

TERM PAPER REPORT

E-COMMERCE DATABASE MANAGEMENT SYSTEM

NAME : BELAGANTI SAHAJA

BRANCH : BIOMEDICAL

SUBJECT: DATABASE MANAGEMENT SYSTEMS (DBMS)

SEMESTER : VI

ROLL NO. : 19111018

Currently, there are many shops which have their business up to limited areas only. Customers have to visit their shop and shopkeepers have to show many products and their details again and again. It is quite time taking so they hire workers to show products. Many customers have to wait outside because the size of the shop is small and due to limited workers. For personal work, shopkeepers have to close shop for a day.

Few problems for shopkeeper:

- Single shop covers business in a specific area
- Expenditures on shop like electricity bills, rent and salary of workers
- Storing of data are on papers
- Bills are manually created on papers
- Presence of shopkeeper at shop is required
- There is certain time limit to have shop open
- For increasing business, he has to open a shop in another area which would have above mentioned all problems.

Few problems for Customers:

- Customers have to manage their timings with the shop and have to personally visit the shop for the details of the product.
- Have to wait for the shopkeeper to be free from other customers.
- Customers can trust pre-users of the same product rather than shopkeepers, here reviews unavailable.
- On the spot selection and comparison of products is quite difficult.

Objectives of the Proposed System :

Proposed system is more beneficial than the current one. As every customer can see the product details and compare two products on their mobile itself. Customers are assured about the product by reading reviews from others who used that product. Shopkeepers do not have to explain again and again. Shopkeeper can manage his personal work along with his business. Shopkeeper can expand his business on a large scale without investing in making different shops.

Benefits of New System for both -

- Expands business on large scale
- Efficient way to select and compare products on customer's free time
- No unauthorized user can access the data
- Maintain detail of products and all record of money transaction and orders
- Less Capital needed on workers, electricity or rent etc
- Handling of many customers is very easy here
- No limits on opening and closing timings of shop
- Shopkeeper can do other personal work along with business
- Prices of product are somewhat less than shops as there no extra expenditures on shops

Definition :

This project aims at maintaining all the information pertaining to the customers, vendors, products, and their categories, orders, and couriers. It enables vendors to set up online shops, customers to browse through the shops, and a system administrator to approve and reject requests for new shops and maintain lists of shop categories. The system manages the items in the shop and also helps customers purchase them online without having to visit the shop

physically. The online shopping system will use the internet as the sole method for selling goods, products, and services to its consumers. The website will show all products in a categorized manner. Customers can browse any product for its price, other details and can order the product by using their registered account. The customer has to pay the order amount at the time of delivery.

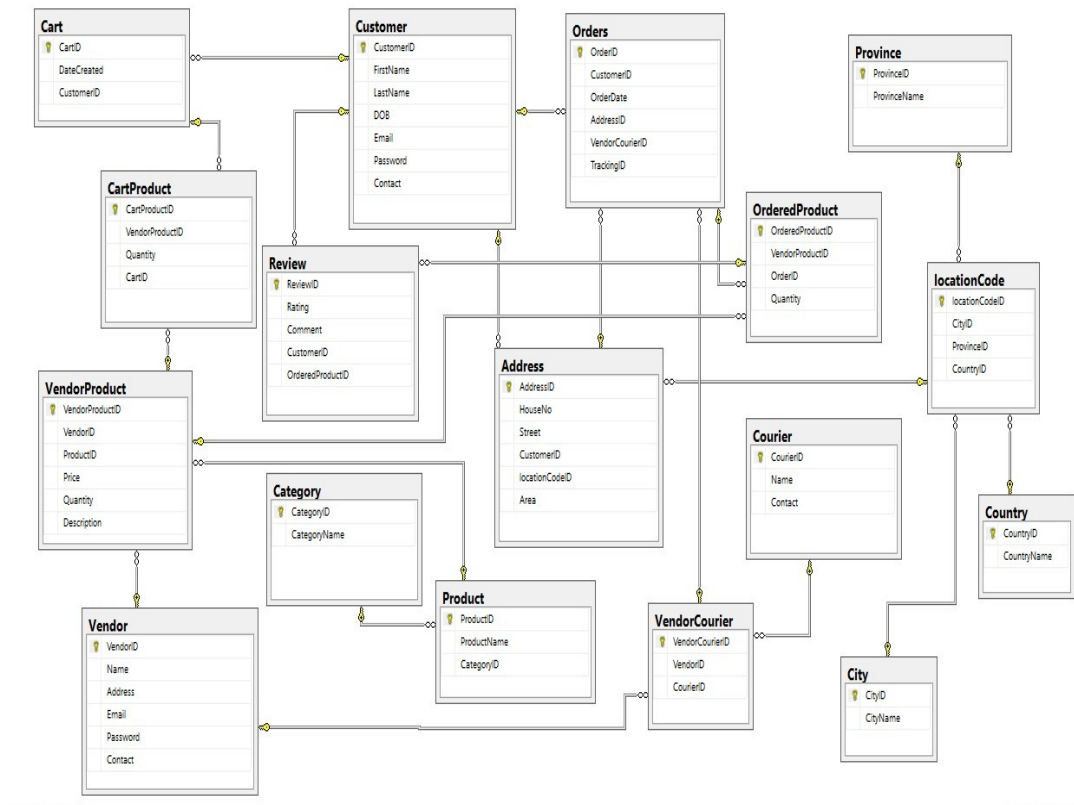
Purpose :

There are many sellers who want to expand their business globally but due to unavailability of capital they can't do it. Many of them have one or two shops and a range of customers within those and neighboring cities only. By this project, those people will get a global platform to sell their product to a vast customer range. Moreover, it's a motivation for small scale sellers to expand their business.

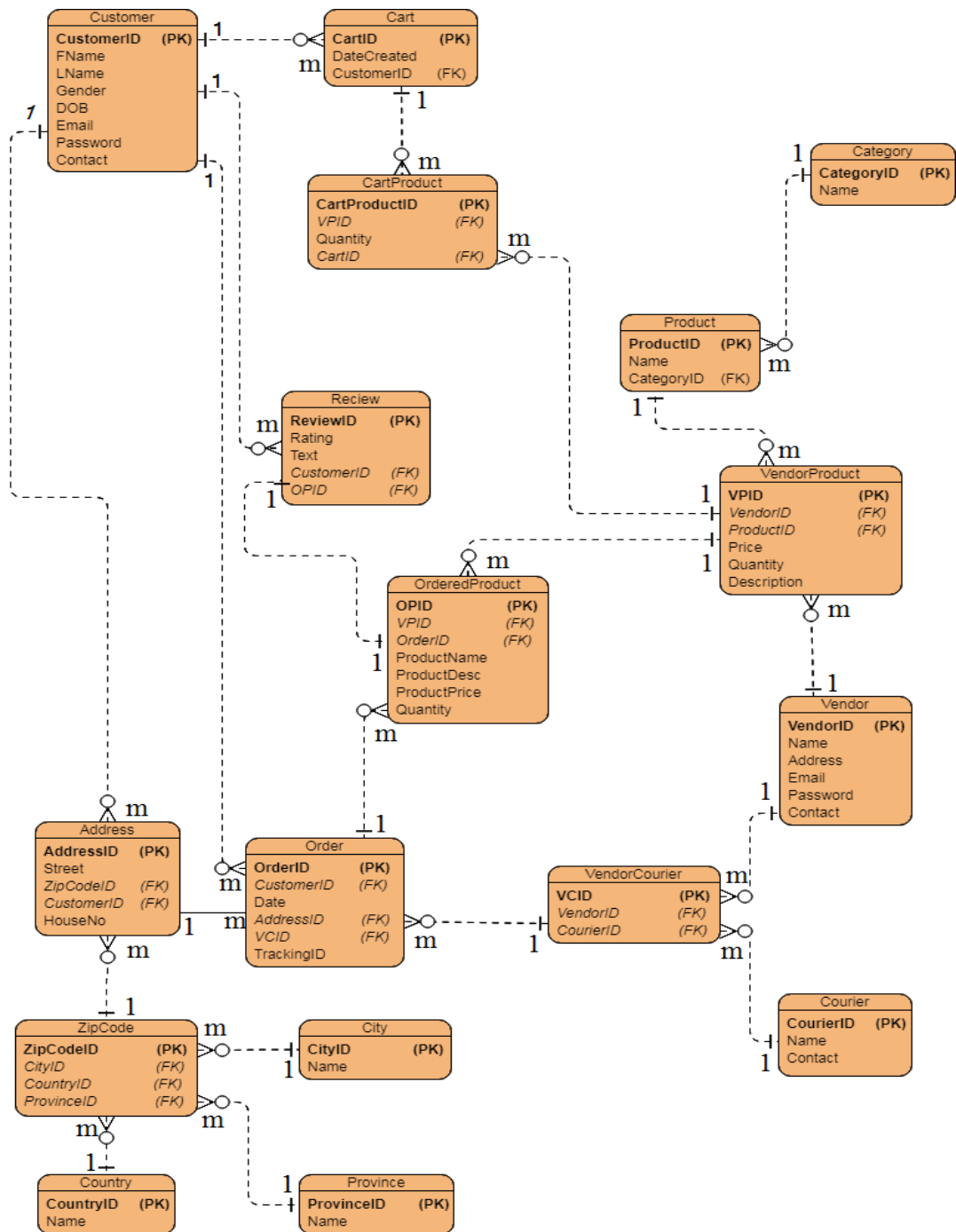
Scope & Objective :

Nowadays, Every person has a personal smartphone. This project is giving a chance to think and decide patiently, which product will be best for them. E-commerce draws on such technologies as electronic funds transfer, supply chain management, Internet marketing, online transaction processing. In a retailer shopping system, customers will not get the actual truth about the product as the details will be given by the shopkeeper itself. But here, customers can compare products, check reviews of the customer who really bought and used that product.

ENTITY RELATIONSHIP DIAGRAM



RELATION DATABASE SCHEMA



DATABASE IMPLEMENTATION

Creating DATABASE and TABLES :

CREATE DATABASE OnlineShopping;

Use OnlineShopping;

CREATE TABLE Customer

```
(
    CustomerID int IDENTITY(1, 1) PRIMARY KEY,
    FirstName varchar(255) NOT NULL,
    LastName varchar(255) NOT NULL,
    DOB date NOT NULL,
    Email varchar(255) NOT NULL,
    Password varchar(255) NOT NULL,
    Contact varchar(255) NOT NULL
);
```

CREATE TABLE Country

```
(
    CountryID int IDENTITY(1, 1) PRIMARY KEY,
    CountryName varchar(255) NOT NULL
);
```

CREATE TABLE Province

```
(
    ProvinceID int IDENTITY(1, 1) PRIMARY KEY,
    ProvinceName varchar(255) NOT NULL
);
```

CREATE TABLE City

```
(
    CityID int IDENTITY(1, 1) PRIMARY KEY,
    CityName varchar(255) NOT NULL
);
```

CREATE TABLE ZipCode

```
(
    ZipCodeID int IDENTITY(1, 1) PRIMARY KEY,
    CityID int FOREIGN KEY REFERENCES City(CityID) NOT NULL,
    ProvinceID int FOREIGN KEY REFERENCES Province(ProvinceID) NOT
NULL,
    CountryID int FOREIGN KEY REFERENCES Country(CountryID) NOT
NULL
);
```

CREATE TABLE Address

```
(
    AddressID int IDENTITY(1, 1) PRIMARY KEY,
    HouseNo varchar(255) NOT NULL,
```

```

        Street int NOT NULL,
        CustomerID int FOREIGN KEY REFERENCES Customer(CustomerID)
NOT NULL,
        ZipCodeID int FOREIGN KEY REFERENCES ZipCode(ZipCodeID) NOT
NULL,
        Area varchar(255) NOT NULL
    );

```

```

CREATE TABLE Category
(
    CategoryID int IDENTITY(1, 1) PRIMARY KEY,
    CategoryName varchar(255) NOT NULL
);

```

```

CREATE TABLE Vendor
(
    VendorID int IDENTITY(1, 1) PRIMARY KEY,
    Name varchar(255) NOT NULL,
    Address text NOT NULL,
    Email varchar(255) NOT NULL,
    Password varchar(255) NOT NULL,
    Contact varchar(255) NOT NULL
);

```

```

CREATE TABLE Product
(
    ProductID int IDENTITY(1, 1) PRIMARY KEY,
    ProductName varchar(255) NOT NULL,
    CategoryID int FOREIGN KEY REFERENCES Category(CategoryID)
NOT NULL
);

```

```

CREATE TABLE VendorProduct
(
    VendorProductID int IDENTITY(1, 1) PRIMARY KEY,
    VendorID int FOREIGN KEY REFERENCES Vendor(VendorID) NOT
NULL,
    ProductID int FOREIGN KEY REFERENCES Product(ProductID) NOT
NULL,
    Price decimal(19, 2) NOT NULL,
    Quantity int NOT NULL,
    Description text NOT NULL
);

```

```

CREATE TABLE Courier

```

```

(
    CourierID int IDENTITY(1, 1) PRIMARY KEY,
    Name varchar(255) NOT NULL,
    Contact varchar(255) NOT NULL
);

CREATE TABLE VendorCourier
(
    VendorCourierID int IDENTITY(1, 1) PRIMARY KEY,
    VendorID int FOREIGN KEY REFERENCES Vendor(VendorID) NOT
NULL,
    CourierID int FOREIGN KEY REFERENCES Courier(CourierID) NOT
NULL
);

CREATE TABLE Orders
(
    OrderID int IDENTITY(1, 1) PRIMARY KEY,
    CustomerID int FOREIGN KEY REFERENCES Customer(CustomerID)
NOT NULL,
    OrderDate date NOT NULL,
    AddressID int FOREIGN KEY REFERENCES Address(AddressID) NOT
NULL,
    VendorCourierID int FOREIGN KEY REFERENCES VendorCourier(VendorCourierID)
NOT NULL,
    TrackingID varchar(255) NOT NULL
);

CREATE TABLE OrderedProduct
(
    OrderedProductID int IDENTITY(1, 1) PRIMARY KEY,
    VendorProductID int FOREIGN KEY REFERENCES VendorProduct(VendorProductID)
NOT NULL,
    OrderID int FOREIGN KEY REFERENCES Orders(OrderID) NOT NULL,
    Quantity int NOT NULL
);

CREATE TABLE Review
(
    ReviewID int IDENTITY(1, 1) PRIMARY KEY,
    Rating tinyint NOT NULL,
    Comment text,
    CustomerID int FOREIGN KEY REFERENCES Customer(CustomerID)
NOT NULL,
    OrderedProductID int FOREIGN KEY REFERENCES OrderedProduct(OrderedProductID)
NOT NULL,

```



```

);

CREATE TABLE Cart
(
    CartID int IDENTITY(1, 1) PRIMARY KEY,
    DateCreated date NOT NULL,
    CustomerID int FOREIGN KEY REFERENCES Customer(CustomerID)
NOT NULL,
);

CREATE TABLE CartProduct
(
    CartProductID int IDENTITY(1, 1) PRIMARY KEY,
    VendorProductID int FOREIGN KEY REFERENCES VendorProduct(VendorProductID)
NOT NULL,
    Quantity int NOT NULL,
    CartID int FOREIGN KEY REFERENCES Cart(CartID) NOT NULL
);

```

INSERTING DATA

Note: All information is fake and generated for the demonstration of the project.

use OnlineShopping;

Insert Into Category Values ('Android Smart TV Box/Air Mouse');

Insert Into City Values ('Karachi');

Insert Into Orders Values ('1','2011-01-27','1','318','77425889862');

Insert Into Product Values ('Air Mouse C120 for Android and Smart TV','1');

Insert Into Province Values ('Punjab');

Insert Into Review Values ('4','Great Product Slow delivery. Such slow much wow','1','1');

QUERIES

Categories available at a particular vendor for selling :

use OnlineShopping

```

select CategoryName from Category
where CategoryID in
    (select Distinct CategoryID from Product
     where ProductID in
        (select ProductID
         from VendorProduct
         where VendorID='3'))

```

Category with highest selling product :

use OnlineShopping;

```

Select VendorName, ProductName, OrderedQuantity, Category.CategoryName
From Category

```

```

    inner join (
    Select VendorName, Product.ProductName, OrderedQuantity, CategoryID
    From Product

```

```

        inner join (
        Select Vendor.Name as VendorName, ProductID, OrderedQuantity
        From Vendor

```

```

            inner join (
            Select vendorID, ProductID, OrderedQuantity
            From VendorProduct

```

```

                inner join (
                Select TOP 1 OrderedProduct.VendorProductID, sum(Quantity)
as OrderedQuantity
                From OrderedProduct Group By VendorProductID Order By
OrderedQuantity Desc
                ) vpfk on vpfk.VendorProductID = VendorProduct.VendorProductID

            ) vpid on vpid.VendorID = vendor.VendorID

        ) vpv on vpv.ProductID = Product.ProductID

    ) pc on pc.CategoryID = Category.CategoryID

```

Customer's Cart :

use OnlineShopping

```

select Name,ProductName,Qty
from Product

```

```
inner join(  
select Vendor.Name,ProductID,Qty from Vendor
```

```
inner join(  
select Vendorid,Productid,Qty from VendorProduct
```

```
inner join(  
select CartProduct.VendorProductID,CartProduct.Quantity as Qty from Cart-  
Product
```

```
inner join(  
Select CartID from Cart where CustomerID='170') cid on cid.CartID=CartProduct.CartID
```

```
) vpid on vpid.VendorProductID=VendorProduct.VendorProductID
```

```
        )vid on vid.vendorid=vendor.vendorid
```

```
)pid on pid.ProductID=product.ProductID
```

First Customer's Vendor's Product :

use OnlineShopping;

```
Select CustomerName, VendorName, ProductName  
From Product
```

```
inner join (  
    Select CustomerName, Name as VendorName, ProductID  
    From Vendor
```

```
inner join (  
    Select CustomerName, VendorID, ProductID  
    From VendorProduct
```

```
inner join (  
    Select CustomerName, VendorProductID  
    From OrderedProduct
```

```
inner join (  
Select FirstName as CustomerName, OrderID From Customer
```

```
inner join (  
    Select CustomerID, OrderID  
    From Orders
```

```
        Where OrderDate = (Select min(OrderDate)  
From Orders)
```

```

) cid on cid.CustomerID = Customer.CustomerID
) opid on opid.OrderID = OrderedProduct.OrderedProductID
) vpid on vpid.VendorProductID = VendorProduct.VendorProductID
) vid on vid.VendorID = Vendor.VendorID
) pid on pid.ProductID = Product.ProductID

```

Customer's with empty cart :

```
use OnlineShopping
```

```
select * from Customer
where Customer.CustomerID NOT IN (select CustomerID from Cart)
```

Customers who never ordered anything :

```
use OnlineShopping
```

```
select * from Customer
where Customer.CustomerID NOT IN (select CustomerID from Orders)
```

Full Address with most orders with customer :

```
use OnlineShopping;
```

```
Select FirstName, LastName, HouseNo, Street, Area, CityName
From City
```

```
inner join (
    Select FirstName, LastName, HouseNo, Street, Area, CityID
    From ZipCode
```

```
inner join (
    Select FirstName, LastName, HouseNo, Street, Area, ZipCodeID
    From Customer
```

```
inner join (
    Select CustomerID, HouseNo, Street, Area, ZipCodeID
    From Address
```

```
inner join (
    Select TOP 1 AddressID
    From Orders
```

```

Where AddressID = (Select max(AddressID) From
Orders)
) adid on adid.AddressID = Address.AddressID
) cid on cid.CustomerID = Customer.CustomerID
) zcid on zcid.ZipCodeID = ZipCode.ZipCodeID
) ctid on ctid.CityID = City.CityID

```

Average age of customers :

use OnlineShopping;

```

select sum(DATEDIFF ( year , customer.DOB , getdate() ))/Count(customerid)
as "Average Age"
from Customer

```

Highest value Order :

use OnlineShopping

```

select Top 1 FirstName, LastName, Contact, Name, ProductName, Qty, Total
from Customer
inner join(
    select Name, ProductName, CustomerID, Qty, Total
    from Orders
    inner join(
        select Name, ProductName, OrderID, Qty, Total
        from Product
        inner join(
            select Name, ProductID, OrderID, Qty, Total
            from Vendor
            inner join(
                select VendorID, ProductID, OrderID, Qty,
Qty*Price as Total
                from VendorProduct
                inner join(
                    select VendorProductID, OrderID,
Quantity as Qty
                    from OrderedProduct)
                vpoid on vpoid.VendorProductID=VendorProduct.VendorProduct.
                voi on voi.VendorID=Vendor.VendorID)
            tp on tp.ProductID=Product.ProductID)
            cip on cip.OrderID=Orders.OrderID)
        cq on cq.CustomerID=Customer.CustomerID

```

Order by Total Desc

Late or Slow Delivery Order's vendors courier products:

use OnlineShopping

```
select "Vendor Name","Vendor Contact", "Vendor Email","Courier Name",
"Courier Contact", TrackingID, CityName, OrderDate, ProductName, Price,
Rating, Comment
from Product
inner join (
    select Rating, Comment, CityName, OrderDate, TrackingID, "Ven-
dor Name", "Vendor Contact", "Vendor Email", "Courier Name", "Courier
Contact", ProductID, Price
    from City
    inner join(
        select Rating, Comment, CityID, OrderDate, TrackingID, "Vendor
Name", "Vendor Contact", "Vendor Email", "Courier Name", "Courier Con-
tact", ProductID, Price
        from ZipCode
        inner join(
            select Rating, Comment, ZipCodeID,
OrderDate, TrackingID, "Vendor Name", "Vendor Contact", "Vendor Email",
"Courier Name", "Courier Contact", ProductID, Price
            from Address
            inner join(
                select Rating, Comment, AddressID,
OrderDate, TrackingID, "Vendor Name", "Vendor Contact", "Vendor Email",
Courier.Name as "Courier Name", Courier.Contact as "Courier Contact", Pro-
ductID, Price
                from Courier
                inner join (
                    select Rating, Comment, AddressID, OrderDate,
TrackingID, Name as "Vendor Name", Contact as "Vendor Contact", Email as
"Vendor Email", CourierID, ProductID, Price
                    from Vendor
                    inner join(
                        select Rating, Comment,
AddressID, OrderDate, TrackingID, VendorID, CourierID, ProductID, Price
                        from VendorProduct
                        inner join(
```

```

                                select Rating,
Comment, AddressID, OrderDate, TrackingID, CourierID, VendorProductID
                                from VendorCourier
                                inner join(
                                select Rating, Comment,
AddressID, OrderDate, TrackingID, VendorCourierID, VendorProductID
                                from Orders
                                inner join(
                                select
Rating, Comment, OrderID, VendorProductID
                                from
OrderedProduct
                                inner
join(
                                select
Rating, Comment, OrderedProductID
                                from
Review
                                where
Comment like '%late%deliver%' OR
                                Comment
like '%slow%deliver%' OR
                                Comment
like '%deliver%late%' OR
                                Comment
like '%deliver%slow%'
                                )
                                rc
on rc.OrderedProductID = OrderedProduct.OrderedProductID
                                )
od on
od.OrderID = Orders.OrderID
                                )
vcid on
vcid.VendorCourierID = VendorCourier.VendorCourierID
                                )
                                vpn on vpn.VendorProductID
= VendorProduct.VendorProductID
                                )
vcn on vcn.VendorID =
Vendor.VendorID
                                )
                                cd on cd.CourierID = Courier.CourierID
                                )
                                aid on aid.AddressID = Address.AddressID
                                )
                                lid on lid.ZipCodeID = ZipCode.ZipCodeID
                                )

```

```

        ccid on ccid.CityID = City.CityID
    )
    Pna on pna.ProductID = Product.ProductID

```

Preferred courier by all vendors :

use OnlineShopping

```

select Courier.Name,Contact,No_of_Orders
from Courier
inner join(
select Top 1 CourierID,count(courierID) as No_of_Orders
from VendorCourier

```

```

inner join(
select VendorCourierID
from Orders) vcid on vcid.VendorCourierID=VendorCourier.CourierID group
by CourierID order by No_of_Orders desc)
cn on cn.CourierID=Courier.CourierID

```

Same vendor courier for any customer :

use OnlineShopping;

```

Select FirstName, LastName, VendorName, Name as CourierName
From Courier

```

```

    inner join (
        Select FirstName, LastName, Name as VendorName, CourierID
        From Vendor

```

```

        inner join (
            Select FirstName, LastName, VendorID, CourierID
            From VendorCourier

```

```

            inner join (
                Select FirstName, LastName, VendorCourierID
                From Customer

```

```

                inner join (
                    Select CustomerID, VendorCourierID
                    From Orders
                Where VendorCourierID = (Select max(VendorCourierID)
From Orders)
            ) cid on cid.CustomerID = Customer.CustomerID

```


) vcid on vcid.VendorCourierID = VendorCourier.VendorCourierID

) vid on vid.VendorID = Vendor.VendorID

) crid on crid.CourierID = Courier.CourierID

Top 3 Cities as per no. of orders :

use OnlineShopping

```
select Top 3 City.CityName,"no of Orders"
from City inner join(
    select ZipCode.CityID,"no of Orders"
    from ZipCode inner join
        (select Address.ZipCodeID,count(Address.ZipCodeID)
as "no of Orders"
        from Address inner join
            (select AddressID
            from Orders) ad
            on ad.AddressID=Address.AddressID
            group by Address.ZipCodeID)
        zpc on zpc.ZipCodeID=ZipCode.ZipCodeID)
    ck on ck.CityID=city.CityID

order by "no of Orders" desc
```

Top 3 most selling products :

use OnlineShopping

```
select Name,Product.ProductName,OrderedQuantity
from Product
```

```
inner join(
select vendor.Name,ProductID,OrderedQuantity
from Vendor
```

```
inner join(
```

```
select vendorID,ProductID,OrderedQuantity
from VendorProduct
inner join (
select TOP 3 OrderedProduct.VendorProductID,sum(Quantity) as OrderedQuan-
tity
```

```

from OrderedProduct group by VendorProductID order by orderedquantity
desc) vpfk on vpfk.VendorProductID=VendorProduct.VendorProductID)
vpid on vpid.VendorID=vendor.VendorID)
vpvp on vvpv.ProductID=Product.ProductID

```

Top 3 products :

use OnlineShopping

```

Select Name, Product.ProductName, OrderedQuantity
From Product

```

```

inner join (
Select Vendor.Name, ProductID, OrderedQuantity
From Vendor

```

```

inner join (
Select vendorID, ProductID, OrderedQuantity
From VendorProduct

```

```

inner join (
Select TOP 3 OrderedProduct.VendorProductID, sum(Quantity) as OrderedQuan-
tity
From OrderedProduct Group By VendorProductID Order By orderedquantity
Desc) vpfk on vpfk.VendorProductID = VendorProduct.VendorProductID)
vpid on vpid.VendorID = vendor.VendorID)
vpvp on vvpv.ProductID = Product.ProductID

```

Top 3 Vendors with worst rating :

use OnlineShopping

```

select TOP 3 Name,Contact,Email,Total_bad_ratings
from Vendor
inner join(
select VendorID,COUNT(rating) as Total_bad_ratings
from VendorProduct
inner join(
select Rating,Comment,OrderedProduct.VendorProductID
from OrderedProduct
inner join(
select Rating,Comment,OrderedProductID
from Review
where Rating<3) r on r.OrderedProductID=OrderedProduct.OrderedProductID)

```

```

vr on vr.VendorProductID=VendorProduct.VendorProductID group by VendorID)
vrt on vrt.VendorID=Vendor.VendorID
Order by Total_bad_ratings desc

```

Total amount of purchase done by customer :

```

use OnlineShopping
select SUM(Price*Qty)as Total_Purchase
from(
select VendorProduct.Price as price,Qty
from VendorProduct
inner join(
select Quantity as Qty,VendorProductID
from OrderedProduct
inner join(
select OrderID
from Orders
where CustomerID='60') op on op.OrderID=OrderedProduct.OrderedProductID)
pp on pp.VendorProductID=VendorProduct.VendorProductID) as ta

```

Total sale done by a vendor between given date range :

```

use OnlineShopping

```

```

select sum(Price) as "Total Sale"
from VendorProduct
inner join(
select VendorProductID
from OrderedProduct
inner join(
select OrderID as oid
from Orders
where OrderDate Between '2013-07-26' And '2014-02-21') ordi on ordi.oid=OrderedProduct.
on vpid.VendorProductID=VendorProduct.VendorProductID
where VendorID='3'

```

Vendors with products having rating 5 :

```

use OnlineShopping;

```

```

Select Name as VendorName, ProductName, Rating
From Vendor

```

```

inner join (
    Select ProductName, VendorID, Rating
    From Product

    inner join (
        Select ProductID, VendorID, Rating
        From VendorProduct

        inner join (
            Select VendorProductID, Rating
            From OrderedProduct

            inner join (
                Select OrderedProductID, Rating
                From Review Where Rating = 5
            ) opid on opid.OrderedProductID = OrderedProd-
uct.OrderedProductID

        ) vpid on vpid.VendorProductID = VendorProduct.VendorProductID

    ) pid on pid.ProductID = Product.ProductID

) vid on vid.VendorID = Vendor.VendorID

```

Order By VendorName, ProductName

Vendors who sell top 3 most sold products :

use OnlineShopping

```

select Name,Contact,email,ProductName,Price
from Vendor

```

```

inner join(
select Product.ProductName,VendorID,Price
from Product
inner join(
select ProductID,VendorID,Price
from VendorProduct
inner join (
select TOP 3 OrderedProduct.VendorProductID,sum(Quantity) as OrderedQuan-
tity
from OrderedProduct group by VendorProductID order by orderedquantity
desc) vpfk on vpfk.VendorProductID=VendorProduct.ProductID
) pro on pro.ProductID=Product.ProductID) vna on vna.VendorID=Vendor.VendorID

```

order by Price