**10 Implement SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym for following relational schema:**

**Borrower(Rollin, Name, DateofIssue, NameofBook, Status)**

-- Create the Borrower table with necessary constraints  
CREATE TABLE borrower (  
Rollin INT AUTO\_INCREMENT PRIMARY KEY, -- Unique Roll number (acts like a sequence)  
Name VARCHAR(100) NOT NULL, -- Name of the borrower  
DateofIssue DATE NOT NULL, -- Book issue date  
NameofBook VARCHAR(100) NOT NULL, -- Title of the borrowed book  
Status ENUM('Issued', 'Returned') NOT NULL -- Book status (Issued/Returned)  
);

-- Insert sample records into Borrower table  
INSERT INTO borrower (Name, DateofIssue, NameofBook, Status) VALUES  
('Amit Sharma', '2025-05-01', 'Data Structures in C', 'Issued'),  
('Pooja Rani', '2025-05-03', 'Operating Systems', 'Returned'),  
('Ravi Kumar', '2025-05-04', 'Database Management', 'Issued'),  
('Meena Verma', '2025-05-05', 'Computer Networks', 'Returned'),  
('Ankit Joshi', '2025-05-06', 'Python Programming', 'Issued');

-- Create a view showing records where books are still issued  
CREATE VIEW IssuedBooks AS  
SELECT  
Rollin,  
Name,  
NameofBook,  
DateofIssue  
FROM  
borrower  
WHERE  
Status = 'Issued';

-- Create index on NameofBook for quick lookup by book title  
CREATE INDEX idx\_BookName ON borrower(NameofBook ASC);

-- Simulate a synonym using a view (alias for full table)  
CREATE VIEW borrower\_synonym AS  
SELECT \* FROM borrower;

-- Optional: View all records using the synonym  
-- SELECT \* FROM borrower\_synonym;

-- Optional: View only currently issued books  
-- SELECT \* FROM IssuedBooks;

;

11 Design at least 10 SQL queries for suitable database application using SQL DML statements: all types of Join, Sub-Query and View.

CREATE DATABASE BankDB;  
USE BankDB;

-- Create Tables  
CREATE TABLE customer (  
CustomerID INT AUTO\_INCREMENT PRIMARY KEY,  
Name VARCHAR(50) NOT NULL,  
Address VARCHAR(100),  
Phone VARCHAR(15)  
);

CREATE TABLE branch (  
BranchID INT AUTO\_INCREMENT PRIMARY KEY,  
BranchName VARCHAR(50) NOT NULL,  
Location VARCHAR(100)  
);

CREATE TABLE account (  
AcctNo INT AUTO\_INCREMENT PRIMARY KEY,  
Balance DECIMAL(10,2) DEFAULT 0,  
CustomerID INT,  
BranchID INT,  
FOREIGN KEY (CustomerID) REFERENCES customer(CustomerID),  
FOREIGN KEY (BranchID) REFERENCES branch(BranchID)  
);

-- Insert Sample Data  
INSERT INTO customer (Name, Address, Phone) VALUES  
('Ravi Kumar', 'Bengaluru', '9876543210'),  
('Anjali Sharma', 'Kolkata', '9123456789'),  
('Arjun Patel', 'Mumbai', '9988776655');

INSERT INTO branch (BranchName, Location) VALUES  
('Bengaluru Branch', 'Bengaluru'),  
('Kolkata Branch', 'Kolkata');

INSERT INTO account (Balance, CustomerID, BranchID) VALUES  
(15000, 1, 1),  
(20000, 2, 2),  
(12000, 3, 1);

-- Queries

-- 1. Inner Join: Customer with their account balance and branch  
SELECT c.Name, a.Balance, b.BranchName  
FROM customer c  
JOIN account a ON c.CustomerID = a.CustomerID  
JOIN branch b ON a.BranchID = b.BranchID;

-- 2. Left Join: All customers with their accounts (if any)  
SELECT c.Name, a.AcctNo, a.Balance  
FROM customer c  
LEFT JOIN account a ON c.CustomerID = a.CustomerID;

-- 3. Subquery: Customers with accounts having balance > 15000  
SELECT Name  
FROM customer  
WHERE CustomerID IN (SELECT CustomerID FROM account WHERE Balance > 15000);

-- 4. Subquery in SELECT: Customers and number of accounts  
SELECT c.Name,  
(SELECT COUNT(\*) FROM account a WHERE a.CustomerID = c.CustomerID) AS AccountCount  
FROM customer c;

-- 5. Create View: Customer account details  
CREATE VIEW CustomerAccountView AS  
SELECT c.Name, a.AcctNo, a.Balance  
FROM customer c  
JOIN account a ON c.CustomerID = a.CustomerID;

-- 6. Query the view  
SELECT \* FROM CustomerAccountView;

-- 7. Aggregate: Total balance per branch  
SELECT b.BranchName, SUM(a.Balance) AS TotalBalance  
FROM account a  
JOIN branch b ON a.BranchID = b.BranchID  
GROUP BY b.BranchName;

-- 8. Simple Inner Join: Accounts with branch info  
SELECT a.AcctNo, a.Balance, b.BranchName  
FROM account a  
JOIN branch b ON a.BranchID = b.BranchID;

-- 9. Find accounts with balance above average balance  
SELECT AcctNo, Balance  
FROM account  
WHERE Balance > (SELECT AVG(Balance) FROM account);

-- 10. List customers without accounts  
SELECT Name  
FROM customer  
WHERE CustomerID NOT IN (SELECT CustomerID FROM account);

14 Implement all SQL DML opeartions with operators, functions, and set operator for given schema:

Account(Acc\_no, branch\_name,balance)

branch(branch\_name,branch\_city,assets)

customer(cust\_name,cust\_street,cust\_city)

Depositor(cust\_name,acc\_no)

Loan(loan\_no,branch\_name,amount)

Borrower(cust\_name,loan\_no)

Solve following query:

1. Find the average account balance at each branch
2. Find no. of depositors at each branch.
3. Find the branches where average account balance > 12000.
4. Find number of tuples in customer relation.

CREATE DATABASE BankDB;  
USE BankDB;

-- Create Tables  
CREATE TABLE branch (  
branch\_name VARCHAR(50) PRIMARY KEY,  
branch\_city VARCHAR(50),  
assets DECIMAL(15,2)  
);

CREATE TABLE customer (  
cust\_name VARCHAR(50) PRIMARY KEY,  
cust\_street VARCHAR(100),  
cust\_city VARCHAR(50)  
);

CREATE TABLE account (  
acc\_no INT PRIMARY KEY,  
branch\_name VARCHAR(50),  
balance DECIMAL(10,2),  
FOREIGN KEY (branch\_name) REFERENCES branch(branch\_name)  
);

CREATE TABLE depositor (  
cust\_name VARCHAR(50),  
acc\_no INT,  
PRIMARY KEY (cust\_name, acc\_no),  
FOREIGN KEY (cust\_name) REFERENCES customer(cust\_name),  
FOREIGN KEY (acc\_no) REFERENCES account(acc\_no)  
);

CREATE TABLE loan (  
loan\_no INT PRIMARY KEY,  
branch\_name VARCHAR(50),  
amount DECIMAL(10,2),  
FOREIGN KEY (branch\_name) REFERENCES branch(branch\_name)  
);

CREATE TABLE borrower (  
cust\_name VARCHAR(50),  
loan\_no INT,  
PRIMARY KEY (cust\_name, loan\_no),  
FOREIGN KEY (cust\_name) REFERENCES customer(cust\_name),  
FOREIGN KEY (loan\_no) REFERENCES loan(loan\_no)  
);

-- Insert Sample Data

INSERT INTO branch VALUES  
('Bengaluru', 'Bengaluru', 1000000),  
('Mumbai', 'Mumbai', 850000),  
('Delhi', 'Delhi', 950000);

INSERT INTO customer VALUES  
('Ravi Kumar', 'MG Road', 'Bengaluru'),  
('Anjali Sharma', 'Park Street', 'Kolkata'),  
('Arjun Patel', 'Marine Drive', 'Mumbai'),  
('Sneha Gupta', 'Connaught Place', 'Delhi');

INSERT INTO account VALUES  
(101, 'Bengaluru', 15000),  
(102, 'Mumbai', 12000),  
(103, 'Delhi', 13000),  
(104, 'Bengaluru', 11000);

INSERT INTO depositor VALUES  
('Ravi Kumar', 101),  
('Arjun Patel', 102),  
('Sneha Gupta', 103),  
('Ravi Kumar', 104);

INSERT INTO loan VALUES  
(201, 'Bengaluru', 500000),  
(202, 'Mumbai', 300000);

INSERT INTO borrower VALUES  
('Ravi Kumar', 201),  
('Arjun Patel', 202);

-- Queries

-- 1. Find the average account balance at each branch  
SELECT branch\_name, AVG(balance) AS avg\_balance  
FROM account  
GROUP BY branch\_name;

-- 2. Find number of depositors at each branch  
SELECT a.branch\_name, COUNT(DISTINCT d.cust\_name) AS no\_of\_depositors  
FROM depositor d  
JOIN account a ON d.acc\_no = a.acc\_no  
GROUP BY a.branch\_name;

-- 3. Find branches where average account balance > 12000  
SELECT branch\_name  
FROM account  
GROUP BY branch\_name  
HAVING AVG(balance) > 12000;

-- 4. Find number of tuples in customer relation  
SELECT COUNT(\*) AS total\_customers FROM customer;

15 Implement all SQL DML opeartions with operators, functions, and set operator for given schema:

Account(Acc\_no, branch\_name,balance)

branch(branch\_name,branch\_city,assets)

customer(cust\_name,cust\_street,cust\_city)

Depositor(cust\_name,acc\_no)

Loan(loan\_no,branch\_name,amount)

Borrower(cust\_name,loan\_no)

Create above tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.

Solve following query:

1. Find the names of all branches in loan relation.
2. Find all loan numbers for loans made at Akurdi Branch with loan amount > 12000.
3. Find all customers who have a loan from bank.
4. Find their names,loan\_no and loan amount.

CREATE DATABASE BankDB;  
USE BankDB;

-- Create Tables with Constraints

CREATE TABLE branch (  
branch\_name VARCHAR(50) PRIMARY KEY,  
branch\_city VARCHAR(50) NOT NULL,  
assets DECIMAL(15,2) NOT NULL CHECK (assets >= 0)  
);

CREATE TABLE customer (  
cust\_name VARCHAR(50) PRIMARY KEY,  
cust\_street VARCHAR(100),  
cust\_city VARCHAR(50)  
);

CREATE TABLE account (  
acc\_no INT PRIMARY KEY,  
branch\_name VARCHAR(50),  
balance DECIMAL(10,2) NOT NULL CHECK (balance >= 0),  
FOREIGN KEY (branch\_name) REFERENCES branch(branch\_name)  
);

CREATE TABLE depositor (  
cust\_name VARCHAR(50),  
acc\_no INT,  
PRIMARY KEY (cust\_name, acc\_no),  
FOREIGN KEY (cust\_name) REFERENCES customer(cust\_name),  
FOREIGN KEY (acc\_no) REFERENCES account(acc\_no)  
);

CREATE TABLE loan (  
loan\_no INT PRIMARY KEY,  
branch\_name VARCHAR(50),  
amount DECIMAL(10,2) NOT NULL CHECK (amount > 0),  
FOREIGN KEY (branch\_name) REFERENCES branch(branch\_name)  
);

CREATE TABLE borrower (  
cust\_name VARCHAR(50),  
loan\_no INT,  
PRIMARY KEY (cust\_name, loan\_no),  
FOREIGN KEY (cust\_name) REFERENCES customer(cust\_name),  
FOREIGN KEY (loan\_no) REFERENCES loan(loan\_no)  
);

-- Insert Sample Data

INSERT INTO branch VALUES  
('Akurdi', 'Pune', 1500000),  
('Bengaluru', 'Bengaluru', 1000000),  
('Mumbai', 'Mumbai', 850000);

INSERT INTO customer VALUES  
('Ravi Kumar', 'MG Road', 'Bengaluru'),  
('Anjali Sharma', 'Park Street', 'Kolkata'),  
('Arjun Patel', 'Marine Drive', 'Mumbai'),  
('Sneha Gupta', 'Connaught Place', 'Delhi');

INSERT INTO account VALUES  
(101, 'Akurdi', 15000),  
(102, 'Mumbai', 12000),  
(103, 'Bengaluru', 13000),  
(104, 'Akurdi', 11000);

INSERT INTO depositor VALUES  
('Ravi Kumar', 101),  
('Arjun Patel', 102),  
('Sneha Gupta', 103),  
('Ravi Kumar', 104);

INSERT INTO loan VALUES  
(201, 'Akurdi', 20000),  
(202, 'Mumbai', 30000),  
(203, 'Akurdi', 10000);

INSERT INTO borrower VALUES  
('Ravi Kumar', 201),  
('Arjun Patel', 202),  
('Anjali Sharma', 203);

-- Queries

-- 1. Find the names of all branches in loan relation  
SELECT DISTINCT branch\_name FROM loan;

-- 2. Find all loan numbers for loans made at Akurdi Branch with loan amount > 12000  
SELECT loan\_no FROM loan  
WHERE branch\_name = 'Akurdi' AND amount > 12000;

-- 3. Find all customers who have a loan from bank  
SELECT DISTINCT cust\_name FROM borrower;

-- 4. Find their names, loan\_no and loan amount  
SELECT b.cust\_name, l.loan\_no, l.amount  
FROM borrower b  
JOIN loan l ON b.loan\_no = l.loan\_no;

**18 Implement all SQL DML opeartions with operators, functions, and set operator for given schema:**

**Account(Acc\_no, branch\_name,balance)**

**branch(branch\_name,branch\_city,assets)**

**customer(cust\_name,cust\_street,cust\_city)**

**Depositor(cust\_name,acc\_no)**

**Loan(loan\_no,branch\_name,amount)**

**Borrower(cust\_name,loan\_no)**

**Create above tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.Solve following query:**

1. **Find all customers who have an account or loan or both at bank.**
2. **Find all customers who have both account and loan at bank.**
3. **Find all customer who have account but no loan at the bank.**
4. **Find average account balance at Akurdi branch.**

CREATE DATABASE BankDB;  
USE BankDB;

-- Create Tables with constraints

CREATE TABLE branch (  
branch\_name VARCHAR(50) PRIMARY KEY,  
branch\_city VARCHAR(50) NOT NULL,  
assets DECIMAL(15,2) NOT NULL CHECK (assets >= 0)  
);

CREATE TABLE customer (  
cust\_name VARCHAR(50) PRIMARY KEY,  
cust\_street VARCHAR(100),  
cust\_city VARCHAR(50)  
);

CREATE TABLE account (  
acc\_no INT PRIMARY KEY,  
branch\_name VARCHAR(50) NOT NULL,  
balance DECIMAL(10,2) NOT NULL CHECK (balance >= 0),  
FOREIGN KEY (branch\_name) REFERENCES branch(branch\_name)  
);

CREATE TABLE depositor (  
cust\_name VARCHAR(50) NOT NULL,  
acc\_no INT NOT NULL,  
PRIMARY KEY (cust\_name, acc\_no),  
FOREIGN KEY (cust\_name) REFERENCES customer(cust\_name),  
FOREIGN KEY (acc\_no) REFERENCES account(acc\_no)  
);

CREATE TABLE loan (  
loan\_no INT PRIMARY KEY,  
branch\_name VARCHAR(50) NOT NULL,  
amount DECIMAL(10,2) NOT NULL CHECK (amount > 0),  
FOREIGN KEY (branch\_name) REFERENCES branch(branch\_name)  
);

CREATE TABLE borrower (  
cust\_name VARCHAR(50) NOT NULL,  
loan\_no INT NOT NULL,  
PRIMARY KEY (cust\_name, loan\_no),  
FOREIGN KEY (cust\_name) REFERENCES customer(cust\_name),  
FOREIGN KEY (loan\_no) REFERENCES loan(loan\_no)  
);

-- Insert Sample Data

INSERT INTO branch VALUES  
('Akurdi', 'Pune', 1500000),  
('Bengaluru', 'Bengaluru', 1000000),  
('Mumbai', 'Mumbai', 850000);

INSERT INTO customer VALUES  
('Ravi Kumar', 'MG Road', 'Bengaluru'),  
('Anjali Sharma', 'Park Street', 'Kolkata'),  
('Arjun Patel', 'Marine Drive', 'Mumbai'),  
('Sneha Gupta', 'Connaught Place', 'Delhi'),  
('Vikram Singh', 'Residency Road', 'Chennai');

INSERT INTO account VALUES  
(101, 'Akurdi', 15000),  
(102, 'Mumbai', 12000),  
(103, 'Bengaluru', 13000),  
(104, 'Akurdi', 11000);

INSERT INTO depositor VALUES  
('Ravi Kumar', 101),  
('Arjun Patel', 102),  
('Sneha Gupta', 103),  
('Vikram Singh', 104);

INSERT INTO loan VALUES  
(201, 'Akurdi', 20000),  
(202, 'Mumbai', 30000),  
(203, 'Akurdi', 10000);

INSERT INTO borrower VALUES  
('Ravi Kumar', 201),  
('Arjun Patel', 202),  
('Anjali Sharma', 203);

-- Queries

-- 1. Find all customers who have an account or loan or both at bank  
SELECT DISTINCT cust\_name  
FROM (  
SELECT cust\_name FROM depositor  
UNION  
SELECT cust\_name FROM borrower  
) AS CustomersWithAccountOrLoan;

-- 2. Find all customers who have both account and loan at bank  
SELECT DISTINCT cust\_name  
FROM depositor  
WHERE cust\_name IN (SELECT cust\_name FROM borrower);

-- 3. Find all customers who have account but no loan at the bank  
SELECT DISTINCT cust\_name  
FROM depositor  
WHERE cust\_name NOT IN (SELECT cust\_name FROM borrower);

-- 4. Find average account balance at Akurdi branch  
SELECT AVG(balance) AS AvgBalanceAtAkurdi  
FROM account  
WHERE branch\_name = 'Akurdi';

**19 Implement all SQL DML operations with operators, functions, and set operator for given schema:**

**Account(Acc\_no, branch\_name,balance)**

**branch(branch\_name,branch\_city,assets)**

**customer(cust\_name,cust\_street,cust\_city)**

**Depositor(cust\_name,acc\_no)**

**Loan(loan\_no,branch\_name,amount)**

**Borrower(cust\_name,loan\_no)**

**Solve following query:**

1. **Calculate total loan amount given by bank.**
2. **Delete all loans with loan amount between 1300 and 1500.**
3. **Delete all tuples at every branch located in Nigdi.**

CREATE DATABASE IF NOT EXISTS BankDB;  
USE BankDB;

-- Create tables if not created already (with constraints)

CREATE TABLE IF NOT EXISTS branch (  
branch\_name VARCHAR(50) PRIMARY KEY,  
branch\_city VARCHAR(50) NOT NULL,  
assets DECIMAL(15,2) NOT NULL CHECK (assets >= 0)  
);

CREATE TABLE IF NOT EXISTS customer (  
cust\_name VARCHAR(50) PRIMARY KEY,  
cust\_street VARCHAR(100),  
cust\_city VARCHAR(50)  
);

CREATE TABLE IF NOT EXISTS account (  
acc\_no INT PRIMARY KEY,  
branch\_name VARCHAR(50) NOT NULL,  
balance DECIMAL(10,2) NOT NULL CHECK (balance >= 0),  
FOREIGN KEY (branch\_name) REFERENCES branch(branch\_name)  
);

CREATE TABLE IF NOT EXISTS depositor (  
cust\_name VARCHAR(50) NOT NULL,  
acc\_no INT NOT NULL,  
PRIMARY KEY (cust\_name, acc\_no),  
FOREIGN KEY (cust\_name) REFERENCES customer(cust\_name),  
FOREIGN KEY (acc\_no) REFERENCES account(acc\_no)  
);

CREATE TABLE IF NOT EXISTS loan (  
loan\_no INT PRIMARY KEY,  
branch\_name VARCHAR(50) NOT NULL,  
amount DECIMAL(10,2) NOT NULL CHECK (amount > 0),  
FOREIGN KEY (branch\_name) REFERENCES branch(branch\_name)  
);

CREATE TABLE IF NOT EXISTS borrower (  
cust\_name VARCHAR(50) NOT NULL,  
loan\_no INT NOT NULL,  
PRIMARY KEY (cust\_name, loan\_no),  
FOREIGN KEY (cust\_name) REFERENCES customer(cust\_name),  
FOREIGN KEY (loan\_no) REFERENCES loan(loan\_no)  
);

-- Sample Data Inserts (optional if tables are empty)

INSERT IGNORE INTO branch VALUES  
('Akurdi', 'Pune', 1500000),  
('Bengaluru', 'Bengaluru', 1000000),  
('Mumbai', 'Mumbai', 850000),  
('Nigdi', 'Pune', 700000);

INSERT IGNORE INTO customer VALUES  
('Ravi Kumar', 'MG Road', 'Bengaluru'),  
('Anjali Sharma', 'Park Street', 'Kolkata'),  
('Arjun Patel', 'Marine Drive', 'Mumbai'),  
('Sneha Gupta', 'Connaught Place', 'Delhi'),  
('Vikram Singh', 'Residency Road', 'Chennai');

INSERT IGNORE INTO account VALUES  
(101, 'Akurdi', 15000),  
(102, 'Mumbai', 12000),  
(103, 'Bengaluru', 13000),  
(104, 'Nigdi', 11000);

INSERT IGNORE INTO depositor VALUES  
('Ravi Kumar', 101),  
('Arjun Patel', 102),  
('Sneha Gupta', 103),  
('Vikram Singh', 104);

INSERT IGNORE INTO loan VALUES  
(201, 'Akurdi', 20000),  
(202, 'Mumbai', 1400),  
(203, 'Nigdi', 1350),  
(204, 'Akurdi', 10000);

INSERT IGNORE INTO borrower VALUES  
('Ravi Kumar', 201),  
('Arjun Patel', 202),  
('Anjali Sharma', 203);

-- Queries for the given problems

-- 1. Calculate total loan amount given by bank  
SELECT SUM(amount) AS TotalLoanAmount FROM loan;

-- 2. Delete all loans with loan amount between 1300 and 1500  
DELETE FROM loan WHERE amount BETWEEN 1300 AND 1500;

-- 3. Delete all tuples at every branch located in Nigdi

-- Step 1: Delete dependent tuples from borrower and depositor related to Nigdi branch

DELETE b  
FROM borrower b  
JOIN loan l ON b.loan\_no = l.loan\_no  
WHERE l.branch\_name = 'Nigdi';

DELETE d  
FROM depositor d  
JOIN account a ON d.acc\_no = a.acc\_no  
WHERE a.branch\_name = 'Nigdi';

-- Step 2: Delete loans and accounts of Nigdi branch

DELETE FROM loan WHERE branch\_name = 'Nigdi';  
DELETE FROM account WHERE branch\_name = 'Nigdi';

-- Step 3: Delete the branch 'Nigdi' from branch table

DELETE FROM branch WHERE branch\_name = 'Nigdi';

**20 Create the following tables.**

1. **Deposit (actno,cname,bname,amount,adate)**
2. **Branch (bname,city)**
3. **Customers (cname, city)**
4. **Borrow(loanno,cname,bname, amount)**

**Add primary key and foreign key wherever applicable.Insert data into the above created tables.**

1. **Display account date of customers “ABC”.**
2. **Modify the size of attribute of amount in deposit**
3. **Display names of customers living in city pune.**
4. **Display name of the city where branch “OBC” is located.**
5. **Find the number of tuples in the *customer* relation**

CREATE DATABASE IF NOT EXISTS BankDB;  
USE BankDB;

-- 1. Create Deposit table  
CREATE TABLE IF NOT EXISTS Deposit (  
actno INT PRIMARY KEY,  
cname VARCHAR(50) NOT NULL,  
bname VARCHAR(50) NOT NULL,  
amount DECIMAL(15,2) NOT NULL,  
adate DATE NOT NULL  
);

-- 2. Create Branch table  
CREATE TABLE IF NOT EXISTS Branch (  
bname VARCHAR(50) PRIMARY KEY,  
city VARCHAR(50) NOT NULL  
);

-- 3. Create Customers table  
CREATE TABLE IF NOT EXISTS Customers (  
cname VARCHAR(50) PRIMARY KEY,  
city VARCHAR(50) NOT NULL  
);

-- 4. Create Borrow table  
CREATE TABLE IF NOT EXISTS Borrow (  
loanno INT PRIMARY KEY,  
cname VARCHAR(50) NOT NULL,  
bname VARCHAR(50) NOT NULL,  
amount DECIMAL(15,2) NOT NULL,  
FOREIGN KEY (cname) REFERENCES Customers(cname),  
FOREIGN KEY (bname) REFERENCES Branch(bname)  
);

-- Insert sample data into Branch  
INSERT IGNORE INTO Branch VALUES  
('OBC', 'Pune'),  
('SBI', 'Mumbai'),  
('HDFC', 'Delhi');

-- Insert sample data into Customers  
INSERT IGNORE INTO Customers VALUES  
('ABC', 'Pune'),  
('XYZ', 'Mumbai'),  
('PQR', 'Pune'),  
('DEF', 'Delhi');

-- Insert sample data into Deposit  
INSERT IGNORE INTO Deposit VALUES  
(1001, 'ABC', 'OBC', 15000.00, '2025-05-01'),  
(1002, 'XYZ', 'SBI', 20000.00, '2025-04-20'),  
(1003, 'PQR', 'OBC', 12000.00, '2025-03-15');

-- Insert sample data into Borrow  
INSERT IGNORE INTO Borrow VALUES  
(5001, 'ABC', 'OBC', 500000.00),  
(5002, 'DEF', 'HDFC', 300000.00);

-- 1. Display account date (adate) of customers named "ABC"  
SELECT adate FROM Deposit WHERE cname = 'ABC';

-- 2. Modify the size of attribute amount in Deposit (increase precision)  
ALTER TABLE Deposit MODIFY amount DECIMAL(18,2);

-- 3. Display names of customers living in city Pune  
SELECT cname FROM Customers WHERE city = 'Pune';

-- 4. Display name of the city where branch "OBC" is located  
SELECT city FROM Branch WHERE bname = 'OBC';

-- 5. Find the number of tuples in the Customers relation  
SELECT COUNT(\*) AS NumberOfCustomers FROM Customers;

**21 Create following tables:**

1. **Deposit (actno,cname,bname,amount,adate)**
2. **Branch (bname,city)**
3. **Customers (cname, city)**
4. **Borrow(loanno,cname,bname, amount)**

**Add primary key and foreign key wherever applicable. Insert data into the above created tables.**

1. **Display customer name having living city Bombay and branch city Nagpur**
2. **Display customer name having same living city as their branch city**
3. **Display customer name who are borrowers as well as depositors and having living city Nagpur.**

CREATE DATABASE BankDB;  
USE BankDB;

-- 6. Create Deposit table  
CREATE TABLE IF NOT EXISTS Deposit (  
actno INT PRIMARY KEY,  
cname VARCHAR(50) NOT NULL,  
bname VARCHAR(50) NOT NULL,  
amount DECIMAL(15,2) NOT NULL,  
adate DATE NOT NULL  
);

-- 7. Create Branch table  
CREATE TABLE IF NOT EXISTS Branch (  
bname VARCHAR(50) PRIMARY KEY,  
city VARCHAR(50) NOT NULL  
);

-- 8. Create Customers table  
CREATE TABLE IF NOT EXISTS Customers (  
cname VARCHAR(50) PRIMARY KEY,  
city VARCHAR(50) NOT NULL  
);

-- 9. Create Borrow table  
CREATE TABLE IF NOT EXISTS Borrow (  
loanno INT PRIMARY KEY,  
cname VARCHAR(50) NOT NULL,  
bname VARCHAR(50) NOT NULL,  
amount DECIMAL(15,2) NOT NULL,  
FOREIGN KEY (cname) REFERENCES Customers(cname),  
FOREIGN KEY (bname) REFERENCES Branch(bname)  
);

-- Insert sample data into Branch  
INSERT IGNORE INTO Branch VALUES  
('OBC', 'Nagpur'),  
('SBI', 'Mumbai'),  
('HDFC', 'Nagpur'),  
('BOB', 'Bombay');

-- Insert sample data into Customers  
INSERT IGNORE INTO Customers VALUES  
('Rahul', 'Bombay'),  
('Sneha', 'Nagpur'),  
('Amit', 'Nagpur'),  
('Neha', 'Mumbai'),  
('Kiran', 'Nagpur');

-- Insert sample data into Deposit  
INSERT IGNORE INTO Deposit VALUES  
(1001, 'Rahul', 'BOB', 15000.00, '2025-05-01'),  
(1002, 'Sneha', 'HDFC', 20000.00, '2025-04-20'),  
(1003, 'Kiran', 'OBC', 12000.00, '2025-03-15'),  
(1004, 'Amit', 'SBI', 17000.00, '2025-02-10');

-- Insert sample data into Borrow  
INSERT IGNORE INTO Borrow VALUES  
(5001, 'Sneha', 'HDFC', 500000.00),  
(5002, 'Amit', 'OBC', 300000.00),  
(5003, 'Kiran', 'OBC', 350000.00);

-- 1. Display customer name having living city Bombay and branch city Nagpur  
SELECT DISTINCT c.cname  
FROM Customers c  
JOIN Deposit d ON c.cname = d.cname  
JOIN Branch b ON d.bname = b.bname  
WHERE c.city = 'Bombay' AND b.city = 'Nagpur';

-- 2. Display customer name having same living city as their branch city  
SELECT DISTINCT c.cname  
FROM Customers c  
JOIN Deposit d ON c.cname = d.cname  
JOIN Branch b ON d.bname = b.bname  
WHERE c.city = b.city;

-- 3. Display customer name who are borrowers as well as depositors and having living city Nagpur  
SELECT DISTINCT c.cname  
FROM Customers c  
JOIN Deposit d ON c.cname = d.cname  
JOIN Borrow br ON c.cname = br.cname  
WHERE c.city = 'Nagpur';

**22 Create the following tables.**

1. **Deposit (actno,cname,bname,amount,adate)**
2. **Branch (bname,city)**
3. **Customers (cname, city)**
4. **Borrow(loanno,cname,bname, amount)**

**Add primary key and foreign key wherever applicable.**

**Insert data into the above created tables.**

1. **Display loan no and loan amount of borrowers having the same branch as that of sunil.**
2. **Display deposit and loan details of customers in the city where pramod is living.**
3. **Display borrower names having deposit amount greater than 1000 and having the same living city as pramod.**
4. **Display branch and living city of ‘ABC’**

CREATE DATABASE IF NOT EXISTS BankDB;  
USE BankDB;

-- 4. Create Deposit table  
CREATE TABLE IF NOT EXISTS Deposit (  
actno INT PRIMARY KEY,  
cname VARCHAR(50) NOT NULL,  
bname VARCHAR(50) NOT NULL,  
amount DECIMAL(15,2) NOT NULL,  
adate DATE NOT NULL  
);

-- 5. Create Branch table  
CREATE TABLE IF NOT EXISTS Branch (  
bname VARCHAR(50) PRIMARY KEY,  
city VARCHAR(50) NOT NULL  
);

-- 6. Create Customers table  
CREATE TABLE IF NOT EXISTS Customers (  
cname VARCHAR(50) PRIMARY KEY,  
city VARCHAR(50) NOT NULL  
);

-- 7. Create Borrow table  
CREATE TABLE IF NOT EXISTS Borrow (  
loanno INT PRIMARY KEY,  
cname VARCHAR(50) NOT NULL,  
bname VARCHAR(50) NOT NULL,  
amount DECIMAL(15,2) NOT NULL,  
FOREIGN KEY (cname) REFERENCES Customers(cname),  
FOREIGN KEY (bname) REFERENCES Branch(bname)  
);

-- Insert sample data into Branch  
INSERT IGNORE INTO Branch VALUES  
('OBC', 'Mumbai'),  
('SBI', 'Pune'),  
('HDFC', 'Nagpur'),  
('BOB', 'Mumbai');

-- Insert sample data into Customers  
INSERT IGNORE INTO Customers VALUES  
('Sunil', 'Mumbai'),  
('Pramod', 'Pune'),  
('Rakesh', 'Nagpur'),  
('ABC', 'Mumbai'),  
('Meena', 'Pune');

-- Insert sample data into Deposit  
INSERT IGNORE INTO Deposit VALUES  
(1001, 'Sunil', 'OBC', 1500.00, '2025-05-01'),  
(1002, 'Pramod', 'SBI', 1200.00, '2025-04-20'),  
(1003, 'Meena', 'SBI', 800.00, '2025-03-15'),  
(1004, 'ABC', 'BOB', 2500.00, '2025-02-10');

-- Insert sample data into Borrow  
INSERT IGNORE INTO Borrow VALUES  
(5001, 'Sunil', 'OBC', 500000.00),  
(5002, 'Rakesh', 'HDFC', 300000.00),  
(5003, 'Meena', 'SBI', 350000.00);

-- 1. Display loan no and loan amount of borrowers having the same branch as that of Sunil  
SELECT br.loanno, br.amount  
FROM Borrow br  
JOIN Borrow sunil\_borrow ON sunil\_borrow.cname = 'Sunil'  
WHERE br.bname = sunil\_borrow.bname;

-- 2. Display deposit and loan details of customers in the city where Pramod is living  
SELECT d.cname, d.actno, d.amount AS DepositAmount, d.adate, b.loanno, b.amount AS LoanAmount  
FROM Customers c  
LEFT JOIN Deposit d ON c.cname = d.cname  
LEFT JOIN Borrow b ON c.cname = b.cname  
WHERE c.city = (SELECT city FROM Customers WHERE cname = 'Pramod');

-- 3. Display borrower names having deposit amount greater than 1000 and having the same living city as Pramod  
SELECT DISTINCT b.cname  
FROM Borrow b  
JOIN Deposit d ON b.cname = d.cname  
JOIN Customers c ON b.cname = c.cname  
WHERE d.amount > 1000  
AND c.city = (SELECT city FROM Customers WHERE cname = 'Pramod');

-- 4. Display branch and living city of ‘ABC’  
SELECT d.bname, c.city  
FROM Customers c  
JOIN Deposit d ON c.cname = d.cname  
WHERE c.cname = 'ABC';

**24 Create the following tables.**

1. **Deposit (actno,cname,bname,amount,adate)**
2. **Branch (bname,city)**
3. **Customers (cname, city)**
4. **Borrow(loanno,cname,bname, amount)**

**Add primary key and foreign key wherever applicable. Insert data into the above created tables.**

1. **Display amount for depositors living in the city where Anil is living.**
2. **Display total loan and maximum loan taken from KAROLBAGH branch.**
3. **Display total deposit of customers having account date later than ‘1-jan-98’.**
4. **Display maximum deposit of customers living in PUNE.**

CREATE DATABASE IF NOT EXISTS BankDB;  
USE BankDB;

-- Create Deposit table  
CREATE TABLE IF NOT EXISTS Deposit (  
actno INT PRIMARY KEY,  
cname VARCHAR(50) NOT NULL,  
bname VARCHAR(50) NOT NULL,  
amount DECIMAL(15,2) NOT NULL,  
adate DATE NOT NULL  
);

-- Create Branch table  
CREATE TABLE IF NOT EXISTS Branch (  
bname VARCHAR(50) PRIMARY KEY,  
city VARCHAR(50) NOT NULL  
);

-- Create Customers table  
CREATE TABLE IF NOT EXISTS Customers (  
cname VARCHAR(50) PRIMARY KEY,  
city VARCHAR(50) NOT NULL  
);

-- Create Borrow table  
CREATE TABLE IF NOT EXISTS Borrow (  
loanno INT PRIMARY KEY,  
cname VARCHAR(50) NOT NULL,  
bname VARCHAR(50) NOT NULL,  
amount DECIMAL(15,2) NOT NULL,  
FOREIGN KEY (cname) REFERENCES Customers(cname),  
FOREIGN KEY (bname) REFERENCES Branch(bname)  
);

-- Insert sample data into Branch  
INSERT IGNORE INTO Branch VALUES  
('KAROLBAGH', 'Delhi'),  
('SION', 'Mumbai'),  
('VADODARA', 'Vadodara'),  
('PUNE', 'Pune');

-- Insert sample data into Customers  
INSERT IGNORE INTO Customers VALUES  
('Anil', 'Delhi'),  
('Rahul', 'Mumbai'),  
('Sunita', 'Pune'),  
('Meena', 'Pune'),  
('Ramesh', 'Delhi');

-- Insert sample data into Deposit  
INSERT IGNORE INTO Deposit VALUES  
(101, 'Anil', 'KAROLBAGH', 1200.00, '1998-01-15'),  
(102, 'Rahul', 'SION', 1500.00, '1997-12-20'),  
(103, 'Sunita', 'PUNE', 2000.00, '1999-06-10'),  
(104, 'Meena', 'PUNE', 3000.00, '1998-08-25'),  
(105, 'Ramesh', 'KAROLBAGH', 1000.00, '1999-02-05');

-- Insert sample data into Borrow  
INSERT IGNORE INTO Borrow VALUES  
(201, 'Anil', 'KAROLBAGH', 50000.00),  
(202, 'Rahul', 'SION', 40000.00),  
(203, 'Sunita', 'PUNE', 30000.00),  
(204, 'Meena', 'PUNE', 20000.00),  
(205, 'Ramesh', 'KAROLBAGH', 60000.00);

-- 1. Display amount for depositors living in the city where Anil is living  
SELECT d.amount  
FROM Deposit d  
JOIN Customers c ON d.cname = c.cname  
WHERE c.city = (SELECT city FROM Customers WHERE cname = 'Anil');

-- 2. Display total loan and maximum loan taken from KAROLBAGH branch  
SELECT  
SUM(amount) AS TotalLoan,  
MAX(amount) AS MaxLoan  
FROM Borrow  
WHERE bname = 'KAROLBAGH';

-- 3. Display total deposit of customers having account date later than ‘1-jan-98’  
SELECT SUM(amount) AS TotalDeposit  
FROM Deposit  
WHERE adate > '1998-01-01';

-- 4. Display maximum deposit of customers living in PUNE  
SELECT MAX(d.amount) AS MaxDeposit  
FROM Deposit d  
JOIN Customers c ON d.cname = c.cname  
WHERE c.city = 'Pune';

**13 Create the instance of the COMPANY which consists of the following tables:**

**EMPLOYEE(Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Dno)**

**DEPARTEMENT(Dname, Dno, Mgr\_ssn, Mgr\_start\_date)**

**DEPT\_LOCATIONS(Dnumber, Dlocation)**

**PROJECT(Pname, Pnumber, Plocation, Dno)**

**WORKS\_ON(Essn, Pno, Hours)**

**DEPENDENT(Essn, Dependent\_name, Sex, Bdate, Relationship)**

**Perform following queries**

1. **For every project located in ‘Stafford’, list the project number, the controlling department number, and the department manager’s last name,address, and birth date.**
2. **Make a list of all project numbers for projects that involve an employee whose last name is ‘Smith’, either as a worker or as a manager of the department that controls the project.**
3. **Retrieve all employees whose address is in Houston, Texas.**
4. **Show the resulting salaries if every employee working on the ‘ProductX’ project is given a 10 percent raise.**

-- =========================

-- 1. CREATE TABLES

-- =========================

CREATE TABLE EMPLOYEE (

Fname VARCHAR(20),

Minit CHAR(1),

Lname VARCHAR(20),

Ssn CHAR(9) PRIMARY KEY,

Bdate DATE,

Address VARCHAR(100),

Sex CHAR(1),

Salary DECIMAL(10,2),

Dno INT

);

CREATE TABLE DEPARTMENT (

Dname VARCHAR(20),

Dno INT PRIMARY KEY,

Mgr\_ssn CHAR(9),

Mgr\_start\_date DATE

);

CREATE TABLE DEPT\_LOCATIONS (

Dnumber INT,

Dlocation VARCHAR(50),

FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dno)

);

CREATE TABLE PROJECT (

Pname VARCHAR(50),

Pnumber INT PRIMARY KEY,

Plocation VARCHAR(50),

Dno INT,

FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dno)

);

CREATE TABLE WORKS\_ON (

Essn CHAR(9),

Pno INT,

Hours DECIMAL(5,2),

FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),

FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber)

);

CREATE TABLE DEPENDENT (

Essn CHAR(9),

Dependent\_name VARCHAR(20),

Sex CHAR(1),

Bdate DATE,

Relationship VARCHAR(20),

FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn)

);

-- =========================

-- 2. INSERT SAMPLE DATA

-- =========================

INSERT INTO EMPLOYEE VALUES

('John', 'B', 'Smith', '123456789', '1980-01-15', 'Houston, Texas', 'M', 50000, 1),

('Jane', 'M', 'Doe', '987654321', '1985-05-25', 'Dallas, Texas', 'F', 60000, 2),

('Mike', 'A', 'Johnson', '456789123', '1975-03-10', 'Austin, Texas', 'M', 75000, 1);

INSERT INTO DEPARTMENT VALUES

('Research', 1, '123456789', '2010-04-01'),

('Development', 2, '987654321', '2015-01-01');

INSERT INTO DEPT\_LOCATIONS VALUES

(1, 'Houston'),

(2, 'Stafford');

INSERT INTO PROJECT VALUES

('ProductX', 10, 'Stafford', 1),

('ProductY', 20, 'Dallas', 2);

INSERT INTO WORKS\_ON VALUES

('123456789', 10, 20),

('987654321', 10, 10),

('456789123', 20, 25);

INSERT INTO DEPENDENT VALUES

('123456789', 'Anna', 'F', '2010-06-01', 'Daughter');

-- =========================

-- 3. REQUIRED QUERIES

-- =========================

-- Query 1: Projects in 'Stafford' with manager details

SELECT P.Pnumber, P.Dno, E.Lname, E.Address, E.Bdate

FROM PROJECT P

JOIN DEPARTMENT D ON P.Dno = D.Dno

JOIN EMPLOYEE E ON D.Mgr\_ssn = E.Ssn

WHERE P.Plocation = 'Stafford';

-- Query 2: Projects involving 'Smith' (as worker or manager)

-- As worker

SELECT DISTINCT P.Pnumber

FROM PROJECT P

JOIN WORKS\_ON W ON P.Pnumber = W.Pno

JOIN EMPLOYEE E ON W.Essn = E.Ssn

WHERE E.Lname = 'Smith'

UNION

-- As manager

SELECT DISTINCT P.Pnumber

FROM PROJECT P

JOIN DEPARTMENT D ON P.Dno = D.Dno

JOIN EMPLOYEE E ON D.Mgr\_ssn = E.Ssn

WHERE E.Lname = 'Smith';

-- Query 3: Employees from Houston, Texas

SELECT \* FROM EMPLOYEE

WHERE Address LIKE '%Houston, Texas%';

-- Query 4: New salaries after 10% raise for employees on ‘ProductX’

SELECT E.Fname, E.Lname, E.Salary,

E.Salary \* 1.10 AS NewSalary

FROM EMPLOYEE E

JOIN WORKS\_ON W ON E.Ssn = W.Essn

JOIN PROJECT P ON W.Pno = P.Pnumber

WHERE P.Pname = 'ProductX';