```
drop database if exists order_processing;
create database order_processing;
use order_processing;

create table if not exists Customers (
cust_id int primary key,
cname varchar(35) not null,
city varchar(35) not null
);

create table if not exists Orders (
      order_id int primary key,
      odate date not null,
      cust_id int,
      order_amt int not null,
      foreign key (cust_id) references Customers(cust_id) on delete cascade
     );

create table if not exists Items (
     item_id int primary key,
     unitprice int not null
     );

create table if not exists OrderItems (
     order_id int not null,
     item_id int not null,
     qty int not null,
     foreign key (order_id) references Orders(order_id) on delete cascade,
     foreign key (item_id) references Items(item_id) on delete cascade
    );

create table if not exists Warehouses (
     warehouse_id int primary key,
     city varchar(35) not null
    );

create table if not exists Shipments (
     order_id int not null,
     warehouse_id int not null,
     ship_date date not null,
     foreign key (order_id) references Orders(order_id) on delete cascade,
     foreign key (warehouse_id) references Warehouses(warehouse_id) on delete cascade
    );

INSERT INTO Customers VALUES
     (0001, "Customer_1", "Mysuru"),
     (0002, "Customer_2", "Bengaluru"),
     (0003, "Kumar", "Mumbai"),
     (0004, "Customer_4", "Dehli"),
     (0005, "Customer_5", "Bengaluru");
```

```
INSERT INTO Orders VALUES
    (001, "2020-01-14", 0001, 2000),
    (002, "2021-04-13", 0002, 500),
    (003, "2019-10-02", 0003, 2500),
    (004, "2019-05-12", 0005, 1000),
    (005, "2020-12-23", 0004, 1200);

INSERT INTO Items VALUES
    (0001, 400),
    (0002, 200),
    (0003, 1000),
    (0004, 100),
    (0005, 500);

INSERT INTO Warehouses VALUES
    (0001, "Mysuru"),
    (0002, "Bengaluru"),
    (0003, "Mumbai"),
    (0004, "Dehli"),
    (0005, "Chennai");

INSERT INTO OrderItems VALUES
    (001, 0001, 5),
    (002, 0005, 1),
    (003, 0005, 5),
    (004, 0003, 1),
    (005, 0004, 12);

INSERT INTO Shipments VALUES
    (001, 0002, "2020-01-16"),
    (002, 0001, "2021-04-14"),
    (003, 0004, "2019-10-07"),
    (004, 0003, "2019-05-16"),
    (005, 0005, "2020-12-23");

SELECT * FROM Customers;
+---------+------------+-----------+
| cust_id | cname      | city      |
+---------+------------+-----------+
|       1 | Customer_1 | Mysuru    |
|       2 | Customer_2 | Bengaluru |
|       3 | Kumar      | Mumbai    |
|       4 | Customer_4 | Dehli     |
|       5 | Customer_5 | Bengaluru |
+---------+------------+-----------+
```

SELECT * FROM Orders;

order_id	odate	cust_id	order_amt
1	2020-01-14	1	2000
2	2021-04-13	2	500
3	2019-10-02	3	2500
4	2019-05-12	5	1000
5	2020-12-23	4	1200

SELECT * FROM OrderItems;

order_id	item_id	qty
1	1	5
2	5	1
3	5	5
4	3	1
5	4	12

SELECT * FROM Items;

item_id	unitprice
1	400
2	200
3	1000
4	100
5	500

SELECT * FROM Shipments;

order_id	warehouse_id	ship_date
1	2	2020-01-16
2	1	2021-04-14
3	4	2019-10-07
4	3	2019-05-16
5	5	2020-12-23

SELECT * FROM Warehouses;

warehouse_id	city
1	Mysuru
2	Bengaluru
3	Mumbai
4	Dehli
5	Chennai

-- List the Order# and Ship_date for all orders shipped from Warehouse# "0001".

select order_id,ship_date from Shipments where warehouse_id=0001;

```
+----------+------------+
| order_id | ship_date  |
+----------+------------+
|        2 | 2021-04-14 |
+----------+------------+
```

-- List the Warehouse information from which the Customer named "Kumar" was supplied his orders. Produce a listing of Order#, Warehouse#

select order_id,warehouse_id from Warehouses natural join Shipments where order_id in (select order_id from Orders where cust_id in (Select cust_id from Customers where cname like "%Kumar%"));

```
+----------+--------------+
| order_id | warehouse_id |
+----------+--------------+
|        3 |            4 |
+----------+--------------+
```

--Produce a listing: Cname, #ofOrders, Avg_Order_Amt, where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer. (Use aggregate functions)

select cname, COUNT(*) as no_of_orders, AVG(order_amt) as avg_order_amt
from Customers c, Orders o
where c.cust_id=o.cust_id
group by cname;

```
+------------+--------------+---------------+
| cname      | no_of_orders | avg_order_amt |
+------------+--------------+---------------+
| Customer_1 |            1 |     2000.0000 |
| Customer_2 |            1 |      500.0000 |
| Kumar      |            1 |     2500.0000 |
| Customer_4 |            1 |     1200.0000 |
| Customer_5 |            1 |     1000.0000 |
+------------+--------------+---------------+
```

-- Delete all orders for customer named "Kumar".

delete from Orders where cust_id = (select cust_id from Customers where cname like "%Kumar%");

SELECT * FROM Orders;

```
+----------+------------+---------+-----------+
| order_id | odate      | cust_id | order_amt |
+----------+------------+---------+-----------+
|        3 | 2019-10-02 |       3 |      2500 |
+----------+------------+---------+-----------+
```

SELECT * FROM OrderItems;

order_id	item_id	qty
3	5	5

SELECT * FROM Shipments;

order_id	warehouse_id	ship_date
3	4	2019-10-07

-- Find the item with the maximum unit price.

select max(unitprice) from Items;

max(unitprice)
1000

-- A tigger that updates order_amount based on quantity and unit price of order_item

```
DELIMITER $$
create trigger UpdateOrderAmt
after insert on OrderItems
for each row
BEGIN
update Orders set order_amt=(new.qty*(select distinct unitprice from Items
NATURAL JOIN OrderItems where item_id=new.item_id)) where
Orders.order_id=new.order_id;
END; $$

DELIMITER;
INSERT INTO Orders VALUES
(006, "2020-12-23", 0004, 1200);
```

SELECT * FROM Orders;

order_id	odate	cust_id	order_amt
1	2020-01-14	1	2000
2	2021-04-13	2	500
4	2019-05-12	5	1000
5	2020-12-23	4	1200
6	2020-12-23	4	2000

INSERT INTO OrderItems VALUES
(006, 0001, 5);
SELECT * FROM OrderItems;

order_id	item_id	qty
1	1	5
2	5	1
4	3	1
5	4	12
6	1	5

-- Create a view to display orderID and shipment date of all orders shipped from a warehouse 5.

create view ShipmentDatesFromWarehouse5 as
select order_id, ship_date
from Shipments
where warehouse_id=5;
select * from ShipmentDatesFromWarehouse5;

order_id	ship_date
5	2020-12-23