Order Processing DB

ORDER PROCESSING DB

CUSTOMER

custno	cname	city

ORDER

orderno	odate	custno	ord_amt

ORDER_ITEM

orderno	itemno	quantity

ITEM

Itemno	unitprice

SHIPMENT

orderno	warehouseno	ship_date

WAREHOUSE

warehouseno	city

Queries Used to CREATE and INSERT

```
    create table customer (custno int primary key, cname varchar(20), city
varchar(20));
    create table Oorder(orderno int  primary key, odate date,custno int
foreign key references customer,ord_amt int);
    create table order_item(orderno int foreign key references Oorder,itemno
int primary key);
    create table item (itemno int foreign key references order_item,
unit_price int);
    create table warehouse(warehouseno int primary key,city varchar(20));
    create table shipment(orderno int foreign key references Oorder,
```

```sql
warehouseno int foreign key references warehouse, ship_date date );

    insert into customer values(1001,'Ben','Bangalore');
    insert into customer values(1002,'Vikas','Delhi');
    insert into customer values(1003,'Tilly','Singapore');
    insert into customer values(1004,'John','Mumbai');
    insert into customer values(1005,'Brad','Kansas');

    insert into Oorder values(2001,'2017-04-13',1001,20000);
    insert into Oorder values(2002,'2017-06-16',1003,30000);
    insert into Oorder values(2003,'2017-10-27',1004,25000);
    insert into Oorder values(2004,'2017-01-30',1001,10000);
    insert into Oorder values(2005,'2017-07-02',1005,15000);

    insert into order_item values(2002,3001);
    insert into order_item values(2003,3002);
    insert into order_item values(2004,3003);
    insert into order_item values(2004,3004);
    insert into order_item values(2005,3005);
    insert into order_item values(2001,3006);
    insert into order_item values(2005,3007);

    insert into item values(3001,3000);
    insert into item values(3002,3000);
    insert into item values(3003,4000);
    insert into item values(3004,5000);
    insert into item values(3005,3000);
    insert into item values(3006,6000);
    insert into item values(3007,7000);

    insert into warehouse values(1,'Bangalore');
    insert into warehouse values(2,'Mysore');
    insert into warehouse values(3,'Tirichy');
    insert into warehouse values(4,'Salem');
    insert into warehouse values(5,'kanyakumari');
    insert into warehouse values(6,'Bangalore');
    insert into warehouse values(7,'Bangalore');

    insert into shipment values(2001,1,'2017-04-13');
    insert into shipment values(2003,6,'2017-04-13');
    insert into shipment values(2005,7,'2017-04-13');
    insert into shipment values(2003,2,'2017-09-10');
    insert into shipment values(2004,3,'2017-03-11');
```

```
    insert into shipment values(2002,4,'2017-01-13');
    insert into shipment values(2005,5,'2017-02-15');
```

WEBSITE USED FOR MYSQL

Link : http://sqlfiddle.com/

Queries

1. Create the above tables by properly specifying the primary keys and foreign keys.

```sql
    create table customer (custno int primary key, cname varchar(20), city
varchar(20));
    create table Oorder(orderno int  primary key, odate date,custno int
foreign key references customer,ord_amt int);
    create table order_item(orderno int foreign key references Oorder,itemno
int primary key);
    create table item (itemno int foreign key references order_item,
unit_price int);
    create table warehouse(warehouseno int primary key,city varchar(20));
    create table shipment(orderno int foreign key references Oorder,
warehouseno int foreign key references warehouse, ship_date date );
```

2. Enter at least five tuples for each relation.

```sql
    insert into customer values(1001,'Ben','Bangalore');
    insert into customer values(1002,'Vikas','Delhi');
    insert into customer values(1003,'Tilly','Singapore');
    insert into customer values(1004,'John','Mumbai');
    insert into customer values(1005,'Brad','Kansas');
```

```sql
insert into Oorder values(2001,'2017-04-13',1001,20000);
insert into Oorder values(2002,'2017-06-16',1003,30000);
insert into Oorder values(2003,'2017-10-27',1004,25000);
insert into Oorder values(2004,'2017-01-30',1001,10000);
insert into Oorder values(2005,'2017-07-02',1005,15000);

insert into order_item values(2002,3001);
insert into order_item values(2003,3002);
insert into order_item values(2004,3003);
insert into order_item values(2004,3004);
insert into order_item values(2005,3005);
insert into order_item values(2001,3006);
insert into order_item values(2005,3007);

insert into item values(3001,3000);
insert into item values(3002,3000);
insert into item values(3003,4000);
insert into item values(3004,5000);
insert into item values(3005,3000);
insert into item values(3006,6000);
insert into item values(3007,7000);

insert into warehouse values(1,'Bangalore');
insert into warehouse values(2,'Mysore');
insert into warehouse values(3,'Tirichy');
insert into warehouse values(4,'Salem');
insert into warehouse values(5,'kanyakumari');
insert into warehouse values(6,'Bangalore');
insert into warehouse values(7,'Bangalore');

insert into shipment values(2001,1,'2017-04-13');
insert into shipment values(2003,6,'2017-04-13');
insert into shipment values(2005,7,'2017-04-13');
insert into shipment values(2003,2,'2017-09-10');
insert into shipment values(2004,3,'2017-03-11');
insert into shipment values(2002,4,'2017-01-13');
insert into shipment values(2005,5,'2017-02-15');
```

3. Produce a listing: custname , No_of_orders , Avg_order_amount , 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.

```
select cname,count(*), avg(ord_amt) from customer,Oorder
where customer.custno = Oorder.custno group by cname;
```

cname	count(*)	avg(ord_amt)
Ben	2	15000
Brad	1	15000
John	1	25000
Tilly	1	30000

4. List the orderno for orders that were shipped from all the warehouses that the company has in a specific city.

```
select distinct orderno from shipment
where warehouseno in
(select all warehouseno from warehouse where city = 'Bangalore');
```

orderno
2001

5. Demonstrate the deletion of an item from the ITEM table and demonstrate a method of handling the rows in the ORDER_ITEM table that contains this particular item.

```
delete from item where itemno = 3007;
```

Before

itemno	unit_price

itemno	unit_price
3001	3000
3002	3000
3003	4000
3004	5000
3005	3000
3006	6000
3007	7000

After

itemno	unit_price
3001	3000
3002	3000
3003	4000
3004	5000
3005	3000
3006	6000

```
update order_item
set orderno = 2002
where itemno = 3006;
```

Before

orderno	itemno
2002	3001
2003	3002
2004	3003
2004	3004
2005	3005
2001	3006
2005	3007

After

orderno	itemno
2002	3001
2003	3002
2004	3003
2004	3004
2005	3005
2002	3006
2005	3007