PES Institute of Technology and Management

Department of Information Science & Engineering

Laboratory Manual

PART A: SQL PROGRAMMING

A. Consider the following schema for a LibraryDatabase:

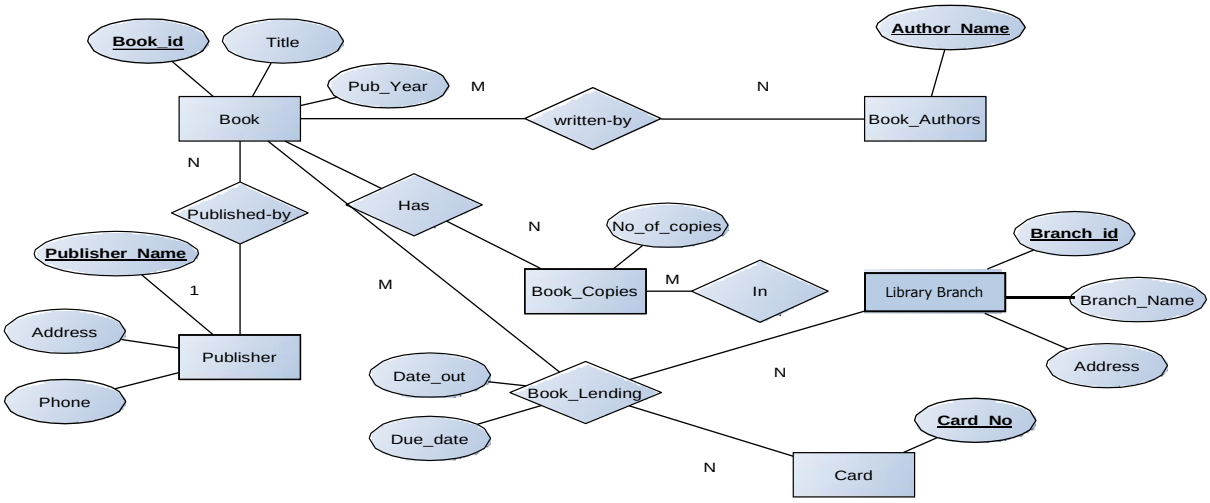
BOOK (*Book_id, Title, Publisher_Name, Pub_Year*)
BOOK_AUTHORS (Book_id, Author_*Name*)
PUBLISHER (*Name, Address, Phone*) BOOK_COPIES
(*Book_id, Branch_id, No-of_Copies*)
BOOK_LENDING (*Book_id, Branch_id, Card_No, Date_Out, Due_Date*)
LIBRARY_BRANCH (*Branch_id, Branch_Name, Address*)

Write SQL queries to

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch,etc.
2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun2017
3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulationoperation.
4. Partition the BOOK table based on year of publication. Demonstrate its working with a simplequery.
5. Create a view of all books and its number of copies that are currently available in the Library.

Solution:

Entity-Relationship Diagram



Schema Diagram

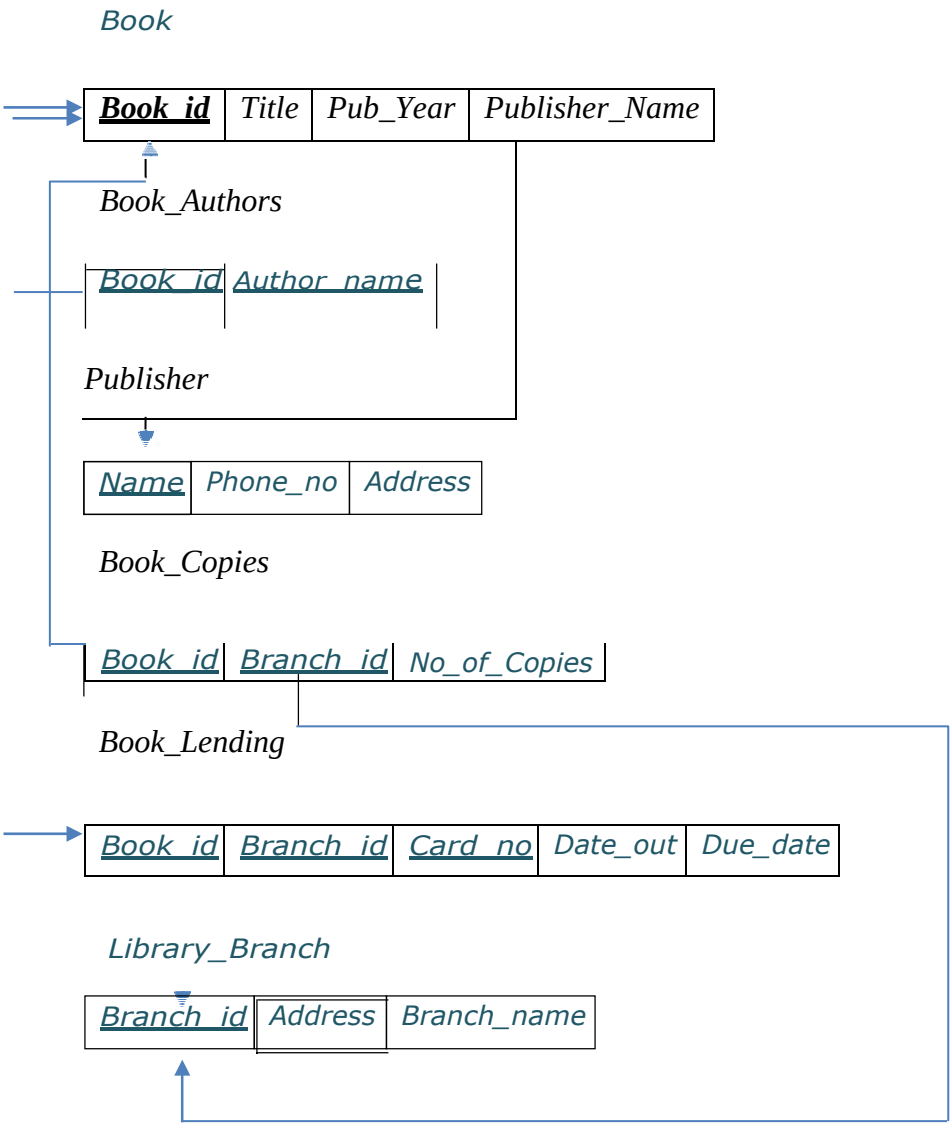


Table Creation

```
CREATE TABLE PUBLISHER
(NAME VARCHAR2 (20) PRIMARY KEY,
PHONE INTEGER,
ADDRESS VARCHAR2 (20));

CREATE TABLEBOOK
(BOOK_ID INTEGER PRIMARYKEY,
TITLE VARCHAR2(20),
PUB_YEAR VARCHAR2 (20),
PUBLISHER_NAME REFERENCES PUBLISHER (NAME) ON DELETE CASCADE);
```

```
CREATE TABLE BOOK_AUTHORS
(AUTHOR_NAME VARCHAR2 (20),
BOOK_ID REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
PRIMARY KEY (BOOK_ID, AUTHOR_NAME));

CREATE TABLE LIBRARY_BRANCH (BRANCH_ID
INTEGER PRIMARY KEY, BRANCH_NAME VARCHAR2
(50),
ADDRESS VARCHAR2 (50));

CREATE TABLE BOOK_COPIES
(NO_OF_COPIES INTEGER,
BOOK_ID REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
BRANCH_ID REFERENCES LIBRARY_BRANCH (BRANCH_ID) ON DELETE
CASCADE,
PRIMARY KEY (BOOK_ID, BRANCH_ID));

CREATE TABLE CARD
(CARD_NO INTEGER PRIMARY KEY);

CREATE TABLE BOOK_LENDING
(DATE_OUT DATE,
DUE_DATE DATE,
BOOK_ID REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
BRANCH_ID REFERENCES LIBRARY_BRANCH (BRANCH_ID) ON DELETE
CASCADE,
CARD_NO REFERENCES CARD (CARD_NO) ON DELETE CASCADE,
PRIMARY KEY (BOOK_ID, BRANCH_ID, CARD_NO));
```

Table Descriptions

DESC PUBLISHER;

```
SQL> desc publisher;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                      NOT NULL VARCHAR2(20)
 PHONE                                              NUMBER(38)
 ADDRESS                                            VARCHAR2(20)
```

DESC BOOK;

```
SQL> DESC BOOK;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOK_ID                                   NOT NULL NUMBER(38)
 TITLE                                             VARCHAR2(20)
 PUB_YEAR                                          VARCHAR2(20)
 PUBLISHER_NAME                                    VARCHAR2(20)
```

DESC BOOK_AUTHORS;

```
SQL> DESC BOOK_AUTHORS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 AUTHOR_NAME                               NOT NULL VARCHAR2(20)
 BOOK_ID                                   NOT NULL NUMBER(38)
```

DESC LIBRARY_BRANCH;

```
SQL> DESC LIBRARY_BRANCH;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BRANCH_ID                                 NOT NULL NUMBER(38)
 BRANCH_NAME                                        VARCHAR2(50)
 ADDRESS                                            VARCHAR2(50)
```

DESC BOOK_COPIES;

```
SQL> DESC BOOK_COPIES;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NO_OF_COPIES                                       NUMBER(38)
 BOOK_ID                                   NOT NULL NUMBER(38)
 BRANCH_ID                                 NOT NULL NUMBER(38)
```

DESC CARD;

```
SQL> DESC CARD;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CARD_NO                                   NOT NULL NUMBER(38)
```

DESC BOOK_LENDING;

```
SQL> desc book_lending;
 Name
 ----------------------------------------------------------------------------------------------------
 DATE_OUT
 DUE_DATE
 BOOK_ID
 BRANCH_ID
 CARD_NO
```

Insertion of Values to Tables

```
INSERT INTO PUBLISHER VALUES (_MCGRAW-HILL‘, 9989076587, _BANGALORE‘);
INSERT INTO PUBLISHER VALUES (_PEARSON‘, 9889076565, _NEWDELHI‘);
INSERT INTO PUBLISHER VALUES (_RANDOM HOUSE‘, 7455679345, _HYDRABAD‘); INSERT
INTO PUBLISHER VALUES (_HACHETTE LIVRE‘, 8970862340, _CHENAI‘);
INSERTINTOPUBLISHERVALUES(_GRUPOPLANETA‘,7756120238,_BANGALORE‘);

INSERT INTO BOOK VALUES (1,‘DBMS‘,‘JAN-2017‘, _MCGRAW-HILL‘); INSERT INTO
BOOK VALUES (2,‘ADBMS‘,‘JUN-2016‘, _MCGRAW-HILL‘); INSERT INTO BOOK
VALUES (3,‘CN‘,‘SEP-2016‘, _PEARSON‘);
INSERT INTO BOOK VALUES (4,‘CG‘,‘SEP-2015‘, _GRUPO PLANETA‘); INSERT
INTO BOOK VALUES (5,‘OS‘,‘MAY-2016‘, _PEARSON‘);

INSERT INTO BOOK_AUTHORS VALUES (‘NAVATHE‘, 1); INSERT INTO
BOOK_AUTHORS VALUES (‘NAVATHE‘, 2); INSERT INTO
BOOK_AUTHORS VALUES (‘TANENBAUM‘, 3); INSERT INTO
BOOK_AUTHORS VALUES (‘EDWARD ANGEL‘, 4); INSERT INTO
BOOK_AUTHORS VALUES (‘GALVIN‘, 5);

INSERT INTO LIBRARY_BRANCH VALUES (10,‘RR NAGAR‘,‘BANGALORE‘); INSERT
INTO LIBRARY_BRANCH VALUES (11,‘RNSIT‘,‘BANGALORE‘);
INSERT INTO LIBRARY_BRANCH VALUES (12,‘RAJAJI NAGAR‘, ‘BANGALORE‘); INSERT INTO
LIBRARY_BRANCH VALUES (13,‘NITTE‘,‘MANGALORE‘);
INSERT INTO LIBRARY_BRANCH VALUES (14,‘MANIPAL‘,‘UDUPI‘);

INSERT INTO BOOK_COPIES VALUES (10, 1, 10);
INSERT INTO BOOK_COPIES VALUES (5, 1,11);
INSERT INTO BOOK_COPIES VALUES (2, 2,12);
INSERT INTO BOOK_COPIES VALUES (5, 2,13);
INSERT INTO BOOK_COPIES VALUES (7, 3,14);
INSERT INTO BOOK_COPIES VALUES (1, 5,10);
INSERT INTO BOOK_COPIES VALUES (3, 4,11);

INSERT INTO CARD VALUES (100);
INSERT INTO CARD VALUES (101);
INSERT INTO CARD VALUES (102);
INSERT INTO CARD VALUES (103);
INSERT INTO CARD VALUES (104);
```

INSERT INTO BOOK_LENDING VALUES (`01-JAN-17`,`01-JUN-17`, 1, 10, 101);
INSERT INTO BOOK_LENDING VALUES (`11-JAN-17`,`11-MAR-17`, 3, 14, 101);
INSERT INTO BOOK_LENDING VALUES (`21-FEB-17`,`21-APR-17`, 2, 13,101);
INSERT INTO BOOK_LENDING VALUES (`15-MAR-17`,`15-JUL-17`, 4, 11,101);
INSERT INTO BOOK_LENDING VALUES (_12-APR-17`,`12-MAY-17`, 1, 11, 104);
SELECT * FROM PUBLISHER;

```
SQL> select * from publisher;

NAME                      PHONE ADDRESS
-------------------- ---------- --------------------
MCGRAW-HILL          9989076587 BANGALORE
PEARSON              9889076565 NEWDELHI
RANDOM HOUSE         7455679345 HYDRABAD
HACHETTE LIVRE       8970862340 CHENAI
GRUPO PLANETA        7756120238 BANGALORE
```

SELEvCT * FROM BOOK;

```
SQL> SELECT * FROM BOOK;

   BOOK_ID TITLE                PUB_YEAR             PUBLISHER_NAME
---------- -------------------- -------------------- --------------------
         1 DBMS                 JAN-2017             MCGRAW-HILL
         2 ADBMS                JUN-2016             MCGRAW-HILL
         3 CN                   SEP-2016             PEARSON
         4 CG                   SEP-2015             GRUPO PLANETA
         5 OS                   MAY-2016             PEARSON
```

SELECT * FROM BOOK_AUTHORS;

```
SQL> SELECT * FROM BOOK_AUTHORS;

AUTHOR_NAME             BOOK_ID
-------------------- ----------
NAVATHE                      1
NAVATHE                      2
TANENBAUM                    3
EDWARD ANGEL                 4
GALVIN                       5
```

SELECT * FROM LIBRARY_BRANCH;

```
SQL> SELECT * FROM LIBRARY_BRANCH;

 BRANCH_ID BRANCH_NAME                                        ADDRESS
---------- -------------------------------------------------- ------------------------------------
        10 RR NAGAR                                           BANGALORE
        11 RNSIT                                              BANGALORE
        12 RAJAJI NAGAR                                       BANGALORE
        13 NITTE                                              MANGALORE
        14 MANIPAL                                            UDUPI
```

SELECT * FROM BOOK_COPIES;

```
SQL> SELECT * FROM BOOK_COPIES;

NO_OF_COPIES    BOOK_ID  BRANCH_ID
------------ ---------- ----------
          10          1         10
           5          1         11
           2          2         12
           5          2         13
           7          3         14
           1          5         10
           3          4         11
```

SELECT * FROM CARD;

```
SQL> SELECT * FROM CARD;

   CARD_NO
----------
       100
       101
       102
       103
       104
```

SELECT * FROM BOOK_LENDING;

```
SQL> select * from book_lending;

DATE_OUT  DUE_DATE     BOOK_ID  BRANCH_ID    CARD_NO
--------- --------- ---------- ---------- ----------
01-JAN-17 01-JUN-17          1         10        101
11-JAN-17 11-MAR-17          3         14        101
21-FEB-17 21-APR-17          2         13        101
15-MAR-17 15-JUL-17          4         11        101
12-APR-17 12-MAY-17          1         11        104
```

Queries:

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

   ```
   SELECT B.BOOK_ID, B.TITLE, B.PUBLISHER_NAME, A.AUTHOR_NAME,
   C.NO_OF_COPIES,L.BRANCH_ID
   FROM BOOK B, BOOK_AUTHORS A, BOOK_COPIES C, LIBRARY_BRANCHL
   WHEREB.BOOK_ID=A.BOOK_ID
   AND B.BOOK_ID=C.BOOK_ID
   AND L.BRANCH_ID=C.BRANCH_ID;
   ```

BOOK_ID	TITLE	PUBLISHER_NAME	AUTHOR_NAME	NO_OF_COPIES	BRANCH_ID
1	DBMS	MCGRAW-HILL	NAVATHE	10	10
1	DBMS	MCGRAW-HILL	NAVATHE	5	11
2	ADBMS	MCGRAW-HILL	NAVATHE	2	12
2	ADBMS	MCGRAW-HILL	NAVATHE	5	13
3	CN	PEARSON	TANENBAUM	7	14
5	OS	PEARSON	GALVIN	1	10
4	CG	GRUPO PLANETA	EDWARD ANGEL	3	11

1. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun2017.

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

CARD_NO
101

2. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulationoperation.

```
DELETE FROM BOOK
WHERE BOOK_ID=3;
```

```
SQL> DELETE FROM BOOK
  2  WHERE BOOK_ID=3;

1 row deleted.

SQL> SELECT * FROM BOOK;
```

BOOK_ID	TITLE	PUB_YEAR	PUBLISHER_NAME
1	DBMS	JAN-2017	MCGRAW-HILL
2	ADBMS	JUN-2016	MCGRAW-HILL
4	CG	SEP-2015	GRUPO PLANETA
5	OS	MAY-2016	PEARSON

.

3. Partition the BOOK table based on year of publication. Demonstrate its working with a simplequery.

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

```
PUB_YEAR
-------------
JAN-2017
JUN-2016
SEP-2016
SEP-2015
MAY-2016
```

4. 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_BRANCH L
WHERE B.BOOK_ID=C.BOOK_ID
AND C.BRANCH_ID=L.BRANCH_ID;
```

BOOK_ID	TITLE	NO_OF_COPIES
1	DBMS	10
1	DBMS	5
2	ADBMS	2
2	ADBMS	5
3	CN	7
5	OS	1
4	CG	3

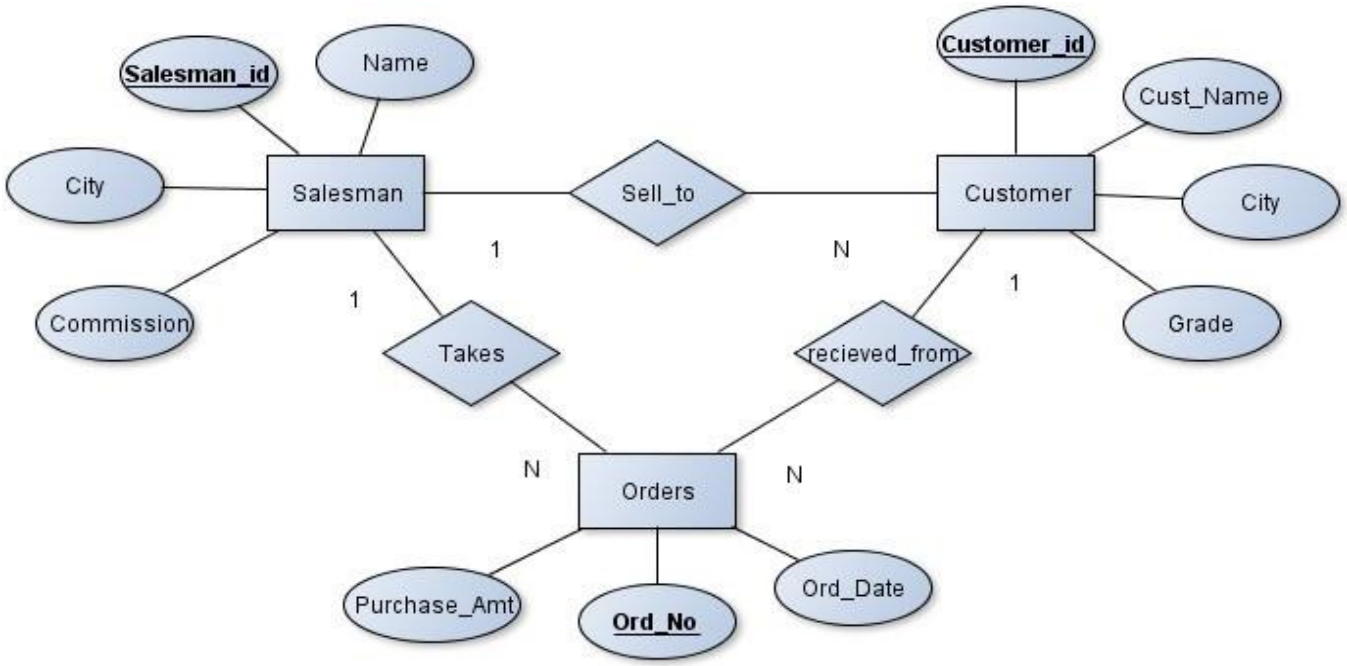
B. Consider the following schema for OrderDatabase:

SALESMAN (*Salesman_id, Name, City, Commission*)
CUSTOMER (*Customer_id, Cust_Name, City, Grade,Salesman_id*)
ORDERS (*Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id*) Write SQL queries to

1. Count the customers with grades above Bangalore'saverage.
2. Find the name and numbers of all salesmen who had more than onecustomer.
3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNIONoperation.)
4. Create a view that finds the salesman who has the customer with the highest order of a day.
5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also bedeleted.

Solution:

Entity-Relationship Diagram



Schema Diagram

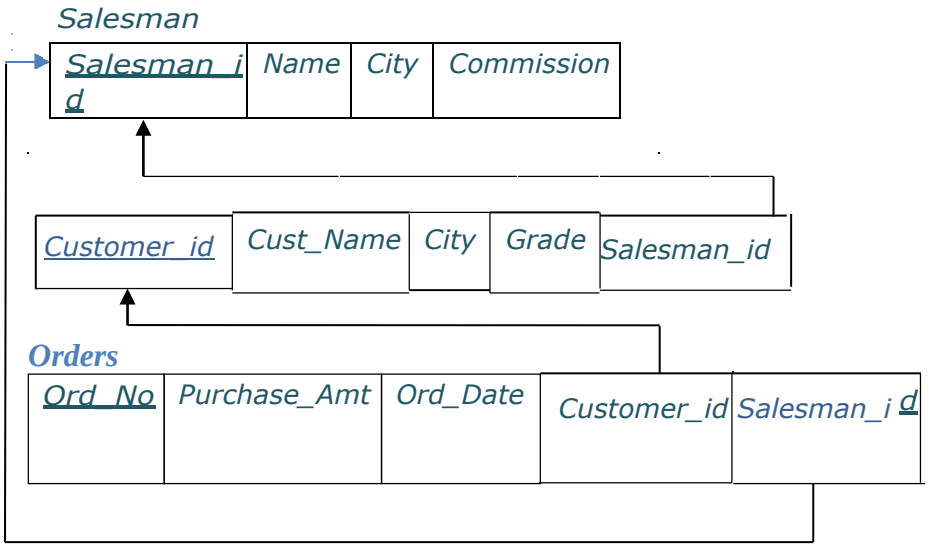


Table Creation

```
CREATE TABLE SALESMAN
(SALESMAN_ID NUMBER (4),
NAME VARCHAR2 (20),
CITY VARCHAR2 (20),
COMMISSION VARCHAR2 (20),
PRIMARYKEY          (SALESMAN_ID));

CREATE TABLE CUSTOMER1
(CUSTOMER_ID NUMBER (4),
CUST_NAME VARCHAR2 (20),
CITY VARCHAR2 (20),
GRADE NUMBER (3),
PRIMARY KEY (CUSTOMER_ID),
SALESMAN_ID REFERENCES SALESMAN (SALESMAN_ID) ON DELETE SET NULL);

CREATE TABLE ORDERS
(ORD_NO NUMBER (5),
PURCHASE_AMT NUMBER (10, 2),
ORD_DATE DATE,
PRIMARY KEY (ORD_NO),
CUSTOMER_ID REFERENCES CUSTOMER1 (CUSTOMER_ID) ON DELETE CASCADE,
SALESMAN_ID REFERENCES SALESMAN (SALESMAN_ID) ON DELETE CASCADE);
```

Table Descriptions

DESC SALESMAN;

```
SQL> DESC SALESMAN;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SALESMAN_ID                               NOT NULL NUMBER(4)
 NAME                                               VARCHAR2(15)
 CITY                                               VARCHAR2(15)
 COMMISSION                                         NUMBER(3,2)
```

DESC CUSTOMER1;

```
SQL> DESC CUSTOMER1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL NUMBER(4)
 CUST_NAME                                          VARCHAR2(15)
 CITY                                               VARCHAR2(15)
 GRADE                                              NUMBER(3)
 SALESMAN_ID                                        NUMBER(4)
```

DESC ORDERS;

```
SQL> DESC ORDERS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORD_NO                                    NOT NULL NUMBER(5)
 PURCHASE_AMT                                       NUMBER(10,2)
 ORD_DATE                                           DATE
 CUSTOMER_ID                                        NUMBER(4)
 SALESMAN_ID                                        NUMBER(4)
```

Insertion of Values to Tables

INSERT INTO SALESMAN VALUES (1000, _JOHN‘,‘BANGALORE‘,‘25 %‘); INSERT INTO SALESMAN VALUES (2000, _RAVI‘,‘BANGALORE‘,‘20 %‘); INSERT INTO SALESMAN VALUES (3000, _KUMAR‘,‘MYSORE‘,‘15 %‘); INSERT INTO SALESMAN VALUES (4000, _SMITH‘,‘DELHI‘,‘30 %‘);
INSERTINTOSALESMANVALUES(5000,_HARSHA‘,‘HYDRABAD‘,‘15%‘);

INSERT INTO CUSTOMER1 VALUES (10, _PREETHI‘,‘BANGALORE‘, 100, 1000);
INSERT INTO CUSTOMER1 VALUES (11, _VIVEK‘,‘MANGALORE‘, 300, 1000);
INSERT INTO CUSTOMER1 VALUES (12, _BHASKAR‘,‘CHENNAI‘, 400, 2000);
INSERT INTO CUSTOMER1 VALUES (13, _CHETHAN‘,‘BANGALORE‘, 200, 2000);
INSERT INTO CUSTOMER1 VALUES (14, _MAMATHA‘,‘BANGALORE‘, 400, 3000);

INSERT INTO ORDERS VALUES (50, 5000, _04-MAY-17‘, 10, 1000);
INSERT INTO ORDERS VALUES (51, 450, _20-JAN-17‘, 10, 2000);

```
INSERTINTOORDERSVALUES(52,1000,_24-FEB-17',13,2000);
INSERTINTOORDERSVALUES(53,3500,_13-APR-17',14,3000);
INSERT INTO ORDERS VALUES (54, 550, _09-MAR-17', 12, 2000);
```

SELECT * FROM SALESMAN;

SALESMAN_ID	NAME	CITY	COMMISSION
1000	JOHN	BANGALORE	25 %
2000	RAVI	BANGALORE	20 %
3000	KUMAR	MYSORE	15 %
4000	SMITH	DELHI	30 %
5000	HARSHA	HYDRABAD	15 %

CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
10	PREETHI	BANGALORE	100	1000
11	VIVEK	MANGALORE	300	1000
12	BHASKAR	CHENNAI	400	2000
13	CHETHAN	BANGALORE	200	2000
14	MAMATHA	BANGALORE	400	3000

SELECT * FROM ORDERS;

ORD_NO	PURCHASE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
50	5000	04-MAY-17	10	1000
51	450	20-JAN-17	10	2000
52	1000	24-FEB-17	13	2000
53	3500	13-APR-17	14	3000
54	550	09-MAR-17	12	2000

Queries:

1. Count the customers with grades above Bangalore'saverage. SELECT GRADE, COUNT(DISTINCT CUSTOMER_ID) FROMCUSTOMER1
   GROUP BY GRADE
   HAVING GRADE > (SELECT AVG(GRADE)
   FROM CUSTOMER1
   WHERE CITY='BANGALORE');

GRADE	COUNT(DISTINCTCUSTOMER_ID)
300	1
400	2

2. Find the name and numbers of all salesmen who had more than onecustomer.

```
SELECT SALESMAN_ID, NAME FROM
 SALESMAN A
 WHERE 1 < (SELECT COUNT (*)
        FROM CUSTOMER1
 WHERE SALESMAN_ID=A.SALESMAN_ID);
```

```
SALESMAN_ID NAME
----------- --------------------
       1000 JOHN
       2000 RAVI
```

3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNIONoperation.)

```
SELECT SALESMAN.SALESMAN_ID, NAME, CUST_NAME, COMMISSION FROM
SALESMAN, CUSTOMER1
WHERE SALESMAN.CITY = CUSTOMER1.CITY
UNION
SELECT SALESMAN_ID, NAME, 'NO MATCH', COMMISSION FROM
SALESMAN
WHERE NOT CITY = ANY
 (SELECT CITY
  FROM CUSTOMER1)
ORDER BY 2 DESC;
```

```
SALESMAN_ID NAME                 CUST_NAME            COMMISSION
----------- -------------------- -------------------- --------------------
       4000 SMITH                NO MATCH             30 %
       2000 RAVI                 CHETHAN              20 %
       2000 RAVI                 MAMATHA              20 %
       2000 RAVI                 PREETHI              20 %
       3000 KUMAR                NO MATCH             15 %
       1000 JOHN                 CHETHAN              25 %
       1000 JOHN                 MAMATHA              25 %
       1000 JOHN                 PREETHI              25 %
       5000 HARSHA               NO MATCH             15 %
```

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

```
CREATE VIEW ELITSALESMAN AS
SELECT B.ORD_DATE, A.SALESMAN_ID, A.NAME FROM
SALESMAN A, ORDERS B
```

```
WHERE A.SALESMAN_ID = B.SALESMAN_ID
AND B.PURCHASE_AMT=(SELECT MAX (PURCHASE_AMT)
                                    FROM ORDERS C
                                        WHERE C.ORD_DATE = B.ORD_DATE);
```

```
ORD_DATE  SALESMAN_ID NAME
--------- ----------- --------------------
04-MAY-17        1000 JOHN
20-JAN-17        2000 RAVI
24-FEB-17        2000 RAVI
13-APR-17        3000 KUMAR
09-MAR-17        2000 RAVI
```

5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also bedeleted.

Use ON DELETE CASCADE at the end of foreign key definitions while creating child table orders and then execute the following:

Use ON DELETE SET NULL at the end of foreign key definitions while creating child table customers and then executes the following:

```
DELETE FROM SALESMAN
WHERE SALESMAN_ID=1000;
```

```
SQL> DELETE FROM SALESMAN
  2  WHERE SALESMAN_ID=1000;

1 row deleted.

SQL> SELECT * FROM SALESMAN;

SALESMAN_ID NAME                 CITY                 COMMISSION
----------- -------------------- -------------------- ----------
       2000 RAVI                 BANGALORE            20 %
       3000 KUMAR                MYSORE               15 %
       4000 SMITH                DELHI                30 %
       5000 HARSHA               HYDRABAD             15 %
```

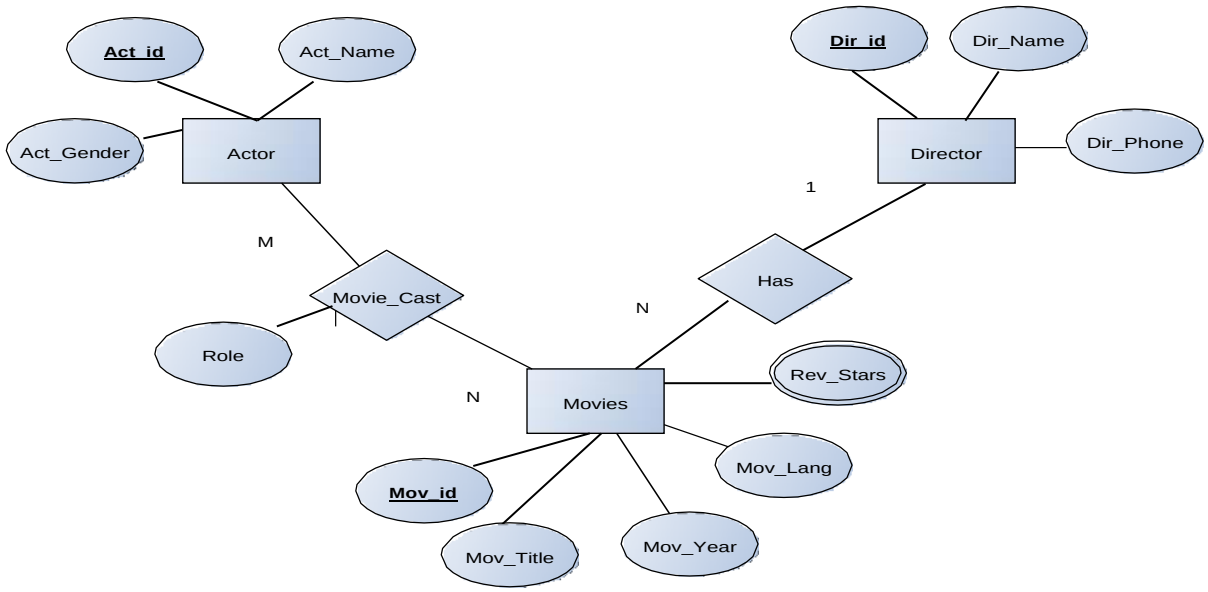
C. Consider the schema for MovieDatabase:

ACTOR (*Act_id, Act_Name, Act_Gender*)
DIRECTOR (*Dir_id, Dir_Name,*
Dir_Phone)
MOVIES (*Mov_id, Mov_Title, Mov_Year, Mov_Lang,*
Dir_id) MOVIE_CAST (*Act_id, Mov_id, Role*)
RATING (*Mov_id,*
Rev_Stars) Write SQL queries to

1. List the titles of all movies directed by 'Hitchcock'.
2. Find the movie names where one or more actors acted in two or more movies.
3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).
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.
5. Update rating of all movies directed by 'Steven Spielberg' to 5.

Solution:

Entity-Relationship Diagram



Schema Diagram

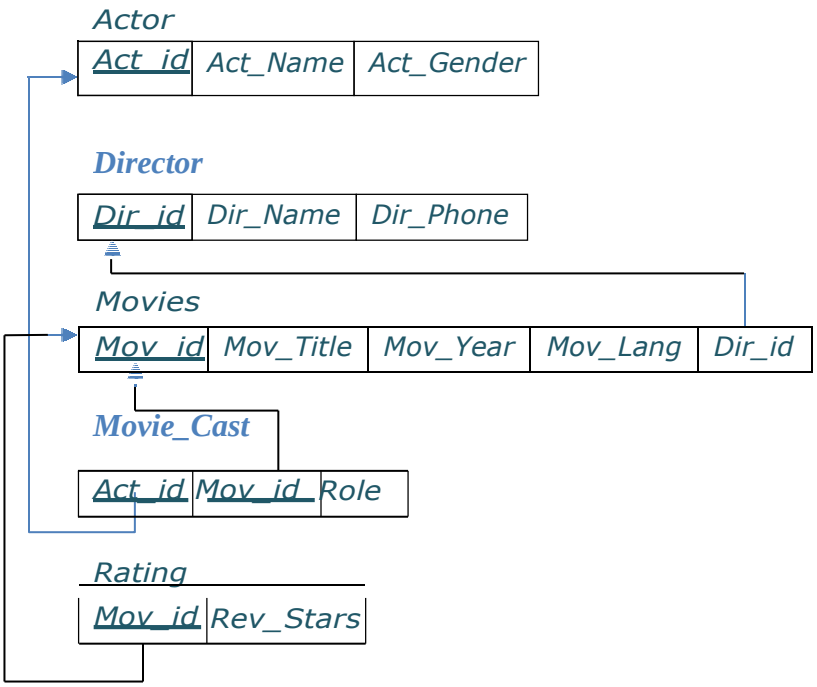


Table Creation

```
CREATE TABLE ACTOR (
ACT_ID NUMBER (3),
ACT_NAME VARCHAR (20),
ACT_GENDER CHAR (1),
PRIMARY KEY (ACT_ID));

CREATE TABLE DIRECTOR (
DIR_ID NUMBER (3),
DIR_NAME VARCHAR (20),
DIR_PHONE NUMBER (10),
PRIMARY KEY (DIR_ID));

CREATE TABLE MOVIES (
MOV_ID NUMBER (4),
MOV_TITLE VARCHAR (25),
MOV_YEAR NUMBER (4),
MOV_LANG VARCHAR (12),
DIR_ID NUMBER (3),
PRIMARY KEY (MOV_ID),
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR (DIR_ID));
```

```
CREATE TABLE MOVIE_CAST ( ACT_ID
NUMBER (3),
MOV_ID NUMBER(4),
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));

CREATE TABLE RATING (
MOV_ID NUMBER (4),
REV_STARS VARCHAR (25),
PRIMARY KEY (MOV_ID),
FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID));
```

Table Descriptions

DESC ACTOR;

```
SQL> DESC ACTOR;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACT_ID                                    NOT NULL NUMBER(3)
 ACT_NAME                                           VARCHAR2(20)
 ACT_GENDER                                         CHAR(1)
```

DESC DIRECTOR;

```
SQL> DESC DIRECTOR;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DIR_ID                                    NOT NULL NUMBER(3)
 DIR_NAME                                           VARCHAR2(20)
 DIR_PHONE                                          NUMBER(10)
```

DESC MOVIES;

```
SQL> DESC MOVIES;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MOV_ID                                    NOT NULL NUMBER(4)
 MOV_TITLE                                          VARCHAR2(25)
 MOV_YEAR                                           NUMBER(4)
 MOV_LANG                                           VARCHAR2(12)
 DIR_ID                                             NUMBER(3)
```

DESC MOVIE_CAST;

```
SQL> DESC MOVIE_CAST;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACT_ID                                    NOT NULL NUMBER(3)
 MOV_ID                                    NOT NULL NUMBER(4)
 ROLE                                               VARCHAR2(10)
```

DESC RATING;

```
SQL> DESC RATING;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MOV_ID                                    NOT NULL NUMBER(4)
 REV_STARS                                          VARCHAR2(25)
```

Insertion of Values to Tables

```
INSERT INTO ACTOR VALUES (301,'ANUSHKA','F');
INSERT INTO ACTOR VALUES (302,'PRABHAS','M');
INSERT INTO ACTOR VALUES (303,'PUNITH','M');
INSERT INTO ACTOR VALUES (304,'JERMY','M');

INSERT INTO DIRECTOR VALUES (60,'RAJAMOULI', 8751611001);
INSERT INTO DIRECTOR VALUES (61,'HITCHCOCK', 7766138911);
INSERT INTO DIRECTOR VALUES (62,'FARAN', 9986776531);
INSERT INTO DIRECTOR VALUES (63,'STEVEN SPIELBERG', 8989776530);

INSERT INTO MOVIES VALUES (1001,'BAHUBALI-2', 2017, _TELAGU', 60);
INSERT INTO MOVIES VALUES (1002,'BAHUBALI-1', 2015, _TELAGU', 60);
INSERT INTO MOVIES VALUES (1003,'AKASH', 2008, _KANNADA', 61); INSERT
INTO MOVIES VALUES (1004,'WAR HORSE', 2011, _ENGLISH', 63);

INSERT INTO MOVIE_CAST VALUES (301, 1002, _HEROINE'); INSERT
INTO MOVIE_CAST VALUES (301, 1001, _HEROINE'); INSERT INTO
MOVIE_CAST VALUES (303, 1003, _HERO'); INSERT INTO
MOVIE_CAST VALUES (303, 1002, _GUEST'); INSERT INTO
MOVIE_CAST VALUES (304, 1004, _HERO');

INSERT INTO RATING VALUES (1001,4);
INSERT INTO RATING VALUES (1002,2);
```

```
INSERT INTO RATING VALUES (1003, 5);
INSERT INTO RATING VALUES (1004, 4);

SELECT * FROM ACTOR;
```

```
SQL> SELECT * FROM ACTOR;

    ACT_ID ACT_NAME              A
---------- -------------------- -
       301 ANUSHKA              F
       302 PRABHAS              M
       303 PUNITH               M
       304 JERMY                M
```

```
SELECT * FROM DIRECTOR;
```

```
SQL> SELECT * FROM DIRECTOR;

    DIR_ID DIR_NAME              DIR_PHONE
---------- -------------------- ----------
        60 RAJAMOULI            8751611001
        61 HITCHCOCK            7766138911
        62 FARAN                9986776531
        63 STEVEN SPIELBERG     8989776530
```

```
SELECT * FROM MOVIES;
```

```
SQL> SELECT * FROM MOVIES;

    MOV_ID MOV_TITLE                MOV_YEAR MOV_LANG         DIR_ID
---------- ------------------------ ---------- ------------ ----------
      1001 BAHUBALI-2                     2017 TELAGU               60
      1002 BAHUBALI-1                     2015 TELAGU               60
      1003 AKASH                          2008 KANNADA              61
      1004 WAR HORSE                      2011 ENGLISH              63
```

```
SELECT * FROM MOVIE_CAST;
```

```
SQL> SELECT * FROM MOVIE_CAST;

    ACT_ID     MOV_ID ROLE
---------- ---------- ----------
       301       1002 HEROINE
       301       1001 HEROINE
       303       1003 HERO
       303       1002 GUEST
       304       1004 HERO
```

SELECT * FROM RATING;

```
SQL> SELECT * FROM RATING;

    MOV_ID REV_STARS
---------- -----------
      1001 4
      1002 2
      1003 5
      1004 4
```

Queries:

1. List the titles of all movies directed by'Hitchcock'.

```
SELECT MOV_TITLE
FROM MOVIES
WHERE DIR_ID IN (SELECT DIR_ID
                 FROM DIRECTOR
                WHERE DIR_NAME = _HITCHCOCK');
```

```
MOV_TITLE
-------------------------
AKASH
```

2. Find the movie names where one or more actors acted in two or moremovies.

```
SELECT MOV_TITLE
FROM MOVIES M, MOVIE_CAST MV
WHERE M.MOV_ID=MV.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;
```

```
MOV_TITLE
------------------------
BAHUBALI-1
```

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

```
SELECT ACT_NAME, MOV_TITLE, MOV_YEAR
```

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

SELECT A.ACT_NAME, A.ACT_NAME, C.MOV_TITLE, C.MOV_YEAR FROM ACTOR
A, MOVIE_CAST B, MOVIES C
WHERE A.ACT_ID=B.ACT_ID
AND B.MOV_ID=C.MOV_ID
AND C.MOV_YEAR NOT BETWEEN 2000 AND 2015;
```

ACT_NAME	MOV_TITLE	MOV_YEAR
ANUSHKA	BAHUBALI-2	2017

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;
```

MOV_TITLE	MAX(REV_STARS)
AKASH	5
BAHUBALI-1	2
BAHUBALI-2	4
WAR HORSE	4

5. Update rating of all movies directed by 'Steven Spielberg' to5
KL

```
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'));
```

```
SQL> SELECT * FROM RATING;

    MOV_ID REV_STARS
---------- -----------------------
      1001 4
      1002 2
      1003 5
      1004 5
```

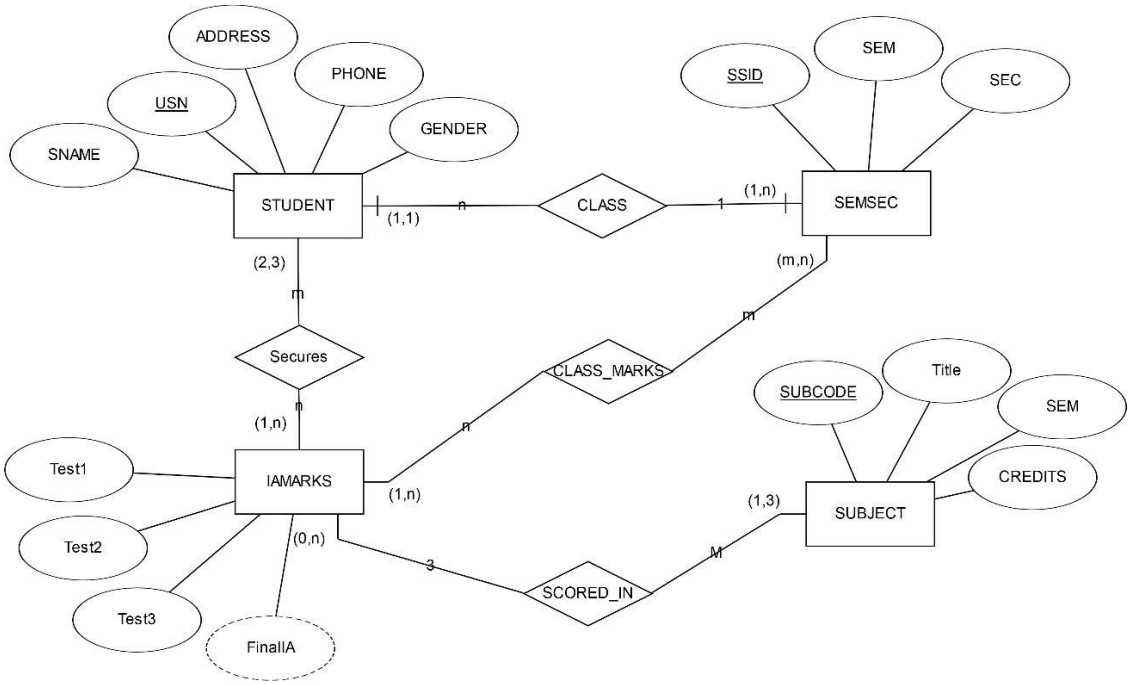
D. Consider the schema for CollegeDatabase:

STUDENT (*USN, SName, Address, Phone, Gender*) SEMSEC (*SSID, Sem, Sec*)
CLASS (*USN, SSID*)
SUBJECT (*Subcode, Title, Sem, Credits*)
IAMARKS (*USN, Subcode, SSID, Test1, Test2, Test3, FinalIA*) Write SQL queries to

1. List all the student details studying in fourth semester 'C'section.
2. Compute the total number of male and female students in each semester and in each section.
3. Create a view of Test1 marks of student USN '1BI15CS101' in allsubjects.
4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for allstudents.
5. Categorize students based on the followingcriterion: 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.

Solution:

Entity - Relationship Diagram



Schema Diagram

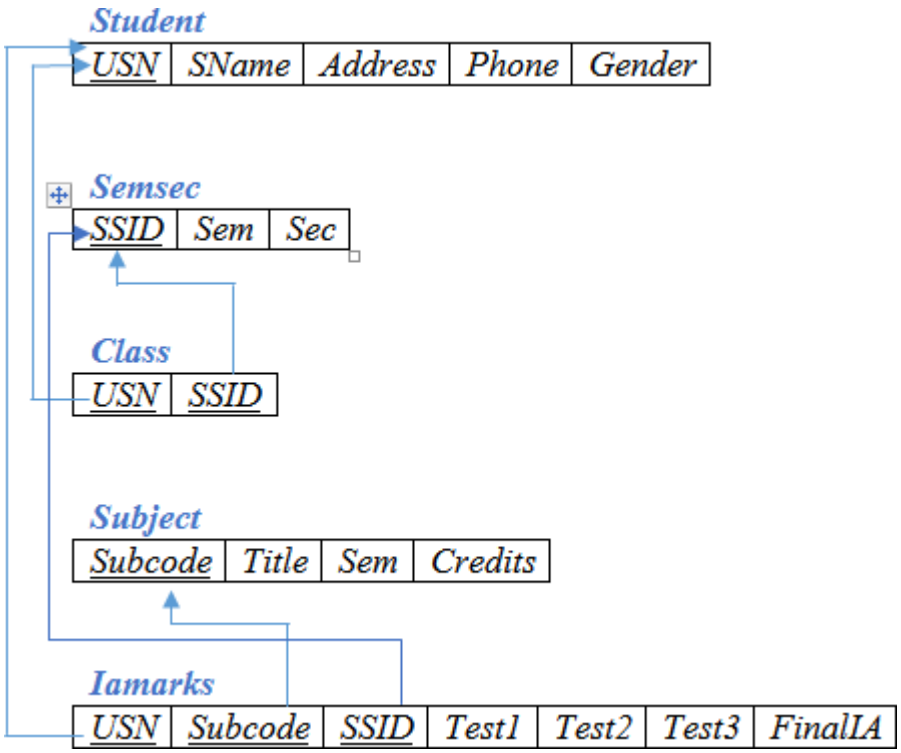


Table Creation

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

CREATE TABLE SEMSEC (
SSID VARCHAR (5) PRIMARY KEY,
SEM NUMBER (2),
SEC CHAR (1));

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

```
CREATE TABLE SUBJECT (
SUBCODE VARCHAR (8),
TITLE VARCHAR (20),
SEM NUMBER (2),
CREDITS NUMBER (2),
PRIMARY KEY (SUBCODE));

 CREATE TABLE IAMARKS (
 USN VARCHAR (10),
 SUBCODE VARCHAR (8),
 SSID VARCHAR(5),
 TEST1 NUMBER(2),
 TEST2 NUMBER(2),
 TEST3 NUMBER(2),
 FINALIA NUMBER (2),
 PRIMARY KEY (USN, SUBCODE, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN),
 FOREIGN KEY (SUBCODE) REFERENCES SUBJECT (SUBCODE),
 FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID));
```

Table Descriptions

DESC STUDENT;

```
Name
--------------------------------------------------------------------------------------------
USN
SNAME
ADDRESS
PHONE
GENDER
```

DESC SEMSEC;

```
SQL> DESC SEMSEC;
 Name
 ------------------------------------------------------------------------------
 SSID
 SEM
 SEC
```

DESC CLASS;

```
SQL> DESC CLASS;
 Name
 -----------------------------------------
 USN
 SSID
```

DESC SUBJECT;

```
SQL> DESC SUBJECT1;
 Name
 ----------------------------------
 SUBCODE
 TITLE
 SEM
 CREDITS
```

DESC IAMARKS;

```
SQL> DESC IAMARKS;
 Name
 -----------------------------------------------
 USN
 SUBCODE
 SSID
 TEST1
 TEST2
 TEST3
 FINALIA
```

Insertion of values to tables

INSERT INTO STUDENT VALUES ('1RN13CS020','AKSHAY','BELAGAVI',
8877881122,'M');
INSERT INTO STUDENT VALUES('1RN13CS062','SANDHYA','BENGALURU',
7722829912,'F');
INSERT INTO STUDENT VALUES('1RN13CS091','TEESHA','BENGALURU',
7712312312,'F');
INSERT INTO STUDENT VALUES('1RN13CS066','SUPRIYA','MANGALURU',
8877881122,'F');
INSERT INTO STUDENTVALUES('1RN14CS010','ABHAY','BENGALURU',
9900211201,'M');
INSERT INTO STUDENT VALUES('1RN14CS032','BHASKAR','BENGALURU',
9923211099,'M');
INSERT INTO STUDENTVALUES ('1RN14CS025','ASMI','BENGALURU', 7894737377,'F');
INSERT INTO STUDENT VALUES ('1RN15CS011','AJAY','TUMKUR', 9845091341,'M');

```
INSERT INTO STUDENT VALUES ('1RN15CS029','CHITRA','DAVANGERE',
7696772121,'F');
INSERT INTO STUDENT VALUES ('1RN15CS045','JEEVA','BELLARY', 9944850121,'M');
INSERT INTO STUDENT VALUES ('1RN15CS091','SANTOSH','MANGALURU',
8812332201,'M');
INSERT INTO STUDENT VALUES('1RN16CS045','ISMAIL','KALBURGI',
9900232201,'M');
INSERT INTO STUDENT VALUES ('1RN16CS088','SAMEERA','SHIMOGA',
9905542212,'F');
INSERT INTO STUDENT VALUES ('1RN16CS122','VINAYAKA','CHIKAMAGALUR',
8800880011,'M');

INSERT INTO SEMSEC VALUES ('CSE8A', 8,'A');
INSERT INTO SEMSEC VALUES (_CSE8B', 8,'B');
INSERTINTOSEMSECVALUES(_CSE8C‘,8,‘C‘);

INSERT INTO SEMSEC VALUES ('CSE7A', 7,‘A‘);
INSERTINTOSEMSECVALUES(_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');

INSERTINTOSEMSECVALUES(_CSE4A‘,4,‘A‘);
INSERT INTO SEMSEC VALUES ('CSE4B', 4,‘B‘);
INSERTINTOSEMSECVALUES(_CSE4C‘,4,'C‘);

INSERT INTO SEMSEC VALUES ('CSE3A', 3,'A');
INSERT INTO SEMSEC VALUES (_CSE3B', 3,'B');
INSERTINTOSEMSECVALUES(_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‘);

INSERTINTOCLASSVALUES(_1RN13CS020‘,‘CSE8A‘);
INSERTINTOCLASSVALUES(_1RN13CS062‘,‘CSE8A‘);
INSERTINTOCLASSVALUES(_1RN13CS066‘,‘CSE8B‘);
INSERTINTOCLASSVALUES(_1RN13CS091‘,‘CSE8C‘);

INSERTINTOCLASSVALUES(_1RN14CS010‘,‘CSE7A‘);
INSERTINTOCLASSVALUES(_1RN14CS025‘,‘CSE7A‘);
INSERTINTOCLASSVALUES(_1RN14CS032‘,‘CSE7A‘);

INSERTINTOCLASSVALUES(_1RN15CS011‘,‘CSE4A‘);
INSERTINTOCLASSVALUES(_1RN15CS029‘,‘CSE4A‘);
INSERTINTOCLASSVALUES(_1RN15CS045‘,‘CSE4B‘);
INSERTINTOCLASSVALUES(_1RN15CS091‘,‘CSE4C‘);

INSERTINTOCLASSVALUES(_1RN16CS045‘,‘CSE3A‘);
INSERTINTOCLASSVALUES(_1RN16CS088‘,‘CSE3B‘);
INSERTINTOCLASSVALUES(_1RN16CS122‘,‘CSE3C‘);

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);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS81','CSE8C', 15, 16,18);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS82','CSE8C', 12, 19,14);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS83','CSE8C', 19, 15,20);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS84','CSE8C', 20, 16,19);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS85','CSE8C', 15, 15,12);

SELECT * FROM STUDENT;
```

```
SQL> SELECT * FROM STUDENT1;

USN        SNAME                     ADDRESS                       PHONE G
---------- ------------------------- ------------------------- ---------- -
1RN13CS020 AKSHAY                    BELAGAVI                  8877881122 M
1RN13CS062 SANDHYA                   BENGALURU                 7722829912 F
1RN13CS091 TEESHA                    BENGALURU                 7712312312 F
1RN13CS066 SUPRIYA                   MANGALURU                 8877881122 F
1RN14CS010 ABHAY                     BENGALURU                 9900211201 M
1RN14CS032 BHASKAR                   BENGALURU                 9923211099 M
1RN15CS011 AJAY                      TUMKUR                    9845091341 M
1RN15CS029 CHITRA                    DAVANGERE                 7696772121 F
1RN15CS045 JEEVA                     BELLARY                   9944850121 M
1RN15CS091 SANTOSH                   MANGALURU                 8812332201 M
1RN16CS045 ISMAIL                    KALBURGI                  9900232201 M
1RN16CS088 SAMEERA                   SHIMOGA                   9905542212 F
1RN16CS122 VINAYAKA                  CHIKAMAGALUR              8800880011 M
1RN14CS025 ASMI                      BENGALURU                 7894737377 F
```

SELECT * FROM SEMSEC;

```
SQL> SELECT * FROM SEMSEC;

SSID         SEM S
----- ---------- -
CSE8A          8 A
CSE8B          8 B
CSE8C          8 C
CSE7A          7 A
CSE7B          7 B
CSE7C          7 C
CSE6A          6 A
CSE6B          6 B
CSE6C          6 C
CSE5A          5 A
CSE5B          5 B
CSE5C          5 C
CSE4A          4 A
CSE4B          4 B
CSE4C          4 C
CSE3A          3 A
CSE3B          3 B
CSE3C          3 C
CSE2A          2 A
CSE2C          2 C
CSE2B          2 B
CSE1A          1 A
CSE1B          1 B
CSE1C          1 C
```

SELECT * FROM CLASS;

```
SQL> SELECT * FROM CLASS;

USN        SSID
---------- -----
1RN13CS020 CSE8A
1RN13CS062 CSE8A
1RN13CS066 CSE8B
1RN13CS091 CSE8C
1RN14CS010 CSE7A
1RN14CS025 CSE7A
1RN14CS032 CSE7A
1RN15CS011 CSE4A
1RN15CS029 CSE4A
1RN15CS045 CSE4B
1RN15CS091 CSE4C
1RN16CS045 CSE3A
1RN16CS088 CSE3B
1RN16CS122 CSE3C

14 rows selected.
```

SELECT * FROM SUBJECT;

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

SELECT * FROM IAMARKS;

```
SQL> SELECT * FROM IAMARKS;
```

USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINALIA
1RN13CS091	10CS81	CSE8C	15	16	18	
1RN13CS091	10CS82	CSE8C	12	19	14	
1RN13CS091	10CS83	CSE8C	19	15	20	
1RN13CS091	10CS84	CSE8C	20	16	19	
1RN13CS091	10CS85	CSE8C	15	15	12	

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';

USN	SNAME	ADDRESS	PHONE	G	SEM	S
1RN15CS091	SANTOSH	MANGALURU	8812332201	M	4	C

2. 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
WHERES.USN = C.USNAND
SS.SSID =C.SSID
GROUP BY SS.SEM, SS.SEC, S.GENDER
ORDER BYSEM;
```

SEM	S	G	COUNT
3	A	M	1
3	B	F	1
3	C	M	1
4	A	F	1
4	A	M	1
4	B	M	1
4	C	M	1
7	A	F	1
7	A	M	2
8	A	F	1
8	A	M	1
8	B	F	1
8	C	F	1

3. Create a view of Test1 marks of student USN '1BI15CS101' in allsubjects.

```
CREATE VIEW STU_TEST1_MARKS_VIEW AS
SELECT TEST1, SUBCODE
FROM IAMARKS
WHERE USN = '1RN13CS091';
```

TEST1	SUBCODE
15	10CS81
12	10CS82
19	10CS83
20	10CS84
15	10CS85

4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for allstudents.

```
CREATE OR REPLACE PROCEDURE AVGMARKS IS
CURSOR C_IAMARKS IS
SELECT GREATEST(TEST1,TEST2) AS A, GREATEST(TEST1,TEST3) AS B,
GREATEST(TEST3,TEST2) ASC
FROM IAMARKS
WHERE FINALIA IS NULL
FOR UPDATE;

  C_ANUMBER;
  C_BNUMBER;
  C_CNUMBER;
  C_SMNUMBER;
  C_AVNUMBER;

BEGIN
  OPEN C_IAMARKS;
  LOOP
  FETCH C_IAMARKS INTO C_A, C_B, C_C;
    EXIT WHEN C_IAMARKS%NOTFOUND;
    --DBMS_OUTPUT.PUT_LINE(C_A || ' ' || C_B || ' ' ||
    C_C); IF (C_A != C_B) THEN
 C_SM:=C_A+C_B;
    ELSE
 C_SM:=C_A+C_C;
    END IF;

    C_AV:=C_SM/2;
    --DBMS_OUTPUT.PUT_LINE('SUM = '||C_SM);
    --DBMS_OUTPUT.PUT_LINE('AVERAGE = '||C_AV);
    UPDATE IAMARKS SET FINALIA=C_AV WHERE CURRENT OF C_IAMARKS;

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

Note: Before execution of PL/SQL procedure, IAMARKS table contents are:

SELECT * FROM IAMARKS;

```
SQL> SELECT * FROM IAMARKS;

USN        SUBCODE  SSID       TEST1      TEST2      TEST3    FINALIA
---------- -------- ----- ---------- ---------- ---------- ----------
1RN13CS091 10CS81   CSE8C          15         16         18
1RN13CS091 10CS82   CSE8C          12         19         14
1RN13CS091 10CS83   CSE8C          19         15         20
1RN13CS091 10CS84   CSE8C          20         16         19
1RN13CS091 10CS85   CSE8C          15         15         12
```

Below SQL code is to invoke the PL/SQL stored procedure from the command line:

```
BEGIN
AVGMARKS;E
ND;
```

```
SQL> select * from IAMARks;

USN        SUBCODE  SSID       TEST1      TEST2      TEST3    FINALIA
---------- -------- ----- ---------- ---------- ---------- ----------
1RN13CS091 10CS81   CSE8C          15         16         18         17
1RN13CS091 10CS82   CSE8C          12         19         14         17
1RN13CS091 10CS83   CSE8C          19         15         20         20
1RN13CS091 10CS84   CSE8C          20         16         19         20
1RN13CS091 10CS85   CSE8C          15         15         12         15
```

.

5. Categorize students based on the followingcriterion: 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.

```
SELECT S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER,
    (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;
```

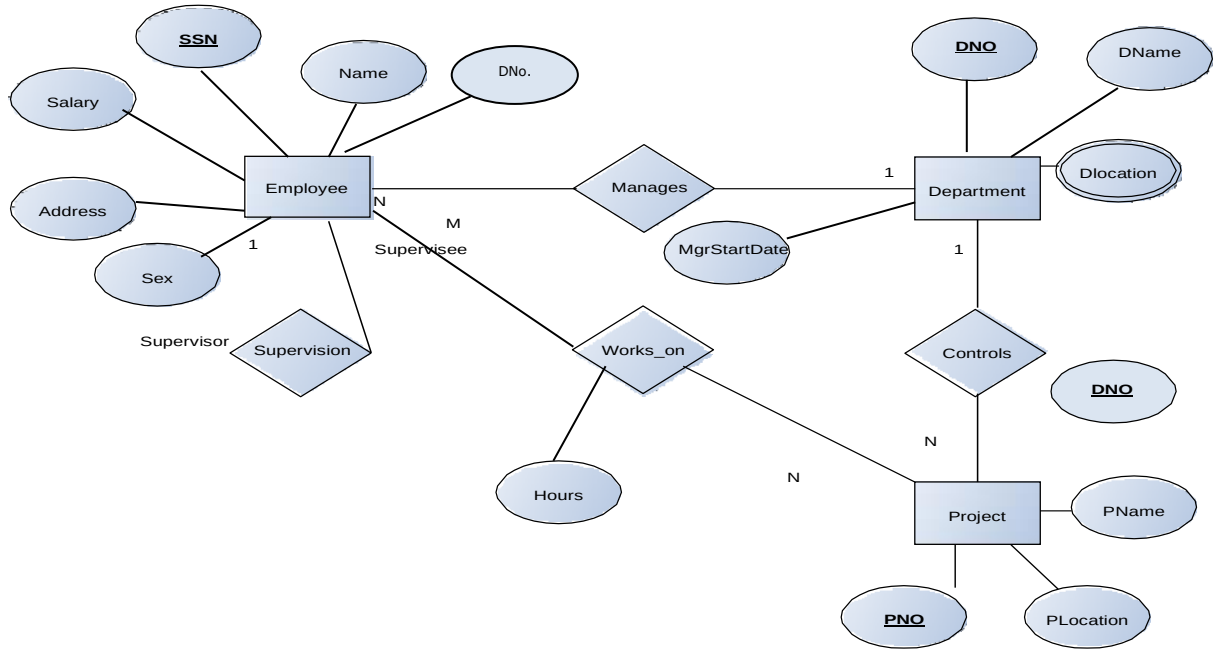
USN	SNAME	ADDRESS	PHONE	G	CAT
1RN13CS091	TEESHA	BENGALURU	7712312312	F	OutStanding
1RN13CS091	TEESHA	BENGALURU	7712312312	F	OutStanding
1RN13CS091	TEESHA	BENGALURU	7712312312	F	OutStanding
1RN13CS091	TEESHA	BENGALURU	7712312312	F	OutStanding
1RN13CS091	TEESHA	BENGALURU	7712312312	F	Average

E. Consider the schema for CompanyDatabase:

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*)
Write SQL queries to

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 theproject.
2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percentraise.
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
4. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator). 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.

Entity-Relationship Diagram



Schema Diagram

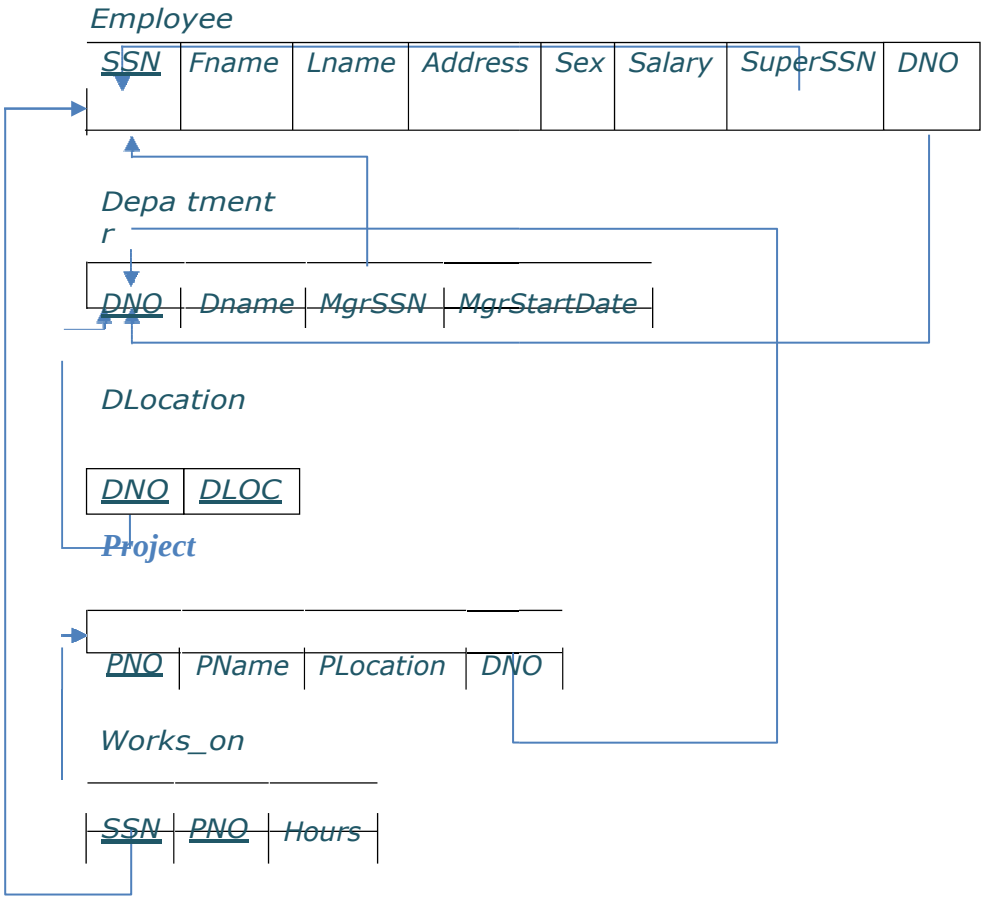


Table Creation

```
CREATE TABLE DEPARTMENT
(DNO VARCHAR2 (20) PRIMARY KEY,
DNAME VARCHAR2 (20), MGRSTARTDATE
DATE);

CREATE TABLE EMPLOYEE
(SSN VARCHAR2 (20) PRIMARYKEY,
FNAME VARCHAR2(20),
LNAME VARCHAR2(20),
ADDRESS VARCHAR2 (20),
SEX CHAR (1), SALARY
INTEGER,
SUPERSSN REFERENCES EMPLOYEE (SSN),
DNO REFERENCES DEPARTMENT (DNO));
```

NOTE: Once DEPARTMENT and EMPLOYEE tables are created we must alter department table to add foreign constraint MGRSSN using sql command

```
ALTER TABLE DEPARTMENT
ADD MGRSSN REFERENCES EMPLOYEE (SSN);
```

```
CREATE TABLE DLOCATION
(DLOC VARCHAR2 (20),
DNO REFERENCES DEPARTMENT (DNO),
PRIMARY KEY (DNO, DLOC));
```

```
CREATE TABLE PROJECT (PNO
INTEGER PRIMARYKEY, PNAME
VARCHAR2(20),
PLOCATION VARCHAR2 (20),
DNO REFERENCES DEPARTMENT (DNO));
```

```
CREATE TABLE WORKS_ON
(HOURS NUMBER (2),
SSN REFERENCES EMPLOYEE (SSN),
PNO REFERENCES PROJECT(PNO),
PRIMARY KEY (SSN, PNO));
```

Table Descriptions

DESC EMPLOYEE;

```
SQL> DESC EMPLOYEE;
 Name
 -----------------------------------------------------------------
 SSN
 FNAME
 LNAME
 ADDRESS
 SEX
 SALARY
 SUPERSSN
 DNO
```

DESC DEPARTMENT;

```
SQL> DESC DEPARTMENT;
 Name
 -----------------------------------------------------
 DNO
 DNAME
 MGRSTARTDATE
 MGRSSN
```

DESC DLOCATION;

```
SQL> DESC DLOCATION;
 Name
 -------------------------
 DLOC
 DNO
```

DESC PROJECT;

```
SQL> DESC PROJECT;
 Name
 -------------------------
 PNO
 PNAME
 PLOCATION
 DNO
```

DESC WORKS_ON;

```
SQL> DESC WORKS_ON;
 Name
 ------------------------------------------------
 HOURS
 SSN
 PNO
```

Insertion of values to tables

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (_RNSECE01‘,‘JOHN‘,‘SCOTT‘,‘BANGALORE‘,‘M‘, 450000);
INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (_RNSCSE01‘,‘JAMES‘,‘SMITH‘,‘BANGALORE‘,‘M‘, 500000);
INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (_RNSCSE02‘,‘HEARN‘,‘BAKER‘,‘BANGALORE‘,‘M‘, 700000);
INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (_RNSCSE03‘,‘EDWARD‘,‘SCOTT‘,‘MYSORE‘,‘M‘, 500000);
INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (_RNSCSE04‘,‘PAVAN‘,‘HEGDE‘,‘MANGALORE‘,‘M‘, 650000);
INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (_RNSCSE05‘,‘GIRISH‘,‘MALYA‘,‘MYSORE‘,‘M‘, 450000);

```
INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES
(_RNSCSE06‘,‘NEHA‘,‘SN‘,‘BANGALORE‘,‘F‘, 800000);
INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES
(_RNSACC01‘,‘AHANA‘,‘K‘,‘MANGALORE‘,‘F‘, 350000);
INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES
(_RNSACC02‘,‘SANTHOSH‘,‘KUMAR‘,‘MANGALORE‘,‘M‘, 300000);
INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES
(_RNSISE01‘,‘VEENA‘,‘M‘,‘MYSORE‘,‘M‘, 600000);
INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES
(_RNSIT01‘,‘NAGESH‘,‘HR‘,‘BANGALORE‘,‘M‘, 500000);


INSERT INTO DEPARTMENT VALUES (_1‘,‘ACCOUNTS‘,‘01-JAN-01‘,‘RNSACC02‘); INSERT
INTO DEPARTMENT VALUES (_2‘,‘IT‘,‘01-AUG-16‘,‘RNSIT01‘);
INSERT INTO DEPARTMENT VALUES (_3‘,‘ECE‘,‘01-JUN-08‘,‘RNSECE01‘); INSERT
INTO DEPARTMENT VALUES (_4‘,‘ISE‘,‘01-AUG-15‘,‘RNSISE01‘); INSERT INTO
DEPARTMENT VALUES (_5‘,‘CSE‘,‘01-JUN-02‘,‘RNSCSE05‘);
```

Note: update entries of employee table to fill missing fields SUPERSSN and DNO

```
UPDATE EMPLOYEE SET
SUPERSSN=NULL,DNO=‘3‘
WHERESSN=‘RNSECE01‘;

UPDATE EMPLOYEE SET
SUPERSSN=‘RNSCSE02‘,DNO=‘5‘
WHERESSN=‘RNSCSE01‘;

UPDATE EMPLOYEE SET
SUPERSSN=‘RNSCSE03‘,DNO=‘5‘
WHERESSN=‘RNSCSE02‘;

UPDATE EMPLOYEE SET
SUPERSSN=‘RNSCSE04‘,DNO=‘5‘
WHERESSN=‘RNSCSE03‘;

UPDATE EMPLOYEE SET DNO=‘5‘,
SUPERSSN=‘RNSCSE05‘
WHERESSN=‘RNSCSE04‘;
```

```
UPDATE EMPLOYEE SET DNO=`5`,
SUPERSSN=`RNSCSE06`
WHERESSN=`RNSCSE05`;

UPDATE EMPLOYEE SET
DNO=`5`, SUPERSSN=NULL
WHERESSN=`RNSCSE06`;

UPDATE EMPLOYEE SET DNO=`1`,
SUPERSSN=`RNSACC02`
WHERESSN=`RNSACC01`;

UPDATE EMPLOYEE SET
DNO=`1`, SUPERSSN=NULL
WHERESSN=`RNSACC02`;

UPDATE EMPLOYEE SET
DNO=`4`, SUPERSSN=NULL
WHERE SSN=`RNSISE01`;

UPDATE EMPLOYEE SET
DNO=`2`, SUPERSSN=NULL
WHERE SSN=`RNSIT01`;

INSERT INTO DLOCATION VALUES (`BANGALORE`, _1`);
INSERT INTO DLOCATION VALUES (`BANGALORE`, _2`);
INSERT INTO DLOCATION VALUES (`BANGALORE`, _3`);
INSERTINTODLOCATIONVALUES(`MANGALORE`,_4`);
INSERTINTODLOCATIONVALUES(`MANGALORE`,_5`);

INSERT INTO PROJECT VALUES (100,`IOT`,`BANGALORE`,`5`); INSERT
INTO PROJECT VALUES (101,`CLOUD`,`BANGALORE`,`5`); INSERT INTO
PROJECT VALUES (102,`BIGDATA`,`BANGALORE`,`5`); INSERT INTO
PROJECT VALUES (103,`SENSORS`,`BANGALORE`,`3`);
INSERT INTO PROJECT VALUES (104,`BANK MANAGEMENT`,`BANGALORE`,`1`); INSERT INTO
PROJECT VALUES (105,`SALARYMANAGEMENT`,`BANGALORE`,`1`); INSERT INTO PROJECT
VALUES(106,`OPENSTACK`,`BANGALORE`,`4`);
INSERT INTO PROJECT VALUES (107,`SMARTCITY`,`BANGALORE`,`2`);
```

```
INSERT INTO WORKS_ON VALUES (4, _RNSCSE01', 100);
INSERT INTO WORKS_ON VALUES (6, _RNSCSE01', 101);
INSERT INTO WORKS_ON VALUES (8, _RNSCSE01', 102);
INSERT INTO WORKS_ON VALUES (10, _RNSCSE02', 100);
INSERT INTO WORKS_ON VALUES (3, _RNSCSE04', 100);
INSERT INTO WORKS_ON VALUES (4, _RNSCSE05', 101);
INSERT INTO WORKS_ON VALUES (5, _RNSCSE06', 102);
INSERT INTO WORKS_ON VALUES (6, _RNSCSE03', 102);
INSERT INTO WORKS_ON VALUES (7, _RNSECE01', 103);
INSERT INTO WORKS_ON VALUES (5, _RNSACC01', 104);
INSERT INTO WORKS_ON VALUES (6, _RNSACC02', 105);
INSERT INTO WORKS_ON VALUES (4, _RNSISE01', 106);
INSERT INTO WORKS_ON VALUES (10, _RNSIT01', 107);
```

```
SELECT * FROM EMPLOYEE;
```

SSN	FNAME	LNAME	ADDRESS	S	SALARY	SUPERSSN	DNO
RNSECE01	JOHN	SCOTT	BANGALORE	M	450000		3
RNSCSE01	JAMES	SMITH	BANGALORE	M	500000	RNSCSE02	5
RNSCSE02	HEARN	BAKER	BANGALORE	M	700000	RNSCSE03	5
RNSCSE03	EDWARD	SCOTT	MYSORE	M	500000	RNSCSE04	5
RNSCSE04	PAVAN	HEGDE	MANGALORE	M	650000	RNSCSE05	5
RNSCSE05	GIRISH	MALYA	MYSORE	M	450000	RNSCSE06	5
RNSCSE06	NEHA	SN	BANGALORE	F	800000		5
RNSACC01	AHANA	K	MANGALORE	F	350000	RNSACC02	1
RNSACC02	SANTHOSH	KUMAR	MANGALORE	M	300000		1
RNSISE01	VEENA	M	MYSORE	M	600000		4
RNSIT01	NAGESH	HR	BANGALORE	M	500000		2

```
SELECT * FROM DEPARTMENT;
```

```
SQL> SELECT * FROM DEPARTMENT;
```

DNO	DNAME	MGRSTARTD	MGRSSN
1	ACCOUNTS	01-JAN-01	RNSACC02
2	IT	01-AUG-16	RNSIT01
3	ECE	01-JUN-08	RNSECE01
4	ISE	01-AUG-15	RNSISE01
5	CSE	01-JUN-02	RNSCSE05

```
SELECT * FROM DLOCATION;
```

DLOC	DNO
BANGALORE	1
BANGALORE	2
BANGALORE	3
MANGALORE	4
MANGALORE	5

SELECT * FROM PROJECT;

PNO	PNAME	PLOCATION	DNO
100	IOT	BANGALORE	5
101	CLOUD	BANGALORE	5
102	BIGDATA	BANGALORE	5
103	SENSORS	BANGALORE	3
104	BANK MANAGEMENT	BANGALORE	1
105	SALARY MANAGEMENT	BANGALORE	1
106	OPENSTACK	BANGALORE	4
107	SMART CITY	BANGALORE	2

SELECT * FROM WORKS_ON;

HOURS	SSN	PNO
4	RNSCSE01	100
6	RNSCSE01	101
8	RNSCSE01	102
10	RNSCSE02	100
3	RNSCSE04	100
4	RNSCSE05	101
5	RNSCSE06	102
6	RNSCSE03	102
7	RNSECE01	103
5	RNSACC01	104
6	RNSACC02	105
4	RNSISE01	106
10	RNSIT01	107

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 controlsthe project.

```
(SELECT DISTINCT P.PNO
FROM PROJECT P, DEPARTMENT D, EMPLOYEE E
WHERE E.DNO=D.DNO
AND  D.MGRSSN=E.SSN
AND E.LNAME='SCOTT')
UNION
(SELECT DISTINCT P1.PNO
FROM PROJECT P1, WORKS_ON W, EMPLOYEE E1
WHERE P1.PNO=W.PNO
AND E1.SSN=W.SSN
AND E1.LNAME='SCOTT');
```

```
     PNO
--------
     100
     101
     102
     103
     104
     105
     106
     107
```

2. Show the resulting salaries if every employee working on the 'IoT' project is given a10 percentraise.

```
SELECT E.FNAME, E.LNAME, 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';
```

```
FNAME                LNAME                INCR_SAL
-------------------- -------------------- ----------
JAMES                SMITH                   550000
HEARN                BAKER                   770000
PAVAN                HEGDE                   715000
```

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 thisdepartment

```
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';
```

```
SUM(E.SALARY) MAX(E.SALARY) MIN(E.SALARY) AVG(E.SALARY)
------------- ------------- ------------- -------------
       650000        350000        300000        325000
```

4. Retrieve the name of each employee who works on all the projects Controlled by department number 5 (use NOT EXISTSoperator).

```
SELECT E.FNAME, E.LNAME
FROM EMPLOYEE E
WHERE NOT EXISTS((SELECT PNO
                  FROM PROJECT
```

```
                    WHERE DNO='5')
                    MINUS (SELECT PNO
                    FROM WORKS_ON
                    WHERE E.SSN=SSN));
```

FNAME	LNAME
JAMES	SMITH

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;
```

DNO	COUNT(*)
5	3

Viva Questions

1. What isSQL?

   Structured Query Language

2. What isdatabase?

   A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specificpurpose.

3. What isDBMS?

   It is a collection of programs that enables user to create and maintain a database. In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

4. What is a Databasesystem?

   The database and DBMS software together is called as Database system.

5. Advantages of DBMS?

   - Redundancy iscontrolled.
   - Unauthorized access isrestricted.
   - Providing multiple userinterfaces.
   - Enforcing integrityconstraints.
   - Providing backup andrecovery.

6. Disadvantage in File ProcessingSystem?

   - Data redundancy &inconsistency.
   - Difficult in accessingdata.
   - Dataisolation.
   - Dataintegrity.
   - Concurrent access is notpossible.
   - SecurityProblems.

7. Describe the three levels of dataabstraction?

   There are three levels of abstraction:

   - Physical level: The lowest level of abstraction describes how data arestored.
   - Logical level*:* The next higher level of abstraction, describes what data are stored in database and what relationship among thosedata.

- View level:The highest level of abstraction describes only part of entiredatabase.

8. Define the "integrityrules"

There are two Integrity rules.

- EntityIntegrity:Statesthat—PrimarykeycannothaveNULLvalue□
- Referential Integrity:States that -Foreign Key can be either a NULL value or should be Primary Key value of otherrelation.

9. What is extension and intension?

Extension - It is the number of tuples present in a table at any instance. This is time dependent.

Intension -It is a constant value that gives the name, structure of table and the constraints laid on it.

10. What is DataIndependence?

Dataindependencemeansthat—theapplicationisindependentofthestoragestructureand access strategy of data□. In other words, The ability to modify the schema definition in onelevel should not affect the schema definition in the next higherlevel.

Two types of Data Independence:

- Physical Data Independence: Modification in physical level should not affect the logicallevel.
- Logical Data Independence: Modification in logical level should affect the view level.

NOTE: Logical Data Independence is more difficult to achieve

11. What is a view? How it is related to dataindependence?

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base table. In other words, there is no stored file that direct represents the view instead a definition of view is stored in data dictionary.

Growth and restructuring of base tables is not reflected in views. Thus the view can insulate users from the effects of restructuring and growth in the database. Hence accounts for logical data independence.

12. What is DataModel?

A collection of conceptual tools for describing data, data relationships data semantics and constraints.

13. What is E-Rmodel?

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

14. What is Object Orientedmodel?

This model is based on collection of objects. An object contains values stored in instance variables within the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

15. What is anEntity?

It is an 'object' in the real world with an independent existence.

16. What is an Entitytype?

It is a collection (set) of entities that have same attributes.

17. What is an Entityset?

It is a collection of all entities of particular entity type in the database.

18. What is an Extension of entity type?

The collections of entities of a particular entity type are grouped together into an entity set.

19. What is anattribute?

It is a particular property, which describes the entity.

20. What is a Relation Schema and aRelation?

A relation Schema denoted by R(A1, A2, …, An) is made up ofthe relation name R and the list of attributes A_i that it contains. A relation isdefined as a set of tuples. Let r be the relation which contains set tuples (t1,t2, t3, ...,tn). Each tuple is an ordered list of n- values t=(v1,v2, ...,vn).

21. What is degree of aRelation?

It is the number of attribute of its relation schema.

22. What isRelationship?

It is an association among two or more entities.

23. What is Relationshipset?

The collection (or set) of similar relationships.

24. *What is Relationshiptype?*

Relationship type defines a set of associations or a relationship set among a given set of entity types.

25. What is degree of Relationshiptype?

It is the number of entity type participating.

26. What is DDL (Data DefinitionLanguage)?

A data base schema is specified by a set of definitions expressed by a special language called DDL.

27. What is VDL (View DefinitionLanguage)?

It specifies user views and their mappings to the conceptual schema.

28. What is SDL (Storage DefinitionLanguage)?

This language is to specify the internal schema. This language may specify the mapping between two schemas.

29. What is Data Storage - DefinitionLanguage?

The storage structures and access methods used by databasesystem are specified by a set of definition in a special type of DDL calleddata storage- definitionlanguage.

30. What is DML (Data ManipulationLanguage)?

This language that enable user to access or manipulate dataasorganized by appropriate datamodel.

- Procedural DML or Low level: DML requires a user to specify what data are needed and how to get thosedata.
- Non-Procedural DML or High level: DML requires a user to specify what data are needed without specifying how to get thosedata.

31. What is DMLCompiler?

It translates DML statements in a query language intolow-level instruction that the query evaluation engine canunderstand.

32. What is RelationalAlgebra?

It is a procedural query language. It consists of a set ofoperationsthat take one or two relations as input and produce a newrelation.

33. What is RelationalCalculus?

It is an applied predicate calculus specifically tailored for relational databases proposed by E.F. Codd. E.g. of languages based on it are DSL, ALPHA,QUEL.

34. What isnormalization?

It is a process of analyzing the given relation schemas basedontheir Functional Dependencies (FDs) and primary key to achieve theproperties

- Minimizingredundancy
- Minimizing insertion, deletion and updateanomalies.

35. What is FunctionalDependency?

A Functional dependency is denotedbyX Y between two sets of attributes X and Y thataresubsetsofRspecifiesaconstraintonthepossibletuplethatcanformarelationstaterof R. The constraint is for any two tuples t1 and t2 in r if t1[X] = t2[X] then they have t1[Y] = t2[Y]. This means the value of X component of a tuple uniquely determines the value of component Y.

36. When is a functional dependency F said to beminimal?

- Every dependency in F has a single attribute for its right handside.
- We cannot replace any dependencyX A in F with adependencyY A where Y is a proper subset of X and still have a set of dependency that is equivalent toF.
- We cannot remove any dependency from F and still have set of dependency that is equivalent toF.

37. What is Multivalueddependency?

Multivalued dependency denoted by X Y specified on relation schema R, where X and Y are both subsets of R, specifies the following constraint on any relation r of R: if two tuples t1 and t2 exist in r such that t1[X] = t2[X] then t3 and t4 should also exist in r with the followingproperties

- t3[x] = t4[X] = t1[X] =t2[X]
- t3[Y] = t1[Y] and t4[Y] =t2[Y]
- t3[Z] = t2[Z] and t4[Z] = t1[Z]

  where [Z = (R-(X U Y))]

38. What is Lossless join property?

It guarantees that the spurious tuple generation does not occur with respect to relation schemas after decomposition.

39. What is 1 NF (NormalForm)?

The domain of attribute must include only atomic (simple, indivisible) values.

40. What is Fully Functionaldependency?

It is based on concept of full functional dependency. A functional dependency X→Y is fully functional dependency if removal of any attribute A from X means that the dependency does not hold anymore.

41. What is2NF?

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

42. What is3NF?

A relation schema R is in 3NF if it is in 2NF and for every FD X →A either of the following istrue

- X is a Super-key ofR.
- A is a prime attribute ofR.

In other words, if every non prime attribute is non-transitively dependent on primary key.

43. What is BCNF (Boyce-Codd NormalForm)?

A relation schema R is in BCNF if it is in 3NF and satisfies additional constraints that for every FD X→ A, X must be a candidate key.

44. What is4NF?

A relation schema R is said to be in 4NF if for every Multivalued dependency X→ Y that holds over R, one of following istrue

- X is subset or equal to (or) XY =R.
- X is a superkey.

45. What is5NF?

A Relation schema R is said to be 5NF if for every join dependency {R1, R2, ...,Rn} that holds R, one the following is true

- Ri = R for somei.
- The join dependency is implied by the set of FD, over R in which the left side is key ofR.

46. What is Domain-Key Normal Form?

A relation is said to be in DKNF if all constraints and dependencies that should hold on the constraint can be enforced by simply enforcing the domain constraint and key constraint on the relation.

47. What are partial, alternate,, artificial, compound and naturalkey?

Partial Key:

It is a set of attributes that can uniquely identify weak entities and that are related to same owner entity. It is sometime called as Discriminator.

Alternate Key:

All Candidate Keys excluding the Primary Key are known as Alternate Keys.

ArtificialKey:

If no obvious key, either stand alone or compound is available, then the last resort is to simply create a key, by assigning a unique number to each record or occurrence. Then this is known as developing an artificial key.

CompoundKey:

If no single data element uniquely identifies occurrences within a construct, then combining multiple elements to create a unique identifier for the construct is known as creating a compound key.

NaturalKey:

When one of the data elements stored within a construct is utilized as the primary key, then it is called the natural key.

48. What is indexing and what are the different kinds ofindexing?

Indexing is a technique for determining how quickly specific data can be found.

- Binary search styleindexing
- B-Treeindexing
- Inverted listindexing
- Memory residenttable
- Table indexing

49. What is system catalog or catalog relation? How is better knownas?

A RDBMS maintains a description of all the data that it contains, information about every relation and index that it contains. This information is stored in a collection of relations maintained by the system called metadata. It is also called datadictionary.

50. What is meant by queryoptimization?

The phase that identifies an efficient execution plan for evaluating a query that has the least estimated cost is referred to as query optimization.

51. What is join dependency and inclusiondependency?

JoinDependency:

A Join dependency is generalization of Multivalued dependency.A JD {R1, R2, ...,Rn} is said to hold over a relation R if R1, R2, R3, ..., Rn is a lossless-join decomposition of R . There is no set of sound and complete inference rules for JD. InclusionDependency:

An Inclusion Dependency is a statement of the form that some columns of a relation are contained in other columns. A foreign key constraint is an example of inclusion dependency.

52. What is durability inDBMS?

Once the DBMS informs the user that a transaction has successfully completed, its effects should persist even if the system crashes before all its changes are reflected on disk. This property is calleddurability.

53. What do you mean by atomicity andaggregation?

Atomicity:

Either all actions are carried out or none are. Users should not have to worry about the effect of incomplete transactions. DBMS ensures this by undoing the actions of incomplete transactions.

Aggregation*:*

A concept which is used to model a relationship between a collection of entities and relationships. It is used when we need to express a relationship among relationships.

54. What is a PhantomDeadlock?

In distributed deadlock detection, the delay in propagating local information might cause the deadlock detection algorithms to identify deadlocks that do not really exist. Such situations are called phantom deadlocks and they lead to unnecessary aborts.

55. What is a checkpoint and when does itoccur?

A Checkpoint is like a snapshot of the DBMS state. By taking checkpoints, the DBMS can reduce the amount of work to be done during restart in the event of subsequent crashes.

56. What are the different phases oftransaction?

Different phases are

- Analysisphase
- RedoPhase
- Undophase

57. What do you mean by flat filedatabase?

It is a database in which there are no programs or user access languages. It has no cross- file capabilities but is user-friendly and provides user-interface management.

58. What is "transparentDBMS"?

It is one, which keeps its Physical Structure hidden from user.

59. Brief theory of Network, Hierarchical schemas and theirproperties

Network schema uses a graph data structure to organize records example for such a database management system is CTCG while a hierarchical schema uses a tree data structure example for such a system is IMS.

60. What is aquery?

A query with respect to DBMS relates to user commands that are used to interact with a data base. The query language can be classified into data definition language and data manipulation language.

61. What do you mean by Correlatedsubquery?

Subqueries, or nested queries, are used to bring back a set of rows to be used by the parentquery.Dependingonhowthesubqueryiswritten,itcanbeexecutedonceforttheparent

query or it can be executed once for each row returned by the parent query. If the subquery is executed for each row of the parent, this is called a *correlated subquery*.

A correlated subquery can be easily identified if it contains any references to the parent subquery columns in its WHERE clause. Columns from the subquery cannot be referenced anywhere else in the parent query. The following example demonstrates a non-correlated subquery.

E.g. Select * From CUST Where '10/03/1990' IN (Select ODATE From ORDER Where CUST.CNUM = ORDER.CNUM)

62. What are the primitive operations common to all record managementsystems?

Addition, deletion and modification.

63. Name the buffer in which all the commands that are typed in

arestored 'Edit'Buffer

64. What are the unary operations in RelationalAlgebra?

PROJECTION and SELECTION.

65. Are the resulting relations of PRODUCT and JOIN operation thesame?

No.

PRODUCT: Concatenation of every row in one relation with every row in another.

JOIN: Concatenation of rows from one relation and related rows from another.

66. What is RDBMSKERNEL?

Two important pieces of RDBMS architecture are the kernel, which is the software, and the data dictionary, which consists of the system-level data structures used by the kernel to manage the database

You might think of an RDBMS as an operating system (or set of subsystems), designed specifically for controlling data access; its primary functions are storing, retrieving, and securing data. An RDBMS maintains its own list of authorized users and their associated privileges; manages memory caches and paging; controls locking for concurrent resource usage; dispatches and schedules user requests; and manages space usage within its table-space structures.

67. Name the sub-systems of aRDBMS

I/O, Security, Language Processing, Process Control, Storage Management, Logging and Recovery, Distribution Control, Transaction Control, Memory Management, Lock Management

68. Which part of the RDBMS takes care of the data dictionary?How

Data dictionary is a set of tables and database objects that is stored in a special area of the database and maintained exclusively by the kernel.

69. What is the job of the information stored indata-dictionary?

The information in the data dictionary validates the existence of the objects, provides access to them, and maps the actual physical storage location.

70. Not only RDBMS takes care of locating data italso

Determines an optimal access path to store or retrieve the data

71. How do you communicate with anRDBMS?

You communicate with an RDBMS using Structured Query Language (SQL)

72. Define SQL and state the differences between SQL and other conventional programmingLanguages

SQL is a nonprocedural language that is designed specifically for data access operations on normalized relational database structures. The primary difference between SQL and other conventional programming languages is that SQL statements specify what data operations should be performed rather than how to perform them.

73. Name the three major set of files on disk that compose a database inOracle

There are three major sets of files on disk that compose a database. All the files are binary. These are

- Database files
- Control files
- Redologs

The most important of these are the database files where the actual data resides. The control files and the redo logs support the functioning of the architecture itself.

All three sets of files must be present, open, and available to Oracle for any data on the database to be useable. Without these files, you cannot access the database, and the database administrator might have to recover some or all of the database using a backup, if there is one.

74. What is an OracleInstance?

The Oracle system processes, also known as Oracle background processes, provide functions for the user processes—functions that would otherwise be done by the user processes themselves

Oracle database-wide system memory is known as the SGA, the system global area or shared global area. The data and control structures in the SGA are shareable, and all the Oracle background processes and user processes can use them.

The combination of the SGA and the Oracle background processes is known as an Oracle instance

75. What are the four Oracle system processes that must always be up and running for the database to beuseable

The four Oracle system processes that must always be up and running for the database to be useable include DBWR (Database Writer), LGWR (Log Writer), SMON (System Monitor), and PMON (Process Monitor).

76. What are database files, control files and log files. How many of these files should a database have at least? Why?

Database Files

The database files hold the actual data and are typically the largest in size. Depending on their sizes, the tables (and other objects) for all the user accounts can go in one database file—but that's not an ideal situation because it does not make the database structure very flexible for controlling access to storage for different users, putting the database on different disk drives, or backing up and restoring just part of the database.

You must have at least one database file but usually, more than one files are used. In terms of accessing and using the data in the tables and other objects, the number (or location) of the files is immaterial.

The database files are fixed in size and never grow bigger than the size at which they werecreated

ControlFiles

The control files and redo logs support the rest of the architecture. Any database must have at least one control file, although you typically have more than one to guard against loss. The control file records the name of the database, the date and time it was created, the location of the database and redoes logs, and the synchronization information to ensure that all three sets of files are always in step. Every time you add a new database or redo log file to the database, the information is recorded in the control files.

Redo Logs

Any database must have at least two redo logs. These are the journals for the database; the redo logs record all changes to the user objects or system objects. If any type of failure occurs, the changes recorded in the redo logs can be used to bring the database to a consistent state without losing any committed transactions. In the case of non-data loss failure, Oracle can apply the information in the redo logs automatically without intervention from theDBA.

The redo log files are fixed in size and never grow dynamically from the size at which they were created.

77. What isROWID?

The ROWID is a unique database-wide physical address for every row on every table. Once assigned (when the row is first inserted into the database), it never changes until the row is deleted or the table is dropped.

The ROWID consists of the following three components, the combination of which uniquely identifies the physical storage location of the row.

- Oracle database file number, which contains the block with therows
- Oracle block address, which contains therow
- The row within the block (because each block can hold manyrows)

The ROWID is used internally in indexes as a quick means of retrieving rows with a particular key value. Application developers also use it in SQL statements as a quick way to access a row once they know the ROWID

78. What is Oracle Block? Can two Oracle Blocks have the sameaddress?

Oracle "formats" the database files into a number of Oracle blocks when they are first created—making it easier for the RDBMS software to manage the files and easier to read data into the memory areas.

The block size should be a multiple of the operating system block size. Regardless of the block size, the entire block is not available for holding data; Oracle takes up some space to manage the contents of the block. This block header has a minimum size, but it can grow.

These Oracle blocks are the smallest unit of storage. Increasing the Oracle block size can improve performance, but it should be done only when the database is first created.

Each Oracle block is numbered sequentially for each database file starting at 1. Two blocks can have the same block address if they are in different database files.

79. What is databaseTrigger?

A database trigger is a PL/SQL block that can defined to automatically execute for insert, update, and delete statements against a table. The trigger can e defined to execute once for the entire statement or once for every row that is inserted, updated, or deleted. For any one table, there are twelve events for which you can define database triggers. A database trigger can call database procedures that are also written in PL/SQL.

80. Name two utilities that Oracle provides, which are use for backup andrecovery.

Along with the RDBMS software, Oracle provides two utilities that you can use to back up and restore the database. These utilities are Exportand Import.

The Export utility dumps the definitions and data for the specified part of the database to an operating system binary file. The Import utility reads the file produced by an export, recreates the definitions of objects, and inserts the data

If Export and Import are used as a means of backing up and recovering the database, all the changes made to the database cannot be recovered since the export was performed. The best you can do is recovering the database to the time when the export was last performed.

81. Name two utilities that Oracle provides, which are use for backup andrecovery.

Along with the RDBMS software, Oracle provides two utilities that you can use to back up and restore the database. These utilities are Exportand Import.

The Export utility dumps the definitions and data for the specified part of the database to an operating system binary file. The Import utility reads the file produced by an export, recreates the definitions of objects, and inserts the data

If Export and Import are used as a means of backing up and recovering the database, all the changes made to the database cannot be recovered since the export was performed. The best you can do is recovering the database to the time when the export was last performed.

82. What are stored-procedures? And what are the advantages of usingthem.

Stored procedures are database objects that perform a user defined operation. A stored procedure can have a set of compound SQL statements. A stored procedure executes the SQL commands and returns the result to the client. Stored procedures are used to reduce network traffic.

83. Tables derived from theERD

a) Are totallyunnormalised

b) Are always in 1NF

c) Can be furtherdenormalised

d) May have multi-valuedattributes

e) Are always in1NF

84. Spurious tuples may occur dueto

i. Badnormalization

ii. Thetajoins

iii. Updating tables fromjoin

a) i&ii b) ii &iii

c)i&iii d) ii &iii

(a) i& iii because theta joins are joins made on keys that are not primary keys.

85. In mapping of ERD toDFD

a) entities in ERD should correspond to an existing entity/store inDFD

b) entity in DFD is converted to attributes of an entity inERD

c) relations in ERD has 1 to 1 correspondence to processes inDFD

d) relationships in ERD has 1 to 1 correspondence to flows inDFD

(a) entities in ERD should correspond to an existing entity/store inDFD

dominant entity is theentity

a) on the N side in a 1 : Nrelationship

b) on the 1 side in a 1 : Nrelationship

c) on either side in a 1 : 1relationship

d) nothing to do with 1 : 1 or 1 : Nrelationship

(b) on the 1 side in a 1 : Nrelationship

87. Select 'NORTH', CUSTOMER From CUST_DTLS Where REGION = 'N' Order By CUSTOMER Union Select 'EAST', CUSTOMER From CUST_DTLS WhereREGION = 'E' Order By CUSTOMER

The above is

a) Not anerror

b) Error - the string in single quotes 'NORTH' and'SOUTH'

c) Error - the string should be in doublequotes

d) Error - ORDER BYclause

(d) Error - the ORDER BY clause. Since ORDER BY clause cannot be used in UNIONS

88. What is StorageManager?

It is a program module that provides the interface between the low-level data stored in database, application programs and queries submitted to the system.

89. What is BufferManager?

It is a program module, which is responsible for fetching data from disk storage into main memory and deciding what data to be cache in memory.

90. What is TransactionManager?

It is a program module, which ensures that database, remains in a consistent state despite system failures and concurrent transaction execution proceeds without conflicting.

91. What is FileManager?

It is a program module, which manages the allocation of space on disk storage and data structure used to represent information stored on a disk.