NAME – R.BECINTO ROSHAN

BATCH – 10am TO 11am

1. Database Setup

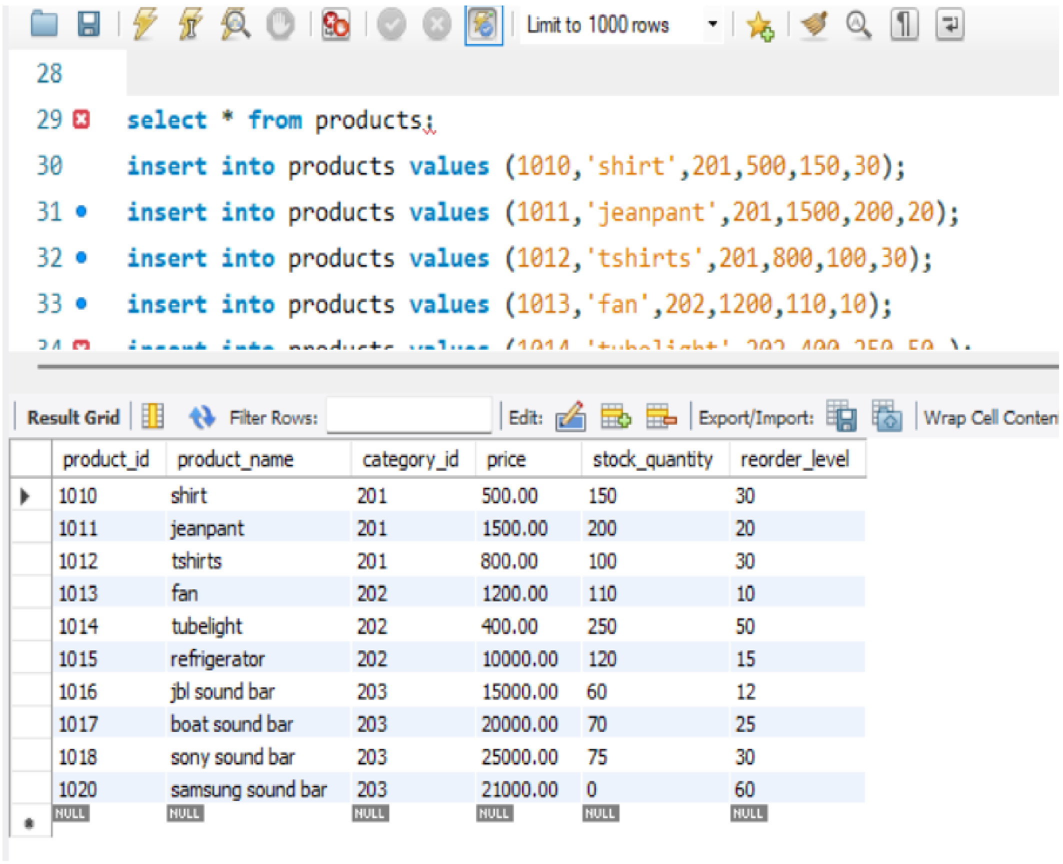
- **Database Name:** InventoryManagement

  Create database inventorymanagement:

  Use inventorymanagement:

Create table products(

```
        product_id INT PRIMARY KEY AUTO_INCREMENT,
        product_name VARCHAR(100) NOT NULL,
        category_id INT NOT NULL,
        price DECIMAL(10, 2) NOT NULL,
        stock_quantity INT NOT NULL,
        reorder_level INT NOT NULL,
            FOREIGN KEY (category_id) REFERENCES Categories(category_id)
            )
```



```

select * from products;
insert into products values (1010,'shirt',201,500,150,30);
insert into products values (1011,'jeanpant',201,1500,200,20);
insert into products values (1012,'tshirts',201,800,100,30);
insert into products values (1013,'fan',202,1200,110,10);
insert into products values (1014,'tubelight',202,400,250,50);
```

product_id	product_name	category_id	price	stock_quantity	reorder_level
1010	shirt	201	500.00	150	30
1011	jeanpant	201	1500.00	200	20
1012	tshirts	201	800.00	100	30
1013	fan	202	1200.00	110	10
1014	tubelight	202	400.00	250	50
1015	refrigerator	202	10000.00	120	15
1016	jbl sound bar	203	15000.00	60	12
1017	boat sound bar	203	20000.00	70	25
1018	sony sound bar	203	25000.00	75	30
1020	samsung sound bar	203	21000.00	0	60
NULL	NULL	NULL	NULL	NULL	NULL

B. Categories Table

Table Name: Categories

Columns:

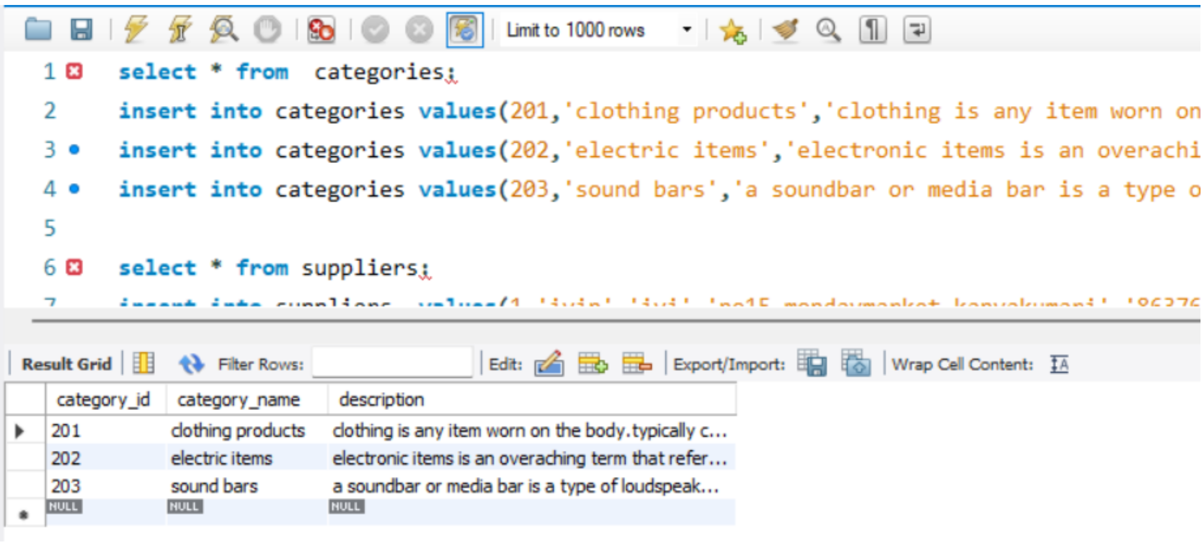
Create table categories (

category_id INT PRIMARY KEY AUTO_INCREMENT,

category_name VARCHAR(100) UNIQUE NOT NULL,

description TEXT

)



```
select * from  categories;
insert into categories values(201,'clothing products','clothing is any item worn on
insert into categories values(202,'electric items','electronic items is an overachi
insert into categories values(203,'sound bars','a soundbar or media bar is a type o

select * from suppliers;
```

	category_id	category_name	description
▶	201	clothing products	clothing is any item worn on the body.typically c...
	202	electric items	electronic items is an overaching term that refer...
	203	sound bars	a soundbar or media bar is a type of loudspeak...
●	NULL	NULL	NULL

C. Suppliers Table

- **Table Name:** Suppliers
- **Columns:**

Create table suppliers (

supplier_id INT PRIMARY KEY AUTO_INCREMENT,

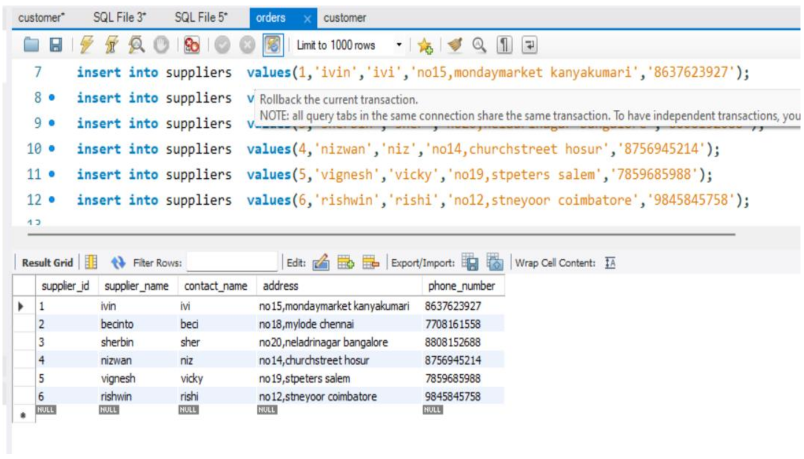
supplier_name VARCHAR(100) NOT NULL,

contact_name VARCHAR(50),

address TEXT,

phone_number VARCHAR(15) UNIQUE

)



```
insert into suppliers  values(1,'ivin','ivi','no15,mondaymarket kanyakumari','8637623927');
insert into suppliers  v
insert into suppliers  v
insert into suppliers  values(4,'nizwan','niz','no14,churchstreet hosur','8756945214');
insert into suppliers  values(5,'vignesh','vicky','no19,stpeters salem','7859685988');
insert into suppliers  values(6,'rishwin','rishi','no12,stneyoor coimbatore','9845845758');
```

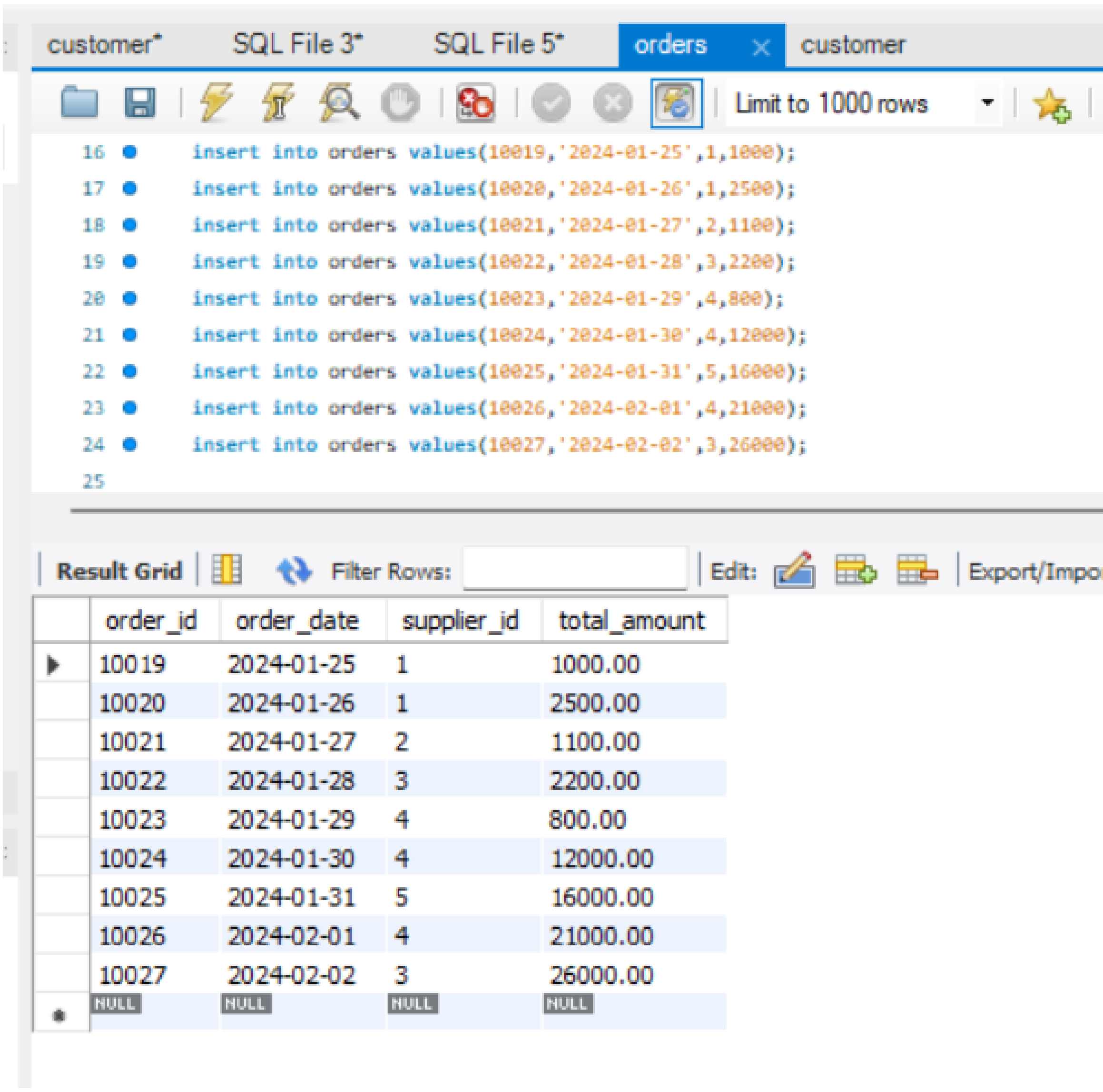
Rollback the current transaction.
NOTE: all query tabs in the same connection share the same transaction. To have independent transactions, you

	supplier_id	supplier_name	contact_name	address	phone_number
▶	1	ivin	ivi	no15,mondaymarket kanyakumari	8637623927
	2	becinto	beci	no18,mylode chennai	7708161558
	3	sherbin	sher	no20,neladrinagar bangalore	8808152688
	4	nizwan	niz	no14,churchstreet hosur	8756945214
	5	vignesh	vicky	no19,stpeters salem	7859685988
	6	rishwin	rishi	no12,stneyoor coimbatore	9845845758
●	NULL	NULL	NULL	NULL	NULL

D. Orders Table

- **Table Name:** Orders
- **Columns:**

```
Create table orders (
order_id INT PRIMARY KEY AUTO_INCREMENT,
order_date DATE NOT NULL,
supplier_id INT NOT NULL,
total_amount DECIMAL(10, 2) NOT NULL,
        FOREIGN KEY (supplier_id) REFERENCES Suppliers(supplier_id)
        )
```



```
insert into orders values(10019,'2024-01-25',1,1000);
insert into orders values(10020,'2024-01-26',1,2500);
insert into orders values(10021,'2024-01-27',2,1100);
insert into orders values(10022,'2024-01-28',3,2200);
insert into orders values(10023,'2024-01-29',4,800);
insert into orders values(10024,'2024-01-30',4,12000);
insert into orders values(10025,'2024-01-31',5,16000);
insert into orders values(10026,'2024-02-01',4,21000);
insert into orders values(10027,'2024-02-02',3,26000);
```

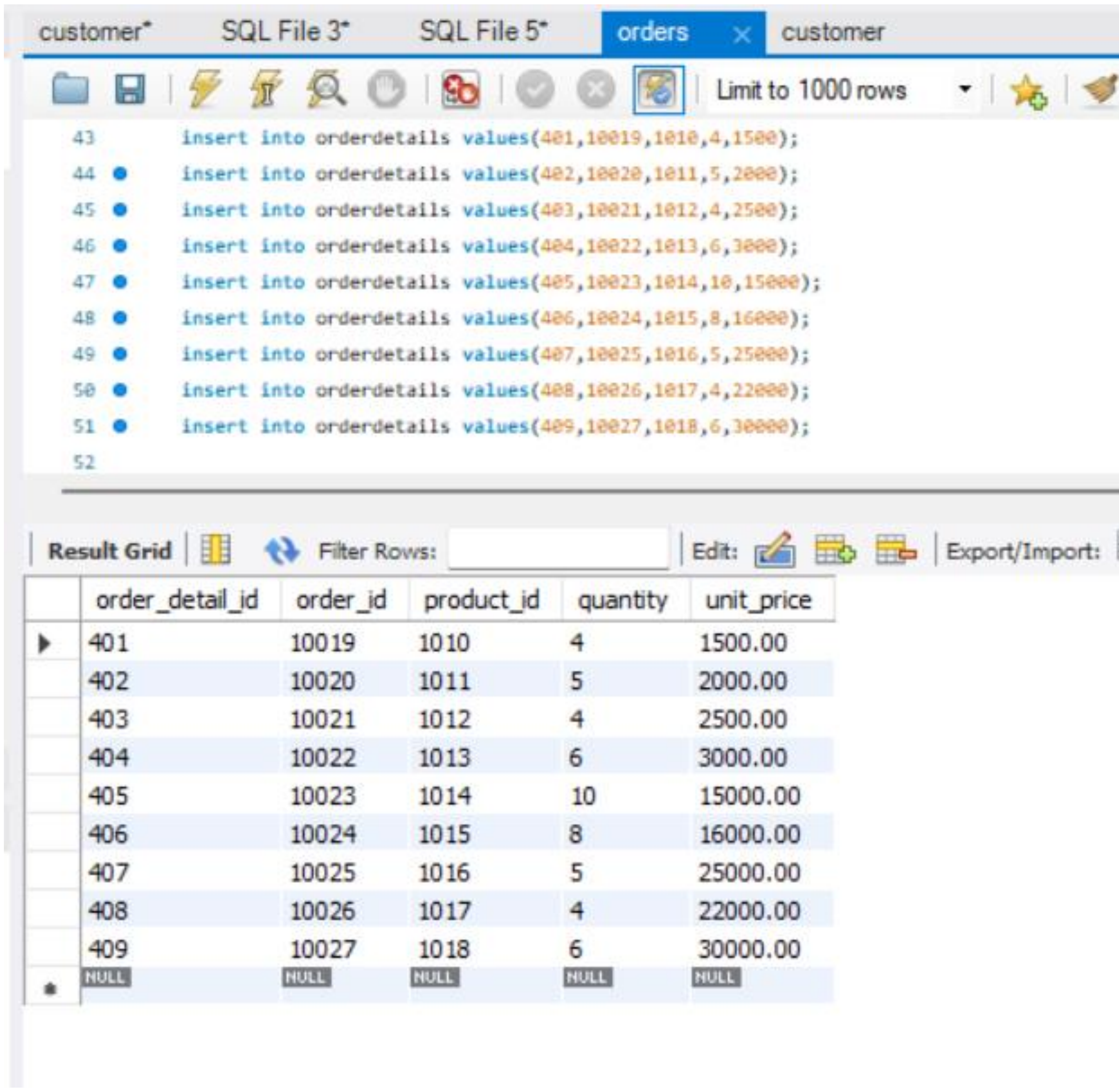
order_id	order_date	supplier_id	total_amount
10019	2024-01-25	1	1000.00
10020	2024-01-26	1	2500.00
10021	2024-01-27	2	1100.00
10022	2024-01-28	3	2200.00
10023	2024-01-29	4	800.00
10024	2024-01-30	4	12000.00
10025	2024-01-31	5	16000.00
10026	2024-02-01	4	21000.00
10027	2024-02-02	3	26000.00
NULL	NULL	NULL	NULL

E. OrderDetails Table

- **Table Name:** OrderDetails

Columns:

```
Create table orderdetails (
order_detail_id INT PRIMARY KEY AUTO_INCREMENT,

order_id INT NOT NULL,

product_id INT NOT NULL,

quantity INT NOT NULL,

unit_price DECIMAL(10, 2) NOT NULL,

        FOREIGN KEY (order_id) REFERENCES Orders(order_id),

        FOREIGN KEY (product_id) REFERENCES Products(product_id));
```



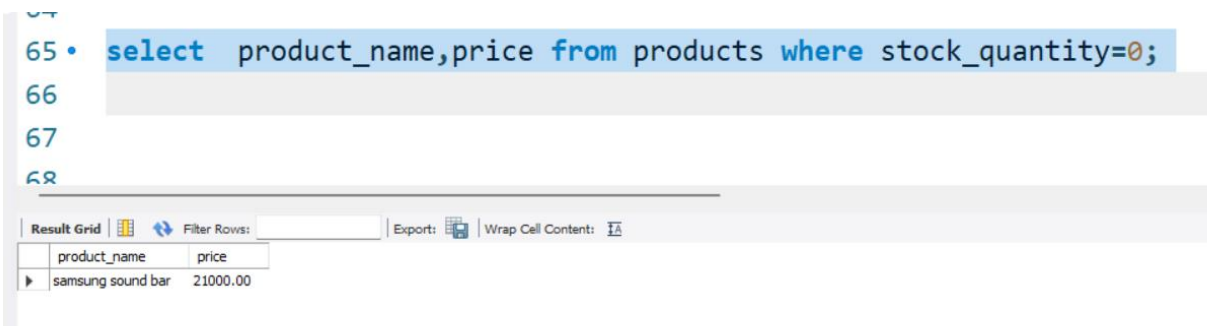
```
insert into orderdetails values(401,10019,1010,4,1500);
insert into orderdetails values(402,10020,1011,5,2000);
insert into orderdetails values(403,10021,1012,4,2500);
insert into orderdetails values(404,10022,1013,6,3000);
insert into orderdetails values(405,10023,1014,10,15000);
insert into orderdetails values(406,10024,1015,8,16000);
insert into orderdetails values(407,10025,1016,5,25000);
insert into orderdetails values(408,10026,1017,4,22000);
insert into orderdetails values(409,10027,1018,6,30000);
```

order_detail_id	order_id	product_id	quantity	unit_price
401	10019	1010	4	1500.00
402	10020	1011	5	2000.00
403	10021	1012	4	2500.00
404	10022	1013	6	3000.00
405	10023	1014	10	15000.00
406	10024	1015	8	16000.00
407	10025	1016	5	25000.00
408	10026	1017	4	22000.00
409	10027	1018	6	30000.00
NULL	NULL	NULL	NULL	NULL

3. SQL Queries

After creating the tables, students should answer the following questions using SQL queries:

1. Retrieve the names and prices of all products that are currently out of stock.

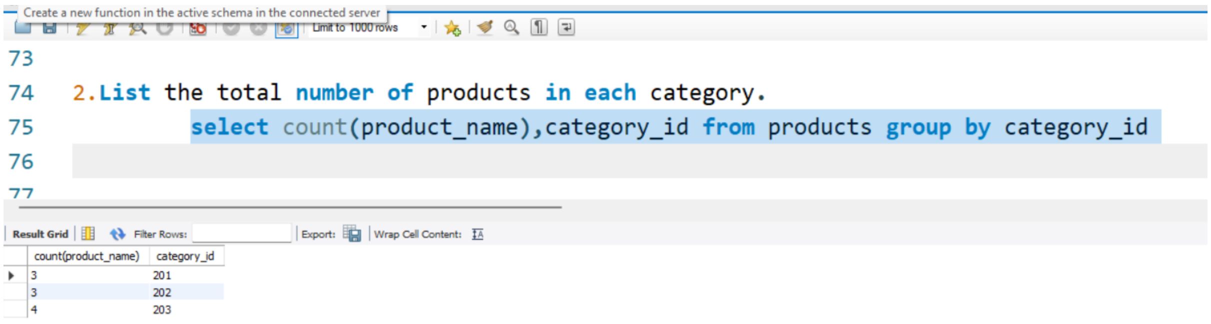


```
select  product_name,price from products where stock_quantity=0;
```

product_name	price
samsung sound bar	21000.00

2.List the total number of products in each category.

select count(product_name),category_id from products group by category_id



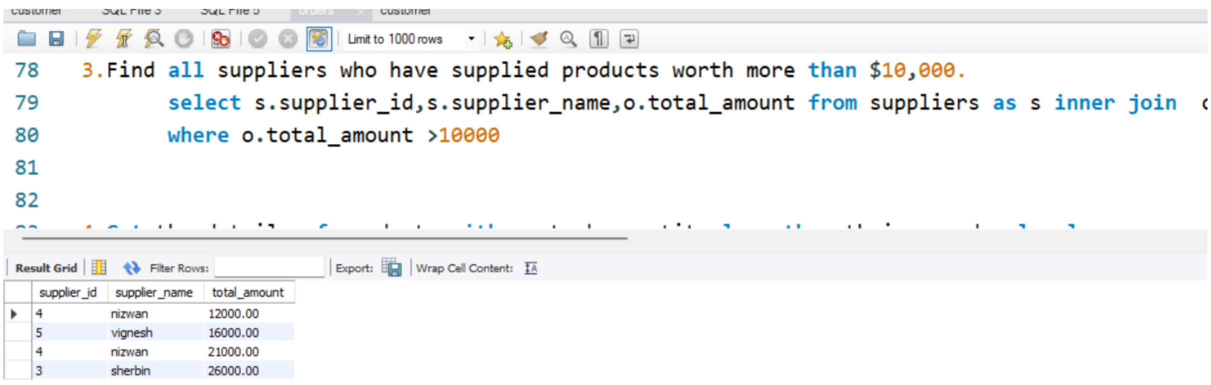
```
2.List the total number of products in each category.
        select count(product_name),category_id from products group by category_id
```

count(product_name)	category_id
3	201
3	202
4	203

3.Find all suppliers who have supplied products worth more than $10,000.

select s.supplier_id,s.supplier_name,o.total_amount from suppliers as s inner join orders as o on s.supplier_id = o.supplier_id

where o.total_amount >10000;



```
3.Find all suppliers who have supplied products worth more than $10,000.
      select s.supplier_id,s.supplier_name,o.total_amount from suppliers as s inner join  o
      where o.total_amount >10000
```

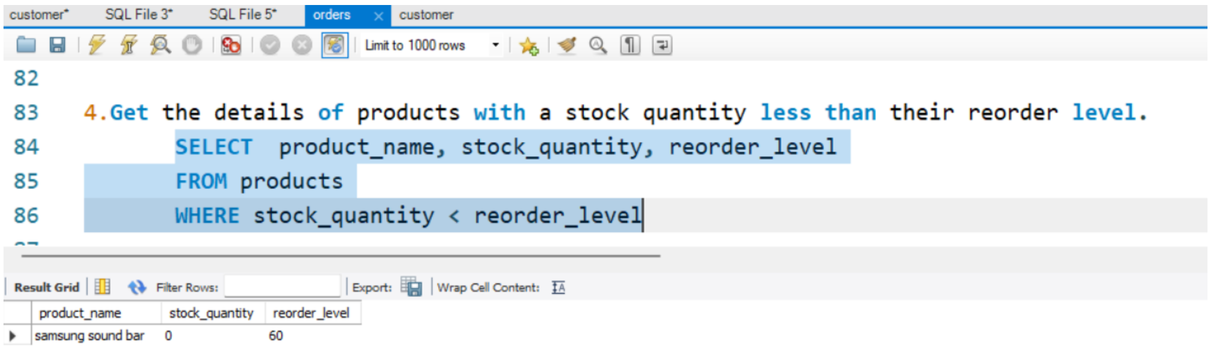
supplier_id	supplier_name	total_amount
4	nizwan	12000.00
5	vignesh	16000.00
4	nizwan	21000.00
3	sherbin	26000.00

4.Get the details of products with a stock quantity less than their reorder level.

SELECT product_name, stock_quantity, reorder_level

FROM products

WHERE stock_quantity < reorder_level;

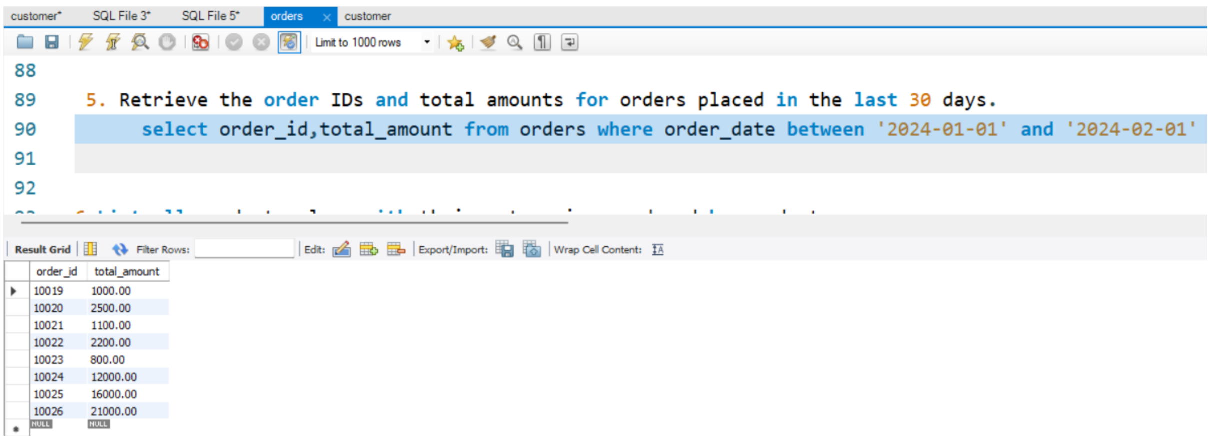


```
4.Get the details of products with a stock quantity less than their reorder level.
        SELECT  product_name, stock_quantity, reorder_level
        FROM products
        WHERE stock_quantity < reorder_level
```

```
product_name       stock_quantity   reorder_level
samsung sound bar  0                60
```

5 .Retrieve the order IDs and total amounts for orders placed in the last 30 days.

select order_id,total_amount from orders where order_date between '2024-01-01' and '2024-02-01';

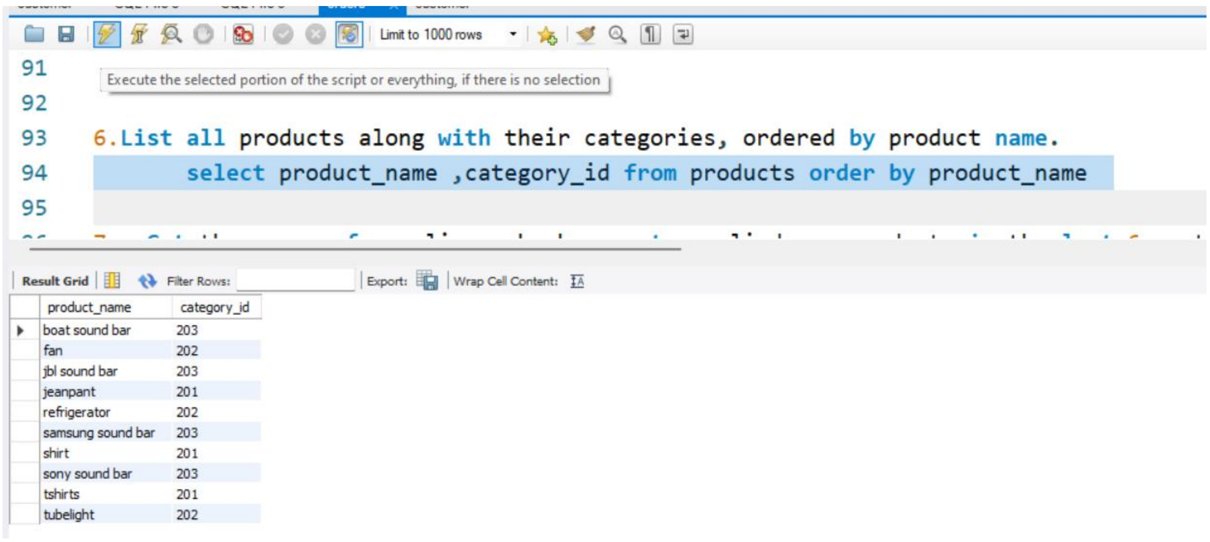


```
5. Retrieve the order IDs and total amounts for orders placed in the last 30 days.
    select order_id,total_amount from orders where order_date between '2024-01-01' and '2024-02-01'
```

```
order_id  total_amount
10019     1000.00
10020     2500.00
10021     1100.00
10022     2200.00
10023     800.00
10024     12000.00
10025     16000.00
10026     21000.00
NULL      NULL
```

6.List all products along with their categories, ordered by product name.

select product_name ,category_id from products order by product_name ;



Execute the selected portion of the script or everything, if there is no selection

```
6.List all products along with their categories, ordered by product name.
    select product_name ,category_id from products order by product_name
```

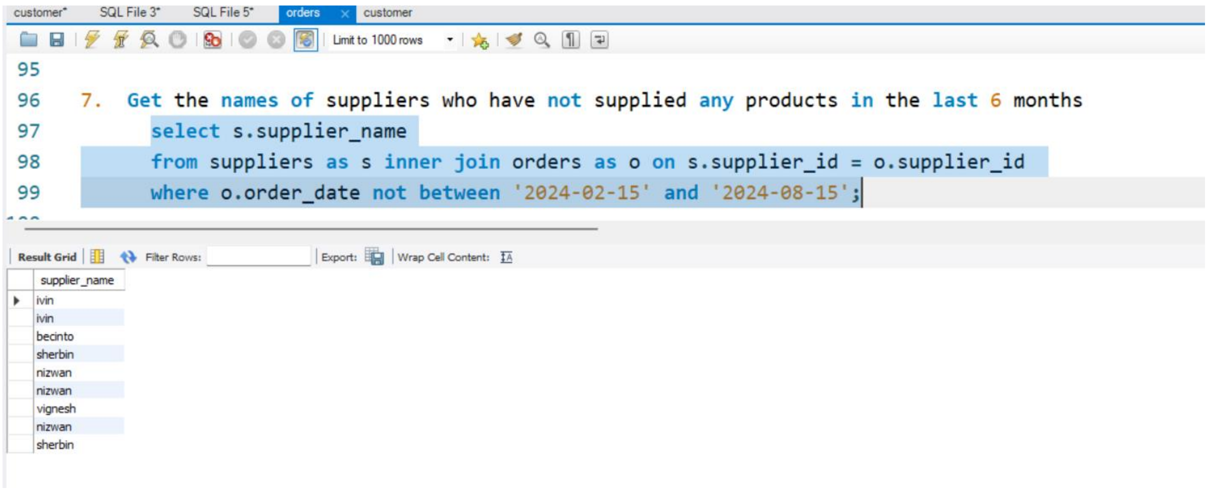
```
product_name        category_id
boat sound bar      203
fan                 202
jbl sound bar       203
jeanpant            201
refrigerator        202
samsung sound bar   203
shirt               201
sony sound bar      203
tshirts             201
tubelight           202
```

7. Get the names of suppliers who have not supplied any products in the last 6 months

select s.supplier_name

from suppliers as s inner join orders as o on s.supplier_id = o.supplier_id

where o.order_date not between '2024-02-15' and '2024-08-15';



```
7.  Get the names of suppliers who have not supplied any products in the last 6 months
    select s.supplier_name
    from suppliers as s inner join orders as o on s.supplier_id = o.supplier_id
    where o.order_date not between '2024-02-15' and '2024-08-15';
```

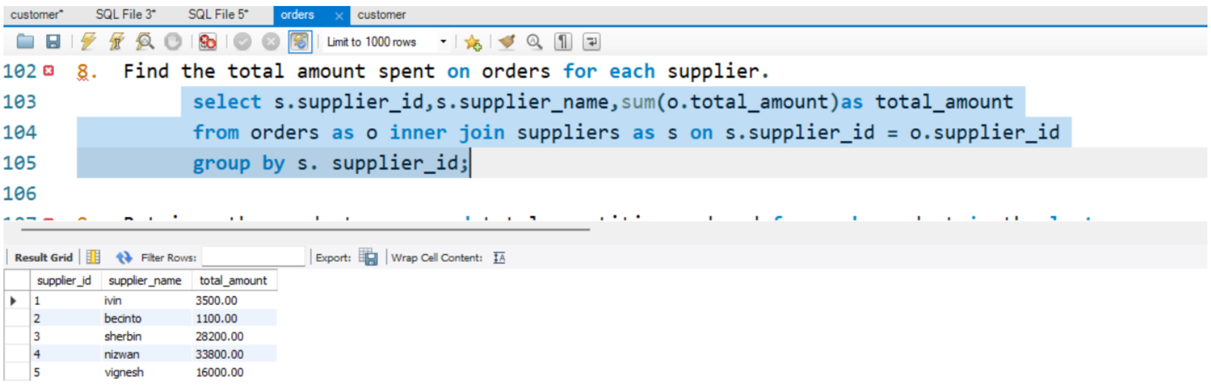
```
supplier_name
ivin
ivin
becinto
sherbin
nizwan
nizwan
vignesh
nizwan
sherbin
```

8.Find the total amount spent on orders for each supplier.

select s.supplier_id,s.supplier_name,sum(o.total_amount)as total_amount

from orders as o inner join suppliers as s on s.supplier_id = o.supplier_id

group by s. supplier_id;



```
8.  Find the total amount spent on orders for each supplier.
            select s.supplier_id,s.supplier_name,sum(o.total_amount)as total_amount
            from orders as o inner join suppliers as s on s.supplier_id = o.supplier_id
            group by s. supplier_id;
```

supplier_id	supplier_name	total_amount
1	ivin	3500.00
2	becinto	1100.00
3	sherbin	28200.00
4	nizwan	33800.00
5	vignesh	16000.00

9.Retrieve the product names and total quantities ordered for each product in the last year.

SELECT P.product_name , SUM(od.Quantity) AS TotalQuantityOrdered

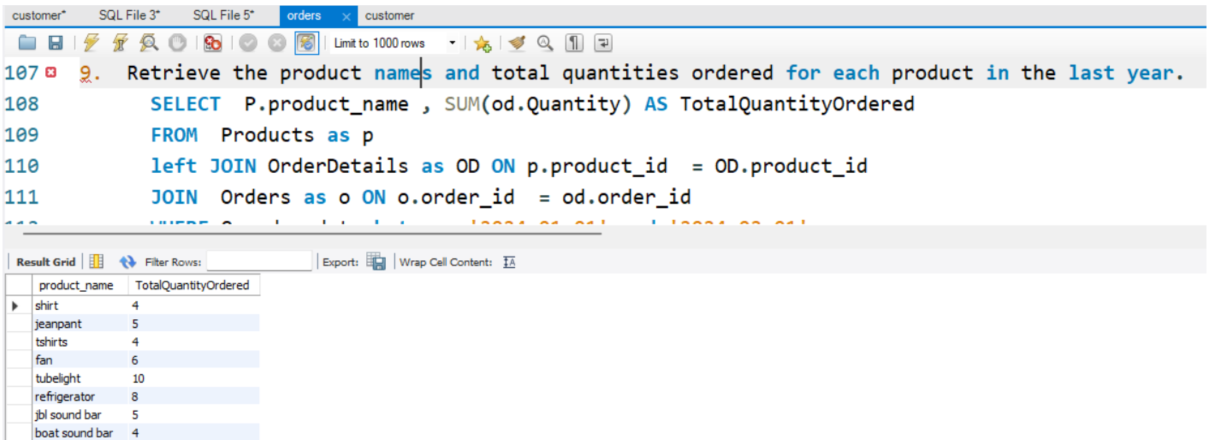
FROM Products as p

left JOIN OrderDetails as OD ON p.product_id = OD.product_id

JOIN Orders as o ON o.order_id = od.order_id

WHERE O.order_date between '2024-01-01' and '2024-02-01'

GROUP BY P.product_name;



```
9.  Retrieve the product names and total quantities ordered for each product in the last year.
        SELECT  P.product_name , SUM(od.Quantity) AS TotalQuantityOrdered
        FROM  Products as p
        left JOIN OrderDetails as OD ON p.product_id  = OD.product_id
        JOIN  Orders as o ON o.order_id  = od.order_id
```

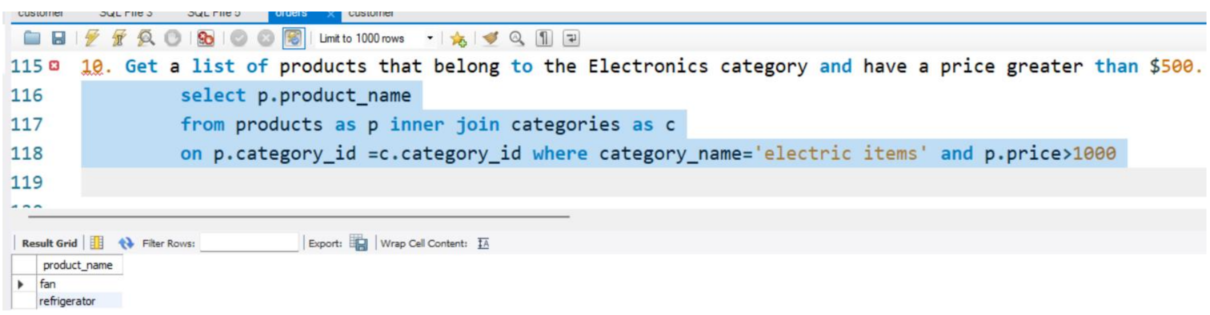
product_name	TotalQuantityOrdered
shirt	4
jeanpant	5
tshirts	4
fan	6
tubelight	10
refrigerator	8
jbl sound bar	5
boat sound bar	4

10.Get a list of products that belong to the Electronics category and have a price greater than $500.

select p.product_name

from products as p inner join categories as c

on p.category_id =c.category_id where category_name='electric items' and p.price>1000;



```
10. Get a list of products that belong to the Electronics category and have a price greater than $500.
        select p.product_name
        from products as p inner join categories as c
        on p.category_id =c.category_id where category_name='electric items' and p.price>1000
```

product_name
fan
refrigerator