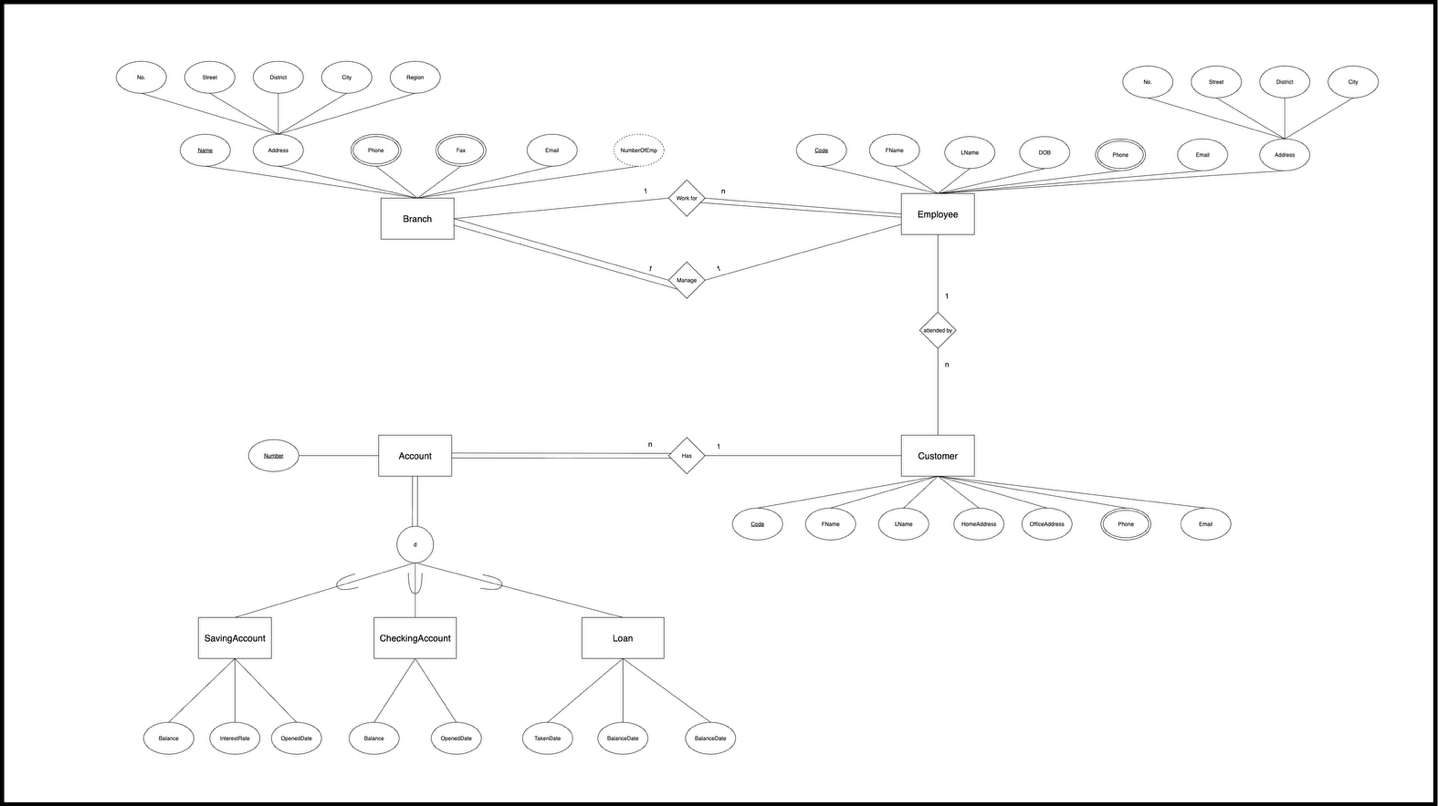
Bài 7

1.Thiết kế Cơ sở dữ liệu ở mức Khái niệm, Luận lý và Vật lý

1. Mức khái niệm



1. Mức luận lý

Branch(**Name**, AddressNo, Street, District, City, Region, Email, MngCode)

BranchPhone(**BraName, Phone**)

BranchFax(**BraName, Fax**)

Eemployee(**Code**, FName, Lname, AddressNo, Street, District, City, DOB, Email, BraName)

EmployeePhone(**EmpCode, Phone**)

Customer(**Code**, FName, Lname, HomeAddress, OfficeAddress, Email, EmpCode)

CustomerPhone(**CusCode, Phone**)

Account(**Number**, CusCode)

SavingAccount(**AccNumber**, Balance, IntersestRate, OpenedDate)

CheckingAccount(**AccNumber**, Balance, OpenedDate)

Loan(**AccNumber**, BalanceDue, IntersestRate, TakenDate)

1. Mức vật lý

Thực hiện cài đặt cơ sở dữ liệu bằng MySQL

-- Create database ABC\_Bank  
DROP DATABASE IF EXISTS ABC\_Bank;  
CREATE DATABASE ABC\_Bank;  
USE ABC\_Bank;  
  
-- Branch  
CREATE TABLE Branch (  
 Name VARCHAR(100) PRIMARY KEY,  
 AddressNo VARCHAR(20),  
 Street VARCHAR(100),  
 District VARCHAR(50),  
 City VARCHAR(50),  
 Region VARCHAR(50),  
 Email VARCHAR(100),  
 MngCode INT  
);  
  
-- Branch Phones  
CREATE TABLE BranchPhone (  
 BraName VARCHAR(100),  
 Phone VARCHAR(20),  
 PRIMARY KEY (BraName, Phone),  
 FOREIGN KEY (BraName) REFERENCES Branch(Name)  
);  
  
-- Branch Fax  
CREATE TABLE BranchFax (  
 BraName VARCHAR(100),  
 Fax VARCHAR(20),  
 PRIMARY KEY (BraName, Fax),  
 FOREIGN KEY (BraName) REFERENCES Branch(Name)  
);  
  
-- Employee  
CREATE TABLE Employee (  
 Code INT PRIMARY KEY,  
 FName VARCHAR(50),  
 LName VARCHAR(50),  
 AddressNo VARCHAR(20),  
 Street VARCHAR(100),  
 District VARCHAR(50),  
 City VARCHAR(50),  
 DOB DATE,  
 Email VARCHAR(100),  
 BraName VARCHAR(100),  
 FOREIGN KEY (BraName) REFERENCES Branch(Name)  
);  
  
-- Link Manager  
ALTER TABLE Branch  
ADD CONSTRAINT fk\_manager FOREIGN KEY (MngCode) REFERENCES Employee(Code);  
  
-- Employee Phones  
CREATE TABLE EmployeePhone (  
 EmpCode INT,  
 Phone VARCHAR(20),  
 PRIMARY KEY (EmpCode, Phone),  
 FOREIGN KEY (EmpCode) REFERENCES Employee(Code)  
);  
  
-- Customer  
CREATE TABLE Customer (  
 Code INT PRIMARY KEY,  
 FName VARCHAR(50),  
 LName VARCHAR(50),  
 HomeAddress VARCHAR(200),  
 OfficeAddress VARCHAR(200),  
 Email VARCHAR(100),  
 EmpCode INT,  
 FOREIGN KEY (EmpCode) REFERENCES Employee(Code)  
);  
  
-- Customer Phones  
CREATE TABLE CustomerPhone (  
 CusCode INT,  
 Phone VARCHAR(20),  
 PRIMARY KEY (CusCode, Phone),  
 FOREIGN KEY (CusCode) REFERENCES Customer(Code)  
);  
  
-- Account  
CREATE TABLE Account (  
 Number INT PRIMARY KEY,  
 CusCode INT,  
 FOREIGN KEY (CusCode) REFERENCES Customer(Code)  
);  
  
-- Saving Account  
CREATE TABLE SavingAccount (  
 AccNumber INT PRIMARY KEY,  
 Balance DECIMAL(15,2),  
 InterestRate DECIMAL(5,2),  
 OpenedDate DATE,  
 FOREIGN KEY (AccNumber) REFERENCES Account(Number)  
);  
  
-- Checking Account  
CREATE TABLE CheckingAccount (  
 AccNumber INT PRIMARY KEY,  
 Balance DECIMAL(15,2),  
 OpenedDate DATE,  
 FOREIGN KEY (AccNumber) REFERENCES Account(Number)  
);  
  
-- Loan  
CREATE TABLE Loan (  
 AccNumber INT PRIMARY KEY,  
 BalanceDue DECIMAL(15,2),  
 InterestRate DECIMAL(5,2),  
 TakenDate DATE,  
 FOREIGN KEY (AccNumber) REFERENCES Account(Number)  
);

2.Query: Insert, Update, Delete, Select

a. You are required to insert valid and meaningful data into the database. Each table has at least 4 rows.

Muốn insert phải gỡ bỏ rang buộc giữa MngCode và Employee

/\*  
You are required to insert valid and meaningful  
data into the database. Each table has at least 4 rows.  
\*/  
-- Branch  
INSERT INTO Branch VALUES  
('HCM01', '12', 'Le Loi', '1', 'HCM', 'South', 'hcm01@abc.com', 1001),  
('HN01', '25', 'Kim Ma', 'Ba Dinh', 'HN', 'North', 'hn01@abc.com', 1002),  
('DN01', '45', 'Tran Phu', 'Hai Chau', 'DN', 'Central', 'dn01@abc.com', 1003),  
('CT01', '78', 'Hoa Binh', 'Ninh Kieu', 'CT', 'West', 'ct01@abc.com', 1004);  
  
-- Employee  
INSERT INTO Employee VALUES  
(1001, 'John', 'Smith', '45', 'Nguyen Trai', '5', 'HCM', '1985-03-10', 'john.smith@abc.com', 'HCM01'),  
(1002, 'Anna', 'White', '22', 'Kim Ma', 'Ba Dinh', 'HN', '1990-07-15', 'anna.white@abc.com', 'HN01'),  
(1003, 'David', 'Johnson', '56', 'Tran Phu', 'Hai Chau', 'DN', '2010-03-20', 'david.j@abc.com', 'DN01'),  
(1004, 'Lisa', 'Green', '12', 'Hoa Binh', 'Ninh Kieu', 'CT', '2010-03-25', 'lisa.g@abc.com', 'CT01');  
  
-- Customer  
INSERT INTO Customer VALUES  
(2001, 'Peter', 'Johnson', 'FPT Tower HN', '12 Hoang Hoa Tham HN', 'peter.j@abc.com', 1002),  
(2002, 'Mary', 'Brown', 'Viettel HCM', '45 Tran Hung Dao HCM', 'mary.b@abc.com', 1001),  
(2003, 'Tom', 'Wilson', 'Sunrise DN', '56 Hai Phong DN', 'tom.w@abc.com', 1003),  
(2004, 'Laura', 'Nguyen', 'VinGroup HN', '89 Nguyen Chi Thanh HN', 'laura.n@abc.com', 1004);  
  
-- Account  
INSERT INTO Account VALUES  
(3001, 2001),  
(3002, 2001),  
(3003, 2002),  
(3004, 2003);  
  
-- Saving Account  
INSERT INTO SavingAccount VALUES  
(3001, 10000, 0.05, '2022-01-01'),  
(3002, 5000, 0.04, '2023-01-10'),  
(3003, 15000, 0.03, '2023-02-20'),  
(3004, 20000, 0.04, '2024-03-01');  
  
-- Checking Account  
INSERT INTO CheckingAccount VALUES  
(3002, 2000, '2023-01-10'),  
(3003, 1000, '2023-02-20'),  
(3004, 500, '2024-03-01'),  
(3001, 300, '2022-01-01');  
  
-- Loan  
INSERT INTO Loan VALUES  
(3001, 20000, 0.07, '2023-05-15'),  
(3002, 15000, 0.06, '2023-06-10'),  
(3003, 10000, 0.08, '2024-02-20'),  
(3004, 5000, 0.07, '2024-03-25');  
  
-- Branch Phone  
INSERT INTO BranchPhone VALUES  
('HCM01', '028-1111-2222'),  
('HN01', '024-3333-4444'),  
('DN01', '0236-5555-6666'),  
('CT01', '0292-7777-8888');  
  
-- Branch Fax  
INSERT INTO BranchFax VALUES  
('HCM01', '028-9999-1111'),  
('HN01', '024-8888-2222'),  
('DN01', '0236-7777-3333'),  
('CT01', '0292-6666-4444');  
  
-- Employee Phone  
INSERT INTO EmployeePhone VALUES  
(1001, '0901-111-111'),  
(1002, '0902-222-222'),  
(1003, '0903-333-333'),  
(1004, '0904-444-444');  
  
-- Customer Phone  
INSERT INTO CustomerPhone VALUES  
(2001, '0911-111-111'),  
(2002, '0912-222-222'),  
(2003, '0913-333-333'),  
(2004, '0914-444-444');

b. Change last name of an employee whose code is “N1080” to “Brown”.

UPDATE Employee  
SET LName = 'BROWN'  
WHERE Code = 'N1080'

c. Delete the employee having code “N1080”. Explain what will happen to the customers who are

severed by this employee.

DELETE FROM Employee  
WHERE Code = 'N1080';

Giải thích:

Xảy ra 2 trường hợp

- Nếu employee có một ràng buộc nào đó với customer thì sẽ không xoá được vì dính tới foreign key, chương trình sẽ bắn ra lỗi.

- Nếu code employee không là khoá ngoài của customer nào đó thì sẽ xoá bình thường.

d. Retrieve all account numbers of a customer whose name is Peter Johnson.

/\*  
Retrieve all account numbers of a customer whose name is Peter Johnson.  
\*/  
SELECT A.Number  
FROM `Account` AS A, Customer AS C  
WHERE C.LName = 'Johnson'  
AND C.FName = 'Peter'  
AND C.Code = A.CusCode

e. Find all employees who have their birthdays in March 2010.

SELECT \*  
FROM Employee  
WHERE *YEAR*(Employee.DOB ) = 2010  
AND *MONTH*(Employee.DOB) = 3;

f. Retrieve the total balance of all saving accounts owned by a customer named “Peter Johnson”

(Supposing that he has many saving accounts)

SELECT *SUM*(SV.Balance) AS 'Total balance of customer named Peter Johnson'  
FROM Account AS A, Customer AS C, SavingAccount AS SV  
WHERE C.FName = 'Peter' AND C.LName = 'Johnson'  
AND A.CusCode = C.Code  
AND SV.AccNumber = A.Number