-- Table Creation with Primary and Foreign Keys

-- 1. Creating Employee table

CREATE TABLE Employee (

person\_name VARCHAR(100) PRIMARY KEY,

street VARCHAR(255),

city VARCHAR(100)

);

-- 2. Creating Works table with foreign key to Employee

CREATE TABLE Works (

person\_name VARCHAR(100),

company\_name VARCHAR(100),

salary DECIMAL(10, 2),

PRIMARY KEY (person\_name, company\_name),

FOREIGN KEY (person\_name) REFERENCES Employee(person\_name)

);

-- 3. Creating Company table

CREATE TABLE Company (

company\_name VARCHAR(100) PRIMARY KEY,

city VARCHAR(100)

);

-- 4. Creating Branch table

CREATE TABLE Branch (

branch\_name VARCHAR(100) PRIMARY KEY,

branch\_city VARCHAR(100),

assets DECIMAL(15, 2)

);

-- 5. Creating Customer table

CREATE TABLE Customer (

customer\_name VARCHAR(100) PRIMARY KEY,

customer\_street VARCHAR(255),

customer\_city VARCHAR(100)

);

-- 6. Creating Loan table with foreign key to Branch

CREATE TABLE Loan (

loan\_number INT PRIMARY KEY,

branch\_name VARCHAR(100),

amount DECIMAL(15, 2),

FOREIGN KEY (branch\_name) REFERENCES Branch(branch\_name)

);

-- 7. Creating Borrower table with foreign keys to Customer and Loan

CREATE TABLE Borrower (

customer\_name VARCHAR(100),

loan\_number INT,

PRIMARY KEY (customer\_name, loan\_number),

FOREIGN KEY (customer\_name) REFERENCES Customer(customer\_name),

FOREIGN KEY (loan\_number) REFERENCES Loan(loan\_number)

);

-- 8. Creating Account table with foreign key to Branch

CREATE TABLE Account (

account\_number INT PRIMARY KEY,

branch\_name VARCHAR(100),

balance DECIMAL(15, 2),

FOREIGN KEY (branch\_name) REFERENCES Branch(branch\_name)

);

-- 9. Creating Depositor table with foreign keys to Customer and Account

CREATE TABLE Depositor (

customer\_name VARCHAR(100),

account\_number INT,

PRIMARY KEY (customer\_name, account\_number),

FOREIGN KEY (customer\_name) REFERENCES Customer(customer\_name),

FOREIGN KEY (account\_number) REFERENCES Account(account\_number)

);

-- Queries and Operations

-- 1. Create a view with customer\_name and customer\_street

CREATE VIEW CustomerAddress AS

SELECT customer\_name, customer\_street

FROM Customer;

-- 2. Create an index for loan\_number in Loan table

CREATE INDEX idx\_loan\_number ON Loan(loan\_number);

-- 3. Create a sequence for Account and insert 4 records using the sequence

CREATE SEQUENCE account\_seq START WITH 1 INCREMENT BY 1;

INSERT INTO Account (account\_number, branch\_name, balance) VALUES (account\_seq.NEXTVAL, 'Main Branch', 1000.00);

INSERT INTO Account (account\_number, branch\_name, balance) VALUES (account\_seq.NEXTVAL, 'Main Branch', 2000.00);

INSERT INTO Account (account\_number, branch\_name, balance) VALUES (account\_seq.NEXTVAL, 'Main Branch', 3000.00);

INSERT INTO Account (account\_number, branch\_name, balance) VALUES (account\_seq.NEXTVAL, 'Main Branch', 4000.00);

-- 4. Create synonyms for Depositor and Borrower tables

CREATE SYNONYM Syn\_Depositor FOR Depositor;

CREATE SYNONYM Syn\_Borrower FOR Borrower;

-- Display records using synonyms

SELECT \* FROM Syn\_Depositor;

SELECT \* FROM Syn\_Borrower;

-- Update a record using the synonym tables

UPDATE Syn\_Depositor

SET account\_number = 1234

WHERE customer\_name = 'John Doe';

UPDATE Syn\_Borrower

SET loan\_number = 5678

WHERE customer\_name = 'Jane Doe';

-- 5. Drop synonym for Depositor table

DROP SYNONYM Syn\_Depositor;