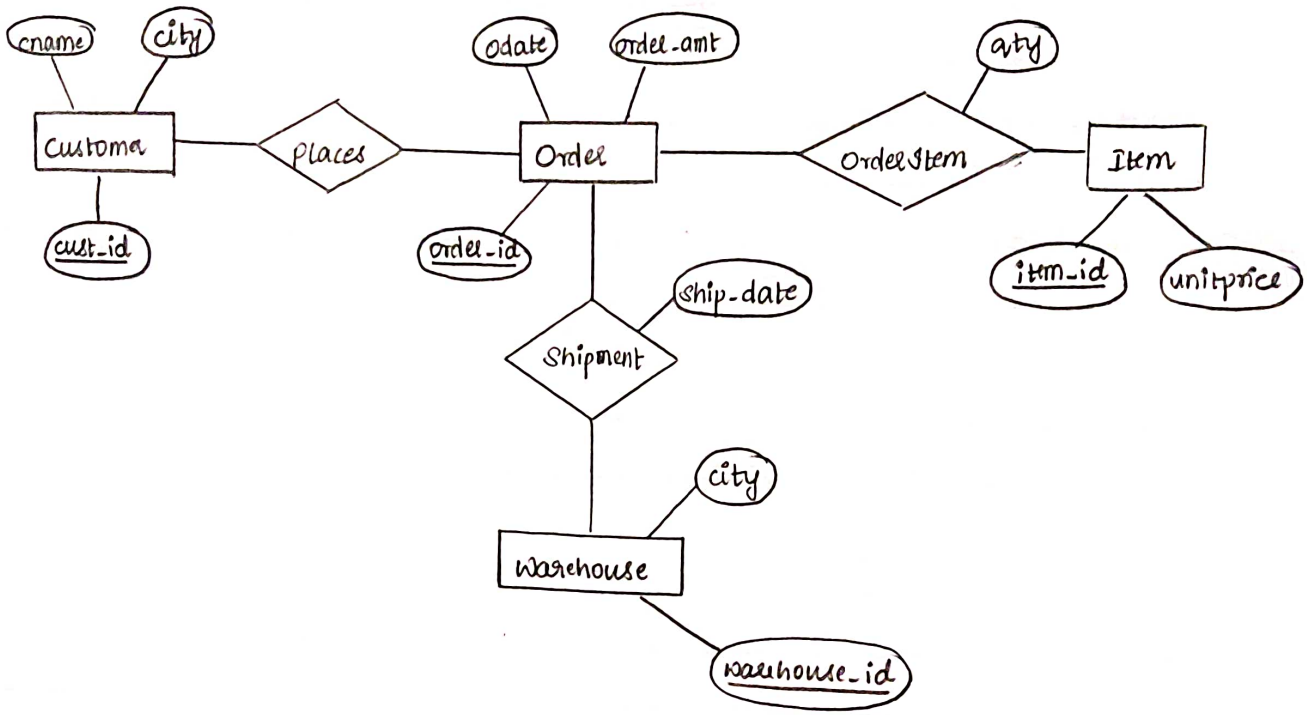
ER diagram :


cname
city
Customer
cust_id
places
odate
order_amt
Order
order_id
ship_date
Shipment
city
Warehouse
warehouse_id
qty
OrderItem
Item
item_id
unitprice

Name of Experiment :..... Date : 7/12/24 Page No. 26

Experiment No : 05 Experiment Result :.....

3. 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 customers (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 ;
```

4. Find the item with the maximum unit price.

```
select max(unitprice)
from Items ;
```

5. Create a view to display orderid and shipment date of all orders shipped from a warehouse 2.

```
create view shipmentDateFromWarehouse2 as
select order_id, shipdate
from shipments
where warehouse_id = 2 ;
select * from shipmentDateFromWarehouse2 ;
```

6. A view that shows the warehouse ids from where the kumar's orders are being shipped.

```
create view WarehouseWithKumarOrders as
select s.warehouse_id
from warehouse w, customers c, Orders o, shipment s
where w.warehouseid = s.warehouse_id and s.order_id = o.order_id and
      o.cust_id = c.cust_id and c.cname = "kumar" ;
select * from WarehouseWithKumarOrders ;
```

Teacher's Signature : _____

Name of Experiment : ..... Date : 7/12/24

Experiment No : 05 Experiment Result : .....

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

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

28/12/24

Teacher's Signature : _____

Name of Experiment : Enrollment database Date : 28/12/24

Experiment No : 06 Experiment Result :

D. Student enrollment in courses and books adopted for each course.

Student (regno : string, name : string, major : string, bdate : date)

course (course# : int, cname : string, dept : string)

Enroll (regno : string, course# : int, sem : int, marks : int)

Book_adoption (course# : int, sem : int, book_isbn : int)

Text : book (book_isbn : int, book_title : string, publisher : string, author : string)

```
create database enrollment ;
use enrollment ;
create table student (
    regno varchar(13) primary key,
    name varchar(25) not null,
    major varchar(25) not null,
    bdate date not null
);

create table course (
    course int primary key,
    cname varchar(30) not null,
    dept varchar(100) not null
);

create table Enroll (
    regno varchar(13),
    course int,
    sem int not null,
    marks int not null,
    foreign key(regno) references student(regno) on update cascade on delete
cascade,
```

Teacher's Signature : _____

Output :

Student table :

regno	name	major	bdate
01HF234	student_5	Computer Economics	2001-10-10
01HF235	student_1	CSE	2001-05-15
01HF254	student_2	Philosophy	2000-04-04
01HF354	student_3	Literature	2002-06-10
01HF653	student_4	History	2003-10-12

Course table :

course	cname	dept
1	DBMS	Computer Science
2	Literature	English
3	Philosophy	Philosophy
4	History	Social science
5	Computer Economics	Computer Science

Enroll table :

regno	course	sem	marks
01HF235	1	5	85
01HF354	2	6	87
01HF254	3	3	95
01HF653	4	3	80
01HF234	5	5	75

```
    foreign key(course) references Course (course) on update cascade on delete
    cascade
);

create table TextBook (
    bookISBN int primary key,
    book-title varchar(40) not null,
    publisher varchar(25) not null,
    author varchar(25) not null
);

create table BookAdoption(
    course int ;
    sem int not null,
    bookISBN int,
    foreign key(bookISBN) references TextBook (bookISBN) on update cascade on delete
                                                        cascade,
    foreign key(course) references Course (course) on update cascade on delete cascade
);

Insert into student values
("01HF235", "student-1", "CSE", "2001-05-15"),
("01HF354", "student-2", "Literature", "2002-06-10"),
("01HF254", "student-3", "Philosophy", "2000-04-04"),
("01HF653", "student-4", "History", "2003-10-12"),
("01HF234", "student-5", "Computer Economics", "2001-10-10");

Insert into Courese values
```

Teacher's Signature : _____

Book Adoption table :

course	sem	booksbn
1	5	241563
2	6	532678
3	3	453723
4	3	278345
1	6	426784

TextBook table :

booksbn	book_title	publisher	author
241563	Operating System	Pearson	Silberschatz
278345	History of the world	The times	Richard Overy
426784	Behavioural Economics	Pearson	David Orrel
453723	Immanuel kant	Delhi classics	Immanuel kant
532678	Complete work of shakespheee	Oxford	Shakespheee

1. TextBook table :

booksbn	book_title	Publisher	author
123456	Wings of Fire	Pearson	Dr APJ Abdul kalam
241563	Operating System	Pearson	Silberschatz
278345	History of the world	The times	Richard Overy
426784	Behavioural Economics	Pearson	David Orrel
453723	Immanuel kant	Delhi classics	Immanuel kant
532678	Complete work of shakespheee	Oxford	Shakespheee

Name of Experiment :..... Date : 28/12/24 Page No. 30

Experiment No : 06 Experiment Result :.....

(001, "DBMS", "CS"),
(002, "Literature", "English"),
(003, "Philosophy", "Philosophy"),
(004, "History", "Social Science"),
(005, "Computer Economics", "CS");

Insert into Enroll values
("01HF235", 001, 5, 85),
("01HF354", 002, 6, 87),
("01HF254", 003, 3, 95),
("01HF653", 004, 3, 80),
("01HF234", 005, 5, 75);

Insert into TextBook values
(241563, "Operating System", "Pearson", "Silberschatz"),
(532678, "complete works of shakespheare", "Oxford", "Shakespheare"),
(453723, "Immanuel kant", "Delhi classics", "Immanuel kant"),
(278345, "History of the world", "The times", "Richard Overy"),
(426784, "Behavioural Economics", "Pearson", "David Orrel");

Insert into BookAdoption values
(001, 5, 241563),
(002, 6, 532678),
(003, 3, 453723),
(004, 3, 278345),
(001, 6, 426784);

select * from student ;

Teacher's Signature : _____

BookAdoption table :

course	sem	booksisbn
1	5	241563
2	6	532678
3	3	453723
4	3	278345
1	6	426784
1	5	123456

2.

course	booksisbn	book_title
1	426784	Behavioural Economics
1	241563	Operating System
1	123456	Wings of Fire

3.

dept
Computer Science

4.

name
student_1

5.

cname	marks
DBMS	85

6.

regno	course	sem	marks
01HF235	1	5	85

Name of Experiment :..... Date : 28/12/24

Experiment No : 06 Experiment Result :.....


```
select * from Course ;
select * from Enroll;
select * from BookAdoption ;
select * from TextBook;
```

1. Demonstrate how you add a new text book to the database and make this book be adopted by some department.

```
insert into TextBook values
(123456, "Wings Of Fire", "Pearson", "Dr APJ Abdul kalam");
insert into BookAdoption values
(001, 5, 123456);
```

2. Produce a list of text books (include course#, Book-ISBN, Book-title) in the alphabetical order for courses offed by the 'cs' department that use more than two books.

```
select c.course , b.bookisbn , b.book-title
from course c, BookAdoption ba, TextBook b
where c.course = ba.course AND ba.book isbn = b.book isbn AND c.dept = 'cs'
      AND 2 < (select count(bookisbn)
               from BookAdoption b
               where c.course = b.course) order by b.book-title;
```

3. List any department that has all its adopted books published by a specific publisher.

```
select distinct c.dept
from course c
where c.dept in (select c.dept
                 from course c, BookAdoption b, TextBook b
```

Teacher's Signature : _____

7.

cname	book-title
DBMS	Operating System
Literature	Complete work of Shakespeare
Philosophy	Immanuel kant
History	History of the world
DBMS	Behavioural Economics
DBMS	Wings of Fire

8.

regno	course	sem	marks
01HF235	1	5	85
01HF254	3	3	95
01HF653	4	3	80
01HF234	5	5	75

9. Error 1644 (45000) : marks below threshold

```
where c.course = b.course AND t.bookISBN = b.bookISBN AND
t.publisher = 'PEARSON') AND c.dept not in
(select c.dept
from course c, BookAdoption b,
TextBook t
where c.course = b.course AND
t.bookISBN = b.bookISBN AND
t.publisher != 'PEARSON');
```

4. List the students who have scored maximum marks in 'DBMS' course

```
select name
from student s, Enroll e, course c
where s.regno = e.regno AND e.course = c.course AND cname = 'DBMS' AND
e.marks in (select max(marks)
from Enroll e1, Course c1
where c1.name = 'DBMS' and c1.course = e1.course);
```

5. Create a view to display all the courses opted by a student along with marks obtained

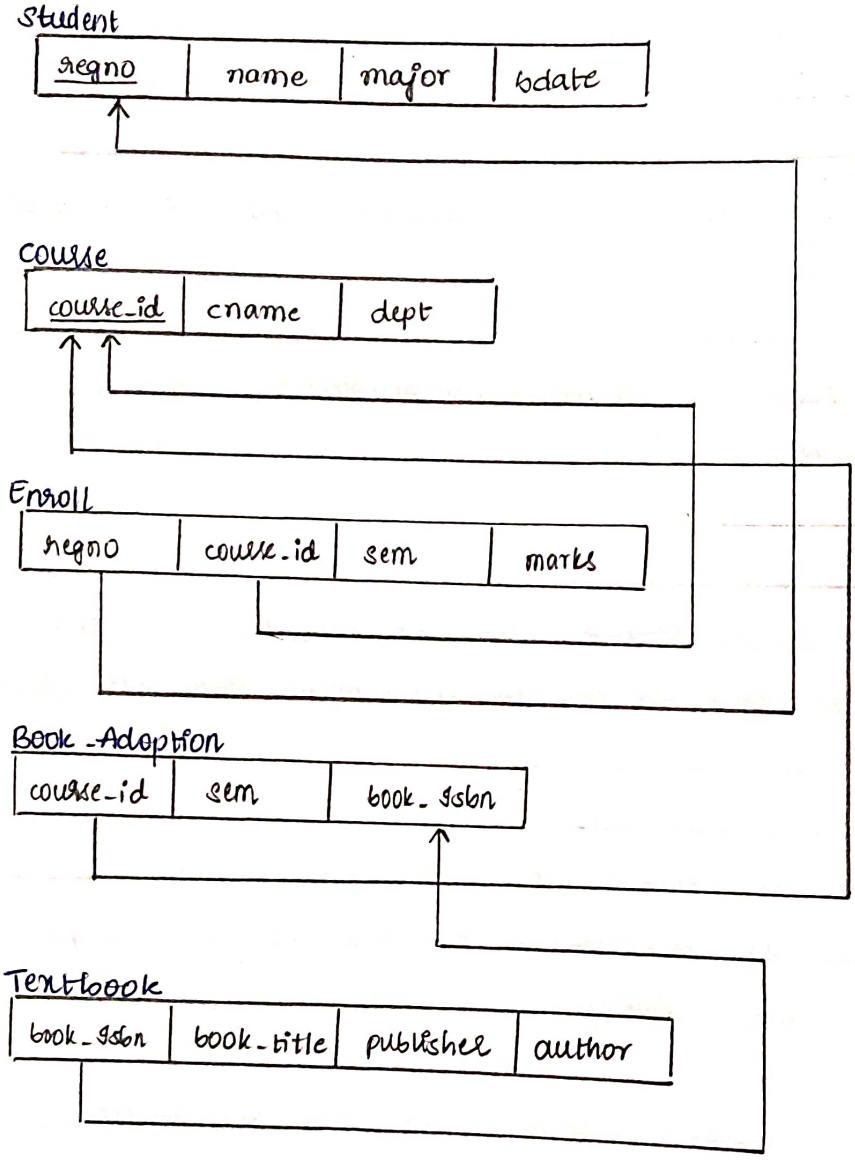
```
create view CoursesOptedByStudent as
select c.cname, e.marks
from course c, Enroll e
where e.course = c.course AND e.regno = "01FE20235";
select * from CoursesOptedByStudent;
```

6. Create a view to show the enrolled details of a student.

```
create view StudentEnrollmentDetails as
select * from Enroll
```

Teacher's Signature : _____

schema diagram :



```
where regno = "01HF235";
select * from StudentEnrollmentDetails;
```

7. Create a view to display course related books from course adoption and textbooks table using bookisbn

```
create view CourseRelatedBooks as
select cname, book_title
from Course c, TextBook tb, BookAdoption ba
where c.course = ba.course and tb.bookisbn = ba.bookisbn;
select * from CourseRelatedBooks;
```

8. Create a trigger such that it deletes all records from enroll table when course is deleted.

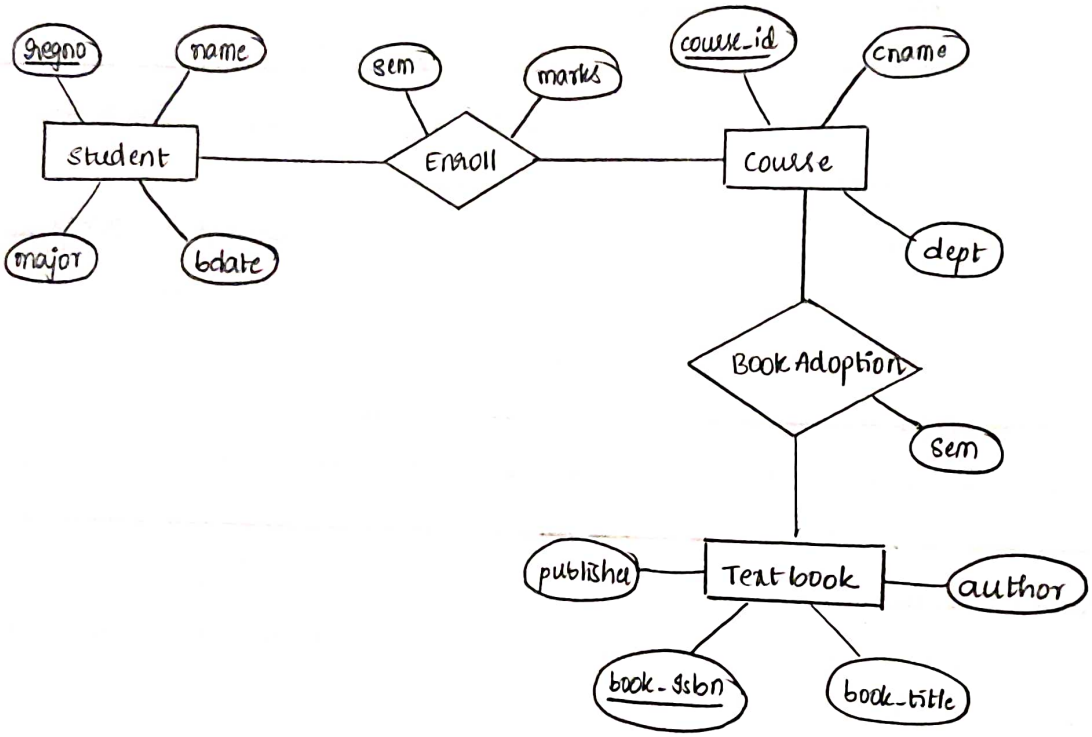
```
DELIMITER //
create trigger DeleteRecords
after delete on Course
for each row
BEGIN
    delete from Enroll where Enroll.course = OLD.course;
END; //
DELIMITER ;
delete from course where course = 2;
```

9. Create a trigger that prevents a student from enrolling in a course if the marks pre-requisit is less than the given threshold

```
DELIMITER //
create trigger PreventEnrollment
before insert on Enroll
```

Teacher's Signature : _____

ER diagram :


regno
name
student
major
bdate
sem
marks
Enroll
course_id
cname
Course
dept
Book Adoption
sem
publisher
Text book
author
book_isbn
book_title

Name of Experiment :..... Date :..28/12/24.....

Experiment No :..06..... Experiment Result :.....


```
for each row
BEGIN
    IF (new.marks <10) THEN
        signal sqlstate '45000' set message_text = 'marks below threshold';
    END IF;
END; //
DELIMITER ;
insert into Enroll values ("01HF235", 002, 5, 5);
```

Teacher's Signature : _____

Name of Experiment : Company database Date : 11/01/2025 Page No. 35

Experiment No : 07 Experiment Result :

E. Company database :

EMPLOYEE (ssn, name, address, sex, salary, superssn, dno)

DEPARTMENT (Dno, Dname, Mgrssn, MgrStartDate)

DLocation (Dno, DLoc)

PROJECT (PNO, PName, PLocation, DNO)

WORKS-ON (ssn, PNo, Hours)

```
create database company;
use company;

create table if not exists Employee (
    ssn varchar(35) primary key,
    name varchar(35) not null,
    address varchar(255) not null,
    sex varchar(7) not null,
    salary int not null,
    super_ssn varchar(35),
    d_no int,
    foreign key(super_ssn) references Employee(ssn) on update cascade on delete
                                                    set null
);

create table if not exists Department (
    d_no int primary key,
    dname varchar(100) not null,
    mgr_ssn varchar(35),
    mgr_start_date date,
    foreign key(mgr_ssn) references Employee(ssn) on delete cascade on update
                                                  cascade
);
```

Teacher's Signature : _____

Output:

Department tables:

d_no	dname	mgr_ssn	mgr_start_date
1	Human Recources	01NB235	2020-10-21
2	Quality Assement	03NB653	2020-10-19
3	System Assessment	04NB234	2020-10-27
4	Accounts	01NB354	2020-09-04
5	Production	02NB254	2020-08-16

Employee tables:

ssn	name	address	sex	salary	super_ssn	d_no
01NB235	Chandan Krishna	Siddartha nagar, Mysuru	Male	1500000	01NB235	5
01NB354	Ananth	Lakshmipuram, Mysuru	Male	1200000	01NB235	2
02NB254	Lakshmi	Pune, Maharashtra	Female	1000000	01NB235	4
03NB653	Akash	Hyderabad, Telangana	Male	2500000	01NB354	5
04NB234	Isha	JP Nagar, Bengaluru	Female	1700000	01NB354	1

Dlocation table:

d_no	d_loc
1	Jayanagar, Bengaluru
2	Vijaynagar, Mysuru
3	Chennai, TamilNadu
4	Mumbai, Maharashtra
5	Kuvempunagar, Mysuru

Name of Experiment :..... Date :...11/01/2025.....

Experiment No :...07..... Experiment Result :.....


```
create table if not exists DLocation (
    d_no int not null,
    d_loc varchar(100) not null,
    foreign key(d_no) references Department(d_no) on delete cascade on
                                update cascade
);

create table if not exists Project (
    p_no int primary key,
    p_name varchar(25) not null,
    p_loc varchar(25) not null,
    d_no int not null,
    foreign key(d_no) references Department(d_no) on delete cascade on
                                update cascade
);

create table if not exists WorksOn (
    ssn varchar(35) not null,
    p_no int not null,
    hours int not null default 0,
    foreign key(ssn) referenced Employee(ssn) on delete cascade on update cascade,
    foreign key(p_no) references Project(p_no) on delete cascade on update cascade
);

insert into Employee values
("01NB235", "chandan krishna", "Siddartha Nagar, Mysuru", "male",
        1500000, "01NB235", 5);
("01NB354", "Ananth", "Lakshmipuram, Mysuru", "Male", 1200000,
```

Teacher's Signature : _____

Project table :

p-no	p-name	p-loc	d-no
241563	System Testing	Mumbai, Maharashtra	4
278345	Yield Increase	Kuvempunagar, Mysuru	5
426784	Product Refinement	Saraswathipuram, Mysuru	2
453723	Product Optimization	Hyderabad, Telangana	5
532678	IOT	JP Nagar, Bengaluru	1

WorksOn table :

ssn	p-no	hours
01NB235	278345	5
01NB354	426784	6
04NB234	532678	3
02NB254	241563	3
03NB653	453723	6

1.

p-no	p-name	name
278345	Yield Increase	Chandan Krishna
453723	Product Optimization	Chandan Krishna

2.

ssn	name	dd-salary	neu-salary
04NB234	Isha	1700000	1870000.0

3.

sal-sum	sal-max	sal-min	sal-avg
1000000	1000000	1000000	1000000.0000

"01NB235", 2),
("02NB254", "Lakshmi", "Pune, Maharashtra", "Female", 1000000,
"01NB235", 4),
("03NB653", "Akash", "Hyderabad, Telangana", "Male", 2500000,
"01NB354", 5),
("04NB234", "Asha", "JP Nagar, Bengaluru", "Female", 1700000,
"01NB354", 1);

insert into Department values
(001, "Human Resources", "01NB235", "2020-10-21"),
(002, "Quality Assessment", "03NB653", "2020-10-19"),
(003, "System Assessment", "04NB234", "2020-10-27"),
(005, "Production", "02NB254", "2020-08-16"),
(004, "Accounts", "01NB354", "2020-09-04");

insert into Dlocation value
(001, "Jayanagar, Bengaluru"),
(002, "Vijaynagar, Mysuru"),
(003, "Chennai, Tamil Nadu"),
(004, "Mumbai, Maharashtra"),
(005, "Kuvempunagar, Mysuru");

insert into Project value
(241563, "System Testing", "Mumbai, Maharashtra", 004),
(532678, "IOT", "JP Nagar, Bengaluru", 001),
(453723, "Product Optimization", "Hyderabad, Telangana", 005),
(278345, "Yield increase", "Kuvempunagar, Mysuru", 005),
(426784, "Product Refinement", "Saraswatipuram, Mysuru", 002);

Teacher's Signature : _____

4.

ssn	name	d-no
04NB234	Isha	1

5.

d-no	count(*)
5	2

6.

name	dname	d_loc
Chandan Krishna	Production	Kuvempunagar, Mysuru
Ananth	Quality Assessment	Vijaynagar, Mysuru
Lakshmi	Accounts	Mumbai, Maharashtra
Akash	Production	Kuvempunagar, Mysuru
Isha	Human Resources	Jayanagar, Bengaluru

7.

p_name	p_loc	dname
System testing	Mumbai, Maharashtra	Accounts
Yield Increase	Kuvempunagar, Mysuru	Production
Product Refinement	Saraswathipuram, Mysuru	Quality Assessment
Product Optimization	Hyderabad, Telangana	Production
IOT	JP Nagar, Bengaluru	Human Resources

Name of Experiment :..... Date :..11/01/2025..... Page No. 38

Experiment No :..07..... Experiment Result :.....

```
insert into WorksOn values
("01NB235", 278345, 5),
("01NB354", 426784, 6),
("04NB234", 532678, 3),
("02NB254", 241563, 3),
("03NB653", 453723, 6);
alter table Employee add constraint foreign key (d_no) references Department
(d_no) on delete cascade on update cascade;

select * from Department;
select * from Employee;
select * from DLocation;
select * from Project;
select * from WorksOn;
```

1. Make a list of all project numbers for projects that involve an employee whose last name is ~~'Scott'~~ 'krishna', either as a worker or as a manager of the department that controls the project

```
select p_no, p_name, name
from Project p, Employee e
where p.d_no = e.d_no and e.name like "%krishna";
```

2. Show the resulting salaries if every employee working on the 'IOT' project is given a 10 percent raise.

```
select w.ssn, name, salary as old_salary, salary * 1.1 as new_salary
from WorksOn w join Employee e
where w.ssn = e.ssn and w.p_no = (select p_no
                                  from Project
                                  where p_name = 'IOT');
```

Teacher's Signature : _____

8.

d_no	d_name	mgr_ssn	mgr_start_date
1	Human Resources	01NB235	2020-10-21
2	Quality Assessment	03NB653	2020-10-19
3	System Assessment	04NB234	2020-10-27
4	Accounts	02NB257	2020-08-16
5	Production	01NB354	2020-09-04
6	R & D	01NB354	2025-01-01

9. Error 1644(45000) : This project has an employee assigned.

3. Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary and the average salary in this department.

```
select sum(salary) as sal_sum, max(salary) as sal_max, min(salary) as sal_min,
       avg(salary) as sal_avg.
from Employee e join Department d on e.d_no = d.d_no
where d.dname = 'Accounts';
```

4. Retrieve the name of each employee who works on all the projects controlled by department number 1 (use not exists operator)

```
select Employee.ssn, name, d_no
from Employee
where not exists (select p_no
                  from Project p
                  where p.dno = 1 and p_no not in (select p_no
                                                   from workson wo
                                                   where wo.ssn = Employee.ssn));
```

5. For each department that has more than one employees, retrieve the department number and the number of its employees who are making more than Rs 6,00,000.

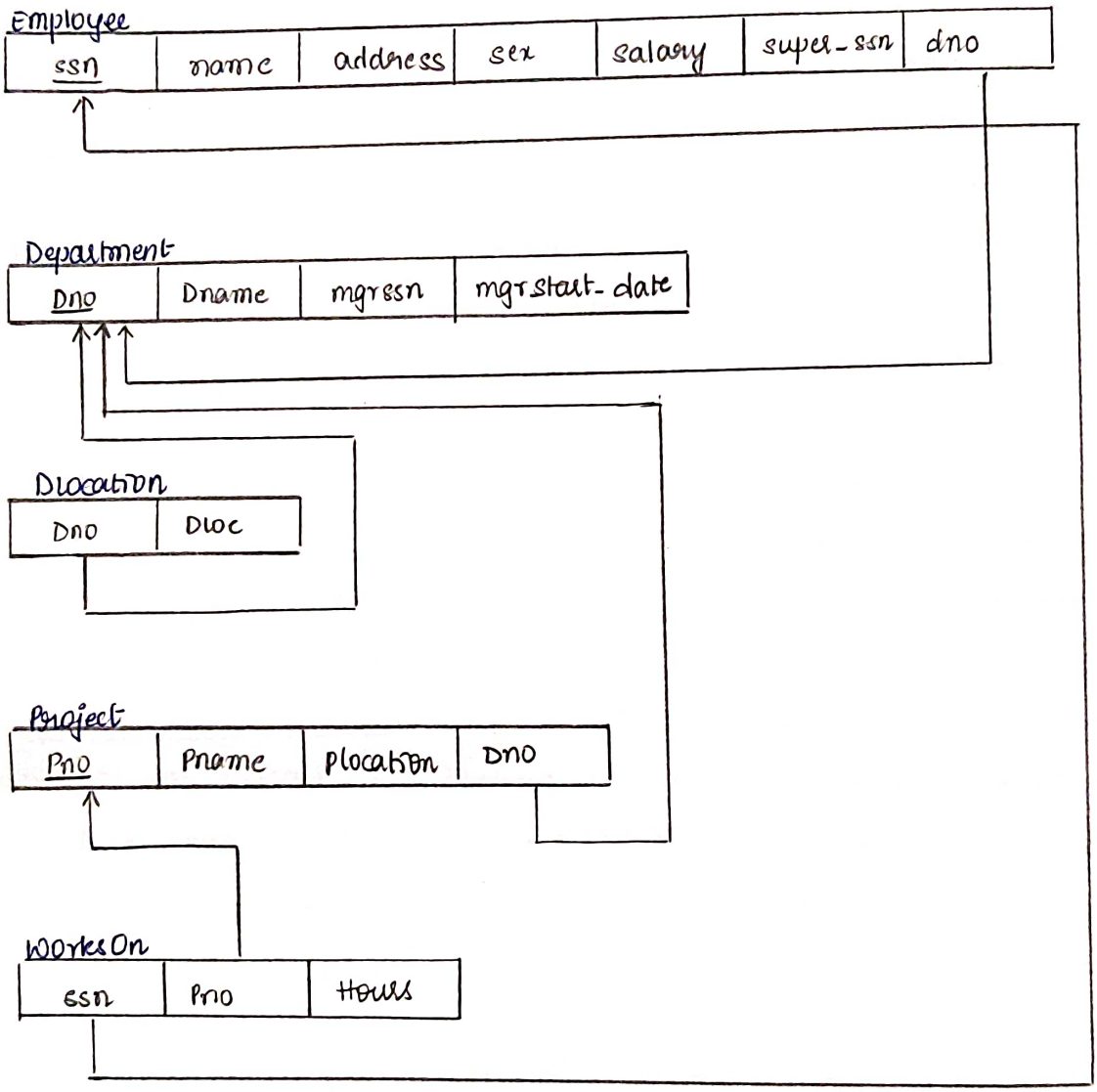
```
select d.d_no count (*)
from Department d join Employee e on e.d_no = d.d_no
where salary > 600000 group by d.d_no having count (*) > 1 ;
```

6. Create a view that shows name, dept name and location of all employees

```
create view emp_details as
select name, dname, d_loc
from Employee e, join Department d on e.d_no and = d.d_no join Dlocation d1
```

Teacher's Signature : _____

schema diagram :



```
    on d.d-no = d1.d-no;
select * from emp-details;
```

7. create a view that shows project name, location and department.

```
create view ProjectDetails as
select p-name, p-loc, dname
from Project p natural join Department d;
select * from ProjectDetails;
```

8. A trigger that automatically updates manager's start date when he is assigned.

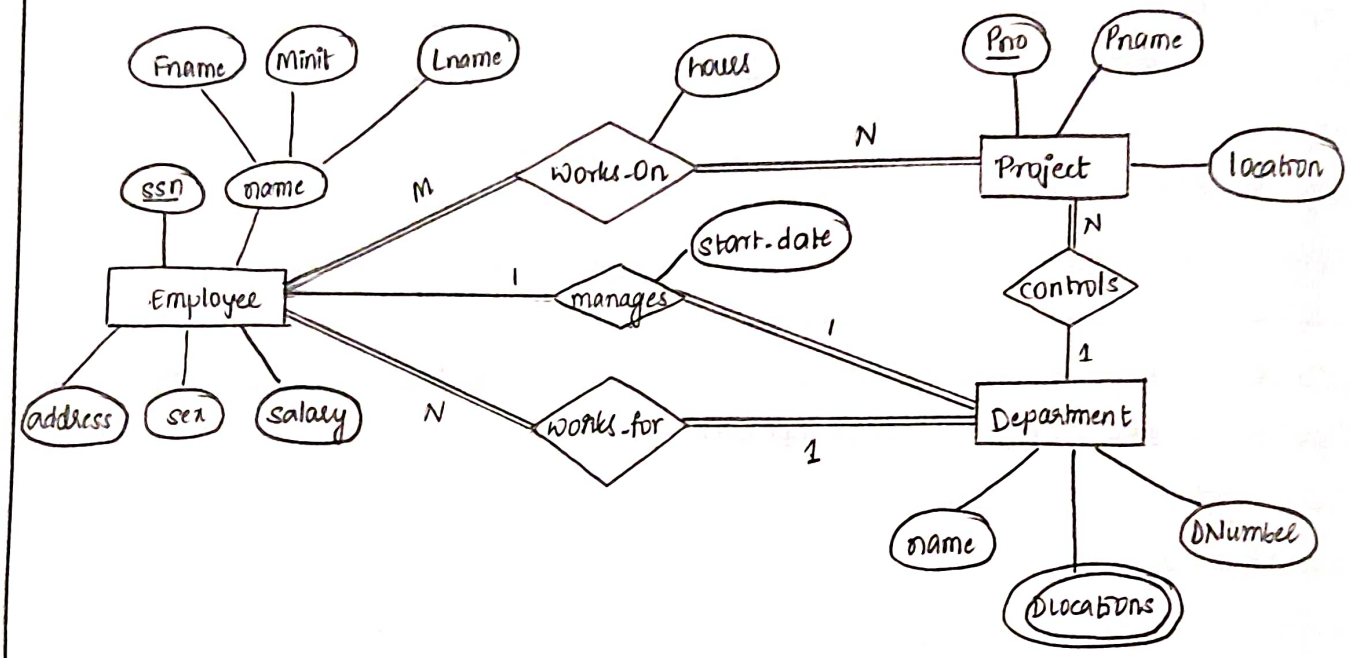
```
DELIMITER //
create trigger UpdateManagerstartDate
before insert on Department
for each row
BEGIN
    SET NEW.mgr-start-date = curdate();
END; //
DELIMITER ;
insert into Department (d-no, dname, mgr-ssn) values
(006, "R&D", "01NB354"); -This will automatically set mgr-start-date to today's date.
```

9. create a trigger that prevents a project from being deleted if it is currently being worked by any employee.

```
DELIMITER //
create trigger PreventDelete
before delete on Project
for each row
BEGIN
```

Teacher's Signature : _____

ER diagram :


Fname
Minit
Lname
ssn
name
hours
Works-On
M
N
Pno
Pname
Project
location
N
controls
1
start-date
manages
1
Employee
address
sex
salary
N
Works-for
1
Department
name
DNumber
Dlocations

Name of Experiment :..... Date : 11/01/2025

Experiment No : 07 Experiment Result :.....


```
IF EXISTS (select * from WorksOn where p_no = old.p_no) THEN
    signal sqlstate '45000' set message_text = 'This project has an employee
                                                assigned';
END IF;
END; //
DELIMITER ;
delete from Project where p_no = 241563; -- will give error.
```

Teacher's Signature : _____