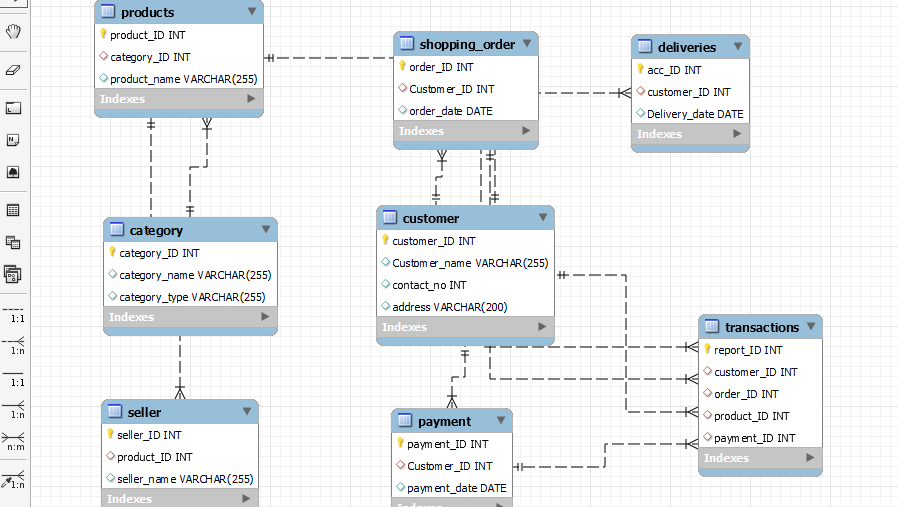
**Assignment 2**

**1. For an online purchasing DB, create entity relationship diagrams. Create a DB object from your entity diagram.**



create database online\_shopping

use online\_shopping

Create Table Customer

(

customer\_ID Int not null,

Customer\_name varchar(255),

contact\_no Int,

address varchar(200),

primary key(customer\_ID)

)

Create table category

(

category\_ID Int not null,

category\_name Varchar(255),

category\_type Varchar(255),

primary key(category\_ID)

)

Create table Shopping\_Order

(

order\_ID Int not null,

Customer\_ID Int,

order\_date Date,

primary key(order\_ID),

foreign key (Customer\_ID) references Customer(Customer\_ID)

)

Create table Deliveries

(

acc\_ID Int not null,

customer\_ID Int,

Delivery\_date Date,

Primary key(acc\_ID),

Foreign key(customer\_ID) references customer(customer\_ID)

)

Create Table Products

(

product\_ID Int,

category\_ID Int,

product\_name Varchar(255),

primary key(product\_ID),

foreign key(category\_ID) references category(category\_ID)

)

Create Table Seller

(

seller\_ID Int not null,

product\_ID Int,

seller\_name Varchar(255),

primary key(seller\_ID),

foreign key(product\_ID) references products(product\_ID)

)

Create Table Payment

(

payment\_ID Int not null,

Customer\_ID int,

payment\_date Date,

primary key(payment\_ID),

foreign key(Customer\_ID) references Customer(Customer\_ID)

)

Create Table Transactions

(

report\_ID int not null,

customer\_ID int,

order\_ID int,

product\_ID int,

payment\_ID int,

primary key(report\_ID),

foreign key(Customer\_ID) references Customer(Customer\_ID),

foreign key(order\_ID) references shopping\_order(order\_ID),

foreign key(product\_ID) references products(product\_ID),

foreign key(payment\_ID) references payment(payment\_ID)

)

**2. Create a sql store process to register the use of the database, complete it with proper validation and transaction rollback and commit?**

Create table store

(

Id int not null.

Name varchar(100) not null,

Number int(10) not null,

)

Insert into store(1,Ayesha,5684255846)

Insert into store(2,kala,45862157995)

Select \* from store

Rollback;

Begin transaction

Insert into store(3,nalan,588674759)

Select \*from store

Rollback;

Begin transaction

Insert into store(3,nalan,2854556996)

Commit;

**3. List the sql aggregate function and demonstrate how to utilize it.**

Count() – Select count(\*) from emp where gendee =’F’

Max() – Select max(salary) from emp

Avg() – Select avg(salary) from emp

Sum() – Select sum(salary) as total from emp

**4. In sql, create a pivot query?**

CREATE TABLE exams (

id int(11) NOT NULL auto\_increment,

name varchar(15),

exam int,

score int,

PRIMARY KEY (id)

);

insert into exams (name,exam,score) values ('Bob',1,70);

insert into exams (name,exam,score) values ('Bob',2,77);

insert into exams (name,exam,score) values ('Bob',3,71);

insert into exams (name,exam,score) values ('Bob',4,70);

insert into exams (name,exam,score) values ('Sue',1,89);

insert into exams (name,exam,score) values ('Sue',2,87);

insert into exams (name,exam,score) values ('Sue',3,88);

insert into exams (name,exam,score) values ('Sue',4,89);

SELECT name,

sum(IF(exam=1, score, NULL)) AS exam1,

sum(IF(exam=2, score, NULL)) AS exam2,

sum(IF(exam=3, score, NULL)) AS exam3,

sum(IF(exam=4, score, NULL)) AS exam4

FROM exams

GROUP BY name;

**4. with an example, describe how to join in sql?**

create database student;

use student

drop table dep

create table dep

(

depid int not null,

depname varchar(100),

dep\_add varchar(100),

primary key(depid)

)

drop table emp

create table emp

(

empid int,

empname varchar(100),

emp\_add varchar(100),

depid int,

joindate date,

salary int,

gender varchar(1),

primary key(empid)

)

Insert into dep value(1,'Analyst','London');

Insert into dep value(2,'Software Developer','India');

Insert into dep value(3,'Programmer','China');

Insert into dep value(4,'Data scientist','Russia');

Insert into dep value(5,'Fornt end developer','Singapore');

insert into emp values(1,'john weber','UK',1,'2020/11/20',32000,'F');

insert into emp values(2,'Potsy weber','USA',1,'2004/12/22',45000,'M');

insert into emp values(3,'Dirk smith','India',3,'2011/10/19',34000,'F');

insert into emp values(4,'Johny','UK',2,'2020/02/18',52000,'F');

insert into emp values(5,'Warrior','Ruissia',2,'2021/12/20',27000,'F');

insert into emp values(6,'Warrior','Ruissia',7,'2015/01/21',42000,'M');

insert into emp values(7,'Johny','UK',1,'2012/09/20',37000,'F');

**Left join**

select \* from emp e left join dep d

ON e.depid = d.depid

**Right join**

select \* from emp e right join dep d

ON e.depid = d.depid

**Cross join**

select \* from dep d cross join emp e

ON e.depid = d.depid

**Inner join**

select \* from dep d left outer join emp e

ON e.depid = d.depid

UNION

select \* from dep d right outer join emp e

ON e.depid = d.depid

6. How to locate the 4th highest value in a column in a row. Create you table

create table emp

(

empid int,

empname varchar(100),

emp\_add varchar(100),

depid int,

joindate date,

salary int,

gender varchar(1),

primary key(empid)

)

insert into emp values(1,'john weber','UK',1,'2020/11/20',32000,'F');

insert into emp values(2,'Potsy weber','USA',1,'2004/12/22',45000,'M');

insert into emp values(3,'Dirk smith','India',3,'2011/10/19',34000,'F');

insert into emp values(4,'Johny','UK',2,'2020/02/18',52000,'F');

insert into emp values(5,'Warrior','Ruissia',2,'2021/12/20',27000,'F');

insert into emp values(6,'Warrior','Ruissia',7,'2015/01/21',42000,'M');

insert into emp values(7,'Johny','UK',1,'2012/09/20',37000,'F');

select top 1 salary from(select top 4 salary from emp order by salary desc) order by salary asc