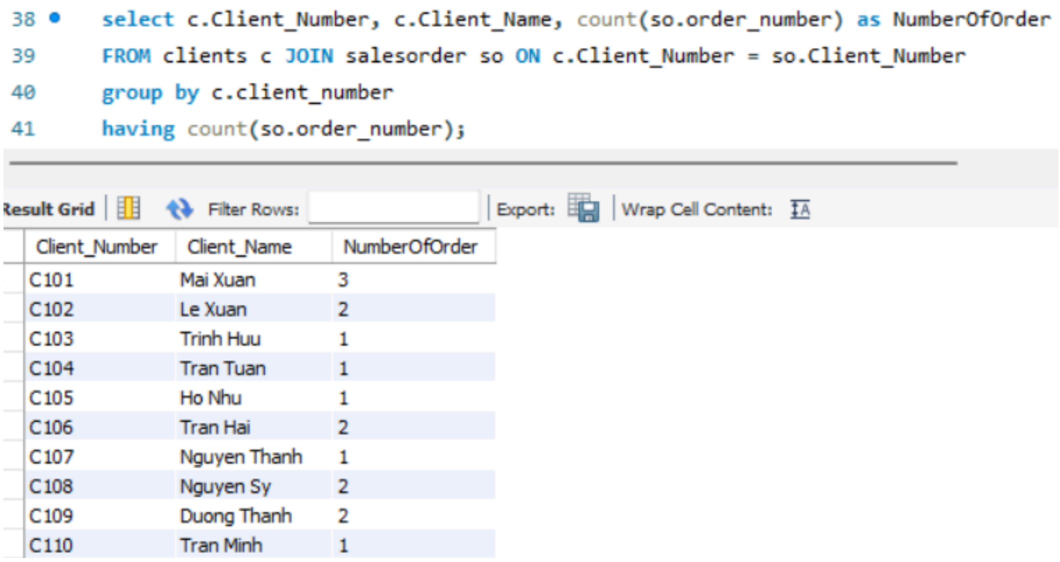


```sql
select client_number, client_name, order_number
from clients natural join salesorder;
```



| client_number | client_name | order_number |
|---|---|---|
| C101 | Mai Xuan | O20001 |
| C101 | Mai Xuan | O20005 |
| C101 | Mai Xuan | O20009 |
| C102 | Le Xuan | O20002 |
| C102 | Le Xuan | O20008 |
| C103 | Trinh Huu | O20003 |
| C104 | Tran Tuan | O20004 |
| C105 | Ho Nhu | O20006 |
| C106 | Tran Hai | O20007 |
| C106 | Tran Hai | O20010 |
| C107 | Nguyen Th... | O20011 |
| C108 | Nguyen Sy | O20012 |
| C108 | Nguyen Sy | O20015 |
| C109 | Duong Thanh | O20013 |
| C109 | Duong Thanh | O20016 |
| C110 | Tran Minh | O20014 |

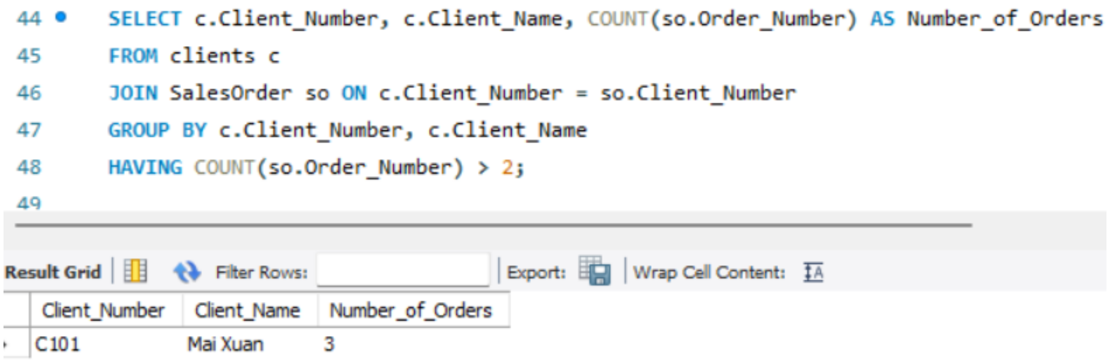
5. Display the details of clients (client_number, client_name) and the number of orders which is paid by them.



```
select c.Client_Number, c.Client_Name, count(so.order_number) as NumberOfOrder
FROM clients c JOIN salesorder so ON c.Client_Number = so.Client_Number
group by c.client_number
having count(so.order_number);
```

| Client_Number | Client_Name | NumberOfOrder |
|---|---|---|
| C101 | Mai Xuan | 3 |
| C102 | Le Xuan | 2 |
| C103 | Trinh Huu | 1 |
| C104 | Tran Tuan | 1 |
| C105 | Ho Nhu | 1 |
| C106 | Tran Hai | 2 |
| C107 | Nguyen Thanh | 1 |
| C108 | Nguyen Sy | 2 |
| C109 | Duong Thanh | 2 |
| C110 | Tran Minh | 1 |

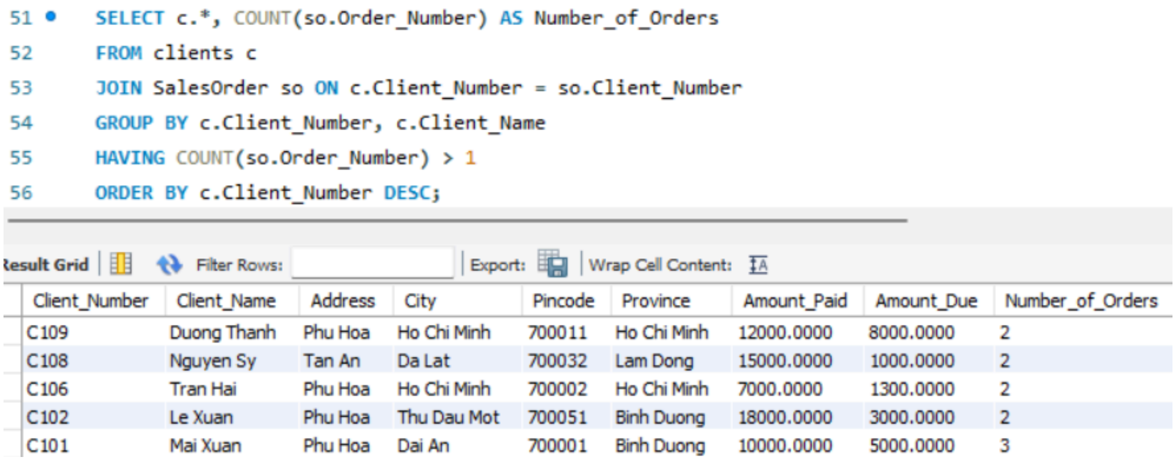
6. Display the details of clients (client_number, client_name) who have paid for more than 2 orders.



```
SELECT c.Client_Number, c.Client_Name, COUNT(so.Order_Number) AS Number_of_Orders
FROM clients c
JOIN SalesOrder so ON c.Client_Number = so.Client_Number
GROUP BY c.Client_Number, c.Client_Name
HAVING COUNT(so.Order_Number) > 2;
```

| Client_Number | Client_Name | Number_of_Orders |
|---|---|---|
| C101 | Mai Xuan | 3 |

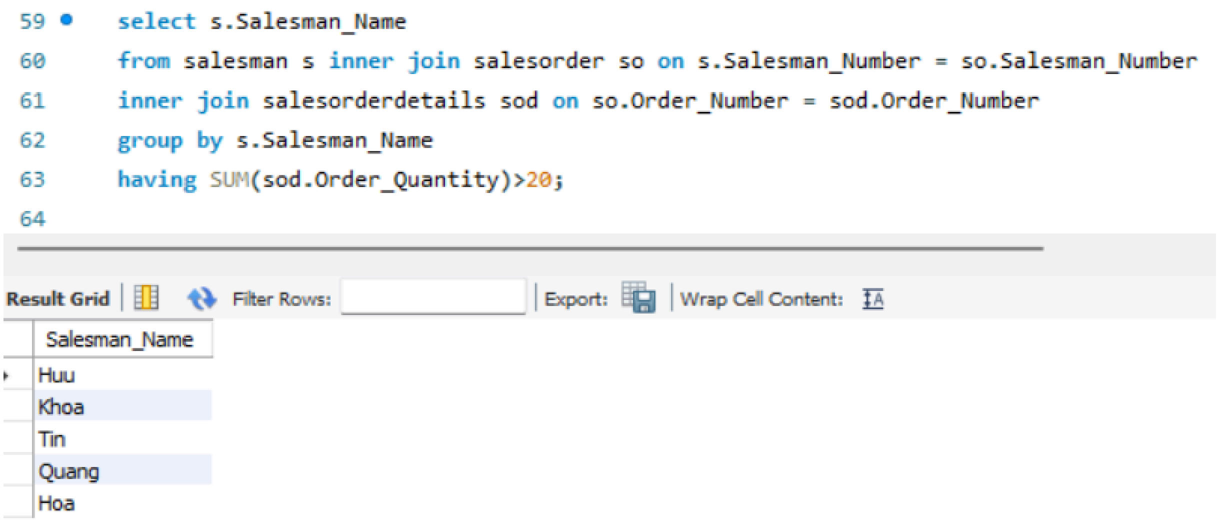
7. Display details of clients who have paid for more than 1 order in descending order of client_number.



```
SELECT c.*, COUNT(so.Order_Number) AS Number_of_Orders
FROM clients c
JOIN SalesOrder so ON c.Client_Number = so.Client_Number
GROUP BY c.Client_Number, c.Client_Name
HAVING COUNT(so.Order_Number) > 1
ORDER BY c.Client_Number DESC;
```

| Client_Number | Client_Name | Address | City | Pincode | Province | Amount_Paid | Amount_Due | Number_of_Orders |
|---|---|---|---|---|---|---|---|---|
| C109 | Duong Thanh | Phu Hoa | Ho Chi Minh | 700011 | Ho Chi Minh | 12000.0000 | 8000.0000 | 2 |
| C108 | Nguyen Sy | Tan An | Da Lat | 700032 | Lam Dong | 15000.0000 | 1000.0000 | 2 |
| C106 | Tran Hai | Phu Hoa | Ho Chi Minh | 700002 | Ho Chi Minh | 7000.0000 | 1300.0000 | 2 |
| C102 | Le Xuan | Phu Hoa | Thu Dau Mot | 700051 | Binh Duong | 18000.0000 | 3000.0000 | 2 |
| C101 | Mai Xuan | Phu Hoa | Dai An | 700001 | Binh Duong | 10000.0000 | 5000.0000 | 3 |

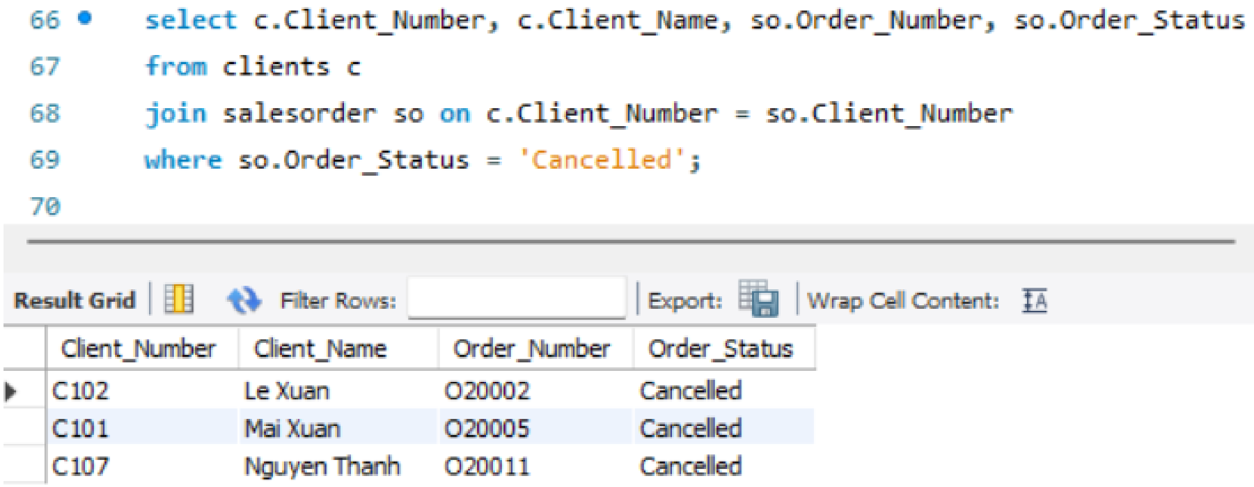
8. Find the salesman names who sells more than 20 products.



```sql
select s.Salesman_Name
from salesman s inner join salesorder so on s.Salesman_Number = so.Salesman_Number
inner join salesorderdetails sod on so.Order_Number = sod.Order_Number
group by s.Salesman_Name
having SUM(sod.Order_Quantity)>20;
```

| Salesman_Name |
|---|
| Huu |
| Khoa |
| Tin |
| Quang |
| Hoa |

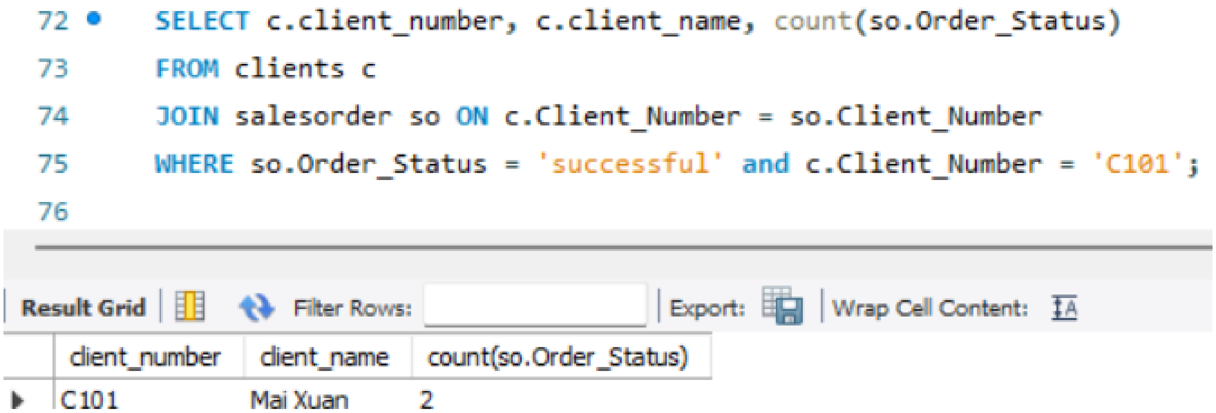
9. Display the client information (client_number, client_name) and order number of those clients who have order status is canceled.



```sql
select c.Client_Number, c.Client_Name, so.Order_Number, so.Order_Status
from clients c
join salesorder so on c.Client_Number = so.Client_Number
where so.Order_Status = 'Cancelled';
```

| Client_Number | Client_Name | Order_Number | Order_Status |
|---|---|---|---|
| C102 | Le Xuan | O20002 | Cancelled |
| C101 | Mai Xuan | O20005 | Cancelled |
| C107 | Nguyen Thanh | O20011 | Cancelled |

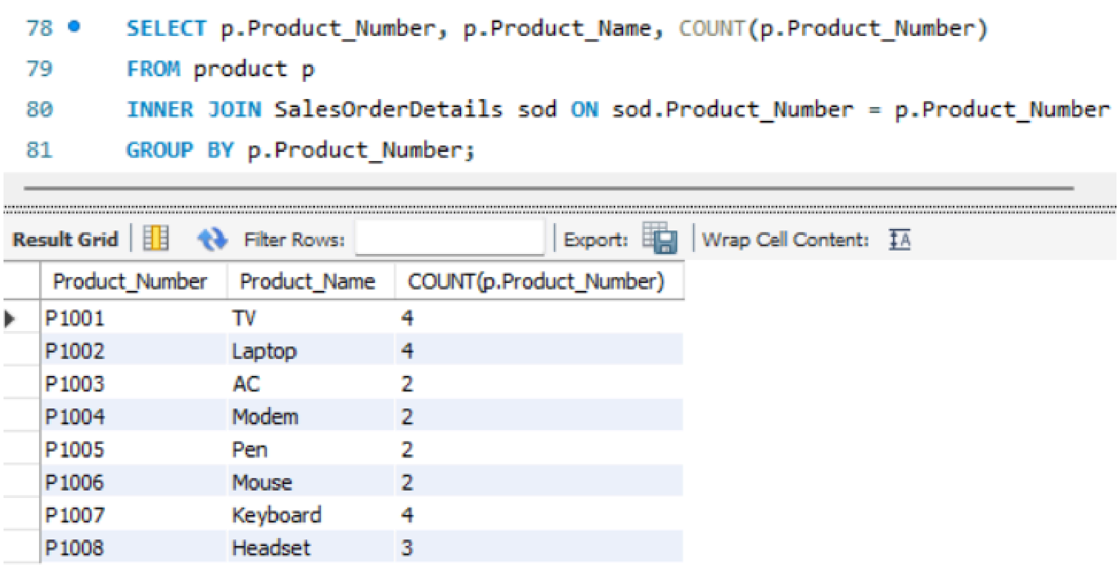
10. Display client name, client number of clients C101 and count the number of orders which were received "successful".



```sql
SELECT c.client_number, c.client_name, count(so.Order_Status)
FROM clients c
JOIN salesorder so ON c.Client_Number = so.Client_Number
WHERE so.Order_Status = 'successful' and c.Client_Number = 'C101';
```

| client_number | client_name | count(so.Order_Status) |
|---|---|---|
| C101 | Mai Xuan | 2 |

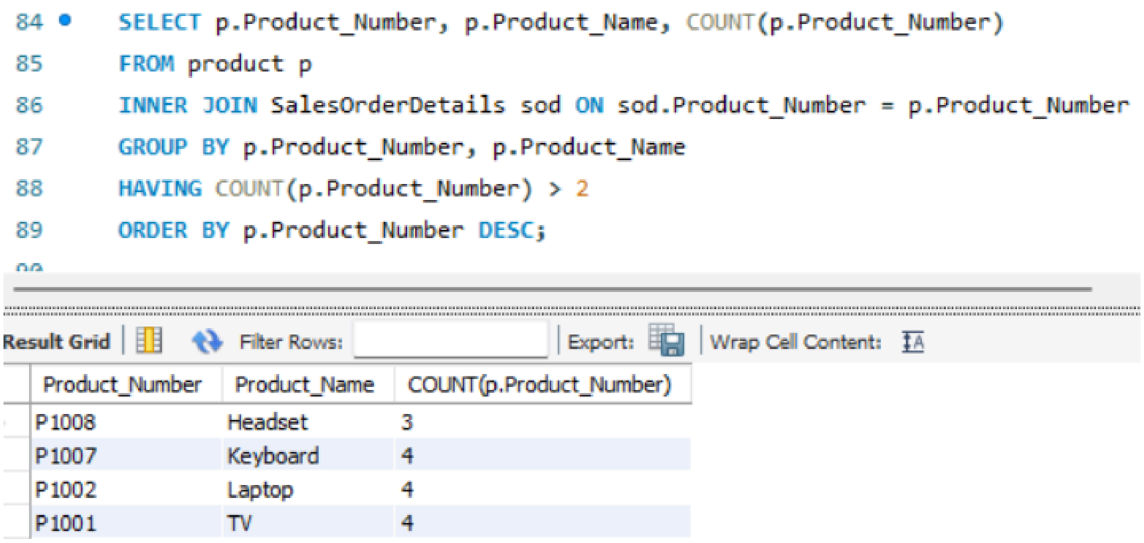
11. Count the number of clients orders placed for each product.



```
SELECT p.Product_Number, p.Product_Name, COUNT(p.Product_Number)
FROM product p
INNER JOIN SalesOrderDetails sod ON sod.Product_Number = p.Product_Number
GROUP BY p.Product_Number;
```

| Product_Number | Product_Name | COUNT(p.Product_Number) |
|---|---|---|
| P1001 | TV | 4 |
| P1002 | Laptop | 4 |
| P1003 | AC | 2 |
| P1004 | Modem | 2 |
| P1005 | Pen | 2 |
| P1006 | Mouse | 2 |
| P1007 | Keyboard | 4 |
| P1008 | Headset | 3 |

12. Find product numbers that were ordered by more than two clients then order in descending by product number.

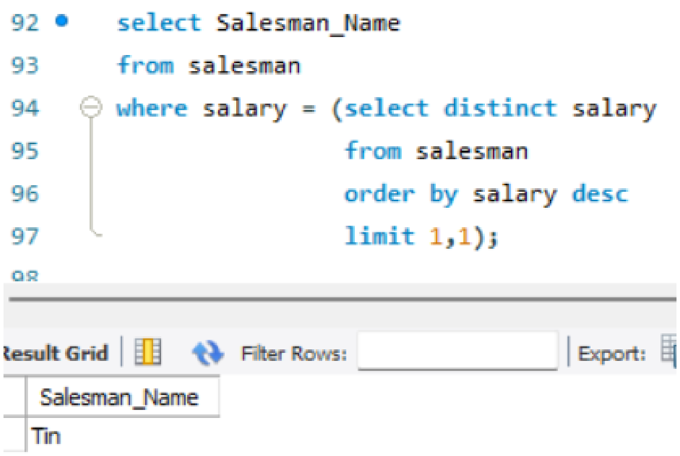


```
SELECT p.Product_Number, p.Product_Name, COUNT(p.Product_Number)
FROM product p
INNER JOIN SalesOrderDetails sod ON sod.Product_Number = p.Product_Number
GROUP BY p.Product_Number, p.Product_Name
HAVING COUNT(p.Product_Number) > 2
ORDER BY p.Product_Number DESC;
```

| Product_Number | Product_Name | COUNT(p.Product_Number) |
|---|---|---|
| P1008 | Headset | 3 |
| P1007 | Keyboard | 4 |
| P1002 | Laptop | 4 |
| P1001 | TV | 4 |

b) *Using nested query with operator (IN, EXISTS, ANY and ALL)*

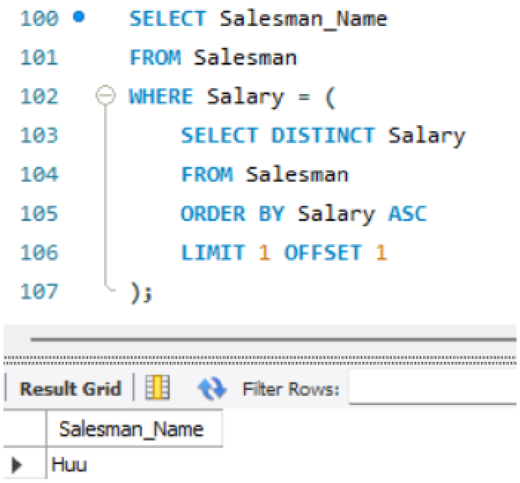
13. Find the salesman's names who is getting the second highest salary.



```
select Salesman_Name
from salesman
where salary = (select distinct salary
                from salesman
                order by salary desc
                limit 1,1);
```

| Salesman_Name |
|---|
| Tin |

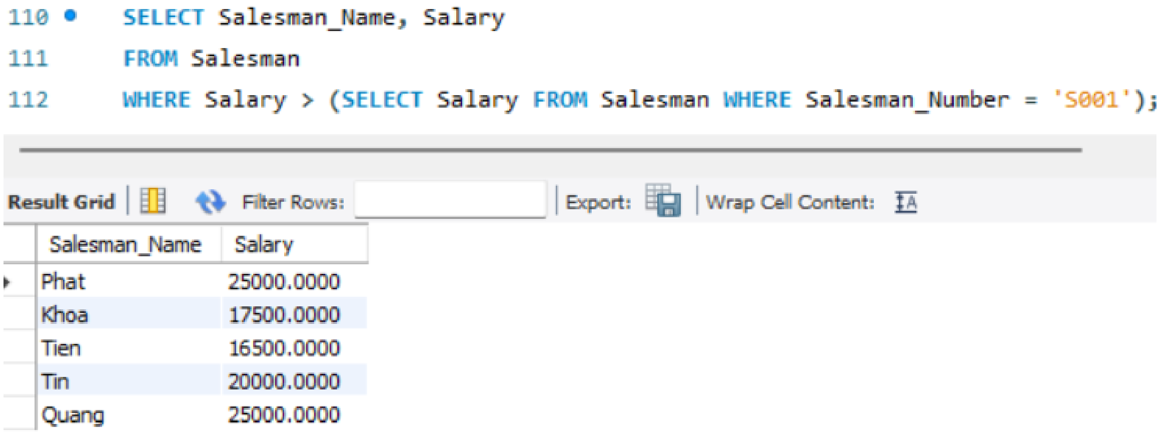
14. Find the salesman's names who is getting second lowest salary.



```sql
SELECT Salesman_Name
FROM Salesman
WHERE Salary = (
    SELECT DISTINCT Salary
    FROM Salesman
    ORDER BY Salary ASC
    LIMIT 1 OFFSET 1
);
```

```
Salesman_Name
Huu
```

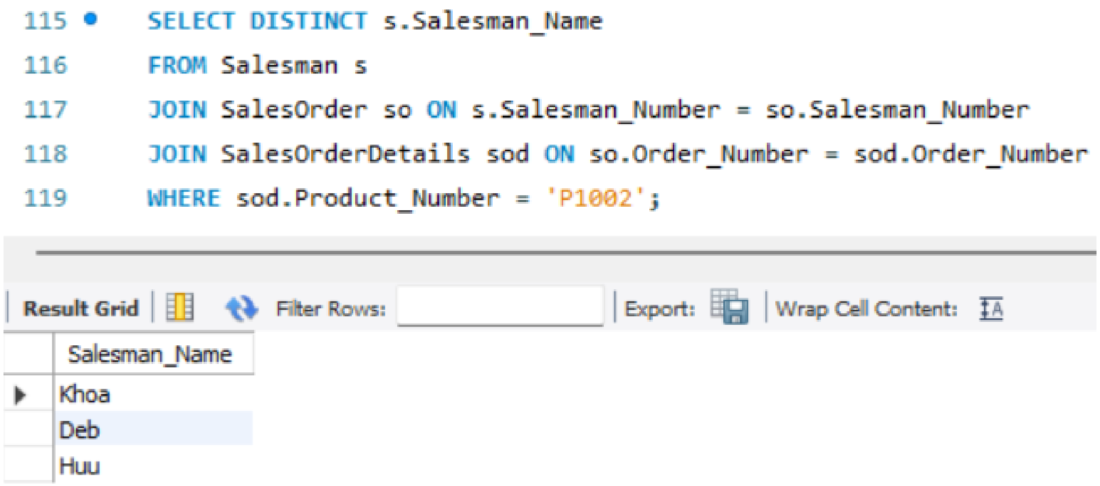
15. Write a query to find the name and the salary of the salesman who have a higher salary than the salesman whose salesman number is S001.



```sql
SELECT Salesman_Name, Salary
FROM Salesman
WHERE Salary > (SELECT Salary FROM Salesman WHERE Salesman_Number = 'S001');
```

```
Salesman_Name  Salary
Phat           25000.0000
Khoa           17500.0000
Tien           16500.0000
Tin            20000.0000
Quang          25000.0000
```

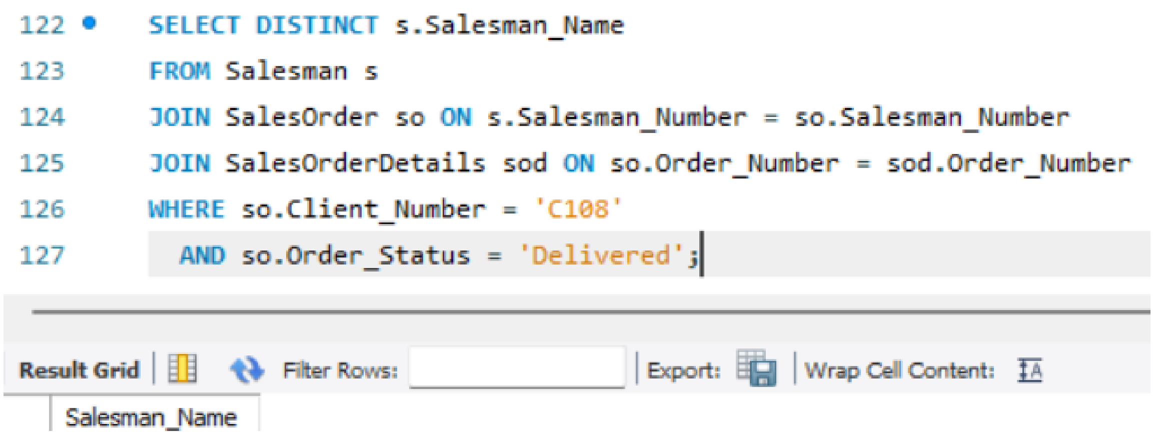
16. Write a query to find the name of all salesman who sold the product has number: P1002.



```sql
SELECT DISTINCT s.Salesman_Name
FROM Salesman s
JOIN SalesOrder so ON s.Salesman_Number = so.Salesman_Number
JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
WHERE sod.Product_Number = 'P1002';
```

```
Salesman_Name
Khoa
Deb
Huu
```

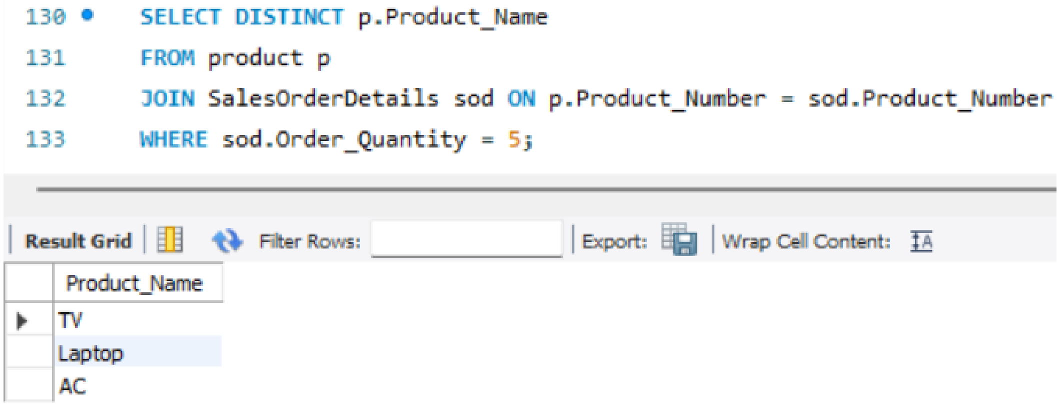
17. Find the name of the salesman who sold the product to client C108 with delivery status is "delivered".



```sql
SELECT DISTINCT s.Salesman_Name
FROM Salesman s
JOIN SalesOrder so ON s.Salesman_Number = so.Salesman_Number
JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
WHERE so.Client_Number = 'C108'
  AND so.Order_Status = 'Delivered';
```

| Salesman_Name |
|---|

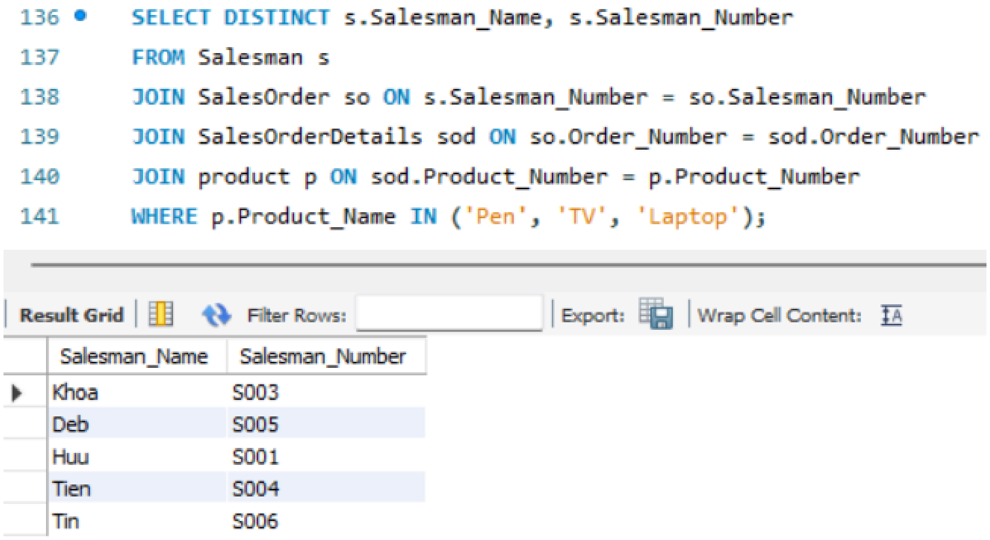
18. Display lists the ProductName in ANY records in the sale Order Details table has Order Quantity equal to 5.



```sql
SELECT DISTINCT p.Product_Name
FROM product p
JOIN SalesOrderDetails sod ON p.Product_Number = sod.Product_Number
WHERE sod.Order_Quantity = 5;
```

| Product_Name |
|---|
| TV |
| Laptop |
| AC |

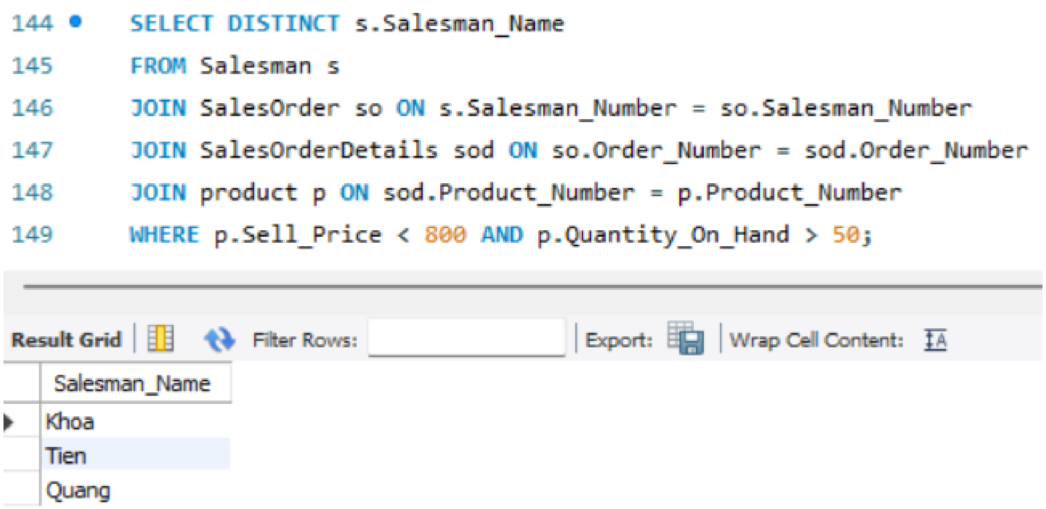
19. Write a query to find the name and number of the salesman who sold pen or TV or laptop.



```sql
SELECT DISTINCT s.Salesman_Name, s.Salesman_Number
FROM Salesman s
JOIN SalesOrder so ON s.Salesman_Number = so.Salesman_Number
JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
JOIN product p ON sod.Product_Number = p.Product_Number
WHERE p.Product_Name IN ('Pen', 'TV', 'Laptop');
```

| Salesman_Name | Salesman_Number |
|---|---|
| Khoa | S003 |
| Deb | S005 |
| Huu | S001 |
| Tien | S004 |
| Tin | S006 |

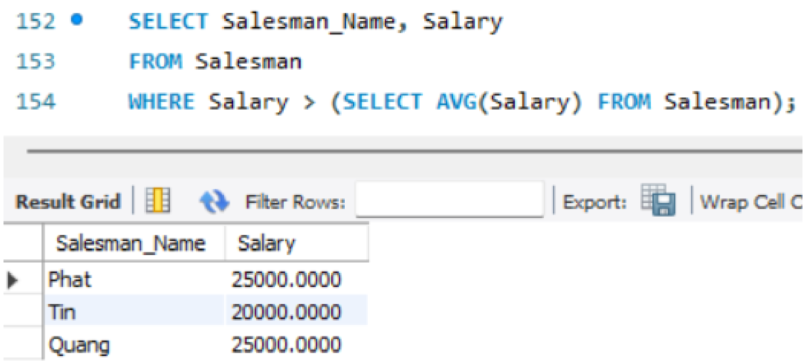
20. Lists the salesman’s name sold product with a product price less than 800 and Quantity_On_Hand more than 50.



```sql
SELECT DISTINCT s.Salesman_Name
FROM Salesman s
JOIN SalesOrder so ON s.Salesman_Number = so.Salesman_Number
JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
JOIN product p ON sod.Product_Number = p.Product_Number
WHERE p.Sell_Price < 800 AND p.Quantity_On_Hand > 50;
```

| Salesman_Name |
|---|
| Khoa |
| Tien |
| Quang |

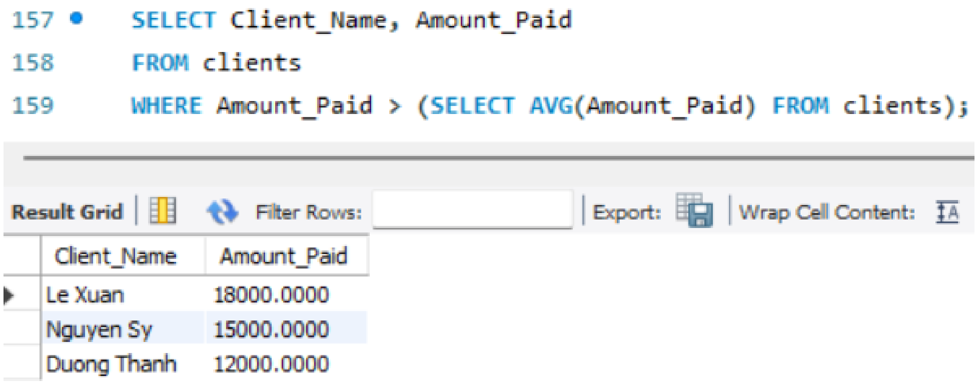
21. Write a query to find the name and salary of the salesman whose salary is greater than the average salary.



```sql
SELECT Salesman_Name, Salary
FROM Salesman
WHERE Salary > (SELECT AVG(Salary) FROM Salesman);
```

| Salesman_Name | Salary |
|---|---|
| Phat | 25000.0000 |
| Tin | 20000.0000 |
| Quang | 25000.0000 |

22. Write a query to find the name and Amount Paid of the clients whose amount paid is greater than the average amount paid.

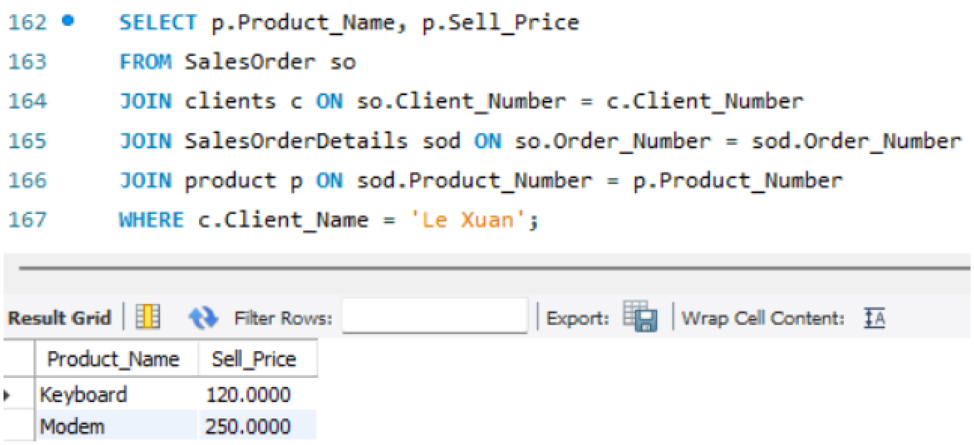


```sql
SELECT Client_Name, Amount_Paid
FROM clients
WHERE Amount_Paid > (SELECT AVG(Amount_Paid) FROM clients);
```

| Client_Name | Amount_Paid |
|---|---|
| Le Xuan | 18000.0000 |
| Nguyen Sy | 15000.0000 |
| Duong Thanh | 12000.0000 |

## II. Additional exercise:

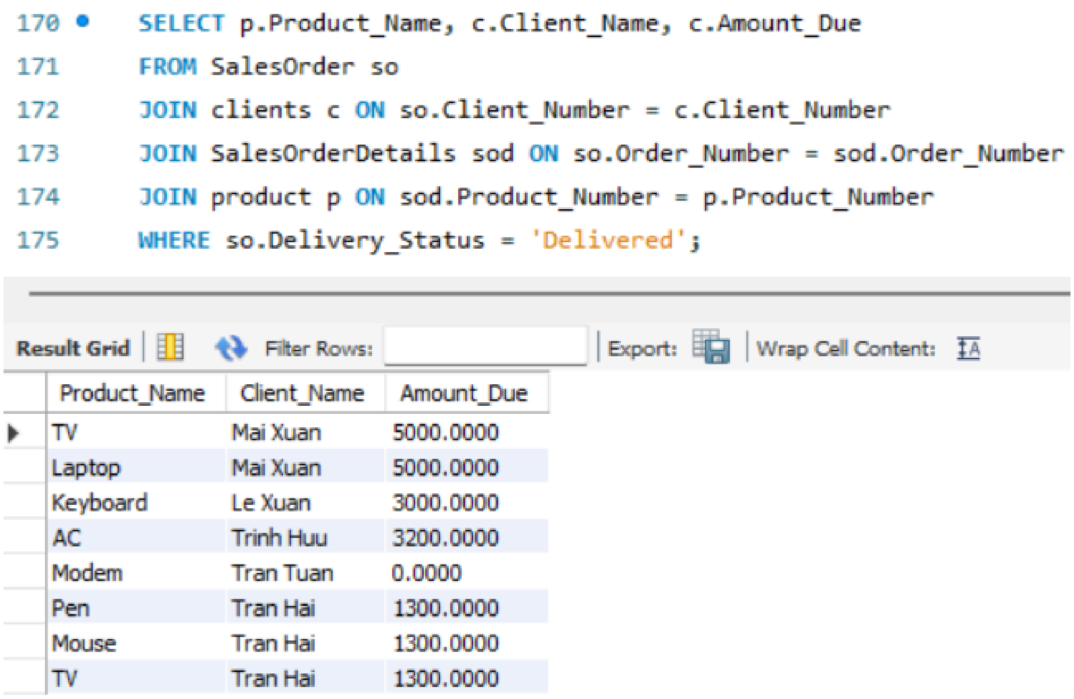
23. Find the product price that was sold to Le Xuan.



```sql
SELECT p.Product_Name, p.Sell_Price
FROM SalesOrder so
JOIN clients c ON so.Client_Number = c.Client_Number
JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
JOIN product p ON sod.Product_Number = p.Product_Number
WHERE c.Client_Name = 'Le Xuan';
```

| Product_Name | Sell_Price |
|---|---|
| Keyboard | 120.0000 |
| Modem | 250.0000 |

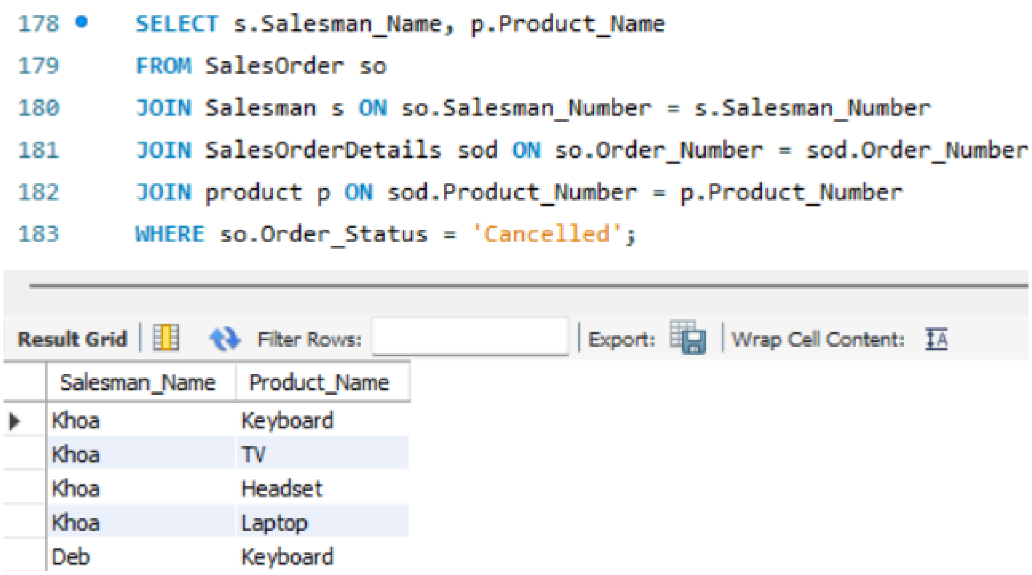
24. Determine the product name, client name and amount due that was delivered.



```sql
SELECT p.Product_Name, c.Client_Name, c.Amount_Due
FROM SalesOrder so
JOIN clients c ON so.Client_Number = c.Client_Number
JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
JOIN product p ON sod.Product_Number = p.Product_Number
WHERE so.Delivery_Status = 'Delivered';
```

| Product_Name | Client_Name | Amount_Due |
|---|---|---|
| TV | Mai Xuan | 5000.0000 |
| Laptop | Mai Xuan | 5000.0000 |
| Keyboard | Le Xuan | 3000.0000 |
| AC | Trinh Huu | 3200.0000 |
| Modem | Tran Tuan | 0.0000 |
| Pen | Tran Hai | 1300.0000 |
| Mouse | Tran Hai | 1300.0000 |
| TV | Tran Hai | 1300.0000 |

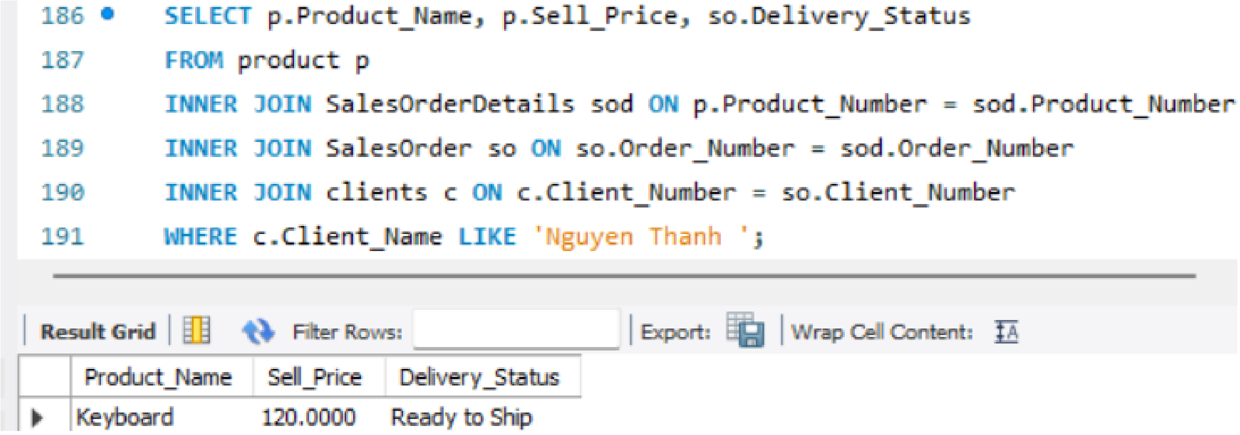
25. Find the salesman's name and their product name which is canceled.



```sql
SELECT s.Salesman_Name, p.Product_Name
FROM SalesOrder so
JOIN Salesman s ON so.Salesman_Number = s.Salesman_Number
JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
JOIN product p ON sod.Product_Number = p.Product_Number
WHERE so.Order_Status = 'Cancelled';
```

| Salesman_Name | Product_Name |
|---|---|
| Khoa | Keyboard |
| Khoa | TV |
| Khoa | Headset |
| Khoa | Laptop |
| Deb | Keyboard |

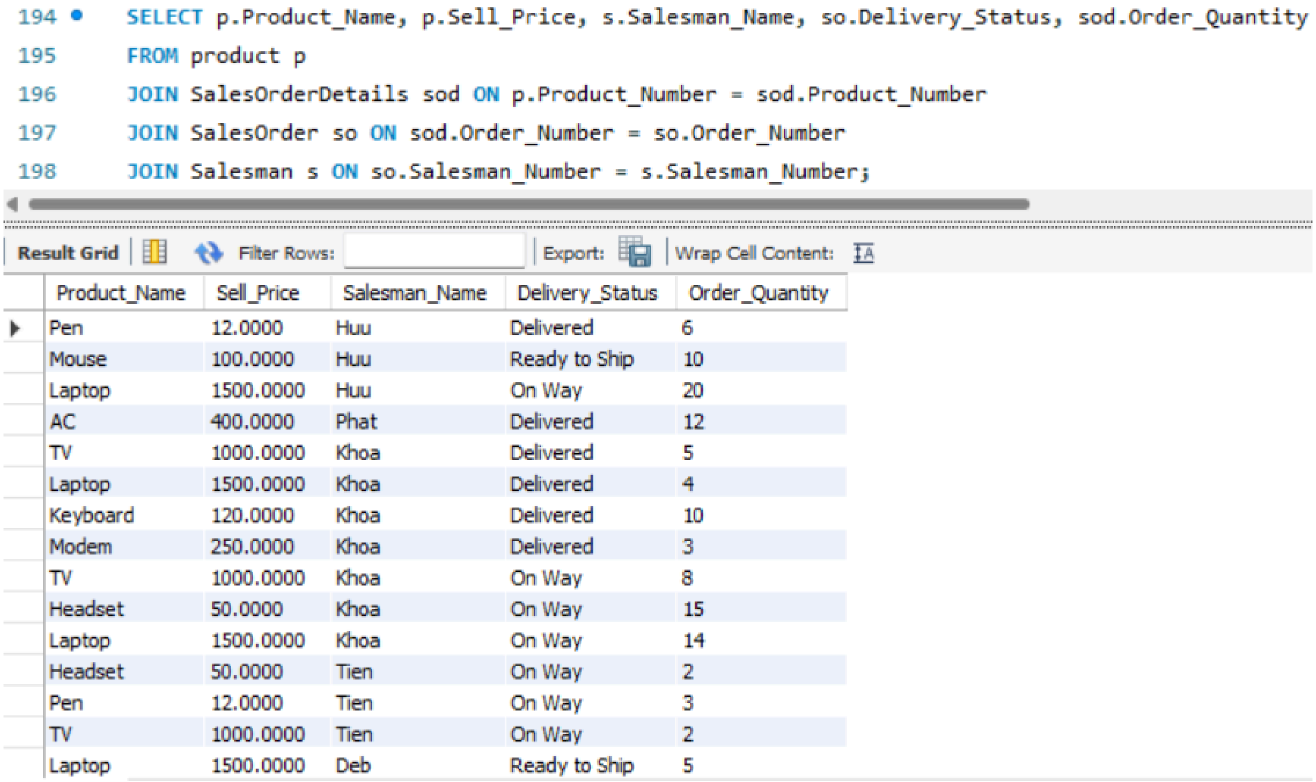
26. Find product names, prices and delivery status for those products purchased by Nguyen Thanh.



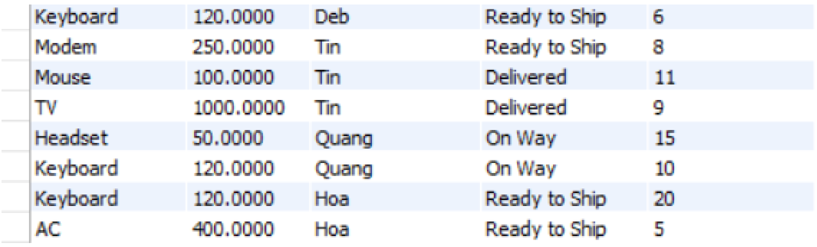
```sql
SELECT p.Product_Name, p.Sell_Price, so.Delivery_Status
FROM product p
INNER JOIN SalesOrderDetails sod ON p.Product_Number = sod.Product_Number
INNER JOIN SalesOrder so ON so.Order_Number = sod.Order_Number
INNER JOIN clients c ON c.Client_Number = so.Client_Number
WHERE c.Client_Name LIKE 'Nguyen Thanh ';
```

| Product_Name | Sell_Price | Delivery_Status |
|---|---|---|
| Keyboard | 120.0000 | Ready to Ship |

27. Display the product name, sell price, salesperson name, delivery status, and order quantity information for each customer.

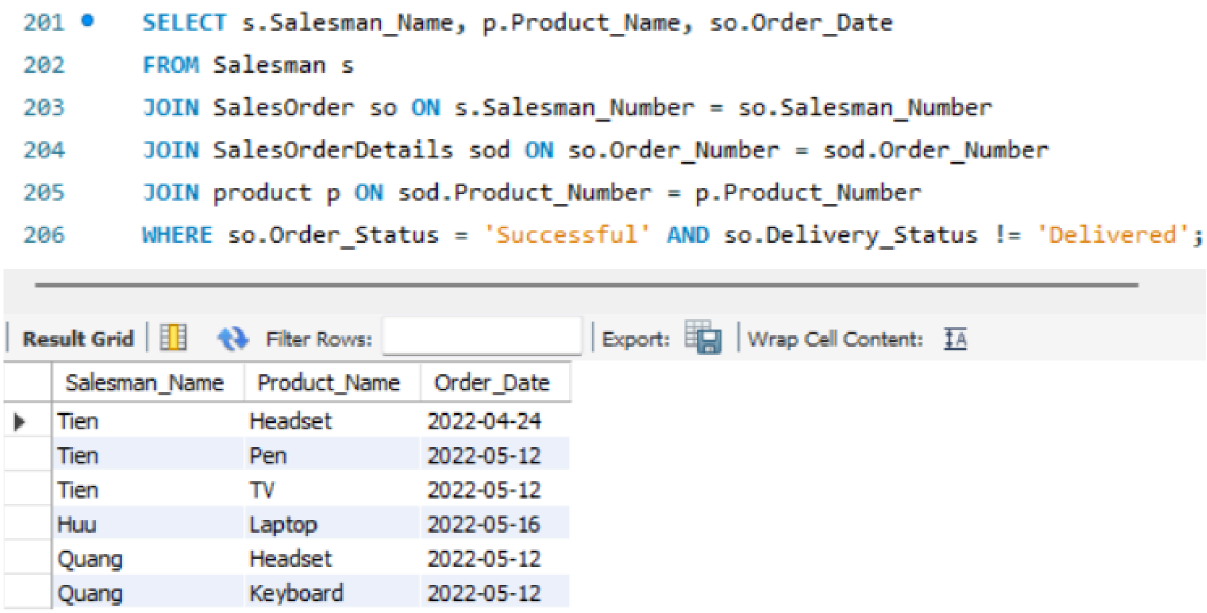


```sql
SELECT p.Product_Name, p.Sell_Price, s.Salesman_Name, so.Delivery_Status, sod.Order_Quantity
FROM product p
JOIN SalesOrderDetails sod ON p.Product_Number = sod.Product_Number
JOIN SalesOrder so ON sod.Order_Number = so.Order_Number
JOIN Salesman s ON so.Salesman_Number = s.Salesman_Number;
```



| Product_Name | Sell_Price | Salesman_Name | Delivery_Status | Order_Quantity |
|---|---|---|---|---|
| Pen | 12.0000 | Huu | Delivered | 6 |
| Mouse | 100.0000 | Huu | Ready to Ship | 10 |
| Laptop | 1500.0000 | Huu | On Way | 20 |
| AC | 400.0000 | Phat | Delivered | 12 |
| TV | 1000.0000 | Khoa | Delivered | 5 |
| Laptop | 1500.0000 | Khoa | Delivered | 4 |
| Keyboard | 120.0000 | Khoa | Delivered | 10 |
| Modem | 250.0000 | Khoa | Delivered | 3 |
| TV | 1000.0000 | Khoa | On Way | 8 |
| Headset | 50.0000 | Khoa | On Way | 15 |
| Laptop | 1500.0000 | Khoa | On Way | 14 |
| Headset | 50.0000 | Tien | On Way | 2 |
| Pen | 12.0000 | Tien | On Way | 3 |
| TV | 1000.0000 | Tien | On Way | 2 |
| Laptop | 1500.0000 | Deb | Ready to Ship | 5 |
| Keyboard | 120.0000 | Deb | Ready to Ship | 6 |
| Modem | 250.0000 | Tin | Ready to Ship | 8 |
| Mouse | 100.0000 | Tin | Delivered | 11 |
| TV | 1000.0000 | Tin | Delivered | 9 |
| Headset | 50.0000 | Quang | On Way | 15 |
| Keyboard | 120.0000 | Quang | On Way | 10 |
| Keyboard | 120.0000 | Hoa | Ready to Ship | 20 |
| AC | 400.0000 | Hoa | Ready to Ship | 5 |

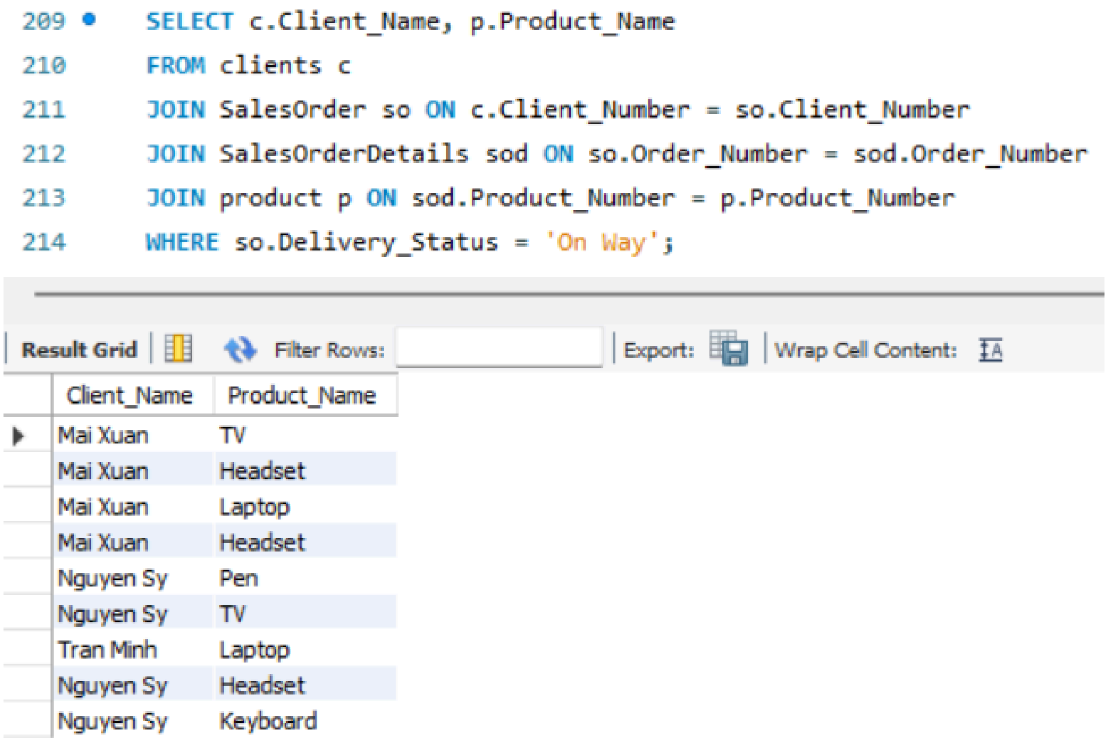
28. Find the names, product names, and order dates of all sales staff whose product order status has been successful but the items have not yet been delivered to the client.



```sql
SELECT s.Salesman_Name, p.Product_Name, so.Order_Date
FROM Salesman s
JOIN SalesOrder so ON s.Salesman_Number = so.Salesman_Number
JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
JOIN product p ON sod.Product_Number = p.Product_Number
WHERE so.Order_Status = 'Successful' AND so.Delivery_Status != 'Delivered';
```

| Salesman_Name | Product_Name | Order_Date |
|---|---|---|
| Tien | Headset | 2022-04-24 |
| Tien | Pen | 2022-05-12 |
| Tien | TV | 2022-05-12 |
| Huu | Laptop | 2022-05-16 |
| Quang | Headset | 2022-05-12 |
| Quang | Keyboard | 2022-05-12 |

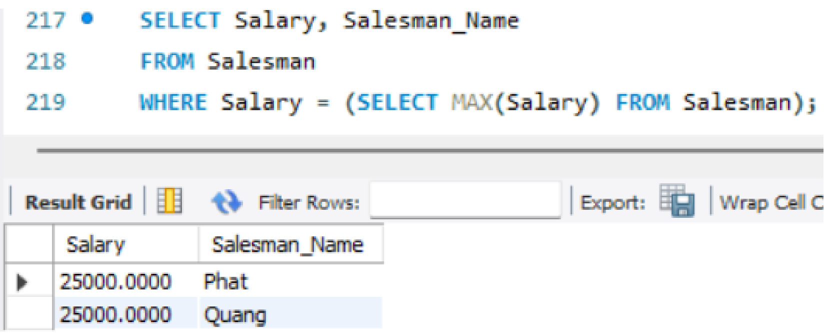
29. Find each clients' product which is on the way.



```sql
SELECT c.Client_Name, p.Product_Name
FROM clients c
JOIN SalesOrder so ON c.Client_Number = so.Client_Number
JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
JOIN product p ON sod.Product_Number = p.Product_Number
WHERE so.Delivery_Status = 'On Way';
```

| Client_Name | Product_Name |
|---|---|
| Mai Xuan | TV |
| Mai Xuan | Headset |
| Mai Xuan | Laptop |
| Mai Xuan | Headset |
| Nguyen Sy | Pen |
| Nguyen Sy | TV |
| Tran Minh | Laptop |
| Nguyen Sy | Headset |
| Nguyen Sy | Keyboard |

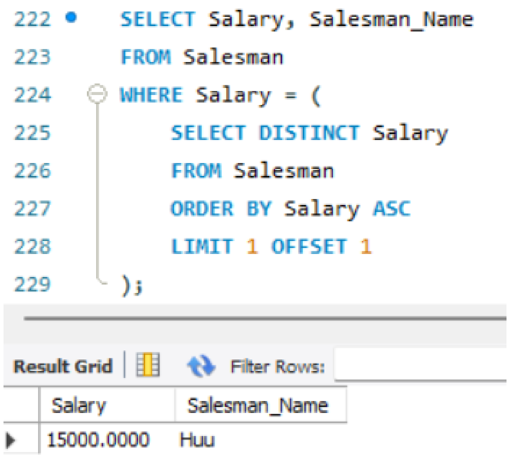
30. Find salary and the salesman's names who is getting the highest salary.



```sql
SELECT Salary, Salesman_Name
FROM Salesman
WHERE Salary = (SELECT MAX(Salary) FROM Salesman);
```

| Salary | Salesman_Name |
|---|---|
| 25000.0000 | Phat |
| 25000.0000 | Quang |

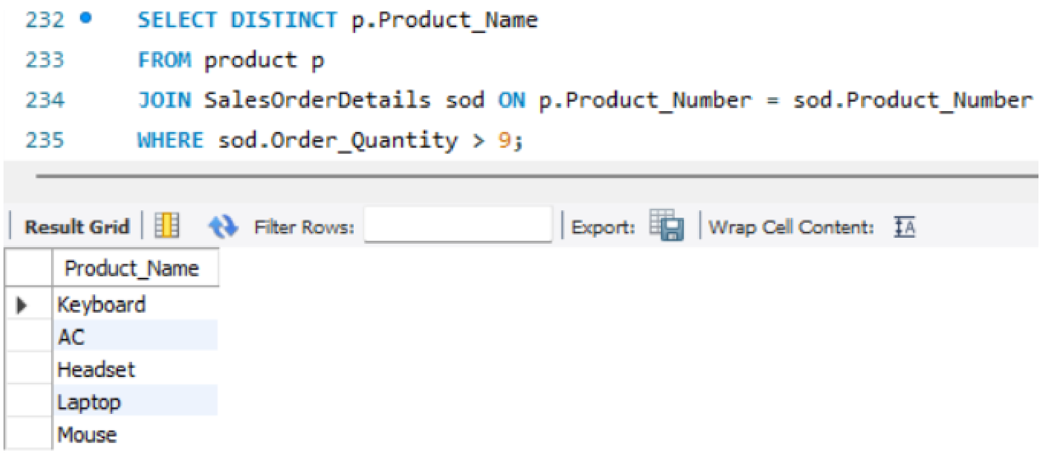
31. Find salary and the salesman's names who is getting the second lowest salary.



```sql
SELECT Salary, Salesman_Name
FROM Salesman
WHERE Salary = (
    SELECT DISTINCT Salary
    FROM Salesman
    ORDER BY Salary ASC
    LIMIT 1 OFFSET 1
);
```

| Salary | Salesman_Name |
|---|---|
| 15000.0000 | Huu |

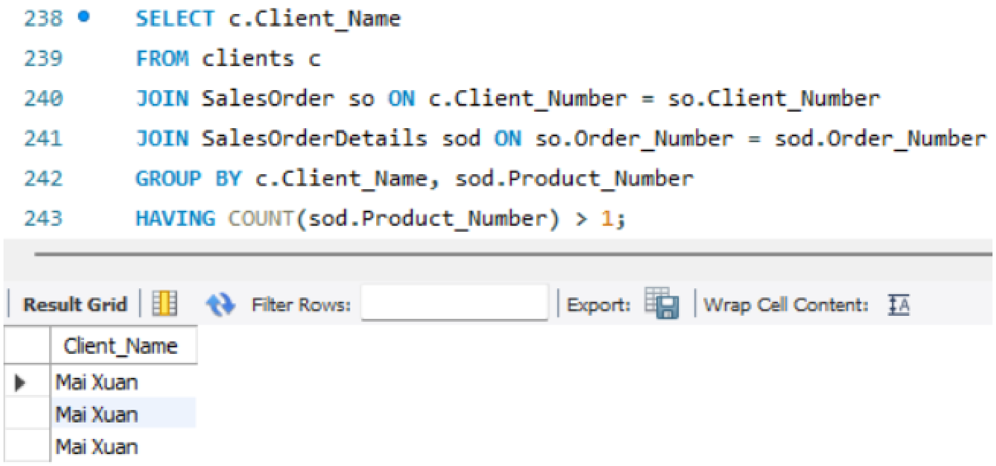
32. Display lists the ProductName in ANY records in the sale Order Details table has Order Quantity more than 9.



```sql
SELECT DISTINCT p.Product_Name
FROM product p
JOIN SalesOrderDetails sod ON p.Product_Number = sod.Product_Number
WHERE sod.Order_Quantity > 9;
```

| Product_Name |
|---|
| Keyboard |
| AC |
| Headset |
| Laptop |
| Mouse |

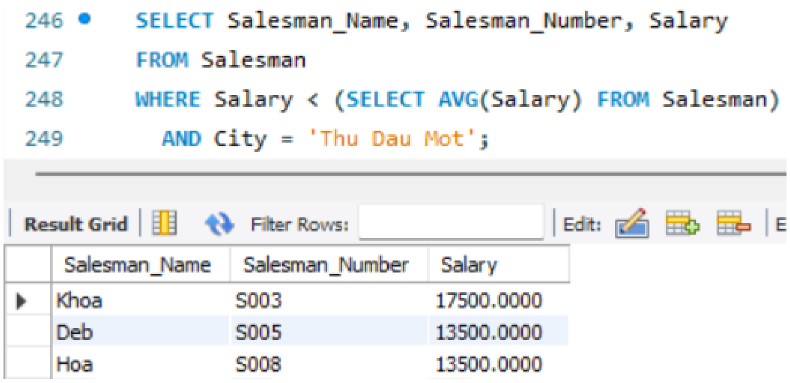
33. Find the name of the customer who ordered the same item multiple times.



```sql
SELECT c.Client_Name
FROM clients c
JOIN SalesOrder so ON c.Client_Number = so.Client_Number
JOIN SalesOrderDetails sod ON so.Order_Number = sod.Order_Number
GROUP BY c.Client_Name, sod.Product_Number
HAVING COUNT(sod.Product_Number) > 1;
```

| Client_Name |
|---|
| Mai Xuan |
| Mai Xuan |
| Mai Xuan |

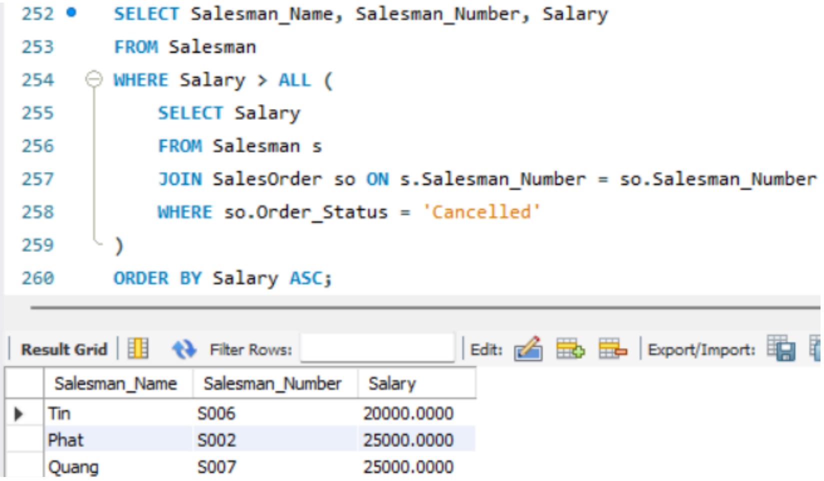
34. Write a query to find the name, number and salary of the salesman who earns less than the average salary and works in any of Thu Dau Mot city.



```sql
SELECT Salesman_Name, Salesman_Number, Salary
FROM Salesman
WHERE Salary < (SELECT AVG(Salary) FROM Salesman)
  AND City = 'Thu Dau Mot';
```

| Salesman_Name | Salesman_Number | Salary |
|---|---|---|
| Khoa | S003 | 17500.0000 |
| Deb | S005 | 13500.0000 |
| Hoa | S008 | 13500.0000 |

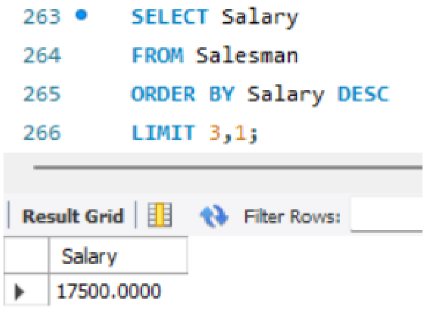
35. Write a query to find the name, number and salary of the salesman who earns a salary that is higher than the salary of all the salesmen have (Order_status = ‘Canceled’). Sort the results of the salary of the lowest to highest.



```sql
SELECT Salesman_Name, Salesman_Number, Salary
FROM Salesman
WHERE Salary > ALL (
    SELECT Salary
    FROM Salesman s
    JOIN SalesOrder so ON s.Salesman_Number = so.Salesman_Number
    WHERE so.Order_Status = 'Cancelled'
)
ORDER BY Salary ASC;
```

| Salesman_Name | Salesman_Number | Salary |
|---|---|---|
| Tin | S006 | 20000.0000 |
| Phat | S002 | 25000.0000 |
| Quang | S007 | 25000.0000 |

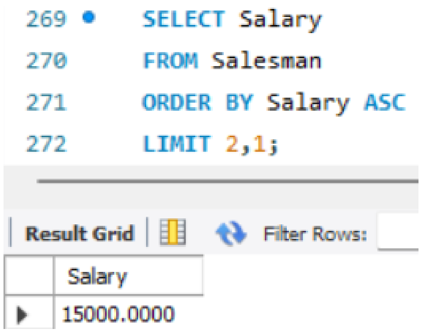
36. Write a query to find the 4th maximum salary on the salesman’s table.



```sql
SELECT Salary
FROM Salesman
ORDER BY Salary DESC
LIMIT 3,1;
```

| Salary |
|---|
| 17500.0000 |

37. Write a query to find the 3th minimum salary in the salesman’s table.



```sql
SELECT Salary
FROM Salesman
ORDER BY Salary ASC
LIMIT 2,1;
```

| Salary |
|---|
| 15000.0000 |