

Mysql-4 Consider following Relation 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, not null etc.

1. Create a View1 to display List all customers in alphabetical order who have loan from Pune_Station branch.
2. Create View2 on branch table by selecting any two columns and perform insert update delete operations.
3. Create View3 on borrower and depositor table by selecting any one column from each table perform insert update delete operations.
4. Create Union of left and right joint for all customers who have an account or loan or both at bank
5. Create Simple and Unique index.
6. Display index Information.

-- Branch Table

```
CREATE TABLE Branch (  
    branch_name VARCHAR(50) PRIMARY KEY,  
    branch_city VARCHAR(50) NOT NULL,  
    assets DECIMAL(15,2)  
);
```

-- Customer Table

```
CREATE TABLE Customer (  
    cust_name VARCHAR(50) PRIMARY KEY,  
    cust_street VARCHAR(50),  
    cust_city VARCHAR(50)  
);
```

-- Account Table

```
CREATE TABLE Account (  
    acc_no INT PRIMARY KEY,  
    branch_name VARCHAR(50),  
    balance DECIMAL(15,2) NOT NULL,  
    FOREIGN KEY (branch_name) REFERENCES Branch(branch_name)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

-- Depositor Table

```
CREATE TABLE Depositor (  
    cust_name VARCHAR(50),  
    acc_no INT,  
    PRIMARY KEY(cust_name, acc_no),  
    FOREIGN KEY (cust_name) REFERENCES Customer(cust_name)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (acc_no) REFERENCES Account(acc_no)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

-- Loan Table

```
CREATE TABLE Loan (  
    loan_no INT PRIMARY KEY,
```

```
    branch_name VARCHAR(50),
    amount DECIMAL(15,2) NOT NULL,
    FOREIGN KEY (branch_name) REFERENCES Branch(branch_name)
        ON DELETE CASCADE ON UPDATE CASCADE
);
```

-- Borrower Table

```
CREATE TABLE Borrower (
    cust_name VARCHAR(50),
    loan_no INT,
    PRIMARY KEY(cust_name, loan_no),
    FOREIGN KEY (cust_name) REFERENCES Customer(cust_name)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (loan_no) REFERENCES Loan(loan_no)
        ON DELETE CASCADE ON UPDATE CASCADE
);
```

-- Branches

```
INSERT INTO Branch VALUES
('Pune_Station','Pune', 500000),
('Viman_Nagar','Pune', 300000),
('Mumbai_Central','Mumbai', 700000);
```

-- Customers

```
INSERT INTO Customer VALUES
('Maya','Street1','Pune'),
('Gopal','Street2','Mumbai'),
('Riya','Street3','Delhi'),
('Kiran','Street4','Pune');
```

-- Accounts

```
INSERT INTO Account VALUES
(101,'Pune_Station', 60000),
(102,'Viman_Nagar', 45000),
(103,'Mumbai_Central', 80000);
```

-- Depositors

```
INSERT INTO Depositor VALUES
('Maya',101),
('Gopal',102),
('Kiran',103);
```

-- Loans

```
INSERT INTO Loan VALUES
(201,'Pune_Station', 120000),
(202,'Viman_Nagar', 80000),
(203,'Mumbai_Central', 150000);
```

```
-- Borrowers
INSERT INTO Borrower VALUES
('Maya',201),
('Gopal',202),
('Kiran',203);
```

Step 7: Test the Queries

Query 1 - Create View1 - List of all customers (alphabetically) who have a loan from Pune_Station branch

```
CREATE OR REPLACE VIEW View1 AS
SELECT DISTINCT c.cust_name, c.cust_city
FROM Customer c
JOIN Borrower b ON c.cust_name = b.cust_name
JOIN Loan l ON b.loan_no = l.loan_no
WHERE l.branch_name = 'Pune_Station'
ORDER BY c.cust_name;
```

```
SELECT * FROM View1;
```

Query 2 - Create View2 on Branch table (2 columns) and perform insert/update/delete

```
CREATE OR REPLACE VIEW View2 AS
SELECT branch_name, branch_city FROM Branch;
```

```
SELECT * FROM View2;
```

```
INSERT INTO View2 VALUES ('Hinjewadi','Pune');
```

```
UPDATE View2 SET branch_city='Mumbai'
WHERE branch_name='Hinjewadi';
```

```
DELETE FROM View2 WHERE branch_name='Hinjewadi';
```

```
SELECT * FROM View2;
```

Query 3 - Create View3 on Borrower + Depositor (1 column from each) (You can't always modify join views, but you can view data.)

```
CREATE OR REPLACE VIEW View3 AS
SELECT b.cust_name AS borrower_name, d.acc_no AS account_no
FROM Borrower b
JOIN Depositor d ON b.cust_name = d.cust_name;
```

```
SELECT * FROM View3;
```

Query 4 - Create UNION of Left and Right Join

To find all customers who have account OR loan OR both (Union of Depositor and Borrower)

-- Left join: all depositors + loan info

```
SELECT c.cust_name
FROM Customer c
LEFT JOIN Depositor d ON c.cust_name = d.cust_name
UNION
SELECT c.cust_name
FROM Customer c
RIGHT JOIN Borrower b ON c.cust_name = b.cust_name;
```

Query 5 - Simple Index (non-unique)

```
CREATE INDEX idx_branch_city ON Branch(branch_city);
```

Unique index

```
CREATE UNIQUE INDEX idx_cust_name ON Customer(cust_name);
```

Query 6- Display Index Information

In MySQL

```
SHOW INDEX FROM Branch;
SHOW INDEX FROM Customer;
```