Database Management System Lab

Assignment - 5

Name :- Harsh Singh

Roll No. :- 244CA018

1. Create the following tables with the following attributes and constraints on them.
   Create all the tables by defining primary key, foreign key and other appropriate constraints.

a. Bank (bk_code, bk_name, bk_address)

```
=>   CREATE TABLE Bank (
     bk_code NUMBER PRIMARY KEY,
     bk_name VARCHAR2(25) NOT NULL,
     bk_address VARCHAR2(25) NOT NULL
     );
```

b. Branch (br_id, br_name, br_address, bk_code)

```
=>   CREATE TABLE Branch (
     br_id VARCHAR2(8) PRIMARY KEY,
     br_name VARCHAR2(25) NOT NULL,
     br_address VARCHAR2(25) NOT NULL,
     bk_code NUMBER,
     CONSTRAINT fk_branch_bank FOREIGN KEY (bk_code)
     REFERENCES Bank(bk_code) ON DELETE CASCADE
     );
```

c. Customer (cust_ID, cust_name, phone_no, address)

```
=>   CREATE TABLE Customer (
     cust_ID NUMBER PRIMARY KEY,
     cust_name VARCHAR2(25) NOT NULL,
     phone_no VARCHAR2(15) UNIQUE,
     address VARCHAR2(25) NOT NULL
     );
```

d. Account (acc_no, acc_type, balance, br_id)

```
=>   CREATE TABLE Account (
     acc_no VARCHAR2(8) PRIMARY KEY,
     acc_type VARCHAR2(25) NOT NULL,
     balance NUMBER(15,2) CHECK (balance >= 0),
     br_id VARCHAR2(8),
     CONSTRAINT fk_account_branch FOREIGN KEY (br_id)
     REFERENCES Branch(br_id) ON DELETE SET NULL
     );
```

e. Customer_Account (cust_ID, acc_no)

```
=>   CREATE TABLE Customer_Account (
     cust_ID NUMBER,
     acc_no VARCHAR2(8),
     PRIMARY KEY (cust_ID, acc_no),
     CONSTRAINT fk_cust_acc_customer FOREIGN KEY (cust_ID)
     REFERENCES Customer(cust_ID) ON DELETE CASCADE,
     CONSTRAINT fk_cust_acc_account FOREIGN KEY (acc_no)
     REFERENCES Account(acc_no) ON DELETE CASCADE
     );
```

f. Loan (loan_ID, loan_type, amount, br_id)

```
=>   CREATE TABLE Loan (
     loan_ID VARCHAR2(8) PRIMARY KEY,
     loan_type VARCHAR2(25) NOT NULL,
     amount NUMBER(15,2) CHECK (amount > 0),
     br_id VARCHAR2(8),
     CONSTRAINT fk_loan_branch FOREIGN KEY (br_id)
     REFERENCES Branch(br_id) ON DELETE SET NULL
     );
```

g. Customer_Loan (cust_ID, loan_ID)

```
=>   CREATE TABLE Customer_Loan (
     cust_ID NUMBER,
     loan_ID VARCHAR2(8),
     PRIMARY KEY (cust_ID, loan_ID),
     CONSTRAINT fk_cust_loan_customer FOREIGN KEY (
     cust_ID) REFERENCES Customer(cust_ID) ON DELETE
     CASCADE,
     CONSTRAINT fk_cust_loan_loan FOREIGN KEY (loan_ID)
     REFERENCES Loan(loan_ID) ON DELETE CASCADE
     );
```

2. Insert atleast five records in each table.

a. Bank

```
=>   INSERT INTO Bank VALUES (201, 'SBI', 'MG Road');
     INSERT INTO Bank VALUES (202, 'HDFC', 'Brigade
     Road');
     INSERT INTO Bank VALUES (203, 'ICICI', 'Whitefield');
     INSERT INTO Bank VALUES (204, 'Axis', 'Electronic
     City');
     INSERT INTO Bank VALUES (205, 'Canara', 'Jayanagar');
```

b. Branch

```
=>   INSERT INTO Branch VALUES ('br_01', 'SBI Main', 'MG
     Road', 201);
     INSERT INTO Branch VALUES ('br_02', 'HDFC North',
     'Koramangala', 202);
     INSERT INTO Branch VALUES ('br_03', 'ICICI West',
     'NITK', 203);
     INSERT INTO Branch VALUES ('br_04', 'Axis South',
     'BTM Layout', 204);
     INSERT INTO Branch VALUES ('br_05', 'Canara East',
     'Indiranagar', 205);
```

c. Customer

```
=>   INSERT INTO Customer VALUES (102, 'Ravi Sharma',
     '9876543210', 'Surathkal');
     INSERT INTO Customer VALUES (103, 'Ananya Reddy',
     '9876543211', 'HSR Layout');
     INSERT INTO Customer VALUES (104, 'Vikram Joshi',
     '9876543212', 'Yelahanka');
     INSERT INTO Customer VALUES (105, 'Neha Kapoor',
     '9876543213', 'Jayanagar');
     INSERT INTO Customer VALUES (106, 'Arjun Rao',
     '9876543214', 'Banashankari');
```

d. Account

```
=>   INSERT INTO Account VALUES ('A301', 'Savings',
     50000.50, 'br_01');
     INSERT INTO Account VALUES ('A302', 'Current',
     75000.00, 'br_02');
     INSERT INTO Account VALUES ('A303', 'Savings',
     20000.25, 'br_03');
     INSERT INTO Account VALUES ('A304', 'Fixed Deposit',
     100000.00, 'br_04');
     INSERT INTO Account VALUES ('A305', 'Savings',
     35000.75, 'br_05');
```

e. Customer_Account

```
=>   INSERT INTO Customer_Account VALUES (102, 'A301');
     INSERT INTO Customer_Account VALUES (103, 'A302');
     INSERT INTO Customer_Account VALUES (104, 'A303');
     INSERT INTO Customer_Account VALUES (105, 'A304');
     INSERT INTO Customer_Account VALUES (106, 'A305');
```

f. Loan

```
=>   INSERT INTO Loan VALUES ('L401', 'Home Loan',
     500000.00, 'br_01');
     INSERT INTO Loan VALUES ('L402', 'Car Loan',
     300000.00, 'br_02');
     INSERT INTO Loan VALUES ('L403', 'Education Loan',
     200000.00, 'br_03');
     INSERT INTO Loan VALUES ('L404', 'Personal Loan',
     150000.00, 'br_04');
     INSERT INTO Loan VALUES ('L405', 'Gold Loan',
     100000.00, 'br_05');
```

g. Customer_Loan

```
=>   INSERT INTO Customer_Loan VALUES (102, 'L401');
     INSERT INTO Customer_Loan VALUES (103, 'L402');
     INSERT INTO Customer_Loan VALUES (104, 'L403');
     INSERT INTO Customer_Loan VALUES (105, 'L404');
     INSERT INTO Customer_Loan VALUES (106, 'L405');
```

3. List the details of all customers.

```
=>   SELECT *
     FROM Customer;
```

4. Find the cust_ID and phone number of customer ‘Ravi’.

```
=>   SELECT cust_id, phone_no
     FROM Customer
     WHERE cust_name like = ‘Ravi%’;
```

5. Find the Address of all branches of br_01.

```
=>   SELECT br_address
     FROM Branch
     WHERE br_id = ‘br_01’;
```

6. Find the details of Customer having ID 103.

```
=>   SELECT *
     FROM Customer
     WHERE cust_id = 103;
```

7. List the account details having balance more than 10000.

```
=>   SELECT *
     FROM Account
     WHERE balance > 10000;
```

8. List the account details of branch br_02.

```
=>   SELECT *
     FROM Account
     WHERE br_id = 'br_02';
```

9. List the loan details of branch br_01.

```
=>   SELECT *
     FROM Loan
     WHERE br_id = 'br_01';
```

10. List the account details with their branch address.

```
=>   SELECT a.*, b.br_address
     FROM Account a JOIN Branch b ON a.br_id = b.br_id;
```

11. List the customer details with their account details.

```
=>   SELECT c.*, a.acc_no, a.acc_type, a.balance, a.br_id
     FROM Customer c JOIN Customer_Account ca ON
     c.cust_id = ca.cust_id
     JOIN Account a on ca.acc_no = a.acc_no;
```

12. List the customer details having account type ‘savings’.

```
=>   SELECT c.*
     FROM Customer c JOIN Customer_Account ca ON
     c.cust_id = ca.cust_id
     JOIN Account a on ca.acc_no = a.acc_no
     WHERE acc_type = ‘Savings’;
```

13. List the customer details having vehicle loan.

```
=>   SELECT c.*
     FROM Customer c JOIN Customer_Loan cl on c.cust_id =
     cl.cust_id
     JOIN Loan l ON cl.loan_id = l.loan_id
     WHERE loan_type = ‘Vehicle Loan’;
```

14. List the branch names of all accounts.

```
=>   SELECT DISTONCT b.br_name
     FROM Account a JOIN Branch b on a.br_id = b.br_id;
```

15. List the customer details going to ‘Surathkal’ branch.

```
=>   SELECT c.*
     FROM Customer c JOIN Customer_Account ca ON
     c.cust_id = ca.cust_id
     JOIN Account a on ca.acc_no = a.acc_no
     JOIN Branch b on a.br_id = b.br_id
     WHERE b.br_address = ‘Surathkal’;
```

16. List the customers having loan account in ‘MG Road’ branch.

```
=>   SELECT c.*
     FROM Customer c JOIN Customer_Loan cl ON
     c.cust_id = cl.cust_id
     JOIN Loan l on cl.loan_id = l.loan_id
     JOIN Branch b on l.br_id = b.br_id
     WHERE b.br_address = ‘MG Road’;
```

17. Find the customers having balance between 1000 to 10000.

```
=>   SELECT c.*
     FROM Customer c JOIN Customer_Account ca ON
     c.cust_id = ca.cust_id
     JOIN Account a on ca.acc_no = a.acc_no
     WHERE a.balance between 1000 AND 100000;
```

18. Give a bonus of rupees 100 to customers having more than 10000 balance.

```
=>   UPDATE Account set balance = balance + 100
     WHERE balance > 10000;
```

19. Deduct 50 rupees from customers having less than 500 rupees in balance.

```
=>   UPDATE Account set balance = balance - 50
     WHERE balance < 500;
```

20. Give the customer details having home loan.

```
=>   SELECT c.*
     FROM Customer c JOIN Customer_Loan cl ON
     c.cust_id = cl.cust_id
     JOIN Loan l on cl.loan_id = l.loan_id
     WHERE l.loan_type = 'Home Loan';
```

21. Give the customer details having home loan in 'NITK' branch.

```
=>   SELECT c.*
     FROM Customer c JOIN Customer_Loan cl ON
     c.cust_id = cl.cust_id
     JOIN Loan l on cl.loan_id = l.loan_id
     JOIN Branch b on l.br_id = b.br_id
     WHERE b.br_address = 'NITK';
```

22. Add a column NOMINEE to the customer table with data type varchar (50).

```
=>   ALTER TABLE Customer
     ADD Nominee VARCHAR(50);
```

23. List all the account numbers in ascending order of their balance.

```
=>   SELECT acc_no
     FROM Account
     ORDER BY balance ASC;
```

24. Count the number of customers having account type savings.

```
=>   SELECT count(*)
     FROM Customer_Account ca JOIN Account a ON ca.acc_no
     = a.acc_no
     WHERE a.acc_type = 'Savings';
```

25. Count the number of customers for each account type.

```
=>   SELECT acc_type, count(*)
     FROM Account
     GROUP BY acc_type;
```

26. Find the total balance in Savings account.

```
=>   SELECT sum(balance)
     FROM Account
     WHERE acc_type = 'Savings';
```

27. Find the average balance of Current account.

```
=>   SELECT avg(balance)
     FROM Account
     WHERE acc_type = 'Current';
```

28. Find the average balance for each account type.

```
=>   SELECT acc_type, avg(balance) as avg_bal
     FROM Account
     GROUP BY acc_type;
```

29. Find the customer details having maximum balance.

```
=>   SELECT c.*
     FROM Customer c JOIN Customer_Account ca ON
     c.cust_id = ca.cust_id
     JOIN Account a on ca.acc_no = a.acc_no
     WHERE rownum < 2
     ORDER BY balance DESC;
```

30. Find the average amount for vehicle loan.

```
=>   SELECT c.*
     FROM Loan
     WHERE loan_type = 'Vehicle Loan';
```

31. Find the average balance in each branch.

```
=>   SELECT br_id, avg(balance)
     FROM Account
     GROUP BY br_id;
```