

Phần 1: CSDL QUẢN LÝ HỌC SINH

Để xây dựng một cơ sở dữ liệu quản lý học sinh, chúng ta cần thiết kế các bảng cơ bản như sau: **Học sinh**, **Lớp học**, **Môn học**, và **Kết quả**. Dưới đây là đề xuất về cách thiết kế các bảng đơn giản như trên kèm dữ liệu mẫu của CSD quản lý học sinh.

I. Thiết kế bảng:

a. Bảng HocSinh

- MaHS** (VARCHAR) - Khóa chính, mã học sinh.
- TenHS** (VARCHAR) - Tên học sinh.
- NgaySinh** (DATE) - Ngày sinh.
- GioiTinh** (VARCHAR) - Giới tính.
- MaLop** (VARCHAR) - Khóa ngoại, liên kết với bảng **LopHoc**.

b. Bảng LopHoc

- MaLop** (VARCHAR) - Khóa chính, mã lớp học.
- TenLop** (VARCHAR) - Tên lớp học.
- NienKhoa** (VARCHAR) - Niên khóa.

c. Bảng MonHoc

- MaMH** (VARCHAR) - Khóa chính, mã môn học.
- TenMH** (VARCHAR) - Tên môn học.

d. Bảng KetQua

- MaHS** (VARCHAR) - Khóa ngoại, liên kết với **HocSinh**.
- MaMH** (VARCHAR) - Khóa ngoại, liên kết với **MonHoc**.
- Diem** (FLOAT) - Điểm số.
- HocKy** (INT) - Học kỳ.

II. Xây dựng cấu trúc bảng:

a. Tạo bảng LopHoc

Bảng này chứa thông tin về các lớp học, bao gồm mã lớp, tên lớp, và niên khóa.

```
CREATE TABLE LopHoc (  
    MaLop VARCHAR(10) PRIMARY KEY,  
    TenLop VARCHAR(50) NOT NULL,  
    NienKhoa VARCHAR(20) NOT NULL  
);
```

b. Tạo bảng HocSinh

Bảng này lưu thông tin của học sinh, bao gồm mã học sinh, tên, ngày sinh, giới tính, và mã lớp mà học sinh đó thuộc về. MaLop là khóa ngoại tham chiếu đến bảng LopHoc.

```
CREATE TABLE HocSinh (  
    MaHS VARCHAR(10) PRIMARY KEY,
```

```
TenHS VARCHAR(50) NOT NULL,  
NgàySinh DATE NOT NULL,  
GioiTinh VARCHAR(10) NOT NULL,  
MaLop VARCHAR(10) NOT NULL,  
FOREIGN KEY (MaLop) REFERENCES LopHoc(MaLop)  
ON DELETE CASCADE
```

);

c. Tạo bảng MonHoc

Bảng này chứa thông tin về các môn học, bao gồm mã môn học và tên môn học.

```
CREATE TABLE MonHoc (  
    MaMH VARCHAR(10) PRIMARY KEY,  
    TenMH VARCHAR(50) NOT NULL
```

);

d. Tạo bảng KetQua

Bảng này ghi lại điểm số của học sinh theo từng môn học và học kỳ. Mỗi bản ghi là duy nhất với sự kết hợp của MaHS, MaMH, và HocKy. MaHS và MaMH là khóa ngoại liên kết tới bảng HocSinh và MonHoc.

```
CREATE TABLE KetQua (  
    MaHS VARCHAR(10),  
    MaMH VARCHAR(10),  
    Diem FLOAT CHECK (Diem >= 0 AND Diem <= 10),  
    HocKy INT,  
    PRIMARY KEY (MaHS, MaMH, HocKy),  
    FOREIGN KEY (MaHS) REFERENCES HocSinh(MaHS)  
    ON DELETE CASCADE,  
    FOREIGN KEY (MaMH) REFERENCES MonHoc(MaMH)  
    ON DELETE CASCADE
```

);

III. Dữ liệu mẫu:

a. Bảng HocSinh

```
INSERT INTO LopHoc (MaLop, TenLop, NienKhoa) VALUES  
( 'L01', '10A1', '2022-2025'),  
( 'L02', '11B2', '2022-2025'),  
( 'L03', '11B5', '2022-2025');
```

b. Bảng LopHoc

```
INSERT INTO HocSinh (MaHS, TenHS, NgàySinh, GioiTinh, MaLop) VALUES  
( 'HS01', 'Nguyen Van A', '2007-01-01', 'Nam', 'L01'),
```

('HS02', 'Tran Thi B', '2007-02-15', 'Nữ', 'L01'),
('HS03', 'Pham Thi Thang', '2006-01-01', 'Nam', 'L03'),
('HS04', 'Duong Truc Linh', '2006-02-15', 'Nữ', 'L02'),
('HS05', 'Nguyen Gia Vy', '2006-01-01', 'Nam', 'L03'),
('HS06', 'Nguyen Viet Anh', '2006-01-01', 'Nam', 'L02'),
('HS07', 'Tran Thi Sot', '2006-02-15', 'Nữ', 'L02'),
('HS08', 'Pham Thi Thao', '2006-01-01', 'Nam', 'L03'),
('HS09', 'Duong Linh Hoi', '2007-02-15', 'Nữ', 'L01'),
('HS10', 'Nguyen Gia Khanh', '2007-01-01', 'Nam', 'L01'),
('HS11', 'Tran Thi Bao', '2007-02-15', 'Nữ', 'L01');

c. Bảng MonHoc

INSERT INTO MonHoc (MaMH, TenMH) VALUES

('MH01', 'Toán'),
('MH05', 'Tin');

d. Bảng KetQua

INSERT INTO KetQua (MaHS, MaMH, Diem, HocKy) VALUES

('HS01', 'MH01', 8.5, 1),
('HS01', 'MH05', 7.0, 1),
('HS02', 'MH01', 8.5, 1),
('HS02', 'MH05', 7.0, 1),
('HS03', 'MH01', 6.0, 1),
('HS03', 'MH05', 8.5, 1),
('HS04', 'MH01', 7.0, 1),
('HS04', 'MH05', 8.5, 1),
('HS05', 'MH01', 7.0, 1),
('HS05', 'MH05', 8.5, 1),
('HS06', 'MH01', 7.0, 1),
('HS06', 'MH05', 8.5, 1),
('HS07', 'MH01', 7.0, 1),
('HS07', 'MH05', 7.0, 1),
('HS08', 'MH01', 7.0, 1),
('HS08', 'MH05', 8.5, 1),
('HS09', 'MH01', 7.0, 1),
('HS09', 'MH05', 8.5, 1),
('HS10', 'MH01', 7.0, 1),
('HS10', 'MH05', 8.5, 1),
('HS11', 'MH01', 7.0, 1),

('HS11', 'MH05', 7.0, 1);

IV. Câu lệnh truy vấn SQL:

1. Cập nhật điểm cho học sinh

Để cập nhật điểm số của học sinh cho một môn học cụ thể trong một học kỳ, bạn có thể sử dụng câu lệnh sau:

```
UPDATE KetQua
SET Diem = 5
WHERE MaHS = 'HS09' AND MaMH = 'MH01' AND HocKy = 1;
```

2. Xóa học sinh

Nếu một học sinh không còn học tại trường nữa, bạn có thể xóa thông tin của học sinh đó khỏi cơ sở dữ liệu:

```
DELETE FROM HocSinh
WHERE MaHS = 'HS01';
```

Lưu ý: Trước khi xóa học sinh, bạn cần đảm bảo rằng đã xóa hoặc cập nhật các bảng có khóa ngoại liên quan để tránh vi phạm ràng buộc toàn vẹn dữ liệu

3. Truy vấn thông tin học sinh theo lớp

```
SELECT *
FROM HocSinh
WHERE MaLop = 'L01';
```

4. Truy vấn thông tin học sinh và điểm của học sinh theo một môn học cụ thể, sau đó sắp xếp điểm giảm dần. Ví dụ điểm môn Toán.

```
SELECT HocSinh.TenHS, KetQua.Diem
FROM HocSinh
INNER JOIN KetQua ON HocSinh.MaHS = KetQua.MaHS
WHERE KetQua.MaMH = 'MH01' -- Giả sử 'MH01' là mã của môn Toán
ORDER BY KetQua.Diem DESC;
```

5. Lấy danh sách học sinh và tên lớp của học sinh đó

```
SELECT HocSinh.TenHS, LopHoc.TenLop
FROM HocSinh
INNER JOIN LopHoc ON HocSinh.MaLop = LopHoc.MaLop
ORDER BY LopHoc.TenLop, HocSinh.TenHS;
```

6. Tìm học sinh có điểm cao nhất trong một môn học

```
SELECT HocSinh.TenHS, KetQua.Diem
FROM HocSinh
INNER JOIN KetQua ON HocSinh.MaHS = KetQua.MaHS
WHERE KetQua.MaMH = 'MH01' -- Giả sử 'MH01' là mã của môn Toán
ORDER BY KetQua.Diem DESC
```

LIMIT 1;

7. Truy vấn học sinh sinh ngày một ngày cụ thể và các điểm của họ theo từng môn

```
SELECT HocSinh.TenHS, MonHoc.TenMH, KetQua.Diem
FROM HocSinh
INNER JOIN KetQua ON HocSinh.MaHS = KetQua.MaHS
INNER JOIN MonHoc ON KetQua.MaMH = MonHoc.MaMH
WHERE year(HocSinh.NgaySinh) = '2006'
-- Giả sử muốn tìm học sinh sinh vào năm 2006
ORDER BY MonHoc.TenMH;
```

8. Liệt kê tất cả học sinh và môn học mà họ đã đạt điểm từ cao xuống thấp

```
SELECT HocSinh.TenHS, MonHoc.TenMH, KetQua.Diem
FROM HocSinh
INNER JOIN KetQua ON HocSinh.MaHS = KetQua.MaHS
INNER JOIN MonHoc ON KetQua.MaMH = MonHoc.MaMH
ORDER BY KetQua.Diem DESC;
```

9. Truy vấn danh sách học sinh, tên lớp và điểm môn Toán trong học kỳ 1

```
SELECT HocSinh.TenHS, LopHoc.TenLop, KetQua.Diem
FROM HocSinh
INNER JOIN LopHoc ON HocSinh.MaLop = LopHoc.MaLop
INNER JOIN KetQua ON HocSinh.MaHS = KetQua.MaHS
WHERE KetQua.MaMH = 'MH01' AND KetQua.HocKy = 1
ORDER BY KetQua.Diem DESC;
```

10. Truy vấn danh sách học sinh sinh trước năm 2007 và điểm trung bình tổng thể của học sinh

```
SELECT HocSinh.TenHS, AVG(KetQua.Diem) AS DiemTB
FROM HocSinh
INNER JOIN KetQua ON HocSinh.MaHS = KetQua.MaHS
WHERE HocSinh.NgaySinh < '2007-01-01'
GROUP BY HocSinh.MaHS
ORDER BY DiemTB DESC;
```

11. Truy vấn điểm trung bình của học sinh theo từng môn học

```
SELECT MaHS, AVG(Diem) AS DiemTrungBinh
FROM KetQua
GROUP BY MaHS, MaMH;
```

12. Liệt kê học sinh và điểm trung bình của học sinh đó

```
SELECT HocSinh.TenHS, AVG(KetQua.Diem) AS DiemTrungBinh
FROM HocSinh
```

```
INNER JOIN KetQua ON HocSinh.MaHS = KetQua.MaHS
GROUP BY HocSinh.MaHS
ORDER BY DiemTrungBinh DESC;
```

13. Truy vấn danh sách các lớp và số lượng học sinh trong từng lớp

```
SELECT LopHoc.TenLop, COUNT(HocSinh.MaHS) AS SoLuongHS
FROM LopHoc JOIN HocSinh
ON LopHoc.MaLop = HocSinh.MaLop
GROUP BY LopHoc.TenLop;
```

14. Truy vấn thông tin học sinh có điểm trung bình cao nhất

```
SELECT HocSinh.MaHS, HocSinh.TenHS, AVG(KetQua.Diem) AS DiemTB
FROM HocSinh JOIN KetQua ON HocSinh.MaHS = KetQua.MaHS
GROUP BY HocSinh.MaHS
ORDER BY DiemTB DESC
LIMIT 1;
```

15. Truy vấn thông tin học sinh và điểm trung bình mỗi học kỳ

Câu truy vấn này sẽ cho bạn biết điểm trung bình của mỗi học sinh theo từng học kỳ:

```
SELECT HocSinh.MaHS, HocSinh.TenHS, KetQua.HocKy, AVG(KetQua.Diem) AS
DiemTrungBinh
FROM HocSinh JOIN KetQua
ON HocSinh.MaHS = KetQua.MaHS
GROUP BY HocSinh.MaHS, KetQua.HocKy;
```

16. Truy vấn danh sách học sinh và điểm trung bình của họ theo lớp, chỉ lấy điểm trung bình trên 8

```
SELECT HocSinh.TenHS, AVG(KetQua.Diem) AS DiemTB
FROM HocSinh
INNER JOIN KetQua ON HocSinh.MaHS = KetQua.MaHS
WHERE KetQua.Diem > 8
GROUP BY HocSinh.TenHS
ORDER BY DiemTB DESC;
```

Phần 2: CSDL QUẢN LÝ THƯ VIỆN

Dưới đây là các câu lệnh SQL để tạo ra cơ sở dữ liệu quản lý thư viện đơn giản, bao gồm 4 bảng chính: **Sach**, **DocGia**, **MuonTra**, và **NhanVien**. Sau đó, tôi sẽ cung cấp một số dữ liệu mẫu cho mỗi bảng.

I. Thiết kế bảng:

a. Bảng Sach

- MaSach**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi quyển sách.
- TenSach**: VARCHAR(100) - Tên của sách.
- TacGia**: VARCHAR(50) - Tên tác giả của sách.
- TheLoai**: VARCHAR(50) - Thể loại của sách.
- NamXuatBan**: INT - Năm xuất bản của sách.
- SoLuong**: INT - Số lượng sách hiện có trong thư viện.

b. Bảng DocGia

- MaDocGia**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi độc giả.
- TenDocGia**: VARCHAR(50) - Tên của độc giả.
- NgaySinh**: DATE - Ngày sinh của độc giả.
- DiaChi**: VARCHAR(100) - Địa chỉ liên lạc của độc giả.
- SoDienThoai**: VARCHAR(15) - Số điện thoại liên lạc của độc giả.

c. Bảng NhanVien

- MaNV**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi nhân viên.
- TenNV**: VARCHAR(50) - Tên của nhân viên.
- ChucVu**: VARCHAR(50) - Chức vụ của nhân viên trong thư viện.
- NgayVaoLam**: DATE - Ngày nhân viên bắt đầu làm việc tại thư viện.

d. Bảng MuonTra

- MaMuonTra**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi giao dịch mượn hoặc trả sách.
- MaDocGia**: VARCHAR(10) - Khóa ngoại, liên kết với bảng **DocGia**.
- MaSach**: VARCHAR(10) - Khóa ngoại, liên kết với bảng **Sach**.
- MaNV**: VARCHAR(10) - Khóa ngoại, liên kết với bảng **NhanVien**.
- NgayMuon**: DATE - Ngày mà sách được mượn.
- NgayTra**: DATE - Ngày mà sách được trả.

II. Xây dựng cấu trúc bảng:

a. Tạo bảng Sach

```
CREATE TABLE Sach (  
    MaSach VARCHAR(10) PRIMARY KEY,
```

TenSach **VARCHAR(100) NOT NULL**,
TacGia **VARCHAR(50) NOT NULL**,
TheLoai **VARCHAR(50) NOT NULL**,
NamXuatBan **INT**,
SoLuong **INT DEFAULT 0**

);

b. Tạo bảng DocGia

CREATE TABLE DocGia (
MaDocGia **VARCHAR(10) PRIMARY KEY**,
TenDocGia **VARCHAR(50) NOT NULL**,
NgaySinh **DATE**,
DiaChi **VARCHAR(100)**,
SoDienThoai **VARCHAR(15)**

);

c. Tạo bảng NhanVien

CREATE TABLE NhanVien (
MaNV **VARCHAR(10) PRIMARY KEY**,
TenNV **VARCHAR(50) NOT NULL**,
ChucVu **VARCHAR(50)**,
NgayVaoLam **DATE**

);

d. Tạo bảngMuonTra

CREATE TABLE MuonTra (
MaMuonTra **VARCHAR(10) PRIMARY KEY**,
MaDocGia **VARCHAR(10) NOT NULL**,
MaSach **VARCHAR(10) NOT NULL**,
MaNV **VARCHAR(10) NOT NULL**,
NgayMuon **DATE NOT NULL**,
NgayTra **DATE**,
FOREIGN KEY (MaDocGia) **REFERENCES** DocGia(MaDocGia)
ON DELETE CASCADE,
FOREIGN KEY (MaSach) **REFERENCES** Sach(MaSach)
ON DELETE CASCADE,
FOREIGN KEY (MaNV) **REFERENCES** NhanVien(MaNV)
ON DELETE CASCADE

);

III. Dữ liệu mẫu:

a. Bảng Sach

INSERT INTO Sach (MaSach, TenSach, TacGia, TheLoai, NamXuatBan, SoLuong)
VALUES

('S001', 'Tôi thấy hoa vàng trên cỏ xanh', 'Nguyễn Nhật Ánh', 'Văn học', 2010, 3),
('S002', 'Dế mèn phiêu lưu ký', 'Tô Hoài', 'Thiếu nhi', 1999, 5),
('S003', 'Mắt biếc', 'Nguyễn Nhật Ánh', 'Văn học', 2004, 2),
('S004', 'Những người khốn khổ', 'Victor Hugo', 'Tiểu thuyết', 1862, 4),
('S005', 'Harry Potter và Hòn đá Phù thủy', 'J.K. Rowling', 'Fantasy', 1997, 6),
('S006', 'Chí Phèo', 'Nam Cao', 'Văn học', 1941, 2),
('S007', 'Sherlock Holmes', 'Arthur Conan Doyle', 'Trình thám', 1892, 5),
('S008', 'Lược sử thời gian', 'Stephen Hawking', 'Khoa học', 1988, 3),
('S009', 'Chiến tranh và hòa bình', 'Leo Tolstoy', 'Tiểu thuyết', 1869, 1),
('S010', 'The Great Gatsby', 'F. Scott Fitzgerald', 'Tiểu thuyết', 1925, 4);

b. Bảng DocGia

INSERT INTO DocGia (MaDocGia, TenDocGia, NgaySinh, DiaChi, SoDienThoai)
VALUES

('D001', 'Nguyễn Văn A', '1990-08-15', 'Hà Nội, Việt Nam', '0987654321'),
('D002', 'Trần Thị B', '1985-07-20', 'TP Hồ Chí Minh, Việt Nam', '0935123456'),
('D003', 'Lê Văn C', '2000-12-01', 'Đà Nẵng, Việt Nam', '0912345678'),
('D004', 'Phạm Thị D', '1998-03-25', 'Cần Thơ, Việt Nam', '0978123456'),
('D005', 'Hoàng Minh E', '1975-05-30', 'Nghệ An, Việt Nam', '0969123456');

c. Bảng NhanVien

INSERT INTO NhanVien (MaNV, TenNV, ChucVu, NgayVaoLam) **VALUES**

('NV01', 'Nguyễn Hữu F', 'Thủ thư', '2015-06-01'),
('NV02', 'Trần Anh G', 'Quản lý', '2013-08-15'),
('NV03', 'Lê Thị H', 'Bảo vệ', '2018-01-10'),
('NV04', 'Phạm Bảo I', 'Thủ thư', '2017-05-20'),
('NV05', 'Võ Minh J', 'Thủ thư', '2019-07-22');

d. Bảng MuonTra

INSERT INTO MuonTra (MaMuonTra, MaDocGia, MaSach, MaNV, NgayMuon, NgayTra) **VALUES**

('MT001', 'D001', 'S001', 'NV01', '2022-06-01', NULL),
('MT002', 'D002', 'S002', 'NV02', '2022-07-15', '2022-08-14'),
('MT003', 'D003', 'S003', 'NV01', '2022-09-10', NULL),

```
('MT004', 'D004', 'S004', 'NV03', '2022-10-20', NULL),  
('MT005', 'D005', 'S005', 'NV04', '2022-11-22', '2022-12-21');
```

IV. Câu lệnh truy vấn SQL:

1. Lấy danh sách sách và tên tác giả

```
SELECT TenSach, TacGia  
FROM Sach;
```

2. Tìm sách theo thể loại

```
SELECT *  
FROM Sach  
WHERE TheLoai = 'Văn học';
```

3. Cập nhật số lượng sách sau khi nhập thêm

```
UPDATE Sach  
SET SoLuong = SoLuong + 10  
WHERE MaSach = 'S001';
```

4. Xóa sách không còn trong kho

```
DELETE FROM Sach  
WHERE SoLuong = 0;
```

5. Lấy thông tin mượn sách của độc giả cụ thể

```
SELECT DocGia.TenDocGia, Sach.TenSach,MuonTra.NgayMuon  
FROM MuonTra  
INNER JOIN DocGia ON MuonTra.MaDocGia = DocGia.MaDocGia  
INNER JOIN Sach ON MuonTra.MaSach = Sach.MaSach  
WHERE DocGia.MaDocGia = 'D001';
```

6. Danh sách các sách được mượn bởi nhân viên cụ thể

```
SELECT NhanVien.TenNV, Sach.TenSach, MuonTra.NgayMuon  
FROM MuonTra INNER JOIN NhanVien  
ON MuonTra.MaNV = NhanVien.MaNV  
INNER JOIN Sach ON MuonTra.MaSach = Sach.MaSach  
WHERE NhanVien.MaNV = 'NV01';
```

7. Lấy danh sách sách mượn chưa được trả

```
SELECT Sach.TenSach, DocGia.TenDocGia, MuonTra.NgayMuon  
FROM MuonTra INNER JOIN Sach ON MuonTra.MaSach = Sach.MaSach  
INNER JOIN DocGia ON MuonTra.MaDocGia = DocGia.MaDocGia  
WHERE MuonTra.NgayTra IS NULL;
```

8. Cập nhật ngày trả sách cho một mượn trả cụ thể

```
UPDATE MuonTra  
SET NgayTra = '2023-02-01'
```

WHERE MaMuonTra = 'MT01';

9. Xóa bản ghi mượn trả khi sách đã bị mất

DELETE FROMMuonTra

WHERE MaMuonTra = 'MT02' **AND** NgayTra **IS NULL**;

10. Tìm sách theo tên sách và sắp xếp theo năm xuất bản giảm dần

SELECT * FROM Sach

WHERE TenSach **LIKE** '%Chiến%'

ORDER BY NamXuatBan **DESC**;

11. Lấy thông tin các độc giả sinh trước năm 1990

SELECT * FROM DocGia

WHERE **YEAR**(NgaySinh) < 1990

ORDER BY NgaySinh;

12. Cập nhật thông tin liên lạc của độc giả

UPDATE DocGia

SET DiaChi = 'Đà Nẵng', SoDienThoai = '0987123456'

WHERE MaDocGia = 'D02';

13. Tìm các sách của tác giả 'Ngô Tất Tố'

SELECT *

FROM Sach

WHERE TacGia = 'Ngô Tất Tố';

14. Xem lịch sử mượn sách của nhân viên

SELECT NhanVien.TenNV, Sach.TenSach, MuonTra.NgayMuon, MuonTra.NgayTra

FROM MuonTra

INNER JOIN NhanVien **ON** MuonTra.MaNV = NhanVien.MaNV

INNER JOIN Sach **ON** MuonTra.MaSach = Sach.MaSach

ORDER BY MuonTra.NgayMuon **DESC**;

15. Liệt kê các sách có số lượng dưới 3 không kể thể loại

SELECT *

FROM Sach

WHERE SoLuong < 3;

16. Cập nhật chức vụ cho nhân viên

UPDATE NhanVien

SET ChucVu = 'Quản lý'

WHERE MaNV = 'NV01';

17. Danh sách sách được mượn nhiều nhất

SELECT Sach.TenSach, **COUNT(*) AS** SoLanMuon

FROM MuonTra **INNER JOIN** Sach

```
ONMuonTra.MaSach = Sach.MaSach
GROUP BY Sach.TenSach
ORDER BY SoLanMuon DESC;
```

18. Lấy danh sách độc giả đã mượn sách trong tháng này

```
SELECT DISTINCT DocGia.TenDocGia
FROM MuonTra INNER JOIN DocGia
ON MuonTra.MaDocGia = DocGia.MaDocGia
WHERE MONTH(MuonTra.NgayMuon) = MONTH(CURRENT_DATE()) AND
YEAR(MuonTra.NgayMuon) = YEAR(CURRENT_DATE());
```

19. Xóa các bản ghi mượn trả quá hạn mà không có thông tin ngày trả

```
DELETE FROM MuonTra WHERE NgayTra IS NULL AND NgayMuon <
DATE_SUB(CURRENT_DATE(), INTERVAL 1 YEAR);
```

20. Lấy danh sách các độc giả mượn sách nhưng chưa trả sau 30 ngày

```
SELECT DocGia.TenDocGia, Sach.TenSach, MuonTra.NgayMuon
FROM MuonTra
INNER JOIN DocGia ON MuonTra.MaDocGia = DocGia.MaDocGia
INNER JOIN Sach ON MuonTra.MaSach = Sach.MaSach
WHERE MuonTra.NgayTra IS NULL AND MuonTra.NgayMuon <=
DATE_SUB(CURRENT_DATE(), INTERVAL 30 DAY);
```

Phần 3: CSDL QUẢN LÝ DU LỊCH

Mục đích cần tạo cơ sở dữ liệu quản lý du lịch để tối ưu hóa hoạt động và nâng cao trải nghiệm khách hàng. CSDL giúp lưu trữ và quản lý thông tin một cách hệ thống, dễ dàng tra cứu và cập nhật. Nó hỗ trợ việc phân tích dữ liệu, dự báo nhu cầu, và lập kế hoạch kinh doanh hiệu quả. Đồng thời, CSDL giúp theo dõi chất lượng dịch vụ và đảm bảo an toàn trong du lịch.

Để xây dựng một cơ sở dữ liệu quản lý du lịch, chúng ta cần thiết kế các bảng cơ bản như sau: **Khách hàng**, **Tour Du Lịch**, **Hướng dẫn viên**, và **Đặt Tour**. Dưới đây là đề xuất về cách thiết kế các bảng đơn giản như trên kèm dữ liệu mẫu của CSD quản lý du lịch.

I. Thiết kế bảng:

a. Bảng KháchHang

- MaKH:** VARCHAR(10) - Khóa chính, mã định danh cho mỗi khách hàng.
- TenKH:** VARCHAR(50) - Tên của khách hàng.
- NgaySinh:** DATE - Ngày sinh của khách hàng.
- DiaChi:** VARCHAR(100) - Địa chỉ liên lạc của khách hàng.
- SoDienThoai:** VARCHAR(15) - Số điện thoại liên lạc của khách hàng.

b. Bảng TourDuLich

- MaTour:** VARCHAR(10) - Khóa chính, mã định danh cho mỗi tour du lịch.
- TenTour:** VARCHAR(100) - Tên của tour.
- MoTa:** TEXT - Mô tả chi tiết về tour du lịch.
- GiaTour:** DECIMAL(10, 2) - Giá của tour.
- NgayKhoiHan:** DATE - Ngày khởi hành của tour.
- NgayKetThuc:** DATE - Ngày kết thúc của tour.

c. Bảng HuongDanVien

- MaHDV:** VARCHAR(10) - Khóa chính, mã định danh cho mỗi hướng dẫn viên.
- TenHDV:** VARCHAR(50) - Tên của hướng dẫn viên.
- SoDienThoai:** VARCHAR(15) - Số điện thoại liên lạc của hướng dẫn viên.
- BangCap:** VARCHAR(50) - Bằng cấp của hướng dẫn viên.

d. Bảng DatTour

- MaDatTour:** VARCHAR(10) - Khóa chính, mã định danh cho mỗi đặt tour.
- MaKH:** VARCHAR(10) - Khóa ngoại, liên kết với bảng KháchHang.
- MaTour:** VARCHAR(10) - Khóa ngoại, liên kết với bảng TourDuLich.
- MaHDV:** VARCHAR(10) - Khóa ngoại, liên kết với bảng HuongDanVien.
- NgayDat:** DATE - Ngày đặt tour.
- SoNguoi:** INT - Số người tham gia tour.
- TongTien:** DECIMAL(10, 2) - Tổng tiền của đặt tour.

Các bảng này đều có các trường thông tin cơ bản phục vụ cho việc quản lý khách hàng, tour du lịch, nhân viên (hướng dẫn viên) và các đặt tour. Mỗi bảng còn có các khóa ngoại để liên kết với nhau, đảm bảo tính toàn vẹn và liên kết dữ liệu giữa các bảng.

II. Xây dựng cấu trúc bảng:

a. Bảng KháchHàng

```
CREATE TABLE KháchHàng (  
    MaKH VARCHAR(10) PRIMARY KEY,  
    TenKH VARCHAR(50) NOT NULL,  
    SoDienThoai VARCHAR(15),  
    DiaChi VARCHAR(100),  
    Email VARCHAR(50)  
);
```

b. Bảng TourDuLich

```
CREATE TABLE TourDuLich (  
    MaTour VARCHAR(10) PRIMARY KEY,  
    TenTour VARCHAR(100) NOT NULL,  
    MoTa TEXT,  
    GiaTour DECIMAL(10, 2),  
    NgayKhoiHanh DATE,  
    NgayKetThuc DATE  
);
```

c. Bảng HuongDanVien

```
CREATE TABLE HuongDanVien (  
    MaHDV VARCHAR(10) PRIMARY KEY,  
    TenHDV VARCHAR(50) NOT NULL,  
    SoDienThoai VARCHAR(15),  
    BangCap VARCHAR(50)  
);
```

d. Bảng DatTour

```
CREATE TABLE DatTour (  
    MaDatTour VARCHAR(10) PRIMARY KEY,  
    MaKH VARCHAR(10) NOT NULL,  
    MaTour VARCHAR(10) NOT NULL,  
    MaHDV VARCHAR(10) NOT NULL,  
    NgayDat DATE NOT NULL,  
    SoLuongNguoi INT,  
    TongTien DECIMAL(10, 2),
```

FOREIGN KEY (MaKH) REFERENCES KháchHang(MaKH),
FOREIGN KEY (MaTour) REFERENCES TourDuLich(MaTour)

);

III. Dữ liệu mẫu:

a. Bảng KháchHang

```
INSERT INTO KháchHang (MaKH, TenKH, SoDienThoai, DiaChi, Email) VALUES
('KH01', 'Nguyen Van An', '0123456789', 'Ha Noi, Viet Nam', 'vana@example.com'),
('KH02', 'Tran Thi Binh', '0987654321', 'Da Nang, Viet Nam', 'thib@example.com'),
('KH03', 'Le Van Chi', '0912345678', 'HCM, Viet Nam', 'vanc@example.com'),
('KH04', 'Pham Thi Dung', '0923456789', 'Can Tho, Viet Nam', 'thid@example.com'),
('KH05', 'Hoang Minh Tuan', '0934567890', 'Hai Phong, Viet Nam', 'minhe@example.com'),
('KH06', 'Phan Van Gia', '0167890123', 'Hue, Viet Nam', 'vang@example.com'),
('KH07', 'Bui Cam Hai', '0198765432', 'Binh Dinh, Viet Nam', 'camh@example.com'),
('KH08', 'Nguyen Tuan Vinh', '0112345678', 'Lam Dong, Viet Nam', 'tuani@example.com'),
('KH09', 'Le Thi Quynh', '0147890123', 'Thai Nguyen, Viet Nam', 'thij@example.com'),
('KH10', 'Tran Duc Khanh', '0187654321', 'Ca Mau, Viet Nam', 'duck@example.com');
```

b. Bảng TourDuLich

```
INSERT INTO TourDuLich (MaTour, TenTour, MoTa, GiaTour, NgayKhoiHanh, NgayKetThuc) VALUES
('TOUR01', 'Tour Ha Long', 'Du lich bien dao', 1500000, '2023-06-01', '2023-06-03'),
('TOUR02', 'Tour Da Nang', 'Tham quan Ngu Hanh Son', 1200000, '2023-06-15', '2023-06-17'),
('TOUR03', 'Tour Nha Trang', 'Nghỉ dưỡng tại bãi biển', 1800000, '2023-07-01', '2023-07-05'),
('TOUR04', 'Tour Sa Pa', 'Trekking núi', 2000000, '2023-08-01', '2023-08-04'),
('TOUR05', 'Tour Phu Quoc', 'Du lich bien dao, ngam san ho', 2500000, '2023-09-01', '2023-09-05'),
('TOUR06', 'Tour Hue', 'Tham quan Dai Noi Hue', 1300000, '2023-10-01', '2023-10-03'),
('TOUR07', 'Tour Mien Tay', 'Kham pha dong song Mekong', 1500000, '2023-10-15', '2023-10-17'),
('TOUR08', 'Tour Da Lat', 'Tham quan thanh pho ngan hoa', 1600000, '2023-11-01', '2023-11-04'),
('TOUR09', 'Tour Quang Binh', 'Kham pha dong Phong Nha', 1700000, '2023-11-15', '2023-11-18'),
('TOUR10', 'Tour Thai Nguyen', 'Tham quan vung che', 1100000, '2023-12-01', '2023-12-03');
```

c. Bảng HuongDanVien

```
INSERT INTO HuongDanVien (MaHDV, TenHDV, SoDienThoai, BangCap) VALUES
('HDV01', 'Doan Quang H', '0123451234', 'Du lich'),
('HDV02', 'Nguyen Thi I', '0987654321', 'Su pham'),
('HDV03', 'Tran Anh K', '0912121212', 'Ngoai ngu'),
('HDV04', 'Le Minh M', '0939393939', 'Am nhac'),
('HDV05', 'Pham Bao N', '0948484848', 'Du lich quoc te'),
('HDV06', 'Dao Quoc O', '0153451234', 'Lich su');
```

```
('HDV07', 'Nguyen Quynh P', '0981654321', 'Van hoa'),  
('HDV08', 'Tran Anh Q', '0912821212', 'Du lich'),  
('HDV09', 'Le Quang R', '0939494939', 'Ngoai ngu'),  
('HDV10', 'Pham Bao S', '0948585848', 'Du lich quoc te');
```

d. Bảng DatTour

```
INSERT INTO DatTour (MaDatTour, MaKH, MaTour, MaHDV, NgayDat, SoLuongNguoi,  
TongTien) VALUES
```

```
('DT01', 'KH01', 'TOUR01', 'HDV01', '2023-05-20', 2, 3000000),  
('DT02', 'KH02', 'TOUR02', 'HDV02', '2023-05-25', 1, 1200000),  
('DT03', 'KH03', 'TOUR03', 'HDV03', '2023-06-15', 3, 5400000),  
('DT04', 'KH04', 'TOUR04', 'HDV04', '2023-07-20', 4, 8000000),  
('DT05', 'KH05', 'TOUR05', 'HDV05', '2023-08-15', 2, 5000000),  
('DT06', 'KH06', 'TOUR06', 'HDV03', '2023-09-20', 2, 2600000),  
('DT07', 'KH07', 'TOUR07', 'HDV02', '2023-10-10', 1, 1500000),  
('DT08', 'KH08', 'TOUR08', 'HDV01', '2023-10-22', 3, 4800000),  
('DT09', 'KH09', 'TOUR09', 'HDV04', '2023-11-10', 4, 6800000),  
('DT10', 'KH10', 'TOUR10', 'HDV05', '2023-11-25', 2, 2200000);
```

IV. Câu lệnh truy vấn SQL:

1. Truy vấn danh sách khách hàng đã đặt tour cụ thể nào đó:

```
SELECT KháchHang.TenKH, DatTour.NgayDat  
FROM KháchHang INNER JOIN DatTour  
ON KháchHang.MaKH = DatTour.MaKH  
WHERE DatTour.MaTour = 'TOUR01'  
ORDER BY DatTour.NgayDat;
```

2. Liệt kê các tour và hướng dẫn viên tương ứng:

```
SELECT TourDuLich.TenTour, HuongDanVien.TenHDV  
FROM TourDuLich  
INNER JOIN DatTour ON TourDuLich.MaTour = DatTour.MaTour  
INNER JOIN HuongDanVien ON DatTour.MaHDV = HuongDanVien.MaHDV  
WHERE TourDuLich.NgayKhoiHanh > '2023-01-01'  
ORDER BY TourDuLich.NgayKhoiHanh;
```

3. Tìm các khách hàng đặt tour trong khoảng thời gian nhất định:

```
SELECT KháchHang.TenKH, DatTour.NgayDat  
FROM KháchHang  
INNER JOIN DatTour ON KháchHang.MaKH = DatTour.MaKH  
WHERE DatTour.NgayDat BETWEEN '2023-01-01' AND '2023-06-30'  
ORDER BY DatTour.NgayDat DESC;
```

4. Danh sách các tour và tổng số người tham gia:

```
SELECT TourDuLich.TenTour, SUM(DatTour.SoLuongNguoi) AS TongSoNguoi
FROM TourDuLich
INNER JOIN DatTour ON TourDuLich.MaTour = DatTour.MaTour
GROUP BY TourDuLich.TenTour
ORDER BY TongSoNguoi DESC;
```

5. Khách hàng đã đặt nhiều tour nhất:

```
SELECT KháchHang.TenKH, COUNT(*) AS SoTourDaDat
FROM KháchHang
INNER JOIN DatTour ON KháchHang.MaKH = DatTour.MaKH
GROUP BY KháchHang.TenKH
ORDER BY SoTourDaDat DESC
LIMIT 1;
```

6. Danh sách tour chưa có ai đặt:

```
SELECT TourDuLich.TenTour
FROM TourDuLich
LEFT JOIN DatTour ON TourDuLich.MaTour = DatTour.MaTour
WHERE DatTour.MaTour IS NULL;
```

7. Tour có giá cao nhất được đặt bởi khách hàng:

```
SELECT KháchHang.TenKH, TourDuLich.TenTour, TourDuLich.GiaTour
FROM KháchHang
INNER JOIN DatTour ON KháchHang.MaKH = DatTour.MaKH
INNER JOIN TourDuLich ON DatTour.MaTour = TourDuLich.MaTour
WHERE TourDuLich.GiaTour = (SELECT MAX(GiaTour) FROM TourDuLich)
ORDER BY KháchHang.TenKH;
```

8. Liệt kê các tour được khởi hành trong tháng 7:

```
SELECT TourDuLich.TenTour, TourDuLich.NgayKhoiHanh
FROM TourDuLich
WHERE MONTH(TourDuLich.NgayKhoiHanh) = 7
ORDER BY TourDuLich.NgayKhoiHanh;
```

9. Danh sách các hướng dẫn viên có bằng cấp 'Du lịch':

```
SELECT HuongDanVien.TenHDV
FROM HuongDanVien
WHERE HuongDanVien.BangCap = 'Du lịch'
ORDER BY HuongDanVien.TenHDV;
```

10. Tour có ít người tham gia nhất:

```
SELECT TourDuLich.TenTour, SUM(DatTour.SoLuongNguoi) AS TongSoNguoiThamGia
```

```
FROM TourDuLich
INNER JOIN DatTour ON TourDuLich.MaTour = DatTour.MaTour
GROUP BY TourDuLich.TenTour
ORDER BY TongSoNguoiThamGia ASC
LIMIT 1;
```

11. Liệt kê thông tin của các tour được khởi hành từ một thành phố cụ thể và sắp xếp theo ngày khởi hành:

```
SELECT TourDuLich.TenTour, TourDuLich.NgayKhoiHanh
FROM TourDuLich
WHERE TourDuLich.TenTour = 'Tour Phu Yen'
ORDER BY TourDuLich.NgayKhoiHanh ASC;
```

12. Tìm hướng dẫn viên có kinh nghiệm dẫn dắt các tour dài ngày:

```
SELECT HuongDanVien.TenHDV, COUNT(DatTour.MaHDV) AS SoTour
FROM HuongDanVien
INNER JOIN DatTour ON HuongDanVien.MaHDV = DatTour.MaHDV
INNER JOIN TourDuLich ON DatTour.MaTour = TourDuLich.MaTour
WHERE NgayKetThuc-NgayKhoiHanh+1 > 3
GROUP BY HuongDanVien.TenHDV
ORDER BY SoTour DESC;
```

13. Liệt kê các tour có giá trên một ngưỡng nhất định và được dẫn dắt bởi một hướng dẫn viên cụ thể:

```
SELECT TourDuLich.TenTour, TourDuLich.GiaTour, HuongDanVien.TenHDV
FROM TourDuLich
INNER JOIN DatTour ON TourDuLich.MaTour = DatTour.MaTour
INNER JOIN HuongDanVien ON DatTour.MaHDV = HuongDanVien.MaHDV
WHERE TourDuLich.GiaTour > 2000000 AND HuongDanVien.TenHDV = 'Nguyen Thi Bich'
ORDER BY TourDuLich.GiaTour DESC;
```

14. Tìm các tour khởi hành trong một khoảng thời gian nhất định và được đặt bởi khách hàng trên 60 tuổi:

```
SELECT TourDuLich.TenTour, TourDuLich.NgayKhoiHanh, KhachHang.TenKH
FROM TourDuLich
INNER JOIN DatTour ON TourDuLich.MaTour = DatTour.MaTour
INNER JOIN KhachHang ON DatTour.MaKH = KhachHang.MaKH
WHERE TourDuLich.NgayKhoiHanh BETWEEN '2023-06-01' AND '2023-12-31'
AND YEAR(CURDATE()) - YEAR(KhachHang.NgaySinh) > 60
ORDER BY TourDuLich.NgayKhoiHanh;
```

15. Danh sách khách hàng đã tham gia ít nhất 3 tour khác nhau:

```
SELECT KháchHang.TenKH, COUNT(DISTINCT DatTour.MaTour) AS SoTourThamGia
FROM KháchHang
INNER JOIN DatTour ON KháchHang.MaKH = DatTour.MaKH
GROUP BY KháchHang.TenKH
HAVING COUNT(DISTINCT DatTour.MaTour) >= 3
ORDER BY SoTourThamGia DESC;
```

Phần 4: CSDL QUẢN LÝ NHÂN SỰ

I. Thiết kế bảng:

a. Bảng NhanVien

- **MaNV**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi nhân viên.
- **TenNV**: VARCHAR(50) - Tên của nhân viên.
- **NgaySinh**: DATE - Ngày sinh của nhân viên.
- **DiaChi**: VARCHAR(100) - Địa chỉ liên lạc của nhân viên.
- **SoDienThoai**: VARCHAR(15) - Số điện thoại liên lạc của nhân viên.
- **MaPB**: VARCHAR(10) - Khóa ngoại, mã phòng ban mà nhân viên thuộc về.

b. Bảng PhongBan

- **MaPB**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi phòng ban.
- **TenPB**: VARCHAR(50) - Tên của phòng ban.

c. Bảng DuAn

- **MaDA**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi dự án.
- **TenDA**: VARCHAR(100) - Tên của dự án.
- **MoTa**: TEXT - Mô tả chi tiết về dự án.

d. Bảng ThamGiaDuAn

- **MaNV**: VARCHAR(10) - Khóa ngoại, mã định danh cho nhân viên tham gia dự án.
- **MaDA**: VARCHAR(10) - Khóa ngoại, mã định danh cho dự án.
- **NgayThamGia**: DATE - Ngày nhân viên bắt đầu tham gia dự án.
- **PRIMARY KEY (MaNV, MaDA)**: Đặt khóa chính là sự kết hợp của **MaNV** và **MaDA**.

II. Xây dựng cấu trúc bảng:

a. Bảng PhongBan

```
CREATE TABLE PhongBan (  
    MaPB VARCHAR(10) PRIMARY KEY,  
    TenPB VARCHAR(50) NOT NULL  
);
```

b. Bảng NhanVien

```
CREATE TABLE NhanVien (  
    MaNV VARCHAR(10) PRIMARY KEY,  
    TenNV VARCHAR(50) NOT NULL,  
    NgaySinh DATE,
```

DiaChi **VARCHAR**(100),
SoDienThoai **VARCHAR**(15),
MaPB **VARCHAR**(10) **NOT NULL**,
FOREIGN KEY (MaPB) **REFERENCES** PhongBan(MaPB) **ON DELETE CASCADE**
);

c. Bảng DuAn

CREATE TABLE DuAn (
MaDA **VARCHAR**(10) **PRIMARY KEY**,
TenDA **VARCHAR**(100) **NOT NULL**,
MoTa **TEXT**
);

d. Bảng ThamGiaDuAn

CREATE TABLE ThamGiaDuAn (
MaNV **VARCHAR**(10),
MaDA **VARCHAR**(10),
NgayThamGia **DATE**,
PRIMARY KEY (MaNV, MaDA),
FOREIGN KEY (MaNV) **REFERENCES** NhanVien(MaNV) **ON DELETE CASCADE**,
FOREIGN KEY (MaDA) **REFERENCES** DuAn(MaDA) **ON DELETE CASCADE**
);

III. Dữ liệu mẫu:

a. Bảng PhongBan

INSERT INTO PhongBan (MaPB, TenPB) **VALUES**
('PB01', 'Phòng Nhân Sự'),
('PB02', 'Phòng Kỹ Thuật'),
('PB03', 'Phòng Tài Chính'),
('PB04', 'Phòng Marketing'),
('PB05', 'Phòng Kinh Doanh');

b. Bảng NhanVien

INSERT INTO NhanVien (MaNV, TenNV, NgaySinh, DiaChi, SoDienThoai, MaPB)
VALUES
('NV01', 'Nguyễn Văn A', '1985-04-12', '123 Đường ABC, Quận 1, TP. HCM', '0901234567', 'PB01'),

```
('NV02', 'Trần Thị B', '1990-06-22', '456 Đường XYZ, Quận 2, TP. HCM', '0912345678', 'PB02'),
('NV03', 'Lê Văn C', '1988-09-15', '789 Đường QWE, Quận 3, TP. HCM', '0923456789', 'PB03'),
('NV04', 'Phạm Thị D', '1992-11-30', '321 Đường RTY, Quận 4, TP. HCM', '0934567890', 'PB04'),
('NV05', 'Ngô Văn E', '1987-01-25', '654 Đường UIO, Quận 5, TP. HCM', '0945678901', 'PB05');
```

c. Bảng DuAn

```
INSERT INTO DuAn (MaDA, TenDA, MoTa) VALUES
('DA01', 'Dự Án Xây Dựng', 'Xây dựng cơ sở hạ tầng cho dự án ABC.'),
('DA02', 'Dự Án Phần Mềm', 'Phát triển phần mềm quản lý cho công ty XYZ.'),
('DA03', 'Dự Án Marketing', 'Chiến dịch marketing cho sản phẩm mới của công ty.'),
('DA04', 'Dự Án Nghiên Cứu', 'Nghiên cứu và phát triển sản phẩm mới.'),
('DA05', 'Dự Án Tài Chính', 'Quản lý tài chính cho dự án lớn của công ty.');
```

d. Bảng ThamGiaDuAn

```
INSERT INTO ThamGiaDuAn (MaNV, MaDA, NgayThamGia) VALUES
('NV01', 'DA01', '2023-01-01'),
('NV02', 'DA02', '2023-02-15'),
('NV03', 'DA03', '2023-03-20'),
('NV04', 'DA04', '2023-04-25'),
('NV05', 'DA05', '2023-05-30'),
('NV01', 'DA02', '2023-06-10'),
('NV02', 'DA03', '2023-07-15'),
('NV03', 'DA04', '2023-08-20'),
('NV04', 'DA05', '2023-09-25'),
('NV05', 'DA01', '2023-10-30');
```

IV. Câu lệnh truy vấn SQL:

1. Liệt kê tên nhân viên và tên phòng ban của họ

```
SELECT NhanVien.TenNV, PhongBan.TenPB
FROM NhanVien
INNER JOIN PhongBan ON NhanVien.MaPB = PhongBan.MaPB
ORDER BY NhanVien.TenNV;
```

2. Liệt kê tên nhân viên và dự án họ tham gia

```
SELECT NhanVien.TenNV, DuAn.TenDA
FROM NhanVien
```

```
INNER JOIN ThamGiaDuAn ON NhanVien.MaNV = ThamGiaDuAn.MaNV
INNER JOIN DuAn ON ThamGiaDuAn.MaDA = DuAn.MaDA
ORDER BY NhanVien.TenNV;
```

3. Liệt kê nhân viên thuộc phòng ban "Phòng Kỹ Thuật"

```
SELECT NhanVien.TenNV
FROM NhanVien
INNER JOIN PhongBan ON NhanVien.MaPB = PhongBan.MaPB
WHERE PhongBan.TenPB = 'Phòng Kỹ Thuật'
ORDER BY NhanVien.TenNV;
```

4. Tìm nhân viên tham gia dự án "Dự Án Xây Dựng"

```
SELECT NhanVien.TenNV
FROM NhanVien
INNER JOIN ThamGiaDuAn ON NhanVien.MaNV = ThamGiaDuAn.MaNV
INNER JOIN DuAn ON ThamGiaDuAn.MaDA = DuAn.MaDA
WHERE DuAn.TenDA = 'Dự Án Xây Dựng'
ORDER BY NhanVien.TenNV;
```

5. Liệt kê tên dự án và mô tả của nó, sắp xếp theo tên dự án

```
SELECT TenDA, MoTa
FROM DuAn
ORDER BY TenDA;
```

6. Liệt kê tên nhân viên và số điện thoại của họ, sắp xếp theo địa chỉ

```
SELECT TenNV, SoDienThoai
FROM NhanVien
ORDER BY DiaChi;
```

7. Tìm nhân viên sinh trước năm 1990

```
SELECT TenNV, NgaySinh
FROM NhanVien
WHERE NgaySinh < '1990-01-01'
ORDER BY NgaySinh;
```

8. Liệt kê tên nhân viên và phòng ban của họ, chỉ lấy nhân viên có địa chỉ chứa từ 'Quận 1'

```
SELECT NhanVien.TenNV, PhongBan.TenPB
FROM NhanVien
INNER JOIN PhongBan ON NhanVien.MaPB = PhongBan.MaPB
WHERE NhanVien.DiaChi LIKE '%Quận 1%'
ORDER BY NhanVien.TenNV;
```

9. Liệt kê các dự án mà nhân viên 'Nguyễn Văn A' tham gia

```
SELECT DuAn.TenDA
```

```
FROM DuAn
INNER JOIN ThamGiaDuAn ON DuAn.MaDA = ThamGiaDuAn.MaDA
INNER JOIN NhanVien ON ThamGiaDuAn.MaNV = NhanVien.MaNV
WHERE NhanVien.TenNV = 'Nguyễn Văn A'
ORDER BY DuAn.TenDA;
```

10. Tìm các nhân viên có số điện thoại bắt đầu bằng '090'

```
SELECT TenNV, SoDienThoai
FROM NhanVien
WHERE SoDienThoai LIKE '090%'
ORDER BY TenNV;
```

11. Liệt kê tất cả phòng ban và số lượng nhân viên trong mỗi phòng

```
SELECT PhongBan.TenPB, COUNT(NhanVien.MaNV) AS SoLuongNhanVien
FROM PhongBan
LEFT JOIN NhanVien ON PhongBan.MaPB = NhanVien.MaPB
GROUP BY PhongBan.TenPB
ORDER BY SoLuongNhanVien DESC;
```

12. Tìm các dự án có mô tả chứa từ 'phần mềm'

```
SELECT TenDA, MoTa
FROM DuAn
WHERE MoTa LIKE '%phần mềm%'
ORDER BY TenDA;
```

13. Liệt kê tên nhân viên và ngày tham gia dự án 'DA01'

```
SELECT NhanVien.TenNV, ThamGiaDuAn.NgayThamGia
FROM NhanVien
INNER JOIN ThamGiaDuAn ON NhanVien.MaNV = ThamGiaDuAn.MaNV
WHERE ThamGiaDuAn.MaDA = 'DA01'
ORDER BY ThamGiaDuAn.NgayThamGia;
```

14. Tìm nhân viên chưa tham gia bất kỳ dự án nào

```
SELECT TenNV
FROM NhanVien
WHERE MaNV NOT IN (SELECT MaNV FROM ThamGiaDuAn)
ORDER BY TenNV;
```

15. Liệt kê nhân viên theo phòng ban và sắp xếp theo ngày sinh

```
SELECT PhongBan.TenPB, NhanVien.TenNV, NhanVien.NgaySinh
FROM NhanVien
INNER JOIN PhongBan ON NhanVien.MaPB = PhongBan.MaPB
ORDER BY PhongBan.TenPB, NhanVien.NgaySinh;
```

16. Tìm nhân viên làm việc trong 'Phòng Marketing' và tham gia dự án 'DA03'

```
SELECT NhanVien.TenNV
FROM NhanVien
INNER JOIN ThamGiaDuAn ON NhanVien.MaNV = ThamGiaDuAn.MaNV
INNER JOIN PhongBan ON NhanVien.MaPB = PhongBan.MaPB
WHERE PhongBan.TenPB = 'Phòng Marketing' AND ThamGiaDuAn.MaDA = 'DA03'
ORDER BY NhanVien.TenNV;
```

17. Liệt kê các dự án có ngày tham gia gần nhất

```
SELECT DuAn.TenDA, MAX(ThamGiaDuAn.NgayThamGia) AS NgayThamGiaGanNhat
FROM DuAn
INNER JOIN ThamGiaDuAn ON DuAn.MaDA = ThamGiaDuAn.MaDA
GROUP BY DuAn.TenDA
ORDER BY NgayThamGiaGanNhat DESC;
```

18. Tìm nhân viên có số điện thoại là '0934567890'

```
SELECT TenNV, DiaChi
FROM NhanVien
WHERE SoDienThoai = '0934567890';
```

19. Liệt kê tên nhân viên và ngày tham gia dự án của họ, sắp xếp theo ngày tham gia

```
SELECT NhanVien.TenNV, ThamGiaDuAn.NgayThamGia
FROM NhanVien
INNER JOIN ThamGiaDuAn ON NhanVien.MaNV = ThamGiaDuAn.MaNV
ORDER BY ThamGiaDuAn.NgayThamGia;
```

20. Tìm các phòng ban có hơn 2 nhân viên

```
SELECT PhongBan.TenPB, COUNT(NhanVien.MaNV) AS SoLuongNhanVien
FROM PhongBan
INNER JOIN NhanVien ON PhongBan.MaPB = NhanVien.MaPB
GROUP BY PhongBan.TenPB
HAVING COUNT(NhanVien.MaNV) > 1
ORDER BY SoLuongNhanVien DESC;
```

Phần 5: CSDL QUẢN LÝ BỆNH NHÂN

Dưới đây là việc xây dựng cơ sở dữ liệu Quản Lý Bệnh Nhân đơn giản gồm 4 bảng: **BenhNhan**, **BacSi**, **PhongKham**, và **LichKham**. Bao gồm các bước từ thiết kế bảng với kiểu dữ liệu đến việc viết câu lệnh SQL để tạo các bảng và thêm dữ liệu mẫu vào mỗi bảng.

I. Thiết kế bảng:

a. Bảng BenhNhan

- MaBN**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi bệnh nhân.
- TenBN**: VARCHAR(50) - Tên của bệnh nhân.
- NgaySinh**: DATE - Ngày sinh của bệnh nhân.
- DiaChi**: VARCHAR(100) - Địa chỉ liên lạc của bệnh nhân.
- SoDienThoai**: VARCHAR(15) - Số điện thoại liên lạc của bệnh nhân.

b. Bảng BacSi

- MaBS**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi bác sĩ.
- TenBS**: VARCHAR(50) - Tên của bác sĩ.
- ChuyenKhoa**: VARCHAR(50) - Chuyên khoa của bác sĩ.
- SoDienThoai**: VARCHAR(15) - Số điện thoại liên lạc của bác sĩ.

c. Bảng PhongKham

- MaPK**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi phòng khám.
- TenPK**: VARCHAR(50) - Tên của phòng khám.
- ViTri**: VARCHAR(100) - Vị trí của phòng khám.

d. Bảng LichKham

- MaKham**: VARCHAR(10) - Khóa chính, mã định danh cho mỗi lượt khám.
- MaBN**: VARCHAR(10) - Khóa ngoại, mã bệnh nhân, liên kết với bảng **BenhNhan**.
- MaBS**: VARCHAR(10) - Khóa ngoại, mã bác sĩ, liên kết với bảng **BacSi**.
- MaPK**: VARCHAR(10) - Khóa ngoại, mã phòng khám, liên kết với bảng **PhongKham**.
- NgayKham**: DATE - Ngày khám bệnh.
- GhiChu**: TEXT - Ghi chú thêm về lượt khám.

II. Xây dựng cấu trúc bảng:

a. Tạo bảng BenhNhan

```
CREATE TABLE BenhNhan (  
    MaBN VARCHAR(10) PRIMARY KEY,  
    TenBN VARCHAR(50) NOT NULL,  
    NgaySinh DATE,  
    DiaChi VARCHAR(100),  
    SoDienThoai VARCHAR(15)  
);
```

b. Tạo bảng BacSi

```
CREATE TABLE BacSi (  
    MaBS VARCHAR(10) PRIMARY KEY,  
    TenBS VARCHAR(50) NOT NULL,  
    ChuyenKhoa VARCHAR(50),  
    SoDienThoai VARCHAR(15)  
);
```

c. Tạo bảng PhongKham

```
CREATE TABLE PhongKham (  
    MaPK VARCHAR(10) PRIMARY KEY,  
    TenPK VARCHAR(50) NOT NULL,  
    ViTri VARCHAR(100)  
);
```

d. Tạo bảng LichKham

```
CREATE TABLE LichKham (  
    MaKham VARCHAR(10) PRIMARY KEY,  
    MaBN VARCHAR(10) NOT NULL,  
    MaBS VARCHAR(10) NOT NULL,  
    MaPK VARCHAR(10) NOT NULL,  
    NgayKham DATE NOT NULL,  
    GhiChu TEXT,  
    FOREIGN KEY (MaBN) REFERENCES BenhNhan(MaBN) ON DELETE CASCADE,  
    FOREIGN KEY (MaBS) REFERENCES BacSi(MaBS) ON DELETE CASCADE,  
    FOREIGN KEY (MaPK) REFERENCES PhongKham(MaPK) ON DELETE CASCADE  
);
```

III. Dữ liệu mẫu:

a. Bảng BenhNhan

```
INSERT INTO BenhNhan (MaBN, TenBN, NgaySinh, DiaChi, SoDienThoai) VALUES  
( 'BN01', 'Nguyen Van A', '1990-01-01', '123 Duong ABC, Ha Noi', '0901234567'),  
( 'BN02', 'Tran Thi B', '1985-02-02', '456 Duong DEF, Ha Noi', '0912345678'),  
( 'BN03', 'Le Van C', '2000-03-03', '789 Duong GHI, Ha Noi', '0923456789'),  
( 'BN04', 'Pham Thi D', '1995-04-04', '123 Duong JKL, Ha Noi', '0934567890'),  
( 'BN05', 'Hoang Van E', '1988-05-05', '456 Duong MNO, Ha Noi', '0945678901'),  
( 'BN06', 'Do Thi F', '1992-06-06', '789 Duong PQR, Ha Noi', '0956789012'),  
( 'BN07', 'Nguyen Van G', '1993-07-07', '123 Duong STU, Ha Noi', '0967890123'),  
( 'BN08', 'Tran Thi H', '1987-08-08', '456 Duong VWX, Ha Noi', '0978901234'),  
( 'BN09', 'Le Van I', '1991-09-09', '789 Duong YZ, Ha Noi', '0989012345'),
```

('BN10', 'Pham Thi J', '1994-10-10', '123 Duong ABC, Ha Noi', '0990123456');

b. Bảng BacSi

INSERT INTO BacSi (MaBS, TenBS, ChuyenKhoa, SoDienThoai) VALUES

('BS01', 'Nguyen Van K', 'Noi Khoa', '0901111222'),
('BS02', 'Tran Thi L', 'Ngoai Khoa', '0911111222'),
('BS03', 'Le Van M', 'Nhi Khoa', '0921111222'),
('BS04', 'Pham Thi N', 'San Phu Khoa', '0931111222'),
('BS05', 'Hoang Van O', 'Than Kinh', '0941111222'),
('BS06', 'Do Thi P', 'Mat', '0951111222'),
('BS07', 'Nguyen Van Q', 'Tai Mui Hong', '0961111222'),
('BS08', 'Tran Thi R', 'Da Lieu', '0971111222'),
('BS09', 'Le Van S', 'Tim Mach', '0981111222'),
('BS10', 'Pham Thi T', 'Tieu Hoa', '0991111222');

c. Bảng PhongKham

INSERT INTO PhongKham (MaPK, TenPK, ViTri) VALUES

('PK01', 'Phong Kham Da Khoa', 'Tang 1, Khu A'),
('PK02', 'Phong Kham Noi Khoa', 'Tang 2, Khu B'),
('PK03', 'Phong Kham Ngoai Khoa', 'Tang 3, Khu C'),
('PK04', 'Phong Kham Nhi Khoa', 'Tang 4, Khu D'),
('PK05', 'Phong Kham San Phu Khoa', 'Tang 5, Khu E'),
('PK06', 'Phong Kham Than Kinh', 'Tang 6, Khu F'),
('PK07', 'Phong Kham Mat', 'Tang 7, Khu G'),
('PK08', 'Phong Kham Tai Mui Hong', 'Tang 8, Khu H'),
('PK09', 'Phong Kham Da Lieu', 'Tang 9, Khu I'),
('PK10', 'Phong Kham Tim Mach', 'Tang 10, Khu J');

d. Bảng LichKham

INSERT INTO LichKham (MaKham, MaBN, MaBS, MaPK, NgayKham, GhiChu) VALUES

('LK01', 'BN01', 'BS01', 'PK01', '2024-05-01', 'Kham dinh ky'),
('LK02', 'BN02', 'BS02', 'PK02', '2024-05-02', 'Kham tai noi'),
('LK03', 'BN03', 'BS03', 'PK03', '2024-05-03', 'Kham ngoai khoa'),
('LK04', 'BN04', 'BS04', 'PK04', '2024-05-04', 'Kham nhi khoa'),
('LK05', 'BN05', 'BS05', 'PK05', '2024-05-05', 'Kham san phu khoa'),
('LK06', 'BN06', 'BS06', 'PK06', '2024-05-06', 'Kham than kinh'),
('LK07', 'BN07', 'BS07', 'PK07', '2024-05-07', 'Kham mat'),
('LK08', 'BN08', 'BS08', 'PK08', '2024-05-08', 'Kham tai mui hong'),
('LK09', 'BN09', 'BS09', 'PK09', '2024-05-09', 'Kham da lieu'),
('LK10', 'BN10', 'BS10', 'PK10', '2024-05-10', 'Kham tim mach');

IV. Câu lệnh truy vấn SQL:

1. Liệt kê tên bệnh nhân và tên bác sĩ của họ

```
SELECT BenhNhan.TenBN, BacSi.TenBS  
FROM BenhNhan  
INNER JOIN LichKham ON BenhNhan.MaBN = LichKham.MaBN  
INNER JOIN BacSi ON LichKham.MaBS = BacSi.MaBS  
ORDER BY BenhNhan.TenBN;
```

2. Liệt kê tên bệnh nhân và phòng khám của họ

```
SELECT BenhNhan.TenBN, PhongKham.TenPK  
FROM BenhNhan  
INNER JOIN LichKham ON BenhNhan.MaBN = LichKham.MaBN  
INNER JOIN PhongKham ON LichKham.MaPK = PhongKham.MaPK  
ORDER BY BenhNhan.TenBN;
```

3. Liệt kê bệnh nhân sinh trước năm 1990 và tên bác sĩ của họ

```
SELECT BenhNhan.TenBN, BacSi.TenBS  
FROM BenhNhan  
INNER JOIN LichKham ON BenhNhan.MaBN = LichKham.MaBN  
INNER JOIN BacSi ON LichKham.MaBS = BacSi.MaBS  
WHERE BenhNhan.NgaySinh < '1990-01-01'  
ORDER BY BenhNhan.TenBN;
```

4. Tìm bệnh nhân đến khám trong một ngày cụ thể

```
SELECT BenhNhan.TenBN, LichKham.NgayKham  
FROM BenhNhan  
INNER JOIN LichKham ON BenhNhan.MaBN = LichKham.MaBN  
WHERE LichKham.NgayKham = '2024-05-01'  
ORDER BY BenhNhan.TenBN;
```

5. Liệt kê bác sĩ chuyên khoa 'Noi Khoa' và bệnh nhân của họ

```
SELECT BacSi.TenBS, BenhNhan.TenBN  
FROM BacSi  
INNER JOIN LichKham ON BacSi.MaBS = LichKham.MaBS  
INNER JOIN BenhNhan ON LichKham.MaBN = BenhNhan.MaBN  
WHERE BacSi.ChuyenKhoa = 'Noi Khoa'  
ORDER BY BacSi.TenBS;
```

6. Liệt kê số lượng bệnh nhân mỗi bác sĩ khám

```
SELECT BacSi.TenBS, COUNT(LichKham.MaBN) AS SoLuongBenhNhan
```

```
FROM BacSi
INNER JOIN LichKham ON BacSi.MaBS = LichKham.MaBS
GROUP BY BacSi.TenBS
ORDER BY SoLuongBenhNhan DESC;
```

7. Liệt kê số lượng bệnh nhân mỗi phòng khám

```
SELECT PhongKham.TenPK, COUNT(LichKham.MaBN) AS SoLuongBenhNhan
FROM PhongKham
INNER JOIN LichKham ON PhongKham.MaPK = LichKham.MaPK
GROUP BY PhongKham.TenPK
ORDER BY SoLuongBenhNhan DESC;
```

8. Liệt kê tên bệnh nhân và số lần khám của họ

```
SELECT BenhNhan.TenBN, COUNT(LichKham.MaKham) AS SoLanKham
FROM BenhNhan
INNER JOIN LichKham ON BenhNhan.MaBN = LichKham.MaBN
GROUP BY BenhNhan.TenBN
ORDER BY SoLanKham DESC;
```

9. Liệt kê các bác sĩ và số lượng phòng khám mà họ làm việc

```
SELECT BacSi.TenBS, COUNT(DISTINCT LichKham.MaPK) AS SoLuongPhongKham
FROM BacSi
INNER JOIN LichKham ON BacSi.MaBS = LichKham.MaBS
GROUP BY BacSi.TenBS
ORDER BY SoLuongPhongKham DESC;
```

10. Tìm bệnh nhân có tên bắt đầu bằng 'Nguyen'

```
SELECT TenBN, DiaChi
FROM BenhNhan
WHERE TenBN LIKE 'Nguyen%'
ORDER BY TenBN;
```

11. Liệt kê các phòng khám có từ 2 lượt khám trở lên

```
SELECT PhongKham.TenPK, COUNT(LichKham.MaKham) AS SoLuongKham
FROM PhongKham
INNER JOIN LichKham ON PhongKham.MaPK = LichKham.MaPK
GROUP BY PhongKham.TenPK
HAVING COUNT(LichKham.MaKham) >= 2
ORDER BY SoLuongKham DESC;
```

12. Tìm bác sĩ và ngày khám gần nhất của họ

```
SELECT BacSi.TenBS, MAX(LichKham.NgayKham) AS NgayKhamGanNhat
FROM BacSi
```

```
INNER JOIN LichKham ON BacSi.MaBS = LichKham.MaBS
GROUP BY BacSi.TenBS
ORDER BY NgayKhamGanNhat DESC;
```

13. Liệt kê tên bệnh nhân và ngày khám gần nhất của họ

```
SELECT BenhNhan.TenBN, MAX(LichKham.NgayKham) AS NgayKhamGanNhat
FROM BenhNhan
INNER JOIN LichKham ON BenhNhan.MaBN = LichKham.MaBN
GROUP BY BenhNhan.TenBN
ORDER BY NgayKhamGanNhat DESC;
```

14. Tìm các phòng khám có lượt khám vào tháng 5 năm 2024

```
SELECT PhongKham.TenPK, COUNT(LichKham.MaKham) AS SoLuongKham
FROM PhongKham
INNER JOIN LichKham ON PhongKham.MaPK = LichKham.MaPK
WHERE LichKham.NgayKham BETWEEN '2024-05-01' AND '2024-05-31'
GROUP BY PhongKham.TenPK
ORDER BY SoLuongKham DESC;
```

15. Liệt kê bệnh nhân và bác sĩ của họ, sắp xếp theo ngày khám

```
SELECT BenhNhan.TenBN, BacSi.TenBS, LichKham.NgayKham
FROM BenhNhan
INNER JOIN LichKham ON BenhNhan.MaBN = LichKham.MaBN
INNER JOIN BacSi ON LichKham.MaBS = BacSi.MaBS
ORDER BY LichKham.NgayKham DESC;
```

16. Tìm số lượt khám trong mỗi ngày trong tháng 5 năm 2024

```
SELECT LichKham.NgayKham, COUNT(LichKham.MaKham) AS SoLuongKham
FROM LichKham
WHERE LichKham.NgayKham BETWEEN '2024-05-01' AND '2024-05-31'
GROUP BY LichKham.NgayKham
ORDER BY LichKham.NgayKham;
```

17. Liệt kê các bác sĩ có chuyên khoa 'Ngoại Khoa' và số lượng bệnh nhân của họ

```
SELECT BacSi.TenBS, COUNT(LichKham.MaBN) AS SoLuongBenhNhan
FROM BacSi
INNER JOIN LichKham ON BacSi.MaBS = LichKham.MaBS
WHERE BacSi.ChuyenKhoa = 'Ngoai Khoa'
GROUP BY BacSi.TenBS
ORDER BY SoLuongBenhNhan DESC;
```

18. Liệt kê bệnh nhân và bác sĩ của họ, chỉ những bệnh nhân đã khám ít nhất 2 lần

```
SELECT BenhNhan.TenBN, BacSi.TenBS, COUNT(LichKham.MaKham) AS SoLanKham
```

```
FROM BenhNhan
INNER JOIN LichKham ON BenhNhan.MaBN = LichKham.MaBN
INNER JOIN BacSi ON LichKham.MaBS = BacSi.MaBS
GROUP BY BenhNhan.TenBN, BacSi.TenBS
HAVING COUNT(LichKham.MaKham) >= 2
ORDER BY SoLanKham DESC;
```

19. Liệt kê phòng khám và tổng số lượt khám trong tháng 5 năm 2024

```
SELECT PhongKham.TenPK, COUNT(LichKham.MaKham) AS TongSoKham
FROM PhongKham
INNER JOIN LichKham ON PhongKham.MaPK = LichKham.MaPK
WHERE LichKham.NgayKham BETWEEN '2024-05-01' AND '2024-05-31'
GROUP BY PhongKham.TenPK
ORDER BY TongSoKham DESC;
```

20. Tìm các bác sĩ có số lượng bệnh nhân khám nhiều hơn 5 người

```
SELECT BacSi.TenBS, COUNT(LichKham.MaBN) AS SoLuongBenhNhan
FROM BacSi
INNER JOIN LichKham ON BacSi.MaBS = LichKham.MaBS
GROUP BY BacSi.TenBS
HAVING COUNT(LichKham.MaBN) > 5
ORDER BY SoLuongBenhNhan DESC;
```

Phần 6: CSDL QUẢN LÝ VÉ TÀU

I. Thiết kế bảng:

a. Bảng Tau

- MaTau (VARCHAR) - Khóa chính mã tàu.
- TenTau (VARCHAR) - Tên tàu.
- LoaiTau (VARCHAR) - Loại tàu.
- SoChoNgoi (INT) - Số chỗ ngồi.

b. Bảng ChuyenTau

- MaChuyen (VARCHAR) - Khóa chính mã chuyến tàu.
- MaTau (VARCHAR) - Khóa ngoại liên kết với bảng Tau.
- GaDi (VARCHAR) - Ga đi.
- GaDen (VARCHAR) - Ga đến.
- NgayDi (DATE) - Ngày đi.
- GioDi (TIME) - Giờ đi.
- GioDen (TIME) - Giờ đến.

c. Bảng KháchHang

- **MaKH** (VARCHAR) - Khóa chính mã khách hàng.
- **TenKH** (VARCHAR) - Tên khách hàng.
- **SoDienThoai** (VARCHAR) - Số điện thoại.
- **Email** (VARCHAR) - Email.
- **DiaChi** (VARCHAR) - Địa chỉ.

d. Bảng VeTau

- **MaVe** (VARCHAR) - Khóa chính mã vé.
- **MaChuyen** (VARCHAR) - Khóa ngoại liên kết với bảng ChuyenTau.
- **MaKH** (VARCHAR) - Khóa ngoại liên kết với bảng KhachHang.
- **MaLoaiVe** (VARCHAR) - Khóa ngoại liên kết với bảng LoaiVe.
- **MaToaTau** (VARCHAR) - Khóa ngoại liên kết với bảng ToaTau.
- **NgayDat** (DATE) - Ngày đặt vé.
- **SoLuongVe** (INT) - Số lượng vé.

e. Bảng LoaiVe

- **MaLoaiVe** (VARCHAR) - Khóa chính mã loại vé.
- **TenLoaiVe** (VARCHAR) - Tên loại vé (ví dụ: Vé thường, vé VIP, vv.).
- **MoTa** (VARCHAR) - Mô tả chi tiết về loại vé.

f. Bảng ToaTau

- **MaToaTau** (VARCHAR) - Khóa chính mã toa tàu.
- **MaTau** (VARCHAR) - Khóa ngoại liên kết với bảng Tau.
- **LoaiToa** (VARCHAR) - Loại toa (ví dụ: "Toa Giường Nằm", "Toa Ghế Mềm", "Toa Ghế Ngồi Cứng"vv.).
- **SoChoNgoiToa** (INT) - Số chỗ ngồi trong toa.

II. Xây dựng cấu trúc bảng:

a. Tạo bảng Tau

```
-- Tạo bảng Tau
CREATE TABLE Tau (
    MaTau VARCHAR(50) PRIMARY KEY,
    TenTau VARCHAR(100),
    LoaiTau VARCHAR(50),
    SoChoNgoi INT
);
```

b. Tạo bảng ChuyenTau

```
-- Tạo bảng ChuyenTau
CREATE TABLE ChuyenTau (
    MaChuyen VARCHAR(50) PRIMARY KEY,
    MaTau VARCHAR(50),
    GaDi VARCHAR(100),
    GaDen VARCHAR(100),
    NgayDi DATE,
```

```
    GioDi TIME,  
    GioDen TIME,  
    FOREIGN KEY (MaTau) REFERENCES Tau (MaTau)  
);
```

c. Tạo bảng KháchHang

```
-- Tạo bảng KháchHang  
CREATE TABLE KháchHang (  
    MaKH VARCHAR(50) PRIMARY KEY,  
    TenKH VARCHAR(100),  
    NgaySinh DATE,  
    SoDienThoai VARCHAR(50),  
    Email VARCHAR(100),  
    DiaChi VARCHAR(200)  
);
```

d. Tạo bảng LoaiVe

```
-- Tạo bảng LoaiVe  
CREATE TABLE LoaiVe (  
    MaLoaiVe VARCHAR(50) PRIMARY KEY,  
    TenLoaiVe VARCHAR(100),  
    MoTa VARCHAR(255)  
);
```

e. Tạo bảng ToaTau

```
-- Tạo bảng ToaTau  
CREATE TABLE ToaTau (  
    MaToaTau VARCHAR(50) PRIMARY KEY,  
    MaTau VARCHAR(50),  
    LoaiToa VARCHAR(50),  
    SoChoNgonToa INT,  
    FOREIGN KEY (MaTau) REFERENCES Tau (MaTau)  
);
```

f. Tạo bảng VeTau

```
-- Tạo bảng VeTau  
CREATE TABLE VeTau (  
    MaVe VARCHAR(50) PRIMARY KEY,  
    MaChuyen VARCHAR(50),  
    MaKH VARCHAR(50),  
    MaLoaiVe VARCHAR(50),  
    MaToaTau VARCHAR(50),  
    NgayDat DATE,  
    SoLuongVe INT,  
    GiaVe DECIMAL(10, 2),  
    FOREIGN KEY (MaChuyen) REFERENCES ChuyenTau (MaChuyen),  
    FOREIGN KEY (MaKH) REFERENCES KháchHang (MaKH),  
    FOREIGN KEY (MaLoaiVe) REFERENCES LoaiVe (MaLoaiVe),  
    FOREIGN KEY (MaToaTau) REFERENCES ToaTau (MaToaTau)
```

);

g. Tạo bảng GiamGia

```
CREATE TABLE GiamGia (  
    MaGiamGia VARCHAR(50) PRIMARY KEY,  
    MoTa VARCHAR(255),  
    PhanTramGiam DECIMAL(5, 2),  
    DieuKien VARCHAR(255) -- Mô tả điều kiện áp dụng giảm giá  
);
```

III. Dữ liệu mẫu:

a. Bảng Tau

-- Đổ dữ liệu vào bảng Tau

```
INSERT INTO Tau (MaTau, TenTau, LoaiTau, SoChoNgon) VALUES  
( 'SE1', 'SE1', 'Express', 500),  
( 'SE2', 'SE2', 'Express', 500),  
( 'SE3', 'SE3', 'Express', 500),  
( 'TN1', 'TN1', 'Thường', 600),  
( 'TN2', 'TN2', 'Thường', 600),  
( 'TN3', 'TN3', 'Thường', 600);
```

b. Bảng ChuyenTau

-- Đổ dữ liệu vào bảng ChuyenTau

```
INSERT INTO ChuyenTau (MaChuyen, MaTau, GaDi, GaDen, NgayDi, GioDi, GioDen) VALUES  
( 'CH001', 'SE1', 'Hà Nội', 'Sài Gòn', '2024-05-20', '06:00:00', '19:00:00'),  
( 'CH002', 'SE2', 'Hà Nội', 'Sài Gòn', '2024-05-21', '07:00:00', '20:00:00'),  
( 'CH003', 'SE3', 'Hà Nội', 'Sài Gòn', '2024-05-22', '08:00:00', '21:00:00'),  
( 'CH004', 'TN1', 'Hà Nội', 'Hải Phòng', '2024-05-23', '09:00:00', '12:00:00'),  
( 'CH005', 'TN2', 'Hà Nội', 'Hải Phòng', '2024-05-24', '10:00:00', '13:00:00'),  
( 'CH006', 'TN3', 'Hà Nội', 'Hải Phòng', '2024-05-25', '11:00:00', '14:00:00');
```

c. Bảng KháchHang

-- Đổ dữ liệu vào bảng KháchHang

-- Ví dụ đổ dữ liệu khách hàng

```
INSERT INTO KháchHang (MaKH, TenKH, NgaySinh, SoDienThoai, Email, DiaChi) VALUES  
( 'KH001', 'Nguyễn Văn A', '1970-01-01', '0123456789', 'a@example.com', 'Hà Nội'),  
( 'KH002', 'Trần Thị B', '1980-02-02', '0123456790', 'b@example.com', 'Hồ Chí Minh'),  
-- Thêm 8 bản ghi nữa...
```

```
INSERT INTO KháchHang (MaKH, TenKH, SoDienThoai, Email, DiaChi) VALUES  
( 'KH001', 'Nguyễn Văn A', '0123456789', 'nguyenvana@example.com', 'Hà Nội'),  
( 'KH002', 'Trần Thị B', '0123456790', 'tranthib@example.com', 'Hồ Chí Minh'),  
( 'KH003', 'Lê Văn C', '0123456791', 'levanc@example.com', 'Đà Nẵng'),  
( 'KH004', 'Phạm Thị D', '0123456792', 'phamthid@example.com', 'Cần Thơ'),  
( 'KH005', 'Hoàng Văn E', '0123456793', 'hoangvane@example.com', 'Hải Phòng'),  
( 'KH006', 'Ngô Thị F', '0123456794', 'ngothif@example.com', 'Nha Trang'),  
( 'KH007', 'Đỗ Văn G', '0123456795', 'dovang@example.com', 'Huế'),  
( 'KH008', 'Bùi Thị H', '0123456796', 'buiithih@example.com', 'Thanh Hóa'),
```

```
('KH009', 'Vũ Văn I', '0123456797', 'vuvani@example.com', 'Bắc Ninh'),  
('KH010', 'Đặng Thị J', '0123456798', 'dangthij@example.com', 'Quảng Ninh');
```

d. Tạo bảng LoaiVe

-- Đổ dữ liệu vào bảng LoaiVe

```
INSERT INTO LoaiVe (MaLoaiVe, TenLoaiVe, MoTa) VALUES  
('LV1', 'Giường nằm', 'Giường nằm mềm'),  
('LV2', 'Ghế mềm', 'Ghế mềm theo toa'),  
('LV3', 'Ghế cứng', 'Ghế ngồi cứng theo toa');
```

e. Tạo bảng ToaTau

-- Đổ dữ liệu vào bảng ToaTau

```
INSERT INTO ToaTau (MaToaTau, MaTau, LoaiToa, SoChoNgoiToa) VALUES  
('TT1', 'SE1', 'Giường nằm', 100),  
('TT2', 'SE2', 'Ghế mềm', 200),  
('TT3', 'SE3', 'Ghế cứng', 200),  
('TT4', 'TN1', 'Giường nằm', 150),  
('TT5', 'TN2', 'Ghế mềm', 250),  
('TT6', 'TN3', 'Ghế cứng', 250);
```

f. Bảng VeTau

-- Đổ dữ liệu vào bảng VeTau

```
INSERT INTO VeTau (MaVe, MaChuyen, MaKH, MaLoaiVe, MaToaTau, NgayDat, SoLuongVe, GiaVe)  
VALUES  
('VE001', 'CH001', 'KH001', 'LV1', 'TT1', '2024-05-20', 2, 500000),  
('VE002', 'CH002', 'KH002', 'LV2', 'TT2', '2024-05-21', 1, 300000),  
('VE003', 'CH003', 'KH003', 'LV3', 'TT3', '2024-05-22', 1, 200000),  
('VE004', 'CH004', 'KH004', 'LV1', 'TT4', '2024-05-23', 3, 500000),  
('VE005', 'CH005', 'KH005', 'LV2', 'TT5', '2024-05-24', 4, 300000),  
('VE006', 'CH006', 'KH006', 'LV3', 'TT6', '2024-05-25', 2, 200000),  
('VE007', 'CH001', 'KH007', 'LV1', 'TT1', '2024-05-20', 2, 450000),  
('VE008', 'CH002', 'KH008', 'LV2', 'TT2', '2024-05-21', 1, 350000),  
('VE009', 'CH003', 'KH009', 'LV3', 'TT3', '2024-05-22', 3, 250000),  
('VE010', 'CH004', 'KH010', 'LV1', 'TT4', '2024-05-23', 2, 600000);
```

g. Bảng GiamGia

```
INSERT INTO GiamGia (MaGiamGia, MoTa, PhanTramGiam, DieuKien) VALUES  
('GG1', 'Giảm giá cho người già', 15.00, 'Áp dụng cho khách hàng từ 60 tuổi trở lên'),  
('GG2', 'Giảm giá cho trẻ em', 20.00, 'Áp dụng cho trẻ em dưới 12 tuổi'),  
('GG3', 'Giảm giá cho vé nhóm', 10.00, 'Áp dụng cho nhóm từ 10 người trở lên');
```

IV. Câu lệnh truy vấn SQL:

1. Tìm tất cả các vé đã đặt bởi một khách hàng cụ thể theo tên:

```
SELECT VeTau.MaVe, ChuyenTau.GaDi, ChuyenTau.GaDen, VeTau.NgayDat, VeTau.SoLuongVe  
FROM VeTau  
INNER JOIN KhachHang ON VeTau.MaKH = KhachHang.MaKH  
INNER JOIN ChuyenTau ON VeTau.MaChuyen = ChuyenTau.MaChuyen
```

```
WHERE KháchHang.TenKH = 'Nguyễn Văn A';
```

2. Liệt kê thông tin về các chuyến tàu đi từ Hà Nội đến Sài Gòn:

```
SELECT MaChuyen, GaDi, GaDen, NgayDi
FROM ChuyenTau
WHERE GaDi = 'Hà Nội' AND GaDen = 'Sài Gòn'
ORDER BY NgayDi;
```

3. Đếm số lượng vé đã bán cho mỗi loại vé:

```
SELECT LoaiVe.TenLoaiVe, COUNT(*) AS SoLuongVe
FROM VeTau
INNER JOIN LoaiVe ON VeTau.MaLoaiVe = LoaiVe.MaLoaiVe
GROUP BY LoaiVe.TenLoaiVe;
```

4. Tìm các khách hàng ở Hà Nội đã đặt vé:

```
SELECT DISTINCT KháchHang.TenKH
FROM KháchHang
INNER JOIN VeTau ON KháchHang.MaKH = VeTau.MaKH
WHERE KháchHang.DiaChi LIKE '%Hà Nội%';
```

5. Tổng số tiền thu được từ các vé đã bán:

```
SELECT SUM(GiaVe * SoLuongVe) AS TongDoanhThu
FROM VeTau;
```

6. Liệt kê các chuyến tàu và tổng số vé đã bán cho mỗi chuyến:

```
SELECT ChuyenTau.MaChuyen, COUNT(*) AS SoVeDaBan
FROM VeTau
INNER JOIN ChuyenTau ON VeTau.MaChuyen = ChuyenTau.MaChuyen
GROUP BY ChuyenTau.MaChuyen;
```

7. Sắp xếp các khách hàng theo số lượng vé đã mua giảm dần:

```
SELECT KháchHang.TenKH, COUNT(*) AS SoVe
FROM VeTau
INNER JOIN KháchHang ON VeTau.MaKH = KháchHang.MaKH
GROUP BY VeTau.MaKH
ORDER BY SoVe DESC;
```

8. Tìm thông tin về toa tàu và loại toa cho mỗi mã tàu:

```
SELECT Tau.TenTau, ToaTau.LoaiToa, ToaTau.SoChoNgoiToa
FROM ToaTau
INNER JOIN Tau ON ToaTau.MaTau = Tau.MaTau;
```

9. Liệt kê các chuyến tàu cùng ngày đi:

```
SELECT a.MaChuyen, b.MaChuyen, a.GaDi, a.GaDen
FROM ChuyenTau a, ChuyenTau b
WHERE a.MaChuyen != b.MaChuyen AND a.NgayDi = b.NgayDi;
```

10. Liệt kê các vé đã bán cho chuyến tàu có mã là 'CH001':

```
SELECT VeTau.MaVe, KháchHang.TenKH, VeTau.SoLuongVe
FROM VeTau
INNER JOIN KháchHang ON VeTau.MaKH = KháchHang.MaKH
WHERE VeTau.MaChuyen = 'CH001';
```

6. Truy vấn thông tin vé tàu và tên khách hàng cho mỗi vé đã mua trên chuyến tàu 'C01':

```
```sql
```

```
SELECT VeTau.MaVe, KháchHang.TenKH
```

```
FROM VeTau
JOIN KhachHang ON VeTau.MaKH = KhachHang.MaKH
WHERE VeTau.MaChuyen = 'C01';
'''
```

7. Truy vấn danh sách các chuyến tàu có giờ khởi hành trước 10:00 sáng:

```
```sql
SELECT MaChuyen, GaDi, GaDen, GioDi
FROM ChuyenTau
WHERE GioDi < '10:00'
ORDER BY GioDi;
'''
```

8. Truy vấn tổng số chỗ ngồi của các tàu loại 'Loại 1':

```
```sql
SELECT SUM(SoChoNgoi) AS TongSoChoLoai1
FROM Tau
WHERE LoaiTau = 'Loại 1';
'''
```

9. Truy vấn số lượng khách hàng từng thành phố:

```
```sql
SELECT DiaChi, COUNT(MaKH) AS SoLuongKhachHang
FROM KhachHang
GROUP BY DiaChi
ORDER BY SoLuongKhachHang DESC;
'''
```

10. Truy vấn chi tiết chuyến đi và tên tàu cho mỗi chuyến:

```
```sql
SELECT ChuyenTau.MaChuyen, Tau.TenTau, ChuyenTau.GaDi, ChuyenTau.GaDen
FROM ChuyenTau
JOIN Tau ON ChuyenTau.MaTau = Tau.MaTau
ORDER BY ChuyenTau.NgayDi;
'''
```

## Phần 7: CSDL QUẢN LÝ BÁN HÀNG CÀ PHÊ

### I. Thiết kế bảng:

#### a. Bảng Sản Phẩm

- **MaSP** (VARCHAR) - Primary Key, Product ID.
- **TenSP** (VARCHAR) - Product Name.
- **DonViTinh** (VARCHAR) - Unit of Measure.
- **GiaBan** (FLOAT) - Sale Price.

#### b. Bảng Nhân Viên

- **MaNV** (VARCHAR) - Primary Key, Employee ID.
- **TenNV** (VARCHAR) - Employee Name.
- **ChucVu** (VARCHAR) - Position.
- **SoDienThoai** (VARCHAR) - Phone Number.

#### c. Bảng Khách Hàng

- **MaKH** (VARCHAR) - Primary Key, Customer ID.
- **TenKH** (VARCHAR) - Customer Name.
- **DiaChi** (VARCHAR) - Address.
- **SoDienThoai** (VARCHAR) - Phone Number.
- **Email** (VARCHAR) - Email.

#### d. Bảng Hóa Đơn

- **MaHD** (VARCHAR) - Primary Key, Invoice ID.
- **MaNV** (VARCHAR) - Foreign Key linked to Nhân Viên.
- **MaKH** (VARCHAR) - Foreign Key linked to Khách Hàng.
- **MaSP** (VARCHAR) - Foreign Key linked to Sản Phẩm
- **NgayLap** (DATE) - Date Issued.
- **TongTien** (FLOAT) - Total Amount.

### II. Xây dựng cấu trúc bảng:

#### a. Tạo bảng Sản Phẩm

```
CREATE TABLE SanPham (
 MaSP VARCHAR(10) PRIMARY KEY,
 TenSP VARCHAR(50) NOT NULL,
 DonViTinh VARCHAR(10) NOT NULL,
 GiaBan FLOAT CHECK (GiaBan > 0)
);
```

#### b. Tạo bảng Nhân Viên

```
CREATE TABLE NhanVien (
 MaNV VARCHAR(10) PRIMARY KEY,
 TenNV VARCHAR(50) NOT NULL,
 ChucVu VARCHAR(30) NOT NULL,
 SoDienThoai VARCHAR(15) NOT NULL
);
```

#### c. Tạo bảng Khách Hàng

```
CREATE TABLE KhachHang (
 MaKH VARCHAR(10) PRIMARY KEY,
```

```
TenKH VARCHAR(50) NOT NULL,
DiaChi VARCHAR(100),
SoDienThoai VARCHAR(15),
Email VARCHAR(50)
);
```

#### d. Tạo bảng Hóa Đơn

```
CREATE TABLE HoaDon (
 MaHD VARCHAR(10) PRIMARY KEY,
 MaNV VARCHAR(10) NOT NULL,
 MaKH VARCHAR(10) NOT NULL,
 MaSP VARCHAR(10) NOT NULL,
 NgayLap DATE NOT NULL,
 TongTien FLOAT NOT NULL,
 FOREIGN KEY (MaNV) REFERENCES NhanVien (MaNV),
 FOREIGN KEY (MaKH) REFERENCES KhachHang (MaKH),
 FOREIGN KEY (MaSP) REFERENCES SanPham (MaSP)
);
```

### III. Dữ liệu mẫu:

#### a. Bảng Sản Phẩm

```
INSERT INTO SanPham (MaSP, TenSP, DonViTinh, GiaBan) VALUES
('SP01', 'Espresso', 'Ly', 40000),
('SP02', 'Cappuccino', 'Ly', 45000),
('SP03', 'Latte', 'Ly', 50000),
('SP04', 'Mocha', 'Ly', 55000),
('SP05', 'Americano', 'Ly', 35000),
('SP06', 'Macchiato', 'Ly', 50000),
('SP07', 'Flat White', 'Ly', 50000),
('SP08', 'Iced Coffee', 'Ly', 45000),
('SP09', 'Vietnamese Coffee', 'Ly', 30000),
('SP10', 'Frappuccino', 'Ly', 60000);
```

#### b. Bảng Nhân Viên

```
INSERT INTO NhanVien (MaNV, TenNV, ChucVu, SoDienThoai) VALUES
('NV01', 'Nguyễn Thảo', 'Manager', '0987123456'),
('NV02', 'Lê Hoàng', 'Barista', '0987234567'),
('NV03', 'Trần Huy', 'Barista', '0987345678'),
('NV04', 'Phạm Linh', 'Waitress', '0987456789'),
('NV05', 'Hoàng Anh', 'Waitress', '0987567890'),
('NV06', 'Trần Khoa', 'Chef', '0987678901'),
('NV07', 'Ngô Quý', 'Security', '0987789012'),
('NV08', 'Luu Mai', 'Cleaner', '0987890123'),
('NV09', 'Đỗ Duy', 'Accountant', '0987901234'),
('NV10', 'Lê Khánh', 'Assistant Manager', '0987012345');
```

#### c. Bảng Khách Hàng

```
INSERT INTO KhachHang (MaKH, TenKH, DiaChi, SoDienThoai, Email) VALUES
('KH01', 'Phạm Vũ', 'Hà Nội', '0912345671', 'phamvu@example.com'),
('KH02', 'Đỗ Trang', 'Đà Nẵng', '0912345672', 'dotrang@example.com'),
('KH03', 'Nguyễn Nam', 'Hồ Chí Minh', '0912345673', 'nguyennam@example.com'),
('KH04', 'Lê Hà', 'Nha Trang', '0912345674', 'leha@example.com'),
('KH05', 'Trần Phương', 'Vũng Tàu', '0912345675', 'tranphuong@example.com'),
('KH06', 'Hoàng Minh', 'Cần Thơ', '0912345676', 'hoangminh@example.com'),
('KH07', 'Lý Khuê', 'Hải Phòng', '0912345677', 'lykhue@example.com'),
('KH08', 'Đặng Hà', 'Quảng Ninh', '0912345678', 'dangha@example.com'),
('KH09', 'Phan Sơn', 'Phú Quốc', '0912345679', 'phanso@example.com'),
('KH10', 'Ngô Lan', 'Bình Dương', '0912345680', 'ngolan@example.com');
```

#### d. Bảng Hóa Đơn

```
INSERT INTO HoaDon (MaHD, MaNV, MaKH, MaSP, NgayLap, TongTien) VALUES
('HD01', 'NV01', 'KH01', 'SP01', '2024-01-01', 150000),
('HD02', 'NV02', 'KH02', 'SP02', '2024-01-02', 120000),
('HD03', 'NV03', 'KH03', 'SP03', '2024-01-03', 180000),
('HD04', 'NV04', 'KH04', 'SP04', '2024-01-04', 90000),
('HD05', 'NV05', 'KH05', 'SP05', '2024-01-05', 210000),
('HD06', 'NV06', 'KH06', 'SP06', '2024-01-06', 130000),
('HD07', 'NV07', 'KH07', 'SP07', '2024-01-07', 80000),
('HD08', 'NV08', 'KH08', 'SP08', '2024-01-08', 200000),
('HD09', 'NV09', 'KH09', 'SP09', '2024-01-09', 150000),
('HD10', 'NV10', 'KH10', 'SP10', '2024-01-10', 170000);
```

#### IV. Câu lệnh truy vấn SQL:

1. Liệt kê tất cả hóa đơn và tên khách hàng tương ứng:

```
SELECT HoaDon.MaHD, KháchHang.TenKH
FROM HoaDon
INNER JOIN KháchHang ON HoaDon.MaKH = KháchHang.MaKH;
```

2. Danh sách tất cả các sản phẩm có giá bán trên 30,000 VND:

```
SELECT TenSP, GiaBan
FROM SanPham
WHERE GiaBan > 30000
ORDER BY GiaBan DESC;
```

3. Danh sách các sản phẩm chưa bao giờ được bán:

```
SELECT SanPham.TenSP
FROM SanPham
LEFT JOIN HoaDon ON SanPham.MaSP = HoaDon.MaSP
WHERE HoaDon.MaHD IS NULL;
```

4. Các khách hàng đã chi tiêu trên 150,000 VND cho một hóa đơn:

```
SELECT KháchHang.TenKH, HoaDon.TongTien
FROM KháchHang
INNER JOIN HoaDon ON KháchHang.MaKH = HoaDon.MaKH
WHERE HoaDon.TongTien > 150000;
```

5. Tổng doanh thu theo ngày:

```
SELECT NgayLap, SUM(TongTien) AS TongDoanhThu
FROM HoaDon
GROUP BY NgayLap
```

## ORDER BY NgayLap;

. Liệt kê tổng doanh thu và số hóa đơn theo từng nhân viên:

```
SELECT NhanVien.TenNV, COUNT(HoaDon.MaHD) AS SoHoaDon, SUM(HoaDon.TongTien) AS
TongDoanhThu
FROM HoaDon
INNER JOIN NhanVien ON HoaDon.MaNV = NhanVien.MaNV
GROUP BY NhanVien.TenNV
ORDER BY TongDoanhThu DESC;
```

Thực hành với CSDL:

I. Thiết kế bảng:

II. Xây dựng cấu trúc bảng:

III. Dữ liệu mẫu:

IV. Câu lệnh truy vấn SQL: