

S7 - SQL Queries (in MySQL)

Problem Statement:

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)

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.
-

Creating the database

```
CREATE DATABASE Bank4;  
USE Bank4;
```

Creating tables:

```
CREATE TABLE Account (  
    acc_no INT,  
    branch_name VARCHAR(255),  
    balance INT,  
    PRIMARY KEY (acc_no)  
);
```

```
CREATE TABLE Branch (  
    branch_name VARCHAR(255),  
    branch_city VARCHAR(255),  
    assets INT,  
    PRIMARY KEY (branch_name)  
);
```

```
CREATE TABLE Customer (  
    cust_name VARCHAR(255),  
    cust_street VARCHAR(255),
```

```

    cust_city VARCHAR(255),
    PRIMARY KEY (cust_name)
);

CREATE TABLE Depositor (
    cust_name VARCHAR(255),
    acc_no INT
);

CREATE TABLE Loan (
    loan_no INT,
    branch_name VARCHAR(255),
    amount INT,
    PRIMARY KEY (loan_no)
);

CREATE TABLE Borrower (
    cust_name VARCHAR(255),
    loan_no INT
);

```

Declaring foreign keys

```

ALTER TABLE Account ADD FOREIGN KEY (branch_name) REFERENCES Branch (branch_name);
ALTER TABLE Depositor ADD FOREIGN KEY (cust_name) REFERENCES Customer (cust_name);
ALTER TABLE Depositor ADD FOREIGN KEY (acc_no) REFERENCES Account (acc_no);
ALTER TABLE Loan ADD FOREIGN KEY (branch_name) REFERENCES Branch (branch_name);
ALTER TABLE Borrower ADD FOREIGN KEY (cust_name) REFERENCES Customer (cust_name);
ALTER TABLE Borrower ADD FOREIGN KEY (loan_no) REFERENCES Loan (loan_no);

```

Inserting data

```

INSERT INTO Branch VALUES
('Pune_Station', 'Pune', 50000),
('PES', 'Pune', 65000),
('Lohegaon', 'Pune', 350000),
('Viman Nagar', 'Pune', 850000);

INSERT INTO Customer VALUES
('Kalas', 'Street 12', 'Pune'),
('Himanshu', 'Street 15', 'Pune'),
('Mehul', 'Street 29', 'Pune'),
('Macho', 'Street 59', 'Mumbai'),
('Gundet', 'Street 40', 'Mumbai'),
('Salvi', 'Street 8', 'Pune'),
('Pintu', 'Street 55', 'Ahemadnagar'),
('Piyush', 'Street 21', 'Assam');

INSERT INTO Account VALUES
(101, 'Lohegaon', 67000),
(102, 'PES', 4324),
(103, 'PES', 54670),
(104, 'Viman Nagar', 5433),
(105, 'Pune_Station', 6462);

```

```
INSERT INTO Depositor VALUES
('Kalas', 101),
('Macho', 104),
('Gundeti', 105),
('Salvi', 105);
```

```
INSERT INTO Loan VALUES
(201, 'Pune_Station', 1800),
(202, 'PES', 8500),
(203, 'PES', 15000),
(204, 'Pune_Station', 5322),
(205, 'Viman Nagar', 1300),
(206, 'Lohegaon', 1450);
```

```
INSERT INTO Borrower VALUES
('Macho', 201),
('Mehul', 202),
('Himanshu', 203),
('Salvi', 204);
```

Queries

1. Create a View1 to display List all customers in alphabetical order who have loan from Pune_Station branch.

```
CREATE VIEW View1 AS SELECT cust_name FROM Borrower INNER JOIN Loan ON
Borrower.loan_no = Loan.loan_no WHERE branch_name = "Pune_Station" ORDER BY
cust_name;
SELECT * FROM View1;
```

2. Create View2 on branch table by selecting any two columns and perform insert update delete operations.

```
-- Creating view
CREATE VIEW View2 AS SELECT branch_name, assets FROM Branch;
SELECT * FROM View2;

-- Insert operation
INSERT INTO View2 VALUES ('Kharadi', 594000);
INSERT INTO View2 VALUES ('Yerwada', 34004);
SELECT * FROM View2;

-- Update operation
UPDATE View2 SET assets = 590000 WHERE branch_name = "Kharadi";
UPDATE View2 SET assets = 24000 WHERE branch_name = "Yerwada";
SELECT * FROM View2;

-- Delete
DELETE FROM View2 WHERE branch_name = "Yerwada";
SELECT * FROM View2;
```

3. Create View3 on borrower and depositor table by selecting any one column from each table perform insert update delete operations.

```
-- Creating view
CREATE VIEW View3 AS SELECT Borrower.cust_name, Depositor.acc_no FROM Borrower JOIN
Depositor ON Borrower.cust_name = Depositor.cust_name;
SELECT * FROM View3;

-- Insert operation
INSERT INTO Borrower (cust_name, loan_no) VALUES ('Pintu', 205);
INSERT INTO Depositor (cust_name, acc_no) VALUES ('Pintu', 102);
SELECT * FROM View3;

-- Update operation
UPDATE View3 SET cust_name = "Piyush" WHERE cust_name = "Pintu";
SELECT * FROM View3;

-- Delete operation
DELETE FROM Borrower WHERE cust_name = 'Macho';
-- This will also remove it from View3. We cannot perform delete operation directly
View3 since it is created using join clause
SELECT * FROM View3;
```

4. Create Union of left and right joint for all customers who have an account or loan or both at bank

```
SELECT Borrower.cust_name FROM Borrower LEFT JOIN Loan ON Borrower.loan_no =
Loan.loan_no UNION SELECT Depositor.cust_name FROM Depositor RIGHT JOIN Account ON
Depositor.acc_no = Account.acc_no;
```

5. Create Simple and Unique index.

```
-- Simple Index
CREATE INDEX loaners ON Borrower(cust_name);

-- Unique Index
CREATE UNIQUE INDEX depos ON Depositor(cust_name);
```

6. Display index Information.

```
SHOW INDEX FROM Borrower;
SHOW INDEX FROM Depositor;
```
