**DBMS Practice Program List**

**1:** Draw an ER for Bank database with atleast 5 entities and convert them into tables.

Perform DDL on above converted tables.

1. Create tables with all constraints
2. Create views on any two tables using join conditions
3. Create index called CustomerId. Entries should be in ascending order by customer
4. name.
5. Create sequence on Acctno

ANSWER:

Create Database q1;

Use q1;

Create table Customer(CustomerID int Primary key ,Name Varchar(50),Address Varchar(50),Phone Varchar(10));

Create table Branch(BranchID INT, BranchName Varchar(50),City Varchar(50));

Create Table Account(AccNo INT Primary Key ,CustomerID int ,BranchID INT,Balance INT,Foreign Key (CustomerID) references Customer(CustomerID),Foreign Key (BranchID) references Branch(BranchID));

Create table Loan(LoanID INT Primary Key,CustomerID Int,Amount int,Foreign Key (CustomerID) references Customer(CustomerID));

**✅ Step 3: Create Views with Join Conditions**

-- View 1: Customer and Account details

CREATE VIEW CustomerAccounts AS

SELECT c.CustomerId, c.Name, a.AcctNo, a.Balance

FROM Customer c

JOIN Account a ON c.CustomerId = a.CustomerId;

-- View 2: Loan details along with customer name

CREATE VIEW CustomerLoans AS

SELECT c.Name, l.LoanId, l.Amount

FROM Customer c

JOIN Loan l ON c.CustomerId = l.CustomerId;

**✅ Step 4: Create Index on CustomerId Sorted by Customer Name**

CREATE INDEX idx\_CustomerName ON Customer(Name ASC);

**✅ Step 5: Create Sequence on AcctNo (if using Oracle/PostgreSQL)**

CREATE SEQUENCE acctno\_seq

START WITH 1001

INCREMENT BY 1;

16: 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. 2. 3. 4. Find the average account balance at each branch

Find no. of depositors at each branch.

Find the branches where average account balance > 12000.

Find number of tuples in customer relation.

Create Table Branch(BranchName Varchar(50) Primary Key,BranchCity Varchar(50),Assets int);

Create Table Account(AccNo Int Primary Key,BranchName Varchar(50),Balance int,Foreign Key (BranchName) references Branch(BranchName));

Create Table Depositor(CustomerName varchar(50),AccNo int,Foreign Key (CustomerName) references Customer(CustomerName),Foreign Key (AccNo) refrences Account(AccNo));

Create table Customer(CustomerName Varchar(50),CustomerStreet varchar(50),CustomerCity varchar(50));

Create table Loan(LoanNo Int Primary key,BranchName varchar(50),Amount int,Foreign Key (BranchName) references Branch(Branchname));

Create table Borrower(CustomerName varchar(50),LoanNo INT Primary Key,Foreign key (CustomerName) references Customer(CustomerName),Foreign Key (LoanNo) references Loan(LoanNo));

SELECT BranchName, COUNT(\*) AS NoOfDepositors

FROM Account

GROUP BY BranchName;

SELECT BranchName

FROM Account

GROUP BY BranchName

HAVING AVG(balance) > 12000;

SELECT COUNT(\*) AS NoOfTuples

FROM Customer;

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

given schema:

1. 2. 3. 4. 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:

Find the names of all branches in loan relation.

Find all loan numbers for loans made at Akurdi Branch with loan amount > 12000.

Find all customers who have a loan from bank.

Find their names,loan\_no and loan amount.

Select DISTINCT BranchName from Loan;

Select LoanNo from Loan

Where BranchName=”Akurdi” AND Amount>12000;

SELECT DISTINCT cust\_name

FROM Borrower;

SELECT b.cust\_name, l.loan\_no, l.amount

FROM Borrower b

JOIN Loan l ON b.loan\_no = l.loan\_no;

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)

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.

SELECT cust\_name FROM Depositor

UNION

SELECT cust\_name FROM Borrower;

SELECT cust\_name

FROM Depositor

WHERE cust\_name IN (SELECT cust\_name FROM Borrower);

SELECT cust\_name

FROM Depositor

WHERE cust\_name NOT IN (SELECT cust\_name FROM Borrower);

SELECT AVG(balance) AS avg\_balance

FROM Account

WHERE branch\_name = 'Akurdi';

20: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.

Select sum(amount) from Loan

Delete from loan where amount between 1300 and 1500

Delete from Branch where BranchName =”Nigdi”;

21: Create the following tables.

Deposit (actno,cname,bname,amount,adate)

Branch (bname,city)

Customers (cname, city)

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”.

SELECT adate

FROM Deposit

WHERE cname = 'ABC';

1. Modify the size of attribute of amount in deposit

ALTER TABLE Deposit

MODIFY amount DECIMAL(12,2);

1. Display names of customers living in city pune.

SELECT cname

FROM Customers

WHERE city = 'Pune';

1. Display name of the city where branch “OBC” is located.

SELECT city

FROM Branch

WHERE bname = 'OBC';

1. Find the number of tuples in the *customer* relation

SELECT COUNT(\*) AS total\_customers

FROM Customers;

23.Create following tables:

Deposit (actno,cname,bname,amount,adate)

Branch (bname,city)

Customers (cname, city)

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.