**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 name.**

**4. Create sequence on Acctno.**

ER Model for Bank Database (Short)

* Entities:
  + Customer (CustomerID, Name, Address, Phone)
  + Branch (BranchID, BranchName, Location)
  + Account (AcctNo, Balance, CustomerID, BranchID)
  + Loan (LoanID, Amount, CustomerID)
  + Transaction (TxnID, AcctNo, TxnType, Amount, TxnDate)
* Relationships:
  + Customer — Account: One-to-Many
  + Branch — Account: One-to-Many
  + Customer — Loan: One-to-Many
  + Account — Transaction: One-to-Many

-- CREATE TABLES

CREATE TABLE Customer (

CustomerID INT AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

Address VARCHAR(200),

Phone VARCHAR(15)

);

CREATE TABLE Branch (

BranchID INT AUTO\_INCREMENT PRIMARY KEY,

BranchName VARCHAR(100) NOT NULL,

Location VARCHAR(100)

);

CREATE TABLE Account (

AcctNo INT AUTO\_INCREMENT PRIMARY KEY,

Balance DECIMAL(12,2) CHECK (Balance >= 0),

CustomerID INT,

BranchID INT,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (BranchID) REFERENCES Branch(BranchID)

);

CREATE TABLE Loan (

LoanID INT AUTO\_INCREMENT PRIMARY KEY,

Amount DECIMAL(10,2) NOT NULL CHECK (Amount > 0),

CustomerID INT,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

CREATE TABLE Transaction (

TxnID INT AUTO\_INCREMENT PRIMARY KEY,

AcctNo INT,

TxnType VARCHAR(20) CHECK (TxnType IN ('Deposit', 'Withdrawal', 'Transfer')),

Amount DECIMAL(10,2) CHECK (Amount > 0),

TxnDate DATE,

FOREIGN KEY (AcctNo) REFERENCES Account(AcctNo)

);

-- INSERT DEMO DATA

-- Customers (Indian names and addresses)

INSERT INTO Customer (Name, Address, Phone) VALUES

('Rajesh Kumar', '123 MG Road, Bengaluru, Karnataka', '9876543210'),

('Anita Sharma', '45 Connaught Place, New Delhi', '9123456780'),

('Suresh Patel', '22 Marine Drive, Mumbai, Maharashtra', '9988776655'),

('Priya Singh', '56 Park Street, Kolkata, West Bengal', '9876512345'),

('Amit Verma', '89 Residency Road, Pune, Maharashtra', '9012345678');

-- Branches

INSERT INTO Branch (BranchName, Location) VALUES

('MG Road Branch', 'Bengaluru'),

('Connaught Place Branch', 'New Delhi'),

('Marine Drive Branch', 'Mumbai'),

('Park Street Branch', 'Kolkata'),

('Residency Road Branch', 'Pune');

-- Accounts (assigning customers and branches)

INSERT INTO Account (Balance, CustomerID, BranchID) VALUES

(50000.00, 1, 1),

(75000.00, 2, 2),

(62000.00, 3, 3),

(83000.00, 4, 4),

(40000.00, 5, 5);

-- Loans

INSERT INTO Loan (Amount, CustomerID) VALUES

(200000.00, 1),

(150000.00, 3),

(300000.00, 4);

-- Transactions

INSERT INTO Transaction (AcctNo, TxnType, Amount, TxnDate) VALUES

(1, 'Deposit', 10000.00, '2025-05-01'),

(2, 'Withdrawal', 5000.00, '2025-05-03'),

(3, 'Deposit', 7000.00, '2025-05-05'),

(4, 'Transfer', 12000.00, '2025-05-10'),

(5, 'Deposit', 8000.00, '2025-05-12');

-- CREATE VIEWS

CREATE VIEW CustomerAccounts AS

SELECT C.CustomerID, C.Name, A.AcctNo, A.Balance, B.BranchName

FROM Customer C

JOIN Account A ON C.CustomerID = A.CustomerID

JOIN Branch B ON A.BranchID = B.BranchID;

CREATE VIEW CustomerLoans AS

SELECT C.CustomerID, C.Name, L.LoanID, L.Amount

FROM Customer C

JOIN Loan L ON C.CustomerID = L.CustomerID;

-- CREATE INDEX

CREATE INDEX CustomerNameIndex ON Customer(Name);

**2: Draw an ER for Company database with atleast 4 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 conditions**

**3. create index called EmployeeId for the department table. Entries should be in ascending order by department id and then by employee id within each department.**

**4. create sequence on Employee id.**

**ER Model (Short)**

* **Entities:**
  + Department (DeptID, DeptName)
  + Employee (EmpID, EmpName, Salary, DeptID)
  + Project (ProjectID, ProjectName, DeptID)
  + Works\_On (EmpID, ProjectID, HoursWorked)
* **Relationships:**
  + Department — Employee: One-to-Many
  + Department — Project: One-to-Many
  + Employee — Project: Many-to-Many (via Works\_On)

-- 1. Create tables with constraints

CREATE TABLE Department (

DeptID INT PRIMARY KEY,

DeptName VARCHAR(100) NOT NULL,

ManagerID INT

);

CREATE TABLE Employee (

EmployeeID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

DeptID INT,

Salary DECIMAL(10,2),

FOREIGN KEY (DeptID) REFERENCES Department(DeptID)

);

-- Update ManagerID foreign key after Employee table creation

ALTER TABLE Department

ADD CONSTRAINT fk\_Manager FOREIGN KEY (ManagerID) REFERENCES Employee(EmployeeID);

CREATE TABLE Project (

ProjectID INT PRIMARY KEY,

ProjectName VARCHAR(100) NOT NULL,

DeptID INT,

FOREIGN KEY (DeptID) REFERENCES Department(DeptID)

);

CREATE TABLE WorksOn (

EmployeeID INT,

ProjectID INT,

Hours DECIMAL(5,2),

PRIMARY KEY (EmployeeID, ProjectID),

FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),

FOREIGN KEY (ProjectID) REFERENCES Project(ProjectID)

);

-- 2. Insert sample data

INSERT INTO Department (DeptID, DeptName) VALUES

(1, 'HR'),

(2, 'IT'),

(3, 'Finance');

INSERT INTO Employee (FirstName, LastName, DeptID, Salary) VALUES

('Rajesh', 'Kumar', 1, 50000),

('Anita', 'Sharma', 2, 60000),

('Sunil', 'Patel', 2, 65000),

('Pooja', 'Singh', 3, 70000);

-- Update ManagerID for Departments

UPDATE Department SET ManagerID = 1 WHERE DeptID = 1;

UPDATE Department SET ManagerID = 2 WHERE DeptID = 2;

UPDATE Department SET ManagerID = 4 WHERE DeptID = 3;

INSERT INTO Project (ProjectID, ProjectName, DeptID) VALUES

(101, 'Project Alpha', 2),

(102, 'Project Beta', 3);

INSERT INTO WorksOn (EmployeeID, ProjectID, Hours) VALUES

(2, 101, 40),

(3, 101, 20),

(4, 102, 35);

-- 3. Create views

CREATE VIEW EmployeeDeptView AS

SELECT E.EmployeeID, E.FirstName, E.LastName, D.DeptName

FROM Employee E

JOIN Department D ON E.DeptID = D.DeptID;

CREATE VIEW ProjectDeptView AS

SELECT P.ProjectID, P.ProjectName, D.DeptName, D.ManagerID

FROM Project P

JOIN Department D ON P.DeptID = D.DeptID;

-- 4. Create index EmployeeId on Department (by DeptID ASC, ManagerID ASC)

CREATE INDEX EmployeeId ON Department (DeptID ASC, ManagerID ASC);

**8 Draw ER for Library database with atleast 5 entities and convert them into tables.**

**Perform DDL on above converted tables.**

**1. Create tables with all constraints (Based on ERD cardinalities)**

**2. Create views on any two tables using join condition**

**3. Create index called Lib\_Index1. Entries should be in ascending order by Author name.**

**4. Create sequence on Bookid.**

Entities: Author, Publisher, Member, Book, Borrow

Keys:

* Author: AuthorID (PK)
* Publisher: PublisherID (PK)
* Member: MemberID (PK)
* Book: BookID (PK), AuthorID (FK), PublisherID (FK)
* Borrow: BorrowID (PK), BookID (FK), MemberID (FK)

Relations:

* Author 1—M Book
* Publisher 1—M Book
* Member 1—M Borrow
* Book 1—M Borrow

-- Create tables with constraints

CREATE TABLE Author (

AuthorID INT PRIMARY KEY,

AuthorName VARCHAR(100) NOT NULL

);

CREATE TABLE Publisher (

PublisherID INT PRIMARY KEY,

PublisherName VARCHAR(100) NOT NULL

);

CREATE TABLE Member (

MemberID INT PRIMARY KEY,

MemberName VARCHAR(100) NOT NULL,

Address VARCHAR(200)

);

CREATE TABLE Book (

BookID INT AUTO\_INCREMENT PRIMARY KEY,

Title VARCHAR(200) NOT NULL,

AuthorID INT,

PublisherID INT,

FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID),

FOREIGN KEY (PublisherID) REFERENCES Publisher(PublisherID)

);

CREATE TABLE Borrow (

BorrowID INT AUTO\_INCREMENT PRIMARY KEY,

BookID INT,

MemberID INT,

BorrowDate DATE,

ReturnDate DATE,

FOREIGN KEY (BookID) REFERENCES Book(BookID),

FOREIGN KEY (MemberID) REFERENCES Member(MemberID)

);

-- Insert sample data

INSERT INTO Author (AuthorID, AuthorName) VALUES

(1, 'Ruskin Bond'),

(2, 'Amish Tripathi'),

(3, 'Arundhati Roy');

INSERT INTO Publisher (PublisherID, PublisherName) VALUES

(1, 'Penguin India'),

(2, 'HarperCollins');

INSERT INTO Member (MemberID, MemberName, Address) VALUES

(101, 'Rajesh Kumar', 'Delhi'),

(102, 'Anita Sharma', 'Mumbai');

INSERT INTO Book (Title, AuthorID, PublisherID) VALUES

('The Blue Umbrella', 1, 1),

('The Immortals of Meluha', 2, 2),

('The God of Small Things', 3, 1);

INSERT INTO Borrow (BookID, MemberID, BorrowDate, ReturnDate) VALUES

(1, 101, '2025-05-01', '2025-05-10'),

(3, 102, '2025-05-05', NULL);

-- Create views using JOIN

CREATE VIEW BookAuthors AS

SELECT B.BookID, B.Title, A.AuthorName

FROM Book B

JOIN Author A ON B.AuthorID = A.AuthorID;

CREATE VIEW MemberBorrowedBooks AS

SELECT M.MemberName, B.Title, BR.BorrowDate, BR.ReturnDate

FROM Member M

JOIN Borrow BR ON M.MemberID = BR.MemberID

JOIN Book B ON BR.BookID = B.BookID;

-- Create index on AuthorName ascending

CREATE INDEX Lib\_Index1 ON Author(AuthorName ASC);