20.Create the tables by Identifying primary and foreign keys.

1. Insert at least 5 records in each table.
2. Delete the record of book titled “Database System Concepts”.
3. Change the Department of the book titled “Discrete Maths” to “CS”.
4. List all books that belong to “CS” department.
5. List all books that belong to “CS” department and are written by author “Navathe”.
6. List all computer (Department=”CS”) that have been issued.
7. List all books which have a price less than 500 or purchased between “01/01/1999” and “01/01/2004”.
8. Alter (increase) the size of Title column in LibraryBooks
9. Change the Department of the book titled “CS” back to “Discrete Maths”.

**Assessment:** Solve the below queries

1. Increase the price of a book identified by its AccessionNumber by 10%.
2. Display the details of all the books whose price is less than 500

Create database ssdcLibrary identified by ssdc123;

Grant connect,resource,dba to ssdcLibrary;

**20.Create the tables by Identifying primary and foreign keys.**

SQL> create table LibraryBooks (accno number(4),title varchar(25),author varchar(10),dept varchar(5),

purchasedate date,price number(10,2),constraint c1 primary key(accno));

Table created.

SQL> create table IssuedBooks ( accno number(4),borrower varchar(10),constraint c2 foreign key(accno) references LibraryBooks(accno));

Table created.

**21. Insert at least 5 records in each table**

SQL> insert into LibraryBooks values(&accno,'&title','&author','&dept','&purchasedate',&price);

Enter value for accno: 1001

Enter value for title: DB System Concepts

Enter value for author: Korth

Enter value for dept: CS

Enter value for purchasedate: 1-Jan-2015

Enter value for price: 499

SQL> /

Enter value for accno: 1002

Enter value for title: Database System

Enter value for author: Navathe

Enter value for dept: CS

Enter value for purchasedate: 1-Jan-1998

Enter value for price: 300

SQL> /

Enter value for accno: 1003

Enter value for title: Discrete Maths

Enter value for author: SC Gupta

Enter value for dept: Maths

Enter value for purchasedate: 1-Nov-2003

Enter value for price: 510

SQL> /

Enter value for accno: 1004

Enter value for title: M.Statistics

Enter value for author: SC Gupta

Enter value for dept: Stats

Enter value for purchasedate: 1-Apr-2021

Enter value for price: 999

SQL> /

Enter value for accno: 1005

Enter value for title: C

Enter value for author: Bala

Enter value for dept: CS

Enter value for purchasedate: 1-Jan-2020

Enter value for price: 700

SQL> select \* from Librarybooks;

Create IssuedBooks (Accession number, Borrower)

SQL> insert into IssuedBooks values(&accno,'&borrower');

Enter value for accno: 1001

Enter value for borrower: Rahul

SQL> /

Enter value for accno: 1005

Enter value for borrower: Suman

SQL> select \* from IssuedBooks;

**22) Delete the record of book titled “Database System Concepts”.**

SQL> Delete from LibraryBooks where title='Database System Concepts';

0 rows deleted.

**23) Change the Department of the book titled “Discrete Maths” to “CS”.**

SQL> update LibraryBooks set dept='CS' where title='Discrete Maths';

**24) List all books that belong to “CS” department.**

SQL> select \* from LibraryBooks where dept='CS';

**25) List all books that belong to “CS” department and are written by author**

**“Navathe”.**

SQL> select \* from LibraryBooks where dept='CS' and author='Navathe';

**26) List all computer books (Department=”CS”) that have been issued.**

SQL> select \* from LibraryBooks where dept='CS' and accno in (select accno from IssuedBooks);

**27) List all books which have a price less than 500 or purchased between**

**“01/01/1999” and “01/01/2004"**

SQL> select \* from LibraryBooks where price<500 or purchasedate between '01-Jan-1999' and '01-Jan-2004';

SQL> commit;

Commit complete.

**28. Alter (increase) the size of Title column in LibraryBooks**

**Desc LibraryBooks;**

Alter table Librarybooks

Modify(title number(25));

**29. Change the Department of the book titled “CS” back to “Discrete Maths”.**

Select \*from LibraryBooks;

Update LibraryBooks set title =’ **Discrete Maths’ where title=’CS’;**

**Create a database having three tables to store the details of students of Computer**

**Department in your college.**

1.Personal information about Student (College roll number, Name of student, Date of

birth, Address, Marks(rounded off to whole number) in percentage at 10 + 2, Phone

number). **College Roll Number is the primary key**

2.Paper Details (Paper code, Name of the Paper). **Paper Code is the Primary Key**

3.Student’s Academic and Attendance details (College roll number, Paper Code,

Attendance, Marks in home examination**). College Roll No is a Foreign key & PaperCode is**

**a Foreign Key.**

1. Identify primary and foreign keys and create the tables
2. Insert at least 5 records in each table.
3. Design a query that will return the records (from the second table) along with the name of student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper2.
4. List all students who live in “Warangal” and have marks greater than 60 in paper1.
5. Find the total attendance and total marks obtained by each student.

**31. Identify primary and foreign keys and create the tables**

create table **StudentInformation**

(RollNo number(12),name varchar(10),dob date,address varchar(15),marks\_plus2 number(3,0),

phno number(10),constraint c3 primary key(rollno));

create table **PaperDetails**

(papercode varchar(6),papername varchar(15),constraint c4 primary key(papercode));

create table **AcadAttendance**(RollNo number(12),papercode varchar(6),attendance number(5,2),marks number(5,2),constraint c5 foreign key(RollNo) references StudentInformation(RollNo),constraint c6 foreign key(papercode) references PaperDetails(papercode));

**32) Insert at least 5 records in each table.**

records in each table.

SQL> insert into StudentInformation values(&RollNo,'&name','&dob','&address',&marks\_plus2,&phno);

Enter value for rollno: 5001

Enter value for name: ABC

Enter value for dob: 1-Jan-2000

Enter value for address: Hyderabad

Enter value for marks\_plus2: 76

Enter value for phno: 9988776655

insert into StudentInformation values(&RollNo,'&name','&dob','&address',&marks\_plus2,&phno)

insert into StudentInformation values(5001,'ABC','1-Jan2000','Hyderabad',76,9988776655)

1 row created.

SQL> /

Enter value for rollno: 5002

Enter value for name: DEF

Enter value for dob: 2-Feb-2001

Enter value for address: Warangal

Enter value for marks\_plus2: 74

Enter value for phno: 8877665544

old 1: insert into StudentInformation

values(&RollNo,'&name','&dob','&address',&marks\_plus2,&phno)

new 1: insert into StudentInformation values(5002,'DEF','2-Feb2001','Warangal',74,8877665544)

1 row created.

SQL> /

Enter value for rollno: 5003

Enter value for name: xyz

Enter value for dob: 4-Mar-2002

Enter value for address: Khammam

Enter value for marks\_plus2: 75

Enter value for phno: 7894561234

insert into StudentInformation values(&RollNo,'&name','&dob','&address',&marks\_plus2,&phno)

insert into StudentInformation values(5003,'xyz','4-Mar2002','Khammam',75,7894561234)

1 row created.

SQL> /

Enter value for rollno: 5004

Enter value for name: pqr

Enter value for dob: 5-Apr-2001

Enter value for address: Kurnool

Enter value for marks\_plus2: 54

Enter value for phno: 8965478654

1 row created.

SQL> /

Enter value for rollno: 5005

Enter value for name: rahul

Enter value for dob: 6-Jun-2001

Enter value for address: Anantapur

Enter value for marks\_plus2: 87

Enter value for phno: 8965478965

SQL> insert into PaperDetails values('&papercode','&papername');

Enter value for papercode: Paper1

Enter value for papername: Prog with C

SQL> /

Enter value for papercode: Paper2

Enter value for papername: Prog with C++

old 1: insert into PaperDetails values('&papercode','&papername')

new 1: insert into PaperDetails values('Paper2','Prog with C++')

1 row created.

SQL> /

Enter value for papercode: Paper3

Enter value for papername: DataStructures

old 1: insert into PaperDetails values('&papercode','&papername')

new 1: insert into PaperDetails values('Paper3','DataStructures')

1 row created.

SQL> /

Enter value for papercode: Paper4

Enter value for papername: DBMS

SQL> /

Enter value for papercode: Paper5

Enter value for papername: Prog with Java

SQL> select \* from PaperDetails;

SQL> /

Enter value for rollno: 5001

Enter value for papercode: Paper1

Enter value for attendance: 60

Enter value for marks: 80

insert into AcadAttendance values(&RollNo,'&PaperCode',&attendance,&marks)

insert into AcadAttendance values(5001,'Paper1',60,80)

1 row created.

SQL> /

Enter value for rollno: 5002

Enter value for papercode: Paper2

Enter value for attendance: 78

Enter value for marks: 90

**33) Design a query that will return the records (from the second table) along with the name**

**of student from the first table, related to students who have more than 75% attendance**

**and more than 60% marks in paper2.**

select unique s.name, a.papercode, a.papername

from StudentInformation s, Paperdetails p, AcadAttendance a

where a.attendance>75 and a.marks>60 and a.papercode='paper2' and s.rollno=a.rollno;

select unique s.name, a.papercode

from StudentInformation s, Paperdetails p, AcadAttendance a

where a.attendance>75 and a.marks>60 and a.papercode='paper2' and s.rollno=a.rollno;

**34) List all students who live in “Warangal” and have marks greater than 60 in paper1.**

select unique s.name

from StudentInformation s, AcadAttendance a

where s.address='Warangal' and a.marks>60 and a.papercode='paper1';

**35) Find the total attendance and total marks obtained by each student.**

select sum(attendance), sum(marks)

from AcadAttendance

group by rollno;

**36) List the name of student who has got the highest marks in paper2**

select name, rollno

from StudentInformation

where rollno=(select rollno from AcadAttendance

where papercode='paper2' and marks=(select max(marks) from AcadAttendance

where papercode='paper2'));

**37.Get the attendance and marks for a specific student in a specific paper**

select rollno

from AcadAttendance

where papercode='paper2' and marks=(select max(marks) from AcadAttendance where

papercode='paper2');

**38.List all students enrolled in a specific paper**

select s.name, a.papercode, a.papername

from StudentInformation s, Paperdetails p,

where a.papercode='paper2'**;**

**39.Find the papers with average marks scored by students**

select \*

from StudentInformation

where papercode='paper2' and marks=(select avg(marks) from AcadAttendance

where papercode='paper2'));

**40.Update attendance by 5% for a specific student in a specific paper**

UPDATE AcadAttendance

SET attendance = attendance + 5

WHERE student\_id = 'student\_id' AND paper\_id = 'paper2';

**Week -9:**

Customer (CustID, email, Name, Phone, ReferrerID)

Bicycle (BicycleID, DatePurchased, Color, CustID, ModelNo)

BicycleModel(ModelNo, Manufacturer, Style)

Service (StartDate, BicycleID, EndDate)

----------------------------------------------------------------------------

In Customer table-> CustID is the primary key & ReferrerId is a foreign key

In Bicycle table -> BicycleID is the primary key , CustId is a foreign key & ModelNo is also

foreign key

In BicycleModel table -> ModelNo is the primary key.

In Service table -> BicycleId is a foreign key.

Create table in this order: (i), (iii), (ii), (iv)

**41) Identify primary and foreign keys. Create the tables and insert at least 5**

Create table Customer

(CustID varchar(4),email varchar(20),name varchar(15),phone number(10),ReferrerId varchar(4),

constraint c31 primary key(CustId));

insert into Customer values('&CustId','&email','&name',&phone,'&ReferrerId');

SQL> insert into Customer values('&CustId','&email','&name',&phone,'&ReferrerId');

Enter value for custid: C1

Enter value for email: abc@gmail.com

Enter value for name: ABC

Enter value for phone: 8899889988

Enter value for referrerid: NA

SQL> /

Enter value for custid: C2

Enter value for email: def@gmail.com

Enter value for name: DEF

Enter value for phone: 7788990099

Enter value for referrerid: C1

SQL> select \* from Customer;

Create table BicycleModel

(ModelNo number(4),Manufacturer varchar(10),Style varchar(10),constraint c32 primary key(ModelNo));

SQL> insert into BicycleModel values(&ModelNo,'&Manufacturer','&style');

Enter value for modelno: 2020

Enter value for manufacturer: Honda

Enter value for style: RoadBike

SQL> /

Enter value for modelno: 2021

Enter value for manufacturer: Hero

Enter value for style: ElectrBike

Create table Bicycle

( BicycleId number(2),DatePurchased date,color varchar(8),CustId varchar(4),ModelNo number(4),

constraint c33 primary key(BicycleId),

constraint c34 foreign key(CustId) references Customer(CustID),

constraint c35 foreign key(ModelNo) references BicycleModel(ModelNo)

);

SQL> insert into Bicycle values(&BicycleId,'&DataPurchased','&color','&CustId',&ModelNo);

Enter value for bicycleid: 21

Enter value for datapurchased: 4-jun-2021

Enter value for color: red

Enter value for custid: C1

Enter value for modelno: 2021

SQL> /

Enter value for bicycleid: 20

Enter value for datapurchased: 6-dec-2020

Enter value for color: black

Enter value for custid: C2

Enter value for modelno: 2020

SQL> select \* from Bicycle;

Create table Service

(startdate date,BicycleId number(2),

enddate date,

constraint c36 foreign key(BicycleId) references Bicycle(BicycleId)

);

SQL> insert into Service values('&startdate',&BicycleId,'&enddate');

Enter value for startdate: 15-jun-2021

Enter value for bicycleid: 21

Enter value for enddate: 30-jun-2021

SQL> /

Enter value for startdate: 1-jan-2021

Enter value for bicycleid: 20

Enter value for enddate: 15-jan-2021

SQL> select \* from Bicycle;

SQL> select \* from Customer;

SQL> select \* from BicycleModel;

SQL> select \* from Bicycle;

SQL> select \* from service;

**43) List all the customers who have the bicycles manufactured by manufacturer “Honda”.**

select modelno

from BicycleModel

where manufacturer='Honda';

select custid

from Bicycle

where modelno=(select modelno from BicycleModel where manufacturer='Honda');

select name

from Customer

where custid=(select custid from Bicycle where modelno=(select modelno from

BicycleModel where manufacturer='Honda'));

**44) List the bicycles purchased by the customers who have been referred by Customer “C1”.**

select Custid from Customer

where ReferrerID='C1';

select \* from Bicycle

where custid=(select Custid from Customer where ReferrerID='C1');

SQL> select \* from Bicycle

where custid=(select Custid from Customer where ReferrerID='C1');

**45) List the manufacturer of red colored bicycles.**

select ModelNo

from Bicycle where color='red';

from BicycleModel

where ModelNo=(select ModelNo from Bicycle where color='red');

**46) List the models of the bicycles given for service.**

select BicycleId

from Service;

select ModelNo

from Bicycle

where BicycleId = (select BicycleId from Service);

select ModelNo

from Bicycle

where BicycleId in (select BicycleId from Service);

**47. Display the names of customer in UPPER case.**

Select upper(name) from customer;

**48. Find all bicycles with their customer details**

**49.Find the total no.of bicycles each customer owns**

**50.Find all services and the bicycles being serviced, along with customer information.**

4. Write a PL/SQL Program to demonstrate Procedure.

Create or replace procedure addvalues(a in number,b in number, c out number)

is

Begin

c:=a+b;

end;

/

--------------------------------------------------

declare

x number;

y number;

z number;

begin

x:=&x;

y:=&y;

addvalues(x,y,z);

dbms\_output.put\_line('Sum of Two number='||z);

end;

/

5. Write a PL/SQL Program to demonstrate Function.

Create or replace function addnum(a number,b number)

return number

is

c number;

Begin

c:=a+b;

return(c);

end;

/

------------------------------------------------------------

declare

x number;

y number;

z number;

begin

x:=&x;

y:=&y;

z:=addnum(x,y);

dbms\_output.put\_line('Sum of Two number='||z);

end;

/

Output:

6. Write a PL/SQL program to Handle Exceptions.

declare

employee emp%rowtype;

employeeno emp.empno%type;

begin

employeeno:=&employeeno;

select \* into employee from emp where empno=employeeno;

dbms\_output.put\_line('employee number is '||employee.empno);

dbms\_output.put\_line('employee name is '||employee.ename);

dbms\_output.put\_line('employee job is '||employee.job);

dbms\_output.put\_line('employee sal is '||employee.sal);

Exception

when NO\_DATA\_FOUND then

dbms\_output.put\_line('No such employee found');

end;

/