Goal of project:

To create Database for Online Shopping system like Amazon.

Project Description:

Consider the following requirement for an online shopping system like Amazon:

- The shopping system starts when the supplier lists his products in the shopping website, the system allows multiple suppliers to sell their products on the website, each supplier is defined by his ID, Name, and Address.
- The website is divided into several categories as: electronics, fashion, home ….etc., each category has an ID and name.
- In each category there are many products, each is identified by an ID, price, and name.
- A customer visiting the website must register his ID, name, contact address and phone number.
- A customer can add several products in one order, each order is giving an ID and a date.
- After placing the order, the customer must be able to pay for his order, each payment is registers in the system by its ID, amount, method, and date.
- After a success payment, the system prints an invoice for the customer, the invoices in the system defined by the invoice ID and contains the (CustomerID, OrderID, ProductID, PaymentID)
- A customer in the system must be able to track his order delivery progress, each delivery has a tracking number, courier name, and a date.

ER diagram:

Data Modeling Tool:

Modeling Tool is:

draw.io

reference is:

http://draw.io

Assumptions:

For products entity: we assume Prod_ID as an attribute to be the primary key.

For supplier entity: we assume supp_ID as an attribute to be the primary key.

For products categories entity: we assume Cate_ID as an attribute to be the primary key.

For order entity: we assume Ord_ID as an attribute to be the primary key.

For Customer entity: we assume Cust_ID as an attribute to be the primary key.

For Invoice entity: we assume Inv_ID as an attribute to be the primary key.

Description: ER Data Model component:

Entities:

products (strong entity):

This entity identifies item in the online shopping, it contains these attributes: Pord_ID Prod_price, Prod_name.

supplier (strong entity):

This entity identifies supplier in the online shopping, it contains these attributes: Supp_ID, Supp_Contact_Address, Supp_Name.

Product Categories (strong entity):

This entity defines product Category, it contains these attributes: Cate_Name, Cate_ID.

Order (strong entity):

This entity identifies the orders in the online shopping, it contain this attribute: Ord_ID.

Payment (strong entity):

This entity accept/reject orders, it contains these attributes: Pay_ID, Pay_Amount, Pay_Method, Pay_Date.

Orders Delivery progress (strong entity):

This entity identifies orders while shipping, it contains these attributes: ODP_Traking_Number, ODP_Courier_Number, ODP_Date.

Customer (strong entity):

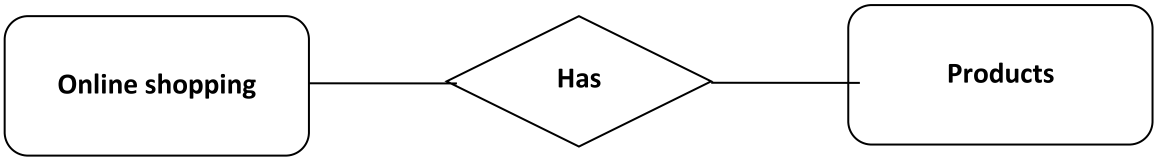
This entity identifies Customer in the online shopping, it contains these attributes: Cust_Contact_Address, Cust_ID, Cust_Phone_Number, Cust_Name.

invoice (Strong entity):

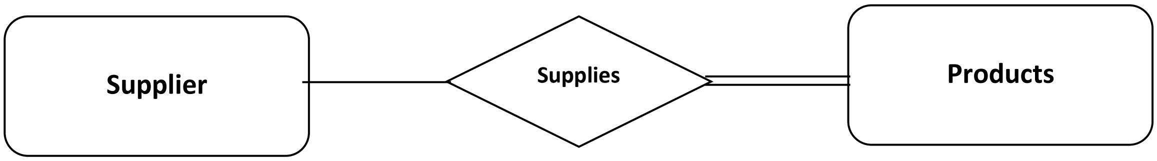
This entity Print orders information, it contains these attributes: Inv_ID, Cust_ID, Ord_ID, Pay_ID, Prod_ID.

Relations:

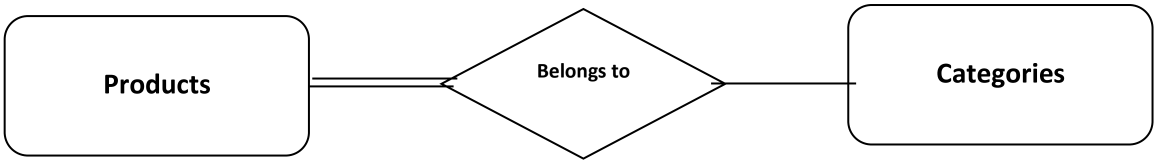
The online shopping should have products to start.



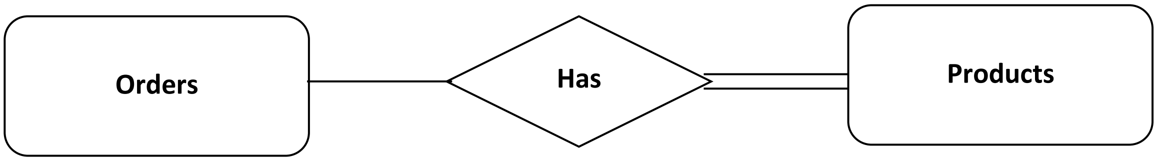
Suppliers supplies the online shopping with products.



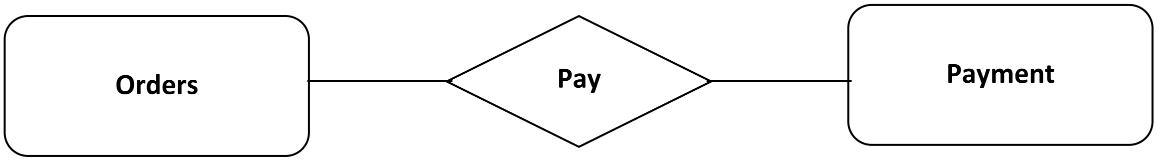
Each product should belong to a category.



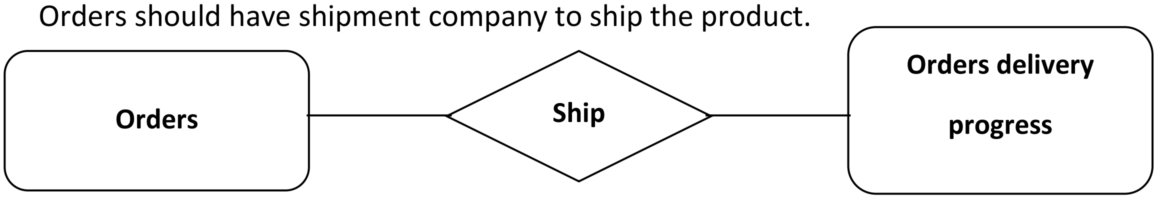
Orders should have products to order from.



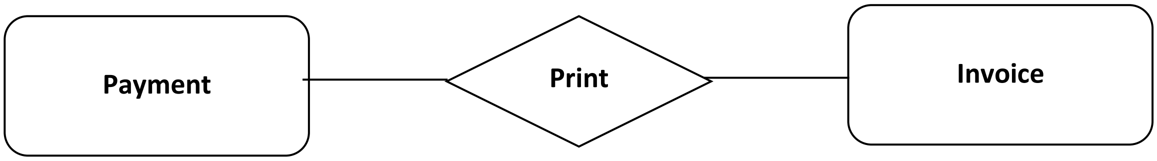
Orders should be able to pay the payment for the order.



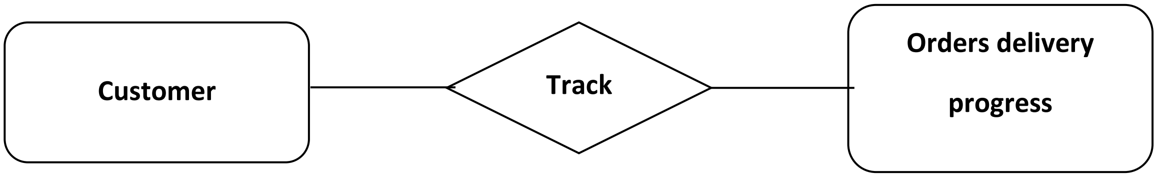
Orders should have shipment company to ship the product.



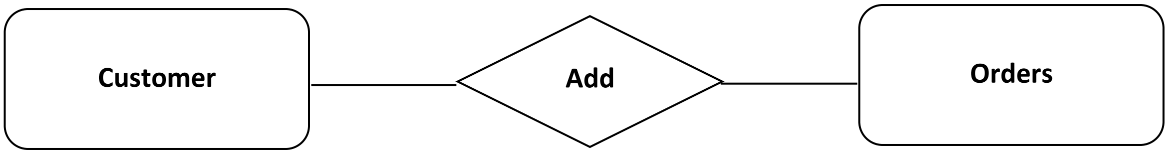
Payment should have invoice contain all the important information about the order.



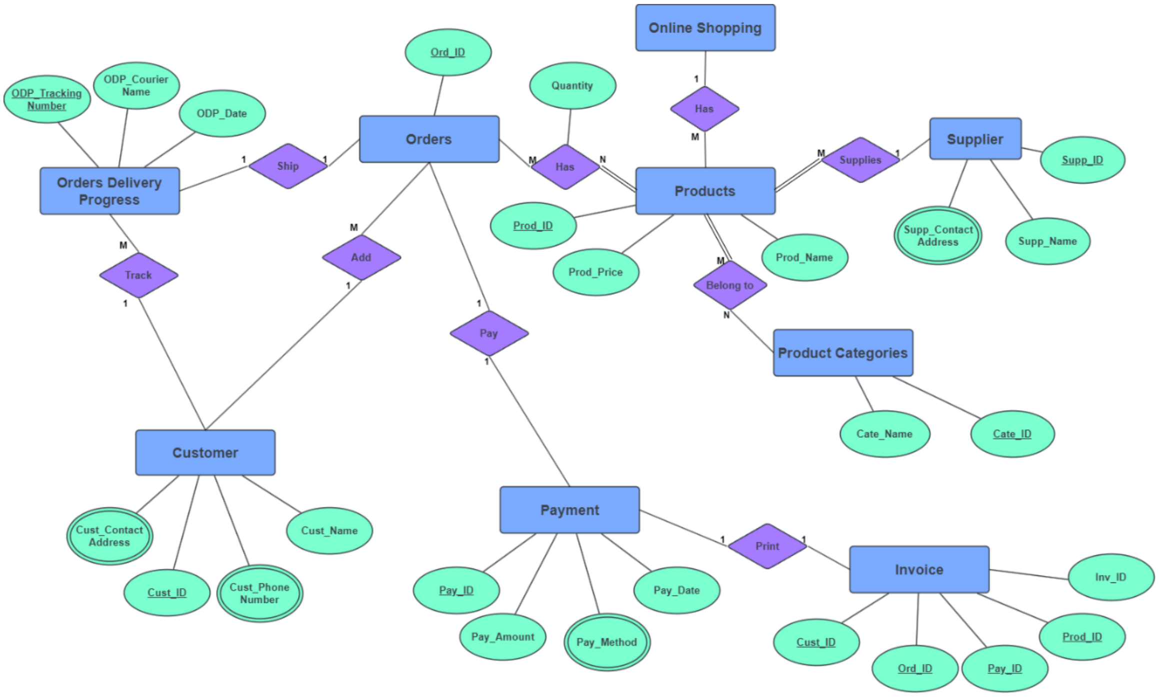
Customer should be able to track his orders.



Customer should be able to place order.



Conceptual Data Model(ER):



Relational Database Schema:

Assumptions

We add cate_ID on the product to know what a product is.

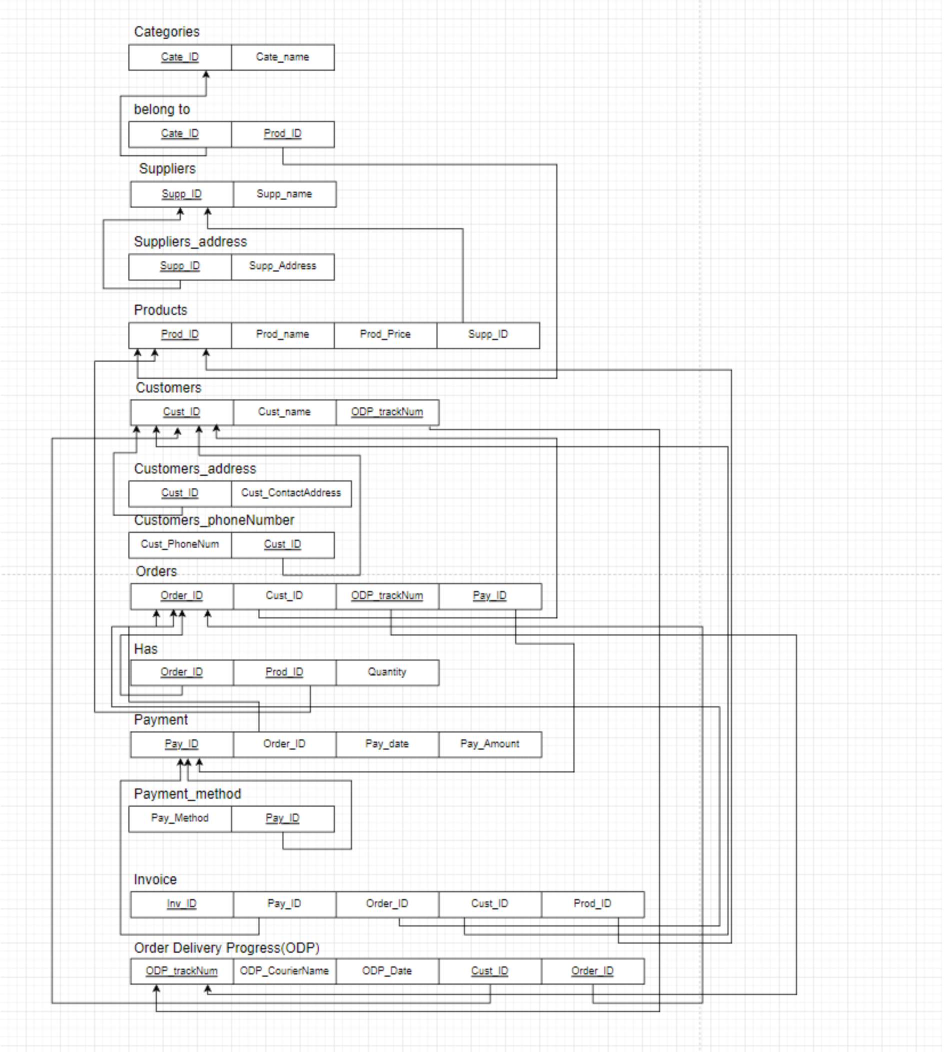
We add cust_ID to the order to know who create this order.

We add prod_ID to know what products have been added.

We add ODP_TracNum to the customer so a customer can track his order.

We add Inv_ID to the order delivery progress to know the invoice information.

Relational Database Schema:



Categories

Cate_ID	Cate_name

belong to

Cate_ID	Prod_ID

Suppliers

Supp_ID	Supp_name

Suppliers_address

Supp_ID	Supp_Address

Products

Prod_ID	Prod_name	Prod_Price	Supp_ID

Customers

Cust_ID	Cust_name	ODP_trackNum

Customers_address

Cust_ID	Cust_ContactAddress

Customers_phoneNumber

Cust_PhoneNum	Cust_ID

Orders

Order_ID	Cust_ID	ODP_trackNum	Pay_ID

Has

Order_ID	Prod_ID	Quantity

Payment

Pay_ID	Order_ID	Pay_date	Pay_Amount

Payment_method

Pay_Method	Pay_ID

Invoice

Inv_ID	Pay_ID	Order_ID	Cust_ID	Prod_ID

Order Delivery Progress(ODP)

ODP_trackNum	ODP_CourierName	ODP_Date	Cust_ID	Order_ID

Schema Definitions and Query Implementations:

Table Creation:

14 tables has been created for the database : (Categories, belong to, Suppliers, Suppliers_address, Products, Customers, Customers_address, Customers_PhoneNumber, Orders, Has, Payment, Payment_method, ODP):

```
create table Categories (
          Cate_ID int primary key auto_increment,
  Cate_name varchar(50) not null);


create table Suppliers (
          Supp_ID int primary key auto_increment,
          Supp_name varchar(50) not null);


create table Suppliers_Address (
          Supp_ID int,
  Supp_Address varchar(100) not null,
  foreign key(Supp_ID) references suppliers(Supp_ID));


create table Products (
          Prod_ID int primary key auto_increment,
  Prod_name varchar(50) not null,
  Prod_price decimal(7,2) not null,
  Cate_ID int,
  Supp_ID int,
  foreign key(Cate_ID) references Categories(Cate_ID),
  foreign key(Supp_ID) references suppliers(Supp_ID));

create table belong_to (
          Cate_ID int,
  Prod_ID int,
  foreign key(Cate_ID) references categories(Cate_ID),
  foreign key(Prod_ID) references Products(Prod_ID));

create table Customers (
          Cust_ID int primary key auto_increment,
  Cust_name varchar(50) not null);

create table Customers_adress (
          Cust_ID int,
  Cust_Address varchar(100) not null,
  foreign key(Cust_ID) references Customers(Cust_ID));

create table Customers_Phone_Number (
          Cust_ID int,
  Cust_Phone_Number varchar(15) not null,
  foreign key(Cust_ID) references Customers(Cust_ID));
```

```
create table Orders (
        Order_ID int primary key auto_increment,
  Cust_ID int,
  foreign key(Cust_ID) references Customers(Cust_ID));

create table Has (
        Order_ID int,
  Prod_ID int,
  Quantity int default 1,
  foreign key(Order_ID) references Orders(Order_ID),
  foreign key(Prod_ID) references Products(Prod_ID));

create table Payment (
        Pay_ID int primary key auto_increment,
  Order_ID int,
  Pay_date Datetime default(current_date),
  Pay_amount int not null,
  foreign key(Order_ID) references Orders(Order_ID));
create table Payment_method (
        Pay_ID int,
  Payment_method varchar(20) not null,
  foreign key(Pay_ID) references payment(Pay_ID));
```

```
create table Invoice (
          invoice_ID int primary key auto_increment,
   Pay_ID int,
   Order_ID int,
   Cust_ID int,
   Prod_ID int,
   foreign key(Pay_ID) references payment(Pay_ID),
   foreign key(Order_ID) references Orders(Order_ID),
   foreign key(Cust_ID) references Customers(Cust_ID),
   foreign key(Prod_ID) references Products(Prod_ID));

create table ODP (
          ODP_Track_Number int primary key auto_increment,
   ODP_Courier_Name varchar(50),
   ODP_Date datetime default(current_date),
   Cust_ID int,
   Order_ID int,
   foreign key(Cust_ID) references Customers(Cust_ID),
   foreign key(Order_ID) references Orders(Order_ID));

alter table orders
ADD ODP_Track_Number int;
```

```
alter table orders
add constraint ODP_Track_Number
FOREIGN KEY (ODP_Track_Number) REFERENCES ODP(ODP_Track_Number);

alter table orders
ADD Pay_ID int;

alter table orders
add constraint Pay_ID
FOREIGN KEY (Pay_ID) REFERENCES payment(Pay_ID);

alter table Customers
add ODP_Track_Number int;

alter table Customers
add constraint FK_Cust_ODP_Track_Number
FOREIGN KEY (ODP_Track_Number) REFERENCES ODP(ODP_Track_Number);
```

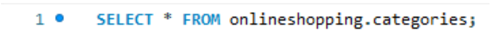
Database State:

```
insert into Categories(Cate_name)
values (' Electronics '),
       ('home'),
       (' computer ');


insert into Suppliers(Supp_name)
value ('ahmed'),
     ('faris'),
     ('ziad'),
     ('khaled'),
     ('peter'),
     ('tamem');


insert into Suppliers_Address(Supp_Address ,Supp_ID )
value (' Al-Qassim  ', 1 ),
     (' al-kharj ',2 ),
     (' hail', 3 ),
     (' al-madinah ', 4 ),
     ('al-namas', 5),
     (' shaqra', 6 );


  insert into Products (Prod_name , Prod_price , Cate_ID , Supp_ID )
   value ('vacuum cleaner', 1999.99,1, 1 ),
        ('boiler', 119.99 , 1 , 2 ),
        ('chair' , 299.99 , 2 , 3),
        ('table', 399.99 , 2 , 4 ),
      ('RTX 3080' , 2999.99 ,  3 , 5 ),
      ('RAM 16GB' , 199.99 , 3 , 6 );

insert into belong_to (Cate_ID ,Prod_ID )
value ( 1 , 1),
     (1 , 2 ),
     (2 ,3 ),
     (2 , 4 ),
     (3 , 5 ),
     (3 , 6 );

insert into Customers (Cust_name , ODP_Track_Number )
value('sraf'),
('Faisal'),
('hanin'),
('Ibrahim'),
('raghed'),
('meske');

insert into Customers_adress ( Cust_Address , Cust_ID  )
value ( 'Riyadh' , 1 ),
( 'Dammam' , 2),
( 'Makkah' , 3 ),
('Jeddah' , 4 ),
('haremila' , 5 ),
('Abha' , 6);
```

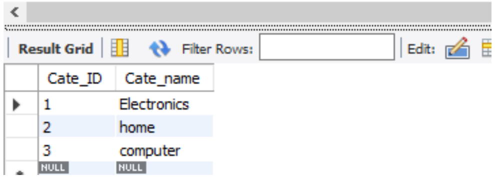
```
insert into Customers_Phone_Number (Cust_ID , Cust_Phone_Number )
value (1 , “966 548 390 231”),
     (2 , “966 523 849 056”),
     (3 , “ 966 564 489 930”),
     (4 , “966 574 483 834”),
     (5 , “966 563 643 743”),
     (6 , “966 532 732 873”);

insert into Orders ( Cust_ID )
value (1),
     (2),
     (3),
     (4),
     (5),
     (6);

 insert into Has(Order_ID , Prod_ID, Quantity )
 value (1,1,1),
      (2,2,3),
      (3,3,1),
      (4,4,1),
      (5,5,10),
      (6,6,2);


 insert into  Payment (Order_ID , Pay_amount  )
value (1 ,1999.99),
     (2,359.97),
     (3,299.99),
     (4,399.99),
     (5,29999.9),
     (6,399.8);

 insert into Payment_method (Pay_ID ,Payment_method )
 value (1, 'cash' ),
 (2 , 'mada'),
 (3 , 'visa'),
 (4 , 'cash'),
 (5 , 'mada'),
 (6 , 'visa');

insert into Invoice (Pay_ID ,Order_ID ,Cust_ID ,Prod_ID )
value (1 , 1 , 1 , 1),
(2,2,2,2),
(3,3,3,3),
(4,4,4,4),
(5,5,5,5),
(6,6,6,6);

insert into ODP( ODP_Courier_Name , Cust_ID , Order_ID , )
value ('hamed' , 1,1),
('saeed',2,2),
('mesfer',3,3),
('Abdullah',4,4),
 ('Abdulrahman',5,5),
 ('saleh',6,6);

 UPDATE onlineshopping.orders
SET ODP_Track_Number = 1, Pay_ID = 1
WHERE (Order_ID = 1);

 UPDATE onlineshopping.orders
 SET ODP_Track_Number = 2, Pay_ID = 2
 WHERE (Order_ID = 2);

 UPDATE onlineshopping.orders
 SET ODP_Track_Number = 3, Pay_ID = 3
 WHERE (Order_ID = 3);

 UPDATE onlineshopping.orders
 SET ODP_Track_Number = 4 , Pay_ID = 4
 WHERE (Order_ID = 4);

 UPDATE onlineshopping.orders
 SET ODP_Track_Number = 5 , Pay_ID = 5
 WHERE (Order_ID = 5);

 UPDATE onlineshopping.orders
 SET ODP_Track_Number = 6 , Pay_ID = 6
 WHERE ( Order_ID = 6 );
 UPDATE onlineshopping.customers
 SET ODP_Track_Number = 1
 WHERE (Cust_ID = 1);

UPDATE onlineshopping.customers
 SET ODP_Track_Number = 2
 WHERE (Cust_ID = 2 );

UPDATE onlineshopping.customers
 SET ODP_Track_Number = 3
 WHERE (Cust_ID = 3);

 UPDATE onlineshopping.customers
 SET ODP_Track_Number = 4
 WHERE (Cust_ID = 4);

UPDATE onlineshopping.customers
 SET ODP_Track_Number = 5
 WHERE (Cust_ID = 5);

UPDATE onlineshopping.customers
 SET ODP_Track_Number = 6
WHERE (Cust_ID = 6);
```

Table Screenshot:

Categories:

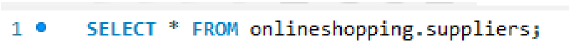


```
SELECT * FROM onlineshopping.categories;
```

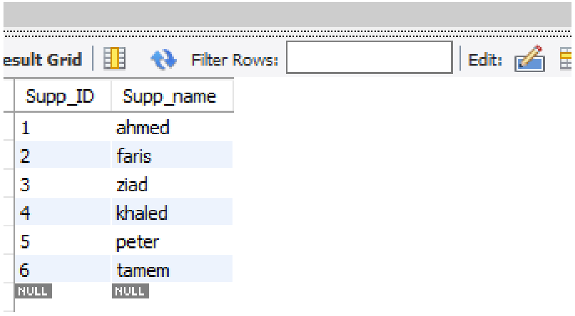


Cate_ID	Cate_name
1	Electronics
2	home
3	computer
NULL	NULL

Suppliers:

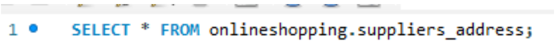


```
SELECT * FROM onlineshopping.suppliers;
```

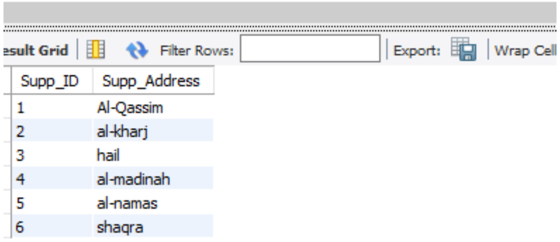


Supp_ID	Supp_name
1	ahmed
2	faris
3	ziad
4	khaled
5	peter
6	tamem
NULL	NULL

Suppliers_Address:

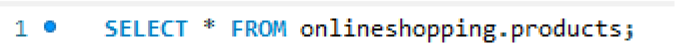


```
SELECT * FROM onlineshopping.suppliers_address;
```

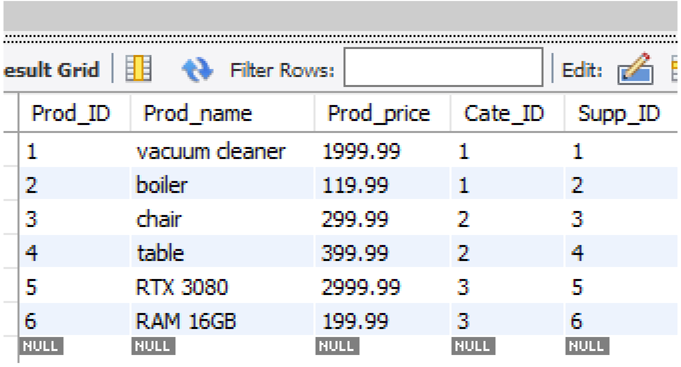


Supp_ID	Supp_Address
1	Al-Qassim
2	al-kharj
3	hail
4	al-madinah
5	al-namas
6	shaqra

Products:

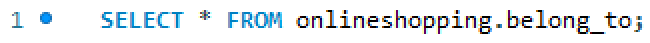


```sql
SELECT * FROM onlineshopping.products;
```

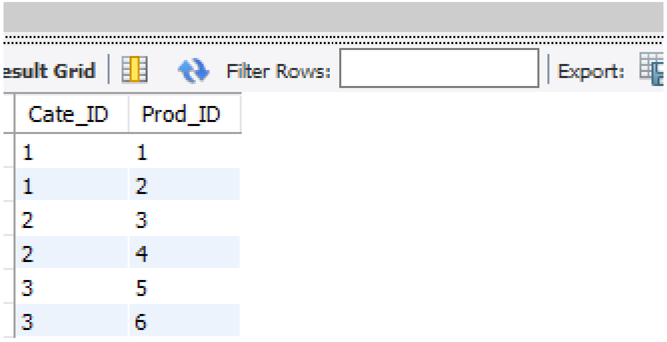


Prod_ID	Prod_name	Prod_price	Cate_ID	Supp_ID
1	vacuum cleaner	1999.99	1	1
2	boiler	119.99	1	2
3	chair	299.99	2	3
4	table	399.99	2	4
5	RTX 3080	2999.99	3	5
6	RAM 16GB	199.99	3	6
NULL	NULL	NULL	NULL	NULL

Belong_to:

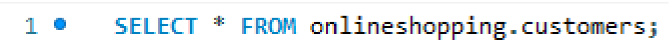


```sql
SELECT * FROM onlineshopping.belong_to;
```

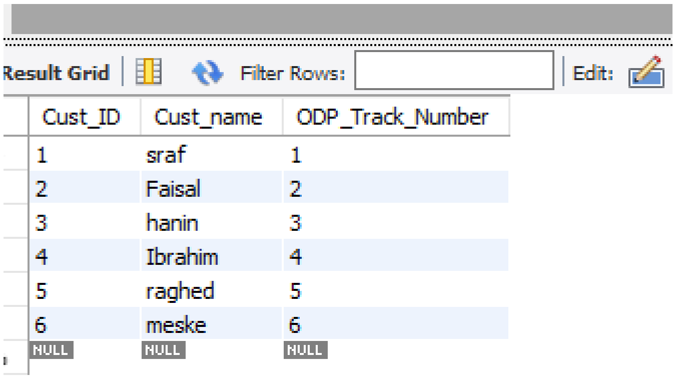


Cate_ID	Prod_ID
1	1
1	2
2	3
2	4
3	5
3	6

Customers:

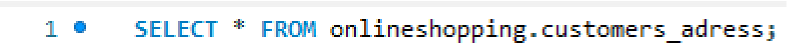


```sql
SELECT * FROM onlineshopping.customers;
```

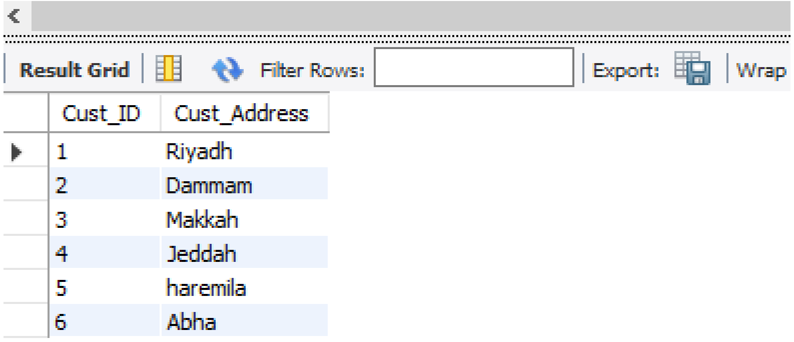


Cust_ID	Cust_name	ODP_Track_Number
1	sraf	1
2	Faisal	2
3	hanin	3
4	Ibrahim	4
5	raghed	5
6	meske	6
NULL	NULL	NULL

Customers_adress:

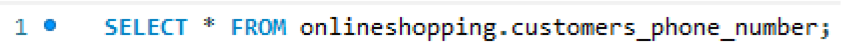


```sql
SELECT * FROM onlineshopping.customers_adress;
```

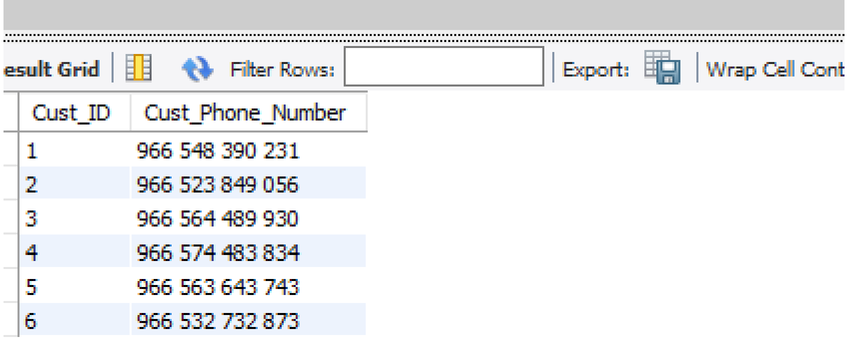


Cust_ID	Cust_Address
1	Riyadh
2	Dammam
3	Makkah
4	Jeddah
5	haremila
6	Abha

Customers_phone_number:

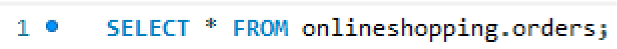


```
SELECT * FROM onlineshopping.customers_phone_number;
```

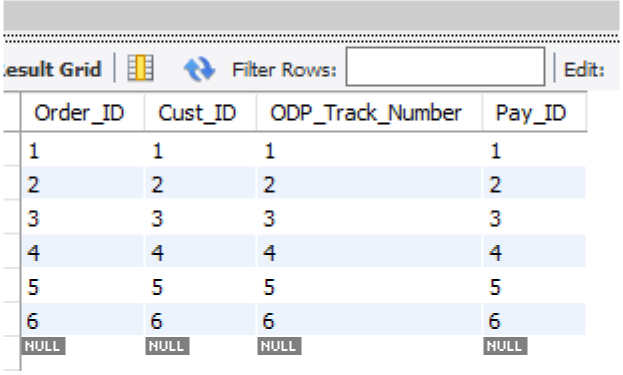


Cust_ID	Cust_Phone_Number
1	966 548 390 231
2	966 523 849 056
3	966 564 489 930
4	966 574 483 834
5	966 563 643 743
6	966 532 732 873

Orders:

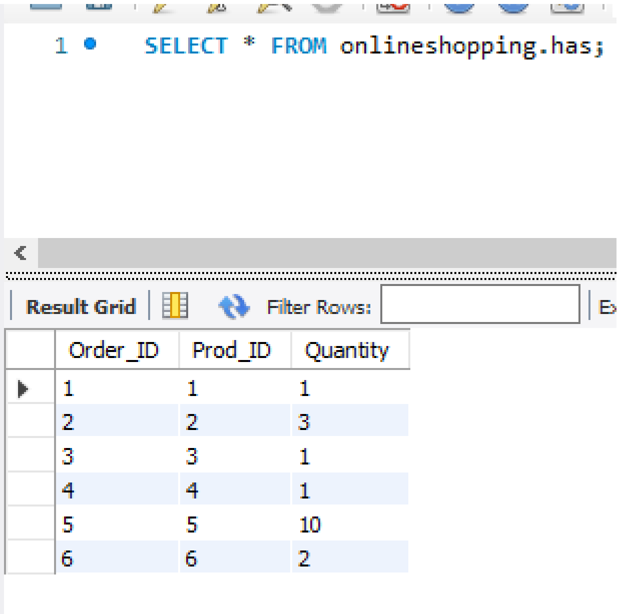


```
SELECT * FROM onlineshopping.orders;
```



Order_ID	Cust_ID	ODP_Track_Number	Pay_ID
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
NULL	NULL	NULL	NULL

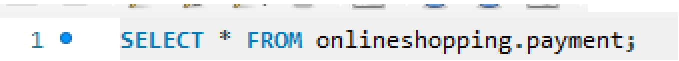
Has:



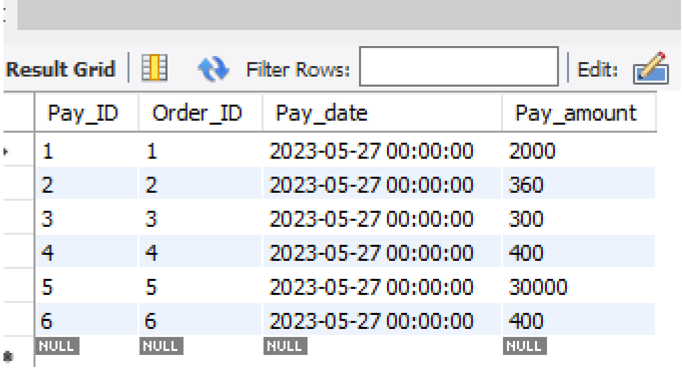
```
SELECT * FROM onlineshopping.has;
```

Order_ID	Prod_ID	Quantity
1	1	1
2	2	3
3	3	1
4	4	1
5	5	10
6	6	2

Payment:

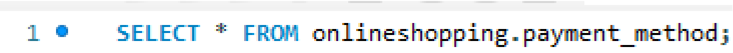


```
SELECT * FROM onlineshopping.payment;
```

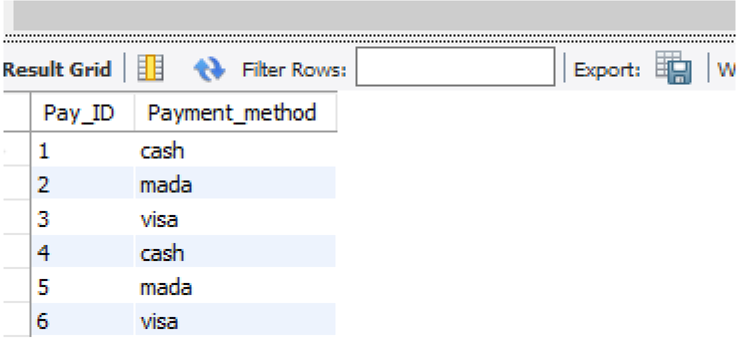


Pay_ID	Order_ID	Pay_date	Pay_amount
1	1	2023-05-27 00:00:00	2000
2	2	2023-05-27 00:00:00	360
3	3	2023-05-27 00:00:00	300
4	4	2023-05-27 00:00:00	400
5	5	2023-05-27 00:00:00	30000
6	6	2023-05-27 00:00:00	400
NULL	NULL	NULL	NULL

Payment_method:

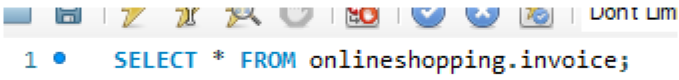


```
SELECT * FROM onlineshopping.payment_method;
```

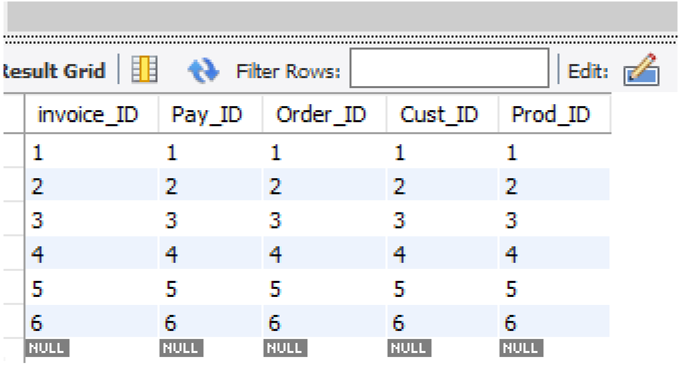


Pay_ID	Payment_method
1	cash
2	mada
3	visa
4	cash
5	mada
6	visa

Invoice:

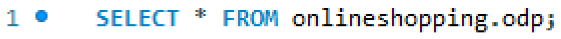


```
SELECT * FROM onlineshopping.invoice;
```

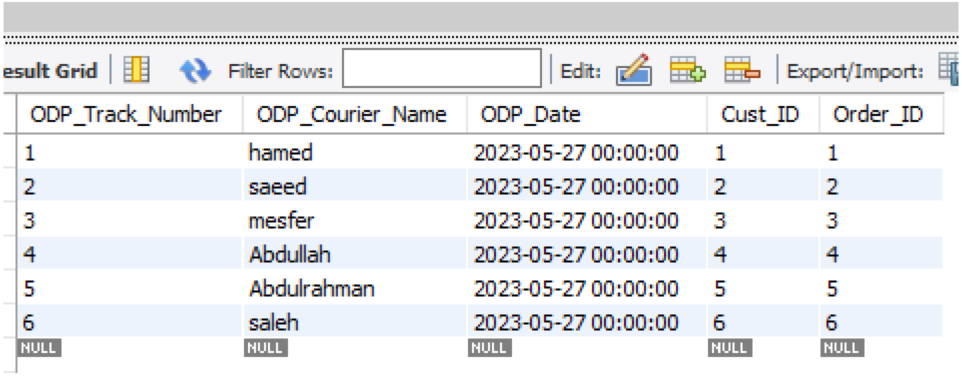


invoice_ID	Pay_ID	Order_ID	Cust_ID	Prod_ID
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
NULL	NULL	NULL	NULL	NULL

ODP:



```
SELECT * FROM onlineshopping.odp;
```



ODP_Track_Number	ODP_Courier_Name	ODP_Date	Cust_ID	Order_ID
1	hamed	2023-05-27 00:00:00	1	1
2	saeed	2023-05-27 00:00:00	2	2
3	mesfer	2023-05-27 00:00:00	3	3
4	Abdullah	2023-05-27 00:00:00	4	4
5	Abdulrahman	2023-05-27 00:00:00	5	5
6	saleh	2023-05-27 00:00:00	6	6
NULL	NULL	NULL	NULL	NULL

Query Implementation:

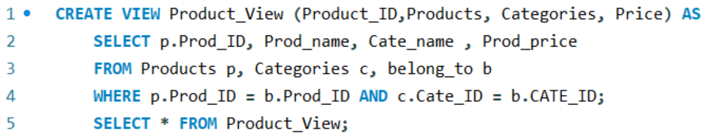
Views list:

1. Product View: this view returns the Product ID, name , price and which category belongs to (for the customers)

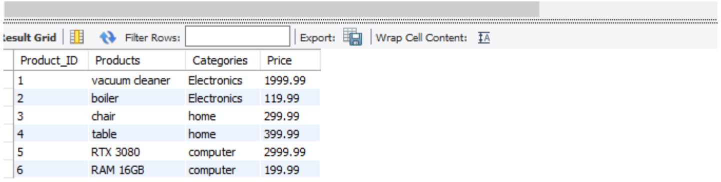
Query:

```
CREATE VIEW Product_View (Product_ID,Products, Categories, Price) AS
 SELECT p.Prod_ID, Prod_name, Cate_name , Prod_price
 FROM Products p, Categories c, belong_to b
 WHERE p.Prod_ID = b.Prod_ID AND c.Cate_ID = b.CATE_ID;
SELECT * FROM Product_View;
```

Result:



```
CREATE VIEW Product_View (Product_ID,Products, Categories, Price) AS
    SELECT p.Prod_ID, Prod_name, Cate_name , Prod_price
    FROM Products p, Categories c, belong_to b
    WHERE p.Prod_ID = b.Prod_ID AND c.Cate_ID = b.CATE_ID;
    SELECT * FROM Product_View;
```



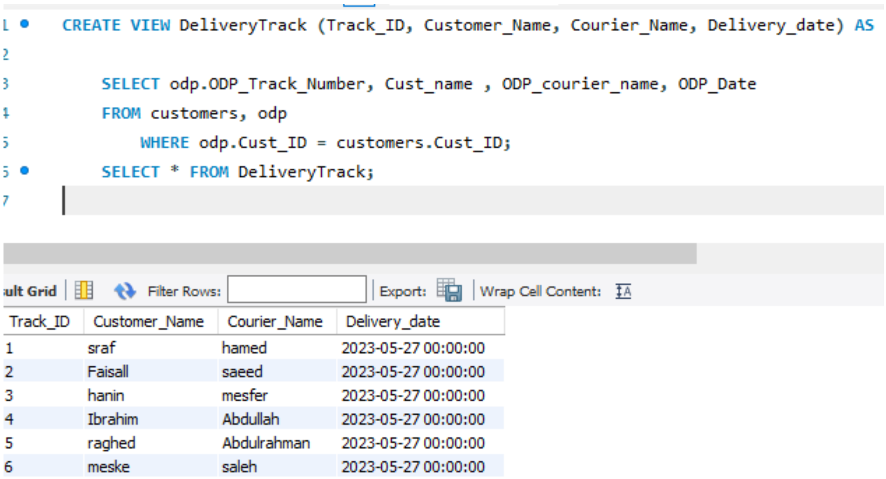
Product_ID	Products	Categories	Price
1	vacuum cleaner	Electronics	1999.99
2	boiler	Electronics	119.99
3	chair	home	299.99
4	table	home	399.99
5	RTX 3080	computer	2999.99
6	RAM 16GB	computer	199.99

2. Delivery track: This view returns the Track ID ,customer name, Courier Name and the Delivery date

Query:

```
CREATE VIEW DeliveryTrack (Track_ID, Customer_Name, Courier_Name, Delivery_date)
AS

SELECT odp.ODP_Track_Number, Cust_name , ODP_courier_name, ODP_Date
FROM customers, odp
WHERE odp.Cust_ID = customers.Cust_ID;
SELECT * FROM DeliveryTrack;
```

Result:



```
CREATE VIEW DeliveryTrack (Track_ID, Customer_Name, Courier_Name, Delivery_date) AS

    SELECT odp.ODP_Track_Number, Cust_name , ODP_courier_name, ODP_Date
    FROM customers, odp
        WHERE odp.Cust_ID = customers.Cust_ID;
    SELECT * FROM DeliveryTrack;
```

Track_ID	Customer_Name	Courier_Name	Delivery_date
1	sraf	hamed	2023-05-27 00:00:00
2	Faisall	saeed	2023-05-27 00:00:00
3	hanin	mesfer	2023-05-27 00:00:00
4	Ibrahim	Abdullah	2023-05-27 00:00:00
5	raghed	Abdulrahman	2023-05-27 00:00:00
6	meske	saleh	2023-05-27 00:00:00

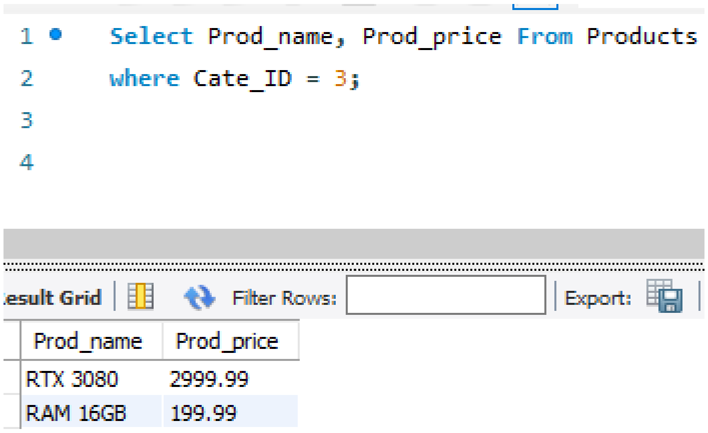
Queries list:

1. Product name and price from the computer category

   Query:

   Select Prod_name, Prod_price From Products
   where Cate_ID = 3;

   Result:



```
Select Prod_name, Prod_price From Products
where Cate_ID = 3;
```

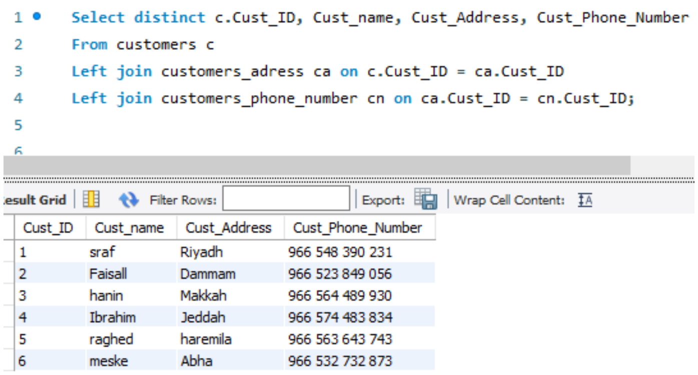
Prod_name	Prod_price
RTX 3080	2999.99
RAM 16GB	199.99

2. List customer ID, name, address, and phone number.

Query:

Select distinct c.Cust_ID, Cust_name, Cust_Address, Cust_Phone_Number
From customers c
Left join customers_adress ca on c.Cust_ID = ca.Cust_ID
Left join customers_phone_number cn on ca.Cust_ID = cn.Cust_ID;

Result:



```
Select distinct c.Cust_ID, Cust_name, Cust_Address, Cust_Phone_Number
From customers c
Left join customers_adress ca on c.Cust_ID = ca.Cust_ID
Left join customers_phone_number cn on ca.Cust_ID = cn.Cust_ID;
```

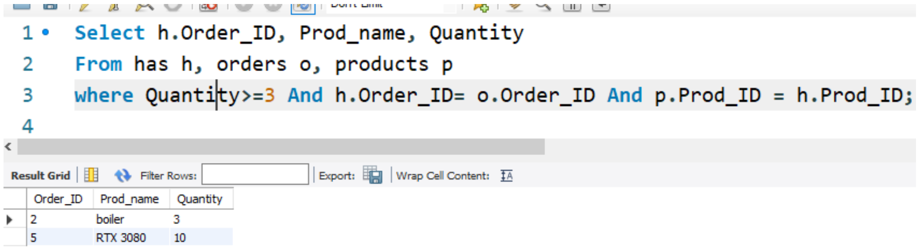
Cust_ID	Cust_name	Cust_Address	Cust_Phone_Number
1	sraf	Riyadh	966 548 390 231
2	Faisall	Dammam	966 523 849 056
3	hanin	Makkah	966 564 489 930
4	Ibrahim	Jeddah	966 574 483 834
5	raghed	haremila	966 563 643 743
6	meske	Abha	966 532 732 873

3. List of Orders with the product name that has at least product quantity of 3

Query:

Select h.Order_ID, Prod_name, Quantity
From has h, orders o, products p
where Quantity>=3 And h.Order_ID= o.Order_ID And p.Prod_ID = h.Prod_ID;

Result:



```
Select h.Order_ID, Prod_name, Quantity
From has h, orders o, products p
where Quantity>=3 And h.Order_ID= o.Order_ID And p.Prod_ID = h.Prod_ID;
```

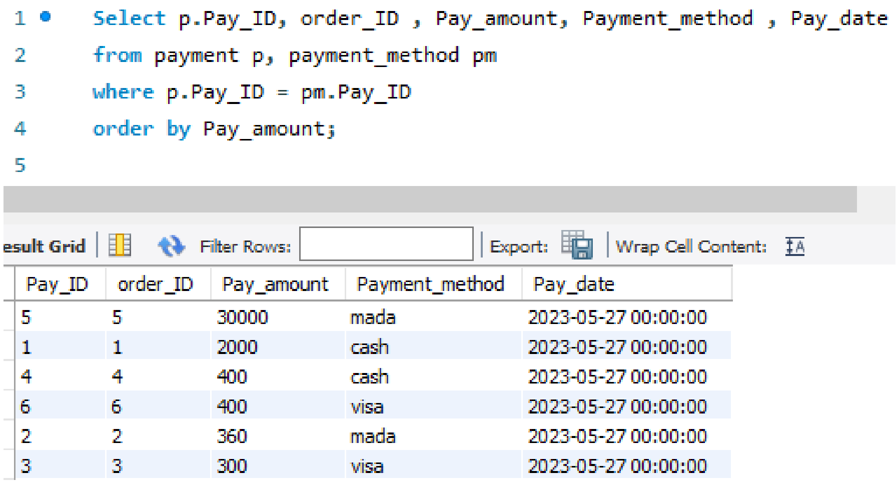
Order_ID	Prod_name	Quantity
2	boiler	3
5	RTX 3080	10

4. List of payment ordered by highest pay amount with the payment method

Query:

```
Select p.Pay_ID, order_ID , Pay_amount, Payment_method , Pay_date
from payment p, payment_method pm
where p.Pay_ID = pm.Pay_ID
order by Pay_amount desc;
```

Result:



```
Select p.Pay_ID, order_ID , Pay_amount, Payment_method , Pay_date
from payment p, payment_method pm
where p.Pay_ID = pm.Pay_ID
order by Pay_amount;
```

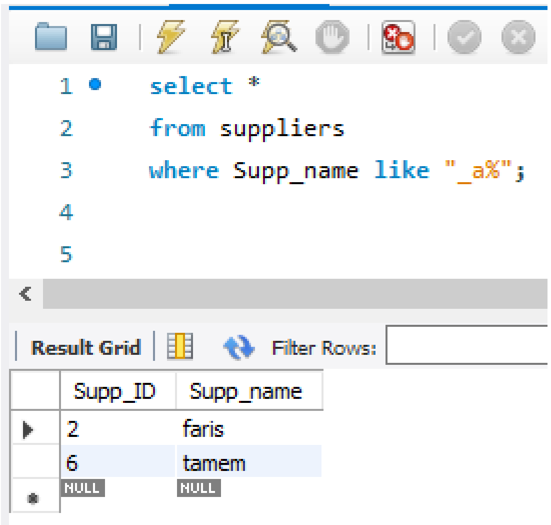
Pay_ID	order_ID	Pay_amount	Payment_method	Pay_date
5	5	30000	mada	2023-05-27 00:00:00
1	1	2000	cash	2023-05-27 00:00:00
4	4	400	cash	2023-05-27 00:00:00
6	6	400	visa	2023-05-27 00:00:00
2	2	360	mada	2023-05-27 00:00:00
3	3	300	visa	2023-05-27 00:00:00

5. Supplier names with the second letter being a

Query:

select *
from suppliers
where Supp_name like "_a%";

Result:



```
select *
from suppliers
where Supp_name like "_a%";
```

Supp_ID	Supp_name
2	faris
6	tamem
NULL	NULL

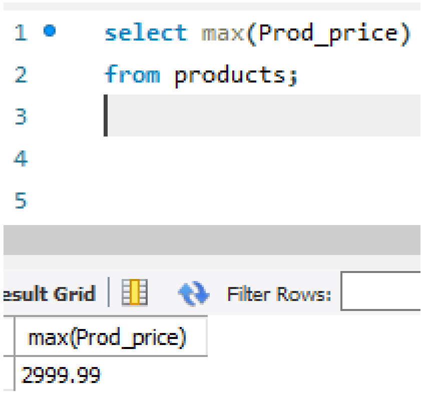
6. List the highest product price

Query:

select max(Prod_price)

from products;

Result:



```
select max(Prod_price)
from products;
```

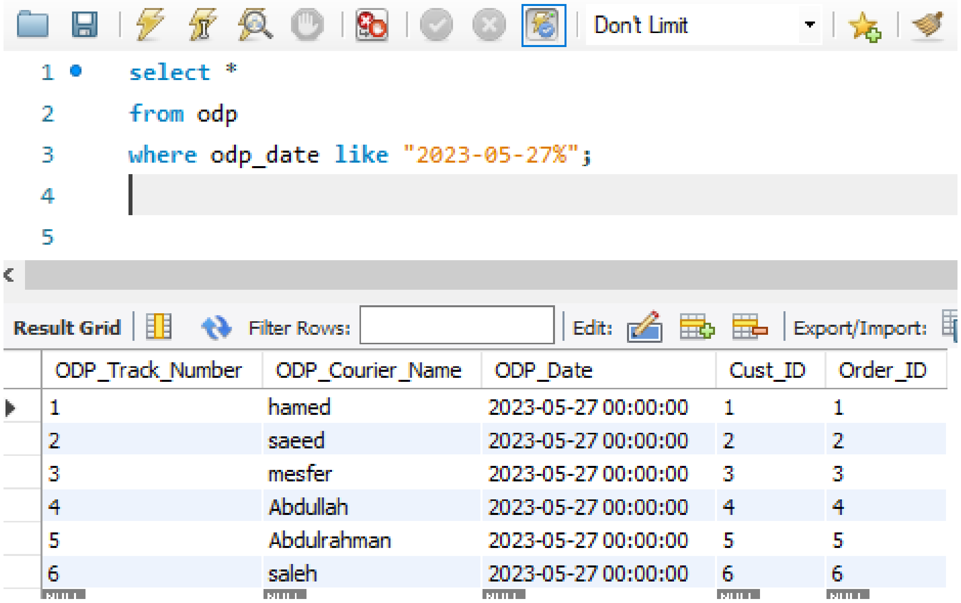
max(Prod_price)
2999.99

7. List all Order deliveries that has the date 2023-05-27

Query:

```
select *

from odp

where odp_date like "2023-05-27%";
```

Result:



```
select *
from odp
where odp_date like "2023-05-27%";
```

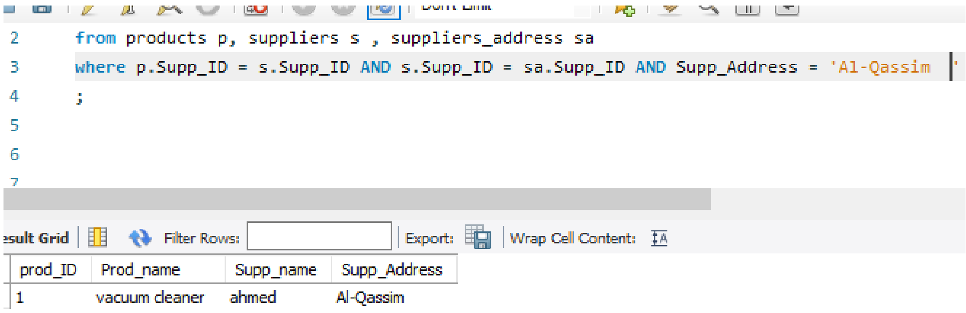
ODP_Track_Number	ODP_Courier_Name	ODP_Date	Cust_ID	Order_ID
1	hamed	2023-05-27 00:00:00	1	1
2	saeed	2023-05-27 00:00:00	2	2
3	mesfer	2023-05-27 00:00:00	3	3
4	Abdullah	2023-05-27 00:00:00	4	4
5	Abdulrahman	2023-05-27 00:00:00	5	5
6	saleh	2023-05-27 00:00:00	6	6

8. List the products that got supplied from a supplier from Al-Qassim with the supplier name

Query:

```
select prod_ID, Prod_name, Supp_name, Supp_Address

from products p, suppliers s , suppliers_address sa

where p.Supp_ID = s.Supp_ID AND s.Supp_ID = sa.Supp_ID AND
Supp_Address = ' Al-Qassim     ';
```

Result:



```
from products p, suppliers s , suppliers_address sa
where p.Supp_ID = s.Supp_ID AND s.Supp_ID = sa.Supp_ID AND Supp_Address = 'Al-Qassim  '
;
```

prod_ID	Prod_name	Supp_name	Supp_Address
1	vacuum cleaner	ahmed	Al-Qassim