

**LAB EXPERIMENTS**

*PART A: SQL PROGRAMMING*



### Consider the following schema for a Library Database:

**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.**

### Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017

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

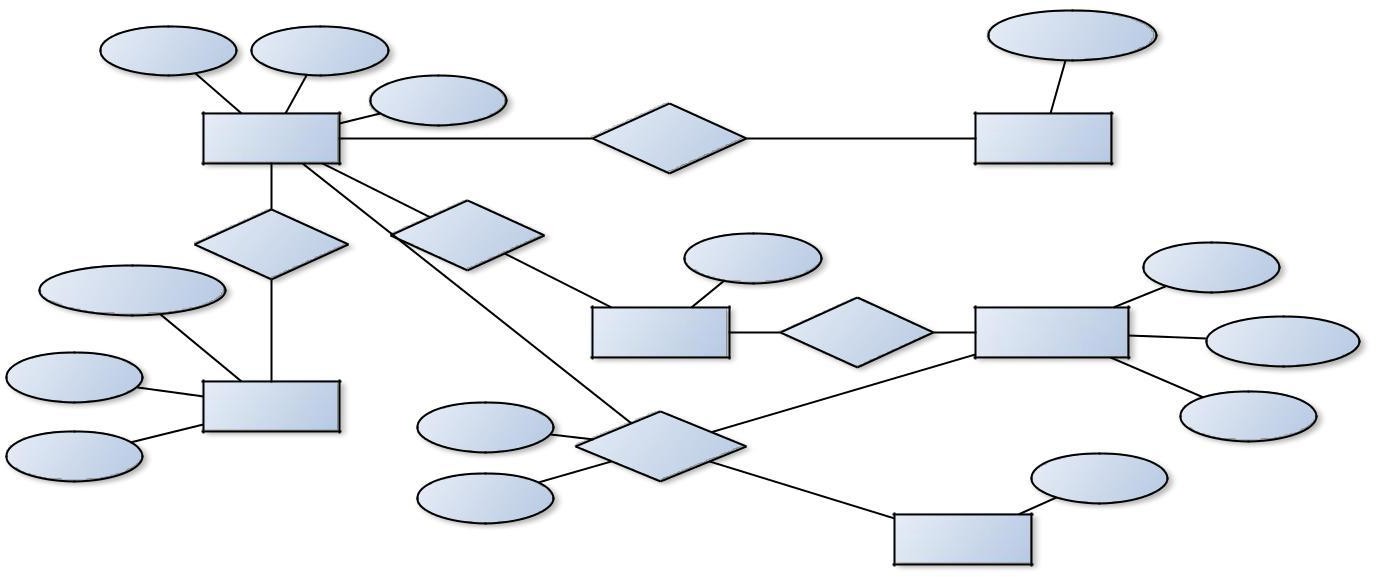
### Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.

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

**Solution:**

**Entity-Relationship Diagram**





**Author\_Name**

**Book\_id**

Title

Pub\_Year M

N

Book written-by Book\_Authors

N

Published-by

Has

N

No\_of\_copies

**Branch\_id**

**Publisher\_Name**

M

N

1

M

Book\_Copies

In

Library\_Branch

Branch\_Name

Address

Publisher

Date\_out

N

Address

Phone

Book\_Lending

**Card\_No**

Due\_date

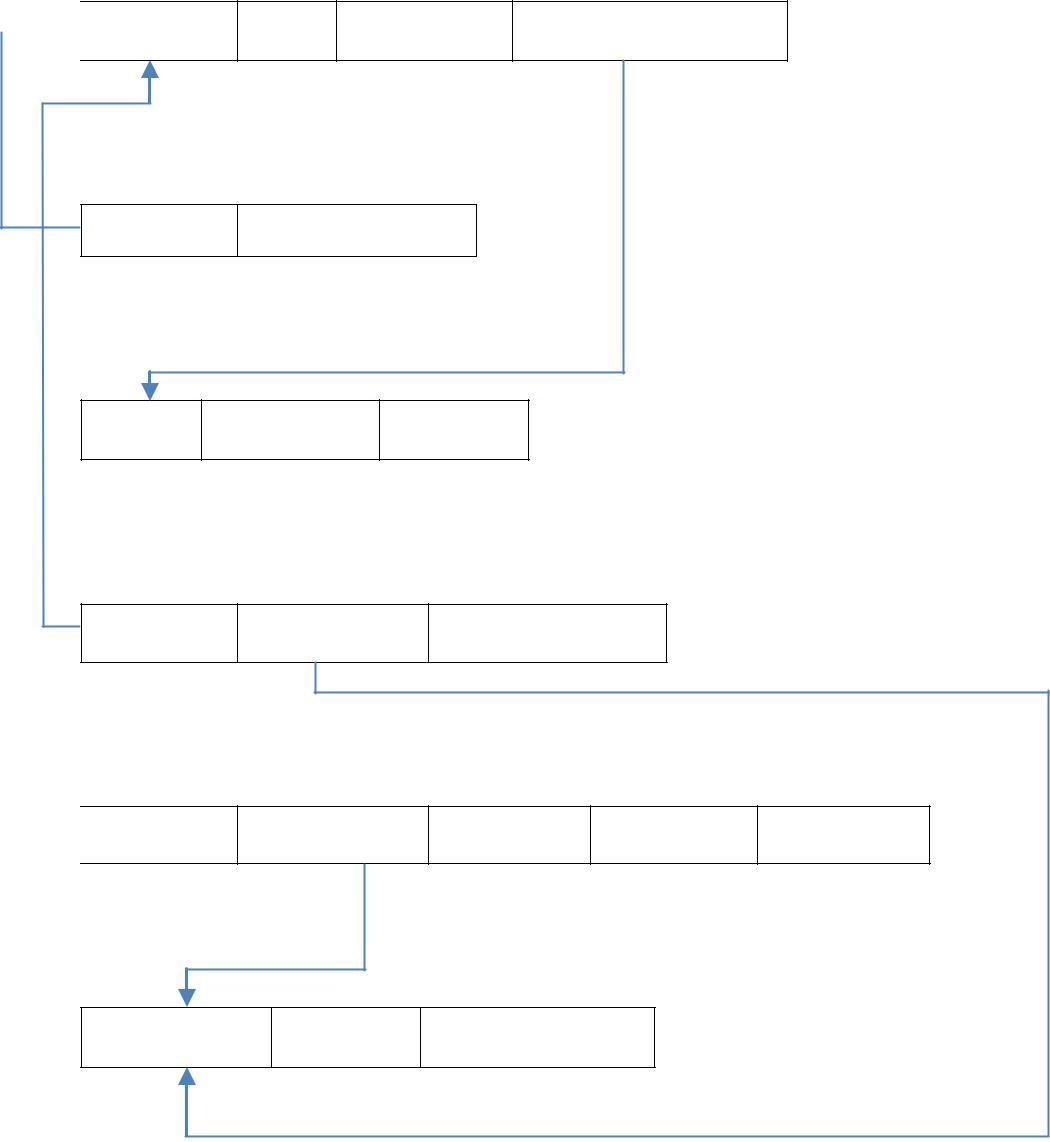
N

Card



**Schema Diagram**

***Book***



***Book\_id*** *Title Pub\_Year Publisher\_Name*

***Book\_Authors***

***Book\_id Author\_name Publisher***

***Name*** *Phone\_no Address*

***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*** *Address Branch\_name*

**Table Creation**

CREATE TABLE PUBLISHER

(NAME VARCHAR2 (20) PRIMARY KEY, PHONE INTEGER,

ADDRESS VARCHAR2 (20));

CREATE TABLE BOOK

(BOOK\_ID INTEGER PRIMARY KEY, 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);

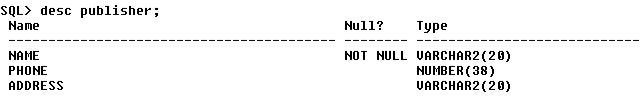
CREATE TABLE BOOK\_LENDING (DATE\_OUT DATE,

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));

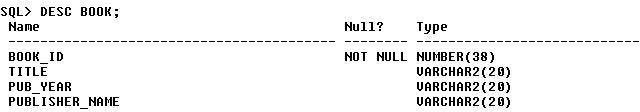
**Table Descriptions**

DESC PUBLISHER;

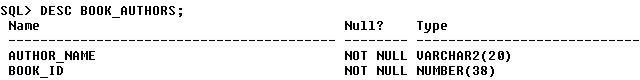
DESC BOOK;



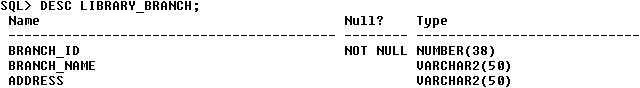




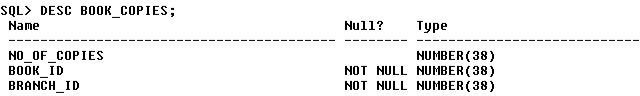
DESC BOOK\_AUTHORS;



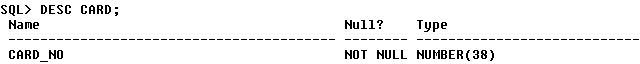
DESC LIBRARY\_BRANCH;



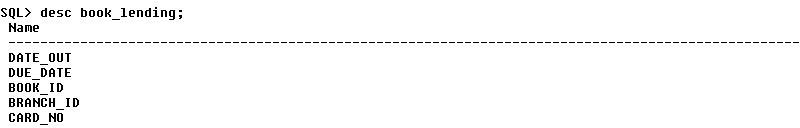
DESC BOOK\_COPIES;



DESC CARD;



DESC BOOK\_LENDING;







**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’); INSERT INTO PUBLISHER VALUES (‘GRUPO PLANETA’, 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,’DEPT OF CS&E,ATRIA I T’,’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);



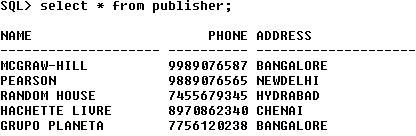


INSERT INTO BOOK\_LENDING VALUES (’01-JAN-17’,’01-JUN-17’, 1, 10, 101);

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);

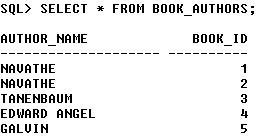
INSERT INTO BOOK\_LENDING VALUES (’15-MAR-17’,’15-JUL-17’, 4, 11, 101);

INSERT INTO BOOK\_LENDING VALUES (‘12-APR-17’,’12-MAY-17’, 1, 11, 104); SELECT \* FROM PUBLISHER;

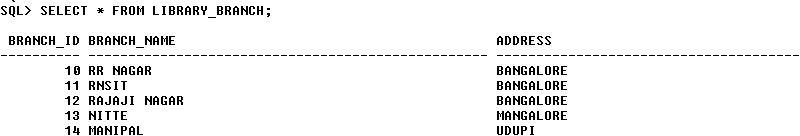
SELECT \* FROM BOOK;



SELECT \* FROM BOOK\_AUTHORS;



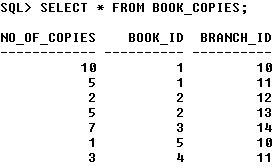
SELECT \* FROM LIBRARY\_BRANCH;



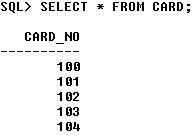
SELECT \* FROM BOOK\_COPIES;



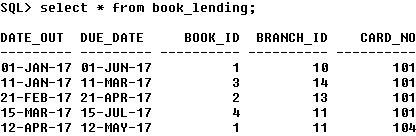




SELECT \* FROM CARD;



SELECT \* FROM BOOK\_LENDING;



**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\_BRANCH L WHERE B.BOOK\_ID=A.BOOK\_ID

AND B.BOOK\_ID=C.BOOK\_ID

AND L.BRANCH\_ID=C.BRANCH\_ID;







### 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

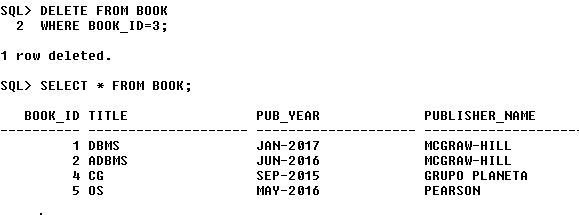
WHERE DATE\_OUT BETWEEN ’01-JAN-2017’ AND ’01-JUL-2017’ GROUP BY CARD\_NO

HAVING COUNT (\*)>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;



### 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;



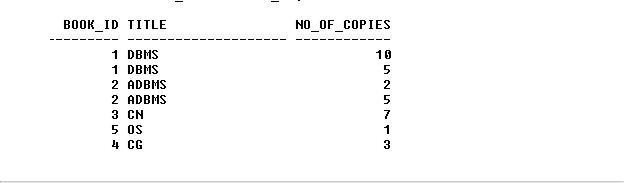


### Create a view of all books and its number of copies that are currently available in the Library.

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;





### Consider the following schema for Order Database:

**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**

### Count the customers with grades above Bangalore’s average.

* 1. **Find the name and numbers of all salesmen who had more than one customer.**

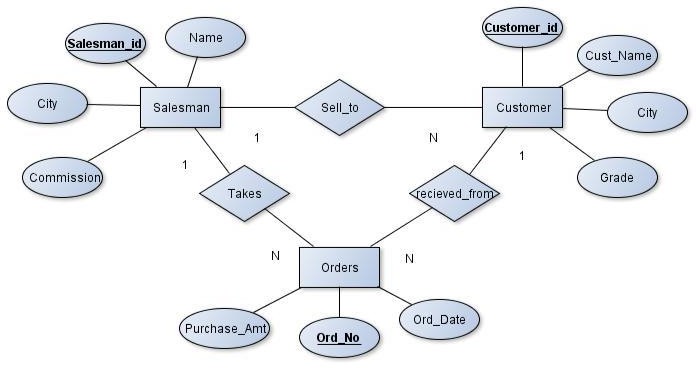
### List all salesmen and indicate those who have and don’t have customers in their cities (Use UNION operation.)

* 1. **Create a view that finds the salesman who has the customer with the highest order of a day.**

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

**Solution:**

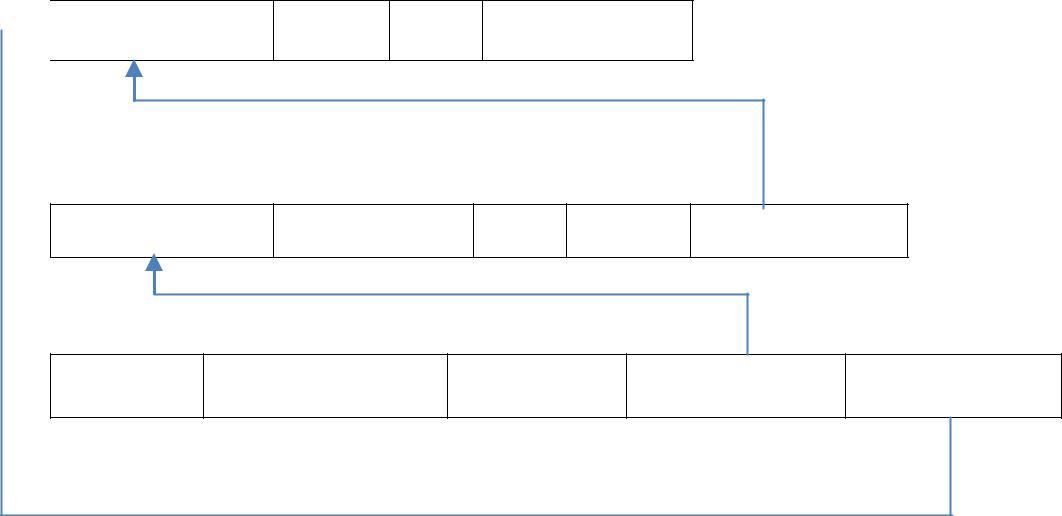
**Entity-Relationship Diagram**





**Schema Diagram**

***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*

**Table Creation**

CREATE TABLE SALESMAN (SALESMAN\_ID NUMBER (4),

NAME VARCHAR2 (20),

CITY VARCHAR2 (20),

COMMISSION VARCHAR2 (20), PRIMARY KEY (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,

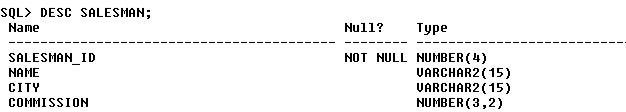
PRIMARY KEY (ORD\_NO),

CUSTOMER\_ID REFERENCES CUSTOMER1 (CUSTOMER\_ID) ON DELETE CASCADE, SALESMAN\_ID REFERENCES SALESMAN (SALESMAN\_ID) ON DELETE CASCADE);

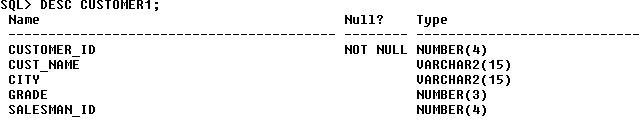


**Table Descriptions**

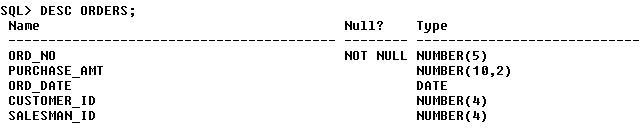
DESC SALESMAN;



DESC CUSTOMER1;



DESC ORDERS;



**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 %’); INSERT INTO SALESMAN VALUES (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);



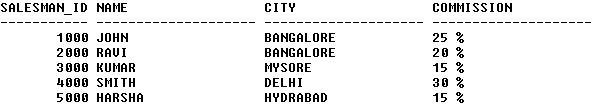


INSERT INTO ORDERS VALUES (52, 1000, ‘24-FEB-17’, 13, 2000);

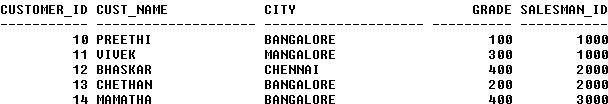
INSERT INTO ORDERS VALUES (53, 3500, ‘13-APR-17’, 14, 3000);

INSERT INTO ORDERS VALUES (54, 550, ‘09-MAR-17’, 12, 2000);

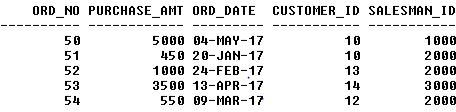
SELECT \* FROM SALESMAN;



SELECT \* FROM CUSTOMER1;



SELECT \* FROM ORDERS;



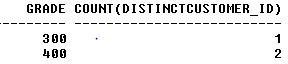
**Queries:**

* + 1. **Count the customers with grades above Bangalore’s average.**

SELECT GRADE, COUNT (DISTINCT CUSTOMER\_ID) FROM CUSTOMER1 GROUP BY GRADE

HAVING GRADE > (SELECT AVG(GRADE) FROM CUSTOMER1

WHERE CITY='BANGALORE');

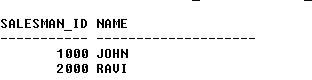




### Find the name and numbers of all salesmen who had more than one customer.

SELECT SALESMAN\_ID, NAME FROM SALESMAN A

WHERE 1 < (SELECT COUNT (\*) FROM CUSTOMER1

WHERE SALESMAN\_ID=A.SALESMAN\_ID);

### List all salesmen and indicate those who have and don’t have customers in their cities (Use UNION operation.)

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;



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

CREATE VIEW ELITSALESMAN AS SELECT B.ORD\_DATE, A.SALESMAN\_ID, A.NAME FROM SALESMAN A, ORDERS B



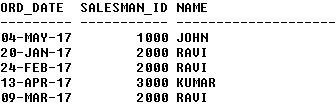


WHERE A.SALESMAN\_ID = B.SALESMAN\_ID

AND B.PURCHASE\_AMT=(SELECT MAX (PURCHASE\_AMT)

FROM ORDERS C

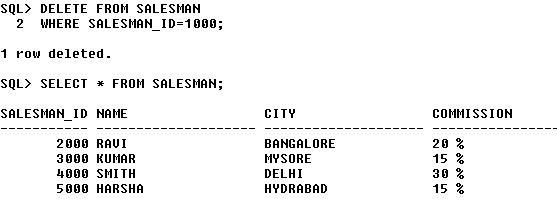
WHERE C.ORD\_DATE = B.ORD\_DATE);



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

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;









### Consider the schema for Movie Database:

**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**

### List the titles of all movies directed by ‘Hitchcock’.

* 1. **Find the movie names where one or more actors acted in two or more movies.**

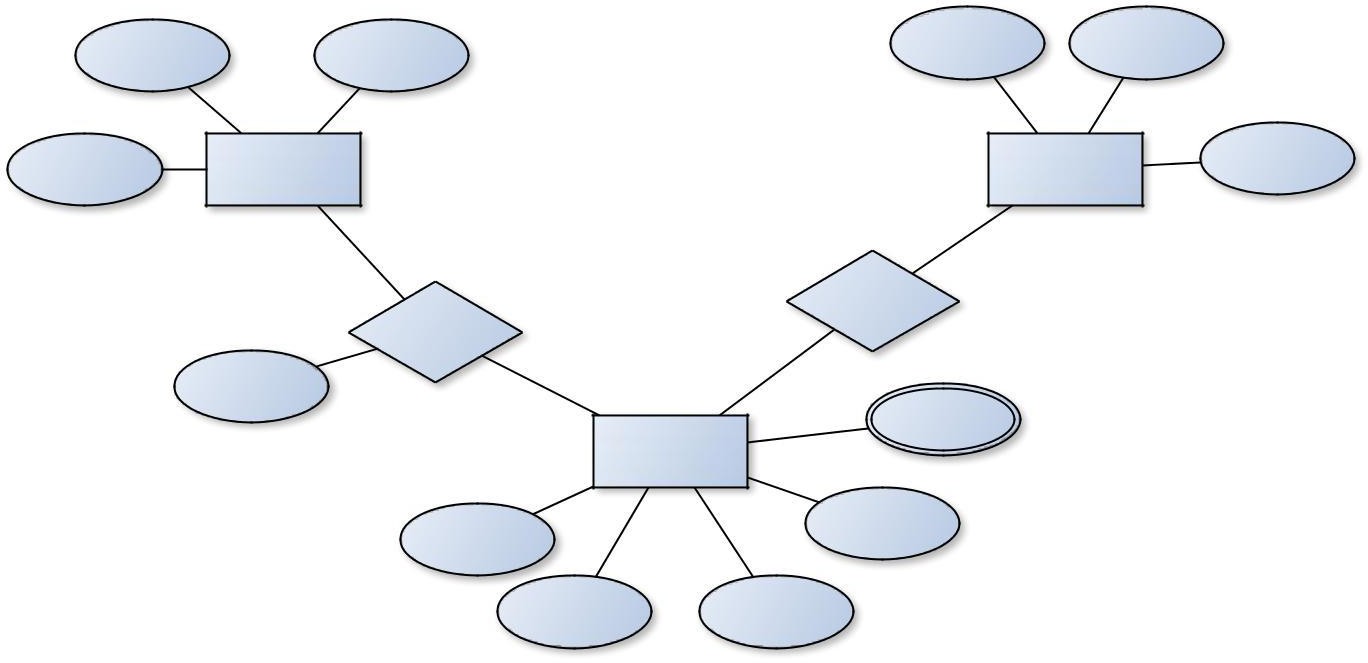
### List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

* 1. **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.**

### Update rating of all movies directed by ‘Steven Spielberg’ to 5.

**Solution:**

**Entity-Relationship Diagram**



**Dir\_id**

**Act\_id**

Dir\_Name

Act\_Name

Act\_Gender

Actor

Director

Dir\_Phone

1

M

Has

Movie\_Cast

N

Role

N

Rev\_Stars

Movies

Mov\_Lang

**Mov\_id**

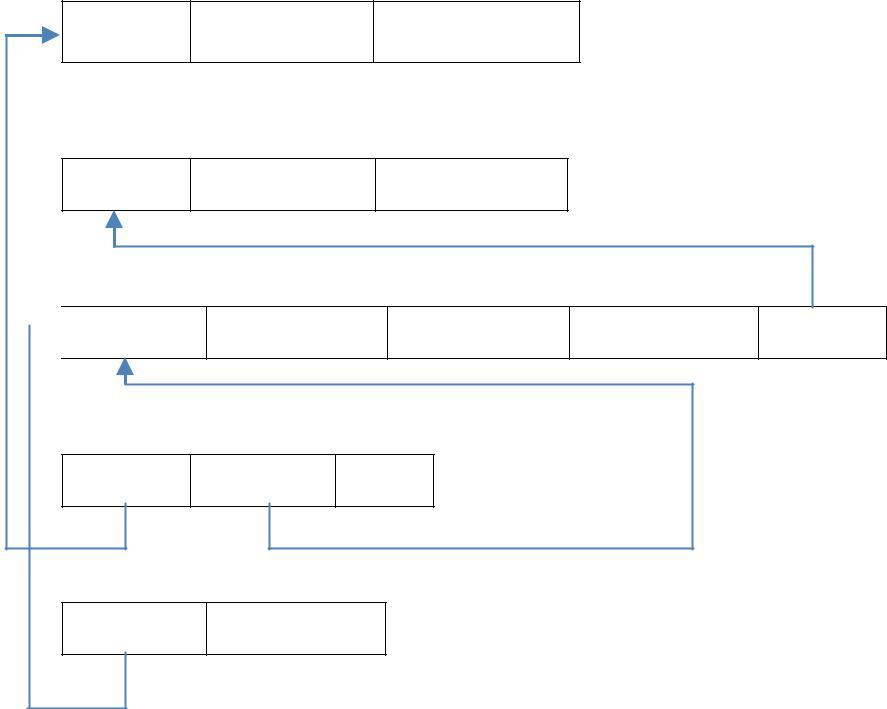
Mov\_Title

Mov\_Year



**Schema Diagram**

***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*

**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));

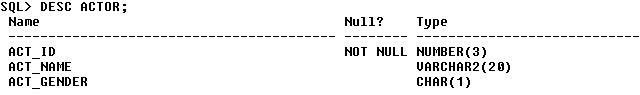
CREATE TABLE RATING ( MOV\_ID NUMBER (4),

REV\_STARS VARCHAR (25), PRIMARY KEY (MOV\_ID),

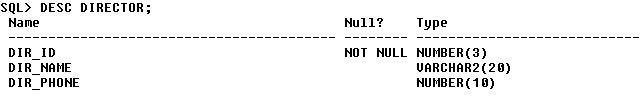
FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID));

**Table Descriptions**

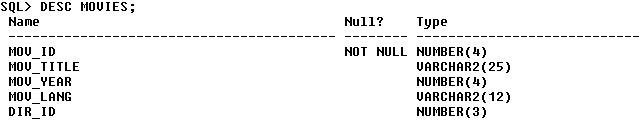
DESC ACTOR;



DESC DIRECTOR;

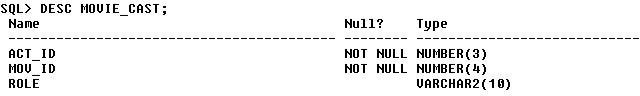


DESC MOVIES;

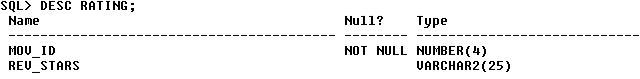




DESC MOVIE\_CAST;



DESC RATING;



**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);

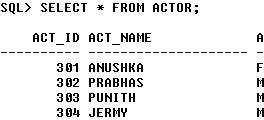




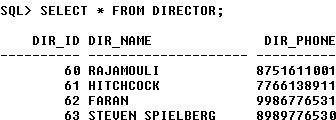
INSERT INTO RATING VALUES (1003, 5);

INSERT INTO RATING VALUES (1004, 4);

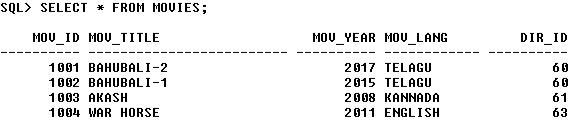
SELECT \* FROM ACTOR;



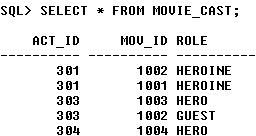
SELECT \* FROM DIRECTOR;



SELECT \* FROM MOVIES;



SELECT \* FROM MOVIE\_CAST;







SELECT \* FROM RATING;



**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’);



1. **Find the movie names where one or more actors acted in two or more movies.**

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;



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

SELECT ACT\_NAME, MOV\_TITLE, MOV\_YEAR





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; 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;

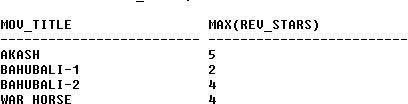


### Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

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;









### Update rating of all movies directed by ‘Steven Spielberg’ to 5

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’));



### Consider the schema for College Database:

**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**

### List all the student details studying in fourth semester ‘C’ section.

* 1. **Compute the total number of male and female students in each semester and in each section.**

### Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.

* 1. **Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.**

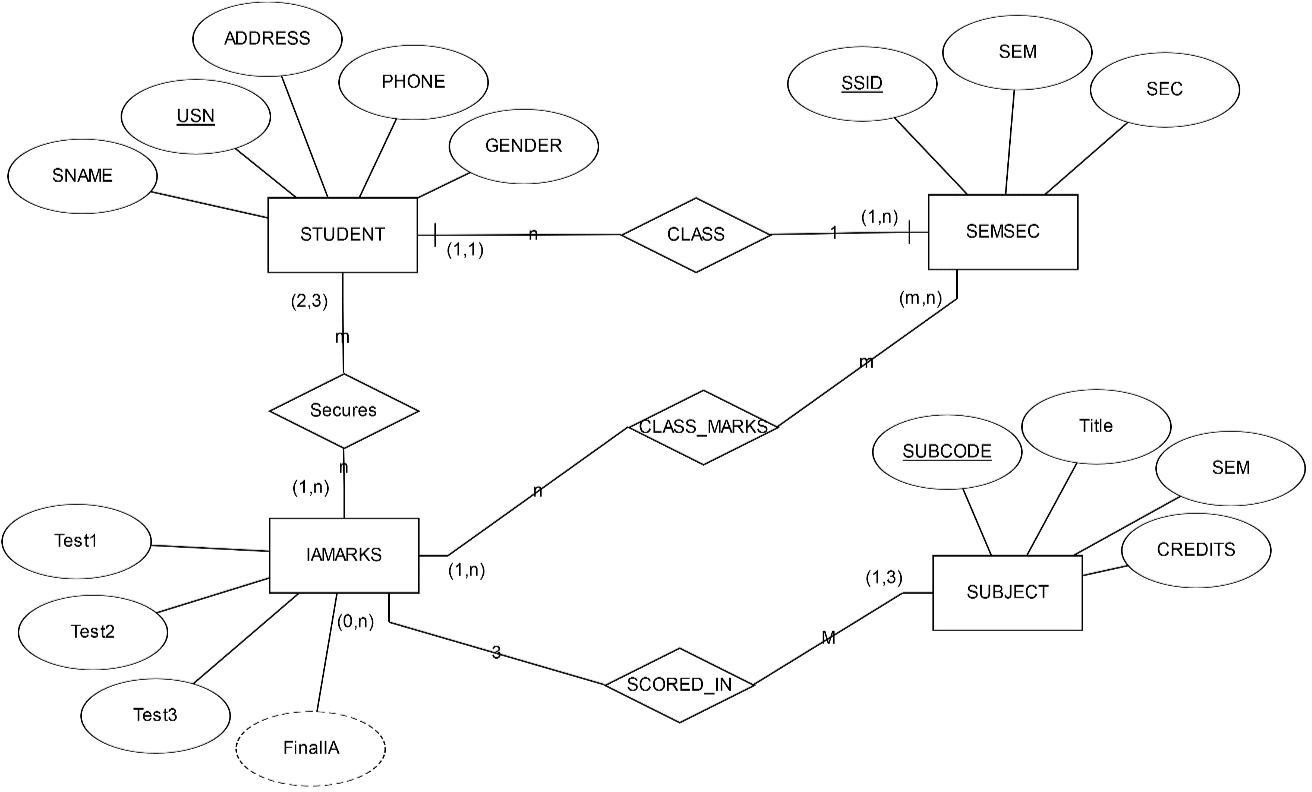
### Categorize students based on the following criterion: If FinalIA = 17 to 20 then CAT = ‘Outstanding’

**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.

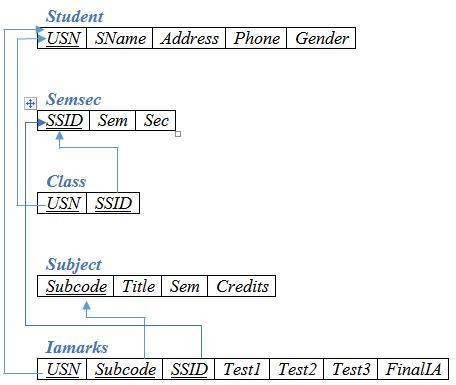
**Solution:**

**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),

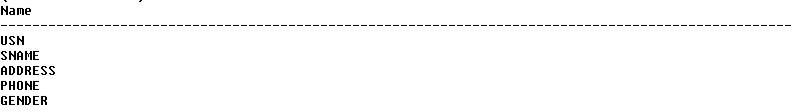
FINALIA NUMBER (2),

PRIMARY KEY (USN, SUBCODE, SSID),

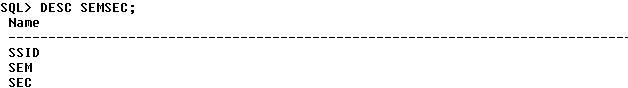
FOREIGN KEY (USN) REFERENCES STUDENT (USN), FOREIGN KEY (SUBCODE) REFERENCES SUBJECT (SUBCODE), FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID));

**Table Descriptions**

DESC STUDENT;



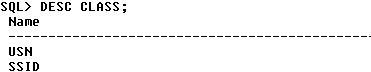
DESC SEMSEC;







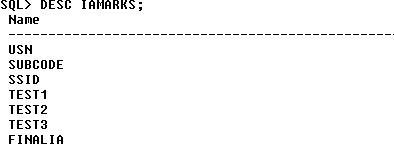
DESC CLASS;



DESC SUBJECT;



DESC IAMARKS;



**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'); 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’);

INSERT INTO CLASS VALUES (‘1RN13CS020’,’CSE8A’); INSERT INTO CLASS VALUES (‘1RN13CS062’,’CSE8A’); INSERT INTO CLASS VALUES (‘1RN13CS066’,’CSE8B’); INSERT INTO CLASS VALUES (‘1RN13CS091’,’CSE8C’);

INSERT INTO CLASS VALUES (‘1RN14CS010’,’CSE7A’); INSERT INTO CLASS VALUES (‘1RN14CS025’,’CSE7A’); INSERT INTO CLASS VALUES (‘1RN14CS032’,’CSE7A’);

INSERT INTO CLASS VALUES (‘1RN15CS011’,’CSE4A’); INSERT INTO CLASS VALUES (‘1RN15CS029’,’CSE4A’); INSERT INTO CLASS VALUES (‘1RN15CS045’,’CSE4B’); INSERT INTO CLASS VALUES (‘1RN15CS091’,’CSE4C’);

INSERT INTO CLASS VALUES (‘1RN16CS045’,’CSE3A’); INSERT INTO CLASS VALUES (‘1RN16CS088’,’CSE3B’); INSERT INTO CLASS VALUES (‘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);

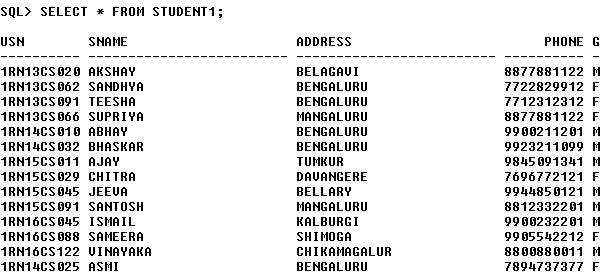
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1RN13CS091','10CS81','CSE8C', 15, 16, 18);

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);

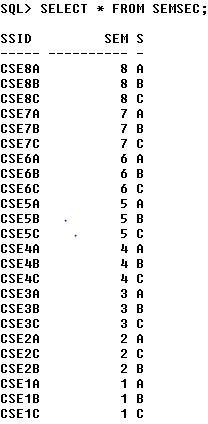
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1RN13CS091','10CS84','CSE8C', 20, 16, 19);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1RN13CS091','10CS85','CSE8C', 15, 15, 12);

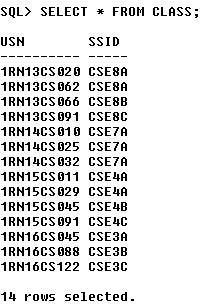
SELECT \* FROM STUDENT;



SELECT \* FROM SEMSEC;



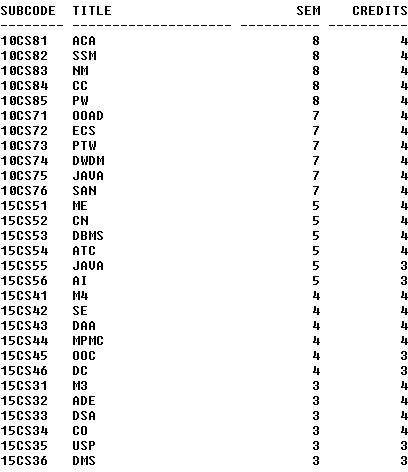
SELECT \* FROM CLASS;



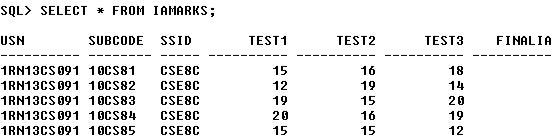




SELECT \* FROM SUBJECT;



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’;

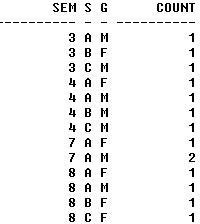


### Compute the total number of male and female students in each semester and in each section.

SELECT SS.SEM, SS.SEC, S.GENDER, COUNT (S.GENDER) AS COUNT FROM STUDENT S, SEMSEC SS, CLASS C WHERES.USN = C.USN AND

SS.SSID = C.SSID

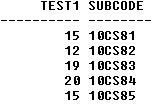
GROUP BY SS.SEM, SS.SEC, S.GENDER ORDER BY SEM;



### Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.

CREATE VIEW STU\_TEST1\_MARKS\_VIEW AS SELECT TEST1, SUBCODE FROM IAMARKS

WHERE USN = '1RN13CS091';







### Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.

CREATE OR REPLACE PROCEDURE AVGMARKS IS

CURSOR C\_IAMARKS IS

SELECT GREATEST(TEST1,TEST2) AS A, GREATEST(TEST1,TEST3) AS B,

GREATEST(TEST3,TEST2) AS C FROM IAMARKS

WHERE FINALIA IS NULL FOR UPDATE;

C\_A NUMBER;

C\_B NUMBER;

C\_C NUMBER;

C\_SM NUMBER;

C\_AV NUMBER;

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;

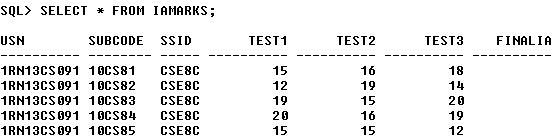
END;

/

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

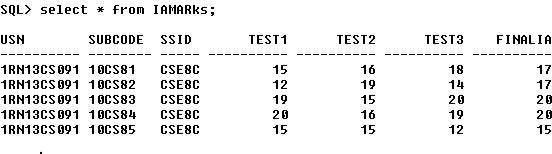


SELECT \* FROM IAMARKS;



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

BEGIN AVGMARKS; END;



### 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’

**th**

### Give these details only for 8 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;











### Consider the schema for Company Database:

**EMPLOYEE (*SSN, Name, Address, Sex, Salary, SuperSSN, DNo*) DEPARTMENT (*DNo, DName, MgrSSN, MgrStartDate*) DLOCATION (*DNo,DLoc*)**

**PROJECT (*PNo, PName, PLocation, DNo*) WORKS\_ON (*SSN, PNo, Hours*)**

### Write SQL queries to

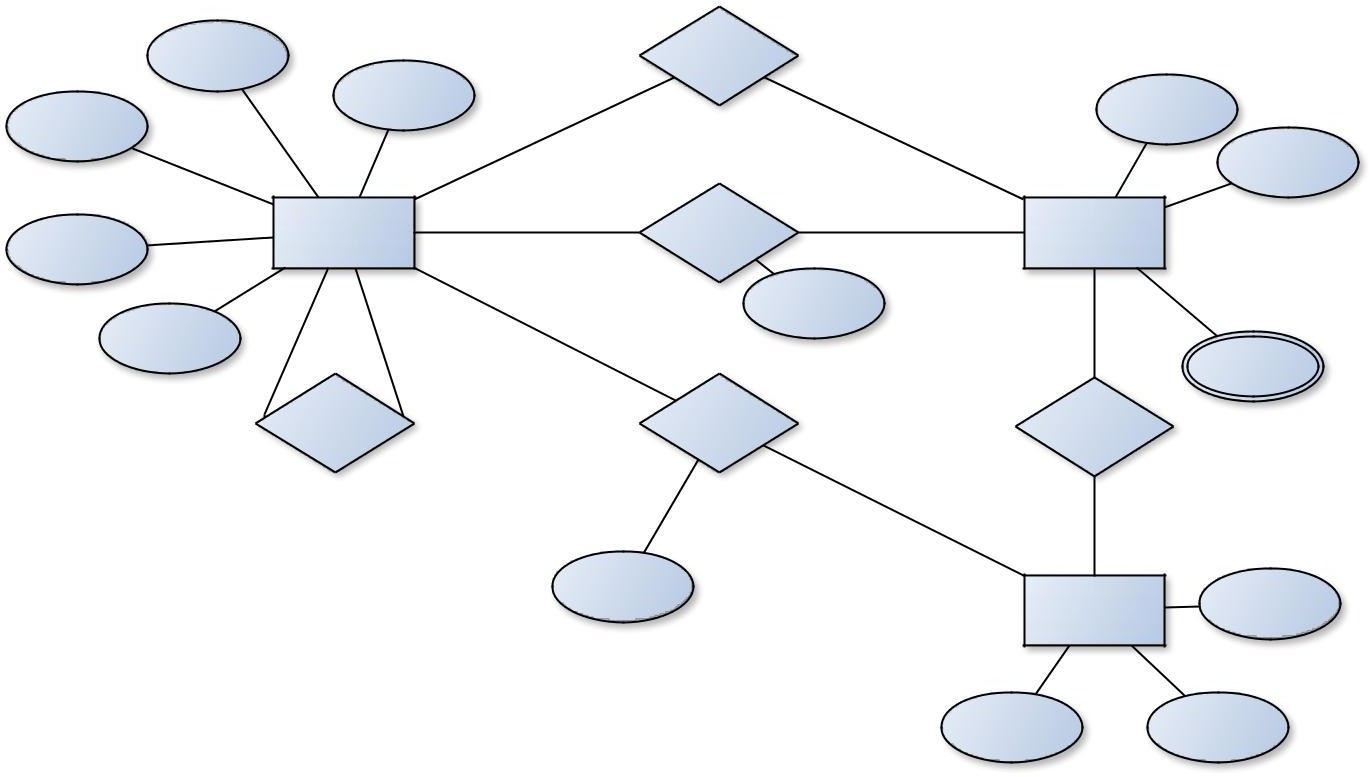
* 1. **Make a list of all project numbers for projects that involve an employee whose last name is ‘Scott’, either as a worker or as a manager of the department that controls the project.**

### Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.

* 1. **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**

### 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**



**SSN**

Controlled\_by

Name

N

1

**DNO**

Salary

DName

1

N

Employee

Manages

Department

Address

MgrStartDate

1

Sex

1

N

M

Dlocation

Supervisor

Supervisee

Supervision

Works\_on

Controls

N

N

Hours

Project

PName

**PNO**

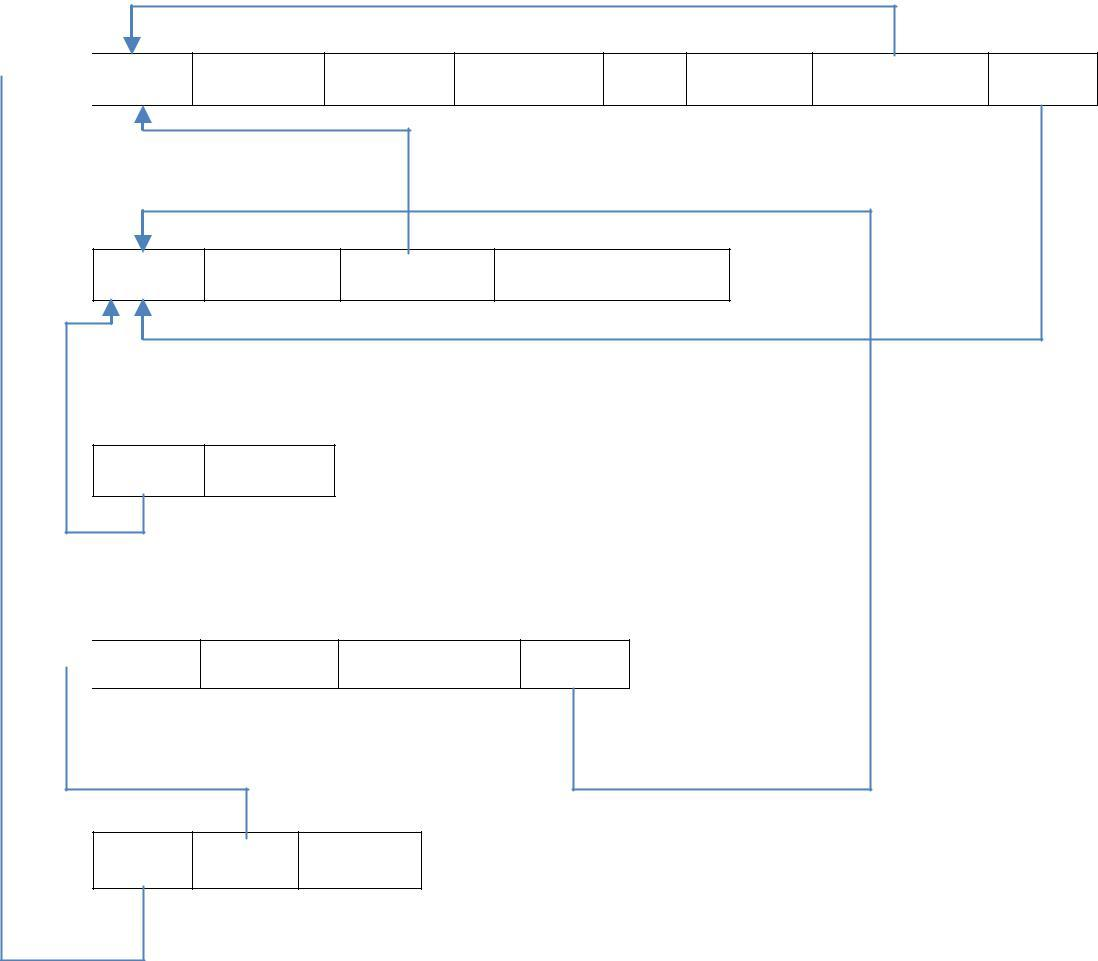
PLocation





**Schema Diagram**

***Employee***



***SSN*** *Fname Lname Address Sex Salary SuperSSN DNO*

***Department***

***DNO*** *Dname MgrSSN MgrStartDate*

***DLocation DNO DLOC***

***Project***

***PNO*** *PName PLocation DNO*

***Works\_on***

***SSN PNO*** *Hours*

**Table Creation**

CREATE TABLE DEPARTMENT

(DNO VARCHAR2 (20) PRIMARY KEY, DNAME VARCHAR2 (20), MGRSTARTDATE DATE);

CREATE TABLE EMPLOYEE

(SSN VARCHAR2 (20) PRIMARY KEY, 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 PRIMARY KEY, PNAME VARCHAR2 (20),

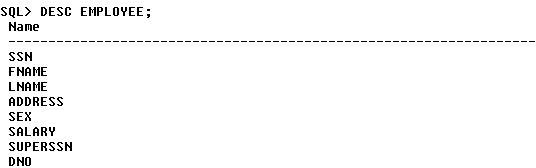
PLOCATION VARCHAR2 (20),

DNO REFERENCES DEPARTMENT (DNO));

CREATE TABLE WORKS\_ON (HOURS NUMBER (2),

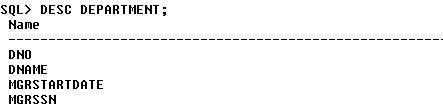
SSN REFERENCES EMPLOYEE (SSN), PNO REFERENCES PROJECT(PNO), PRIMARY KEY (SSN, PNO));

**Table Descriptions**

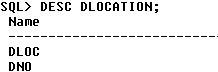
DESC EMPLOYEE;



DESC DEPARTMENT;



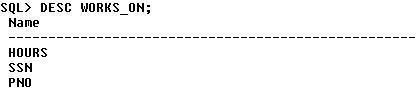
DESC DLOCATION;



DESC PROJECT;



DESC WORKS\_ON;



**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 (‘DEPT OF CS&E,ATRIA I T01’,’NAGESH’,’HR’,’BANGALORE’,’M’, 500000);

INSERT INTO DEPARTMENT VALUES (‘1’,’ACCOUNTS’,’01-JAN-01’,’RNSACC02’); INSERT INTO DEPARTMENT VALUES (‘2’,’IT’,’01-AUG-16’,’DEPT OF CS&E,ATRIA I T01’);

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’ WHERE SSN=’RNSECE01’;

UPDATE EMPLOYEE SET SUPERSSN=’RNSCSE02’, DNO=’5’ WHERE SSN=’RNSCSE01’;

UPDATE EMPLOYEE SET SUPERSSN=’RNSCSE03’, DNO=’5’ WHERE SSN=’RNSCSE02’;

UPDATE EMPLOYEE SET SUPERSSN=’RNSCSE04’, DNO=’5’ WHERE SSN=’RNSCSE03’;

UPDATE EMPLOYEE SET DNO=’5’, SUPERSSN=’RNSCSE05’ WHERE SSN=’RNSCSE04’;



UPDATE EMPLOYEE SET DNO=’5’, SUPERSSN=’RNSCSE06’ WHERE SSN=’RNSCSE05’;

UPDATE EMPLOYEE SET DNO=’5’, SUPERSSN=NULL WHERE SSN=’RNSCSE06’;

UPDATE EMPLOYEE SET DNO=’1’, SUPERSSN=’RNSACC02’ WHERE SSN=’RNSACC01’;

UPDATE EMPLOYEE SET DNO=’1’, SUPERSSN=NULL WHERE SSN=’RNSACC02’;

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

UPDATE EMPLOYEE SET DNO=’2’, SUPERSSN=NULL

WHERE SSN=’DEPT OF CS&E,ATRIA I T01’;

INSERT INTO DLOCATION VALUES (’BANGALORE’, ‘1’); INSERT INTO DLOCATION VALUES (’BANGALORE’, ‘2’); INSERT INTO DLOCATION VALUES (’BANGALORE’, ‘3’); INSERT INTO DLOCATION VALUES (’MANGALORE’, ‘4’); INSERT INTO DLOCATION VALUES (’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,’SALARY MANAGEMENT’,’BANGALORE’,’1’); INSERT INTO PROJECT VALUES (106,’OPENSTACK’,’BANGALORE’,’4’); INSERT INTO PROJECT VALUES (107,’SMART CITY’,’BANGALORE’,’2’);



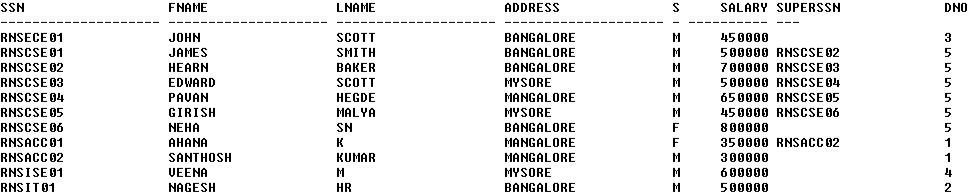




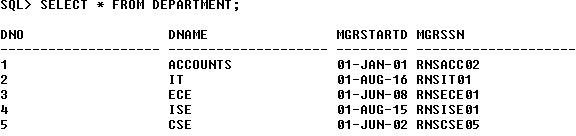
INSERT INTO WORKS\_ON VALUES (4, ‘RNSCSE01’, 100); INSERT INTO WORKS\_ON VALUES (6, ‘RNSCSE01’, 101); INSERT INTO WORKS\_ON VALUES (8, ‘RNSCSE01’, 102); INSERT INTO WORKS\_ON VALUES (10, ‘RNSCSE02’, 100); INSERT INTO WORKS\_ON VALUES (3, ‘RNSCSE04’, 100); INSERT INTO WORKS\_ON VALUES (4, ‘RNSCSE05’, 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, ‘DEPT OF CS&E,ATRIA I T01’, 107);

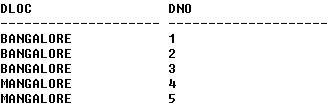
SELECT \* FROM EMPLOYEE;



SELECT \* FROM DEPARTMENT;

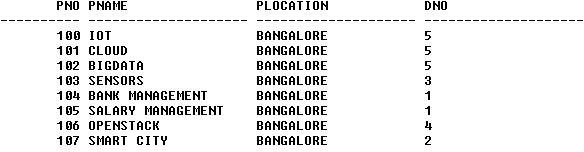


SELECT \* FROM DLOCATION;

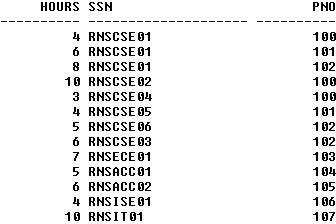




SELECT \* FROM PROJECT;



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.MGRSSN=E.SSN AND E.LNAME=’SCOTT’) UNION

(SELECT DISTINCT P1.PNO

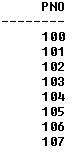
FROM PROJECT P1, WORKS\_ON W, EMPLOYEE E1 WHERE P1.PNO=W.PNO

AND E1.SSN=W.SSN

AND E1.LNAME=’SCOTT’);







### Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.

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’;



### 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’;



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

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));



### 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;









**Viva Questions**

1. **What is SQL?**

Structured Query Language

### What is database?

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 specific purpose.

### What is DBMS?

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.

### What is a Database system?

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

1. **Advantages of DBMS?**

# 

Redundancy is controlled.

* + Unauthorized access is restricted.
  + Providing multiple user interfaces.
  + Enforcing integrity constraints.
  + Providing backup and recovery.

1. **Disadvantage in File Processing System?**

# 

Data redundancy & inconsistency.

* + Difficult in accessing data.
  + Data isolation.
  + Data integrity.
  + Concurrent access is not possible.
  + Security Problems.

1. **Describe the three levels of data abstraction?**

There are three levels of abstraction:

# 

Physical level: The lowest level of abstraction describes how data are stored.

# 

Logical level*:* The next higher level of abstraction, describes what data are stored in database and what relationship among those data.



DEPT OF CS&E,ATRIA I T, Bangalore Page 47



# 

View level:The highest level of abstraction describes only part of entire database.

### Define the "integrity rules"

There are two Integrity rules.

# 

Entity Integrity:States that “Primary key cannot have NULL value”

# 

Referential Integrity:States that “Foreign Key can be either a NULL value or should be Primary Key value of other relation.

### 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.

### What is Data Independence?

Data independence means that “the application is independent of the storage structure and access strategy of data”. In other words, The ability to modify the schema definition in one level should not affect the schema definition in the next higher level.

Two types of Data Independence:

# 

Physical Data Independence: Modification in physical level should not affect the logical level.

# 

Logical Data Independence: Modification in logical level should affect the view level.

NOTE: Logical Data Independence is more difficult to achieve

### What is a view? How it is related to data independence?

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.

### What is Data Model?



DEPT OF CS&E,ATRIA I T, Bangalore Page 48



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

### What is E-R model?

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.

### What is Object Oriented model?

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.

### What is an Entity?

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

### What is an Entity type?

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

### What is an Entity set?

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

### 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 an attribute?** |  |  |
|  | It is a particular property, which describes the entity. |  |  |
| **20.** | **What is a Relation Schema and a Relation?** |  |  |
|  | A relation Schema denoted by R(A1, A2, …, An) is made up of the relation | | name |
| R and the list of attributes Ai that it contains. A relation is defined 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 a Relation?** |  |  |
|  | It is the number of attribute of its relation schema. |  |  |
| **22.** | **What is Relationship?** |  |  |
|  | It is an association among two or more entities. |  |  |
| **23.** | **What is Relationship set?** |  |  |
| DEPT OF CS&E,ATRIA I T, Bangalore | |  | Page 49 |



The collection (or set) of similar relationships.

1. ***What is Relationship type?***

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

### What is degree of Relationship type?

It is the number of entity type participating.

### What is DDL (Data Definition Language)?

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

### What is VDL (View Definition Language)?

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

### What is SDL (Storage Definition Language)?

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

### What is Data Storage - Definition Language?

The storage structures and access methods used by database system are specified by a set of definition in a special type of DDL called data storage- definition language.

### What is DML (Data Manipulation Language)?

This language that enable user to access or manipulate data as organized by appropriate data model.

# 

Procedural DML or Low level: DML requires a user to specify what data are needed and how to get those data.

# 

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

### What is DML Compiler?

|  |  |  |
| --- | --- | --- |
| It translates DML statements in a query language into low-level instruction |  | that the |
| query evaluation engine can understand. |  |  |
| **32. What is Relational Algebra?** |  |  |
| It is a procedural query language. It consists of a set of operations that | take | one or |
| two relations as input and produce a new relation. |  |  |
| **33. What is Relational Calculus?** |  |  |
| DEPT OF CS&E,ATRIA I T, Bangalore |  | Page 50 |



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 is normalization?** |  |
|  | It is a process of analyzing the given relation schemas based on their | Functional |
| Dependencies (FDs) and primary key to achieve the properties | |  |
|  |   Minimizing redundancy |  |
|  |   Minimizing insertion, deletion and update anomalies. |  |
| **35.** | **What is Functional Dependency?** |  |

A Functional dependency is denoted by X Y between sets of attributes X and Y

that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of

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.

1. **When is a functional dependency F said to be minimal?**

# 

Every dependency in F has a single attribute for its right hand side.

# 

We cannot replace any dependency X A in F a dependency Y A where Y is a proper

subset of X and still have a set of dependency that is equivalent to F.

# 

We cannot remove any dependency from F and still have set of dependency that is equivalent to F.

### What is Multivalued dependency?

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 following properties

* 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)) ]

### What is Lossless join property?

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





### What is 1 NF (Normal Form)?

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

### What is Fully Functional dependency?

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 any more.

### What is 2NF?

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.

### What is 3NF?

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

* X is a Super-key of R.
* A is a prime attribute of R.

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

### What is BCNF (Boyce-Codd Normal Form)?

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.

### What is 4NF?

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

* X is subset or equal to (or) XY = R.
* X is a super key.

### What is 5NF?

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 some i.
* The join dependency is implied by the set of FD, over R in which the left side is key of R.





### 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.

### What are partial, alternate,, artificial, compound and natural key?

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:

ArtificialKey:

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

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.

### What is indexing and what are the different kinds of indexing?

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

* + Binary search style indexing
  + B-Tree indexing
  + Inverted list indexing
  + Memory resident table
  + Table indexing





### What is system catalog or catalog relation? How is better known as?

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 data dictionary.

### What is meant by query optimization?

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

### What is join dependency and inclusion dependency?

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.

### What is durability in DBMS?

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 called durability.

### What do you mean by atomicity and aggregation?

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.





### What is a Phantom Deadlock?

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.

### What is a checkpoint and when does it occur?

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.

### What are the different phases of transaction?

Different phases are

* + Analysis phase
  + Redo Phase
  + Undo phase

### What do you mean by flat file database?

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.

### What is "transparent DBMS"?

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

### Brief theory of Network, Hierarchical schemas and their properties

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.

### What is a query?

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.

### What do you mean by Correlated subquery?

Subqueries, or nested queries, are used to bring back a set of rows to be used by the parent query. Depending on how the subquery is written, it can be executed once for the parent



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)

1. **What are the primitive operations common to all record management systems?** Addition, deletion and modification.

### Name the buffer in which all the commands that are typed in are stored

1. **What are the unary operations in Relational Algebra?**

PROJECTION and SELECTION.

### Are the resulting relations of PRODUCT and JOIN operation the same?

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.

### What is RDBMS KERNEL?

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.

### Name the sub-systems of a RDBMS

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



### 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.

### What is the job of the information stored in data-dictionary?

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

1. **Not only RDBMS takes care of locating data it also** D~~etermines~~  an optimal access path to store or retrieve the data

### How do you communicate with an RDBMS?

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

### Define SQL and state the differences between SQL and other conventional programming Languages

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.

### Name the three major set of files on disk that compose a database in Oracle

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

* + Database files
  + Control files
  + Redo logs

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.

### What is an Oracle Instance?

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

### What are the four Oracle system processes that must always be up and running for the database to be useable

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).

### 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 were created

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 the DBA.

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

### What is ROWID?

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 the rows
  + Oracle block address, which contains the row
  + The row within the block (because each block can hold many rows)

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

### What is Oracle Block? Can two Oracle Blocks have the same address?

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.



### What is database Trigger?

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.

### Name two utilities that Oracle provides, which are use for backup and recovery.

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.

### Name two utilities that Oracle provides, which are use for backup and recovery.

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.

### What are stored-procedures? And what are the advantages of using them.

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.





### Tables derived from the ERD

1. Are totally unnormalised
2. Are always in 1NF
3. Can be further denormalised
4. May have multi-valued attributes
5. Are always in 1NF

### Spurious tuples may occur due to

1. Bad normalization
2. Theta joins
3. Updating tables from join
   1. 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.

### In mapping of ERD to DFD

1. entities in ERD should correspond to an existing entity/store in DFD
2. entity in DFD is converted to attributes of an entity in ERD
3. relations in ERD has 1 to 1 correspondence to processes in DFD
4. relationships in ERD has 1 to 1 correspondence to flows in DFD
5. entities in ERD should correspond to an existing entity/store in DFD

### A dominant entity is the entity

1. on the N side in a 1 : N relationship
2. on the 1 side in a 1 : N relationship
3. on either side in a 1 : 1 relationship
4. nothing to do with 1 : 1 or 1 : N relationship
5. on the 1 side in a 1 : N relationship

### Select 'NORTH', CUSTOMER From CUST\_DTLS Where REGION = 'N' Order By CUSTOMER Union Select 'EAST', CUSTOMER From CUST\_DTLS Where REGION

**= 'E' Order By CUSTOMER**

The above is

1. Not an error
2. Error - the string in single quotes 'NORTH' and 'SOUTH'





1. Error - the string should be in double quotes
2. Error - ORDER BY clause

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

### What is Storage Manager?

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.

### What is Buffer Manager?

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.

### What is Transaction Manager?

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

### What is File Manager?

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.

### What is Authorization and Integrity manager?

It is the program module, which tests for the satisfaction of integrity constraint and checks the authority of user to access data.

### What are stand-alone procedures?

Procedures that are not part of a package are known as stand-alone because they independently defined. A good example of a stand-alone procedure is one written in a SQL\*Forms application. These types of procedures are not available for reference from other Oracle tools. Another limitation of stand-alone procedures is that they are compiled at run time, which slows execution.

### What are cursors give different types of cursors.

PL/SQL uses cursors for all database information accesses statements. The language supports the use two types of cursors

* *Implicit*
* *Explicit*

1. **What is cold backup and hot backup (in case of Oracle)?**

# 

Cold Backup:



Itis copying the three sets of files (database files, redo logs, and control file) when the instance is shut down. This is a straight file copy, usually from the disk directly to tape. You must shut down the instance to guarantee a consistent copy.

If a cold backup is performed, the only option available in the event of data file loss is restoring all the files from the latest backup. All work performed on the database since the last backup is lost.

* Hot Backup:

Some sites (such as worldwide airline reservations systems) cannot shut down the database while making a backup copy of the files. The cold backup is not an available option.

So different means of backing up database must be used — the hot backup. Issue a SQL command to indicate to Oracle, on a tablespace-by-tablespace basis, that the files of the tablespace are to backed up. The users can continue to make full use of the files, including making changes to the data. Once the user has indicated that he/she wants to back up the tablespace files, he/she can use the operating system to copy those files to the desired backup destination.

The database must be running in ARCHIVELOG mode for the hot backup option. If a data loss failure does occur, the lost database files can be restored using the hot backup and the online and offline redo logs created since the backup was done. The database is restored to the most consistent state without any loss of committed transactions.

### How can you find the minimal key of relational schema?

Minimal key is one which can identify each tuple of the given relation schema uniquely. For finding the minimal key it is required to find the closure that is the set of all attributes that are dependent on any given set of attributes under the given set of functional dependency.

+

**Algo. I** Determining X , closure for X, given set of FDs F

+

1. Set X = X
2. Set Old X+ = +

X

+ +

3. For each FD Y

Z in F and if Y belongs to X then add Z to X

+ +

4. Repeat steps 2 and 3 until Old X = X

**Algo.II**Determining minimal K for relation schema R, given set of FDs F 1. Set K to R that is make K a set of all attributes in R







1. For each attribute A in K

+

* 1. Compute (K – A) with respect to F

+ +

* 1. If (K – A) = R then set K = (K – A)

### What do you understand by dependency preservation?

Given a relation R and a set of FDs F, dependency preservation states that the closure of the union of the projection of F on each decomposed relation Ri is equal to the closure of F. i.e.,

((R1(F)) U … U (Rn(F)))+= F+

if decomposition is not dependency preserving, then some dependency is lost in the decomposition.

### What is meant by Proactive, Retroactive and Simultaneous Update.

*Proactive Update:*

The updates that are applied to database before it becomes effective in real world.

*Retroactive Update:*

The updates that are applied to database after it becomes effective in real world.

*Simultaneous Update:*

The updates that are applied to database at the same time when it becomes effective in real world

### .What are the different types of JOIN operations?

Equi Join: This is the most common type of join which involves only equality comparisons. The disadvantage in this type of join is that there





**SQL Questions:**

1. **Which is the subset of SQL commands used to manipulate Oracle Database structures, including tables?**

Data Definition Language (DDL)

### What operator performs pattern matching?

LIKE operator

### What operator tests column for the absence of data?

IS NULL operator

### Which command executes the contents of a specified file?

START <filename> or @<filename>

### What is the parameter substitution symbol used with INSERT INTO command?

&

### Which command displays the SQL command in the SQL buffer, and then executes it?

RUN

### What are the wildcards used for pattern matching?

For single character substitution and % for multi-character substitution

### State true or false. EXISTS, SOME, ANY are operators in SQL.

True

### State true or false. !=, <>, ^= all denote the same operation. True

1. **What are the privileges that can be granted on a table by a user to others?**

Insert, update, delete, select, references, index, execute, alter, all

### What command is used to get back the privileges offered by the GRANT command?

REVOKE

### Which system tables contain information on privileges granted and privileges obtained?

USER\_TAB\_PRIVS\_MADE, USER\_TAB\_PRIVS\_RECD

### Which system table contains information on constraints on all the tables created?

USER\_CONSTRAINTS

### TRUNCATE TABLE EMP; DELETE FROM EMP;

**Will the outputs of the above two commands differ?**





Both will result in deleting all the rows in the table EMP.

### What the difference is between TRUNCATE and DELETE commands?

TRUNCATE is a DDL command whereas DELETE is a DML command. Hence DELETE operation can be rolled back, but TRUNCATE operation cannot be rolled back. WHERE clause can be used with DELETE and not with TRUNCATE.

### What command is used to create a table by copying the structure of another table?

***Answer:***

CREATE TABLE AS SELECT command

### Explanation:

To copy only the structure, the WHERE clause of the SELECT command should contain a FALSE statement as in the following.

CREATE TABLE NEWTABLE AS SELECT \* FROM EXISTINGTABLE WHERE

1=2;

If the WHERE condition is true, then all the rows or rows satisfying the condition will be

copied to the new table.

### What will be the output of the following query?

SELECT REPLACE (TRANSLATE(LTRIM(RTRIM('!! ATHEN !!','!'), '!'), 'AN', '\*\*'),'\*','TROUBLE') FROM DUAL;

TROUBLETHETROUBLE

### What will be the output of the following query?

SELECT DECODE(TRANSLATE('A','1234567890','1111111111'), '1','YES', 'NO' );

### Answer :

NO

### Explanation :

The query checks whether a given string is a numerical digit.

### What does the following query do?

*SELECT SAL + NVL(COMM,0) FROM EMP;*

This displays the total salary of all employees. The null values in the commission column will be replaced by 0 and added to salary.

### Which date function is used to find the difference between two dates?

MONTHS\_BETWEEN





### Why does the following command give a compilation error?

DROP TABLE &TABLE\_NAME;

Variable names should start with an alphabet. Here the table name starts with an '&'

symbol.

### What is the advantage of specifying WITH GRANT OPTION in the GRANT command?

The privilege receiver can further grant the privileges he/she has obtained from the owner to any other user.

### What is the use of the DROP option in the ALTER TABLE command?

It is used to drop constraints specified on the table.

### What is the value of ‘comm’ and ‘sal’ after executing the following query if the initial value of ‘sal’ is 10000?

*UPDATE EMP SET SAL = SAL + 1000, COMM = SAL\*0.1;*

sal = 11000, comm = 1000

### What is the use of DESC in SQL?

DESC has two purposes. It is used to describe a schema as well as to retrieve rows from table in descending order.

The query SELECT \* FROM EMP ORDER BY ENAME DESC will display the output sorted on ENAME in descending order.

### What is the use of CASCADE CONSTRAINTS?

When this clause is used with the DROP command, a parent table can be dropped even when a child table exists.

### Which function is used to find the largest integer less than or equal to a specificvalue?

FLOOR

1. **What is the output of the following query?** SELECT TRUNC(1234.5678,-2) FROM DUAL; 1200







**SQL HANDSON**

**Questions Based On Data Definition Language Commands:**

* 1. **Create A Table EMP And DEPT Using The Following Information.**

### DEPT:

COLUMN NAME DATATYPE (SIZE)

DEPTNO DNAME LOC

NUMBER (2)

VARCHAR2 (14)

VARCHAR2 (14)

### EMP:

COLUMN NAME DATATYPE(SIZE)

EMPNO ENAME JOB MGR

HIREDATE SAL COMM DEPTNO

NUMBER (4)

VARCHAR2 (10)

VARCHAR2 (9)

NUMBER (4) DATE NUMBER (7, 2)

NUMBER (7, 2)

NUMBER (2)

### Check the Default Size of a Number, Char and Date Data types.

* 1. **Describe the Structure of the Table and EMP Table.**
  2. **Add two columns to the table EMP with the following information in one single**

### ALTER COMMAND.

COLUMN NAME DATATYPE (SIZE)

SEX CHAR (1)

PLACE CHAR (15)





### Modify the column job present in the EMP table with the following information given below:

COLUMN NAME DATATYPE (SIZE)

JOB VARCHAR2 (15)

### modify the column ENAME present in the EMP table with the following information given below:

COLUMN NAME DATATYPE (SIZE)

ENAME CHAR (15)

### Decrease the size for the column EMPNO with the following information:-

COLUMN NAME DATATYPE (SIZE)

EMPNO NUMBER (2)

### Modify the column name of EMPNO to EMPLOYEE\_NUMBER present in the EMP table verify the result.

* 1. **Add a new column nationality placed between JOB and MGR columns and verify the result**

### Drop the table DEPT and EMP.

* 1. **What is the data type of the column HIREDATE and how many bytes it occupies.**

### Create a table EMP and DEPT using the following information.

|  |  |  |
| --- | --- | --- |
| **a. DEPT:** |  |  |
| COLUMN NAME | DATATYPE(SIZE) | |
|  | | |
| DEPTNO | NUMBER(2) CONSTRAINT | PK\_DEPTNO |
|  | PRIMARY KEY |  |
| DNAME | VARCHAR2(14) |  |
| LOC | VARCHAR2(14) |  |
| **b.EMP:** |  |  |
| COLUMN NAME | DATATYPE(SIZE) |  |
|  |  |  |
| DEPT OF CS&E,ATRIA I T, Bangalore |  | Page 69 |



|  |  |  |  |
| --- | --- | --- | --- |
| EMPNO | NUMBER(4)CONSTRAINTPK\_EMPNO | | |
|  | PRIMARY KEY |  |  |
| ENAME | VARCHAR2(10) CONSTRAINT | | UQ\_DEPTNO |
|  | UNIQUE |  |  |
| JOB | VARCHAR2(9) |  |  |
| MGR | NUMBER(4) |  |  |
| HIREDATE | DATE DEFAULT SYSDATE | |  |
| SAL | NUMBER(7,2) | CONSTRAINT | CK\_SAL |
|  | CHECK(SAL) |  |  |
| COMM | NUMBER(7,2) |  |  |
| DEPTNO | NUMBER(2)CONSTRAINTFK\_DEPTNO | | |
|  | REFERENCE |  |  |

* 1. **Select all the constraints in the EMP table**

SOL: SELECT \* FROM USER\_CONSTRAINTS WHERE TABLE\_NAME=’EMP’;

### select the owner, constraints name, constraints type, table name, status for DEPT table

SOL: SELECT OWNER, CONSTRAINTS\_ NAME, CONSTRAINTS\_TYPE, TABLE\_ NAME, STATUS FROM USER\_CONSTRAINTS WHERE TABLE\_NAME=’DEPT’;

### Drop the constraints UQ\_FMANE from EMP table

SOL: ALTER TABLE EMP DROP CONSTRAINT UQ\_FNAME;

### Add a new column PINCODE with not null constraints to the existing table DEPT

SOL: ALTER TABLE DEPT ADD(PINCODE NUMBER(6) NOT NULL);

### Disable the constraints PK\_DEPTNO present in the DEPT table

SOL: ALTER TABLE DEPT DISABLE CONSTRAINTS PK\_DEPTNO;

### Enable the constraints PK\_DEPTNO which is defined in the DEPTNO column of DEPT table;

SOL: ALTER TABLE DEPT ENABLE CONSTRAINTS PK\_DEPTNO;

### Insert the given values into the tables: EMP:





(i) 7369, SMITH, CLERK, 7902, 17 – DEC – 80, 800, NULL, 20

(ii) 7499, ALLEN SALEMAN, 7698, 20 – FEB – 81, 1600, 300, 30

(iii) 7521, WARD, SALESMAN, 7698, 22 – FEB – 81, 1250, 500, 30

(iv) 7566, JONES, MANAGER, 7839, 02 – APR – 81, 2975, NULL, 20

(v) 7654, MARTIN, SALESMAN, 7698, 28 – SEP – 81, 1250, 1400, 30

(vi) 7698, BLAKE, MANAGER, 7839, 01 – MAY – 81, 2850, NULL, 30

(vii) 7782, CLERK, MANAGER, 7839, 09 – JUN – 81, 2450, NULL, 10

1. 7788, SCOTT, ANALYST, 7566, 19 – NOV – 96, 3000, NULL, 20
2. 7839, KING, PRESIDENT, NULL, 17 – NOV – 81, 5000, NULL, 10 (x) 7844, TURNER, SALESMAN, 7698, 08 –SEP – 81, 1500, 0, 30

(xi) 7876, ADAMS, CLERK , 7788, 23 – DEC – 96, 1100, NULL, 20

(xii) 7900, JAMES, CLERK, 7698, 03 – DEC – 81, 950, NULL, 30

(xiii) 7902, FORD, ANALYST, 7566, 03 – DEC – 81, 3000, NULL, 20

(xiv) 7934, MILLER, CLERK, 7782, 23 – JAN – 82, 1300, NULL, 10 (xv) 7943, JOHN, CLERK, 7943, 10 – DEC – 83, 2000, NULL, 50

### DEPT:

1. 10, ACCOUNTING, NEW YORK
2. 20, RESEARCH, DALLAS (iii)30, SALES, CHICAGO
3. 40, OPERATIONS, BOSTON
4. 50, COMPUTER, AMERICA

### Insert only the records of employee number, name, salary into EMP table

SOL: INSERT INTO EMP (EMPNO, ENAME, SAL) VALUES (‘7955’, ‘PAUL’, 1200);

### insert two rows into EMP table using parameter substitution

SOL: INSERT INTO EMP VALUES (&EMPNO, ‘&ENAME’, ‘&JOB’, &MGR, ‘&IIIREDATE’, &SAL, &COMM, &DEPTNO);

### insert the current transaction date temporary table

SOL: INSERT INTO TEMP VALUES (SYSDATE);





**Problems on Select Command:**

* 1. **List the Information of all Employees**

SOL: SELECT \* FROM EMP;

### List the information of all the departments

SOL: SELECT \* FROM DEPT;

### LIST THE DEPARTMENT NUMBERS, EMPLOYEE NUMBERS AND THEIR MANAGERS NUMBERS

SOL: SELECT DEPT. DEPTNO, EMP. EMPNO, EMP. MGR FROM EMP, DEPT;

### List department name and locations from DEPT table

SOL: SELECT DNAME, LOC, FROM DEPT;

* 1. **List the information of employees and their departments in a single DMI command** SOL: SELECT \*. FROM EMP;

### Copy all the records of their columns EMPNO, ENAME, JOB from EMP table and insert the records into a temp table with column names same as EMPNO, ENAME, JOB

SOL: INSERT INTO TEMP (EMPNO, ENAME, JOB) SELECT EMPNO, NAME, JOB

### List the details of both the tables

SOL: SELECT \* FROM EMP, DEPT;

### List the information of all the employees present in the user named SCOTT

SOL: SELECT \* FROM SCOTT. EMP;

### List the information of the departments from your BATCHMATES DEPT table

SOL: SELECT \* FROM ORA252P. DEPT;

### List out the table names in your schema

SOL: SELECT \* FROM TAB;

### List all the system tables

SOL: SELECT \*FROM SYS. DICTIONARY;

### Get the information of the maximum available blocks allotted to a particular user for creating tables from the system tables

SOL: SELECT \* FROM USER\_TS\_QUOTAS;

### List out all the privileges given to a particular user

* 1. **List out all the tables which start with ‘S’**



SOL: SELECT TABLE\_NAME FROM USER\_TABLES WHERE TABLE\_NAME LIKE ‘S%’;

### Copy the structure of dept table alone to a temporary table called TEMP1

SOL: CREAT TABLE TEMP1, AS SELECT \* FROM DEPT WHERE 1=2;

**Problems on update command:**

* 1. **Update the salary by 10% hike to analysts working in department number 20 and 30**

SOL: UPDATE EMP SET SAL = SAL + 0. 1 WHERE DEPTNO IN (10,20) AND JOB

=‘ANALYST’;

### Give 5% raise in salary to all the salesman

SOL: UPDATE EMP SET SAL=SAL+.5 WHERE JOB = ‘SALESMAN’;

* 1. PROMOTE ALL THE EMPLOYEES DISCRIBED AS SALESMAN TO SALES OFFICER IF THEIR GROSS SALARY PER MOUNTH IS GREATER THEN 3000 SOL:

UPDATE EMP SET JOB = ‘SALESOFFICER’ WHERE JOB = ‘SALESMAN’

AND SAL>3000;

* 1. **Give all the employees of commission of rs500**

SOL: UPDATE EMP SET COMM = 500;

### Change the department of JAMES to 20

SOL: UPDATE EMP SET DEPTNO = 20 WHERE ENAME = ‘JAMES’;

### Calculate all the employees total salary with commission

SOL: SELECT SALES SAL +NVL (COMM) “TOTAL” FROM EMP;

**Problems on delete command:**

* 1. **Delete all the records of employees**

SOL: DELETE FROM EMP;

### Get back the original records of employees back

SOL: ROLLBACK;

### Allen’s record only

SOL: DELETE FROM EMP WHERE ENAME = ‘ALLEN’;

### Delete records of ename column only and verify it





SOL: NOT POOSIBLE

### Delete the records of employee number 7782

SOL: DELETE FROM EMP WHERE EMPNO=7782;

### Delete the employee’s records who doesn’t have commission

SOL: DELETE FROM EMP WHERE COMM IS NULL;

### Get back the original records back

SOL: ROLLBACK;

### Delete the duplicate records of the employee table

SOL: DELETE FROM EMP A WHERE ROWID<> (SELECT MIN (ROWID) FROM EMP B WHERE A. EMPNO=B. EMPNO);

### Delete the first five records of employee table

SOL: DELETE FROM EMP X WHERE 5 > (SELECT COUNT (ROWID) FROM EMP Y WHERE Y. ROWID < X.ROWID);

### Delete the rows of the temp table permanently

SOL: TURNCATE TABLE TEMP;

**Problems on transactional commands:**

* 1. **Update a record of EMP table and save the changes permanently in the database**

SOL: UPDATE EMP SET SAL=SAL+100 WHERE EMPNO=100; COMMIT;

### Sql \* plus has the facility to automatically save all the records without issuing the TCL command which is that?

SOL: SET AUTOCOMMIT ON

### Give all the privileges you have of a database object to another

SOL: GRANT ALL ON EMP TO ORA253A;

### Give only select, insert privileges to another user

SOL: GRANT SELECT, INSERT ON EMP TO ORA267A;

### List the user’s id and which database object you have granted

SOL: SELECT \* FROM USE\_TAB\_PRIVS;







**ADDITIONAL EXERCISES ON DATABASE APPLICATION LABORATORY**

**Exercise 1:** To understand some simple Database Applications and build Conceptual Data Model.

1. Select an enterprise that you are familiar with (for example, a school, a college, a company, a small business, a club or association). List all the information that this enterprise uses.
2. Describe the steps involved in the database design process using E-R Modeling: *Requirements Analysis, Identify Entity Sets, Identify Relationship Sets, Value Sets and Attributes, Specifying Primary keys, Building E-R diagram, Implementation*
3. For the following mini-world example database applications, Design and Develop Conceptual Data Model (E-R Diagram) with all the necessary entities, attributes, constraints and relationships.
   1. ***Medical Clinic Database*** – The clinic has a number of regular patients and new patients come to the clinic regularly. Patients make appointments to see one of the doctors; several doctors attend the clinic and they each have their own hours. Some doctors are General Practitioners (GPs) while others are specialists(cardiologists, dermatologists etc.,). Patients have families and the family relationships are important. A medical record of each patient needs to be maintained. Information on prescriptions, insurance, allergies, etc needs to be maintained. Different doctors may charge different fees. Billing has to be done for Patients.
   2. ***University Database*** - The Visvesvaraya Technological University (VTU) is a large Institution with several campuses scattered across Karnataka. Academically, the university is divided into a number of faculties, such as Faculty of Engineering, Faculty of Architecture, Faculty of Management and Faculty of Science. Some of the Faculties operate on a number of campuses. Faculties, in turn, are divided into schools; for example, the School of Architecture, the School of Information Technology. Each school is headed by a Director and has a number of teaching and non-teaching staff. Each school offers many courses. Each course consists of a fixed core of subjects and a number of electives from other courses. Each student in the University is enrolled in a single course of study. A subject is taught to the students



who have registered for that subject by a teacher. A student is awarded a grade in each subject taken.

* 1. ***Construction Company Database*** - A construction company has many branches spread all over the country. The company has two types of constructions to offer: Housing and Commercial. The housing company provides low-income housing, medium-style housing, and high-end housing schemes, while commercial side, it offers multiplexes and shopping zones. The customers of the company may be individuals or corporate clients. Company stores the information about employees’ works for it.
  2. ***Time Table Preparation*** - An Engineering College has a number of Branches. Each Branch has number sections, a number of courses and a number of faculty members teaching the courses. Each branch has a number of class rooms and laboratories. Each course may be scheduled in a class room at a particular time.

***Note***: Similar applications may be explored and given as assignments to students in a group.

**Exercise 2:** Design and build Relational Data Model for each of the application scenarios of exercise 1 specifying all possible constraints. Extend the same for a database application of students' choice.

**Exercise 3** To understand and demonstrate DDL, DML and DCL Commands of SQL

1. Create a table called EMP with the following structure and describe it.

Name Type

EMPNO NUMBER (6)

ENAME VARCHAR2 (20)

DOB DATE

JOB VARCHAR2 (10)

DEPTNO NUMBER (2)

SAL NUMBER (7,2)





Allow NULL for all columns except ENAME and JOB. EMPNO is the Primary

Key

1. Add a column EXPERIENCE of type NUMERIC to the EMP table. Allow NULL to it.
2. Modify the column width of the JOB field of EMP table.
3. Create DEPT table with the following structure and describe it Name Type

DEPTNO NUMBER (2)

DNAME VARCHAR2 (15)

LOCN VARCHAR2 (10)

DEPTNO is the Primary Key and DNAME cannot be NULL

1. Add constraint to check the SAL value of EMP Table. SAL must be > 6000.
2. Drop a column EXPERIENCE from the EMP table.
3. Insert a single record into DEPT table. Repeat this for inserting at least 3 records
4. Insert more than a record into EMP table using a single insert command. Insert at least 10 records
5. Update the EMP table to set the salary of all employees to Rs. 30000/- for a given JOB type
6. Create a pseudo table EMPLOYEE with the same structure as the table EMP using SELECT clause.
7. Delete employees from EMP table for a given JOB type. Delete the first five records of EMP table
8. Grant all/some privileges of EMP table to DEPT table
9. Revoke some/all privileges of EMP table from DEPT table
10. Truncate the EMP table and drop the DEPT table
11. Demonstrate the use of COMMIT, SAVEPOINT and ROLLBACK commands



