Experiment No. : 5 Date : 14/05/2025

Consider the Banking database - Customer, Branch, Account and Transcation. An account can be a Savings account or a current account. Customer can have both types of accounts. the transcations can be a deposit or a withdrawal Mention the constraints neatly.

a) Design the ER diagram for the problem Statement

b) State the Schema diagram for the ER diagram

c) Create the above tables, insert suitable tuples and perform the following operation in SQL:

i. obtain the details of customer who have both savings and current Account

ii Retrive the details of branches and the number of accounts in each branch

iii Obtain the details of customers who have ~~both~~ performed at least 3 transcaction.

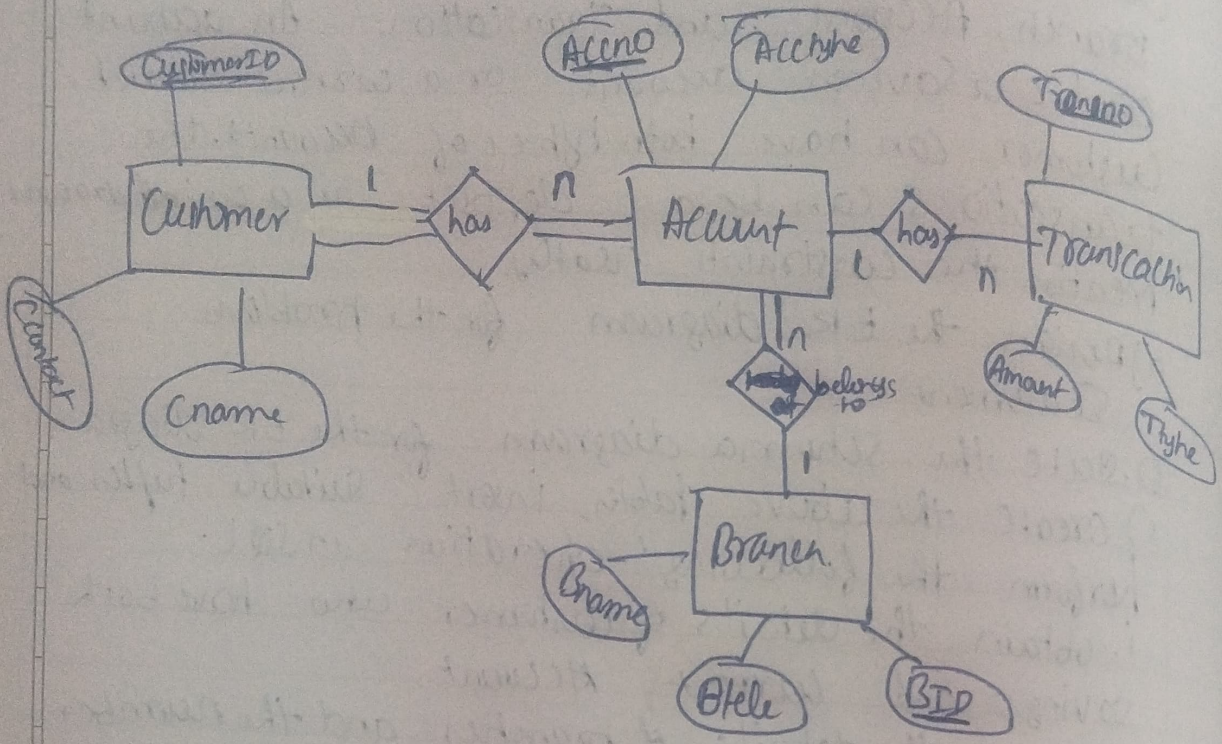
iv. List the details of branches where the number of accounts is less than the average number of account in all branches.

d. Create the table, Insert suitable tuples, and perform the following operations using MongoDB

i Find the branch name for a given Branch_ID

ii List the total number of accounts for each Customer

(a)



(b)

Customer

CustomerID	CContact	Cname

Account

Accno	Acctype	Bid	CustomerID

Branch

Bid	Bname	Btele

Transcation

TransID	Accno	Amount	Ttype

Experiment No. Date

(C)

Create table Customer (CustomerID integer primary key, Cname varchar(20), Contact varchar(13));

create table Branch (Bname varchar(30), Bid integer primary key, Bcity varchar(20));

Create table account (Accno integer Primary key, Acctype varchar(20), CustomerID integer, Foreign key(CustomerID) References Customer(CustomerID), Bid integer, Foreign key(Bid) references Branch(Bid));

Create table transaction (TranNo integer Primary key, Accno Integer, Amount Float, Ttype varchar(20), foreign key(Accno) References account(Accno));

Insert values

Insert into Customer values(&Customerid, &name, &Contactno);

Insert into Branch values(&Bname, &Bid, &Bcity);

Insert into Account values(&Accno, &Acctype, &Bid, &CustomerID);

Insert into Transaction value(&Tranno, &Accno, &Amount, &Ttype);

Select * from Customer;
Select * from Branch;
Select * from Account;
Select * from transaction;

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

(a) The entities are
CUSTOMER
BRANCH
ACCOUNT
TRANSACTION

The attributes of customer are
CustomerID, Cname, Ccontact
The attributes of Branch are
Bname, BID, Bcity
The attributes of Account are
ACCno, Acctype
The attributes of Transaction are
Tranno, Amount, Ttype

Primary key
CustomerID in Customer
Bid in Branch
ACCno in ACCount
Transactionno i.e Tranno in transaction.

foreign key

CustomerID in Account references CustomerID in Customer
Bid in Account references Bid in Branch

ACCno in Transaction references ACCno in ACCount.

Cusnmerid	Cname	Ceontact
1	Alice	9467634377
2	bob	7677614488
3	Charlie	7164832167
4	Dina	8164321606
5	Etna	7163248366
6	Fiona	7164178666
7	Geurse	7164787378
8	Hanna	7782481478
9	Ian	7283632582
10	Julia	7787723699

Bname	Bid	bkle
MG Rood	101	1234567
Indiranagar	102	324767
White field	103	212323
Koramangala	104	476232
Jayanagar	105	123456

ACCNO	Acctyle	CusnmerID	Bid
1001	Savings	1	101
1002	Current	1	101
1003	savings	2	102
1004	Current	2	102
1005	Savings	3	103
1006	Current	3	103
1007	Saving	4	104
1008	savings	5	105
1009	Current	6	101
1010	Savings	7	102
1011	Current	9	103
1012	saving	10	104
1013	Current		105

Tranno	ACCNO	Amount	Htyke
2001	1001	101	Deposit
2002	1001	191	withdrawal
2009	1006	103	Deposit
2010	1007	104	Deposit
2011	1007	104	withdrawal
2012	1007	101	Deposit
2013	1004	101	deposit
2016	1009	101	withdrawal
2017	1007		Deposite

Experiment No. : Date :

(i)

Select * from customer where customerID in (select customerid from account where acctype = 'current' intersect select customerid from account where acctype = 'Savings');

(ii) Select b.bname, b.bid, b.btele, count(a.accno) as Accountcount from branch b Join Accant a ON a.bid = b.bid Group by b.bname, b.bid, b.btele;

(iii)

Select * from customer where customerid in (Select *customerid from account where accno in (select accno from transcation group by accno having count(tranno) >=3));

(v) Select * from branch where bid in (select bid from account group by bid having count(accno) <(select avg (count(accno)) from account group by bid));

Output of query

①

CustomerID	Cname	Ccontact
1	Alice	9467634377
2	Bob	7677614484
3	charlie	7164832167

②

Bname	Bid	Bkle	Accountcount
Jayanagar	105	634567	2
Indiranagar	102	324767	3
MG road	101	1234567	3
Koramangala	104	476232	2
whitefield	103	212323	3

③

CustomerID	Cname	Ccontact
4	Diana	8164321606
6	Fiona	7164778666

④

Bname	Bid	Bkle
Koramangala	104	476732
Jayanagar	105	123456

(d)(i) {
"Bname": Jayanagar
}

(ii) {"_id": 4567, "total accounts": 2}

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

(d)

(i)

```
>db.createCollection("Bank");

>db.bank.insert(
        {"Bname:Jayanagar", "Bid:105", "Btele
        :123456"})

>db.bank.find(
        {bid:105},
        {_id:0, bname:1}
    )
```

(ii) db.createCollection("Account");

```
>db.Account.insert({
        "cid":1,
        "cname":ravi,
        "accno":100,
        "acctype":"Savings",
        "contact":"96476376"
        "bid":104
    });

db.account.aggregate(
    [
        {$group:{
            _id:"$cid",
            totalAccountH:{$sum:1}
        }
    ]);
```

①

```
{
    "_id" : ObjectId("6849094689901e9edd6cdd6"),
    "SSN" : 4561,
     "Name" :  "James"
     "DeptNO" : "XYZ",
      "ProjectNO" : 101
}
{
     "_id" : objectId("6849094789901e9ad6"),
      "SSN" : 3256,
      "Name" :  "Jack"
       "DeptNO" : XYZ,
       "ProjectNO" : 104
}
```

②

```
{ "_id" : ObjectId("6849094789901e9ad6"),
    "SSN" : 3256,
    "Name" :  "Jack",
      "DeptNO" : "XYZ"
       "ProjectNO" : 104
}
```

Experiment No. : 1 Date : 26/03/25

Experiment 1:

e) Create the table, insert suitable tuples & perform the following operations using MongoDB

i) List all the employees of Department named #Dept name.

ii) name the employees working in ~~the Department~~ on Project number: #Project-no

> db.createCollection("Employee");

> db.Employee.insert({"SSN: 4567, "Name": "James", "DeptNo: "XY", "ProjectNO: 101 });

(i)

db.Employee.find({"DeptNo": "XY"}).pretty()

(ii)

db.Employee.find({"ProjectNO": 104 }).pretty()

(4)

Output

writeResult({"nMatched": 1, "nUpserted": 0, "nModified": 1})

db.warehouse.find().pretty()

{

"_id" : ObjectId("68490a1289a010e9edd6cdd4"),

"Pno" : 1946

"Pname" : "bloth",

"Color" : "Black",

"Sno" : 1234,

"Sname" : "ABC"

}

Experiment No. : 2 Date : 09/04/25

E

e) Create the table, insert suitable tuples & perform the following operation using MongoDB

(i) Update the details of parts for a given identifier pid

(ii) Display all suppliers who supply the part with part identifier : pid

Inserting values

db.createCollection("warehouse")

```
db.warehouse.insert({
        "Pno": 1947
    "Paname": "bolts",
    "Color": "Black"
    "Sno": 1234
    "Sname": "ABC"});
```

(i)
```
db.warehouse.update({"Pno": 1947},
    {
      $set: {"Pno": 1946}
    },
    {multi: true});
```

db.warehouse.find().pretty

(ii)
```
db.warehouse.find({"Pno": 1946}).pretty();
```

Output

(i) {

"_id" : ObjectId("8760914b6e0gd6768g"),
"Bid" : 9987,
"Bname" : "Scare"
"Color" : "Black"
"Sname" : "Sucre"
"Sid" : 1234

}

(ii) {

"_id" : ObjectId("8904632106aed")
"Bid" : 9988,
"Bname" "ABC"
"Color" : "Black"
"Sname" : "John"
"Sid" : 1234

}

Experiment No. 03 Date : 07/05/2025

(e) Create the table, insert suitable tuples & perform the following operation using MongoDB

```
db.createCollection("Boatres")

db.boatres.insert ({
                    "Bid": 9988,
                    "Bname": "ABC",
                    "Color": "Black"
                     "Sname": "John"
                    "SID": 1234
                    })
```

(i) `db.boatres.find ({"Sname": "Lucre"}).count()`

(ii) `db.boatres.find ({"Color": "Black"}).pretty()`

Output:

```
Enter value for n: 49
old 2: n number := &n;
new 2: n number := 49;
49 is not prime number

PL/SQL procedure successfully completed
```

```
Enter value for n: 47
old 2: n number := &n;
new 2: n number := 47;
47 is a prime number

PL/SQL procedure successfully completed
```

Experiment No. :..... Date : 04/06/2025

6) Write a PL/SQL Program to check whether a given number is Prime or not

```
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;
/
```

Output

before

Employee

Name	Salary	DNO
John	30000	5
Franklin	40000	5
Alicia	24750	4
Jennifer	42570	4
Ramesh	38000	5
Joyce	28000	5
Ahmad	24780	4
James	55000	1

@Pl.sql

number of rows updated are 4

PL/SQL procedure successfully completed

after Employee table

Name	Salary	DNO
John	34800	5
Franklin	46000	5
Alicia	24780	4
Jennifer	42570	4
Ramesh	43700	5
Joyce	28 780	5
Ahmad	24780	4
James	55000	1

Experiment No. : 1 Date : 04/06/2025

B) Write a program that gives all employees in department #number a 15%. Pay increase Display a message displaying how many employees were awarded the increase.

```
Set ServerOutput on
begin
Update employee set salary=(1.15*salary) where deptno=10;
dbms_output.putline("number of rows updated are' ||sql%.rowcount);
end;
/
```

Output

3 rows inserted into the table copy_part1

Select * from copy_part1;

Pno	Pname	colour
10	nuts	black
20	bolts	gray
30	screw	green

Experiment No. : 2 Date : 04/06/2025

B) Write a PL/SQL Program to copy the contents of the Shipment table to another table for maintaining records for specific Part number.

```
Create table Part1 (Pno int, Pname char(20), color char(20),
                    Primary key(Pno));
Create table Copy-Part1 (Pno int, Pname char(20), Colour
Char(20), Primary key (Pno));

Insert into Part1 value (10, 'nuts', 'black');
Insert into Part1 values (20, 'bolts', 'grey');
Insert into Part1 values (20, '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;
insertinto copy-Part1 values(rows.Pno, rows.Pname, rows.Colour);
end loop;
Counter := curr%.row_count;
 Close curr;
dbms_output.putline(Counter || ' rows inserted into the table
                    Copy-Part1');

end
```

Output

3 rows inserted into the table copy_parts1

Pno	Pname	color
10	nuts	black
20	bolts	grey
30	screw	green

Experiment No. : 4 Date : 04/06/2023

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

```
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;
row part1%rowtype;
begin
open curr;
loop
fetch curr into rows;
exit when curr%notfound;

insert into copy_Part1 values (rows.pno, row.Pname, rows.Colour);

end loop;
Counter := curr%rowcount;
close curr;
dbms_output.put_line(Counter || 'rows inserted into the table copy_Part1');
end;
/
```

Experiment No. : 5 Date : 04/06/2025

Consider the Book lending System from the library -BOOKS, STUDENT, BORROWS. the students are allowed to borrow any number of books on a given date from the library. The details of the books 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 the problem Statement

c) State the Schema diagram for the ER diagram

d) Create the above tables, insert suitable tuples & perform the following Operations in SQL:

i. obtain the names of the student who has borrowed either book bearing ISBN '123' or ISBN '124'

ii. Obtain the names of the student who has borrowed either book bearing ISBN '123' or ISBN '124'

iii. Obtain the name of female students who have borrowed "Database" books

iii) Find the number of books borrowed by each student Display the student details along with the number of books

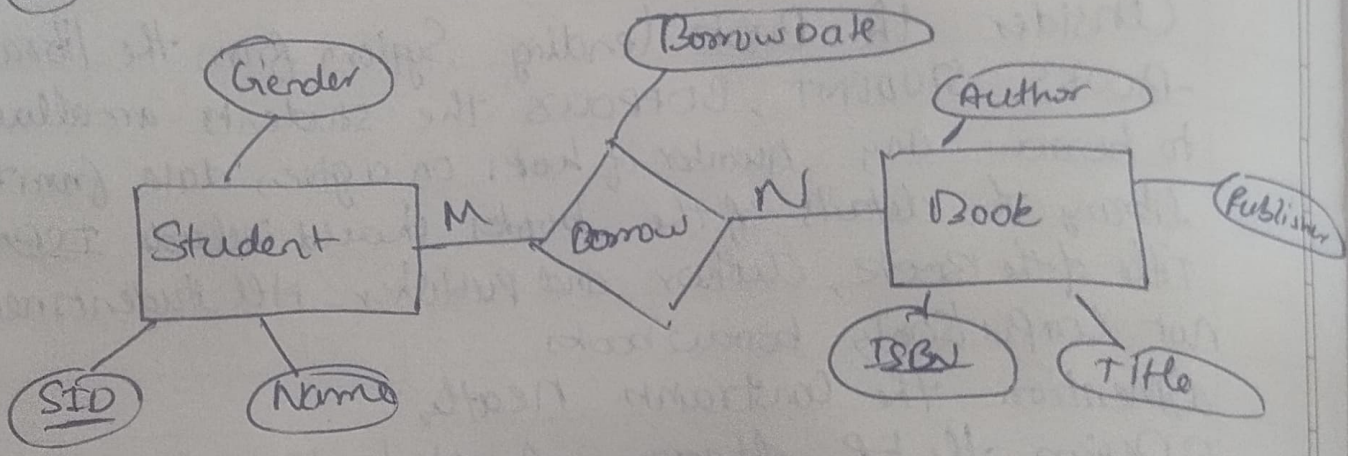
e) Create the table, insert suitable tuples & perform the following operation using MONGODB

i obtain the book details authored by "author-name"

ii Obtain the name of students who have borrowed "Database" books

f) Write a PL/SQL Procedure to print the first 8 Fibonacci numbers and a program to call a same.

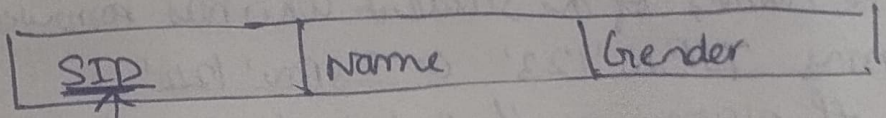
(b)



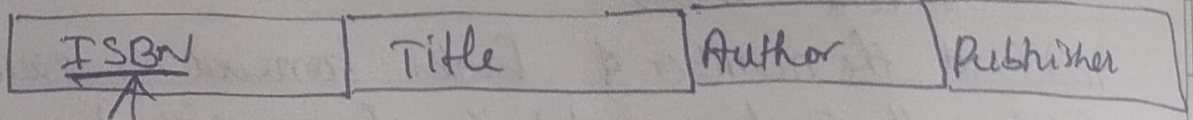
ER diagram

(c)

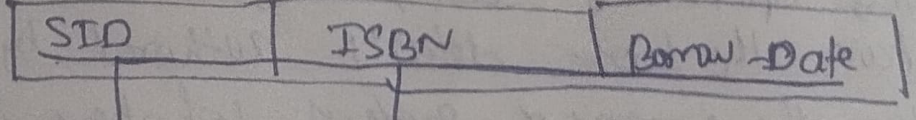
Student



Book



Borrow



Schema diagram

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

(a) Constraint

Book table
1) ISBN (Primary key)
2) Title
3) Author
4) Publisher

Student table
1) SID (primary key)
2) Name
3) Gender

Borrow table
1) ISBN } (primary key)
2) Borrow Date }
3) SID

foreign key ISBN references ISBN of Book table
foreign key SID references SID of student table

Student table

SID	Name	Gender
1	Adithya	M
2	Shena	F
3	Rahul	M
4	Priya	F

Book table

ISBN	title	Author	Publisher
123	Database	Navathe	Pearson
124	OS	Silberschaz	Mcgrawhill
125	CN	tanenbaum	Pearson
126	Java	Herbert	Tata McGraw

Borrow table

SID	ISBN	Date of Borrow
1	123	01-06-25
2	124	01-06-25
2	123	08-10-25
3	125	09-10-25
4	123	10-10-25

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

(2) Create table Student (sid int Primary key, Name varchar(30), Gender varchar(10));

Create table Book (ISBN varchar(10) primary key, title varchar(20), Author varchar(20), publisher varchar(20));

Create table Borrow (ISBN varchar(10), SID int, Date_of_Borrow varchar(10), primary key (SID, ISBN, Date_of_Borrow), Foreign key (sid) references student(SID), Foreign key (ISBN) references Book (ISBN));

Insert into student values (&sid, 'fname', '&Gender');

Insert into Book values ('ISBN', 'title', 'Author', 'Publisher');

Insert into Borrow values ('ISBN', 'SID', 'Date_of_Borrow');

Select * from student;
Select * from Book;
Select * from Borrow;

(i)

Name
Adithya
Sneha
Priya

(ii)

Name
Sneha
Priya

(iii)

SID	Name	Count(B.ISBN)
1	Adithya	1
2	Sneha	2
3	Rahul	1
4	Priya	1

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

(i) Select Name from Student where SID in (Select SID from borrow where ISBN = '123' Union Select SID from borrow where ISBN = '124');

(ii) Select Distinct name from Student where Gender = 'F' and SID in (Select SID from borrow where ISBN in (Select ISBN from Book where title = 'Database'));

(iii) Select S.sid, S.name count(b.ISBN) from student S left Join Borrow b on s.SID = b.sid group by s.sid, S.name;

(c)

db.createCollection("books")
db.create Collection ("Students")

db.books.InsertMany([

{bid: 1, title: "Database", author: "Adithya"},
{bid: 2, title: "OS", author: "Hemanth"},
{bid: 3, title: "CN", author: "Harsendra"}
]);

db.~~books~~students.insertmany([
{sid: 101, name: "Alice", borrowed_books: "Database"},
{sid: 102, name: "Sahil", borrowed_book: "OS"}
]);

(i)

```
{  "_id" :  ObjectID ("bbb46f8c 5c6802f26124562"),
   "bid" : 2
   "title" : "OS"
   "author" : "Anish"
}
```

(ii)

```
[
  { "name_id" : ObjectID ("abc*5687fde*"),
    "sid" : 101
    "name" : " Alice"
    "borrowed_books : "database"
  }
]
```

Output

0
1
1
2
3
5
8
13

Experiment No. :

Date :

(i) db.books.find({author : "Anish"});

(ii) db.student.find({title: "Database"});

(8) Write a PL/SQL Procedure to print the first 8 fibonacci numbers & a programs to call the same

```
Set Server output on
declare
a number;
b number;
c number;
n number;
i number;
begin
n:=8;
a:=0; b:=1;
dbms_output.putline(a); dbms_output.putline(b);
for i in 1..n-2
loop
c:=a+b;
dbms_output.put_line(c);
a:=b;
b:=c;
end loop;
end;
/
```