**`CONNECTING TO THE MY-SQL COMMAND LINE**

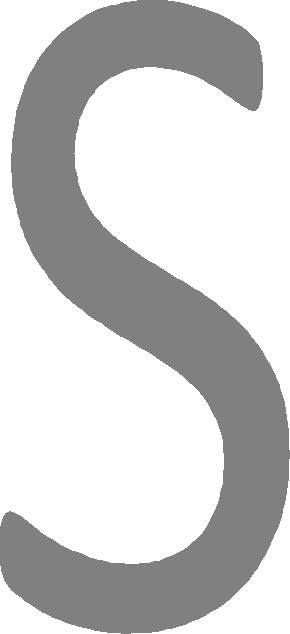
Click on start->All Programs-> Oracle Database 10g Express Edition-> Run SQL Command Line





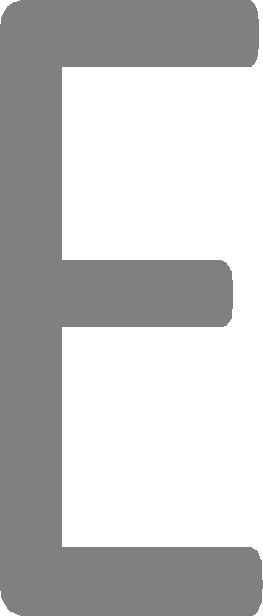






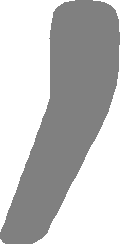














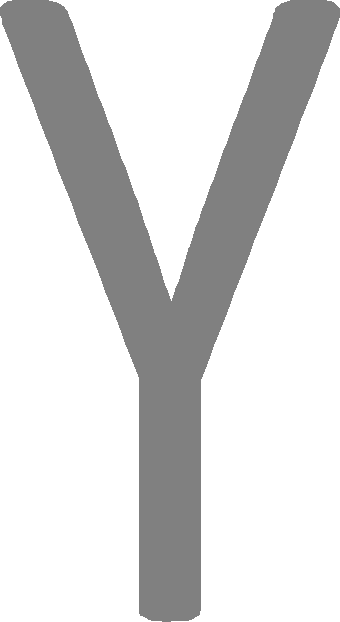






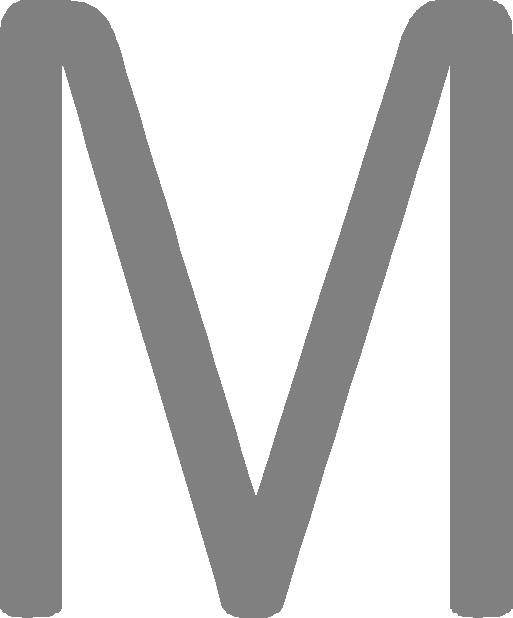






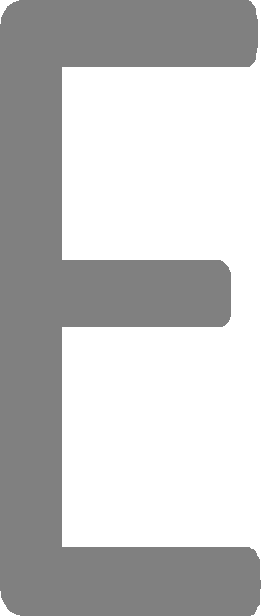






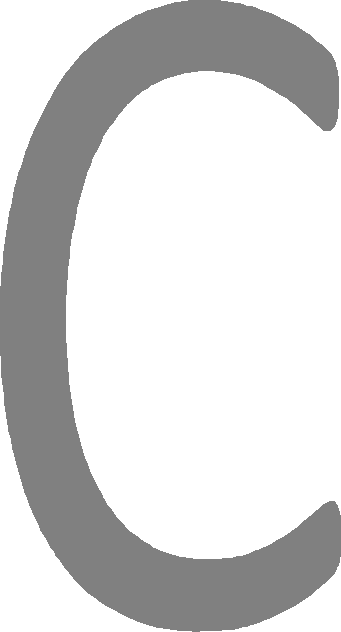




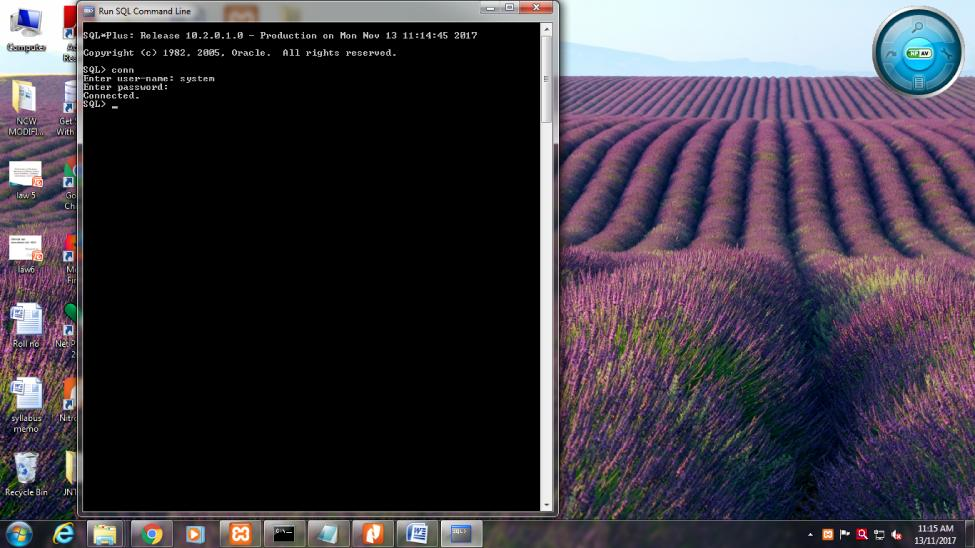












SQL> conn

Enter user-name: system

Enter password:12345

Connected.

**EXPERIMENT-1**

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

**BOOK(Book\_id, Title, Name, Pub\_Year)**

**BOOK\_AUTHORS(Book\_id, Author\_Name)**

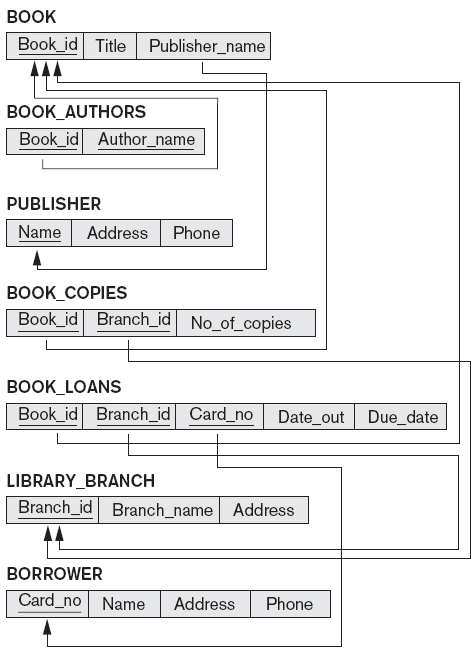
**PUBLISHER(Name, Address, Phone)**

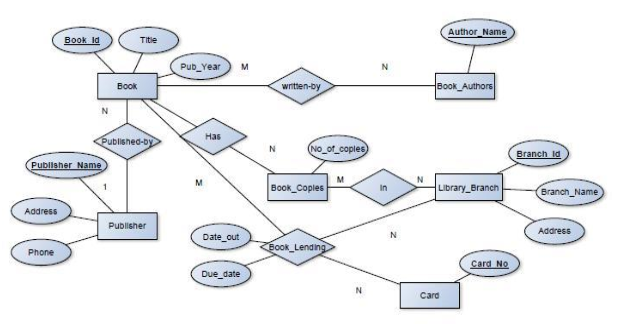
**BOOK\_COPIES(Book\_id, Branch\_id, No-of\_Copies)**

**BOOK\_LENDING(Book\_id, Branch\_id, Card\_No, Date\_Out, Due\_Date)**

**LIBRARY\_BRANCH(Branch\_id, Branch\_Name, Address)**

**SCHEMA DIAGRAM OF LIBRARY DATABASE**



****

|  |
| --- |
| Publisher    sql> create table publisher(  name varchar(18) primary key,  address varchar(10),  phone varchar(10));  table created. |
| Book  sql> create table book(  bkid number primary key,  title varchar(20),  pname varchar(20)references publisher(pname)on delete  casade,  pyear number(4));  table created. |
| book\_authors    sql> create table book\_authors(  bkid number references book(bkid) on delete cascade,  aname varchar(20),  primary key(book\_id));  table created. |
| library  sql> create table library(  branchid number primary key,  branch\_name varchar(18),  address varchar(15));  table created. |
| bookcopies    sql> create table bookcopies(  book\_id number references book(bkid) on delete cascade,  branch\_id number references library(branchid) on deletecascade,  numcopies number);  table created. |

**DESCRIPTION**

|  |
| --- |
| **SQL> desc book;**  Name Null? Type  ----------------------------------------- -------- --------------------  BKID NOT NULL VARCHAR2(5)  TITLE VARCHAR2(10)  PNAME VARCHAR2(10)  PYEAR NUMBER(4) |
| **SQL> desc bauthor;**  Name Null? Type  ----------------------------------------- -------- ---------------  BKID VARCHAR2(5)  ANAME VARCHAR2(10) |
| **SQL> desc bklending;**  Name Null? Type  ----------------------------------------- -------- --------------------  BKID VARCHAR2(5)  BRANCHID VARCHAR2(10)  CARDNO NUMBER(4)  DATE\_OUT DATE  DUE\_DATE DATE |
| **SQL> desc bookcopies;**  Name Null? Type  ----------------------------------------- -------- ----------------  BKID VARCHAR2(10)  BRANCHID VARCHAR2(10)  NUMCOPIES NUMBER |
| **SQL> desc publ;**  Name Null? Type  ----------------------------------------- -------- ---------------  PNAME NOT NULL VARCHAR2(10)  ADDRESS VARCHAR2(20)  PHONE NUMBER(10) |
| **SQL> desc library;**  Name Null? Type  ----------------------------------------- -------- ------------------  BRANCHID NOT NULL VARCHAR2(5)  BRNAME VARCHAR2(10)  ADDR VARCHAR2(10) |

**INSERT AND DISPLAY COMMANDS**

1. Book table

|  |
| --- |
| SQL> Insert into book values(&bkid,’&title’,’&pname’,&pyear)  SQL> select \* from book;  BKID TITLE PNAME PYEAR  ----- ---------- ---------- ----------  cs111 C pearson 2010  cs112 c++ pearson 2012  cs113 DBMS PHI 2012  cs115 oops pearson 2015  cs116 C PHI 2013  cs117 C++ gold 2016  cs118 cn gold 2015  7 rows selected |

1. Publisher table

|  |
| --- |
| SQL> Insert into publ values(‘&pname,’&paddress’,&phone);  SQL> select \* from publ;  PNAME ADDRESS PHONE  ---------- -------------------- ----------  PHI delhi 9877777777  pearson delhi 9812345678  gold bengaluru 984412345  star mysore 9898986666 |

1. Bookcopies table

|  |
| --- |
| SQL> Insert into bookcopies values(&bkid,&branched,&numcopies);  SQL> select \* from bookcopies;  BKID BRANCHID NUMCOPIES  ---------- ---------- ----------  cs111 ryis1 5  cs112 ryis1 4  cs113 ryis1 8  cs115 rym2 10  cs117 rym2 7 |

1. Book lending table

|  |
| --- |
| SQL> g  SQL> select \* from bklending;  BKID BRANCHID CARDNO DATE\_OUT DUE\_DATE  ----- ---------- --------- --------- ---------  cs111 ryis1 2222 17-AUG-16 20-SEP-16  cs113 rym2 2223 03-JUL-16 04-AUG-16  cs115 ryis1 2226 03-MAY-16 05-JUN-16  cs116 ryis1 2224 03-JUL-16 05-AUG-16  cs113 ryis1 2223 04-JUN-17 05-JUL-17  cs113 rym2 2223 03-JAN-17 04-FEB-17  cs116 rym2 2223 05-MAR-17 07-APR-17  cs118 rym2 2223 06-MAY-17 28-MAY-17 |

8 rows selected.

1. Library Table

|  |
| --- |
| SQL> Insert into libraray values(‘&branchid’,’&bname’,’&addr’);  SQL> select \* from library;  BRANC BRNAME ADDR  ----- ------ ----------  ryis1 ISE ise dept  rym2 cse cse dept  rym3 enc enc dept  rym4 eee eee dept  rym5 civ civ dept |

1. Book Author Table

|  |
| --- |
| SQL> Insert into bauthor(&bkid’,’&aname’);  SQL> select \* from bauthor;  BKID ANAME  ----- ----------  cs111 brain  cs112 scott  cs117 harry  cs113 navathe  cs115 subhash |

**QUERIES**

1. **Retrieve details of all the books in the library as (BKID,TITLE,PUBLISHER NAME ,AUTHOR ,NO\_OF\_COPIES)**

|  |
| --- |
| SQL> select b.bkid,b.title,b.pname,ba.aname,bc.branchid,bc.numcopies  2 from book b,bauthor ba,bookcopies bc  3 where b.bkid=ba.bkid and  4 b.bkid=bc.bkid;  BKID TITLE PNAME ANAME BRANCHID NUMCOPIES  ----- ---------- ---------- ---------- ---------- ----------  cs111 C pearson brain ryis1 5  cs112 c++ pearson scott ryis1 4  cs117 C++ gold harry rym2 7  cs113 DBMS PHI navathe ryis1 8  cs115 oops pearson subhash rym2 10 |

1. **Get the particulatrs of borrowers who have borrowed more than three books but from JAN-2017 to jun-2017**

|  |
| --- |
| SQL> select b.bkid,bl.branchid,bl.cardno  2 from book b,bklending bl  3 where b.bkid=bl.bkid and bl.cardno  4 in  5 (select cardno  6 from bklending bl  7 where due\_date between '1-jan-2017' and '1-jul-2017' and  8 date\_out between '1-jan-2017' and '1-jul-2017'  9 group by cardno having count(cardno)>=3);  BKID BRANCHID CARDNO  ----- ---------- ----------  cs118 rym2 2223  cs116 rym2 2223  cs113 rym2 2223  cs113 rym2 2223 |

1. **Delete a book in BOOK table.update the contents of other tables to**

**reflect this manipulatio operation.**

|  |
| --- |
| **sql> delete from books**  **whree bkid=115;**  **I row deleted** |

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

|  |
| --- |
| select bkid, btitle, pname, pyear  from book  group by pyear, bkid, btitle, pname; |

1. **create a view of all books and its no of copies that are currently**

**available in the library;**

|  |
| --- |
| **create view LIBRARY\_BOOKS\_DB**  **as**  **select b.bkid,b.title,bc.numcopies**  **from book b,bookcopies bc**  **where b.bkid=bc.bkid;**  **view cerated** |

**Out put of view**

|  |
| --- |
| **SQL> select \* from LIBRARY\_BOOKS\_DB;**  **BKID TITLE NUMCOPIES**  **----- ---------- ----------**  **cs111 C 5**  **cs112 c++ 4**  **cs113 DBMS 8**  **cs115 oops 10**  **cs117 C++ 7** |

**EXPERIMENT-2**

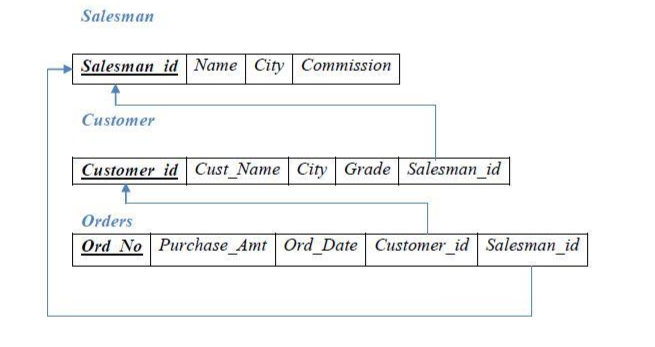
2) Consider the following schema for Order Database:

SALESMAN(Salesman\_id, Name, City, Commission)

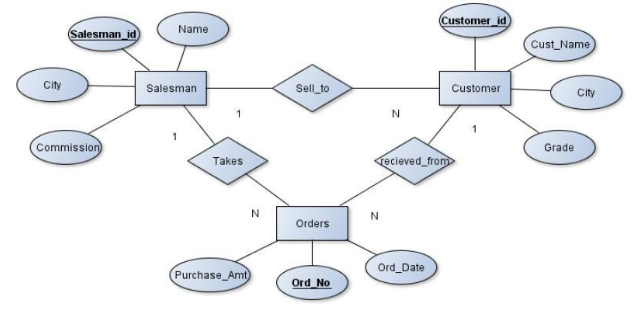
CUSTOMER(Customer\_id, Cust\_Name, City, Grade, Salesman\_id)

ORDERS(Ord\_No, Purchase\_Amt, Ord\_Date, Customer\_id, Salesman\_id)

SCHEMA DIAGRAM



E-R DIAGRAM



**CREATION OF TABLES:**

1. Sales

|  |
| --- |
| SQL> create table sales  2 (sid number primary key,  3 sname varchar(10),  4 scity varchar(10),  5 commission number);  Table created. |
| SQL> desc sales  Name Null? Type  ----------------------------------------- -------- ------------------------  SID NOT NULL NUMBER  SNAME VARCHAR2(10)  SCITY VARCHAR2(10)  COMMISSION NUMBER |

1. Customer

|  |
| --- |
| SQL> create table customer  2 (cid number primary key,  3 cname varchar(10),city varchar(10),  4 grade number(5),  5 sid number references sales(sid) on delete cascade);  Table created. |
| SQL> desc customer;  Name Null? Type  ----------------------------------------- -------- -------------------  CID NOT NULL NUMBER  CNAME VARCHAR2(10)  CITY VARCHAR2(10)  GRADE NUMBER(5)  SID NUMBER |

1. Orders

|  |
| --- |
| SQL> create table orders  2 (ono number(5) primary key,  3 pamount number,  4 odate date,  5 cid number references customer(cid) on delete cascade,  6. sid number references sales(sid) on delete cascade);  Table created. |
| SQL> desc orders;  Name Null? Type  ----------------------------------------- -------- -------------------------  ONO NOT NULL NUMBER(5)  PAMOUNT NUMBER  ODATE DATE  CID NUMBER  SID NUMBER |

**Insertion And Display Of Tables:**

SQL> insert into sales(&sid,'&sname','&scity',&commission);

SQL> select \* from sales;

|  |
| --- |
| SID SNAME CITY COMM |
| ---------- ---------- ---------- ---------- |
| 111 David blore 10 |
| 112 sam mysore 20 |
| 113 bob mumbai 20 |
| 114 ram bly 20 |
|  |
|  |

2) SQL> insert into customer values(&cid,'&cname','&city',&grade,&sid);

SQL> select \* from customer;

|  |
| --- |
| SQL> select \* from customer; |
|  |
| CID CNAME CITY GRADE SID |
| ---------- ---------- ---------- ---------- ---------- |
| 221 priya blore 100 111 |
| 222 suma bly 200 111 |
| 223 mary blore 200 111 |
| 224 raj blore 100 111 |
| 225 ravi mysore 500 112 |

SQL> select \* from orders;

|  |
| --- |
| SQL> select \* from orders;  ONO PAMOUNT ODATE CID SID  ---------- ---------- --------- ---------- ----------  551 6574 02-JAN-17 221 111  552 43251 05-FEB-17 222 113  553 3526 06-MAR-17 221 111  554 3527 16-MAR-17 224 113  556 2300 2-JAN-17 225 111  557 3242 05-APR-17 226 112  558 2435 20-JAN-17 224 114 |

**Queries**

Q1) Count the customers with grades above the bangalore's average.

|  |
| --- |
| SQL> select grade,count(\*) as NO\_OF\_CUSTOMERS  2 from customer  3 group by grade  4 having grade>(select avg(grade)  5 from customer  6 where city = 'blore' );  GRADE NO\_OF\_CUSTOMERS  ---------- ---------------  500 1  300 1  200 2 |

Q2) find the name and no of all salesman who had more than one customer.

|  |
| --- |
| SQL> select sname,sid  2 from sales  3 where sid in  4 (select sid  5 from orders  6 group by sid  7 having count(cid)>1);  SNAME SID  ---------- ----------  David 111  bob 113 |

Q3) List all the salesman and indicate those who have and dont have

cutomers in their city(use UNION operation)

|  |
| --- |
| SQL> (select s.sid ,s.sname,s.scity,c.cid,c.city,c.cname  2 from sales s,customer c  3 where s.sid=c.sid and c.city=s.scity)  4 UNION  5 (select s.sid ,s.sname,s.scity,c.cid,c.city,'NO CUSTOMERS IN CITY'  6 from sales s,customer c  7 where s.sid=c.sid and c.city!=s.scity);  SID SNAME SCITY CID CITY CNAME  ---------- ---------- ---------- ---------- ---------- --------------------  111 David blore 221 blore priya  111 David blore 222 bly NO CUSTOMERS IN CITY  111 David blore 223 blore mary  111 David blore 224 blore raj  112 sam mysore 225 mysore ravi  112 sam mysore 226 mysore alice  114 ram bly 227 blore NO CUSTOMERS IN CITY  7 rows selected. |

Q4) Create a view that finds the salesman who has the customer with the highest order of a day.

|  |
| --- |
| SQL> create view HIGHEST\_ORDERS  2 as select s.sid,s.sname,o1.odate,o1.pamount  3 from sales s,orders o1  4 where s.sid=o1.sid and o1.pamount=  5 (select max(o2.pamount) from orders o2 where  6 o1.odate=o2.odate);  View created.  SQL> select \* from HIGHEST\_ORDERS;  SID SNAME ODATE PAMOUNT  ---------- ---------- --------- ----------  111 David 02-JAN-17 6574  113 bob 05-FEB-17 43251  111 David 06-MAR-17 3526  113 bob 16-MAR-17 3527  112 sam 05-APR-17 3242  SQL> select \* from orders;  ONO PAMOUNT ODATE CID SID  ---------- ---------- --------- ---------- ----------  551 6574 02-JAN-17 221 111  552 43251 05-FEB-17 222 113  553 3526 06-MAR-17 221 111  554 3527 16-MAR-17 224 113  556 2300 02-JAN-17 225 111  557 3242 05-APR-17 226 112  558 2435 02-JAN-17 224 114  7 rows selected. |

Q5) Demonstrate the DELETE operation by removing salesman with ID 1000.all his orders must also be deleted

|  |
| --- |
| sql> delete from sales  where sid=118; |

**LAB PROGRAM 3: MOVIES DATABASE**

**C. 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**

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

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

**3. List all actors who acted in a movie before 2000 and also in a movie after**

**2015 (use JOIN operation).**

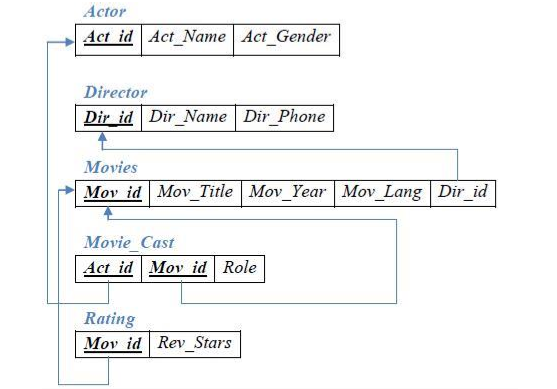
**4. Find the title of movies and number of stars for each movie that has at least one**

**rating and find the highest number of stars that movie received. Sort the result by**

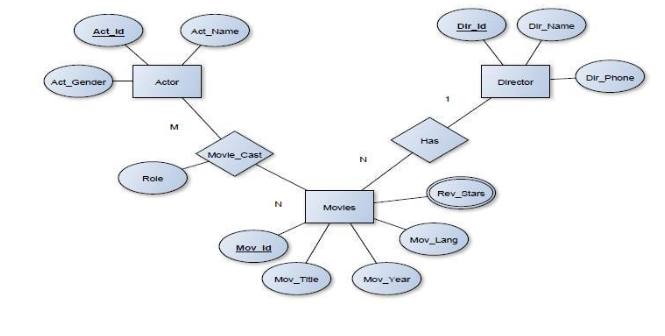
**movie title.**

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

**SCHEMA -DIAGRAM**

****

**E-R DIAGRAM**

****

**CREATION OF TABLES**

1. ACTOR

|  |
| --- |
| SQL> create table actor  2 (actid number primary key,  3 aname varchar(10),  4 gender varchar(6));  Table created. |
| SQL> desc actor;  Name Null? Type  ----------------------------------------- -------- ------------------  ACTID NOT NULL NUMBER  ANAME VARCHAR2(10)  GENDER VARCHAR2(6) |

1. DIRECTOR

|  |
| --- |
| SQL> create table actor  2 (actid number primary key,  3 aname varchar(10),  4 gender varchar(6));  Table created. |
| SQL> desc director;  Name Null? Type  ----------------------------------------- -------- ---------------------  DID NOT NULL NUMBER(5)  DNAME VARCHAR2(10)  PHONE NUMBER(10) |

1. Movies

|  |
| --- |
| SQL> create table movies  2 (mid number primary key,  3 mtitle varchar(10),myear number,  4 mlang varchar(10),  5 did number(5) references director(did) on delete cascade);  Table created. |
| SQL> desc movies;  Name Null? Type  ----------------------------------------- -------- -------------------  MID NOT NULL NUMBER  MTITLE VARCHAR2(10)  MYEAR NUMBER  MLANG VARCHAR2(10)  DID NUMBER(5) |

1. Movie\_cast

|  |
| --- |
| SQL> create table movie\_cast  2 (actid number references actor(actid) on delete cascade,  3 mid number references movies(mid) on delete cascade,  4 role varchar(10));  Table created. |
| SQL> desc movie\_cast;  Name Null? Type  ----------------------------------------- -------- ----------------------  ACTID NUMBER  MID NUMBER  ROLE VARCHAR2(10) |

1. Rating

|  |
| --- |
| SQL> create table rating  2 (mid number references movies(mid) on delete cascade,  3 revstars number);  Table created. |
| SQL> desc rating;  Name Null? Type  ----------------------------------------- -------- ---------------  MID NUMBER  REVSTARS NUMBER |

Insert and display commands

1. Actor

SQL> insert into actor values(&aid,'&aname','&gender');

SQL> select \* from actor;

|  |
| --- |
| ACTID ANAME GENDER  ---------- ---------- ------  111 sam male  112 bob male  113 ariyana female  114 david male  115 jim male  116 kim female  117 puneeth male |

1. Director

insert into director values(&did,'&dname',&phone);

|  |
| --- |
| SQL> select \* from director;  DID DNAME PHONE  ---------- ---------- ----------  2111 hitchcock 5647382  2112 steven 657484  2113 johnwatts 56767  2114 santosh 56764 |

1. Movies

SQL> insert into movies values(&mid,'&mtitle',&myear,'&mlang',&did)

|  |
| --- |
| SQL> select \* from movies;  MID MTITLE MYEAR MLANG DID  ---------- ---------- ---------- ---------- ----------  500 jpark 2013 english 2112  501 rwindow 1954 english 2111  503 spiderman 2017 english 2113  504 rajkumara 2017 kannada 2114 |

1. Movie cast

SQL> insert into moive\_cast values(&aid,&mid,'&role');

|  |
| --- |
| SQL> select \* from movie\_cast;  ACTID MID ROLE  ---------- ---------- ----------  111 501 mainlead  112 503 supporting  113 501 heroin  114 503 mainlead  115 503 negative  116 503 heroin  117 504 hero  111 503 supporting  112 501 hero  114 500 mainlead  115 500 supporting |

1. Rating

SQL> insert into rating values(&mid,&revstars);

|  |
| --- |
| SQL> select \* from rating;  MID REVSTARS  ---------- ----------  500 5  501 3  503 3  504 4  500 3  501 2  504 4  503 4 |

**III Queries:**

Q1) List the titles of all the movies directed by "hitchcock".

|  |
| --- |
| SQL> select m.mid ,m.mtitle  2 from movies m,director d  3 where m.did=d.did and d.dname='hitchcock';  MID MTITLE  ---------- ----------  501 rwindow |

Q2)Find the movie names where one or more actors acted in two or more movies**.**

|  |
| --- |
| SQL> select m.mtitle  2 from movies m, movie\_cast mc  3 where m.mid=mc.mid and mc.actid in  4 (select actid from movie\_cast  5 group by actid  6 having count(actid)>=2)  7 group by m.mtitle  8 having count(\*)>2;  MTITLE  ----------  spiderman |

Q3) List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

New query out put

|  |
| --- |
| SQL> select a.aname,m1.mtitle,m1.myear,m2.myear  2 from actor a, movies m1, movies m2, movie\_cast c1, movie\_cast c2  3 where m1.myear < 2000 and m2.myear > 2015 and a.actid = c1.actid  4 and c1.mid = m1.mid and a.actid = c2.actid and c2.mid = m2.mid;  ANAME MTITLE MYEAR MYEAR  ---------- ---------- ---------- ----------  bob rwindow 1954 2017  sam rwindow 1954 2017 |

Old query out put

|  |
| --- |
| SQL> select a.aname,m.mtitle,m.myear  2 from actor a join movie\_cast mc on a.actid=mc.actid  3 join movies m on m.mid=mc.mid  4 where m.myear not between 2000 and 2015;    ANAME MTITLE MYEAR  ---------- ---------- ----------  sam rwindow 1954  bob spiderman 2017  ariyana rwindow 1954  david spiderman 2017  jim spiderman 2017  kim spiderman 2017  puneeth rajkumara 2017  sam spiderman 2017  bob rwindow 1954 |

Q4) Find the title of movies and number of stars for each movie that has at least one rating, find the highest number of stars that movie received. Return the movie title and number of stars. Sort by movie title.

|  |
| --- |
| SQL> select m.mtitle,max(r.revstars) from rating r  2 ,movies m  3 where m.mid=r.mid  4 group by m.mtitle  5 order by m.mtitle;  MTITLE MAX(R.REVSTARS)  ---------- ---------------  jpark 5  rajkumara 4  rwindow 3  spiderman 4 |

Q5) Update rating of all movies directed by ‘Steven Spielberg’ to 5

|  |
| --- |
| SQL> update rating set revstars=5 where mid in  2 (select m.mid  3 from movies m,director d  4 where m.did=d.did and dname='steven');  2 rows updated.  SQL> select \* from rating; |

**LAB-PGM:4 COLLGE DATABASE**

**4. 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**

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

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

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

**4. Calculate the FinalIA (average of best two test marks) and update the**

**corresponding table for all students.**

**5. Categorize students based on the following criterion:**

**If FinalIA = 17 to 20 then CAT = ‘Outstanding’**

**If FinalIA = 12 to 16 then CAT = ‘Average’**

**If FinalIA< 12 then CAT = ‘Weak’**

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

**Solution:**

**P.K**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USN** | SNAME | ADDRESS | PHONE | GENDER |

**P.K**

|  |  |  |
| --- | --- | --- |
| **SSID** | SEM | SEC |

**F.K F.K**

|  |  |
| --- | --- |
| USN | SSID |

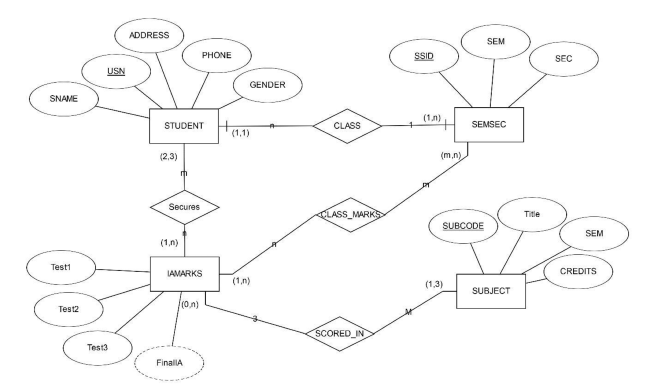
**P.K**

|  |  |  |  |
| --- | --- | --- | --- |
| **SCODE** | TITLE | SEM | CREDITS |

**F.K F.K F.K**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| USN | SCODE | SSID | TEST1 | TEST2 | TEST3 | FINALIA |

**E-R DIAGRAM**

****

**CREATION OF TABLES**

1. **COLLEGE TABLE**

|  |
| --- |
| SQL> create table student  2 (usn varchar(12) primary key,  3 sname varchar(10),  4 address varchar(10),phone number,gender number);  Table created. |

1. SEMSEC

|  |
| --- |
| SQL> create table semsec  2 (ssid number primary key,  3 sem number,  4 sec varchar(5)  5 );  Table created. |

1. Class

|  |
| --- |
| SQL> create table class  2 (usn varchar(12) references student(usn) on delete cascade,  3 ssid number references semsec(ssid) on delete cascade);  Table created. |

1. Subject

|  |
| --- |
| SQL> create table subject  2 (scode number primary key,  3 title varchar(10),sem number,  4 credits number);  Table created. |

1. IA Marks

|  |
| --- |
| SQL> create table iamarks  2 (usn varchar(12) references student(usn) on delete cascade,  3 scode varchar(10) references subject(scode) on delete cascade,  4 ssid number references semsec(ssid) on delete cascade,  5 test1 number,test2 number,test3 number,finalia number);  Table created. |

**Descriptons of tables**

|  |
| --- |
| **SQL> desc student;**  **Name Null? Type**  **----------------------------------------- -------- ----------------------------**  **USN NOT NULL VARCHAR2(12)**  **SNAME VARCHAR2(10)**  **ADDRESS VARCHAR2(10)**  **PHONE NUMBER**  **GENDER NUMBER** |
| **SQL> desc semsec;**  **Name Null? Type**  **----------------------------------------- -------- ----------------------------**  **SSID NOT NULL NUMBER**  **SEM NUMBER**  **SEC VARCHAR2(5)** |
| **SQL> desc class;**  **Name Null? Type**  **----------------------------------------- -------- ----------------------------**  **USN VARCHAR2(12)**  **SSID NUMBER** |
| **SQL> desc subject;**  **Name Null? Type**  **----------------------------------------- -------- ----------------------------**  **SCODE NOT NULL NUMBER**  **TITLE VARCHAR2(10)**  **SEM NUMBER**  **CREDITS NUMBER** |
| **SQL> desc iamarks;**  **Name Null? Type**  **----------------------------------------- -------- ----------------------**  **USN VARCHAR2(12)**  **SCODE NUMBER**  **SSID NUMBER**  **TEST1 NUMBER**  **TEST2 NUMBER**  **TEST3 NUMBER**  **FINALIA NUMBER** |

**III INSERTION**

1. **Student**

insert into student values('&usn','&name','&address',&phone,'&gender');

|  |
| --- |
| SQL> select \* from student;  USN SNAME ADDRESS PHONE GENDER  ------------ ---------- ---------- ---------- ------  3vc15is001 priya bly 4786476478 female  3vc15is002 santosh hospet 4656472443 male  3vc15is003 kiran bly 4736576 male  3vc15is004 sam up 3625476786 male  3vc15is005 kim blore 787846845 female  3vc15is006 sinchana bly 7673453782 female  3vc15is007 shiv blor 35246574 male  3vc16is001 swetha bly 347856478 female  3vc16is002 ravi blore 342652678 male  3vc16is003 steven hospet 34786574 male  3vc16is004 shilpa blore 647836 female  3vc16is005 vijay bly 434786 male |

1. Semsec

|  |
| --- |
| SQL> select \* from semsec;  SSID SEM SEC  ---------- ---------- -----  500 3 a  501 3 b  502 3 c  503 4 c  504 4 b  505 4 a  6 rows selected. |

1. Class

|  |
| --- |
| SQL> select \* from class;  USN SSID  ------------ ----------  3vc15is001 500  3vc15is002 500  3vc15is003 503  3vc15is004 502  3vc15is005 503  3vc15is006 500  3vc15is007 502  3vc16is001 504  3vc16is002 505  3vc16is003 505  3vc16is004 504  3vc16is005 503  12 rows selected. |

1. Subject

|  |
| --- |
| SQL> select \* from subject;  SCODE TITLE SEM CREDITS  --------- - ---------- ---------- ----------  15is41 m4 4 5  15is42 SE 4 5  15is43 DAA 4 5  15is44 MC 4 5  15is61 usp 6 5  15is62 cn 6 5  15is63 ST 6 5  15is64 FS 6 5  8 rows selected. |

1. Iamarks

|  |
| --- |
| SQL> select \* from iamarks;  USN SCODE SSID TEST1 TEST2 TEST3 FINALIA  ------------ ---------- ---------- ---------- ---------- ---------- ----------  3vc15is001 15is41 500 10 17 18 18  3vc15is001 15is42 500 15 18 19 19  3vc15is001 15is43 500 10 10 11 11  3vc15is002 15is41 500 15 16 5 16  3vc15is002 15is42 500 19 15 17 18  3vc15is002 15is43 500 14 15 13 15  3vc15is003 15is41 503 16 18 17 17  3vc15is003 15is42 503 16 17 18 18  3vc15is004 15is41 502 11 18 17 18  3vc15is004 15is42 502 5 6 7 7  3vc15is004 15is43 502 17 19 18 19  3vc16is001 15is61 504 16 17 17 17  3vc16is001 15is62 504 14 13 1 14  3vc16is001 15is63 504 16 13 17 17  3vc16is002 15is63 505 16 13 14 15  3vc16is002 15is64 505 17 18 4 18  3vc16is003 15is61 505 3 5 6 6  3vc16is003 15is63 505 13 12 11 12  3vc16is004 15is61 504 15 11 14 15  19 rows selected. |

**VI QUERIES**

Q1) List all the student details studying in fourth semester ‘C’ section.

|  |
| --- |
| SQL> select s.\*  2 from student s,semsec ss,class c  3 where ss.ssid=c.ssid and s.usn=c.usn and  4 ss.sem=4 and ss.sec='c';  USN SNAME ADDRESS PHONE GENDER  ------------ ---------- ---------- ---------- ------  3vc15is004 sam up 3625476786 male  3vc15is007 shiv blor 35246574 male |

Q2) Compute the total number of male and female students in each semester and in each section.

|  |
| --- |
| SQL> select ss.sem,ss.sec,s.gender,count(s.gender) as COUNT  2 from student s,semsec ss,class c  3 where c.usn=s.usn and ss.ssid=c.ssid  4 group by ss.sem,ss.sec,s.gender  5 order by sem;  SEM SEC GENDER COUNT  ---------- ----- ------ ----------  4 a female 2  4 a male 1  4 c male 2  6 a male 2  6 b female 2  6 c female 1  6 c male 2  7 rows selected. |

Q3) Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.

|  |
| --- |
| SQL> create view TEST1\_15is01  2 as  3 select test1,scode  4 from iamarks where usn='3vc15is001';  View created.  SQL> select \* from TEST1\_15is01;  TEST1 SCODE  ---------- ----------  10 15is41  15 15is42  10 15is43 |

Q4) Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.

Before executing the procedure STUDENT\_AVG(NOTE:FINALIA column contents are null)

|  |
| --- |
| **SQL> select \* from iamarks;**  **USN SCODE SSID TEST1 TEST2 TEST3 FINALIA**  **------------ ---------- ---------- ---------- ---------- ---------- ----------**  **3vc15is001 15is41 500 10 17 18**  **3vc15is001 15is42 500 15 18 19**  **3vc15is001 15is43 500 10 10 11**  **3vc15is002 15is41 500 15 16 5**  **3vc15is002 15is42 500 19 15 17**  **3vc15is002 15is43 500 14 15 13**  **3vc15is003 15is41 503 16 18 17**  **3vc15is003 15is42 503 16 17 18**  **3vc15is004 15is41 502 11 18 17**  **3vc15is004 15is42 502 5 6 7**  **3vc15is004 15is43 502 17 19 18**  **3vc16is001 15is61 504 16 17 17**  **3vc16is001 15is62 504 14 13 1**  **3vc16is001 15is63 504 16 13 17**  **3vc16is002 15is63 505 16 13 14**  **3vc16is002 15is64 505 17 18 4**  **3vc16is003 15is61 505 3 5 6**  **3vc16is003 15is63 505 13 12 11**  **3vc16is004 15is61 504 15 11 14**  **19 rows selected.** |

CREATION OF STORED PROCEDURE STUDENT\_AVG

|  |
| --- |
| **SQL> CREATE OR REPLACE PROCEDURE STUDENT\_AVG**  **2 IS**  **3 CURSOR C\_IAMARKS IS**  **4**  **5 SELECT GREATEST(TEST1,TEST2) AS A,GREATEST(TEST1,TEST3) AS B,**  **6 GREATEST(TEST3,TEST2) AS C**  **7 FROM IAMARKS**  **8 WHERE FINALIA IS NULL**  **9 FOR UPDATE;**  **10 C\_A NUMBER;**  **11 C\_B NUMBER;**  **12 C\_C NUMBER;**  **13 C\_SM NUMBER;**  **14 C\_AV NUMBER;**  **15**  **16 BEGIN**  **17 OPEN C\_IAMARKS;**  **18**  **19 LOOP**  **20 FETCH C\_IAMARKS INTO C\_A,C\_B,C\_C;**  **21**  **22 EXIT WHEN C\_IAMARKS%NOTFOUND;**  **23**  **24 DBMS\_OUTPUT.PUT\_LINE(C\_A||' '||C\_B||' '||C\_C);**  **25 IF(C\_A!=C\_B) THEN**  **26 C\_SM:=C\_A+C\_B;**  **27 ELSE**  **28 C\_SM:=C\_A+C\_C;**  **29 END IF;**  **30**  **31 C\_AV:=C\_SM/2;**  **32**  **33 DBMS\_OUTPUT.PUT\_LINE('SUM='||C\_SM);**  **34 DBMS\_OUTPUT.PUT\_LINE('AVERAGE='||C\_AV);**  **35**  **36 UPDATE IAMARKS**  **37 SET FINALIA=C\_AV**  **38**  **39 WHERE CURRENT OF C\_IAMARKS;**  **40 END LOOP;**  **41**  **42 CLOSE C\_IAMARKS;**  **43 END STUDENT\_AVG;**  **44 /**  **Procedure created.** |

|  |
| --- |
| SQL> BEGIN  2 STUDENT\_AVG;  3 END;  4 /  PL/SQL procedure successfully completed. |

|  |
| --- |
| SQL> SELECT \* FROM IAMARKS;  USN SCODE SSID TEST1 TEST2 TEST3 FINALIA  ------------ ---------- ---------- ---------- ---------- ---------- ----------  3vc15is001 15is41 500 10 17 18 17.5  3vc15is001 15is42 500 15 18 19 18.5  3vc15is001 15is43 500 10 10 11 10.5  3vc15is002 15is41 500 15 16 5 15.5  3vc15is002 15is42 500 19 15 17 18  3vc15is002 15is43 500 14 15 13 14.5  3vc15is003 15is41 503 16 18 17 17.5  3vc15is003 15is42 503 16 17 18 17.5  3vc15is004 15is41 502 11 18 17 17.5  3vc15is004 15is42 502 5 6 7 6.5  3vc15is004 15is43 502 17 19 18 18.5  3vc16is001 15is61 504 16 17 17 17  3vc16is001 15is62 504 14 13 1 13.5  3vc16is001 15is63 504 16 13 17 16.5  3vc16is002 15is63 505 16 13 14 15  3vc16is002 15is64 505 17 18 4 17.5  3vc16is003 15is61 505 3 5 6 5.5  3vc16is003 15is63 505 13 12 11 12.5  3vc16is004 15is61 504 15 11 14 14.5  19 rows selected. |

5. Categorize students based on the following criterion:

If FinalIA = 17 to 20 then CAT = ‘Outstanding’

If FinalIA = 12 to 16 then CAT = ‘Average’

If FinalIA< 12 then CAT = ‘Weak’

Give these details only for 8

th

semester A, B, and C section students.

|  |
| --- |
| SQL> select s.usn,s.sname,s.address,s.phone,s.gender,  2 (case  3 when ia.finalia between 17 and 20 then 'outstanding'  4 when ia.finalia between 12 and 16 then 'average'  5 else 'weak'  6 end) as cat  7 from student s, semsec ss, iamarks ia, subject sub  8 where s.usn = ia.usn and  9 ss.ssid = ia.ssid and  10 sub.scode = ia.scode and  11 sub.sem = 6;  USN SNAME ADDRESS PHONE GENDER CAT  ------------ ---------- ---------- ---------- ------ -----------  3vc16is001 swetha bly 347856478 female outstanding  3vc16is001 swetha bly 347856478 female average  3vc16is001 swetha bly 347856478 female weak  3vc16is002 ravi blore 342652678 male average  3vc16is002 ravi blore 342652678 male outstanding  3vc16is003 steven hospet 34786574 male weak  3vc16is003 steven hospet 34786574 male average  3vc16is004 shilpa blore 647836 f emale average  8 rows selected. |

**LAB PROGRAM -5 COMPANY DATABASE**

**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.**

**2. Show the resulting salaries if every employee working on the ‘IoT’ project is given a**

**10 percent raise.**

**3. Find the sum of the salaries of all employees of the ‘Accounts’ department, as well**

**as the maximum salary, the minimum salary, and the average salary in this**

**department**

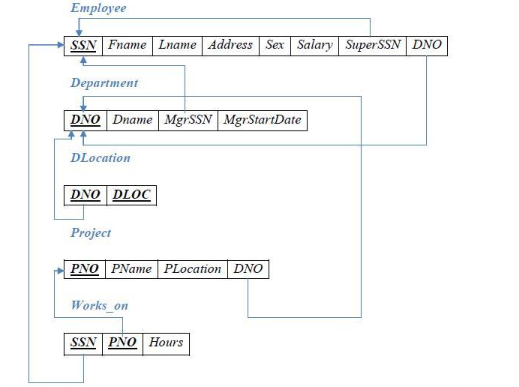
**4. Retrieve the name of each employee who works on all the projects controlled by**

**department number 5 (use NOT EXISTS operator). For each department that has**

**more than five employees, retrieve the department number and the number of its**

**employees who are making more than Rs. 6,00,000.**

**SCHEMA-DIAGRAM**



**E-R DIAGRAM**

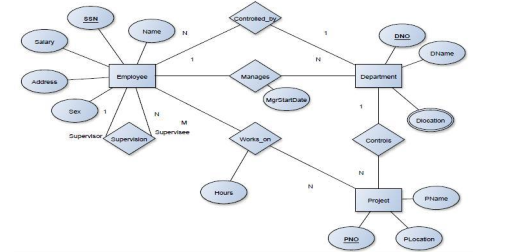


TABLE CREATION

**TABLES CREATION**

1. DEPARTMNET

|  |
| --- |
| SQL> create table department  2 (dno number primary key,  3 dname varchar(20)  4 );  Table created. |

1. EMPLOYEE

|  |
| --- |
| SQL> create table employee  2 (ssn number primary key,  3 name varchar(10),  4 address varchar(20),  5 gender varchar(10),  6 salary number,  7 dno number references department(dno) on delete cascade);  Table created. |

Alter command to add foreign keys in employee and department tables

|  |
| --- |
| SQL> alter table department add(mgrssn number references employee(ssn)  2 on delete cascade,  3 mgr\_start\_date date);  Table altered.  SQL> alter table employee add  2 (superssn number references employee(ssn) on delete cascade,  3 dno number references department(dno) on delete cascade);  Table altered. |

1. DLOCATION

|  |
| --- |
| SQL> create table dlocation  2 (dno number references department(dno) on delete cascade,  3 dlocation varchar(20) );  Table created. |

1. PROJECT

|  |
| --- |
| SQL> create table project  2 ( pno number primary key,  3 pname varchar(20),  4 plocation varchar(20),  5 dno number references department(dno) on delete cascade);  Table created. |

1. Works\_on

|  |
| --- |
| SQL>create table works\_on  2(ssn number references employee(ssn) on delete cascade,  3 pno number references project(pno) on delete cascade,  4.hrs number);  Table created |

Description of tables

|  |
| --- |
| SQL> desc employee;  Name Null? Type  ----------------------------------------- -------- -----------------------  SSN NOT NULL NUMBER  NAME VARCHAR2(10)  ADDRESS VARCHAR2(20)  GENDER VARCHAR2(10)  SALARY NUMBER |
| SQL> desc department;  Name Null? Type  ----------------------------------------- -------- ----------------------  DNO NOT NULL NUMBER  DNAME VARCHAR2(20)  MGRSSN NUMBER  MSR\_START\_DATE DATE |
| SQL> desc dlocation;  Name Null? Type  ---------------------------------------  DNO  NUMBER  DLOCATION VARCHAR2(20) |
| SQL> desc project;  Name Null? Type  ---------------------------------------------------------------  PNO NOT NULL NUMBER  PNAME VARCHAR2(20)  PLOCATION VARCHAR2(20)  DNO NUMBER |
| SQL> desc works\_on;  Name Null? Type  ----------------------------------------  SSN NUMBER  PNO NUMBER  HRS NUMBER |

INSERT COMMANDS

|  |
| --- |
| INSERT INTO DEPARTMENT VALUES(&DNO,'&DNAME')  Note: use alter table commands to add mgrssn and mgrstart\_date columns after insering values in employee table |
| INSERT INTO EMPLOYEE  VALUES('&SSN','&NAME','&ADDRESS','&GENDER',&SALARY)  Note: use alter table commands to add super\_ssn and dno after entering the vales of ssn and dno |
| INSERT INTO DLOCATION VALUES(&DNO,'&DLOC'); |
| INSERT INTO PROJECT VALUES(&PNO,'&PNAME','&PLOCATION',’&DNO’); |
| INSERT INTO WORKS\_ON VALUES('&SSN',&PNO,&HRS); |

TABLES –DISPLAY

EMPLOYEE TABLE

|  |
| --- |
| SQL> select \* from employee;  SSN NAME ADDRESS GENDER SALARY SUPERSSN DNO  ---------- ---------- -------------------- ---------- -------- -------------- ---------  111 sam blore male 54637  112 kim mysore female 65748  113 bob delhi male 54637  114 priya mysore female 65747  115 smith blore male 53666  116 john bly male 54637  117 wong bly male 543656  **Note** Use update command to fill the superssn and dno columns  SQL> select \* from employee;  SSN NAME ADDRESS GENDER SALARY SUPERSSN DNO  --------- ---------- -------------------- ---------- --------- - ---------- ----------  111 sam blore male 54637 116 30  112 kim mysore female 65748 116 10  113 bob delhi male 675849 114 10  114 priya mysore female 65747 116 10  115 smith blore male 53666 114 20  116 john bly male 54637 117 10  117 wong bly male 543656 NULL 40  118 scott blore male 64738 113 10  8 rows selected. |
| SQL> select \* from department; DNO DNAME MGRSSN MSR\_START  ---------- -------------------- ---------- ---------  10 sales 113  20 HQ 115  30 admin 111  40 marketing 117  SQL> update department set msr\_start\_date='12-mar-2000' where dno=20;  1 row updated.  SQL> update department set msr\_start\_date='1-may-2003' where dno=10;  1 row updated.  SQL> update department set msr\_start\_date='11-jan-2003' where dno=30;  1 row updated.  SQL> update department set msr\_start\_date='1-jan-2003' where dno=40;  1 row updated.  SQL> select \* from department;  DNO DNAME MGRSSN MSR\_START  ---------- -------------------- ---------- ---------  10 accounts 113 01-MAY-03  20 HQ 115 12-MAR-00  30 admin 111 11-JAN-03  40 marketing 117 01-JAN-03 |
| SQL> select \* from dlocation;  DNO DLOCATION  ---------- --------------------  10 bagalore  20 mumbai  20 bangalore  30 bangalore  40 bangalore |
| SQL> select \* from project;  PNO PNAME PLOCATION DNO  ---------- -------------------- -------------------- ----------  1234 productA bangalore 10  222 productB mumbai 20  223 productC bangalore 10  224 iot mumbai 30  225 iot bangalore 30 |
| SQL> select \* from works\_on;  SSN PNO HRS  ---------- ---------- ----------  111 1234 10  111 224 7  112 225 8  112 1234 9  113 225 7  114 222 6  114 1234 8  115 224 8 |

**QUERIES**

Q1) 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.

|  |
| --- |
| SQL> 1 (select distinct pno  2 from project p, department d,  3 employee e where p.dno=d.dno and  4 ssn=mgrssn and  5 name='sam')  6 union  7 (select distinct p.pno  8 from project p, works\_on w,  9 employee e where p.pno=w.pno and  10 w.ssn=e.ssn and  11 name='sam')  12 ;  PNO  ----------  224  225 |

Q2) 2. Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.

|  |
| --- |
| 1 select e.name, 1.1\*e.salary as incr\_sal  2 from employee e, works\_on w, project p  3 where e.ssn=w.ssn and  4 w.pno=p.pno and  5 p.pname='iot'  6 ;  NAME INCR\_SAL  ---------- ----------  sam 60100.7  kim 72322.8  bob 60100.7  smith 59032.6 |

Q3) 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.

|  |
| --- |
| SQL> select sum(salary), max(salary), min(salary),  2 avg(salary) from employee e, department d  3 where dname='accounts' and  4 d.dno=e.dno  5 ;  SUM(SALARY) MAX(SALARY) MIN(SALARY) AVG(SALARY)  ----------- ----------- - ---------- -----------  186132 65748 54637 62044 |

Q4) .Retrieve the name of each employee who works on all the projects controlled by

department number 5 (use NOT EXISTS operator).

|  |
| --- |
| 1 select name from  2 employee e  3 where not exists(  4 (select pno  5 from project  6 where  7 dno=5)  8 minus  9 (select pno  10 from works\_on w  11\* where e.ssn=w.ssn))  12 ;  NAME  ----------  sam  kim  bob  priya  smith  john  wong  7 rows selected. |

Q5) 5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.

|  |
| --- |
| SQL> select dno,count(ssn)  2 from employee  3 where salary>600000 and dno  4 in(select dno  5 from employee  6 group by dno  7 having count(ssn)>=5)  8 group by dno  9 ;  DNO COUNT(SSN)  ---------- ----------  10 1 |