**DATABASE MANAGEMENT SYSTEMS LAB-MANUAL**

|  |  |
| --- | --- |
| **Course Code** : **IS415L** | **Credits** : **0:0:2** |
| **Prerequisite**s: **Nil** | **Contact Hours : 28** |
| **Course coordinator(s): Savita Shetty** |  |

**Course objectives:**

* Execute SQL commands
* Implement simple exercises on relational database schema
* Design a relational database schema for specific database application using SQL and PL/SQL
* Apply the normalization procedure on relational database schema
* Develop the database system to handle the real world problem

**Course Contents:**

The Database Management Systems Lab consists of two components viz. Part A: Program Execution and Part B: Mini Project.

**Part A**

**Execute the following exercises in SQL**

1. Consider the relations

EMPLOYEE(SSN, Name, DeptNo),

ASSIGNED\_TO(USN , ProjectNo)

PROJECT(ProjectNo, ProjectArea).

Create the above tables, insert suitable tuples and perform the following operations in SQL:

1. Obtain the SSN of employees assigned to database projects.
2. Find the number of employees working in each department
3. Update the ProjectNo of Employee bearing SSN=1 to ProjectNo=20
4. Consider the relations

PART(PNO, PNAME, COLOUR),

SUPPLIER( SNO,SNAME,ADDRESS)

SUPPLY(PNO,SNO,QUANTITY)

Create the above tables, insert suitable tuples and perform the following operations in SQL:

1. Obtain the PNO of parts supplied by supplier ‘Ram’.
2. Obtain the Names of suppliers who supply bolts
3. Delete the parts which are green in colour
4. Consider the relations

BOAT(BID, BNAME, COLOUR),

SAILOR(SID, SNAME, AGE, RATING)

RESERVES(BID,SID, DAY)

Create the above tables, insert suitable tuples and perform the following operations in SQL:

1. Obtain the bid of the boats reserved by ‘Ram’.
2. Retrieve the bid of the boats reserved by all the sailors.
3. Find the number of boats reserved by each sailor
4. Consider the relations

PART(PNO, PNAME, COLOUR),

WAREHOUSE( WNO,WNAME,CITY)

SHIPMENT(PNO,WNO,QUANTITY,DATE)

Create the above tables, insert suitable tuples and perform the following operations in SQL:

1. Obtain the Names of warehouses which have shipped red coloured parts.
2. Retrieve the PNO of the parts shipped by all the warehouses.
3. Find the number of parts supplied by each warehouse
4. Consider the relations

BOOK(ISBN, TITLE,AUTHOR,PUBLISHER)

STUDENT(USN, NAME, SEM, DEPTNO),

BORROW(ISBN, USN, DATE)

Create the above tables, insert suitable tuples and perform the following operations in SQL:

1. Obtain the name of the student who has borrowed the book bearing ISBN ‘123’
2. Obtain the Names of students who have borrowed database books.
3. Find the number of books borrowed by each student.

**Execute the following programs in PL/SQL**

1. Consider the following database for a BANK system

BRANCH(Code, Name, Assets)

CUSTOMER(SSN, Name, Place)

ACCOUNT(AccNo, SSN, Code, Balance)

1. Create the above tables by stating the primary and foreign keys
2. Insert the following tuples to the tables

BRANCH CUSTOMER

|  |  |  |
| --- | --- | --- |
| **Code** | **Name** | **Assets** |
| B1 | MSR | 10000 |
| B2 | RNR | 20000 |
| B3 | SMR | 15000 |
| B4 | SKR | 25000 |

|  |  |  |
| --- | --- | --- |
| **SSN** | **Name** | **Place** |
| 1 | Ram | BNG |
| 2 | Asha | MNG |
| 3 | Usha | MYS |
| 4 | Sri | DEL |

ACCOUNT

|  |  |  |  |
| --- | --- | --- | --- |
| **AccNo** | **SSN** | **Code** | **Balance** |
| A1 | 1 | B1 | 100000 |
| A2 | 1 | B1 | 200000 |
| A3 | 2 | B2 | 100000 |
| A4 | 3 | B2 | 100000 |
| A5 | 3 | B2 | 100000 |
| A5 | 3 | B2 | 100000 |
| A7 | 4 | B2 | 200000 |

Write a PL/SQL program to display the contents of the above tables and then update the balance of a few accounts.

1. (a) Write a PL/SQL program to check whether a given number is prime or not

(b) Using cursors demonstrate the process of copying the contents of one table to a new table

1. (a) Write a program that gives all employees in department 10 a 15% pay increase. Display a message displaying how many employees were awarded the increase.

(b) Create a program that accepts two numbers. If the first is larger than the second raise an exception called e\_bigger and display an appropriate message.

1. (a) Write a PL/SQL program to print the first 8 fibonacci numbers

(b) Write a PL/SQL program to display the day of the week taking system

date as the input

1. (a) Write a PL/SQL procedure to find the factorial of a given number and a program

to call the same

(b) Consider the following relation schema.

EMPLOYEE (SSN, Name, sal, DeptNo)

Display SSN and name of employees of the department entered by the user as input

1. (a) Write a PL/SQL program to check whether a given number is palindrome or not

(b) Consider the following relation schema.

EMPLOYEE (SSN, Name, sal, DeptNo)

Write a trigger to raise an error if the table is modified on a specific day (Eg., Saturday or Sunday) of the week.

**Part B**

**Mini Project**

A team project in groups of maximum 4 students has to be carried out. The work and a project report should be submitted for demonstration and evaluation. Related to the case study identified by the students, they should be able to design and develop a database application using the following approach.

* Data Requirements collection
* Functional Requirements Collection
* Entity Relationship Model Design
* ER-to-Relational mapping and Normalization
* SQL :Data definition-Create the tables that have been designed. Populate the table
* SQL:Data manipulation
* Create Views and Triggers
* SQL:Queries- Execute the queries
* Complex SQL queries
* Front end design

**Text Book**

Benjamin Rosenzweig, Elena Silvestrova Rakhimov : Oracle PL/SQL by Example, 4th Edition, 2010.

**Course Delivery:**

The Course will be delivered through Demonstration, Class assignments, Presentations, Project based learning (Mini Project)

**Course Assessment and Evaluation**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **What** | | **To whom** | **When/ Where (Frequency in the course)** | **Max marks** | **Evidence collected** | **Contributing to Course Outcomes** |
| **Direct Assessment Methods** | **CIE** | Lab Internal Assessment Tests on exercises | Students | Twice(Average of two will be computed) | 25 | Blue books | C01, C02, C03, C04 and C05 |
| Regular evaluation on the exercises | Weekly(Average of all the weeks) | 5 | Attendance Register and Record book | C01, C02, C03, C04 and C05 |
| Mini Project Regular Evaluation | Weekly (Average of all the weeks) | 5 | Attendance register and record book | CO1,CO4,CO5 |
|  | Mini Project Final Demo | Once | 15 | Report | CO4,CO5 |
| **SEE** | Standard Examination | End of course (Answering 5 of 10 questions) | 50 | Answer scripts | C01, C02, C03, C04 and C05 |
| **Indirect Assessment Methods** | **Students feedback** | | Students | Middle of the course | - | Feedback forms | C01, C02, C03 |
| **End of course survey** | | End of course | - | Question-naire | C01, C02, C03, C04 and C05 |

Questions for CIE and SEE will be designed to evaluate the various educational components (Bloom’s taxonomy) such as:

CIE and SEE evaluation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Levels of Bloom’s Taxonomy** | **LabTest1** | **Lab Test2** | **MiniProject** | **SEE** |
| **1** | **Remember** | 10 | 10 | 10 | 10 |
| **2** | **Understand** | 10 | 10 | 10 | 30 |
| **3** | **Apply** | 20 | 20 | 20 | 30 |
| **4** | **Analyze** | 20 | 20 | 20 | 10 |
| **5** | **Evaluate** | 20 | 20 | 20 | 10 |
| **6** | **Create** | 20 | 20 | 20 | 10 |

**Course outcomes:**

**Students will be able to**

1. Execute SQL commands
2. Implement simple exercises on relational database schema
3. Design a relational database schema for specific database application using SQL and PL/SQL
4. Apply the normalization procedure on relational database schema
5. Develop the database system to handle the real world problem

**Mapping Course Outcomes with Program Outcomes:**

| **Course Outcomes** | **Program Outcomes\*** | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** |
| Execute SQL commands | X | X |  |  |  |  |  |  |  |  |  |  |
| Implement simple exercises on relational database schema |  |  |  | X | X |  |  |  |  |  | X | X |
| Design a relational database schema for specific database application using SQL and PL/SQL | X | X |  | X | X |  |  |  |  |  |  | X |
| Apply the normalization procedure on relational database schema | X |  |  |  | X |  |  |  |  |  |  |  |
| Develop the database system to handle the real world problem |  | X |  |  | X |  |  |  |  |  | X | X |

QUESTIONS AND SOLUTIONS:

1.Consider the relations

EMPLOYEE(SSN, Name, DeptNo),

ASSIGNED\_TO(USN , ProjectNo)

PROJECT(ProjectNo, ProjectArea).

Create the above tables, insert suitable tuples and perform the following operations in SQL:

1. Obtain the SSN of employees assigned to database projects.
2. Find the number of employees working in each department
3. Update the ProjectNo of Employee bearing SSN=1 to ProjectNo=20

SOLUTION:

create table e  
 (  
 ssn varchar(6),  
 name varchar(10),  
 deptno int,  
 primary key(ssn)  
 );  
  
 Table created.  
  
 create table p  
 (  
 projectno varchar(10),  
 projectarea varchar(20),  
 primary key(projectno)  
 );  
  
 Table created.  
  
 create table a  
 (  
 usn varchar(6),  
 projectno varchar(10),  
 foreign key(usn)references e(ssn),  
 foreign key(projectno)references p(projectno)  
 );  
  
 Table created.  
  
 insert into e  
 values('01','abc',10);  
  
 1 row updated  
  
 insert into e  
 values('02','xyz',20);  
  
 1 row updated  
  
 insert into e  
 values('03','pqr',30);  
  
 1 row updated  
  
  
 insert into e  
 values('04','lmn',40);  
  
 1 row updated  
  
 insert into p  
 values('100','database');  
  
 1 row updated  
  
 insert into p  
 values('200','network');  
  
 1 row updated  
  
 insert into p  
 values('300','android');  
  
 1 row updated  
  
 insert into a  
 values('01','100');  
  
 1 row updated  
  
  
 insert into a  
 values('02','200');  
  
 1 row updated  
  
 insert into a  
 values('03','300');  
  
 1 row updated  
  
 insert into a  
 values('01','200');  
  
 1 row updated  
  
 select \*  
 from e;  
  
 SSN  NAME  DEPTNO  
  
 01       abc         10  
  
 02        xyz         20  
  
 03        pqr         30  
  
 04        lmn         40  
  
 select \*  
 from p;  
  
 PROJECTNOPROJECTAREA100database200network300android  
  
 select \*  
 from a;  
  
 USN       PROJECTNO  
  
 01           100  
  
 02           200  
  
 03           300  
  
 01           200  
  
 select ssn from e  
 where ssn=(select usn  
            from a   
             where projectno=(select projectno from p  
                              where projectarea='database'));  
  
 SSN  
  
 01  
  
 select count(ssn),deptno  
 from e group by deptno;  
  
 COUNT(SSN)    DEPTNO  
  
 1                         30  
  
 1                         20  
  
 1                         40  
  
1                          10  
  
 update a  
 set projectno='200' where  
 usn='03';  
  
 1 row updated  
  
 select \*  
  
 from a;  
  
 usn       projectno  
  
 01             100  
  
 02              200  
  
 03              200  
  
 04              200

2. Consider the relations

PART(PNO, PNAME, COLOUR),

SUPPLIER( SNO,SNAME,ADDRESS)

SUPPLY(PNO,SNO,QUANTITY)

Create the above tables, insert suitable tuples and perform the following operations in SQL:

1. Obtain the PNO of parts supplied by supplier ‘Ram’.
2. Obtain the Names of suppliers who supply bolts
3. Delete the parts which are green in colour

SOLUTION:

create table part  
 (  
 pno number(10),  
 pname varchar(20),  
 colour varchar(20),  
 primary key(pno)  
 );  
  
 table created.  
  
    
  
 Create table supplier  
 (  
 sno number(10),  
 sname varchar(20),  
 address varchar(20),  
 primary key(sno)  
 );  
  
 table created.  
  
 create table supply  
 (  
 pno number(10),  
 sno number(10),  
 quantity varchar(20),  
 primary key(pno,sno),  
 foreign key(pno) references part(pno)on delete cascade,  
 foreign key(sno) references supplier(sno)on delete cascade  
 );  
  
 Table created.  
  
 insert into part values(1,'plug','black');  
  
 1 row(s) inserted.  
  
 insert into part values(2,'bolt','blue');  
  
 1 row(s) inserted.  
  
 insert into part values(3,'nut','green');  
  
 1 row(s) inserted.  
  
 insert into supplier values(10,'Anoop','udupi');  
  
 1 row(s) inserted.insert into supplier values(15,'Bharath','mangalore');1 row(s) inserted.insert nto supplier values(20,'Ram','bangalore');1 row(s) inserted.insert into supply values(1,10,50);

1 row(s) inserted.  
 insert into supply values(2,10,30);1 row(s) inserted.  
 insert into supply values(1,15,70);1 row(s) inserted.insert into supply values(3,15,40);  
 1 row(s) inserted.  
 insert into supply values(1,20,55);1 row(s) inserted.  
 insert into supply values(2,20,65);1 row(s) inserted.  
 insert into supply values(3,20,75);1 row(s) inserted.

select \*  
 from part;

PNO PNAME COLOUR1 plug black2 bolt blue3 nut green

select \*  
 from supply ;

PNOSNOQUANTITY  
  
  1 10 50   
 2 10 30   
  
 1 15 70  
  
 31540  
  
 12055  
  
 22065  
  
 32075  
  
 select \*  
 from supplier ;  
  
 SNO    SNAME   ADDRESS  
  
 10       Anoop            udupi  
  
 15      Bharath          mangalore  
  
 20      Ram               Bangalore  
  
 answer 2)select sname,pname   
 from supplier,supply,part   
 where pname='bolt' AND supply.sno=supplier.sno AND part.pno=supply.pno;  
  
 SNAME    PNAME  
  
 Anoop       bolt  
  
 Ram           bolt  
  
 answer 1)select pno   
 from supply   
 where sno IN(select sno from   
              supplier where  
              sname='Ram');  
  
 PNO  
  
 1  
  
 2  
  
 3  
  
 delete from part where colour='green';  
  
 1 row(s) deleted.

select \*  
 from part;PNO PNAME COLOUR1 plug black2 bolt blue

select \*  
 from supply;PNO SNO QUANTITY 1 10 50 2 10 30

3.Consider the relations

BOAT(BID, BNAME, COLOUR),

SAILOR(SID, SNAME, AGE, RATING)

RESERVES(BID,SID, DAY)

Create the above tables, insert suitable tuples and perform the following operations in SQL:

1. Obtain the bid of the boats reserved by ‘Ram’.
2. Retrieve the bid of the boats reserved by all the sailors.
3. Find the number of boats reserved by each sailor

SOLUTION:

Create table BOAT  
  
 (  
  
 BID varchar(6) NOT NULL,  
  
 BNAME varchar(20),  
  
 COLOUR varchar(10),  
  
 PRIMARY KEY(BID)  
  
 );  
  
 Create table SAILOR  
  
 (  
  
 SID varchar(6)NOT NULL,  
  
 SNAME varchar(20),  
  
 AGE varchar(3),  
  
 RATING varchar(2),  
  
 PRIMARY KEY(SID)  
  
 );  
  
 Create table RESERVES   
  
 (  
  
 BID varchar(6),  
  
 SID varchar(6),  
  
 DAY varchar(10),  
  
 FOREIGN KEY(BID) references BOAT(BID) ON DELETE CASCADE,  
  
 FOREIGN KEY(SID) references SAILOR(SID) ON DELETE CASCADE  
  
 );  
  
 INSERT INTO BOAT  
  
 VALUES('01','ABC','RED');  
  
 INSERT INTO BOAT  
  
 VALUES('02','XYZ','YELLOW');  
  
 INSERT INTO BOAT  
  
 VALUES('03','PQR','GREEN');  
  
 INSERT INTO BOAT  
  
 VALUES('04','LMN','BLACK');  
  
 INSERT INTO BOAT  
  
 VALUES('05','DEF','BLUE');  
  
 INSERT INTO SAILOR  
  
 VALUES('10','RAM','30','5');  
  
 INSERT INTO SAILOR  
  
 VALUES('20','RAVI','25','4');  
  
 INSERT INTO SAILOR  
  
 VALUES('30','MISHRA','22','3');  
  
 INSERT INTO SAILOR  
  
 VALUES('40','CHANDRA','24','2');  
  
 INSERT INTO SAILOR  
  
 VALUES('50','SHIVA','36','1');  
  
 INSERT INTO SAILOR  
  
 VALUES('60','KRISHNA','40','6');  
  
 INSERT INTO RESERVES  
  
 VALUES('01','20','MONDAY');  
  
 INSERT INTO RESERVES  
  
 VALUES('02','30','TUESDAY');  
  
 INSERT INTO RESERVES  
  
 VALUES('03','50','WEDNESDAY');  
  
 INSERT INTO RESERVES  
  
 VALUES('04','10','THURSDAY');  
  
  
 INSERT INTO RESERVES  
  
 VALUES('05','20','FRIDAY');  
  
 SELECT \*   
  
 FROM RESERVES  
  
 ;  
  
 BID SID DAY  
  
 01 20 MONDAY  
  
 02 30 TUESDAY  
  
 03 50 WEDNESDAY  
  
 04 10 THURSDAY  
  
 05 20 FRIDAY  
  
 SELECT \*   
  
 FROM SAILOR  
  
;  
  
 SID SNAME AGE RATING  
  
 10 RAM 30 5  
  
 20 RAVI 25 4  
  
 30 MISHRA 22 3  
  
 40 CHANDRA 24 2  
  
 50 SHIVA 36 1  
  
 60 KRISHNA 40 6  
  
 SELECT \*  
  
 FROM BOAT  
  
 ;  
  
 BID BNAME COLOUR  
  
 01 ABC RED  
  
 02 XYZ YELLOW  
  
 03 PQR GREEN  
 04 LMN BLACK  
  
 05 DEF BLUE  
  
 ans1  
  
SELECT BID  
  
 FROM RESERVES   
  
 WHERE SID IN( SELECT SID FROM   
  
               SAILOR WHERE SNAME='RAM');  
  
 BID  
  
 04  
  
 ans2  
  
 SELECT BID,SNAME  
  
 FROM RESERVES r join SAILOR s  
  
 on (r.SID=s.SID);  
  
 BID SNAME  
  
 04 RAM  
  
 05 RAVI  
  
 01 RAVI  
  
 02 MISHRA  
  
 03 SHIVA  
  
 ans 3  
  
 SELECT COUNT(BID),SID  
  
 FROM RESERVES   
  
 GROUP BY SID;  
  
 COUNT(BID) SID  
  
 1 50  
  
 2 20  
  
 1 10  
  
 1 30

4.Consider the relations

PART(PNO, PNAME, COLOUR),

WAREHOUSE( WNO,WNAME,CITY)

SHIPMENT(PNO,WNO,QUANTITY,DATE)

Create the above tables, insert suitable tuples and perform the following operations in SQL:

1. Obtain the Names of warehouses which have shipped red coloured parts.
2. Retrieve the PNO of the parts shipped by all the warehouses.

Find the number of parts supplied by each warehouse

SOLUTION:

CREATE TABLE PARTT  
 (  
 PNO VARCHAR(6) NOT NULL,  
 PNAME VARCHAR(10) ,  
 COLOUR VARCHAR(10),  
 PRIMARY KEY(PNO)  
 );  
  
 CREATE TABLE WAREHOUSE  
 (  
 WNO VARCHAR(6) NOT NULL,  
 WNAME VARCHAR(10),  
 CITY VARCHAR(10),  
 PRIMARY KEY(WNO)  
 );  
  
 CREATE TABLE SHIPMENT  
 (  
 PNO VARCHAR(6),  
 WNO VARCHAR(6),  
 QUANTITY NUMBER,  
 DATEE DATE ,  
 FOREIGN KEY(PNO) REFERENCES PARTT(PNO) ON DELETE CASCADE,  
 FOREIGN KEY(WNO) REFERENCES WAREHOUSE(WNO) ON DELETE CASCADE  
 );  
  
  
 INSERT INTO PARTT  
 VALUES('01','ABC','RED');  
  
 INSERT INTO PARTT  
VALUES('02','DEF','BLUE');  
  
 INSERT INTO PARTT  
 VALUES('03','LMN','GREEN');  
  
 INSERT INTO PARTT  
 VALUES('04','PQR','YELLOW');  
  
INSERT INTO PARTT  
 VALUES('05','XYZ','PINK');  
  
 INSERT INTO WAREHOUSE   
 VALUES('10','AAA','KUMTA');  
  
 INSERT INTO WAREHOUSE   
 VALUES('20','BBB','MUMBAI');  
  
 INSERT INTO WAREHOUSE   
 VALUES('30','CCC','BANGALORE');  
  
 INSERT INTO WAREHOUSE   
 VALUES('40','DDD','UDUPI');  
  
 INSERT INTO WAREHOUSE   
 VALUES('50','EEE','KARWAR');  
  
  
 SELECT \*  
 FROM PARTT;  
  
 PNO PNAME COLOUR  
 01 ABC RED  
 02 DEF BLUE  
 03 LMN GREEN  
 04 PQR YELLOW  
 05 XYZ PINK  
  
 SELECT \*  
 FROM WAREHOUSE;  
  
 WNO WNAME CITY  
 10 AAA KUMTA  
 20 BBB MUMBAI  
 30 CCC BANGALORE  
 40 DDD UDUPI  
 50 EEE KARWAR  
  
 INSERT INTO SHIPMENTS  
 VALUES('01','20','300','28-FEB-2013');  
  
 INSERT INTO SHIPMENTS  
 VALUES('02','30','400','30-JAN-2013');  
  
 INSERT INTO SHIPMENTS  
 VALUES('03','10','00','31-JAN-2013');  
  
 INSERT INTO SHIPMENTS  
 VALUES('04','40','600','31-MARCH-2013');  
  
 INSERT INTO SHIPMENTS  
 VALUES('05','50','100','31-DEC-2013');  
  
  
 SELECT \*  
 FROM SHIPMENTS;  
 PNO WNO QUANTITY DATEE  
 01 20 300 28-FEB-13  
 02 30 400 30-JAN-13  
 03 10 00 31-JAN-13  
 04 40 600 31-MAR-13  
 05 50 100 31-DEC-13  
  
  
 ans a)  
 SELECT WNAME FROM WAREHOUSE  
 WHERE WNO IN(SELECT WNO FROM SHIPMENTS WHERE PNO=(SELECT PNO ROM PARTT WHERE COLOUR='RED'));  
  
 WNAME  
 BBB  
  
 ans b)  
 SELECT PNO,WNAME  
 FROM SHIPMENTS s join WAREHOUSE w  
 on(s.WNO=w.WNO);  
  
  
 PNO WNAME  
 03 AAA  
 01 BBB  
 02 CCC  
 04 DDD  
 05 EEE  
  
 c)  
 SELECT COUNT(PNO),WNO   
ROM SHIPMENTS   
 GROUP BY WNO;  
  
 COUNT(PNO) WNO  
 1 50  
 1 20  
 1 10  
 1 40  
 1 30  
5.Consider the relations

BOOK(ISBN, TITLE,AUTHOR,PUBLISHER)

STUDENT(USN, NAME, SEM, DEPTNO),

BORROW(ISBN, USN, DATE)

Create the above tables, insert suitable tuples and perform the following operations in SQL:

1. Obtain the name of the student who has borrowed the book bearing ISBN ‘123’
2. Obtain the Names of students who have borrowed database books.

Find the number of books borrowed by each student

SOLUTION:

1. create table books  
    (  
    ISBN varchar(10),  
    Title varchar(10),  
    Author varchar(10),  
    Publisher varchar(10),  
    primary key(ISBN)  
    );  
     
    Insert into books values(  
     
    ISBN TITLE AUTHOR PUBLISHER  
    001 T1 A1 P1  
    002 T2 A2 P2  
    003 T3 A3 P3  
    004 T4 A4 P4  
    005 T5 A5 P5  
     
    create table student1  
    (  
    usn varchar(10),  
   name varchar(10),  
    sem int,  
    dept varchar(3),  
    primary key(usn)  
    );  
     
    Insert into student1 values(  
     
    SN NAME  SEM DEPT  
    111 aaa 3  ISE  
    222 bbb 4 CSE  
    333 ccc 3  CSE  
    444 ddd       4 ISE  
    555 eee 4  ISE  
     
    create table borrow  
    (  
    ISBN varchar(10),  
    usn varchar(10),  
    dates varchar(10),  
    foreign key(ISBN) references books(ISBN),  
    foreign key(usn) references student1(usn)  
   );  
     
    Insert into borrow values(  
     
    ISBN  USN DATES  
    001 222 1/2/13  
   002 333 2/2/13  
    003 111 3/2/13  
    005 444 4/2/13  
   003 555   5/2/13  
     
    Queries:  
    1: select NAME from student1  
    where USN=(select USN from borrow where ISBN='001');  
     
    NAME  
    bbb  
     
     
    2: select NAME from student1  
    where USN=(select USN from borrow where ISBN=(select ISBN from books where TITLE='T2'));  
     
    NAME  
    ccc  
     
    3: select count(ISBN) from borrow  
    group by USN;  
     
    COUNT(ISBN)  
    1  
    1  
    1  
    1  
    1  
   6.Consider the following database for a BANK system

BRANCH(Code, Name, Assets)

CUSTOMER(SSN, Name, Place)

ACCOUNT(AccNo, SSN, Code, Balance)

1. Create the above tables by stating the primary and foreign keys
2. Insert the following tuples to the tables

BRANCH CUSTOMER

|  |  |  |
| --- | --- | --- |
| **Code** | **Name** | **Assets** |
| B1 | MSR | 10000 |
| B2 | RNR | 20000 |
| B3 | SMR | 15000 |
| B4 | SKR | 25000 |

|  |  |  |
| --- | --- | --- |
| **SSN** | **Name** | **Place** |
| 1 | Ram | BNG |
| 2 | Asha | MNG |
| 3 | Usha | MYS |
| 4 | Sri | DEL |

ACCOUNT

|  |  |  |  |
| --- | --- | --- | --- |
| **AccNo** | **SSN** | **Code** | **Balance** |
| A1 | 1 | B1 | 100000 |
| A2 | 1 | B1 | 200000 |
| A3 | 2 | B2 | 100000 |
| A4 | 3 | B2 | 100000 |
| A5 | 3 | B2 | 100000 |
| A5 | 3 | B2 | 100000 |
| A7 | 4 | B2 | 200000 |

Write a PL/SQL program to display the contents of the above tables and then update the balance of a few accounts.

SOLUTION:

6a)

create table branch(code varchar(2),name varchar(10),assets int,primary key(code));

create table customer(ssn int,name varchar(12),place varchar(10),primary key(ssn));

create table account(accno varchar(2),ssn int,code varchar(2),balance int,primary key(accno,ssn,code),foreign key(

code) references branch(code) ON DELETE CASCADE,foreign key(ssn) references customer(ssn) ON DELETE CASCADE);

insert into branch values('b1','msr',10000);

insert into branch values('b2','rnr',20000);

insert into branch values('b3','smr',15000);

insert into branch values('b4','skr',25000);

insert into customer values(1,'ram','bng');

insert into customer values(2,'asha','mng');

insert into customer values(3,'usha','mys');

insert into customer values(4,'sri','del');

insert into account values('a1',1,'b1',100000);

insert into account values('a2',1,'b1',200000);

insert into account values('a3',2,'b2',100000);

insert into account values('a4',3,'b2',100000);

insert into account values('a5',3,'b2',100000);

insert into account values('a6',3,'b2',100000);

insert into account values('a7',4,'b2',200000);

DECLARE

CURSOR CC IS

SELECT \* FROM BRANCH;

V\_CC CC%ROWTYPE;

BEGIN

OPEN CC;

LOOP

FETCH CC INTO V\_CC ;

EXIT WHEN CC %NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('CODE '||V\_CC.CODE ||' NAME '||V\_CC.NAME ||' ASSETS '||V\_CC.ASSETS);

END LOOP;

CLOSE CC;

END;

/

DECLARE

CURSOR CC1 IS

SELECT \* FROM CUSTOMER;

V\_CC1 CC1%ROWTYPE;

BEGIN

OPEN CC1;

LOOP

FETCH CC1 INTO V\_CC1;

EXIT WHEN CC1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('SSN '||V\_CC1.SSN ||' NAME '||V\_CC1.CNAME ||' PLACE '||V\_CC1.PLACE);

END LOOP;

CLOSE CC1;

END;

/

DECLARE

CURSOR CC2 IS

SELECT \* FROM account;

V\_CC2 CC2%ROWTYPE;

BEGIN

OPEN CC2;

LOOP

FETCH CC2 INTO V\_CC2;

EXIT WHEN CC2%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('accno '||V\_CC2.accno ||' ssn '||V\_CC2.ssn ||' code '||V\_CC2.code ||’balance’||V\_CC2.balance);

END LOOP;

CLOSE CC2;

END;

/

b)DECLARE

V\_INC NUMBER:=10;

BEGIN

UPDATE ACCOUNT

SET BALANCE=BALANCE+(BALANCE\*0.1)

WHERE ssn=1;

COMMIT;

END;

/

7.(a) Write a PL/SQL program to check whether a given number is prime or not

(b) Using cursors demonstrate the process of copying the contents of one table to a new table

SOLUTION:

7 a)

SET SERVEROUTPUT ON

DECLARE

n number:=&n;

j number:=2;

counter number:=0;

BEGIN

WHILE(j<=n/2) loop

if mod(n,j)=0 then

dbms\_output.put\_line(n ||' is not prime number');

counter:=1;

exit ;

else

j:=j+1;

end if;

end loop;

if counter=0 then

dbms\_output.put\_line( n || ' is a prime number');

end if;

end;

/

7)b)

create table part1(pno int,pname char(20),colour char(20),primary key(pno));

create table copy\_part1(pno int,pname char(20),colour char(20),primary key(pno));

insert into part1 values(10,'nuts','black');

insert into part1 values(20,'bolts','grey');

insert into part1 values(30,'screw','green');

set serveroutput on

declare

cursor curr is select \*from part1;

counter int;

rows part1%rowtype;

begin

open curr;

loop

fetch curr into rows ;

exit when curr%notfound;

insert into copy\_part1 values(rows.pno,rows.pname,rows.colour);

end loop;

counter := curr%rowcount;

close curr;

dbms\_output.put\_line(counter||' rows inserted into the table copy\_part1 ');

end;

/

8.(a) Write a program that gives all employees in department 10 a 15% pay increase. Display a message displaying how many employees were awarded the increase.

(b) Create a program that accepts two numbers. If the first is larger than the second raise an exception called e\_bigger and display an appropriate message.

SOLUTION:

8)a)set serveroutput on

begin

update employee1

set salary=(1.15\*salary) where deptno=10;

dbms\_output.put\_line('number of rows updated are'||sql%rowcount);

end;

/

8)b)

declare

n1 number(10);

n2 number(10);

ebigger exception;

begin

n1:= & sv1;

n2:= & sv2;

if n1<n2 then

dbms\_output.put\_line('no error because n1 is smaller');

else

raise ebigger;

end if;

exception when ebigger then

dbms\_output.put\_line('exception caught ,n1 is bigger than n2');

END;

/

9.(a) Write a PL/SQL program to print the first 8 fibonacci numbers

(b) Write a PL/SQL program to display the day of the week taking system

date as the input

SOLUTION:

9)a)

declare

a number;

b number;

c number;

n number;

i number;

begin

n:=8;

a:=0;

b:=1;

dbms\_output.put\_line(a);

dbms\_output.put\_line(b);

for i in 1..n-2

loop

c:=a+b;

dbms\_output.put\_line(c);

a:=b;

b:=c;

end loop;

end;

/

9)b)

declare

t\_date date;

current\_day varchar2(9);

begin

t\_date:=sysdate;

current\_day:=to\_char(t\_date,'day');

current\_day:=initcap(current\_day);

current\_day:=rtrim(current\_day);

dbms\_output.put\_line('Today is:'||current\_day);

end;

/

10.(a) Write a PL/SQL procedure to find the factorial of a given number and a program

to call the same

(b) Consider the following relation schema.

EMPLOYEE (SSN, Name, sal, DeptNo)

Display SSN and name of employees of the department entered by the user as input

SOLUTION:

10)b)

create table employee1(ssn varchar(20),name char(20),deptno number(10),salary int,primary key(ssn));

insert into employee1 values(1,'pra',10,1000);

insert into employee1values(2,'aaa',20,2000);

insert into employee1 values(3,'bbb',10,4000);

insert into employee1 values(4,'cc',30,10000);

insert into employee1 values(5,'bgg',10,2000);

select \*from employee1;

set serveroutput on

declare

cursor cur is select \*from employee1;

rows employee1%rowtype;

t int;

begin

open cur;

t:= & t1;

loop

fetch cur into rows;

exit when cur%notfound;

if rows.deptno=t

then

dbms\_output.put\_line('the ssn and names are '|| rows.ssn ||','||rows.name);

end if;

end loop;

close cur;

end;

/

11.(a) Write a PL/SQL program to check whether a given number is palindrome or not

(b) Consider the following relation schema.

EMPLOYEE (SSN, Name, sal, DeptNo)

Write a trigger to raise an error if the table is modified on a specific day (Eg., Saturday or Sunday) of the week.

SOLUTION:

11a)

Declare

n number(10);

i number(10);

sum1 number(10);

k number(10);

Begin

sum1:=0;

n:=&n;

k:=n;

while (n>0) loop

i:=mod(n,10);

sum1:=(sum1\*10)+i;

n:=trunc(n/10);

end loop;

if(k=sum1) then

dbms\_output.put\_line('Given Number is a Palindrome Number');

else

dbms\_output.put\_line('Given Number is not a Palindrome Number');

end if;

end;

/