INDEX

RELATIONAL DATABASE MANAGEMENT SYSTEM

STUDENTS DATABASE WITH CURSOR

TABLE CREATION:

Design a student database with the following tables:
Stu (regno,name,addr,city)

```
SQL> create table stu(regno char (10)primary key, name varchar2 (25)not null, addr varchar2 (20),city
varchar2 (10));
```

Table created.

Cou (courseid,cname,staffid)

```
SQL> create table cou(courseid varchar2 (10),cname varchar2 (25),staffid number);
```

Table created.

Staff (deptid, deptname,staffid,staffname)

```
SQL> create table staff(deptid number, deptname varchar2 (30),staffid number,staffname varchar2 (30));
```

Table created.

Mark (regno,courseid,sub1,sub2,sub3,sub4,sub5)

```
SQL> create table mark(regno char (10), courseid varchar2 (30), sub1 number, sub2 number, sub3 number,
sub4 number, sub5 number);
```

Table created.

CONSTRAINTS ADDED TO TABLES:

```
SQL> alter table staff add constraint staff10_pk primary key (staffid,deptid);
```

Table altered.

```
SQL> alter table mark add constraint chk10_mks check (sub1>0 and sub1<100);
```

Table altered.

```
SQL> alter table mark add constraint chk20_mks check (sub2>0 and sub2<100);
```

Table altered.

```
SQL> alter table mark add constraint chk30_mks check (sub3>0 and sub3<100);
```

Table altered.

```
SQL> alter table mark add constraint chk40_mks check (sub4>0 and sub4<100);
```

Table altered.

```
SQL> alter table mark add constraint chk50_mks check (sub5>0 and sub5<100);
```

Table altered.

```
SQL> alter table mark add constraint fok_rno foreign key (regno) references stu(regno) on delete cascade;
```

Table altered.

```
SQL> alter table cou add constraint cot_course primary key (courseid);
```

Table altered.

SQL> alter table mark add constraint fok_cid foreign key (courseid)references cou(courseid);

Table altered.

TABLE DECRIPTIONS:

SQL> desc stu;

Name	Null?	Type
REGNO	NOT NULL	CHAR (10)
NAME	NOT NULL	VARCHAR2 (25)
ADDR		VARCHAR2 (20)
CITY		VARCHAR2 (10)

SQL> desc cou;

Name	Null?	Type
COURSEID	NOT NULL	VARCHAR2 (10)
CNAME		VARCHAR2 (25)
STAFFID		NUMBER

SQL> desc mark;

Name	Null?	Type
REGNO		CHAR (10)
COURSEID		VARCHAR2 (30)
SUB1		NUMBER
SUB2		NUMBER
SUB3		NUMBER
SUB4		NUMBER
SUB5		NUMBER

SQL> desc staff;

Name	Null?	Type
DEPTID	NOT NULL	NUMBER
DEPTNAME		VARCHAR2 (30)
STAFFID	NOT NULL	NUMBER
STAFFNAME		VARCHAR2 (30)

INSERTING VALUES IN STU TABLE:

SQL> insert into stu values ('&reg', '&name', '&addr', '&city');
Enter value for reg: 01
Enter value for name: anu
Enter value for addr: mtp road
Enter value for city: cbe
old 1: insert into stu values ('&reg', '&name', '&addr', '&city')
new 1: insert into stu values ('01', 'anu', 'mtp road', 'cbe')

1 row created.

SQL> /
Enter value for reg: 02
Enter value for name: bani
Enter value for addr: rs puram
Enter value for city: chennai

old 1: insert into stu values ('&reg', '&name', '&addr', '&city')
new 1: insert into stu values ('02', 'bani', 'rs puram', 'chennai')

1 row created.

SQL> /
Enter value for reg: 03
Enter value for name: ram
Enter value for addr: ganapathy
Enter value for city: trichy
old 1: insert into stu values ('&reg', '&name', '&addr', '&city')
new 1: insert into stu values ('03', 'ram', 'ganapathy', 'trichy')

1 row created.

INSERTING VALUES IN COU TABLE:

SQL> insert into cou values ('1011', 'java', 9100);

1 row created.

SQL> insert into cou values ('1022', 'graphics', 9200);

1 row created.

SQL> insert into cou values ('1033', 'php', 9300);

1 row created.

INSERTING VALUES IN STAFF TABLE:

SQL> insert into staff values (20, 'computer science', 9200, 'rajesh');

1 row created.

SQL> insert into staff values (20, 'computer science', 9300, 'arthi');

1 row created.

SQL> insert into staff values (15, 'chemistry', 9100, 'kala');

1 row created.

INSERTING VALUES IN MARK TABLE:

SQL> insert into mark values ('01', '1011', 76, 90, 56, 40, 89);

1 row created.

SQL> insert into mark values ('02', '1022', 85, 90, 95, 96, 87);

1 row created.

SQL> insert into mark values ('03', '1033', 78, 89, 90, 98, 96);

1 row created.

DISPLAYING TABLE CONTENTS:

SQL> select * from stu;

```
REGNO     NAME                 ADDR              CITY
---------- ------------------------ -------------------- ----------
01        anu                  mtp road          cbe
02        bani                 rs puram          chennai
03        ram                  ganapathy          trichy

SQL> select * from cou;

COURSEID   CNAME                    STAFFID
---------- ------------------------ ----------
1011       java                     9100
1022       graphics                   9200
1033       php                      9300

SQL> select * from staff;

   DEPTID DEPTNAME                       STAFFID STAFFNAME
---------- ------------------------------ ---------- ------------------------------
        20 computer science                 9200 rajesh
        20 computer science                 9300 arthi
        15 chemistry                      9100 kala

SQL> select * from mark;

REGNO     COURSEID       SUB1      SUB2      SUB3      SUB4      SUB5
---------- ---------- ---------- ---------- ---------- ---------- ----------
01        1011            76        90        56        40        89
02        1022            85        90        95        96        87
03         1033            78        89        90        98        96
```

DML QUERIES:

a. Display the details of all the staff in all courses with staff ID, Name, Dept Name, and group by course.

```
SQL> create or replace view stu_view1 as (select * from staff join cou using (staffid));
```

View created.

b. Delete a student from stu table.

Before Deletion:

```
SQL> delete from stu where regno=&regno;
Enter value for regno: 03
old   1: delete from stu where regno=&regno
new   1: delete from stu where regno=03
```

1 row deleted.

After Deletion:

```
SQL> select * from stu;

REGNO     NAME                 ADDR              CITY
---------- ------------------------ -------------------- ----------
01        anu                  mtp road          cbe
02        banu                  rs puram          chennai
```

c. Update a record in stu table

```
SQL> update stu set name = 'brintha' where regno='01';

1 row updated.
```

CREATE A CURSOE TO DISPLAY THE DETAILS OF STUDENTS WHO HAS PASSED WITH DISTINCTION.

```
SQL> set serveroutput on
SQL> declare cursor distinctions is select *  from stu where regno in(select regno from mark
where((sub1+sub2+sub3+sub4+sub5)/5)>75);
  2  begin for dr in distinctions loop
dbms_output.put_line(dr.regno||','||dr.name||','||dr.addr||','||dr.city);
  3  end loop;
  4  end;
  5  /

02       ,banu,rs puram,chennai

PL/SQL procedure successfully completed.
```

EMPLOYEE DATABASE WITH CURSOR

TABLE CREATION:

Design a student database with the following tables:
Emp (name,eno,deptno,addr,dob,sex,salary)

```
SQL> create table emp (name varchar2 (25) not null, eno number primary key, deptno number, addr
varchar2 (50), dob date, sex char (1), salary number (9,2));
```

Table created.

Dept (dname, dno, location)

```
SQL> create table dept (dname varchar2 (20), dno number primary key, location varchar2 (15));
```

Table created.

Project (pno, pname, dno, plocation)

```
SQL> create table project (pno varchar2 (10) primary key, pname varchar2 (20), dno number, plocation
varchar2 (15));
```

Table created.

Works (eno, pno, hours)

```
SQL> create table works (eno number, pno varchar2 (10), hours number);
```

Table created.

CONSTRAINTS ADDED TO TABLES:

```
SQL> alter table emp add constraint fk_dept foreign key (deptno) references dept (dno);
```

Table altered.

```
SQL> alter table project add constraint fk_pr_dept foreign key (dno) references dept (dno);
```

Table altered.

```
SQL> alter table works add constraint fk_wrk_emp foreign key (eno) references emp (eno);
```

Table altered.

```
SQL> alter table works add constraint fk_wrk_proj foreign key (pno) references project (pno);
```

Table altered.

TABLE DESCRIPTIONS:

```
SQL> desc emp;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                      NOT NULL VARCHAR2 (25)
 ENO                                       NOT NULL NUMBER
 DEPTNO                                             NUMBER
 ADDR                                               VARCHAR2 (50)
 DOB                                                DATE
 SEX                                                CHAR (1)
 SALARY                                             NUMBER (9,2)

SQL> desc dept;
```

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
DNAME                                              VARCHAR2 (20)
DNO                                       NOT NULL NUMBER
LOCATION                                           VARCHAR2 (15)

SQL> desc project;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
PNO                                       NOT NULL VARCHAR2 (10)
PNAME                                              VARCHAR2 (20)
DNO                                                NUMBER
PLOCATION                                          VARCHAR2 (15)

SQL> desc works;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
ENO                                                NUMBER
PNO                                                VARCHAR2 (10)
HOURS                                              NUMBER
```

INSERTING VALUES IN DEPT TABLE:

```
SQL> insert into dept values ('research', 5000, 'chennai');

1 row created.

SQL> insert into dept values ('research', 5010, 'bangalore');

1 row created.

SQL> insert into dept values ('admin', 5001, 'chennai');

1 row created.

SQL> insert into dept values ('auditor', 5100, 'coimbatore');

1 row created.
```

INSERTING VALUES IN EMP TABLE:

```
SQL> insert into emp values ('sathya', 1000, 5000 , 'saibaba colony, coimbatore', '15-sep-1974', 'f', 10000);

1 row created.

SQL> insert into emp values ('ramya', 1002, 5000 , 'cluddalore main road, cuddalore', '19-jan-1986', 'f',
12000);

1 row created.

SQL> insert into emp values ('suresh', 1001, 5010 , 'ramasamy nagar,coimbatore', '19-nov-1985', 'm', 10500);

1 row created.

SQL> insert into emp values ('rameshan', 1003, 5100 , 'nasr road, coimbatore', '15-sep-1989', 'm', 11000);

1 row created.

SQL> insert into emp values ('prakash', 1004, 5001, 'mgp road, madurai', '25-oct-1984', 'm', 15000);
```

1 row created.

INSERTING VALUES IN PROJECT TABLE:

```
SQL> insert into project values ('p1', 'algorithms', 5000, 'rajasthan');

1 row created.

SQL> insert into project values ('p2', 'trees', 5000, 'bhopal');

1 row created.

SQL> insert into project values ('p3', 'neuralnetworks', 5010, 'delhi');

1 row created.

SQL> insert into project values ('p4', 'networks', 5000, 'delhi');

1 row created.

SQL> insert into project values ('p5', 'embedded', 5010, 'rajasthan');

1 row created.
```

INSERTING VALUES IN WORKS TABLE:

```
SQL> insert into works (eno,pno,hours) values (1000,'p1',8);

1 row created.

SQL> insert into works (eno,pno,hours) values (1001,'p2',6);

1 row created.

SQL> insert into works (eno,pno,hours) values (1002,'p1',7);
1 row created.

SQL> insert into works (eno,pno,hours) values (1004,'p1',12);

1 row created.
```

DISPLAYING TABLE CONTENTS:

```
SQL> select * from dept;
DNAME                DNO LOCATION
-------------------- ---------- --------------
research              5000 chennai
research              5010 bangalore
admin                 5001 chennai
auditor               5100 coimbatore

SQL> select * from emp;
NAME                    ENO    DEPTNO
----------------------- ---------- ----------
ADDR                                DOB      S    SALARY
-------------------------------------------------- --------- - ----------
sathya                  1000      5000
saibaba colony, coimbatore                  15-SEP-74 f     10000
```

```
ramya                    1002      5000
cluddalore main road, cuddalore              19-JAN-86 f      12000

suresh                   1001      5010
ramasamy nagar,coimbatore                    19-NOV-85 m      10500

rameshan                 1003      5100
nasr road, coimbatore                        15-SEP-89 m      11000

prakash                  1004      5001
mgp road, madurai                            25-OCT-84 m      15000

SQL> select * from project;
PNO       PNAME                DNO PLOCATION
---------- -------------------- ---------- ---------------
p1        algorithms             5000 rajasthan
p2        trees                5000 bhopal
p3        neuralnetworks         5010 delhi
p4        networks             5000 delhi
p5        embedded               5010 rajasthan

SQL> select * from works;
     ENO PNO          HOURS
---------- ---------- ----------
    1000 p1              8
    1001 p2              6
    1002 p1              7
    1004 p1             12
```

DML QUERIES:

a. Retrieve the name of all the employee who work for 'research' department.

```
SQL> select eno,name from emp where deptno in(select dno from dept where dname='research');

     ENO NAME
---------- -------------------------
    1000 sathya
    1002 ramya
    1001 suresh
```

b. Retrieve the total no of employee in dept 'research' and dept 'admin'.

```
SQL> select count(eno) as tot_emp_in_rsrch_admin from emp where deptno in (select dno from dept where
dname='research' or dname='admin');

TOT_EMP_IN_RSRCH_ADMIN
----------------------
                     4
```

c. Create a view to count the no of distinct salary for each department.

```
SQL> create or replace view sal_view as (select distinct count(salary) distinct_sal_nos,deptno from emp
group by deptno);
```

View created.

```
SQL> select * from sal_view;

DISTINCT_SAL_NOS    DEPTNO
---------------- ----------
```

```
2	5000
1	5001
1	5010
1	5100
```

CREATE A CURSOR TO DISPLAY THE DETAILS OF ALL EMPLOYEES WHO WORK MORE THAN 8 HRS FOR PROJECT 'P1'.

```
SQL> set serveroutput on
SQL> declare emp_row emp%rowtype;
  2  cursor emphr_cur is select * from emp where eno in (select eno from works where pno='p1' and
hours>8);
  3  begin for cr in emphr_cur
  4  loop
  5  dbms_output.put_line(cr.name||','||cr.eno);
  6  end loop;
  7  end;
  8  /

prakash,1004

PL/SQL procedure successfully completed.
```

BOOK DATABASE WITH CURSOR

TABLE CREATION:

Design a book database with the following tables:
Publisher10 (pubid, name, city)

```
SQL> create table publisher10 (pubid number primary key, name varchar2 (30) not null, city varchar (20) not null);
```

Table created.

Book10 (id, title, publid, year, price)

```
SQL> create table book10 (id number primary key, title varchar (40) not null, pubid number not null, year date not null, price number (9,2));
```

Table created.

CONSTRAINTS ADDED TO TABLES:

```
SQL> alter table book10 add constraint pub_fk foreign key (pubid) references publisher10 (pubid);
```

Table altered.

TABLE DESCRIPTIONS:

```
SQL> desc publisher10;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PUBID                                     NOT NULL NUMBER
 NAME                                      NOT NULL VARCHAR2 (30)
 CITY                                      NOT NULL VARCHAR2 (20)

SQL> desc book10;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 TITLE                                     NOT NULL VARCHAR2 (40)
 PUBID                                     NOT NULL NUMBER
 YEAR                                      NOT NULL DATE
 PRICE                                              NUMBER (9,2)
```

INSERTING VALUES INTO PUBLISHER10 TABLE:

```
SQL> insert into publisher10 values (100, 'microsoft', 'washington');
```

1 row created.

```
SQL> insert into publisher10 values (101, 'sun microsystem', 'sanfrasncisco');
```

1 row created.

```
SQL> insert into publisher10 values (102, 'oracle', 'britan');
```

1 row created.

```
SQL> insert into publisher10 values (103, 'tatamcgraw hill', 'delhi');
```

1 row created.

```
SQL> insert into publisher10 values (104, 'samba', 'chennai');
```

1 row created.

INSERTING VALUES INTO BOOK10 TABLE:

```
SQL> insert into book10 values (500, 'java', 100, '20-jun-2000', 505);
```

1 row created.

```
SQL> insert into book10 values (501, 'programing in c', 101, '10-aug-2005', 2350);
```

1 row created.

```
SQL> insert into book10 values (504, 'dotnet', 104, '21-jun-2005', 35);
```

1 row created.

```
SQL> insert into book10 values (502, 'perl', 102, '15-dec-2002', 450);
```

1 row created.

```
SQL> insert into book10 values (506, 'cobol', 102, '12-sep-2007', 1525);
```

1 row created.

DISPLAYING TABLE CONTENTS:

```
SQL> select * from publisher10;

    PUBID NAME                      CITY
---------- ------------------------------ --------------------
       100 microsoft                  washington
       101 sun microsystem                sanfrasncisco
       102 oracle                     britan
       103 tatamcgraw hill                delhi
       104 samba                      chennai

SQL> select * from book10;

       ID TITLE                                PUBID YEAR
---------- ---------------------------------------- ---------- ---------
    PRICE
----------
       500 java                                   100 20-JUN-00
       505

       501 programing in c                        101 10-AUG-05
      2350

       504 dotnet                                 104 21-JUN-05
        35

       502 perl                                   102 15-DEC-02
       450

       506 cobol                                  102 12-SEP-07
      1525
```

DML QUERIES:

a. Get the titles and publisher names of all the books that are priced above 500.

```
SQL> select title, name from book10 join publisher10 using (pubid) where price > 500;

TITLE                          NAME
-------------------------------------- ------------------------------
java                           microsoft
programing in c                sun microsystem
cobol                          oracle
```

b. Get the title and price of all books published after the year 2006 priced above 400.

```
SQL> select title,price from book10 where price>400 and extract (year from year) > 2006;

TITLE                                  PRICE
-------------------------------------- ----------
cobol                                   1525
```

c. Get the no of books published each year.

```
SQL> select count (id), extract(year from year) as year_of_release from book10 group by extract(year from
year);

 COUNT(ID) YEAR_OF_RELEASE
---------- ---------------
         2            2005
         1            2007
         1            2000
         1            2002
```

d. Get the title and price of all books whose price is less than the average price of all books.

```
SQL> select title, price from book10 where price < (select avg(price) from book10);

TITLE                                  PRICE
-------------------------------------- ----------
java                                    505
dotnet                                     35
perl                                    450
```

WRITE A PL/SQL CURSOR TO DISPLAY THE DETAILS OF THE BOOKS WITH THE HIGHEST PRICE.

```
SQL> set serveroutput on;
SQL> declare cursor book_cur
  2  is
  3  select * from book10 where price>=(select max(price) from book10);
  4  begin
  5  for cr in book_cur
  6  loop dbms_output.put_line('book
id:'||cr.id||','||'title:'||cr.title||','||'pubid:'||cr.pubid||','||'year'||cr.year||','||'price'||cr.price);
  7  end loop;
  8  end;
  9  /

book id:501,title:programing in c,pubid:101,year10-AUG-05,price2350

PL/SQL procedure successfully completed.
```

EMPLOYEE DATABASE WITH TRIGGER

TABLE CREATION:

Design an employee database with the following tables:

Emp (name,eno,deptno,addr,dob,sex,salary)

```
SQL> create table emp (name varchar2 (25) not null,eno number primary key,deptno number,addr varchar2
(50),dob date,sex char (1),salary number (9,2));
```

Table created.

Dept (dname,do,location)

```
SQL> create table dept(dname varchar2 (20),dno number primary key,location varchar2 (15));
```

Table created.

Project (pno,pname,dno,plocation)

```
SQL> create table project(pno varchar2 (10) primary key,pname varchar2 (20),dno number,plocation
varchar2 (15));
```

Table created.

Works (eno,pno,hours)

```
SQL> create table works (eno number,pno varchar2 (10),hours number);
```

Table created.

CONSTRAINTS ADDED TO TABLES:

```
SQL> alter table emp add constraint fk_dept foreign key (deptno) references dept(dno);
```

Table altered.

```
SQL> alter table project add constraint fk_pr_dept foreign key (dno) references dept(dno);
```

Table altered.

```
SQL> alter table works add constraint fk_wrk_emp foreign key (eno) references emp(eno);
```

Table altered.

```
SQL> alter table works add constraint fk_wrk_proj foreign key (pno) references project(pno);
```

Table altered.

TABLE DESCRIPTIONS:

```
SQL> desc emp;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                      NOT NULL VARCHAR2 (25)
 ENO                                       NOT NULL NUMBER
 DEPTNO                                             NUMBER
 ADDR                                               VARCHAR2 (50)
 DOB                                                DATE
 SEX                                                CHAR (1)
 SALARY                                             NUMBER (9,2)

SQL> desc dept;
```

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
DNAME                                              VARCHAR2 (20)
DNO                                       NOT NULL NUMBER
LOCATION                                           VARCHAR2 (15)

SQL> desc project;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
PNO                                       NOT NULL VARCHAR2 (10)
PNAME                                              VARCHAR2 (20)
DNO                                                NUMBER
PLOCATION                                          VARCHAR2 (15)

SQL> desc works;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
ENO                                                NUMBER
PNO                                                VARCHAR2 (10)
HOURS                                              NUMBER
```

INSERTING VALUES INTO DEPT TABLE:

```
SQL> insert into dept values ('research', 5000, 'chennai');

1 row created.

SQL> insert into dept values ('research', 5010, 'bangalore');

1 row created.

SQL> insert into dept values ('admin', 5001, 'chennai');

1 row created.

SQL> insert into dept values ('auditor', 5100, 'coimbatore');

1 row created.
```

INSERTING VALUES INTO EMP TABLE:

```
SQL> insert into emp values ('sathya',1000,5000,'saibaba colony, coimbatore', '15-sep-1974', 'f', 10000);

1 row created.

SQL> insert into emp values ('ramya',1002,5000,'ramasamy colony, cuddalore', '19-jan-1985', 'f', 12000);
1 row created.

SQL> insert into emp values ('suresh',1001,5010,'kalaam nagar, coimbatore', '04-nov-1988', 'm', 10500);

1 row created.

SQL> insert into emp values ('rameshan',1003,5100,'nsr road, coimbatore', '30-jul-1956', 'm', 10800);

1 row created.

SQL> insert into emp values ('sujith',1004,5001,'ganthi nagar, madhurai', '23-mar-1977', 'm', 9500);

1 row created.
```

INSERTING VALUES IN PROJECT TABLE:

```
SQL> insert into project values ('p1','algorithms',5000,'rajasthan');
```

1 row created.

```
SQL> insert into project values ('p2','trees',5000,'bhopal');
```

1 row created.

```
SQL> insert into project values ('p3','neuralnetworks',5010,'delhi');
```

1 row created.

```
SQL> insert into project values ('p4','networks',5000,'delhi');
```

1 row created.

```
SQL> insert into project values ('p5','embedded',5010,'rajasthan');
```

1 row created.

INSERTING VALUES IN WORKS TABLE:

```
SQL> insert into works (eno,pno,hours) values (1000,'p1',8);
```

1 row created.

```
SQL> insert into works (eno,pno,hours) values (1001,'p2',6);
```

1 row created.

```
SQL> insert into works (eno,pno,hours) values (1002,'p1',7);
```

1 row created.

```
SQL> insert into works (eno,pno,hours) values (1004,'p1',12);
```

1 row created.

DISPLAYING TABLE CONTENTS:

```
SQL> select * from dept;

DNAME                DNO LOCATION
-------------------- ---------- --------------
research             5000 chennai
research             5010 bangalore
admin                5001 chennai
auditor              5100 coimbatore

SQL> select * from emp;

NAME                      ENO    DEPTNO
------------------------- ---------- ----------
ADDR                                 DOB      S    SALARY
---------------------------------------------------- --------- - ----------
sathya                    1000      5000
saibaba colony, coimbatore                   15-SEP-74 f     10000
```

```
ramya                1002      5000
ramasamy colony, cuddalore                    19-JAN-85 f      12000

suresh               1001      5010
kalaam nagar, coimbatore                      04-NOV-88 m      10500

rameshan             1003      5100
nsr road, coimbatore                          30-JUL-56 m      10800

sujith               1004      5001
ganthi nagar, madhurai                        23-MAR-77 m      9500

SQL> select * from project;

PNO       PNAME                DNO PLOCATION
---------- -------------------- ---------- ---------------
p1        algorithms             5000 rajasthan
p2        trees                  5000 bhopal
p3        neuralnetworks         5010 delhi
p4        networks               5000 delhi
p5        embedded               5010 rajasthan

SQL> select * from works;

      ENO PNO           HOURS
---------- ---------- ----------
     1000 p1                 8
     1001 p2                 6
     1002 p1                 7
     1004 p1                12
```

WRITE A TRIGGER THAT GETS ACTIVATED WHENEVER A RECORD IS DELETED FROM EMP TABLE ENFORCING CASCADING DELETION.

```
SQL> create or replace trigger emdel_trig
  2  before delete on emp for each row
  3  begin
  4  dbms_output.put_line('the employee and his corresponding works are also deleted');
  5  delete from works where eno=:old.eno;
  6  end;
  7  /
```

Trigger created.

BEFORE TRIGGER EXECUTION:

```
SQL> select * from emp;

NAME                      ENO     DEPTNO
------------------------- ---------- ----------
ADDR                                 DOB      S    SALARY
-------------------------------------------------- --------- - ----------
sathya                1000      5000
saibaba colony,coimbatore                     15-SEP-74 f      10000

ramya                 1002      5000
ramasamy colony,cuddalore                     19-JAN-85 f      12000
```

suresh 1001 5010
kalaam nagar,coimbatore 04-NOV-88 m 10500

rameshan 1003 5100
nsr road,coimbatore 30-JUL-56 m 10800

sujith 1004 5001
ganthi nagar,madhurai 23-MAR-77 m 9500

EXECUTION:

```
SQL> begin
  2  delete from emp where eno=&employee_number;
  3  end;
  4  /

Enter value for employee_number: 1001
old   2: delete from emp where eno=&employee_number;
new   2: delete from emp where eno=1001;
the employee and his corresponding works are also deleted

PL/SQL procedure successfully completed.
```

AFTER TRIGGER EXECUTION:

```
SQL> select * from emp;

NAME                      ENO     DEPTNO
------------------------- ---------- ----------
ADDR                                DOB       S     SALARY
-------------------------------------------------- --------- - ----------
sathya                    1000       5000
saibaba colony,coimbatore           15-SEP-74 f      10000

ramya                     1002       5000
ramasamy colony,cuddalore           19-JAN-85 f      12000

rameshan                  1003       5100
nsr road,coimbatore                 30-JUL-56 m      10800

sujith                    1004       5001
ganthi nagar,madhurai               23-MAR-77 m      9500
```

BEFORE TRIGGER EXECUTION:

```
SQL> select * from works;

      ENO PNO           HOURS
---------- ---------- ----------
     1000 p1               8
     1002 p1               7
     1004 p1              12
```

EXECUTION:

```
SQL> begin delete from emp where eno=&employee_number;
  2  end;
  3  /

Enter value for employee_number: 1002
old   1: begin delete from emp where eno=&employee_number;
new   1: begin delete from emp where eno=1002;
```

the employee and his corresponding works are also deleted

PL/SQL procedure successfully completed.

AFTER TRIGGER EXECUTION:

```
SQL> select * from works;

      ENO PNO           HOURS
---------- ---------- ----------
     1000 p1               8
     1004 p1              12


SQL> select * from emp;

NAME                         ENO     DEPTNO
-------------------------- ---------- ----------
ADDR                                    DOB      S     SALARY
-------------------------------------------------- --------- - ----------
sathya                      1000       5000
saibaba colony,coimbatore                      15-SEP-74 f      10000

rameshan                    1003       5100
nsr road,coimbatore                         30-JUL-56 m      10800

sujith                      1004       5001
ganthi nagar,madhurai                       23-MAR-77 m       9500
```

HOTEL DATABASE WITH TRIGGER

TABLE CREATION:

Design a hotel database with the following tables:

hotel10 (hotelno,hotelname, city)

```
SQL> create table hotel10 (hotelno char (2) primary key, hotelname varchar2 (30), city varchar2 (25));
```

Table created.

room10 (roomno,type,price)

```
SQL> create table room10 (roomno number primary key, hotelno char (2),type varchar2 (15), price number (10,2));
```

Table created.

guest10 (guestno, guestname)

```
SQL> create table guest10 (guestno number primary key, guestname varchar2 (25));
```

Table created.

booking10 (hotelno, guestno, datefrom,dateto,roomno)

```
SQL> create table booking10 (hotelno char (2), guestno number, datefrom date, dateto date, roomno number);
```

Table created.

CONSTRAINTS ADDED TO TABLES:

```
SQL> alter table room10 add constraint hotel_fk foreign key (hotelno) references hotel10 (hotelno);
```

Table altered.

```
SQL> alter table booking10 add constraint fk_gt foreign key (guestno) references guest10 (guestno);
```

Table altered.

```
SQL> alter table booking10 add constraint htl_fk foreign key (hotelno) references hotel10 (hotelno);
```

Table altered.

```
SQL> alter table booking10 add constraint fk_room foreign key (roomno) references room10 (roomno);
```

Table altered.

TABLE DSCRIPTIONS:

```
SQL> desc hotel10;
 Name                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 HOTELNO                   NOT NULL CHAR (2)
 HOTELNAME                          VARCHAR2 (30)
 CITY                               VARCHAR2 (25)

SQL> desc room10;
 Name                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROOMNO                    NOT NULL NUMBER
```

```
HOTELNO                                   CHAR (2)
TYPE                                      VARCHAR2 (15)
PRICE                                     NUMBER (10,2)

SQL> desc guest10;
 Name                                     Null?    Type
 ----------------------------------------- -------- ----------------------------
 GUESTNO                                  NOT NULL NUMBER
 GUESTNAME                                         VARCHAR2 (25)

SQL> desc booking10;
 Name                                     Null?    Type
 ----------------------------------------- -------- ----------------------------
 HOTELNO                                           CHAR (2)
 GUESTNO                                           NUMBER
 DATEFROM                                          DATE
 DATETO                                            DATE
 ROOMNO                                            NUMBER
```

INSERTING VALUES IN HOTEL10 TABLE:

```
SQL> insert into hotel10 values ('h1','paradise inn','chennai');

1 row created.

SQL> insert into hotel10 values ('h2','taj hotel','chennai');

1 row created.

SQL> insert into hotel10 values ('h3','paradise inn','bangalore');

1 row created.

SQL> insert into hotel10 values ('h4','taj hotel','mumbai');

1 row created.

SQL> insert into hotel10 values ('h5','oberai','pune');

1 row created.
```

INSERTING VALUES IN ROOM10 TABLE:

```
SQL> insert into room10 values (100, 'h1','deluxe',5500);

1 row created.

SQL> insert into room10 values (101, 'h2','deluxe', 1450);

1 row created.

SQL> insert into room10 values (102, 'h3','deluxe', 2750);

1 row created.

SQL> insert into room10 values (104, 'h4','deluxe', 5500);

1 row created.
```

SQL> insert into room10 values (105, 'h5','deluxe', 2000);

1 row created.

SQL> insert into room10 values (106, 'h1','ac', 1500);

1 row created.

INSERTING VALUES IN GUEST10 TABLE:

SQL> insert into guest10 (guestno, guestname) values (1, 'xyz');

1 row created.

SQL> insert into guest10 (guestno, guestname) values (2, 'divi');

1 row created.

SQL> insert into guest10 (guestno, guestname) values (3, 'sugan');

1 row created.

SQL> insert into guest10 (guestno, guestname) values (4, 'kani');

1 row created.

SQL> insert into guest10 (guestno, guestname) values (5, 'viji');

1 row created.

SQL> insert into guest10 (guestno, guestname) values (6, 'nithi');

1 row created.

INSERTING VALUES IN BOOKING10 TABLE:

SQL> insert into booking10 values ('h1',2, '5-feb-2009', '15-feb-2009', 100);

1 row created.

SQL> insert into booking10 values ('h1',2, '1-jan-2009', '1-jan-2009', 106);

1 row created.

SQL> insert into booking10 values ('h2',1, '9-feb-2009', '16-feb-2009', 101);

1 row created.

SQL> insert into booking10 values ('h3',3, '1-mar-2009', '5-mar-2009', 102);

1 row created.

SQL> insert into booking10 values ('h4',5, '7-mar-2009', '15-mar-2009', 104);

1 row created.

DISPLAYING TABLE CONTENTS:

```
SQL> select * from hotel10;

HO HOTELNAME                    CITY
-- ------------------------------ ------------------------
h1 paradise inn                 chennai
h2 taj hotel                    chennai
h3 paradise inn                 bangalore
h4 taj hotel                    mumbai
h5 oberai                       pune

SQL> select * from room10;

   ROOMNO HO TYPE               PRICE
---------- -- --------------- ----------
      100 h1 deluxe              5500
      101 h2 deluxe              1450
      102 h3 deluxe              2750
      104 h4 deluxe              5500
      105 h5 deluxe              2000
      106 h1 ac                  1500
```

6 rows selected.

```
SQL> select * from guest10;

  GUESTNO GUESTNAME
---------- -------------------------
         1 xyz
         2 divi
         3 sugan
         4 kani
         5 viji
         6 nithi
```

6 rows selected.

```
SQL> select * from booking10;

HO    GUESTNO DATEFROM  DATETO       ROOMNO
-- ---------- --------- --------- ----------
h1         2 05-FEB-09 15-FEB-09       100
h1         2 01-JAN-09 01-JAN-09       106
h2         1 09-FEB-09 16-FEB-09       101
h3         3 01-MAR-09 05-MAR-09       102
h4         5 07-MAR-09 15-MAR-09       104
```

DML QUERIES:

a. List the guest staying in hotel 'paradise inn'.

```
SQL> select distinct (guestname)
  2  from guest10
  3  join booking10 using(guestno)
  4  join hotel10 using(hotelno)
  5  where guestno = guestno and
  6  hotel10.hotelname='paradise inn';

GUESTNAME
-------------------------
```

divi
sugan

b. List the no of rooms in each hotel.

```
SQL> select count(roomno) as no_of_rooms
  2  from room10
  3  group by hotelno;

NO_OF_ROOMS
-----------
         2
         1
         1
         1
         1
```

c. Display the details of the hotels in which the guest 'xyz' stayed

```
SQL> select * from hotel10
  2  join booking10 using (hotelno)
  3  join guest10 using(guestno)
  4  where guestname='xyz';

  GUESTNO HO HOTELNAME                    CITY                    DATEFROM
---------- -- ------------------------------ ------------------------- ---------
DATETO       ROOMNO GUESTNAME
--------- ---------- -------------------------
         1 h2 taj hotel                  chennai                 09-FEB-09
16-FEB-09        101 xyz
```

d. List the average price of every room in hotel no 'h1'.

```
SQL> select avg(price),type as room_type
  2  from room10
  3  where hotelno = 'h1'
  4  group by type;

AVG(PRICE) ROOM_TYPE
---------- ---------------
      5500 deluxe
      1500 ac
```

e. List the name of the cities where hotel 'taj' is located.

```
SQL> select hotelname,city from hotel10
  2  where hotelname like 'taj%';

HOTELNAME                      CITY
------------------------------ -------------------------
taj hotel                      chennai
taj hotel                      mumbai
```

<u>WRITE A PL/SQL TRIGGER TO CHECK WHETHER THE PRICE IS GREATER THAN ZERO WHENEVER RECORD IS INSERTED INTO THE TABLE.</u>

```
SQL> set serveroutput on
SQL> create or replace trigger price_trig5
  2  before insert on room10 for each row
  3  declare price_zero_exception exception;
  4  begin if(:new.price<=0) then
  5  dbms_output.put_line('The price of a room should be not be less than or equal
```

```
  6
  7  to zero');
  8  raise price_zero_exception;
  9  end if;
 10  end;
 11  /
```

Trigger created.

BEFORE TRIGGER EXECUTION:

```
SQL> select * from room10;

   ROOMNO HO TYPE              PRICE
---------- -- --------------- ----------
      100 h1 deluxe             5500
      101 h2 deluxe             1450
      102 h3 deluxe             2750
      104 h4 deluxe             5500
      105 h5 deluxe             2000
      106 h1 ac                 1500
```

6 rows selected.

EXECUTION:

```
SQL> begin
  2  commit;
  3  insert into room10 values (&room_no,&hotel_no,
  4  &room_type,&price);
  5  exception
  6  when others then
  7  rollback;
  8  end;
  9  /
Enter value for room_no: 120
Enter value for hotel_no: 'h4'
old   3: insert into room10 values (&room_no,&hotel_no,
new   3: insert into room10 values (120,'h4',
Enter value for room_type: 'E class'
Enter value for price: 1200
old   4: &room_type,&price);
new   4: 'E class',1200);
```

PL/SQL procedure successfully completed.

AFTER TRIGGER EXECUTION:

```
SQL> select * from room10;

   ROOMNO HO TYPE              PRICE
---------- -- --------------- ----------
      100 h1 deluxe             5500
      101 h2 deluxe             1450
      102 h3 deluxe             2750
      104 h4 deluxe             5500
      105 h5 deluxe             2000
      106 h1 ac                 1500
      120 h4 E class            1200
```

7 rows selected.

TRIGGER EXECUTION:

```
SQL> begin
  2  commit;
  3  insert into room10 values (&room_no,&hotel_no,
  4  &room_type,&price);
  5  exception
  6  when others then
  7  rollback;
  8  end;
  9  /
Enter value for room_no: 124
Enter value for hotel_no: 'h6'
old   3: insert into room10 values (&room_no,&hotel_no,
new   3: insert into room10 values (124,'h6',
Enter value for room_type: 'A Class'
Enter value for price: 0
old   4: &room_type,&price);
new   4: 'A Class',0);
The price of a room should be not be less than or equal

to zero

PL/SQL procedure successfully completed.
```

BOOK DATABASE WITH TRIGGER

TABLE CREATION:

Design a book database with the following tables:
Publisher1 (pubid,name,city)

```
SQL> create table publisher1 (pubid number primary key, name varchar2 (30) not null, city varchar2 (20) not null);
```

Table created.

Book10 (id,title,publid,year,price)

```
SQL> create table books10 (id number primary key, title varchar2 (40) not null, pubid number not null, year date not null, price num
ber (9,2));
```

Table created.

CONTRAINTS ADDED TO TABLES:

```
SQL> alter table books10 add constraint pub_frk foreign key (pubid) references publisher10 (pubid);
```

Table altered.

INSERTING VALUES INTO PUBLISHER1 TABLE:

```
SQL> insert into publisher1 values (100, 'microsoft', 'washington');
```

1 row created.

```
SQL> insert into publisher1 values (101, 'sun microsystem', 'sanfrasncisco');
```

1 row created.

```
SQL> insert into publisher1 values (102, 'oracle', 'britan');
```

1 row created.

```
SQL> insert into publisher1 values (103, 'tatacgraw hill', 'delhi');
```

1 row created.

```
SQL> insert into publisher1 values (104, 'samba', 'chennai');
```

1 row created.

INSERTING VALUES INTO BOOK10 TABLE:

```
SQL> insert into books10 values (500,'java',100,'20-jun-2000',505);
```

1 row created.

```
SQL> insert into books10 values (501,'programming in c',101,'10-aug-2005',2350);
```

1 row created.

```
SQL> insert into books10 values (504,'dotnet',104,'20-jan-2005',35);
```

1 row created.

```
SQL> insert into books10 values (502,'perl',102,'15-dec-2002',450);
```

1 row created.

```
SQL> insert into books10 values (506,'cobol',102,'12-sep-2007',1525);
```

1 row created.

DISPLAYING TABLE CONTENTS:

```
SQL> select * from publisher10;

    PUBID NAME                           CITY
---------- ------------------------------ --------------------
       100 microsoft                      washington
       101 sun microsystem                sanfrasncisco
       102 oracle                         britan
       103 tatamcgraw hill                delhi
       104 samba                          chennai

SQL> select * from book10;

       ID TITLE                                    PUBID YEAR
---------- ---------------------------------------- ---------- ---------
    PRICE
----------
       500 java                                    100 20-JUN-00
       505

       501 programing in c                         101 10-AUG-05
      2350

       504 dotnet                                  104 21-JUN-05
        35

       502 perl                                    102 15-DEC-02
       450

       506 cobol                                   102 12-SEP-07
      1525
```

WRITE A PL/SQL TRIGGER WHICH GETS ACTIVATED WHENEVER A NEW RECORD IS INSERTED INTO PUBLISHER TABLE. IT SHOULD CHANGE THE PUBLISHER NAME AND CITY TO UPPERCASE.

```
SQL> create or replace trigger pubins_tri
  2  before insert on publisher10 for each row
  3  declare
  4  pid publisher10.pubid%type:=(:new.pubid);
  5  pname publisher10.name%type:=upper (:new.name);
  6  pcity publisher10.city%type:=upper (:new.city);
  7  begin
  8  dbms_output.put_line(pid);
  9  dbms_output.put_line(pname);
 10  dbms_output.put_line(pcity);
 11  :new.pubid:=pid;
 12  :new.name:=pname;
 13  :new.city:=pcity;
 14  end;
```

15 /

Trigger created.

BEFORE TRIGGER EXECUTION:

```
SQL> select * from publisher1;

    PUBID NAME                           CITY
---------- ------------------------------ --------------------
       100 microsoft                      washington
       101 sun microsystem                    sanfrasncisco
       102 oracle                         britan
       103 tatacgraw hill                   delhi
       104 samba                          chennai
```

EXECUTION:

```
SQL> insert into publisher10 values ('500', 'ahamed', 'banagalore');
500
AHAMED
BANAGALORE
```

1 row created.

AFTER TRIGGER EXECUTION:

```
SQL> select * from publisher10;

    PUBID NAME                           CITY
---------- ------------------------------ --------------------
       100 microsoft                      washington
       101 sun microsystem                sanfrasncisco
       102 oracle                         britan
       103 tatamcgraw hill                delhi
       104 samba                          chennai
       500 AHAMED                           BANAGALORE
```

6 rows selected.

BOOK DATABASE WITH PROCEDURE

TABLE CREATION:

Design a book database with the following tables:
Publisher (pubid, name, city)

```
SQL> create table publisher (pubid number primary key, name varchar2 (30) not null, city varchar2 (20) not null);
```

Table created.

Book (id,title,publid,year,price)

```
SQL> create table book (id number primary key, title varchar2 (40) not null, pubid number not null, year date not null, price number (9,2));
```

Table created.

CONSTRAINTS ADDED TO TABLES:

```
SQL> alter table book add constraint pub_fk foreign key(pubid) references publisher (pubid);
```

Table altered.

TABLE DESCRIPTIONS:

```
SQL> desc publisher;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PUBID                                     NOT NULL NUMBER
 NAME                                      NOT NULL VARCHAR2 (30)
 CITY                                      NOT NULL VARCHAR2 (20)

SQL> desc book;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 TITLE                                     NOT NULL VARCHAR2 (40)
 PUBID                                     NOT NULL NUMBER
 YEAR                                      NOT NULL DATE
 PRICE                                              NUMBER (9,2)
```

INSERTING VALUES INTO PUBLISHER TABLE:

```
SQL> insert into publisher values (100,'microsoft','washington');
```

1 row created.

```
SQL> insert into publisher values (101, 'sun microsystem', 'sanfrasncisco');
```

1 row created.

```
SQL> insert into publisher values (102, 'oracle', 'britan');
```

1 row created.

```
SQL> insert into publisher values (103,'tatamcgraw hill','delhi');
```

1 row created.

```
SQL> insert into publisher values (104,'samba','chennai');
```

1 row created.

INSERTING VALUES INTO BOOK TABLE:

```
SQL> insert into book values (500,'java',100,'20-jan-2000',505);

1 row created.

SQL> insert into book values (501,'programming in c',101,'10-aug-2005',2350);

1 row created.

SQL> insert into book values (504,'dotnet',104,'20-jan-2005',35);

1 row created.

SQL> insert into book values (506,'cobol',102,'12-sep-2007',1525);

1 row created.

SQL> insert into book values (502,'perl',102,'15-dec-2002',450);

1 row created.
```

DISPLAYING TABLE CONTENTS:

```
SQL> select * from publisher;
    PUBID NAME                           CITY
---------- ------------------------------ --------------------
       100 microsoft                      washington
       101 sun microsystem                sanfrasncisco
       102 oracle                         britan
       103 tatamcgraw hill                delhi
       104 samba                          chennai
SQL> select * from book;

       ID TITLE                                PUBID YEAR
---------- ---------------------------------------- ---------- ---------
    PRICE
----------
       500 java                                   100 20-JAN-00
       505

       501 programming in c                       101 10-AUG-05
      2350

       504 dotnet                                 104 20-JAN-05
        35
       506 cobol                                  102 12-SEP-07
      1525

       502 perl                                   102 15-DEC-02
       450
```

WRITE A PL/SQL PROCEDURE TO INSERT THE RECORDS IN THE BOOKS TABLE:

```
SQL> create or replace procedure book_insert (bno in number, bname in varchar2,pubno
  2  in number,dt in date, cost in number) as
  3  begin
  4  insert into book values (bno, bname, pubno, dt, cost);
```

```
 5  end book_insert;
 6  /
```

Procedure created.

BEFORE PROCEDURE EXECUTION:

```
SQL> select * from book;
      ID TITLE                                  PUBID YEAR
---------- ---------------------------------------- ---------- ---------
    PRICE
----------
      500 java                                  100 20-JAN-00
      505

      501 programming in c                      101 10-AUG-05
     2350

      504 dotnet                                104 20-JAN-05
       35

      506 cobol                                 102 12-SEP-07
     1525

      502 perl                                  102 15-DEC-02
      450
```

PROCEDURE EXECUTION:

```
SQL> begin
  2  book_insert(&book_number,&book_name,&publisher_number,&date,&cost);
  3  end;
  4  /

Enter value for book_number: 510
Enter value for book_name: 'C SHARP'
Enter value for publisher_number: 102
Enter value for date: '12-JUN-07'
Enter value for cost: 3244
old   2: book_insert(&book_number,&book_name,&publisher_number,&dae,&cost);
new   2: book_insert(510,'C SHARP',102,'12-JUN-07',3244);
```

PL/SQL procedure successfully completed.

COMPANY DATABASE WITH PROCEDURE

TABLE CREATION:

Design a company database with the following tables:

Customer10 (custno,cname,city)

```
SQL> create table customner10 (custno number (10) primary key, cname varchar2 (25) not null, city varchar2 (10) not null);
```

Table created.

corder10 (orderno,orderdate,custno,ordamt)

```
SQL> create table corder10 (orderno number (10) primary key,orderdate date not null, custno number (10), ordamt number (8,2));
```

Table created.

orderitem10 (orderno,itemno,qty)

```
SQL> create table orderitem10 (orderno number (10) not null,itemno number (10) not null, qty number (10)not null);
```

Table created.

item10 (itemno,unitprice)

```
SQL> create table item10 (itemno number (10) primary key, unitprice number (8,2));
```

Table created.

shipment10 (orderno,warehouseno,shipdate)

```
SQL> create table shipment10 (orderno number (10), warehouseno char (2), shipdate date);
```

Table created.

CONSTRAINTS ADDED TO TABLES:

```
SQL> alter table corder10 add constraint fk_cust2 foreign key (custno) references customer10 (custno);
```

Table altered.

```
SQL> alter table orderitem10 add constraint fk1_ord foreign key (orderno) references corder10 (orderno);
```

Table altered.

```
SQL> alter table orderitem10 add constraint fk3_itm foreign key (itemno) references item10 (itemno);
```

Table altered.

TABLE DESCRIPTIONS:

```
SQL> desc customer10;
 Name                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTNO                    NOT NULL NUMBER (10)
 CNAME                     NOT NULL VARCHAR2 (25)
 CITY                      NOT NULL VARCHAR2 (10)

SQL> desc corder10;
```

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDERNO                                   NOT NULL NUMBER (10)
 ORDERDATE                                 NOT NULL DATE
 CUSTNO                                             NUMBER (10)
 ORDAMT                                             NUMBER (8,2)

SQL> desc orderitem10;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDERNO                                   NOT NULL NUMBER (10)
 ITEMNO                                    NOT NULL NUMBER (10)
 QTY                                       NOT NULL NUMBER (10)

SQL> desc item10;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEMNO                                    NOT NULL NUMBER (10)
 UNITPRICE                                          NUMBER (8,2)

SQL> desc shipment10;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDERNO                                            NUMBER (10)
 WAREHOUSENO                                        CHAR (2)
 SHIPDATE                                           DATE
```

INSERTING VALUES IN CUSTOMER10 TABLE:

```
SQL> insert into customer10 values (1001,'roots','comibatore');
```

1 row created.

```
SQL> insert into customer10 values (1002,'pricol','chennai');
```

1 row created.

```
SQL> insert into customer10 values (1003, 'lmw', 'chennai');
```

1 row created.

INSERTING VALUES IN CORDER10 TABLE:

```
SQL> insert into corder10 values (1, '15-jan-1999',1001,1500.75);
```

1 row created.

```
SQL> insert into corder10 values (2, '5-july-1998', 1002,1750.23);
```

1 row created.

```
SQL> insert into corder10 values (4,'20-dec-1999',1002,359.75);
```

1 row created.

INSERTING VALUES IN ITEM10 TABLE:

```
SQL> insert into item10 (itemno,unitprice)values (501,400.25);
```

1 row created.

SQL> insert into item10 (itemno,unitprice)values (205,220.75);

1 row created.

SQL> insert into item10 (itemno,unitprice)values (503,4550.75);

1 row created.

SQL> insert into item10 (itemno,unitprice)values (505,100.20);

1 row created.

INSERTING VALUES IN ORDERITEM10 TABLE:

SQL> insert into orderitem10 values (1,501,55);

1 row created.

SQL> insert into orderitem10 values (2,205,50);

1 row created.

SQL> insert into orderitem10 values (4,501,78);

1 row created.

INSERTING VALUES IN SHIPMENT10 TABLE:

SQL> insert into shipment10 values (1,'w2','25-mar-1995');

1 row created.

SQL> insert into shipment10 values (2,'w2','1-jan-1998');

1 row created.

SQL> insert into shipment10 values (3,'w2','20-sep-2000');

1 row created.

SQL> insert into shipment10 values (4,'w2','16-feb-2005');

1 row created.

SQL> insert into shipment10 values (5,'w2','15-aug-1998');

1 row created.

DISPLAYING TABLE CONTENTS:

SQL> select * from customer10;

CUSTNO	CNAME	CITY
1001	roots	comibatore
1002	pricol	chennai
1003	lmw	chennai

```
SQL> select * from corder10;

  ORDERNO ORDERDATE     CUSTNO     ORDAMT
---------- --------- ---------- ----------
       1 15-JAN-99       1001    1500.75
       2 05-JUL-98       1002    1750.23
       4 20-DEC-99       1002     359.75

SQL> select * from orderitem10;

  ORDERNO     ITEMNO       QTY
---------- ---------- ----------
       1       501        55
       2       205        50
       4       501        78

SQL> select * from item10;

   ITEMNO  UNITPRICE
---------- ----------
     501     400.25
     205     220.75
     503    4550.75
     505      100.2

SQL> select * from shipment10;

  ORDERNO WA SHIPDATE
---------- -- ---------
       1 w2 25-MAR-95
       2 w2 01-JAN-98
       3 w2 20-SEP-00
       4 w2 16-FEB-05
       5 w2 15-AUG-98
```

DML QUERIES:

a. List the orderno and shipdate for all orders shipped from warehouse 'w2'.

```
SQL> select orderno,shipdate from shipment10 where warehouseno='w2';

  ORDERNO SHIPDATE
---------- ---------
       1 25-MAR-95
       2 01-JAN-98
       3 20-SEP-00
       4 16-FEB-05
       5 15-AUG-98
```

b. Create a view consistinf custname, no of_orders, avg_order_amt where no of orders is the total number of orders by the customer and avg_order_amt is the average order amount of the customer.

```
SQL> create or replace view cust_ord_view as (select cname,count(*) as noof_orders, avg(ordamt) as
avg_order_amt from corder10,customer10 where corder10.custno=customer10.custno group by cname);

View created.
SQL> select * from cust_ord_view;

CNAME                NOOF_ORDERS AVG_ORDER_AMT
------------------------ ---------- -------------
pricol                        2      1054.99
```

roots 1 1500.75

A. WRITE A PROCEDURE TO INSERT THE DETAILS INTO ORDER10 TABLE.

```
SQL> create or replace procedure ins_proc(orno number, ordt date, csno number,ordamt number)
 2  is
 3  begin
 4  insert into corder10 (orderno,orderdate,custno,ordamt)values (orno,ordt,csno,ordamt);
 5  end ins_proc;
 6  /
```

Procedure created.

BEFORE PROCEDURE EXECUTION:

```
SQL> select * from corder10;

  ORDERNO ORDERDATE    CUSTNO    ORDAMT
---------- --------- ---------- ----------
       1 15-JAN-99      1001    1500.75
       2 05-JUL-98      1002    1750.23
       4 20-DEC-99      1002     359.75
```

PROCEDURE EXECUTION:

```
SQL> begin
 2  ins_proc (5,'31-dec-2009',1003,1525.32);
 3  dbms_output.put_line ('data inserted into the order table successfully');
 4  end;
 5  /
```

PL/SQL procedure successfully completed.

AFTER PROCEDURE EXECUTION:

```
SQL> select * from corder10;

  ORDERNO ORDERDATE    CUSTNO    ORDAMT
---------- --------- ---------- ----------
       1 15-JAN-99      1001    1500.75
       2 05-JUL-98      1002    1750.23
       4 20-DEC-99      1002     359.75
       5 31-DEC-09      1003    1525.32
```

B. WRITE A PROCEDURE TO DELETE THE RECORD FROM SHIPMENT WHERE ORDER NOT SHIPPED WITHIN 30 DAYS OF ORDERING.

```
SQL> create or replace procedure ship_del is
 2  begin
 3  delete from shipment10
 4  where orderno in (select corder10.orderno from corder10,shipment10
where(months_between(orderdate,shipment10.shipdate) >= 1));end ship_del;
 5  /
```

Procedure created.

BEFORE PROCEDURE EXECUTION:

```
SQL> select * from shipment10;

  ORDERNO WA SHIPDATE
---------- -- ---------
       1 w2 25-MAR-95
       2 w2 01-JAN-98
       3 w2 20-SEP-00
```

4 w2 16-FEB-05
5 w2 15-AUG-98

PROCEDURE EXECUTION:

```
SQL> declare begin ship_del();
  2  end;
  3  /
```

PL/SQL procedure successfully completed.

AFTER PROCEDURE EXECUTION:

```
SQL> select * from shipment10;

   ORDERNO WA SHIPDATE
---------- -- ---------
         3 w2 20-SEP-00
```

COMPANY DATABASE WITH PACKAGE

TABLE CREATION:

Design a company database with the following tables:

SQL> create table customer10 (custno number (10) primary key, cname varchar2 (25) not null, city varchar2 (10) not null);

Table created.

corder10 (orderno,orderdate,custno,ordamt)

SQL> create table corder10 (orderno number (10) primary key,orderdate date not null, custno number (10), ordamt number (8,2));

Table created.

orderitem10 (orderno,itemno,qty)

SQL> create table orderitem10 (orderno number (10) not null,itemno number (10) not null, qty number (10)not null);

Table created.

item10 (itemno,unitprice)

SQL> create table item10 (itemno number (10) primary key, unitprice number (8,2));

Table created.

shipment10 (orderno,warehouseno,shipdate)

SQL> create table shipment10 (orderno number (10), warehouseno char (2), shipdate date);

Table created.

CONSTRAINTS ADDED TO TABLES:

SQL> alter table orderitem10 add constraint fk1_ord foreign key (orderno) references corder10 (orderno);

Table altered.

SQL> alter table orderitem10 add constraint fk3_itm foreign key (itemno) references item10 (itemno);

Table altered.

TABLE DESCRIPTIONS:

SQL> desc customer10;

Name	Null?	Type
CUSTNO	NOT NULL	NUMBER (10)
CNAME	NOT NULL	VARCHAR2 (25)
CITY	NOT NULL	VARCHAR2 (10)

SQL> desc corder10;

Name	Null?	Type
ORDERNO	NOT NULL	NUMBER (10)
ORDERDATE	NOT NULL	DATE
CUSTNO		NUMBER (10)
ORDAMT		NUMBER (8,2)

```
SQL> desc orderitem10;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDERNO                                   NOT NULL NUMBER (10)
 ITEMNO                                    NOT NULL NUMBER (10)
 QTY                                       NOT NULL NUMBER (10)

SQL> desc item10;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEMNO                                    NOT NULL NUMBER (10)
 UNITPRICE                                          NUMBER (8,2)

SQL> desc shipment10;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDERNO                                            NUMBER (10)
 WAREHOUSENO                                        CHAR (2)
 SHIPDATE                                           DATE
```

INSERTING VALUES IN CUSTOMER10 TABLE:

```
SQL> insert into customer10 values (1001,'roots','comibatore');
```

1 row created.

```
SQL> insert into customer10 values (1002,'pricol','erode');
```

1 row created.

```
SQL> insert into customer10 values (1003, 'lmw', 'chennai');
```

1 row created.

INSERTING VALUES IN CORDER10 TABLE:

```
SQL> insert into corder10 values (1, '15-jan-1999',1001,1500.75);
```

1 row created.

```
SQL> insert into corder10 values (2, '5-july-1998', 1002,1750.23);
```

1 row created.

```
SQL> insert into corder10 values (4,'20-dec-1999',1002,359.75);
```

1 row created.

INSERTING VALUES IN ITEM10 TABLE:

```
SQL> insert into item10 (itemno,unitprice)values (501,400.25);
```

1 row created.

```
SQL> insert into item10 (itemno,unitprice)values (205,220.75);
```

1 row created.

```
SQL> insert into item10 (itemno,unitprice)values (503,4550.75);
```

1 row created.

SQL> insert into item10 (itemno,unitprice)values (505,100.20);

1 row created.

INSERTING VALUES IN ORDERITEM10 TABLE:

SQL> insert into orderitem10 values (1,501,55);

1 row created.

SQL> insert into orderitem10 values (2,205,50);

1 row created.

SQL> insert into orderitem10 values (4,501,78);

1 row created.

INSERTING VALUES IN SHIPMENT10 TABLE:

SQL> insert into shipment10 values (1,'w2','25-mar-1995');

1 row created.

SQL> insert into shipment10 values (2,'w2','1-jan-1998');

1 row created.

SQL> insert into shipment10 values (3,'w2','20-sep-2000');

1 row created.

SQL> insert into shipment10 values (4,'w2','16-feb-2005');

1 row created.

SQL> insert into shipment10 values (5,'w2','15-aug-1998');

1 row created.

DISPLAYING TABLE CONTENTS:

SQL> select * from customer10;

CUSTNO	CNAME	CITY
1001	roots	comibatore
1002	pricol	erode
1003	lmw	chennai

SQL> select * from corder10;

ORDERNO	ORDERDATE	CUSTNO	ORDAMT
1	15-JAN-99	1001	1500.75
2	05-JUL-98	1002	1750.23
4	20-DEC-99	1002	359.75

```
SQL> select * from orderitem10;

  ORDERNO     ITEMNO       QTY
---------- ---------- ----------
       1       501        55
       2       205        50
       4       501        78

SQL> select * from item10;

   ITEMNO  UNITPRICE
---------- ----------
      501     400.25
      205     220.75
      503    4550.75
      505      100.2

SQL> select * from shipment10;

  ORDERNO WA SHIPDATE
---------- -- ---------
       1 w2 25-MAR-95
       2 w2 01-JAN-98
       3 w2 20-SEP-00
       4 w2 16-FEB-05
       5 w2 15-AUG-98
```

DML QUERIES:

I) A PROCEDURE TO UPDATE THE DETAILS INTO THE ITEM10 TABLE.
II) A FUNCTION WHICH TAKE ITEMNO AS INPUT AND RETURNS THE NUMBER OF ORDERS FOR THAT ITEM.

```
SQL> create or replace package ex2_pkg as procedure
  2  upd_itms (ino number, prc number, ono number);
  3  function get_orders (ino number) return number;
  4  end ex2_pkg;
  5  /
```

Package created.

```
SQL> create or replace package body ex2_pkg as
  2  procedure upd_itms (ino number, prc number, ono number) is
  3  begin update item10 set itemno=ino, unitprice=prc
  4  where itemno=ono;
  5  end upd_itms;
  6  function get_orders (ino number) return number is ct number;
  7  begin select count(*) into ct from orderitem10 where itemno=ino;
  8  return ct;
  9  end get_orders;
 10  end ex2_pkg;
 11  /
```

Package body created.

BEFORE EXECUTION:

```
SQL> select * from item10;

   ITEMNO  UNITPRICE
```

```
---------- ----------
       501     400.25
       205     220.75
       503    4550.75
       505      100.2
```

PACKAGE EXECUTION:

```
SQL> begin ex2_pkg.upd_itms (501,175.97,501);
  2  end;
  3  /
```

PL/SQL procedure successfully completed.

AFTER PACKAGE EXECUTION:

```
SQL> select * from item10;

    ITEMNO  UNITPRICE
---------- ----------
       501     175.97
       205     220.75
       503    4550.75
       505      100.2



SQL> set serveroutput on
SQL> declare res number;
  2  begin res:=ex2_pkg.get_orders (501);
  3  dbms_output.put_line ('The number of orders for ' || '501,'||'is'||res);
  4  end;
  5  /
The number of orders for 501,is2
```

PL/SQL procedure successfully completed.

HOTEL DATABASE WITH PACKAGE

TABLE CREATION:

Design a company database with the following tables:
hotel10 (hotelno, hotelname, city)

```
SQL> create table hotel10 (hotelno char (2) primary key, hotelname varchar2 (30), city varchar2 (25));
```

Table created.

room10 (roomno, type, price)

```
SQL> create table room10 (roomno number primary key, hotelno char (2),type varchar2 (15), price number (10,2));
```

Table created.

guest10 (guestno, guestname)

```
SQL> create table guest10 (guestno number primary key, guestname varchar2 (25));
```

Table created.

booking10 (hotelno, guestno, datefrom, dateto, roomno)

```
SQL> create table booking10 (hotelno char (2), guestno number, datefrom date, dateto date, roomno number);
```

Table created.

CONSTRAINTS ADDED TO TABLES:

```
SQL> alter table room10 add constraint hotel_fk foreign key (hotelno) references hotel10 (hotelno);
```

Table altered.

```
SQL> alter table booking10 add constraint fk_gt foreign key (guestno) references guest10 (guestno);
```

Table altered.

```
SQL> alter table booking10 add constraint htl_fk foreign key (hotelno) references hotel10 (hotelno);
```

Table altered.

```
SQL> alter table booking10 add constraint fk_room foreign key (roomno) references room10 (roomno);
```

Table altered.

TABLE DSCRIPTIONS:

```
SQL> desc hotel10;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 HOTELNO                                   NOT NULL CHAR (2)
 HOTELNAME                                          VARCHAR2 (30)
 CITY                                               VARCHAR2 (25)

SQL> desc room10;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROOMNO                                    NOT NULL NUMBER
 HOTELNO                                            CHAR (2)
```

```
TYPE                                    VARCHAR2 (15)
PRICE                                   NUMBER (10,2)

SQL> desc guest10;
 Name                          Null?    Type
 ----------------------------------------- -------- ----------------------------
 GUESTNO                         NOT NULL NUMBER
 GUESTNAME                                VARCHAR2 (25)

SQL> desc booking10;
 Name                          Null?    Type
 ----------------------------------------- -------- ----------------------------
 HOTELNO                                  CHAR (2)
 GUESTNO                                  NUMBER
 DATEFROM                                  DATE
 DATETO                                  DATE
 ROOMNO                                   NUMBER
```

INSERTING VALUES IN HOTEL10 TABLE:

```
SQL> insert into hotel10 values ('h1','paradise inn','chennai');

1 row created.

SQL> insert into hotel10 values ('h2','taj hotel','chennai');

1 row created.

SQL> insert into hotel10 values ('h3','paradise inn','bangalore');

1 row created.

SQL> insert into hotel10 values ('h4','taj hotel','mumbai');

1 row created.

SQL> insert into hotel10 values ('h5','oberai','pune');

1 row created.
```

INSERTING VALUES IN ROOM10 TABLE:

```
SQL> insert into room10 values (100, 'h1','deluxe',5500);

1 row created.

SQL> insert into room10 values (101, 'h2','deluxe', 1450);

1 row created.

SQL> insert into room10 values (102, 'h3','deluxe', 2750);

1 row created.

SQL> insert into room10 values (104, 'h4','deluxe', 5500);

1 row created.

SQL> insert into room10 values (105, 'h5','deluxe', 2000);
```

1 row created.

SQL> insert into room10 values (106, 'h1','ac', 1500);

1 row created.

INSERTING VALUES IN GUEST10 TABLE:

SQL> insert into guest10 (guestno, guestname) values (1, 'xyz');

1 row created.

SQL> insert into guest10 (guestno, guestname) values (2, 'divi');

1 row created.

SQL> insert into guest10 (guestno, guestname) values (3, 'sugan');

1 row created.

SQL> insert into guest10 (guestno, guestname) values (4, 'kani');

1 row created.

SQL> insert into guest10 (guestno, guestname) values (5, 'viji');

1 row created.

SQL> insert into guest10 (guestno, guestname) values (6, 'nithi');

1 row created.

INSERTING VALUES IN BOOKING10 TABLE:

SQL> insert into booking10 values ('h1',2, '5-feb-2009', '15-feb-2009', 100);

1 row created.

SQL> insert into booking10 values ('h1',2, '1-jan-2009', '1-jan-2009', 106);

1 row created.

SQL> insert into booking10 values ('h2',1, '9-feb-2009', '16-feb-2009', 101);

1 row created.

SQL> insert into booking10 values ('h3',3, '1-mar-2009', '5-mar-2009', 102);

1 row created.

SQL> insert into booking10 values ('h4',5, '7-mar-2009', '15-mar-2009', 104);

1 row created.

DISPLAYING TABLE CONTENTS:

SQL> select * from hotel10;

HO HOTELNAME CITY

```
-- ------------------------------ -------------------------
h1 paradise inn               chennai
h2 taj hotel                 chennai
h3 paradise inn               bangalore
h4 taj hotel                 mumbai
h5 oberai                    pune

SQL> select * from room10;

   ROOMNO HO TYPE               PRICE
---------- -- --------------- ----------
      100 h1 deluxe             5500
      101 h2 deluxe             1450
      102 h3 deluxe             2750
      104 h4 deluxe             5500
      105 h5 deluxe             2000
      106 h1 ac                1500
```

6 rows selected.

```
SQL> select * from guest10;

  GUESTNO GUESTNAME
---------- -------------------------
       1 xyz
       2 divi
       3 sugan
       4 kani
       5 viji
       6 nithi
```

6 rows selected.

```
SQL> select * from booking10;

HO    GUESTNO DATEFROM  DATETO       ROOMNO
-- ---------- --------- --------- ----------
h1         2 05-FEB-09 15-FEB-09       100
h1         2 01-JAN-09 01-JAN-09       106
h2         1 09-FEB-09 16-FEB-09       101
h3         3 01-MAR-09 05-MAR-09       102
h4         5 07-MAR-09 15-MAR-09       104
```

<u>WRITE A PL/SQL PACKAGE WITH:</u>
I) A PROCEDURE TO INSERT THE DETAILS INTO THE HOTEL10 TABLE.
II) A FUNCTION WHICH TAKE HOTELNO AS INPUT AND RETURNS THE TOTAL INCOME OF THAT HOTEL.

```
SQL> create or replace package htl_pkg
  2  as
  3  procedure htl_insert(hno char, hna varchar2,cit varchar2);
  4  function ret_htl_dtls(htlno in char) return number;
  5  end htl_pkg;
  6  /
```

Package created.

```
SQL> create or replace package body htl_pkg
```

```
 2  as
 3  procedure htl_insert (hno char, hna varchar2, cit varchar2) as
 4  begin
 5  insert into hotel10 values (hno,hna,cit);
 6  end htl_insert;
 7  function ret_htl_dtls(htlno in char) return number as income number;
 8  begin
 9  select sum(price) into income
10  from room10
11  where roomno in
12  (
13  select roomno
14  from booking10
15  where hotelno=htlno);
16  return income;
17  end ret_htl_dtls;
18  end htl_pkg;
19  /
```

Package body created.

BEFORE EXECUTION:

```
SQL> select * from hotel10;

HO HOTELNAME                    CITY
-- ------------------------------ ------------------------
h1 paradise inn                   chennai
h2 taj hotel                    chennai
h3 paradise inn                   bangalore
h4 taj hotel                    mumbai
h5 oberai                       pune
```

PACKAGE EXECUTION:

```
SQL> set serveroutput on
SQL> begin
  2  dbms_output.put_line('total income is:'||htl_pkg.ret_htl_dtls(&htl_no));
  3  end;
  4  /
Enter value for htl_no: 'h1'
old   2: dbms_output.put_line('total income is:'||htl_pkg.ret_htl_dtls(&htl_no));
new   2: dbms_output.put_line('total income is:'||htl_pkg.ret_htl_dtls('h1'));
total income is:7000
```

PL/SQL procedure successfully completed.

```
SQL> begin
  2  htl_pkg.htl_insert(&hotel_no, &hotel_name, &city);
  3  end;
  4  /
Enter value for hotel_no: 'h6'
Enter value for hotel_name: 'residency'
Enter value for city: 'cbe'
old   2: htl_pkg.htl_insert(&hotel_no, &hotel_name, &city);
new   2: htl_pkg.htl_insert('h6', 'residency', 'cbe');
```

PL/SQL procedure successfully completed.

AFTER PACKAGE EXECUTION:

```
SQL> select * from hotel10;

HO HOTELNAME                    CITY
-- ------------------------------ ------------------------
h1 paradise inn                 chennai
h2 taj hotel                    chennai
h3 paradise inn                 bangalore
h4 taj hotel                    mumbai
h5 oberai                       pune
h6 residency                    cbe

6 rows selected.
```