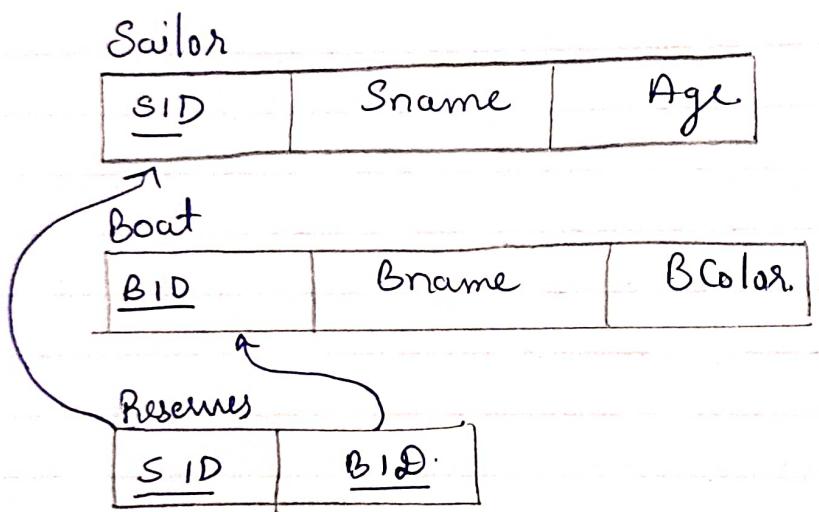


c) Schema Diagram



Exercise - III

Consider the relations BOAT, SAILOR and RESERVES. The relation BOAT identifies the features of a boat such as unique identifier, color, and a name. The list of sailors with attributes such as SailorID, name, age etc., are stored in the relation SAILOR. The sailors are allowed to reserve any number of boats on any day of the week and the records are to be updated in the RESERVES table.

- a) Mention the constraints neatly.
- b) Design the ER diagram for the ER diagram.
- c) State the schema diagram for the ER diagram.
- d) Create the tables, insert suitable tuples and perform the following operations in

S Q1 :

1. Obtain the details of the boats reserved by #Sailor\_Name.
2. Retrieve the BID of the boats reserved necessarily by all the sailors.
3. Find the number of boats reserved by each sailor. Display the Sailor\_Name along with the number of boats reserved.
4. Identify which boats have same name as their sailors.

Output:

⇒ SELECT \* FROM Boat;

BID	Bname	B Color
01	Vikrant	Blue
02	Maria	Black
03	Santa	White
04	Cholan	Red
05	Kalan	Black

⇒ SELECT \* FROM Sailor;

SID	Sname	Age
21	Rajan	25
22	Christopher	55
23	Sarfrazay	45
24	Shivan	35
25	Mosse	35
26	Vikrant	40

⇒ SELECT \* FROM reserves;

SID	BID
01	21
03	21
05	21
01	22
01	23
02	23
01	24
04	24
03	24
01	25

D

```
CREATE TABLE BOAT (
    BID number(3) PRIMARY KEY,
    Bname varchar(10),
    Bcolor varchar(10)
);
```

```
CREATE TABLE Sailor(
    SID number(3) PRIMARY KEY
    Sname varchar(10),
    age number(2)
);
```

```
CREATE TABLE RESERVES(
    BID number(3) references Boat(BID) delete on cascade,
    SID number(3) references Sailor(SID) delete on cascade,
);
```

INSERT ALL

```
INTO Boat VALUES(01, 'Vikrant', 'Blue')
INTO Boat VALUES(02, 'Melia', 'Black')
INTO Boat VALUES(03, 'Santa', 'White')
INTO Boat VALUES(04, 'Cholan', 'Red')
INTO Boat VALUES(05, 'Kalan', 'Black')
SELECT * FROM DUAL;
```

INSERT ALL

```
INTO Sailor VALUES(21, 'Rajan', 25)
INTO Sailor VALUES(22, 'Christopher', 55)
INTO Sailor VALUES(23, 'Surfay', 45)
```

d)

i)

<u>BID</u>	<u>Bname</u>	<u>Bcolor</u>
01	Vikrant	Blue
03	Santa	White
05	Kalan	Black.

ii).

<u>BID</u>	<u>Bname</u>	<u>BColor</u>
01	Vikrant	Blue.

iii)

<u>Sailor.Sname</u>	<u>Count(BID)</u>
Christopher	1
Bogie Morse	2
Rajan.	3
Sarfaraj	2.
Shivam	3

iv)

<u>BID</u>	<u>Bname</u>	<u>Bcolor</u>
01	Vikrant	Blue.

```

    INTO Sailor    VALUES(24, 'Shivan', 35)
    INTO Sailor    VALUES(25, 'Molse', 35)
    INTO Sailor    VALUES(26, 'Vikram', 40) Select * FROM dual;
    INSERT ALL
    INTO RESERVES  VALUES(01, 21)
    INTO Reserves   VALUES(03, 21)
    INTO Reserves   VALUES(05, 21)
    INTO Reserves   VALUES(07, 22)
    INTO Reserves   VALUES(01, 23)
    INTO Reserves   VALUES(02, 23)
    INTO Reserves   VALUES(01, 24)
    INTO Reserves   VALUES(04, 24)
    INTO Reserves   VALUES(03, 24)
    INTO Reserves   VALUES(01, 25)
    INTO Reserves   VALUES(02, 25)
    SELECT * FROM DUAL;

```

(i)

```

SELECT Boat.BID, Bname, Bcolor
FROM Boat, Sailor, reserves
WHERE Boat.BID = Sailor.SID AND Reserves.BID
AND Reserves.SID = Sailor.SID
AND Sailor.Sname = 'Rajan';

```

(ii)

```

SELECT *
FROM boat
WHERE NOT EXISTS(
    SELECT SID
    FROM SAILOR
    MINUS
    SELECT SID FROM reserves
    WHERE Boat.BID = reserves.bid);

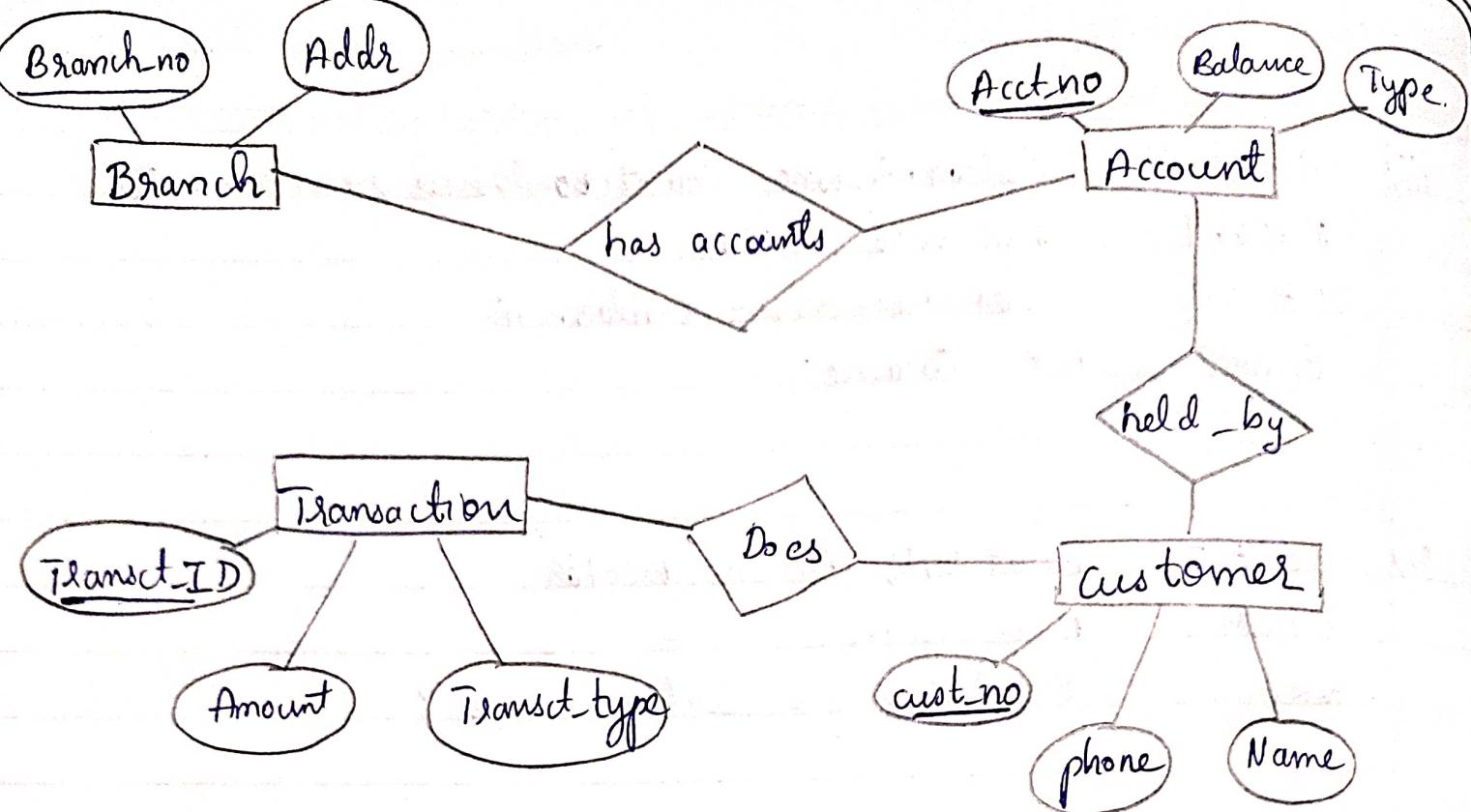
```

Experiment No. ....

Date : .....

iii) SELECT Sailor.Sname, Count(Boat.BID) reserves.BID  
FROM reserves, Sailor  
WHERE reserves.SID = Sailor.SID  
GROUP BY Sname;

N) SELECT Boat.BID, Bname, Bcolor  
FROM Boat, sailor  
WHERE Boat.Bname = Sailor.Sname ;



Branch

Branch-no

Addr.

Account

Acct-no

Balance

Type.

Customer

cust-no

phone

name.

Acctno

Branch-no

Transaction

Transact-ID

Amount

Acctno

transct-type

EXERCISE - IV

Consider the Banking database - CUSTOMER, BRANCH, ACCOUNT, AND TRANSACTION. An account can be a savings account or a current account. Customer can have both types of accounts. The transactions can be a deposit or a withdrawal. Mention constraints neatly.

a) Design the ER diagram for problem statement

b) State the schema diagram for ER diagram

c) Create the above tables, insert suitable types & perform the following operations in SQL

1. Obtain the details of customers who ~~not~~ have both Savings and Current Account

2. Retrieve the details of branches & number of accounts in each branch.

3. Obtain the details of customers who have performed at least 3 transactions.

4. List the details of branches whose number of accounts is less than average number of accounts in all branches

c)

~~ds.~~ create BranchF

Create Table BranchC

~~ds.~~ Branchno number(5) PRIMARY KEY,

Addr varchar(50),

;

Create Table AccountC

Acct-no number(12) PRIMARY KEY

Balance number(15),

Type varchar(20));

Output:

⇒ Select \* from Branch;

Branch-no	Addr
01	Bengaluru
02	Mysuru
03	Kodagu
04	Mandya
05	Hassan

⇒ Select \* from Customer;

Cust_no	Name	phone	Branch-no	Acct_no
101	Raj	1234567891	01	58901
101	Raj	1234567891	01	66901
102	Kriti	9999999999	02	56902
103	Karan	8888888888	02	56903
104	Manoj	7777777777	03	56904
105	Ram	6666666666	04	56905
106	Rahul	5555555555	05	56906
106	Rahul	5555555555	05	66902

⇒ Select \* from Account;

Acct_no	Balance	Acct_type
56901	10000	Savings
66901	50000	current
56902	12000	Savings
56903	17000	Savings
56904	25000	Savings
56905	160000	Savings
56906	30000	Savings
66902	75000	current

Create table Acc

Create table Customer(

Cust-no number(10), PRIMARY KEY,

Name varchar(20),

Phone number(10)

Branch-no number(5) references Branch(Branch-no) on  
delete cascade,

Account-no number(12) references Account(Acct-no) on  
delete cascade

);

Create table Transaction(

Account-no number(12) references Account(Acct-no) on  
delete cascade,

Amount number(15),

Trans-type varchar(20),

Transct-ID number(20) PRIMARY KEY,

);

INSERT ALL

INTO Branch Values(01, 'Bengaluru')

INTO Branch Values(02, 'Mysuru')

INTO Branch Values(03, 'Kodagu')

INTO Branch Values(04, 'Mandiya')

INTO Branch Values(05, 'Hassan')

Select \* From dual;

⇒ Select \* from transaction;

Acct_no	Amount	Transct type	Transct_id.
56901	500 0.	Deposit	30001
56901	300 0	withdraw	30002
56901	2000	deposit	30003
56902	3000	Withdraw	30004
56903	3000	deposit	30005
56904	4000	deposit	30006
56904	3500	withdraw	30007

d)

i)	cust_no	Name	phone	Branch_no	Ans
	101	Raj	1234567891	01	
	106	Rahul	5555555555	05	

ii)	Branch_no	Addr	Count (Acc_no)
	01	Bengaluru	2
	02	Mysuru	2
	03	Kodagu	1
	04	Mandy	1
	05	Hassan	2

iii)	Cust_no & Name	phone	Branch_no
	101 Raj	1234567891	01

## INSERT ALL

```

INTO Customer Values(101, 'Raj', 123456789, 01, 56901)
INTO Customer Values(101, 'Raj', 12356789, 01, 56901)
INTO Customer Values(102, 'Kriti', 9999999999, 02, 56902)
INTO Customer Values(103, 'Karan', 8888888888, 02, 56903)
INTO Customer Values(104, 'Manoj', 7777777777, 03, 56904)
INTO Customer Values(105, 'Ram', 6666666666, 04, 56905)
INTO Customer Values(106, 'Babul', 5555555555, 05, 56906)
INTO Customer Values(106, 'Rahul', 5555555555, 05, 66902)

```

Select \* from dual;

## INSERT ALL

```

INTO Customer Account Values( 56901, 10000, 'Savings')
INTO Account Values( 66901, 50000, 'Current')
INTO Account Values( 56902, 12000, 'Savings')
INTO Account Values( 56903, 17000, 'Savings')
INTO Account Values( 56904, 25000, 'Savings')
INTO Account Values( 56905, 160000, 'Savings')
INTO Account Values( 56906, 30000, 'Savings')
INTO Account Values( 56902, 75000, 'Current')

```

Select \* From dual;

## INSERT ALL

```

INTO Transaction Values(56901, 500, 'Deposit', 30001)
INTO Transaction Values(56901, 3000, 'Withdraw', 30002)
INTO Transaction Values(56901, 2000, 'Deposit', 30003)
INTO Transaction Values(56902, 3000, 'Withdraw', 30004)
INTO Transaction Values(56903, 3000, 'Deposit', 30005)
INTO Transaction Values(56904, 4000, 'Deposit', 30006)
INTO Transaction Values(56904, 3500, 'Withdraw', 30007)

```

iv).

<u>Branch no</u>	<u>Addr</u>
03	Kodagu
04	Mandyā

INTO Transaction Values (56905, 1500, 'withdraw', 30008)  
 Select \* From dual;

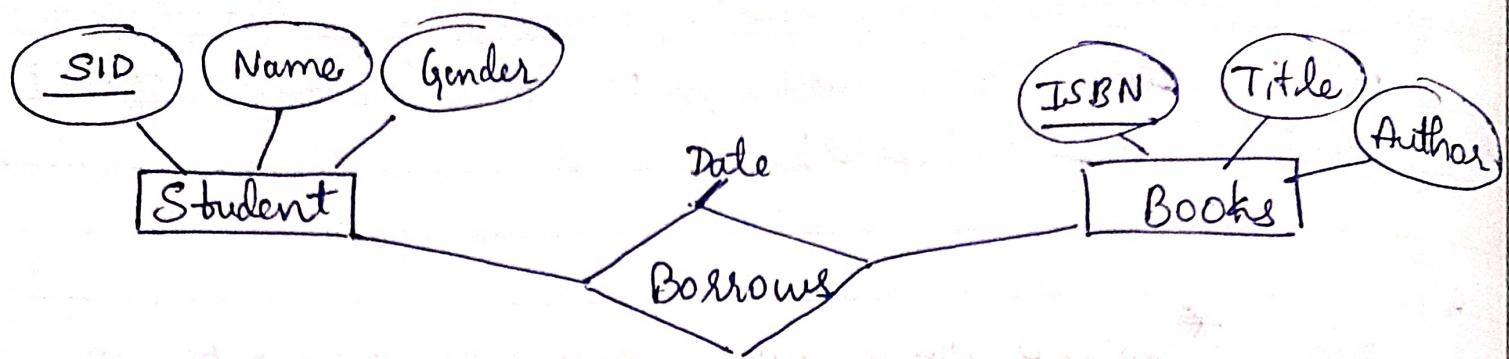
C

i) Select \*  
 From customer, account  
 where cust\_id in (select cust\_id from account  
 where acctype = 'S'  
 Intersect  
 select cust\_id from account  
 where acctype = 'C')  
 );

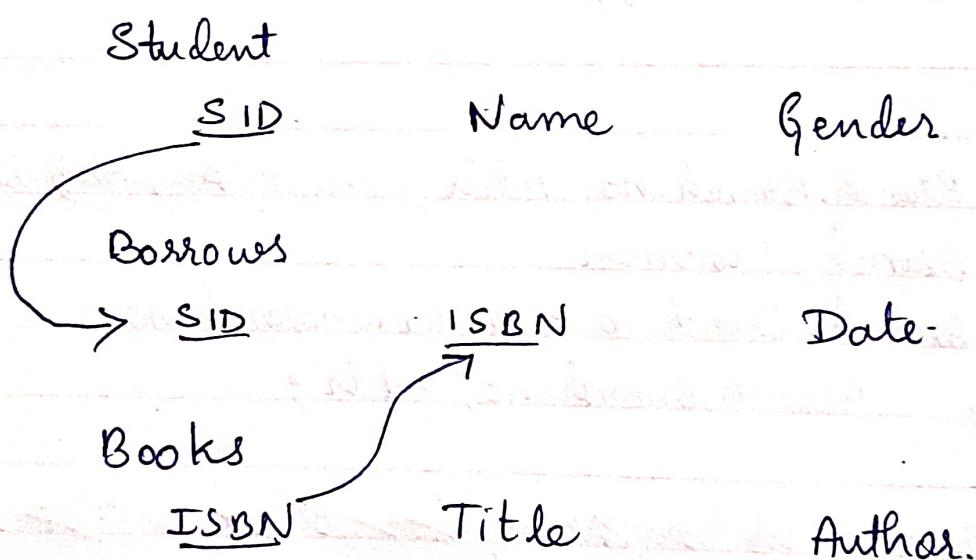
ii) Select Branch.Branch-no, Addr, count(Acc-no) Customer.Acc-no  
 from Branch, Customer  
 where Branch.Branch-no = Customer.Branch-no  
 group by Branch.Branch-no, Addr;

iii) Select Customer.cust\_no, Name, phone, Branch-no, count(Transaction.Transct-ID)  
 Select custno, Name, phone, count(Transaction.Transct-ID)  
 from Customer, Transaction  
 group by Transaction.Acc-no, name, phone  
 where Transaction.Acc-no = Customer.Acc-no  
 Having count(Transaction.Transct-ID) <= 3;

iv) Select branch.bno, addr, count(account.accno)  
 from branch, account  
 where branch.bno = account.bno  
 group by branch.bno, addr  
 having count(account.accno) < (select avg(count(accno)) from  
 account group by bno);



## Q) Schema



Exercise - V.

Consider the Book Lending System from the library - Books, Student, Borrow. The students are allowed to borrow any number of books on a given date from the library. The details of the book should include ISBN, Title of the Book, author and publisher. All students need not compulsorily borrow books.

- a) Mention the constraints neatly
- b) Design the ER diagram for problem statement
- c) State schema diagram for ER diagram.
- d) Create above tables, insert suitable types and perform the following operations in SQL:
  1. Obtain names of student who has borrowed either book bearing ISBN '123' or ISBN '124'
  2. obtain Names of female students who have borrowed "Database" books
  3. Find number of books borrowed by each student. Display student details along with number of books
  4. list the books that begin with the letters "DA" and has never been borrowed by any student

Output:  $\Rightarrow$  Select \* from Books;

ISBN	Title	Author
001	Database	A1
002	DataScience	A2
003	Python	A3
004	Java	A4
005	C++	A5

$\Rightarrow$  Select \* from Student;

SID	Name	gender
101	Meredith	F
102	Taylor	F
103	Kamlesh	M
104	Klaman	M
105	Benjamin	M

$\Rightarrow$  Select \* from Borrowers;

SID	ISBN	Date
101	001	1/2/22
101	003	1/2/22
102	001	16/3/22
102	004	3/6/22
103	005	7/7/22
104	001	4/4/22
105	003	3/4/22

d) Create table Books (

ISBN number(10), primary key,

Title varchar(10),

Author varchar(10)

);

Create table Student (

sid number(10) primary key,

name varchar(20),

gender varchar(3)

);

Create table Borrows (

sid number(10) references Student (sid)

on delete cascade,

ISBN number(10) references Books (ISBN)

on delete cascade,

Date varchar(10),

primary key (sid, ISBN)

);

INSERT ALL

Into Books Values (001, 'Database', 'A1')

Into Books Values (002, 'DataMining', 'A2')

Into Books Values (003, 'Python', 'A3')

Into Books Values (004, 'Java', 'A4')

Into Books Values (005, 'C++', 'A5')

Select \* from dual;

d)

i)

<u>SID</u>	<u>Name</u>
101	Meredith.
102	Taylor
105	Benjamin.

ii)

<u>SID</u>	<u>Name</u>
101	Meredith
102	Taylor.
104	Kraman.

iii)

<u>SID</u>	<u>Name</u>	<u>gender</u>	<u>Count(borrows, isbn)</u>
101	Meredith	F	2
102	Taylor	F	2
103	Kamlesh	M	1
104	Kraman	M	1
105	Benjamin	M	1

iv)

ISBN

002.

INSERT ALL

Into Student Values(101, 'Meredith', 'F')  
 Into Student Values(102, 'Taylos', 'F')  
 Into Student Values(103, 'Kamalesh', 'M')  
 Into Student Values(104, 'Kraman', 'M')  
 Into Student Values(105, 'Benjamin', 'M')  
 Select \* from dual;

INSERT ALL

INTO Borrows Values(101, 001, '1/2/22')  
 INTO Borrows Values(101, 003, '1/2/22')  
 Into Borrows Values(102, 001, '16/3/22')  
 Into Borrows Values(102, 004, '3/6/22')  
 Into Borrows Values(103, 005, '7/7/22')  
 Into Borrows Values(104, 001, '4/4/22')  
 Into Borrows Values(105, 003, '3/4/22')

d)

i) Select student.sid, name  
 from student, borrows  
 where student.sid = borrows.sid  
 and borrows.isbn in (003, 004)

ii) Select student.sid, name.  
 from student, borrows, books  
 where student.sid = borrows.sid,  
 and borrows.isbn = books.isbn  
 and student.gender = 'F'  
 and books.title = 'Database'

Experiment No.: ..... Date : .....

iii) Select student.sid, name, gender, count(borrows.isbn)  
from student, borrows  
where student.sid = Borrows.sid  
group by student.sid, name, gender;

iv) Select isbn  
from books  
where upper(title) like 'DA%'  
Minus  
select isbn  
from borrows;

Output: Select \* from employee;

ssn	fname	Salary	ph-no	DeptNo
123	Krishna	30000	966665555	3
124	Kristen	30000	977776666	3
125	Harish	45000	877775555	1
126	Kashmi	20000	866665555	2
127	Haus	45000	466665555	1
128	Kranti	20000	777773277	2

SQL># Set serveroutput on;

SQL> @file1.sql  
plsql execution completed successfully

Number of employees awarded hike are : 2.

SQL> Select ssn, salary,dept.no from employee

ssn	Salary	DeptNo
123	30000	3
124	30000	3
125	45000	1
126	23000	2
127	45000	1
128	23000	2

PLSQL

Q) Write a program that gives all employees  
whose e in Department #number a 15% pay  
increase. Display a message displaying how  
many employees were awarded the increase.  
→ edit file1.sql

Begin.

```
update employee
set salary = salary + (0.15*salary)
where dept_no = 2;
dbms_output.put_line ('Number of employees awarded
hike are: '||sql%rowcount);
end;
```

⇒ Select \* from Supply;

P ID	S ID	quantity
37	4	15
39	4	
32	1	3
39	3	6
34	3	3
33	2	
39	1	10
36	2	10
38	2	5
36	3	10
39	3	5
35	2	3
37	1	5

SQL> set serveroutput on;

SQL> @ copy.sql  
Execution completed successfully.

SQL> select \* from Shipment;

P ID	S ID	quantity
32	1	3.
34	3	3.
33	2	5

2) Write a PL/SQL program to copy contents of Shipment table to another table for maintaining for specific part number.

SQL> Create table Shipment as  
 (Select \*  
 from supply  
 where 1 = 2);

SQL> edit copy.sql

Declare

cursor c1 is select \* from supply;  
v\_rec employee%rowtype;

Begin

open c1;

loop

fetch c1 into v\_rec;

exit when c1%notfound;

insert into chapter 8pp Shipment values(v\_rec.pid,  
v\_rec.sid, v\_rec.quantity);

where v\_rec.pid in (31, 32, 33, 34);

end loop;

close c1;

end;

/

Output:

```
SQL> set serveroutput on;
SQL> &@ prime-sql.
Enter the value of n: 4
4 is not a prime number.
```

3) Write a PL/SQL program to check for a given number is prime or not.

SQL> cd\it prime.sql.

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 no.');

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;

/

⇒ Select \* from Branch

<u>Branch-no</u>	<u>Addr</u>
01	Bengaluru
02	Mysuru
03	Kodagu
04	Mandya
05	Hassan

SQL> set serveroutput on;

SQL> @ copy Branch.sql  
Execution completed successfully.

SQL> select \* from Branch\_Copy;

<u>Branch-no</u>	<u>Addr</u>
01	Bengaluru
02	Mysuru
03	Kodagu
04	Mandya
05	Hassan

4) Using cursors demonstrate the process of copying the contents of one table to a new table.

SQL> Create table BranchCopy as  
(Select \*  
from Branch where 1=2);

SQL> edit copyBranch.sql

Declare

cursor c1 is select \* from Branch;  
v\_rec Branch%rowtype;

Begin.

open c1;

loop

fetch c1 into v\_rec;

exit when c1%notfound;

insert into BranchCopy Values (v\_rec.Branch\_no,  
v\_rec.Addr);

end loop;

close c1;

end;

/

Output :

SQL> set serveroutput on;

SQL> @ fb.sql.

Fibonacci series

0

1

1

2

3

5

8

13.

- 5) Write PL/SQL procedure to print first 8 Fibonacci numbers and  
5 a program to call the same

SQL> edit fib.sql

Declare

```
a    number := 0;  
b    number := 1;  
c    number := 0;  
counter    number := 2;
```

procedure fibonacci is

begin

while (counter < 8)

loop

c := a+b;

dbms\_output.put\_line(c);

a := b;

b := c;

counter := counter + 1;

end loop;

end;

begin

dbms\_output.put\_line(a);

dbms\_output.put\_line(b);

fibonacci;

end;

/