**Bộ giáo dục và đào tạo**

**Trường Đại học Sài Gòn**

--🙢🕮🙠--



**Báo cáo: Đồ án cuối kì**

**Môn: Công nghệ phần mềm**

# [BaiTap] Thiết kế cơ sở dữ liệu

Sinh viên thực hiện: Đặng Nguyễn Quốc Dương MSSV:3121411045

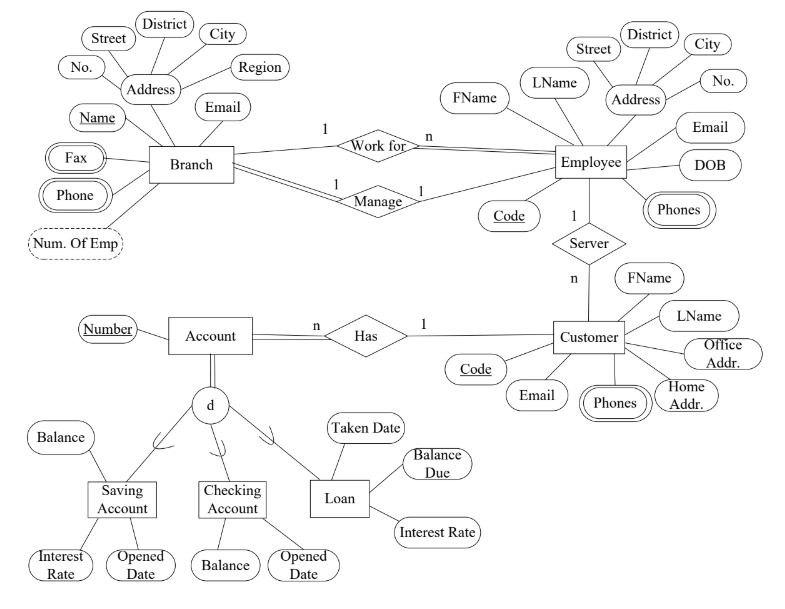
Giáo viên hướng dẫn:TS.Đỗ Như Tài

**TP.HCM,Tháng 4, năm 202**

**QUESTION 1. ABC BANK**

**1. Tạo bảng:**

**a) ERD**



**b) Mức luận lý (Lược đồ quan hệ)**

BRANCH (Name, AddressNo, Street, District, City, Region, Email, MngCode)

BRANCHPHONE (BraName, Phone)

BRANCHFAX (BraName, Fax)

EMPLOYEE (Code, FName, LName, AddressNo, Street, District, City, DOB, Email, BraName)

EMPLOYEEPHONE (EmpCode, Phone)

CUSTOMER (Code, FName, LName, OfficeAddress, HomeAddress, Email, EmpCode)

CUSTOMERPHONE (CusCode, Phone)

ACCOUNT (Number, CusCode)

SAVINGACCOUNT (AccNumber, Balance, InterestRate, OpenedDate)

CHECKINGACCOUNT (AccNumber, Balance, OpenedDate)

LOAN (AccNumber, BalanceDue, InterestRate, TakenDate)

**c) Mức vật lý(tạo bảng)**

-- BRANCH

CREATE TABLE BRANCH (

Name VARCHAR(50) PRIMARY KEY,

AddressNo VARCHAR(10),

Street VARCHAR(100),

District VARCHAR(50),

City VARCHAR(50),

Region VARCHAR(50),

Email VARCHAR(100),

MngCode VARCHAR(20)

);

-- BRANCHPHONE

CREATE TABLE BRANCHPHONE (

BraName VARCHAR(50),

Phone VARCHAR(15),

FOREIGN KEY (BraName) REFERENCES BRANCH(Name)

);

-- BRANCHFAX

CREATE TABLE BRANCHFAX (

BraName VARCHAR(50),

Fax VARCHAR(15),

FOREIGN KEY (BraName) REFERENCES BRANCH(Name)

);

-- EMPLOYEE

CREATE TABLE EMPLOYEE (

Code VARCHAR(20) PRIMARY KEY,

FName VARCHAR(50),

LName VARCHAR(50),

AddressNo VARCHAR(10),

Street VARCHAR(100),

District VARCHAR(50),

City VARCHAR(50),

DOB DATE,

Email VARCHAR(100),

BraName VARCHAR(50),

FOREIGN KEY (BraName) REFERENCES BRANCH(Name)

);

-- EMPLOYEEPHONE

CREATE TABLE EMPLOYEEPHONE (

EmpCode VARCHAR(20),

Phone VARCHAR(15),

FOREIGN KEY (EmpCode) REFERENCES EMPLOYEE(Code)

);

-- CUSTOMERPHONE

CREATE TABLE CUSTOMERPHONE (

CusCode VARCHAR(20),

Phone VARCHAR(15),

FOREIGN KEY (CusCode) REFERENCES CUSTOMER(Code)

);

-- CUSTOMER

CREATE TABLE CUSTOMER (

Code VARCHAR(20) PRIMARY KEY,

FName VARCHAR(50),

LName VARCHAR(50),

OfficeAddress VARCHAR(200),

HomeAddress VARCHAR(200),

Email VARCHAR(100),

EmpCode VARCHAR(20),

FOREIGN KEY (EmpCode) REFERENCES EMPLOYEE(Code)

);

-- ACCOUNT

CREATE TABLE ACCOUNT (

Number VARCHAR(20) PRIMARY KEY,

CusCode VARCHAR(20),

FOREIGN KEY (CusCode) REFERENCES CUSTOMER(Code)

);

-- SAVINGACCOUNT

CREATE TABLE SAVINGACCOUNT (

AccNumber VARCHAR(20) PRIMARY KEY,

Balance MONEY,

InterestRate FLOAT,

OpenedDate DATE,

FOREIGN KEY (AccNumber) REFERENCES ACCOUNT(Number)

);

-- CHECKINGACCOUNT

CREATE TABLE CHECKINGACCOUNT (

AccNumber VARCHAR(20) PRIMARY KEY,

Balance MONEY,

OpenedDate DATE,

FOREIGN KEY (AccNumber) REFERENCES ACCOUNT(Number)

);

-- LOAN

CREATE TABLE LOAN (

AccNumber VARCHAR(20) PRIMARY KEY,

BalanceDue MONEY,

InterestRate FLOAT,

TakenDate DATE,

FOREIGN KEY (AccNumber) REFERENCES ACCOUNT(Number)

);

**2. Query: Insert, Update, Delete, Select**

**a) thêm dữ liệu**

-- BRANCH

INSERT INTO BRANCH VALUES

('BranchA', '123', 'Main St', 'District 1', 'HCMC', 'South', 'brancha@bank.com', 'MNG001'),

('BranchB', '456', 'Second St', 'District 2', 'HCMC', 'South', 'branchb@bank.com', 'MNG002'),

('BranchC', '789', 'Third St', 'District 3', 'Hanoi', 'North', 'branchc@bank.com', 'MNG003'),

('BranchD', '101', 'Fourth St', 'District 4', 'Danang', 'Central', 'branchd@bank.com', 'MNG004'),

('BranchE', '202', 'Fifth St', 'District 5', 'Can Tho', 'South', 'branche@bank.com', 'MNG005');

-- BRANCHPHONE

INSERT INTO BRANCHPHONE VALUES

('BranchA', '0281234567'),

('BranchB', '0282345678'),

('BranchC', '0243456789'),

('BranchD', '0234567890'),

('BranchE', '0298765432');

-- BRANCHFAX

INSERT INTO BRANCHFAX VALUES

('BranchA', '0287654321'),

('BranchB', '0288765432'),

('BranchC', '0249876543'),

('BranchD', '0234987654'),

('BranchE', '0291122334');

-- EMPLOYEE

INSERT INTO EMPLOYEE VALUES

('EMP001', 'John', 'Doe', '1', 'Main St', 'District 1', 'HCMC', '1990-01-01', 'john.doe@bank.com', 'BranchA'),

('EMP002', 'Jane', 'Smith', '2', 'Second St', 'District 2', 'HCMC', '1985-02-02', 'jane.smith@bank.com', 'BranchB'),

('EMP003', 'Alice', 'Brown', '3', 'Third St', 'District 3', 'Hanoi', '1992-03-03', 'alice.brown@bank.com', 'BranchC'),

('EMP004', 'Bob', 'White', '4', 'Fourth St', 'District 4', 'Danang', '1988-04-04', 'bob.white@bank.com', 'BranchD'),

('EMP005', 'Carol', 'Black', '5', 'Fifth St', 'District 5', 'Can Tho', '1995-05-05', 'carol.black@bank.com', 'BranchE');

-- EMPLOYEEPHONE

INSERT INTO EMPLOYEEPHONE VALUES

('EMP001', '0901111111'),

('EMP002', '0902222222'),

('EMP003', '0903333333'),

('EMP004', '0904444444'),

('EMP005', '0905555555');

-- CUSTOMER

INSERT INTO CUSTOMER VALUES

('C001', 'Tom', 'Lee', '1 Office St', '1 Home St', 'tom.lee@mail.com', 'EMP001'),

('C002', 'Jerry', 'Nguyen', '2 Office St', '2 Home St', 'jerry.nguyen@mail.com', 'EMP002'),

('C003', 'Lily', 'Tran', '3 Office St', '3 Home St', 'lily.tran@mail.com', 'EMP003'),

('C004', 'Henry', 'Pham', '4 Office St', '4 Home St', 'henry.pham@mail.com', 'EMP004'),

('C005', 'Anna', 'Vo', '5 Office St', '5 Home St', 'anna.vo@mail.com', 'EMP005');

-- CUSTOMERPHONE

INSERT INTO CUSTOMERPHONE VALUES

('C001', '0911111111'),

('C002', '0912222222'),

('C003', '0913333333'),

('C004', '0914444444'),

('C005', '0915555555');

-- ACCOUNT

INSERT INTO ACCOUNT VALUES

('A001', 'C001'),

('A002', 'C002'),

('A003', 'C003'),

('A004', 'C004'),

('A005', 'C005');

-- SAVINGACCOUNT

INSERT INTO SAVINGACCOUNT VALUES

('A001', 5000000, 5.0, '2023-01-01'),

('A002', 10000000, 4.8, '2023-02-01'),

('A003', 7000000, 5.2, '2023-03-01'),

('A004', 8000000, 5.1, '2023-04-01'),

('A005', 9000000, 4.9, '2023-05-01');

-- CHECKINGACCOUNT

INSERT INTO CHECKINGACCOUNT VALUES

('A001', 2000000, '2023-01-01'),

('A002', 1500000, '2023-02-01'),

('A003', 2500000, '2023-03-01'),

('A004', 3000000, '2023-04-01'),

('A005', 1000000, '2023-05-01');

-- LOAN

INSERT INTO LOAN VALUES

('A001', 10000000, 7.0, '2023-06-01'),

('A002', 12000000, 6.8, '2023-06-15'),

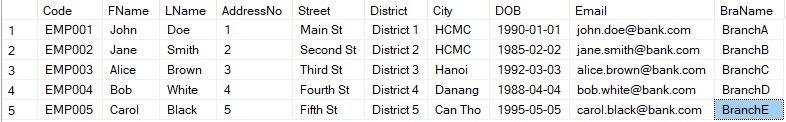
('A003', 15000000, 7.2, '2023-07-01'),

('A004', 8000000, 6.5, '2023-07-15'),

('A005', 11000000, 7.1, '2023-08-01');

**b) Đổi họ của nhân viên có mã "EMP002" thành "Brown"**

**Trước khi thay đổi:**



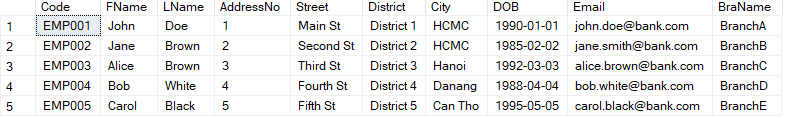
**Hàm thay đổi:**

UPDATE EMPLOYEE

SET LName = 'Brown'

WHERE Code =’ EMP002’;

**Sau khi thay đổi:**



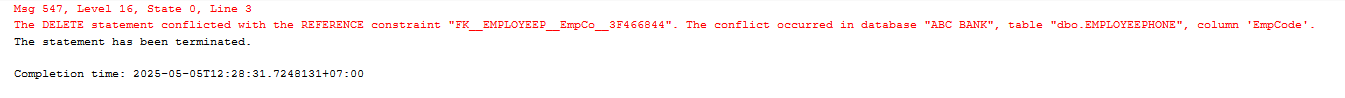
**c) Xóa nhân viên có mã "EMP002" + Giải thích**

**Hàm thay đổi:**

DELETE FROM EMPLOYEE

WHERE Code = 'N1080';

**Kết quả:**



**Giải thích:**

Bạn đang cố xóa một bản ghi trong bảng EMPLOYEE.

Nhưng có một hoặc nhiều bản ghi trong bảng EMPLOYEEPHONE đang tham chiếu đến dòng đó thông qua cột EmpCode.

Ràng buộc FOREIGN KEY FK\_\_EMPLOYEEP\_\_EmpCo\_\_3F466844 ngăn không cho bạn xóa nhân viên nếu vẫn còn số điện thoại liên kết với người đó trong bảng EMPLOYEEPHONE.

**d) Truy vấn tất cả số tài khoản của khách hàng tên "Peter Johnson"**

**Hàm truy vấn:**

SELECT A.Number

FROM ACCOUNT A

JOIN CUSTOMER C ON A.CusCode = C.Code

WHERE C.FName = 'Anna' AND C.LName = 'Vo';

**Kết quả:**



**e) Tìm nhân viên có sinh nhật trong tháng 3 năm 2010**

**Hàm truy vấn:**

SELECT \*

FROM EMPLOYEE

WHERE MONTH(DOB) = 3 AND YEAR(DOB) = 2010;

**Kết quả:**

None, vì bảng không có nhân viên sinh tháng 3 năm 2010

**f) Tổng số dư các tài khoản tiết kiệm của khách hàng "Peter Johnson"**

**Hàm truy vấn:**

SELECT SUM(SA.Balance) AS TotalBalance

FROM SAVINGACCOUNT SA

JOIN ACCOUNT A ON SA.AccNumber = A.Number

JOIN CUSTOMER C ON A.CusCode = C.Code

WHERE C.FName = 'Peter' AND C.LName = 'Johnson';

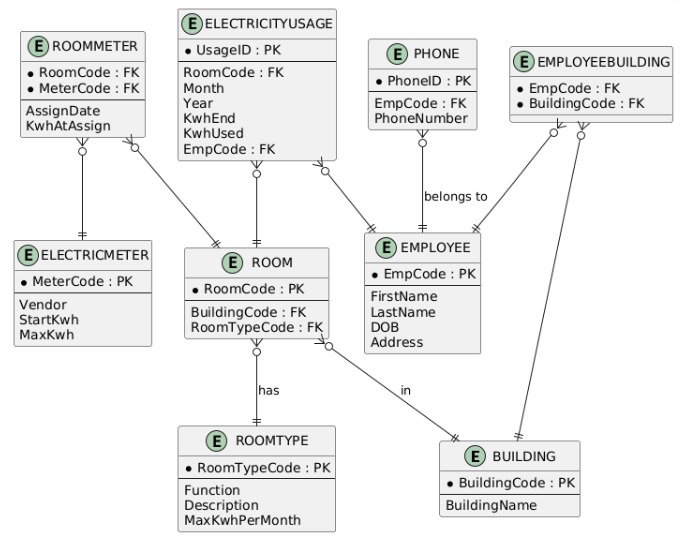
**Kết quả:**



**QUESTION 2. Electricity Management**

**1. Thiết kế cơ sở dữ liệu**

**a) ERD**



**b) Mức luận lý (Lược đồ quan hệ)**

BUILDING(BuildingCode PK, BuildingName)

ROOM(RoomCode PK, BuildingCode FK, RoomTypeCode FK)

ROOMTYPE(RoomTypeCode PK, Function, Description, MaxKwhPerMonth)

EMPLOYEE(EmpCode PK, FirstName, LastName, DOB, Address)

PHONE(PhoneID PK, EmpCode FK, PhoneNumber)

EMPLOYEEBUILDING(EmpCode FK, BuildingCode FK)

ELECTRICMETER(MeterCode PK, Vendor, StartKwh, MaxKwh)

ROOMMETER(RoomCode FK, MeterCode FK, AssignDate, KwhAtAssign)

ELECTRICITYUSAGE(UsageID PK, RoomCode FK, Month, Year, KwhEnd, KwhUsed, EmpCode FK)

**c) mực vật lý tạo bảng**

CREATE TABLE Building (

BuildingCode VARCHAR(10) PRIMARY KEY,

BuildingName NVARCHAR(100)

);

CREATE TABLE RoomType (

RoomTypeCode VARCHAR(10) PRIMARY KEY,

RoomTypeFunction NVARCHAR(50),

Description NVARCHAR(100),

MaxKwhPerMonth INT

);

CREATE TABLE Room (

RoomCode VARCHAR(10) PRIMARY KEY,

BuildingCode VARCHAR(10),

RoomTypeCode VARCHAR(10),

FOREIGN KEY (BuildingCode) REFERENCES Building(BuildingCode),

FOREIGN KEY (RoomTypeCode) REFERENCES RoomType(RoomTypeCode)

);

CREATE TABLE Employee (

EmpCode VARCHAR(10) PRIMARY KEY,

FirstName NVARCHAR(50),

LastName NVARCHAR(50),

DOB DATE,

Address NVARCHAR(100)

);

CREATE TABLE Phone (

PhoneID INT PRIMARY KEY,

EmpCode VARCHAR(10),

PhoneNumber VARCHAR(20),

FOREIGN KEY (EmpCode) REFERENCES Employee(EmpCode)

);

CREATE TABLE EmployeeBuilding (

EmpCode VARCHAR(10),

BuildingCode VARCHAR(10),

PRIMARY KEY (EmpCode, BuildingCode),

FOREIGN KEY (EmpCode) REFERENCES Employee(EmpCode),

FOREIGN KEY (BuildingCode) REFERENCES Building(BuildingCode)

);

CREATE TABLE ElectricMeter (

MeterCode VARCHAR(10) PRIMARY KEY,

Vendor NVARCHAR(50),

StartKwh INT,

MaxKwh INT

);

CREATE TABLE RoomMeter (

RoomCode VARCHAR(10),

MeterCode VARCHAR(10),

AssignDate DATE,

KwhAtAssign INT,

PRIMARY KEY (RoomCode, MeterCode),

FOREIGN KEY (RoomCode) REFERENCES Room(RoomCode),

FOREIGN KEY (MeterCode) REFERENCES ElectricMeter(MeterCode)

);

CREATE TABLE ElectricityUsage (

UsageID INT PRIMARY KEY,

RoomCode VARCHAR(10),

Month INT,

Year INT,

KwhEnd INT,

KwhUsed INT,

EmpCode VARCHAR(10),

FOREIGN KEY (RoomCode) REFERENCES Room(RoomCode),

FOREIGN KEY (EmpCode) REFERENCES Employee(EmpCode)

);

**2. Truy vấn SQL**

**a. Thêm dữ liệu vào mỗi bảng (ít nhất 4 dòng)**

-- RoomType

INSERT INTO RoomType VALUES ('RT01', 'Computer Lab', 'Phòng lab máy tính', 1000);

INSERT INTO RoomType VALUES ('RT02', 'Office', 'Văn phòng', 500);

INSERT INTO RoomType VALUES ('RT03', 'Small Study', 'Phòng học nhỏ', 300);

INSERT INTO RoomType VALUES ('RT04', 'Large Study', 'Phòng học lớn', 800);

-- Building

INSERT INTO Building VALUES ('B1', 'Building A');

INSERT INTO Building VALUES ('B2', 'Building B');

INSERT INTO Building VALUES ('B3', 'Building C');

INSERT INTO Building VALUES ('B4', 'Building D');

-- Employee

INSERT INTO Employee VALUES ('N1080', 'Peter', 'Johnson', '1970-01-01', '123 Main St');

INSERT INTO Employee VALUES ('N1081', 'Anna', 'Lee', '1980-02-02', '456 Park Ave');

INSERT INTO Employee VALUES ('N1082', 'John', 'Smith', '1990-03-03', '789 King Rd');

INSERT INTO Employee VALUES ('N1083', 'Sarah', 'Brown', '1985-04-04', '987 Queen St');

-- Phone

INSERT INTO Phone VALUES (1, 'N1080', '0123456789');

INSERT INTO Phone VALUES (2, 'N1080', '0987654321');

INSERT INTO Phone VALUES (3, 'N1081', '0223344556');

INSERT INTO Phone VALUES (4, 'N1082', '0111222333');

-- EmployeeBuilding

INSERT INTO EmployeeBuilding VALUES ('N1080', 'B4');

INSERT INTO EmployeeBuilding VALUES ('N1081', 'B1');

INSERT INTO EmployeeBuilding VALUES ('N1082', 'B2');

INSERT INTO EmployeeBuilding VALUES ('N1083', 'B3');

-- Room

INSERT INTO Room VALUES ('R1', 'B4', 'RT01');

INSERT INTO Room VALUES ('R2', 'B4', 'RT02');

INSERT INTO Room VALUES ('R3', 'B1', 'RT01');

INSERT INTO Room VALUES ('R4', 'B2', 'RT03');

-- ElectricMeter

INSERT INTO ElectricMeter VALUES ('M1', 'EVN', 0, 1200);

INSERT INTO ElectricMeter VALUES ('M2', 'EVN', 100, 1500);

INSERT INTO ElectricMeter VALUES ('M3', 'GE', 200, 1800);

INSERT INTO ElectricMeter VALUES ('M4', 'Siemens', 50, 1300);

-- RoomMeter

INSERT INTO RoomMeter VALUES ('R1', 'M1', '2010-03-01', 0);

INSERT INTO RoomMeter VALUES ('R2', 'M2', '2010-03-01', 100);

INSERT INTO RoomMeter VALUES ('R3', 'M3', '2010-03-01', 200);

INSERT INTO RoomMeter VALUES ('R4', 'M4', '2010-03-01', 50);

-- ElectricityUsage

INSERT INTO ElectricityUsage VALUES (1, 'R1', 3, 2010, 300, 300, 'N1080');

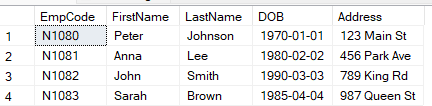
INSERT INTO ElectricityUsage VALUES (2, 'R2', 3, 2010, 350, 250, 'N1080');

INSERT INTO ElectricityUsage VALUES (3, 'R3', 3, 2010, 600, 400, 'N1081');

INSERT INTO ElectricityUsage VALUES (4, 'R4', 3, 2010, 100, 50, 'N1082');

**b. Update DOB của nhân viên có mã "N1080"**

**Trước khi thay đổi:**



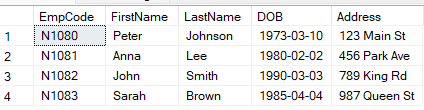
**Hàm thay đổi:**

UPDATE Employee

SET DOB = '1973-03-10'

WHERE EmpCode = 'N1080';

**Kết quả:**



**c. Xóa nhân viên có mã "N1080" + Giải thích**

**Hàm thay đổi:**

DELETE FROM Employee WHERE EmpCode = 'N1080';

**Kết quả:**



**Giải thích:** có ràng buộc khóa ngoại từ EmployeeBuilding hoặc ElectricityUsage đến Employee, thì bạn sẽ nhận lỗi vi phạm khóa ngoại.

**d) Truy vấn thông tin nhân viên quản lý tòa B4**

**Hàm truy vấn:**

SELECT e.EmpCode, e.FirstName, e.LastName, e.Address, e.DOB, p.PhoneNumber

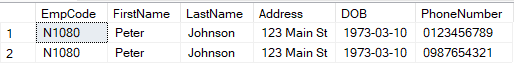
FROM Employee e

JOIN EmployeeBuilding eb ON e.EmpCode = eb.EmpCode

JOIN Phone p ON e.EmpCode = p.EmpCode

WHERE eb.BuildingCode = 'B4';

**Kết quả:**



**e) Các phòng loại "Computer Lab" được quản lý bởi Peter Johnson**

**Hàm truy vấn:**

SELECT r.\*

FROM Room r

JOIN RoomType rt ON r.RoomTypeCode = rt.RoomTypeCode

JOIN Building b ON r.BuildingCode = b.BuildingCode

JOIN EmployeeBuilding eb ON b.BuildingCode = eb.BuildingCode

JOIN Employee e ON eb.EmpCode = e.EmpCode

WHERE rt.RoomTypeFunction = 'Computer Lab'

AND e.FirstName = 'Peter' AND e.LastName = 'Johnson';

**Kết quả:**



**f) Tính tổng Kwh sử dụng trong tháng 3 năm 2010 của các phòng “Computer Lab”**

**Hàm truy vấn:**

SELECT SUM(eu.KwhUsed) AS TotalKwh

FROM ElectricityUsage eu

JOIN Room r ON eu.RoomCode = r.RoomCode

JOIN RoomType rt ON r.RoomTypeCode = rt.RoomTypeCode

WHERE rt.RoomTypeFunction = 'Computer Lab'

AND eu.Month = 3 AND eu.Year = 2010;

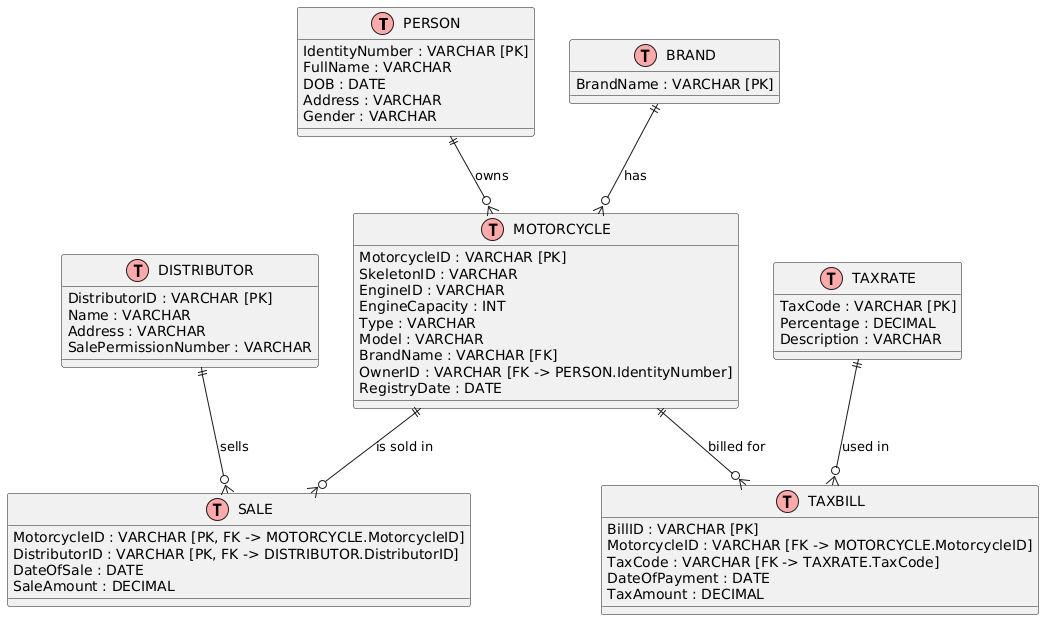
**Kết quả:**



**QUESTION 3. Motorcycle Registration Department**

**1. Thiết kế cơ sở dữ liệu**

**a. ERD**



**b. Mức luận lý**

PERSON (IdentityNumber, FullName, DOB, Address, Gender)

MOTORCYCLE (MotorcycleID, SkeletonID, EngineID, EngineCapacity, Type, Model, BrandName, OwnerID, RegistryDate)

BRAND (BrandName)

DISTRIBUTOR (DistributorID, Name, Address, SalePermissionNumber)

SALE (MotorcycleID, DistributorID, DateOfSale, SaleAmount)

TAXRATE (TaxCode, Percentage, Description)

TAXBILL (BillID, MotorcycleID, TaxCode, DateOfPayment, TaxAmount)

**c. Mức vật lý**

CREATE TABLE PERSON (

IdentityNumber VARCHAR(20) PRIMARY KEY,

FullName NVARCHAR(100),

DOB DATE,

Address NVARCHAR(200),

Gender CHAR(1)

);

CREATE TABLE BRAND (

BrandName VARCHAR(50) PRIMARY KEY

);

CREATE TABLE DISTRIBUTOR (

DistributorID VARCHAR(10) PRIMARY KEY,

Name NVARCHAR(100),

Address NVARCHAR(200),

SalePermissionNumber VARCHAR(20)

);

CREATE TABLE MOTORCYCLE (

MotorcycleID VARCHAR(20) PRIMARY KEY,

SkeletonID VARCHAR(30),

EngineID VARCHAR(30),

EngineCapacity INT,

Type NVARCHAR(50),

Model NVARCHAR(50),

BrandName VARCHAR(50),

OwnerID VARCHAR(20),

RegistryDate DATE,

FOREIGN KEY (BrandName) REFERENCES BRAND(BrandName),

FOREIGN KEY (OwnerID) REFERENCES PERSON(IdentityNumber)

);

CREATE TABLE SALE (

MotorcycleID VARCHAR(20),

DistributorID VARCHAR(10),

DateOfSale DATE,

SaleAmount MONEY,

PRIMARY KEY (MotorcycleID),

FOREIGN KEY (MotorcycleID) REFERENCES MOTORCYCLE(MotorcycleID),

FOREIGN KEY (DistributorID) REFERENCES DISTRIBUTOR(DistributorID)

);

CREATE TABLE TAXRATE (

TaxCode VARCHAR(10) PRIMARY KEY,

Percentage FLOAT,

Description NVARCHAR(200)

);

CREATE TABLE TAXBILL (

BillID VARCHAR(20) PRIMARY KEY,

MotorcycleID VARCHAR(20),

TaxCode VARCHAR(10),

DateOfPayment DATE,

TaxAmount MONEY,

FOREIGN KEY (MotorcycleID) REFERENCES MOTORCYCLE(MotorcycleID),

FOREIGN KEY (TaxCode) REFERENCES TAXRATE(TaxCode)

);

**2. Query – Insert, Update, Delete, Select**

**a. Thêm dữ liệu mẫu vào các bảng (ít nhất 4 dòng mỗi bảng)**

-- BRAND

INSERT INTO BRAND VALUES ('Honda'), ('Yamaha'), ('VMEP'), ('SYM');

-- PERSON

INSERT INTO PERSON VALUES

('ID001', 'Nguyen Van A', '1980-05-01', '123 ABC St', 'M'),

('ID002', 'Tran Thi B', '1990-10-12', '456 DEF St', 'F'),

('ID003', 'Le Van C', '1975-03-20', '789 XYZ St', 'M'),

('ID004', 'Pham Thi D', '1995-08-15', '321 GHI St', 'F');

-- DISTRIBUTOR

INSERT INTO DISTRIBUTOR VALUES

('D001', 'Tung Motor', '12 District 1', 'SP001'),

('D002', 'Phuoc Motor', '34 District 2', 'SP002'),

('D003', 'Thanh Motor', '56 District 3', 'SP003'),

('D004', 'Viet Motor', '78 District 4', 'SP004');

-- MOTORCYCLE

INSERT INTO MOTORCYCLE VALUES

('MC001', 'SK001', 'EN001', 110, 'Scooter', 'Vision', 'VMEP', 'ID001', '2010-03-15'),

('MC002', 'SK002', 'EN002', 150, 'Sport', 'Winner', 'Honda', 'ID002', '2011-06-20'),

('MC003', 'SK003', 'EN003', 125, 'Cub', 'Sirius', 'Yamaha', 'ID003', '2009-12-01'),

('MC004', 'SK004', 'EN004', 100, 'Standard', 'Joyride', 'SYM', 'ID004', '2010-03-05');

-- SALE

INSERT INTO SALE VALUES

('MC001', 'D001', '2009-12-01', 2500),

('MC002', 'D002', '2010-01-01', 3500),

('MC003', 'D001', '1989-01-01', 1200), -- để test DELETE

('MC004', 'D003', '2010-03-01', 2800);

-- TAXRATE

INSERT INTO TAXRATE VALUES

('TX01', 0.1, 'Standard rate'),

('TX02', 0.08, 'Discounted'),

('TX03', 0.12, 'Luxury'),

('TX04', 0.05, 'Old model');

-- TAXBILL

INSERT INTO TAXBILL VALUES

('B001', 'MC001', 'TX01', '2010-03-16', 250),

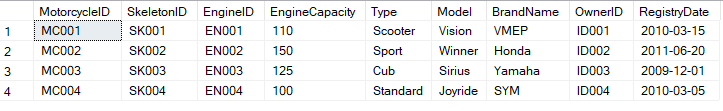
('B002', 'MC002', 'TX01', '2010-01-10', 350),

('B003', 'MC003', 'TX04', '1989-01-05', 60),

('B004', 'MC004', 'TX02', '2010-03-15', 224);

**b) Cập nhật branch name của một xe máy từ "VMEP" thành "SYM"**

**Trước khi thay đổi:**



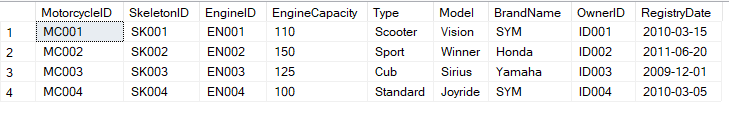
**Hàm thay đổi:**

UPDATE MOTORCYCLE

SET BrandName = 'SYM'

WHERE BrandName = 'VMEP';

**Kết quả:**



**c. Xóa xe có ngày bán trước 04/05/1990. Giải thích hậu quả**

**Hàm thay đổi:**

DELETE FROM MOTORCYCLE

WHERE MotorcycleID IN (

SELECT MotorcycleID FROM SALE WHERE DateOfSale < '1990-05-04'

);

**Kết quả:**



**Giải thích:** Nếu MOTORCYCLEID được tham chiếu bởi TAXBILL, việc xóa sẽ bị lỗi do ràng buộc khóa ngoại.

**Giải pháp:** xóa TAXBILL trước → SALE → MOTORCYCLE hoặc dùng ON DELETE CASCADE.

**d. Truy vấn thông tin distributor bán nhiều xe nhất**

**Hàm truy vấn:**

SELECT TOP 1 D.DistributorID, D.Name, COUNT(S.MotorcycleID) AS NumSold

FROM SALE S

JOIN DISTRIBUTOR D ON S.DistributorID = D.DistributorID

GROUP BY D.DistributorID, D.Name

ORDER BY NumSold DESC;

**Kết quả:**



**e. Liệt kê các hãng xe trả cùng mức thuế với “Honda”**

**Hàm truy vấn:**

SELECT DISTINCT M.BrandName

FROM MOTORCYCLE M

JOIN TAXBILL TB1 ON M.MotorcycleID = TB1.MotorcycleID

WHERE TB1.TaxCode IN (

SELECT TB2.TaxCode

FROM MOTORCYCLE M2

JOIN TAXBILL TB2 ON M2.MotorcycleID = TB2.MotorcycleID

WHERE M2.BrandName = 'Honda'

) AND M.BrandName <> 'Honda';

**Kết quả:**



**f. Tính tổng tiền thuế của các xe Yamaha đăng ký tháng 3/2010**

**Hàm truy vấn:**

SELECT SUM(TB.TaxAmount) AS TotalTax

FROM TAXBILL TB

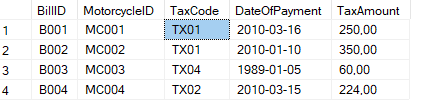
JOIN MOTORCYCLE M ON TB.MotorcycleID = M.MotorcycleID

WHERE M.BrandName = 'Yamaha' AND MONTH(TB.DateOfPayment) = 3 AND YEAR(TB.DateOfPayment) = 2010;

**Kết quả:**



**Giải thích:**

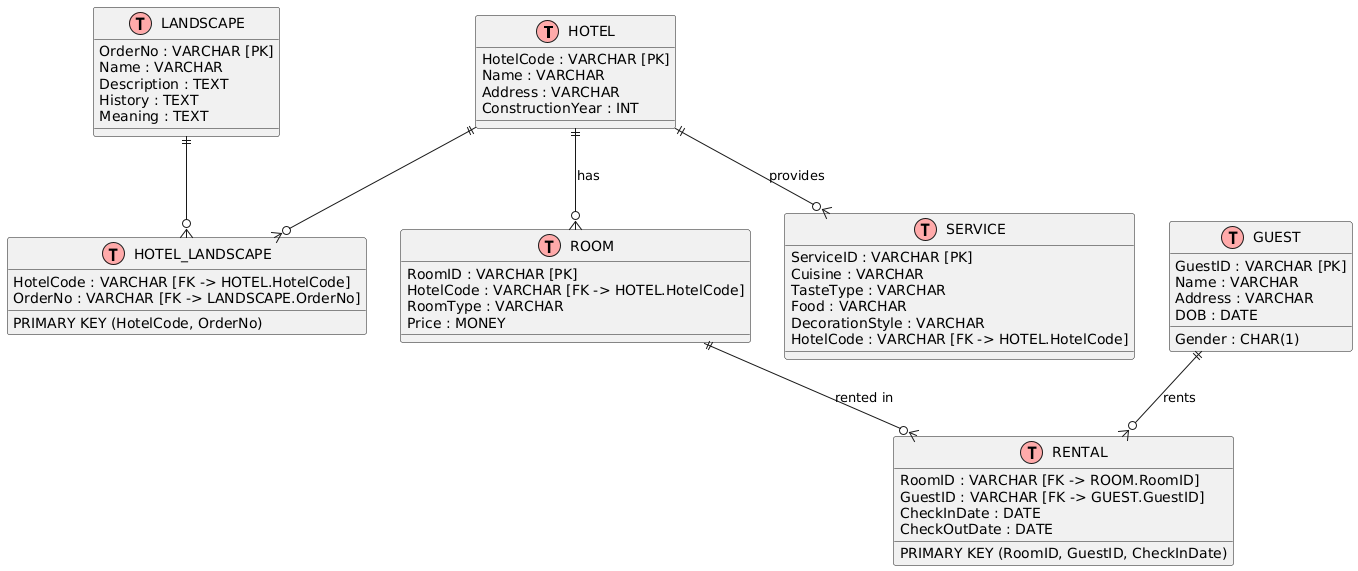


Vì trong bảng có **MotocrycleID** = MC003 tức là xe yamaha không nằm ở tháng 3/2010 mà nằm ở tháng 1/1989.

**QUESTION 4. Hotel in Oslo**

**1. Thiết kế Cơ sở dữ liệu**

**a) ERD**



**b. Mức luận lý**

HOTEL(HotelCode PK, Name, Address, ConstructionYear)

SERVICE(ServiceID PK, Cuisine, TasteType, Food, DecorationStyle, HotelCode FK)

LANDSCAPE(OrderNo PK, Name, Description, History, Meaning)

HOTEL\_LANDSCAPE(HotelCode FK, OrderNo FK)

ROOM(RoomID PK, HotelCode FK, RoomType, Price)

GUEST(GuestID PK, Name, Gender, Address, DOB)

RENTAL(RoomID FK, GuestID FK, CheckInDate, CheckOutDate)

**c. Mức vật lý**

CREATE TABLE HOTEL (

HotelCode INT PRIMARY KEY,

Name VARCHAR(50),

Address VARCHAR(100),

ConstructionYear INT

);

CREATE TABLE SERVICE (

ServiceID INT PRIMARY KEY,

Cuisine VARCHAR(50),

TasteType VARCHAR(50),

Food VARCHAR(50),

DecorationStyle VARCHAR(50),

HotelCode INT FOREIGN KEY REFERENCES HOTEL(HotelCode)

);

CREATE TABLE LANDSCAPE (

OrderNo INT PRIMARY KEY,

Name VARCHAR(50),

Description TEXT,

History TEXT,

Meaning TEXT

);

CREATE TABLE HOTEL\_LANDSCAPE (

HotelCode INT FOREIGN KEY REFERENCES HOTEL(HotelCode),

OrderNo INT FOREIGN KEY REFERENCES LANDSCAPE(OrderNo),

PRIMARY KEY (HotelCode, OrderNo)

);

CREATE TABLE ROOM (

RoomID INT PRIMARY KEY,

HotelCode INT FOREIGN KEY REFERENCES HOTEL(HotelCode),

RoomType VARCHAR(50),

Price DECIMAL(10,2)

);

CREATE TABLE GUEST (

GuestID INT PRIMARY KEY,

Name VARCHAR(50),

Gender VARCHAR(10),

Address VARCHAR(100),

DOB DATE

);

CREATE TABLE RENTAL (

RoomID INT FOREIGN KEY REFERENCES ROOM(RoomID),

GuestID INT FOREIGN KEY REFERENCES GUEST(GuestID),

CheckInDate DATE,

CheckOutDate DATE,

PRIMARY KEY(RoomID, GuestID, CheckInDate)

);

**2. Query: Insert, Update, Delete, Select**

**a. Insert dữ liệu mẫu**

-- HOTEL

INSERT INTO HOTEL VALUES (1, 'Rex', 'Oslo Center', 2001);

INSERT INTO HOTEL VALUES (2, 'Grand', 'Oslo East', 1998);

INSERT INTO HOTEL VALUES (3, 'Ocean View', 'Oslo Bay', 2005);

INSERT INTO HOTEL VALUES (4, 'Nordic Light', 'Oslo North', 2010);

-- SERVICE

INSERT INTO SERVICE VALUES (101, 'Asian', 'Spicy', 'Pho', 'Traditional', 1);

INSERT INTO SERVICE VALUES (102, 'European', 'Mild', 'Pasta', 'Modern', 2);

INSERT INTO SERVICE VALUES (103, 'American', 'Fast Food', 'Burger', 'Urban', 3);

INSERT INTO SERVICE VALUES (104, 'Scandinavian', 'Fresh', 'Salmon', 'Minimal', 4);

-- LANDSCAPE

INSERT INTO LANDSCAPE VALUES (201, 'Vinh Long Bridge', 'Long bridge in Oslo', 'Built 1990', 'Connects two parts of city');

INSERT INTO LANDSCAPE VALUES (202, 'Cantho Ferry', 'Old ferry', 'Since 1980', 'Historical transport');

INSERT INTO LANDSCAPE VALUES (203, 'Snow Park', 'Popular winter spot', 'Since 2000', 'Tourist attraction');

INSERT INTO LANDSCAPE VALUES (204, 'Lake Oslo', 'Freshwater lake', 'Natural', 'Relaxation');

-- HOTEL\_LANDSCAPE

INSERT INTO HOTEL\_LANDSCAPE VALUES (1, 201);

INSERT INTO HOTEL\_LANDSCAPE VALUES (1, 202);

INSERT INTO HOTEL\_LANDSCAPE VALUES (2, 203);

INSERT INTO HOTEL\_LANDSCAPE VALUES (3, 204);

-- ROOM

INSERT INTO ROOM VALUES (301, 1, 'Deluxe', 150.00);

INSERT INTO ROOM VALUES (302, 2, 'Standard', 100.00);

INSERT INTO ROOM VALUES (303, 3, 'Suite', 200.00);

INSERT INTO ROOM VALUES (304, 4, 'Economy', 80.00);

-- GUEST

INSERT INTO GUEST VALUES (401, 'Alice', 'Female', 'Oslo', '1980-05-04');

INSERT INTO GUEST VALUES (402, 'Bob', 'Male', 'Bergen', '1975-08-15');

INSERT INTO GUEST VALUES (403, 'Charlie', 'Male', 'Oslo', '1990-03-20');

INSERT INTO GUEST VALUES (404, 'Dana', 'Female', 'Trondheim', '1985-11-11');

-- RENTAL

INSERT INTO RENTAL VALUES (301, 401, '2000-03-05', '2000-03-10');

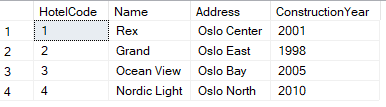
INSERT INTO RENTAL VALUES (301, 402, '2001-04-01', '2001-04-05');

INSERT INTO RENTAL VALUES (302, 403, '2000-07-01', '2000-07-04');

INSERT INTO RENTAL VALUES (303, 404, '2000-03-08', '2000-03-12');

**b. Đổi tên khách sạn “Rex” thành “Sheraton”**

**Trước khi thay đổi:**



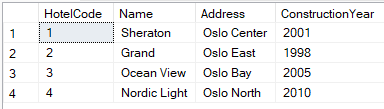
**Hàm thay đổi:**

UPDATE HOTEL

SET Name = 'Sheraton'

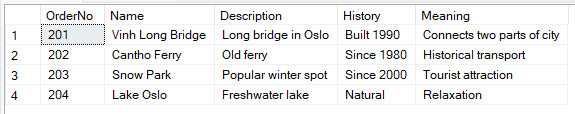
WHERE Name = 'Rex';

**Kết quả:**



**c. Xoá cảnh quan “Cantho Ferry”**

**Trước khi thay đổi:**

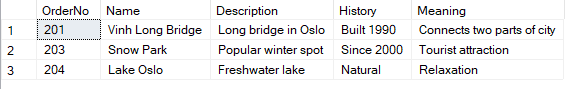


**Hàm thay đổi:**

DELETE FROM HOTEL\_LANDSCAPE WHERE OrderNo = 202;

DELETE FROM LANDSCAPE WHERE Name = 'Cantho Ferry';

**Kết quả:**



**d. Thông tin khách sạn có giá phòng cao nhất**

**Hàm truy vấn:**

SELECT H.\*

FROM HOTEL H

JOIN ROOM R ON H.HotelCode = R.HotelCode

WHERE R.Price = (

SELECT MAX(Price) FROM ROOM

);

**Kết quả:**



**e. Liệt kê món ăn của khách sạn có giá phòng thấp nhất**

**Hàm truy vấn:**

SELECT S.Food

FROM SERVICE S

JOIN HOTEL H ON S.HotelCode = H.HotelCode

JOIN ROOM R ON H.HotelCode = R.HotelCode

WHERE R.Price = (

SELECT MIN(Price) FROM ROOM

);

**Kết quả:**



**f. Đếm số lượng khách đã lưu trú gần cầu “Vinh Long” trong năm 2000**

**Hàm truy vấn:**

SELECT COUNT(DISTINCT G.GuestID) AS GuestCount

FROM GUEST G

JOIN RENTAL R ON G.GuestID = R.GuestID

JOIN ROOM Ro ON R.RoomID = Ro.RoomID

JOIN HOTEL\_LANDSCAPE HL ON Ro.HotelCode = HL.HotelCode

JOIN LANDSCAPE L ON HL.OrderNo = L.OrderNo

WHERE L.Name = 'Vinh Long Bridge'

AND YEAR(R.CheckInDate) = 2000;

**Kết quả:**

