Practical:-4

Design SQL queries for suitable database application using SQL DML statements: all types of Join, Sub-Query.

a)Branch (B_NO, B_name, B_city, assest)
Customer (C_No,C_Name,C_City,Street)
Loan(Loan_No,B_Name, amount)
Account (Acc_No, B_name, Balance)
Borrower (C_No, Loan_No)
Depositer (C_No,Acc_No)

```
mysql> CREATE TABLE Branch(B_No int,B_Name VARCHAR(20),B_City
VARCHAR(20),Assest VARCHAR(20));

mysql> Describe Branch;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| B_No   | int         | NO   | PRI | NULL    |       |
| B_Name | varchar(20) | YES  |     | NULL    |       |
| B_City | varchar(20) | YES  |     | NULL    |       |
| Assest | varchar(20) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

```
mysql> CREATE TABLE Customer2(C_No int PRIMARY KEY,C_Name VARCHAR(20),C_City
VARCHAR(20),Streat VARCHAR(20));

mysql> Describe Customer2;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| C_No   | int         | NO   | PRI | NULL    |       |
| C_Name | varchar(20) | YES  |     | NULL    |       |
| C_City | varchar(20) | YES  |     | NULL    |       |
| Streat | varchar(20) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

```
mysql> CREATE TABLE Loan1(Loan_No INT PRIMARY KEY,B_Name VARCHAR(20),Amount
int);
mysql> Describe Loan1;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Loan_No | int         | NO   | PRI | NULL    |       |
| B_Name  | varchar(20) | YES  |     | NULL    |       |
| Amount  | int         | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
```

```
mysql> CREATE TABLE Account1(Acc_No INT PRIMARY KEY,B_Name
VARCHAR(20),Balance INT);

mysql> Describe Account1;
```

```
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Acc_No  | int         | NO   | PRI | NULL    |       |
| B_Name  | varchar(20) | YES  |     | NULL    |       |
| Balance | int         | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+

mysql> CREATE TABLE Borrower1(C_No INT,FOREIGN KEY (C_No) REFERENCES
Customer2(C_No),Loan_No INT,FOREIGN KEY (Loan_No) REFERENCES Loan1(Loan_No));
mysql> Describe Borrower1;
+---------+------+------+-----+---------+-------+
| Field   | Type | Null | Key | Default | Extra |
+---------+------+------+-----+---------+-------+
| C_No    | int  | YES  | MUL | NULL    |       |
| Loan_No | int  | YES  | MUL | NULL    |       |
+---------+------+------+-----+---------+-------+


mysql> CREATE TABLE Depositor1(C_No INT,FOREIGN KEY (C_No) REFERENCES
Customer2(C_No),Acc_No INT,FOREIGN KEY (Acc_No) REFERENCES Account1(Acc_No));
mysql> Describe Depositor1;
+--------+------+------+-----+---------+-------+
| Field  | Type | Null | Key | Default | Extra |
+--------+------+------+-----+---------+-------+
| C_No   | int  | YES  | MUL | NULL    |       |
| Acc_No | int  | YES  | MUL | NULL    |       |
+--------+------+------+-----+---------+-------+


mysql> Select * From Branch;
+-------+----------+--------+-------------------+
| B_No  | B_Name   | B_City | Assest            |
+-------+----------+--------+-------------------+
| 11111 | Navipeth | Pune   | Security/Loans    |
| 22222 | Deccan   | Pune   | Loans             |
| 33333 | Swarget  | Pune   | Security/Reserves |
| 44444 | Thana    | Mumbai | Security          |
| 55555 | CST      | Mumbai | Loans             |
+-------+----------+--------+-------------------+


mysql> Select * From Customer2;
+------+--------+--------+-------------+
| C_No | C_Name | C_City | Streat      |
+------+--------+--------+-------------+
|  101 | Ram    | Pune   | Vimannager  |
|  102 | Sham   | Pune   | Hadapsar    |
|  103 | Veer   | Pune   | Sinhgadroad |
|  104 | Rakesh | Mumbai | Thana       |
|  105 | Jai    | Mumbai | CST         |
+------+--------+--------+-------------+


mysql> Select * From Loan1;
```

```
+---------+----------+--------+
| Loan_No | B_Name   | Amount |
+---------+----------+--------+
|       1 | Navipeth |   2000 |
|       2 | Lokmanya |   4000 |
|       3 | Manvel   |   6000 |
|       4 | Manvel   |  18000 |
|       5 | Manvel   |  21000 |
+---------+----------+--------+
```

```
mysql> Select * From Account1;
+--------+----------+---------+
| Acc_No | B_Name   | Balance |
+--------+----------+---------+
|    121 | Navipeth |   50000 |
|    122 | Lokmanya |   45000 |
|    124 | Manvel   |    6000 |
|    125 | Manvel   |    7000 |
|    126 | Manvel   |    3000 |
+--------+----------+---------+
```

```
mysql> Select * From Borrower1;
+------+---------+
| C_No | Loan_No |
+------+---------+
|  101 |       1 |
|  102 |       2 |
|  103 |       3 |
|  104 |       4 |
|  105 |       5 |
+------+---------+
```

```
mysql> Select * From Depositor1;
+------+--------+
| C_No | Acc_No |
+------+--------+
|  101 |    121 |
|  102 |    122 |
|  103 |    126 |
|  104 |    124 |
|  105 |    125 |
+------+--------+
```

1. Find loan data, ordered by decreasing amounts, then increasing loan numbers.

```
mysql> SELECT * FROM Loan1 ORDER BY Loan_No ASC,Amount DESC;
+---------+----------+--------+
| Loan_No | B_Name   | Amount |
+---------+----------+--------+
|       1 | Navipeth |   2000 |
|       2 | Lokmanya |   4000 |
|       3 | Manvel   |   6000 |
|       4 | Manvel   |  18000 |
|       5 | Manvel   |  21000 |
+---------+----------+--------+
```

```
mysql> SELECT * FROM Loan1 ORDER BY Amount DESC,Loan_No ASC;
```

```
+---------+----------+--------+
| Loan_No | B_Name   | Amount |
+---------+----------+--------+
|       5 | Manvel   |  21000 |
|       4 | Manvel   |  18000 |
|       3 | Manvel   |   6000 |
|       2 | Lokmanya |   4000 |
|       1 | Navipeth |   2000 |
+---------+----------+--------+
```

2. Find the pairs of names of different customers who live at the same address but have accounts at different branches.

```
mysql> SELECT DISTINCT C_Name FROM Customer2 WHERE C_City IN (SELECT B_City
FROM Branch WHERE B_Name IN (SELECT DISTINCT B_Name FROM Branch));
+---------+
| C_Name  |
+---------+
| Ram     |
| Sham    |
| Veer    |
| Rakesh  |
| Jai     |
| Pratap  |
| Vikrant |
+---------+
```

3. Find the names and address of customers who have loan for an amount exceeding 3 times their current balance.

```
mysql> SELECT * FROM Customer2 WHERE C_No IN (SELECT C_No FROM Borrower1
WHERE Loan_No NOT IN (SELECT Loan1.Loan_No FROM Loan1 INNER JOIN Account1 ON
Loan1.B_Name=Account1.B_Name WHERE Loan1.Amount>3*Account1.Balance));
+------+--------+--------+-------------+
| C_No | C_Name | C_City | Streat      |
+------+--------+--------+-------------+
|  101 | Ram    | Pune   | Vimannager  |
|  102 | Sham   | Pune   | Hadapsar    |
|  103 | Veer   | Pune   | Sinhgadroad |
+------+--------+--------+-------------+
```

b) Create the tables by using the following Schema:
Customer(cid, cname ,DOB ,city ,Bname)
Account(acno, cid, balance, actype)
Loan(loan_no, cid ,amount ,duration , roi)
Transaction(ac_no, dc , amt ,dot)
Employee(eid, ename, mgr_id ,doj, bname)
Branch(Bname, city)

1.)List customer details along with balance amount in ascending order ofbalance use natural join.

```
mysql> SELECt * FROm Customer1 Natural Join Account Order By balance ASC;
+-----+---------+------------+--------+------------+------+---------+-------------+
| cid | cname   | DOB        | city   | Bname      | acno | balance | actype      |
+-----+---------+------------+--------+------------+------+---------+-------------+
| 103 | Virat   | 2002-01-11 | Pune   | Navipeth   |  577 |   12000 | Fixedeposit |
| 102 | Rahul   | 2001-04-15 | Mumbai | Lokhandvala |  570 |   18000 | Fixedeposit |
| 104 | Kuldeep | 2002-11-11 | Noida  | Sector11   |  899 |   20000 | Current     |
| 101 | Shikhar | 2001-01-10 | Pune   | Navipeth   |  547 |   50000 | Savings     |
| 105 | Sehvag  | 2001-07-15 | Mumbai | Thana      |  541 |   50000 | Savings     |
+-----+---------+------------+--------+------------+------+---------+-------------+
```

2.)List customer details who belong to Bangalore along with account type they hold. Use inner join with on clause.

```
mysql> SELECT * FROm Customer1 INNER JOIN Account ON Customer1.cid=Account.cid WHERE
city=&apos;Bangluru&apos;;
+-----+-------+------------+----------+-------+------+------+---------+---------+
| cid | cname | DOB        | city     | Bname | acno | cid  | balance | actype  |
+-----+-------+------------+----------+-------+------+------+---------+---------+
| 106 | Kolhi | 2002-01-10 | Bangluru | Bang  |  544 |  106 |   45000 | Savings |
+-----+-------+------------+----------+-------+------+------+---------+---------
```

3.) List all customer names and their date of birth also display the loan details ifat all they have taken any loan.

```
mysql> SELECT cname,DOB FROM Customer1 WHERE cid IN (SELECT cid FROM Loan);
+---------+------------+
| cname   | DOB        |
+---------+------------+
| Shikhar | 2001-01-10 |
| Rahul   | 2001-04-15 |
| Virat   | 2002-01-11 |
| Kuldeep | 2002-11-11 |
| Sehvag  | 2001-07-15 |
+---------+------------+
```

4.) Show account number and type and transaction details on those accountalong with all transactions carried out at the bank.

```
mysql> SELECT Account.acno,Account.actype,Transaction.dc,Transaction.amt,Transaction.Date_Of_Transection FROM
Account INNER JOIN Transaction WHERE Account.acno=Transaction.acno;
+------+-------------+------+-------+---------------------+
| acno | actype      | dc   | amt   | Date_Of_Transection |
+------+-------------+------+-------+---------------------+
|  547 | Savings     | d    |  2000 | 2018-11-05          |
|  577 | Fixedeposit | c    | 10000 | 2018-10-04          |
|  899 | Current     | c    | 15000 | 2018-11-14          |
|  570 | Fixedeposit | d    | 10000 | 2018-09-04          |
+------+-------------+------+-------+---------------------+
```

5.) List customer details with his loan amount and duration use innner join andusing clause.

```
mysql> SELECt Customer1.*,Loan.amount FROM Customer1 INNER JOIN Loan WHERE Customer1.cid=Loan.cid;
+-----+---------+------------+--------+------------+--------+
| cid | cname   | DOB        | city   | Bname      | amount |
+-----+---------+------------+--------+------------+--------+
| 102 | Rahul   | 2001-04-15 | Mumbai | Lokhandvala | 200000 |
| 104 | Kuldeep | 2002-11-11 | Noida  | Sector11    | 150000 |
| 105 | Sehvag  | 2001-07-15 | Mumbai | Thana       | 150000 |
| 101 | Shikhar | 2001-01-10 | Pune   | Navipeth    | 600000 |
| 103 | Virat   | 2002-01-11 | Pune   | Navipeth    | 100000 |
+-----+---------+------------+--------+-------------+--------+
```

6.) List customer details who hold savings account using natural join

```
mysql> SELECt Customer1.*,Account.actype FROM Customer1 NATURAL JOIN Account WHERE
Customer1.cid=Account.cid;
+-----+---------+------------+----------+-------------+-------------+
| cid | cname   | DOB        | city     | Bname       | actype      |
+-----+---------+------------+----------+-------------+-------------+
| 101 | Shikhar | 2001-01-10 | Pune     | Navipeth    | Savings     |
| 102 | Rahul   | 2001-04-15 | Mumbai   | Lokhandvala | Fixedeposit |
| 103 | Virat   | 2002-01-11 | Pune     | Navipeth    | Fixedeposit |
| 104 | Kuldeep | 2002-11-11 | Noida    | Sector11    | Current     |
| 105 | Sehvag  | 2001-07-15 | Mumbai   | Thana       | Savings     |
| 106 | Kolhi   | 2002-01-10 | Bangluru | Bang        | Savings     |
+-----+---------+------------+----------+-------------+-------------+
```

7.) List the customer names whose account has been credited with more than50000 in a single transaction.

```
mysql> SELECT cname,cid FROM Customer1 WHERE cid IN (SELECT Account.cid FROM Account INNER JOIN Transaction
ON Account.acno=Transaction.acno WHERE Account.balance>50000 AND Account.balance-Transaction.amt<50000);
+---------+-----+
| cname   | cid |
+---------+-----+
| Virat   | 103 |
| Kuldeep | 104 |
+---------+-----+
```

8.) Find the customer who has account in abc-chowk branch and who has morethan 20000 balance.

```
mysql> SELECt Customer1.*,Account.balance FRom Customer1 INNER JOIN Account ON Customer1.cid=Account.cid
WHERE Account.balance>20000;
+-----+---------+------------+----------+----------+---------+
| cid | cname   | DOB        | city     | Bname    | balance |
+-----+---------+------------+----------+----------+---------+
| 105 | Sehvag  | 2001-07-15 | Mumbai   | Thana    |   50000 |
| 106 | Kolhi   | 2002-01-10 | Bangluru | Bang     |   45000 |
| 101 | Shikhar | 2001-01-10 | Pune     | Navipeth |   50000 |
| 103 | Virat   | 2002-01-11 | Pune     | abc-chok |   55000 |
| 104 | Kuldeep | 2002-11-11 | Noida    | Sector11 |   56000 |
+-----+---------+------------+----------+----------+---------+
```

9.) Find the customer whose account branch is in Pune.

```
mysql> SELECt * FRom Customer1 WHERE city=&apos;Pune&apos;;
+-----+---------+------------+------+----------+
| cid | cname   | DOB        | city | Bname    |
+-----+---------+------------+------+----------+
| 101 | Shikhar | 2001-01-10 | Pune | Navipeth |
| 103 | Virat   | 2002-01-11 | Pune | abc-chok |
+-----+---------+------------+------+----------+
```

10.) Find employee whose date of joining is more than date of joining of abc-chowk branch employee.

```
mysql> SELECT * FROM Employee1 WHERE DATEDIFF(CURDATE(),DOJ)>DATEDIFF(CURDATE(),(SELECT DOJ FROM
Employee1 WHERE Bname=&apos;abc-chok&apos;));
+-----+-------+--------+------------+----------+
| eid | ename | mgr_id | DOJ        | Bname    |
+-----+-------+--------+------------+----------+
| 121 | Ram   |    501 | 2005-11-01 | Navipeth |
| 122 | Sham  |    504 | 2004-10-01 | Navipeth |
| 124 | Jai   |    501 | 2004-10-04 | Noida    |
+-----+-------+--------+------------+----------+
```

11.) List customer whose balance is more than 10000.

```
mysql> SELECt Customer1.*,Account.balance FRom Customer1 INNER JOIN Account WHERE
Customer1.cid=Account.cid AND Account.balance>10000;
+-----+---------+------------+----------+-------------+---------+
| cid | cname   | DOB        | city     | Bname       | balance |
+-----+---------+------------+----------+-------------+---------+
| 105 | Sehvag  | 2001-07-15 | Mumbai   | Thana       |   50000 |
| 106 | Kolhi   | 2002-01-10 | Bangluru | Bang        |   45000 |
| 101 | Shikhar | 2001-01-10 | Pune     | Navipeth    |   50000 |
| 102 | Rahul   | 2001-04-15 | Mumbai   | Lokhandvala |   18000 |
| 103 | Virat   | 2002-01-11 | Pune     | abc-chok    |   55000 |
| 104 | Kuldeep | 2002-11-11 | Noida    | Sector11    |   56000 |
+-----+---------+------------+----------+-------------+---------+
```

12.) List the customers who have both account and loan.

```
mysql> SELECT Customer1.* FROM Customer1 WHERE cid IN (SELECT Account.cid FROM Account INNER JOIN Loan
WHERE Account.cid=Loan.cid);
+-----+---------+------------+--------+-------------+
| cid | cname   | DOB        | city   | Bname       |
+-----+---------+------------+--------+-------------+
| 101 | Shikhar | 2001-01-10 | Pune   | Navipeth    |
| 102 | Rahul   | 2001-04-15 | Mumbai | Lokhandvala |
| 103 | Virat   | 2002-01-11 | Pune   | abc-chok    |
| 104 | Kuldeep | 2002-11-11 | Noida  | Sector11    |
| 105 | Sehvag  | 2001-07-15 | Mumbai | Thana       |
+-----+---------+------------+--------+-------------+
```

13.) Create a view which will contain branch name and number of customers in ithaving proper column name.

```
mysql> CREATE VIEW View2 AS SELECt Bname,count(Bname) AS NO_Of_Customer FRom Customer1 GROUP BY
Bname;
Query OK, 0 rows affected (0.16 sec)
mysql> SELECt * FRom View2;
+-------------+----------------+
| Bname       | NO_Of_Customer |
+-------------+----------------+
| abc-chok    |              1 |
| Bang        |              1 |
| Lokhandvala |              1 |
| Navipeth    |              1 |
| Sector11    |              1 |
| Thana       |              1 |
+-------------+----------------+
```

14.) Create a view which will show average balance and total balance of each account type.

```
mysql> CREATE VIEW View3 AS SELECt actype,AVG(balance) AS Average,SUM(balance) AS Total FRom Account GROUP
BY actype;
Query OK, 0 rows affected (0.08 sec)
mysql> SELECt * FRom View3;
+-------------+------------+--------+
| actype      | Average    | Total  |
+-------------+------------+--------+
| Savings     | 48333.3333 | 145000 |
| Fixedeposit | 36500.0000 |  73000 |
| Current     | 56000.0000 |  56000 |
+-------------+------------+--------+
```

15.) Find the customer who has not taken loan.

```
mysql> SELECt * FRom Customer1 WHERE cid NOT IN (SELECt cid FROM Loan);
+-----+-------+------------+----------+-------+
| cid | cname | DOB        | city     | Bname |
+-----+-------+------------+----------+-------+
| 106 | Kolhi | 2002-01-10 | Bangluru | Bang  |
+-----+-------+------------+----------+-------+
```

16.) Display customer name in uppercase name whose nameconsists ofJi and alsoprint them in reverse order.

```
mysql> SELECT UPPER(cname),REVERSE(cname) FROM Customer1 WHERE cname LIKE &apos;%Ra%&apos;;
+--------------+----------------+
| UPPER(cname) | REVERSE(cname) |
+--------------+----------------+
| RAHUL        | luhaR          |
| VIRAT        | tariV          |
+--------------+----------------+
```

17.) Print customer name and branch name who has got highest balance in hisaccount using nested query.

```
mysql> SELECT Customer1.cname,Customer1.Bname,Account.balance FROM Customer1 INNER JOIN Account ON
Customer1.cid=Account.cid WHERE Account.bala
nce=(SELECT MAX(balance) FROM Account);
+---------+----------+---------+
| cname   | Bname    | balance |
+---------+----------+---------+
| Kuldeep | Sector11 |   56000 |
+---------+----------+---------+
```

18.) Display employee name and his respective managers name.

```
mysql> SELECt ename,mgr_id FRom Employee1;
+-------+--------+
| ename | mgr_id |
+-------+--------+
| Ram   |    501 |
| Sham  |    504 |
| Bheem |    503 |
| Jai   |    501 |
+-------+--------+
```