DBMS 5TH SEM LAB

DATABASE: 01(LIBRARY DATA BASE)

CREATE TABLE

```
CREATE TABLE PUBLISHER
(NAME VARCHAR(20) PRIMARY KEY,
PHONE INTEGER,
ADDRESS VARCHAR(20));

DESC PUBLISHER;

CREATE TABLE BOOK
(BOOK_ID INTEGER PRIMARY KEY,
TITLE VARCHAR(20),
PUB_YEAR VARCHAR(20),
PUB_NAME VARCHAR(20),
FOREIGN KEY (PUB_NAME) REFERENCES PUBLISHER(NAME) ON DELETE CASCADE);


DESC BOOK;

CREATE TABLE BOOK_AUTHORS
(AUTHOR_NAME VARCHAR(20),
BOOK_ID INTEGER,
FOREIGN KEY (BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,
PRIMARY KEY(BOOK_ID, AUTHOR_NAME));

DESC BOOK_AUTHORS;


CREATE TABLE LIBRARY_PROGRAMME
(PROGRAMME_ID INTEGER PRIMARY KEY,
PROGRAMME_NAME VARCHAR(50),
ADDRESS VARCHAR(50));
```

```
DESC LIBRARY_PROGRAMME;

CREATE TABLE BOOK_COPIES
(NO_OF_COPIES INTEGER,
BOOK_ID INTEGER,
PROGRAMME_ID INTEGER,
FOREIGN KEY (BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY(PROGRAMME_ID) REFERENCES LIBRARY_PROGRAMME(PROGRAMME_ID) ON
DELETE CASCADE,
PRIMARY KEY (BOOK_ID,PROGRAMME_ID));

DESC BOOK_COPIES;

CREATE TABLE CARD
(CARD_NO INTEGER PRIMARY KEY);

DESC CARD;

CREATE TABLE BOOK_LENDING
(BOOK_ID INTEGER,
PROGRAMME_ID INTEGER,
CARD_NO INTEGER,
DATE_OUT DATE,
DUE_DATE DATE,
FOREIGN KEY (BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY (PROGRAMME_ID) REFERENCES LIBRARY_PROGRAMME(PROGRAMME_ID) ON
DELETE CASCADE,
FOREIGN KEY (CARD_NO) REFERENCES CARD(CARD_NO) ON DELETE CASCADE,
PRIMARY KEY (BOOK_ID,PROGRAMME_ID,CARD_NO));

DESC BOOKLENDING;
```

INSERT INTO TABLES

```
INSERT INTO PUBLISHER VALUES('SAPNA',912121212,'BANGALORE');
INSERT INTO PUBLISHER VALUES('PENGUIN',921212121,'NEW YORK');
INSERT INTO PUBLISHER VALUES('PEARSON',913131313,'HYDERABAD');
INSERT INTO PUBLISHER VALUES('OZONE',931313131,'CHENNAI');
INSERT INTO PUBLISHER VALUES('PLANETZ',914141414,'BANGALORE');

SELECT * FROM PUBLISHER;

INSERT INTO BOOK VALUES(1,'BASICS OF EXCEL','JAN-2017','SAPNA');
INSERT INTO BOOK VALUES(2,'PROGRAMMING MINDSET','JUN-2018','PLANETZ');
INSERT INTO BOOK VALUES(3,'BASICS OF SQL','SEP-2016','PEARSON');
INSERT INTO BOOK VALUES(4,'DBMS FOR BEGINNERS','SEP-2015','PLANETZ');
INSERT INTO BOOK VALUES(5,'WEB SERVICES','MAY-2017','OZONE');

SELECT * FROM BOOK;

INSERT INTO BOOK_AUTHORS VALUES('SRI DEVI',1);
INSERT INTO BOOK_AUTHORS VALUES('DEEPAK',2);
INSERT INTO BOOK_AUTHORS VALUES('PRAMOD',3);
INSERT INTO BOOK_AUTHORS VALUES('SWATHI',4);
INSERT INTO BOOK_AUTHORS VALUES('PRATHIMA',5);

SELECT * FROM BOOK_AUTHORS;

INSERT INTO LIBRARY_PROGRAMME VALUES(100,'HSR LAYOUT','BANGALORE');
INSERT INTO LIBRARY_PROGRAMME VALUES(101,'KENGERI','BANGALORE');
INSERT INTO LIBRARY_PROGRAMME VALUES(102,'BANASHANKARI','BANGALORE');
INSERT INTO LIBRARY_PROGRAMME VALUES(103,'SHANKARA NAGAR','MANGALORE');
INSERT INTO LIBRARY_PROGRAMME VALUES(104,'MANIPAL','UDUPI');

SELECT * FROM LIBRARY_PROGRAMME;
```

```
INSERT INTO BOOK_COPIES VALUES(10,1,100);
INSERT INTO BOOK_COPIES VALUES(16,1,101);
INSERT INTO BOOK_COPIES VALUES(20,2,102);
INSERT INTO BOOK_COPIES VALUES(6,2,103);
INSERT INTO BOOK_COPIES VALUES(4,3,104);
INSERT INTO BOOK_COPIES VALUES(7,5,100);
INSERT INTO BOOK_COPIES VALUES(3,4,101);

SELECT * FROM BOOK_COPIES;

INSERT INTO CARD VALUES(500);
INSERT INTO CARD VALUES(501);
INSERT INTO CARD VALUES(502);
INSERT INTO CARD VALUES(503);
INSERT INTO CARD VALUES(504);

SELECT * FROM CARD;

INSERT INTO BOOK_LENDING VALUES(1, 100, 501, '2017-01-01','2017-01-31');
INSERT INTO BOOK_LENDING VALUES(3, 104, 501, '2017-01-11','2017-03-01');
INSERT INTO BOOK_LENDING VALUES(2, 103, 501, '2017-02-21','2017-04-21');
INSERT INTO BOOK_LENDING VALUES(4, 101, 501, '2017-03-11','2017-06-11');
INSERT INTO BOOK_LENDING VALUES(1, 101, 504, '2017-04-09','2017-07-08');

SELECT * FROM BOOK_LENDING;
```

QURIES

1) **--Retrieve details of all books in the library – id, title, name of publisher, authors, --number of copies in each Programme, etc.**

```
SELECT B.BOOK_ID, B.TITLE, B.PUB_NAME,
A.AUTHOR_NAME,C.NO_OF_COPIES,L.PROGRAMME_ID
FROM BOOK B, BOOK_AUTHORS A, BOOK_COPIES C, LIBRARY_PROGRAMME L
WHERE B.BOOK_ID=A.BOOK_ID
AND B.BOOK_ID=C.BOOK_ID
AND L.PROGRAMME_ID=C.PROGRAMME_ID;
```

2) **--Get the particulars of borrowers who have borrowed more than 3 books, but --from Jan 2017 to Jun 2017.**

```
SELECT CARD_NO
FROM BOOK_LENDING
WHERE DATE_OUT BETWEEN '2017-01-01' AND '2017-06-01'
GROUP BY CARD_NO
HAVING COUNT(*)>3;
```

3) **--Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.**

```
DELETE FROM BOOK
WHERE BOOK_ID=3;
SELECT * FROM BOOK;
SELECT * FROM BOOK_AUTHORS;
```

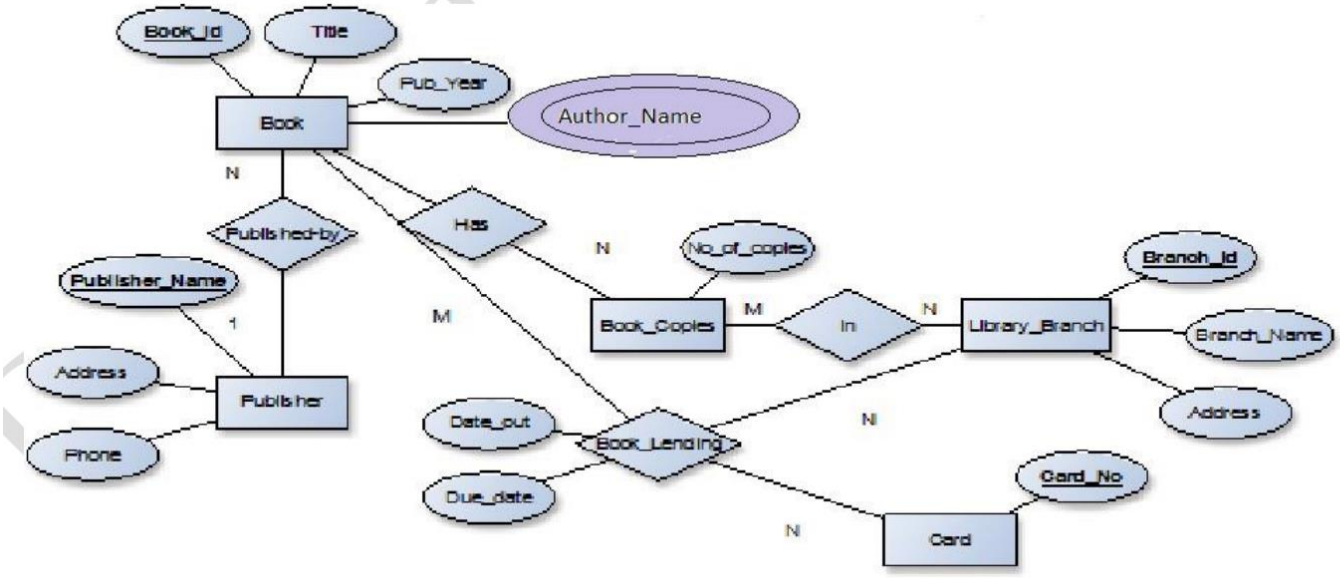
4) **--Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.**

```
CREATE VIEW V_PUBLICATION AS SELECT
PUB_YEAR
FROM BOOK;
SELECT * FROM V_PUBLICATION;
```

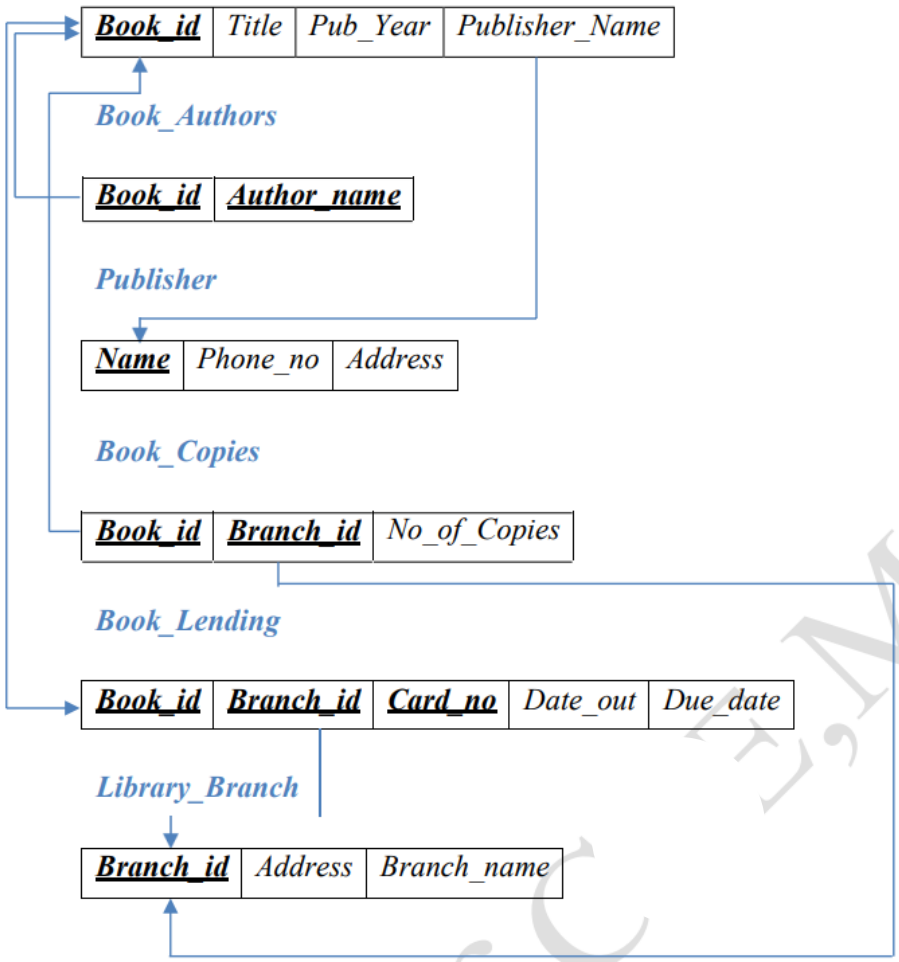
5) **--Create a view of all books and its number of copies that are currently available in the Library.**

```
CREATE VIEW V_BOOKS AS
SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES
FROM
BOOK B, BOOK_COPIES C, LIBRARY_PROGRAMME L
WHERE B.BOOK_ID=C.BOOK_ID
AND C.PROGRAMME_ID=L.PROGRAMME_ID;
SELECT * FROM V_BOOKS;
```

Entity-Relationship Diagram



SCHEMA DIAGRAM



DATABASE: 02(ORDER DATA BASE)

TABEL CREATION

```
CREATE TABLE SALESMAN(
SALESMAN_ID INTEGER PRIMARY KEY,
NAME VARCHAR(20),
CITY VARCHAR(20),
COMMISSION VARCHAR(20));

DESC SALESMAN;

CREATE TABLE CUSTOMER(
CUSTOMER_ID INTEGER PRIMARY KEY,
CUST_NAME VARCHAR(20),
CITY VARCHAR(20),
GRADE INTEGER,
```

```
SALESMAN_ID INTEGER,
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN(SALESMAN_ID) ON DELETE SET
NULL);

DESC CUSTOMER;

CREATE TABLE ORDERS(
ORDER_NO INTEGER PRIMARY KEY,
PURCHASE_AMOUNT DECIMAL(10,2),
ORDER_DATE DATE,
CUSTOMER_ID INTEGER,
SALESMAN_ID INTEGER,
FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID)ON DELETE CASCADE,
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN(SALESMAN_ID) ON DELETE
CASCADE);

DESC ORDERS;
```

INSERT INTO TABLE

```
INSERT INTO SALESMAN VALUES(1000,'RAHUL','BANGALORE','20%');
INSERT INTO SALESMAN VALUES(2000,'ANKITA','BANGALORE','25%');
INSERT INTO SALESMAN VALUES(3000,'SHARMA','MYSORE','30%');
INSERT INTO SALESMAN VALUES(4000,'ANJALI','DELHI','15%');
INSERT INTO SALESMAN VALUES(5000,'RAJ','HYDERABAD','15%');

SELECT * FROM SALESMAN;

INSERT INTO CUSTOMER VALUES(1,'ADYA','BANGALORE',100,1000);
INSERT INTO CUSTOMER VALUES(2,'BANU','MANGALORE',300,1000);
INSERT INTO CUSTOMER VALUES(3,'CHETHAN','CHENNAI',400,2000);
INSERT INTO CUSTOMER VALUES(4,'DANISH','BANGALORE',200,2000);
INSERT INTO CUSTOMER VALUES(5,'ESHA','BANGALORE',400,3000);
```

```
SELECT * FROM CUSTOMER;

INSERT INTO ORDERS VALUES(201,5000,'2020-06-02',1,1000);
INSERT INTO ORDERS VALUES(202,450,'2020-04-09',1,2000);
INSERT INTO ORDERS VALUES(203,1000,'2020-03-15',3,2000);
INSERT INTO ORDERS VALUES(204,3500,'2020-07-09',4,3000);
INSERT INTO ORDERS VALUES(205,550,'2020-05-05',2,2000);

SELECT * FROM ORDERS;
```

QUERIES

1) -- Count the customers with grades above Bangalore's average

```
SELECT GRADE,COUNT(DISTINCT CUSTOMER_ID)
FROM CUSTOMER
GROUP BY GRADE
HAVING GRADE>(SELECT AVG(GRADE)
FROM CUSTOMER
WHERE CITY='BANGALORE');
```

2) --Find the name and numbers of all salesman who had more than one customer

```
SELECT SALESMAN_ID, NAME
FROM SALESMAN S
WHERE (SELECT COUNT(*)
FROM CUSTOMER C
WHERE C.SALESMAN_ID=S.SALESMAN_ID) > 1;
```

3) --List all the salesman and indicate those who have and don't have customers in their cities (Use UNION operation.)

```
SELECT S.SALESMAN_ID, S.NAME, C.CUST_NAME, S.COMMISSION
FROM SALESMAN S, CUSTOMER C
```

```
WHERE S.CITY=C.CITY
UNION
SELECT S.SALESMAN_ID,S.NAME,'NO MATCH',S.COMMISSION
FROM SALESMAN S
WHERE CITY NOT IN
(SELECT CITY
FROM CUSTOMER)
ORDER BY 1 ASC;
```

4) --Create a view that finds the salesman who has the customer with the highest order of a day.

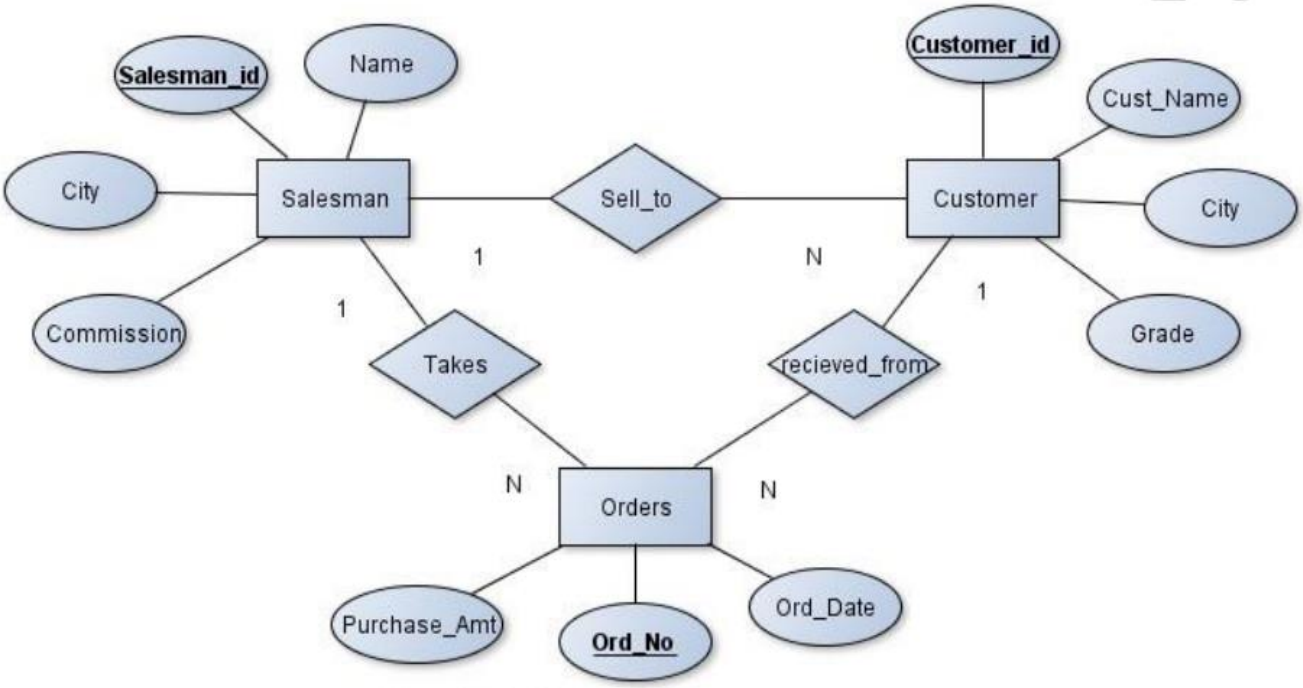
```
SELECT S.SALESMAN_ID, S.NAME, C.CUST_NAME, S.COMMISSION
FROM SALESMAN S, CUSTOMER C
WHERE S.CITY=C.CITY
UNION
SELECT S.SALESMAN_ID,S.NAME,'NO MATCH',S.COMMISSION
FROM SALESMAN S
WHERE CITY NOT IN
(SELECT CITY
FROM CUSTOMER)
ORDER BY 1 ASC;
```

5) --Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

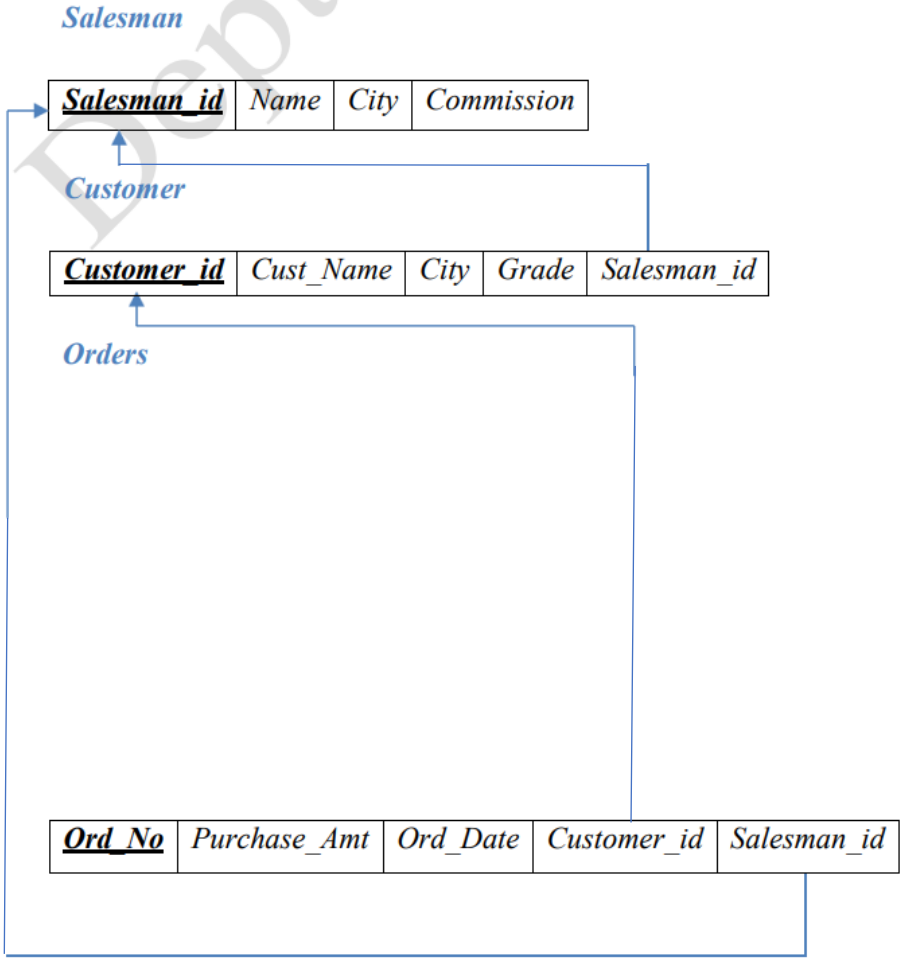
```
CREATE VIEW V_SALESMAN AS
SELECT O.ORDER_DATE, S.SALESMAN_ID, S.NAME
FROM SALESMAN S,ORDERS O
WHERE S.SALESMAN_ID = O.SALESMAN_ID
AND O.PURCHASE_AMOUNT= (SELECT MAX(PURCHASE_AMOUNT)
FROM ORDERS C
WHERE C.ORDER_DATE=O.ORDER_DATE);

SELECT * FROM V_SALESMAN;
```

```
DELETE FROM SALESMAN
WHERE SALESMAN_ID=1000;

SELECT * FROM SALESMAN;
SELECT * FROM ORDERS;
```

Entity-Relationship Diagram



Schema Diagram



DATABASE: 03(MOVIE DATA BASE)

TABLE CREATION

```
CREATE TABLE ACTOR (
ACT_ID INTEGER PRIMARY KEY,
ACT_NAME VARCHAR(20),
ACT_GENDER CHAR(1));

DESC ACTOR;

CREATE TABLE DIRECTOR(
DIR_ID INTEGER PRIMARY KEY,
DIR_NAME VARCHAR(20),
DIR_PHONE INTEGER);

DESC DIRECTOR;

CREATE TABLE MOVIES(
MOV_ID INTEGER PRIMARY KEY,
MOV_TITLE VARCHAR(25),
MOV_YEAR INTEGER,
MOV_LANG VARCHAR(15),
DIR_ID INTEGER,
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR(DIR_ID));

DESC MOVIES;

CREATE TABLE MOVIE_CAST(
ACT_ID INTEGER,
MOV_ID INTEGER,
ROLE VARCHAR(10),
PRIMARY KEY (ACT_ID,MOV_ID),
FOREIGN KEY (ACT_ID) REFERENCES ACTOR(ACT_ID),
FOREIGN KEY (MOV_ID) REFERENCES MOVIES(MOV_ID));

DESC MOVIE_CAST;

CREATE TABLE RATING(
MOV_ID INTEGER PRIMARY KEY,
REV_STARS VARCHAR(25),
FOREIGN KEY (MOV_ID) REFERENCES MOVIES(MOV_ID));


DESC RATING;
```

INSERTION TABLE

```
INSERT INTO ACTOR VALUES(101,'RAHUL','M');
INSERT INTO ACTOR VALUES(102,'ANKITHA','F');
INSERT INTO ACTOR VALUES(103,'RADHIKA','F');
INSERT INTO ACTOR VALUES(104,'CHETHAN','M');
INSERT INTO ACTOR VALUES(105,'VIVAN','M');

SELECT * FROM ACTOR;

INSERT INTO DIRECTOR VALUES(201,'ANUP',918181818);
INSERT INTO DIRECTOR VALUES(202,'HITCHCOCK',918181812);
INSERT INTO DIRECTOR VALUES(203,'SHASHANK',918181813);
INSERT INTO DIRECTOR VALUES(204,'STEVEN SPIELBERG',918181814);
INSERT INTO DIRECTOR VALUES(205,'ANAND',918181815);

SELECT * FROM DIRECTOR;

INSERT INTO MOVIES VALUES(1001,'MANASU',2017,'KANNADA',201);
INSERT INTO MOVIES VALUES(1002,'AAKASHAM',2015,'TELUGU',202);
INSERT INTO MOVIES VALUES(1003,'KALIYONA',2008,'KANNADA',201);
INSERT INTO MOVIES VALUES(1004,'WAR HORSE',2011,'ENGLISH',204);
INSERT INTO MOVIES VALUES(1005,'HOME',2012,'ENGLISH',205);

SELECT * FROM MOVIES;

INSERT INTO MOVIE_CAST VALUES(101,1002,'HERO');
INSERT INTO MOVIE_CAST VALUES(101,1001,'HERO');
INSERT INTO MOVIE_CAST VALUES(103,1003,'HEROINE');
INSERT INTO MOVIE_CAST VALUES(103,1002,'GUEST');
INSERT INTO MOVIE_CAST VALUES(104,1004,'HERO');

SELECT * FROM MOVIE_CAST;

INSERT INTO RATING VALUES(1001,4);
INSERT INTO RATING VALUES(1002,2);
INSERT INTO RATING VALUES(1003,5);
INSERT INTO RATING VALUES(1004,4);
INSERT INTO RATING VALUES(1005,3);

SELECT * FROM RATING;
```

QUERIES

1)--List the titles of all movies directed by ‘Hitchcock’.

```
 SELECT MOV_TITLE

FROM MOVIES
WHERE DIR_ID = (SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME='HITCHCOCK');
```

2) --Find the movie names where one or more actors acted in two or more movies.

```
SELECT MOV_TITLE

FROM MOVIES M,MOVIE_CAST MC
WHERE M.MOV_ID=MC.MOV_ID AND ACT_ID IN (SELECT ACT_ID
FROM MOVIE_CAST GROUP BY ACT_ID
HAVING COUNT(ACT_ID)>1)
GROUP BY MOV_TITLE
HAVING COUNT(*)>1;
```

3) --List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

```
SELECT ACT_NAME

FROM ACTOR A
JOIN MOVIE_CAST C
ON A.ACT_ID=C.ACT_ID
JOIN MOVIES M
ON C.MOV_ID=M.MOV_ID
WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;
```

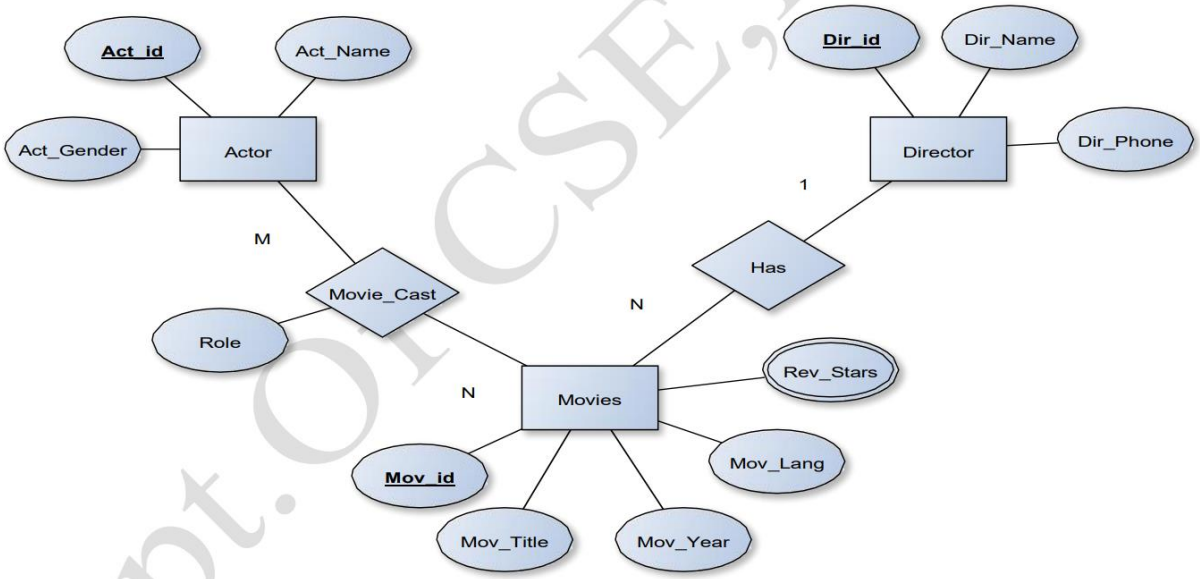
4) --Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

```
SELECT MOV_TITLE,MAX(REV_STARS)

FROM MOVIES
INNER JOIN RATING USING (MOV_ID)
GROUP BY MOV_TITLE
HAVING MAX(REV_STARS)>0
ORDER BY MOV_TITLE;
```

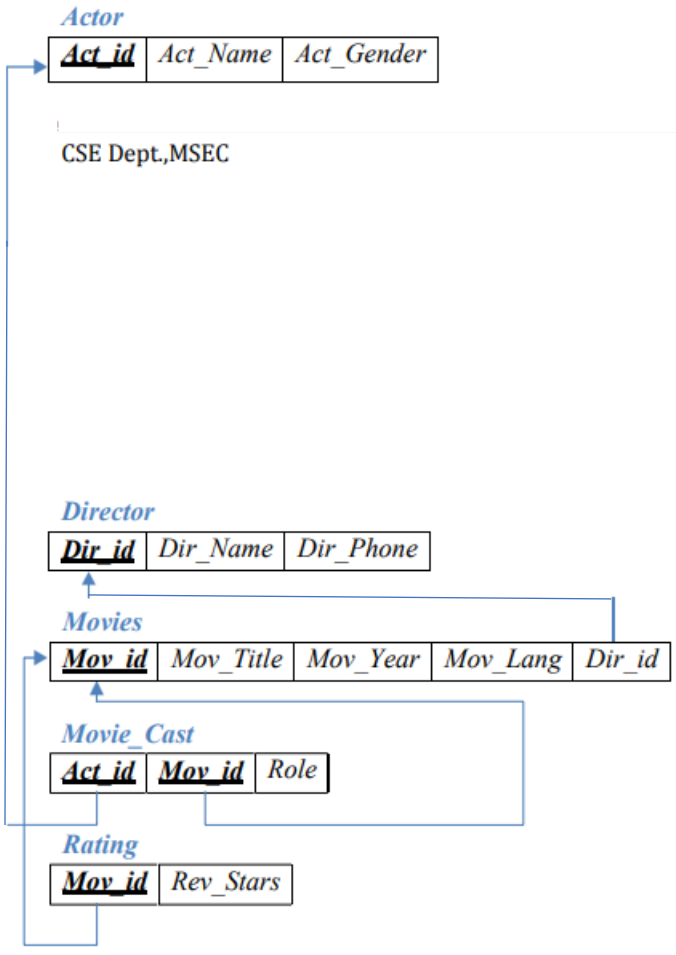
5) --Update rating of all movies directed by 'Steven Spielberg' to 5.

```
UPDATE RATING

SET REV_STARS=5
WHERE MOV_ID IN (SELECT MOV_ID FROM MOVIES
WHERE DIR_ID IN (SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME='STEVEN SPIELBERG'));
```

Entity-Relationship Diagram



Schema Diagram



DATABASE: 04(COLLEGE DATA BASE)CREATE TABLE

```
CREATE TABLE STUDENT(
USN VARCHAR(10) PRIMARY KEY,
SNAME VARCHAR(25),
ADDRESS VARCHAR(25),
PHONE INTEGER,
GENDER CHAR(1));

DESC STUDENT;

CREATE TABLE SEMSEC(
SSID VARCHAR(5) PRIMARY KEY,
SEM INTEGER,
SEC CHAR(1));

DESC SEMSEC;

CREATE TABLE CLASS(
USN VARCHAR(10) PRIMARY KEY,
SSID VARCHAR(5),
FOREIGN KEY(USN) REFERENCES STUDENT(USN),
FOREIGN KEY(SSID) REFERENCES SEMSEC(SSID));

DESC CLASS;

CREATE TABLE SUBJECT(
SUBCODE VARCHAR(8) PRIMARY KEY,
TITLE VARCHAR(20),
SEM INTEGER,
CREDITS INTEGER);

DESC SUBJECT;

CREATE TABLE IAMARKS(
USN VARCHAR(10),
SUBCODE VARCHAR(8),
SSID VARCHAR(5),
TEST1 INTEGER,
TEST2 INTEGER,
TEST3 INTEGER,
FINALIA INTEGER,
PRIMARY KEY(SUBCODE,USN,SSID),
FOREIGN KEY(USN) REFERENCES STUDENT(USN),
FOREIGN KEY(SUBCODE) REFERENCES SUBJECT(SUBCODE),
FOREIGN KEY(SSID) REFERENCES SEMSEC(SSID));

DESC IAMARKS;
```

INSERT TABLE

```
INSERT INTO STUDENT VALUES ('1BI13CS020','ANAND','BELAGAVI', 1233423,'M');
```

```
INSERT INTO STUDENT VALUES ('1BI13CS062','BABIITHA','BENGALURU',43123,'F');
INSERT INTO STUDENT VALUES ('1BI15CS101','CHETHAN','BENGALURU', 534234,'M');
INSERT INTO STUDENT VALUES ('1BI13CS066','DIVYA','MANGALURU',534432,'F');
INSERT INTO STUDENT VALUES ('1BI14CS010','EESHA','BENGALURU', 345456,'F');
INSERT INTO STUDENT VALUES ('1BI14CS032','GANESH','BENGALURU',574532,'M');
INSERT INTO STUDENT VALUES ('1BI14CS025','HARISH','BENGALURU', 235464,'M');
INSERT INTO STUDENT VALUES ('1BI15CS011','ISHA','TUMKUR', 764343,'F');
INSERT INTO STUDENT VALUES ('1BI15CS029','JOEY','DAVANGERE', 235653,'M');
INSERT INTO STUDENT VALUES ('1BI15CS045','KAVYA','BELLARY', 865434,'F');
INSERT INTO STUDENT VALUES ('1BI15CS091','MALINI','MANGALURU',235464,'F');
INSERT INTO STUDENT VALUES ('1BI16CS045','NEEL','KALBURGI', 856453,'M');
INSERT INTO STUDENT VALUES ('1BI16CS088','PARTHA','SHIMOGA', 234546,'M');
INSERT INTO STUDENT VALUES ('1BI16CS122','REEMA','CHIKAMAGALUR', 853333,'F');

SELECT * FROM STUDENT;

INSERT INTO SEMSEC VALUES ('CSE8A', 8,'A');
INSERT INTO SEMSEC VALUES ('CSE8B', 8,'B');
INSERT INTO SEMSEC VALUES ('CSE8C', 8,'C');
INSERT INTO SEMSEC VALUES ('CSE7A', 7,'A');
INSERT INTO SEMSEC VALUES ('CSE7B', 7,'B');
INSERT INTO SEMSEC VALUES ('CSE7C', 7,'C');
INSERT INTO SEMSEC VALUES ('CSE6A', 6,'A');
INSERT INTO SEMSEC VALUES ('CSE6B', 6,'B');
INSERT INTO SEMSEC VALUES ('CSE6C', 6,'C');
INSERT INTO SEMSEC VALUES ('CSE5A', 5,'A');
INSERT INTO SEMSEC VALUES ('CSE5B', 5,'B');
INSERT INTO SEMSEC VALUES ('CSE5C', 5,'C');
INSERT INTO SEMSEC VALUES ('CSE4A', 4,'A');
INSERT INTO SEMSEC VALUES ('CSE4B', 4,'B');
INSERT INTO SEMSEC VALUES ('CSE4C', 4,'C');
INSERT INTO SEMSEC VALUES ('CSE3A', 3,'A');
INSERT INTO SEMSEC VALUES ('CSE3B', 3,'B');
INSERT INTO SEMSEC VALUES ('CSE3C', 3,'C');
INSERT INTO SEMSEC VALUES ('CSE2A', 2,'A');
INSERT INTO SEMSEC VALUES ('CSE2B', 2,'B');
INSERT INTO SEMSEC VALUES ('CSE2C', 2,'C');
INSERT INTO SEMSEC VALUES ('CSE1A', 1,'A');
INSERT INTO SEMSEC VALUES ('CSE1B', 1,'B');
INSERT INTO SEMSEC VALUES ('CSE1C', 1,'C');

SELECT * FROM SEMSEC;

INSERT INTO CLASS VALUES ('1BI13CS020','CSE8A');
INSERT INTO CLASS VALUES ('1BI13CS062','CSE8A');
INSERT INTO CLASS VALUES ('1BI13CS066','CSE8B');
INSERT INTO CLASS VALUES ('1BI15CS101','CSE8C');
INSERT INTO CLASS VALUES ('1BI14CS010','CSE7A');
INSERT INTO CLASS VALUES ('1BI14CS025','CSE7A');
INSERT INTO CLASS VALUES ('1BI14CS032','CSE7A');
INSERT INTO CLASS VALUES ('1BI15CS011','CSE4A');
INSERT INTO CLASS VALUES ('1BI15CS029','CSE4A');
```

```
INSERT INTO CLASS VALUES ('1BI15CS045','CSE4B');
INSERT INTO CLASS VALUES ('1BI15CS091','CSE4C');
INSERT INTO CLASS VALUES ('1BI16CS045','CSE3A');
INSERT INTO CLASS VALUES ('1BI16CS088','CSE3B');
INSERT INTO CLASS VALUES ('1BI16CS122','CSE3C');

SELECT * FROM CLASS;

INSERT INTO SUBJECT VALUES ('10CS81','ACA', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS82','SSM', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS83','NM', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS84','CC', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS85','PW', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS71','OOAD', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS72','ECS', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS73','PTW', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS74','DWDM', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS75','JAVA', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS76','SAN', 7, 4);
INSERT INTO SUBJECT VALUES ('15CS51','ME', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS52','CN', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS53','DBMS', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS54','ATC', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS55','JAVA', 5, 3);
INSERT INTO SUBJECT VALUES ('15CS56','AI', 5, 3);
INSERT INTO SUBJECT VALUES ('15CS41','M4', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS42','SE', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS43','DAA', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS44','MPMC', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS45','OOC', 4, 3);
INSERT INTO SUBJECT VALUES ('15CS46','DC', 4, 3);
INSERT INTO SUBJECT VALUES ('15CS31','M3', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS32','ADE', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS33','DSA', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS34','CO', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS35','USP', 3, 3);
INSERT INTO SUBJECT VALUES ('15CS36','DMS', 3, 3);

SELECT * FROM SUBJECT;

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES
('1BI15CS101','10CS81','CSE8C', 15, 16, 18);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES
('1BI15CS101','10CS82','CSE8C', 12, 19, 14);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES
('1BI15CS101','10CS83','CSE8C', 19, 15, 20);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES
('1BI15CS101','10CS84','CSE8C', 20, 16, 19);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES
('1BI15CS101','10CS85','CSE8C', 15, 15, 12);

SELECT * FROM IAMARKS;
```

QUERIES

1) --List all the student details studying in fourth semester ‘C’ section.

```
SELECT S.*, SS.SEM, SS.SEC
FROM STUDENT S, SEMSEC SS, CLASS C
WHERE S.USN = C.USN AND
SS.SSID = C.SSID AND
SS.SEM = 4 AND SS.SEC='C';
```

3) --Compute the total number of male and female students in each semester and in each section.

```
SELECT SS.SEM, SS.SEC, S.GENDER, COUNT(S.GENDER) AS COUNT
FROM STUDENT S, SEMSEC SS, CLASS C
WHERE S.USN = C.USN AND
SS.SSID = C.SSID
GROUP BY SS.SEM, SS.SEC, S.GENDER
ORDER BY SEM;
```

4) --Create a view of Test1 marks of student USN ‘1BI15CS101’ in all Courses.

```
DELIMITER //
CREATE PROCEDURE AVG_MARKS()
BEGIN
DECLARE C_A INTEGER;
DECLARE C_B INTEGER;
DECLARE C_C INTEGER;
DECLARE C_SUM INTEGER;
DECLARE C_AVG INTEGER;
DECLARE C_USN VARCHAR(10);
DECLARE C_SUBCODE VARCHAR(8);
DECLARE C_SSID VARCHAR(5);

DECLARE C_IAMARKS CURSOR FOR
SELECT GREATEST(TEST1,TEST2) AS A, GREATEST(TEST1,TEST3) AS B,
GREATEST(TEST3,TEST2) AS C, USN, SUBCODE, SSID
FROM IAMARKS
WHERE FINALIA IS NULL
FOR UPDATE;

OPEN C_IAMARKS;
LOOP

FETCH C_IAMARKS INTO C_A, C_B, C_C, C_USN, C_SUBCODE, C_SSID;

IF (C_A != C_B) THEN
    SET C_SUM=C_A+C_B;
```

```
ELSE
    SET C_SUM=C_A+C_C;
END IF;

SET C_AVG=C_SUM/2;

UPDATE IAMARKS SET FINALIA = C_AVG
WHERE USN = C_USN AND SUBCODE = C_SUBCODE AND SSID = C_SSID;

END LOOP;
CLOSE C_IAMARKS;
END;
//
```

5) -- Categorize students based on the following criterion:

-- If FinalIA = 17 to 20 then CAT = 'Outstanding'

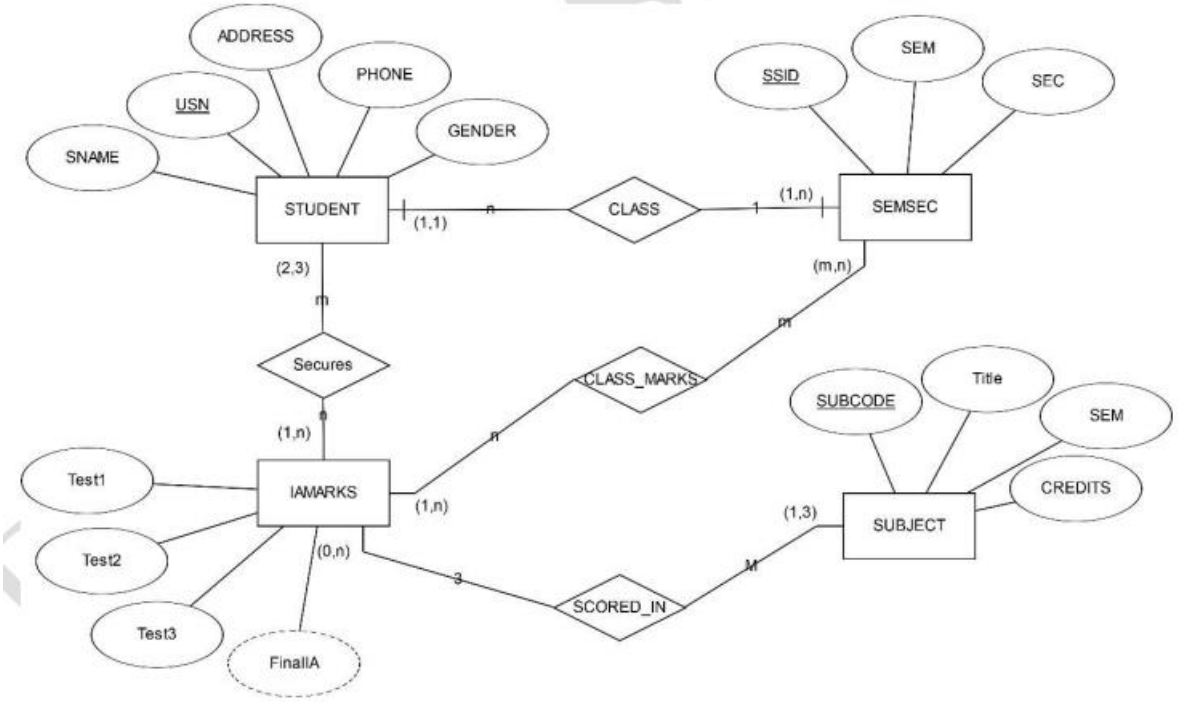
-- If FinalIA = 12 to 16 then CAT = 'Average'

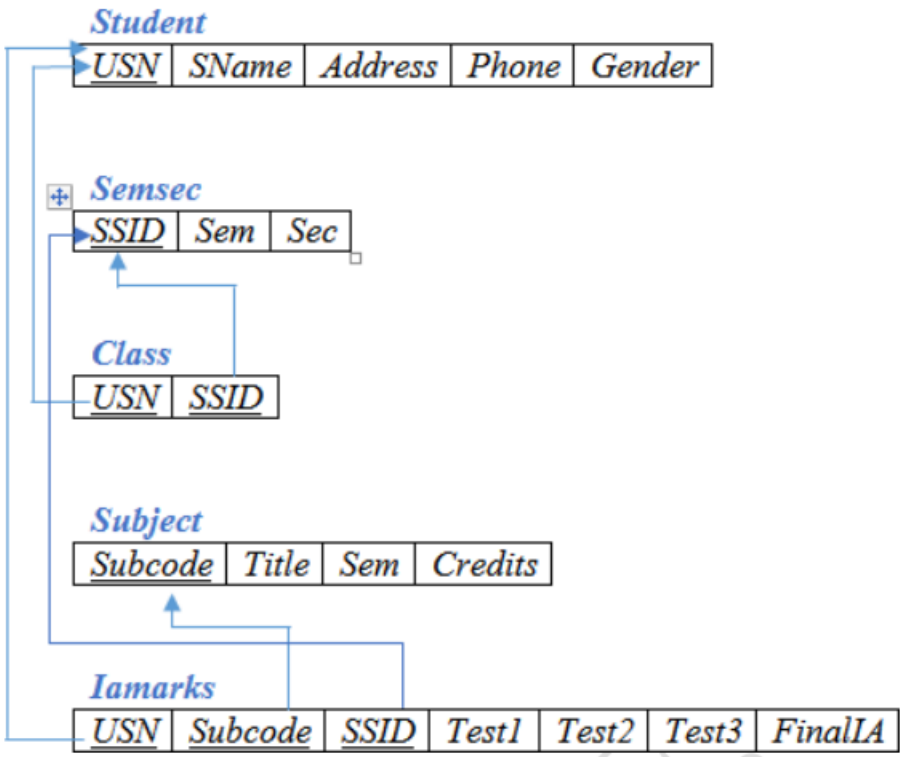
-- If FinalIA< 12 then CAT = 'Weak'

-- Give these details only for 8th semester A, B, and C section students.

```
CALL AVG_MARKS();

SELECT * FROM IAMARKS;

SELECT S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER, IA.SUBCODE,
(CASE
WHEN IA.FINALIA BETWEEN 17 AND 20 THEN 'OUTSTANDING'
WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE'
ELSE 'WEAK'
END) AS CAT
FROM STUDENT S, SEMSEC SS, IAMARKS IA, SUBJECT SUB
WHERE S.USN = IA.USN AND
SS.SSID = IA.SSID AND
SUB.SUBCODE = IA.SUBCODE AND
SUB.SEM = 8;
```

Entity - Relationship Diagram





DATABASE: 05(COMPANY DATA BASE)

TABEL CREATION

```
CREATE TABLE DEPARTMENT
(DNO VARCHAR(20) PRIMARY KEY,
DNAME VARCHAR(20),
MGR_SSN VARCHAR(20),
MGR_START_DATE DATE);

DESC DEPARTMENT;

CREATE TABLE EMPLOYEE
(SSN VARCHAR(20) PRIMARY KEY,
NAME VARCHAR(20),
ADDRESS VARCHAR(20),
SEX CHAR(1),
SALARY INTEGER,
SUPERSSN VARCHAR(20),
DNO VARCHAR(20),
FOREIGN KEY (SUPERSSN) REFERENCES EMPLOYEE (SSN),
FOREIGN KEY (DNO) REFERENCES DEPARTMENT (DNO));

DESC EMPLOYEE;

ALTER TABLE DEPARTMENT
ADD FOREIGN KEY (MGR_SSN) REFERENCES EMPLOYEE(SSN);

CREATE TABLE DLOCATION
(DLOC VARCHAR(20),
DNO VARCHAR(20),
FOREIGN KEY (DNO) REFERENCES DEPARTMENT(DNO),
PRIMARY KEY (DNO, DLOC));

DESC DLOCATION;

CREATE TABLE PROJECT
(PNO INTEGER PRIMARY KEY,
PNAME VARCHAR(20),
PLOCATION VARCHAR(20),
DNO VARCHAR(20),
FOREIGN KEY (DNO) REFERENCES DEPARTMENT(DNO));

DESC PROJECT;
CREATE TABLE WORKS_ON
(HOURS INTEGER,
SSN VARCHAR(20),
PNO INTEGER,
FOREIGN KEY (SSN) REFERENCES EMPLOYEE(SSN),
FOREIGN KEY (PNO) REFERENCES PROJECT(PNO),
PRIMARY KEY (SSN, PNO));

DESC WORKS_ON;
```

INSERT INTO TABLE

```
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC01','BEN SCOTT','BANGALORE','M', 450000);
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC02','HARRY SMITH','BANGALORE','M', 500000);
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC03','LEAN BAKER','BANGALORE','M', 700000);
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC04','MARTIN SCOTT','MYSORE','M', 500000);
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC05','RAVAN HEGDE','MANGALORE','M', 650000);
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC06','GIRISH HOSUR','MYSORE','M', 450000);
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC07','NEELA SHARMA','BANGALORE','F', 800000);
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC08','ADYA KOLAR','MANGALORE','F', 350000);
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC09','PRASANNA KUMAR','MANGALORE','M', 300000);
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC10','VEENA KUMARI','MYSORE','M', 600000);
INSERT INTO EMPLOYEE (SSN, NAME, ADDRESS, SEX, SALARY) VALUES
 ('ABC11','DEEPAK RAJ','BANGALORE','M', 500000);

 SELECT * FROM EMPLOYEE;

INSERT INTO DEPARTMENT VALUES ('1','ACCOUNTS','ABC09', '2016-01-03');
INSERT INTO DEPARTMENT VALUES ('2','IT','ABC11', '2017-02-04');
INSERT INTO DEPARTMENT VALUES ('3','HR','ABC01', '2016-04-05');
INSERT INTO DEPARTMENT VALUES ('4','HELPDESK', 'ABC10', '2017-06-03');
INSERT INTO DEPARTMENT VALUES ('5','SALES','ABC06', '2017-01-08');

SELECT * FROM DEPARTMENT;

UPDATE EMPLOYEE SET
SUPERSSN=NULL, DNO='3'
WHERE SSN='ABC01';

UPDATE EMPLOYEE SET
SUPERSSN='ABC03', DNO='5'
WHERE SSN='ABC02';

UPDATE EMPLOYEE SET
SUPERSSN='ABC04', DNO='5'
WHERE SSN='ABC03';

UPDATE EMPLOYEE SET
SUPERSSN='ABC06', DNO='5'
WHERE SSN='ABC04';

UPDATE EMPLOYEE SET
DNO='5', SUPERSSN='ABC06'
```

```
WHERE SSN='ABC05';

UPDATE EMPLOYEE SET
DNO='5', SUPERSSN='ABC07'
WHERE SSN='ABC06';

UPDATE EMPLOYEE SET
DNO='5', SUPERSSN=NULL
WHERE SSN='ABC07';

UPDATE EMPLOYEE SET
DNO='1', SUPERSSN='ABC09'
WHERE SSN='ABC08';

UPDATE EMPLOYEE SET
DNO='1', SUPERSSN=NULL
WHERE SSN='ABC09';

UPDATE EMPLOYEE SET
DNO='4', SUPERSSN=NULL
WHERE SSN='ABC10';

UPDATE EMPLOYEE SET
DNO='2', SUPERSSN=NULL
WHERE SSN='ABC11';

SELECT * FROM EMPLOYEE;

INSERT INTO DLOCATION VALUES ('BENGALURU', '1');
INSERT INTO DLOCATION VALUES ('BENGALURU', '2');
INSERT INTO DLOCATION VALUES ('BENGALURU', '3');
INSERT INTO DLOCATION VALUES ('MYSORE', '4');
INSERT INTO DLOCATION VALUES ('MYSORE', '5');

SELECT * FROM DLOCATION;

INSERT INTO PROJECT VALUES (1000,'IOT','BENGALURU','5');
INSERT INTO PROJECT VALUES (1001,'CLOUD','BENGALURU','5');
INSERT INTO PROJECT VALUES (1002,'BIGDATA','BENGALURU','5');
INSERT INTO PROJECT VALUES (1003,'SENSORS','BENGALURU','3');
INSERT INTO PROJECT VALUES (1004,'BANK MANAGEMENT','BENGALURU','1');
INSERT INTO PROJECT VALUES (1005,'SALARY MANAGEMENT','BANGALORE','1');
INSERT INTO PROJECT VALUES (1006,'OPENSTACK','BENGALURU','4');
INSERT INTO PROJECT VALUES (1007,'SMART CITY','BENGALURU','2');

SELECT * FROM PROJECT;

INSERT INTO WORKS_ON VALUES (4, 'ABC02', 1000);
INSERT INTO WORKS_ON VALUES (6, 'ABC02', 1001);
INSERT INTO WORKS_ON VALUES (8, 'ABC02', 1002);
INSERT INTO WORKS_ON VALUES (10,'ABC03', 1000);
INSERT INTO WORKS_ON VALUES (3, 'ABC05', 1000);
```

```
INSERT INTO WORKS_ON VALUES (4, 'ABC06', 1001);
INSERT INTO WORKS_ON VALUES (5, 'ABC07', 1002);
INSERT INTO WORKS_ON VALUES (6, 'ABC04', 1002);
INSERT INTO WORKS_ON VALUES (7, 'ABC01', 1003);
INSERT INTO WORKS_ON VALUES (5, 'ABC08', 1004);
INSERT INTO WORKS_ON VALUES (6, 'ABC09', 1005);
INSERT INTO WORKS_ON VALUES (4, 'ABC10', 1006);
INSERT INTO WORKS_ON VALUES (10,'ABC11', 1007);

SELECT * FROM WORKS_ON;
```

QUERIES

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

```
SELECT DISTINCT P.PNO

FROM PROJECT P, DEPARTMENT D, EMPLOYEE E
WHERE E.DNO=D.DNO
AND D.MGR_SSN=E.SSN
AND E.NAME LIKE '%SCOTT'
UNIONZ
SELECT DISTINCT P1.PNO
FROM PROJECT P1, WORKS_ON W, EMPLOYEE E1
WHERE P1.PNO=W.PNO
AND E1.SSN=W.SSN
AND E1.NAME LIKE '%SCOTT';
```

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

```
SELECT E.NAME, 1.1*E.SALARY AS INCR_SAL

FROM EMPLOYEE E, WORKS_ON W, PROJECT P
WHERE E.SSN=W.SSN
AND W.PNO=P.PNO
AND P.PNAME='IOT';
```

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(E.SALARY), MAX(E.SALARY), MIN(E.SALARY), AVG(E.SALARY)

FROM EMPLOYEE E, DEPARTMENT D
WHERE E.DNO=D.DNO
AND D.DNAME='ACCOUNTS';
```

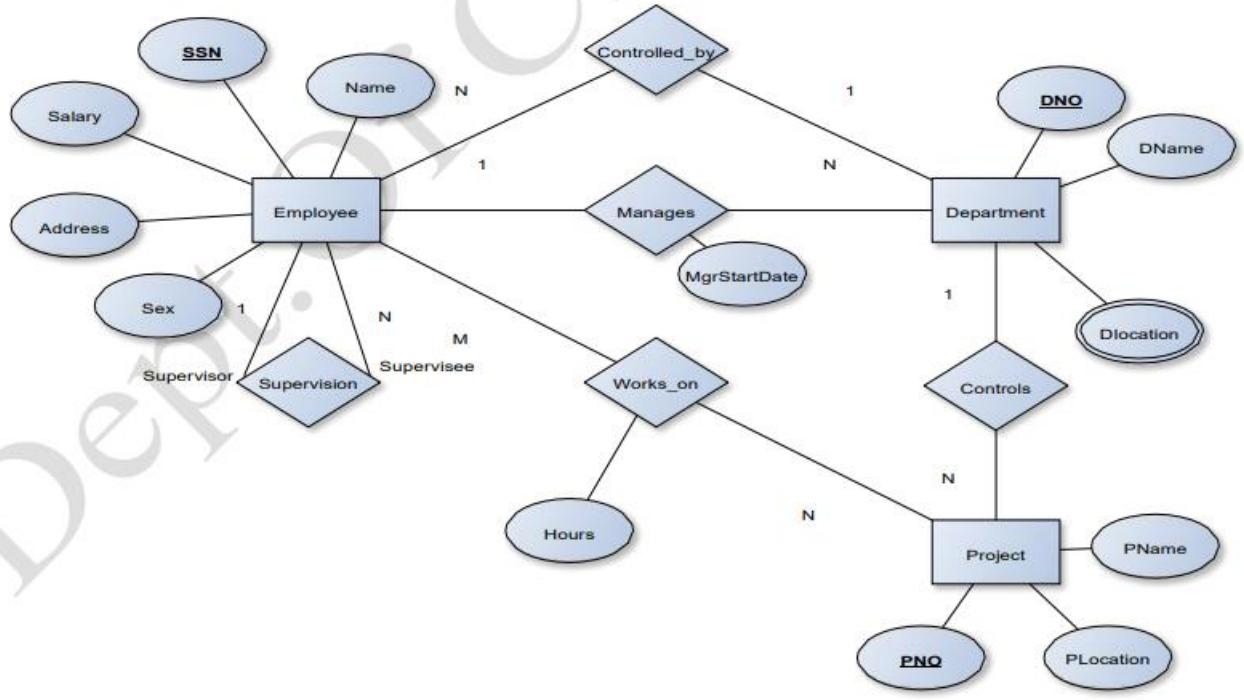
4) -Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator).

```
SELECT E.NAME

FROM EMPLOYEE E
WHERE NOT EXISTS(SELECT PNO FROM PROJECT WHERE DNO='5' AND PNO NOT IN (SELECT
PNO FROM WORKS_ON
WHERE E.SSN=SSN));
```

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

```
SELECT D.DNO, COUNT(*)

FROM DEPARTMENT D, EMPLOYEE E
WHERE D.DNO=E.DNO
AND E.SALARY > 600000
AND D.DNO IN (SELECT E1.DNO
FROM EMPLOYEE E1
GROUP BY E1.DNO
HAVING COUNT(*)>5)
GROUP BY D.DNO;
```

Entity-Relationship Diagram



Schema Diagram