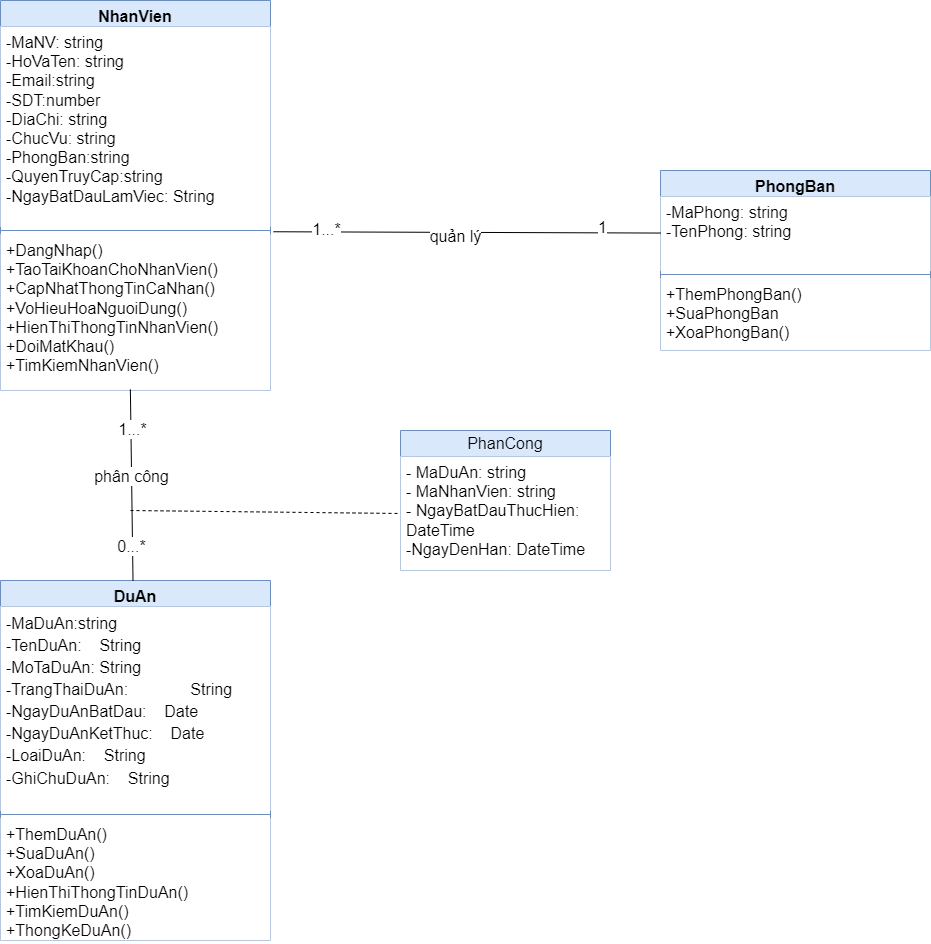
# Thiết kế lớp

## Sơ đồ lớp



## Đặc tả phương thức lớp

### Lớp NhanVien

#### *DangNhap()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaNV * MatKhau | * String (20) * String (50) |
| **Outputs** | Đăng nhập thành công | Boolean |
| **Algorithm specification** | 1. Kiểm tra xem tài khoản và mật khẩu được cung cấp có khớp với dữ liệu trong cơ sở dữ liệu hay không. 2. Nếu khớp, cập nhật trạng thái đăng nhập của nhân viên thành "đã đăng nhập".  * Trả về kết quả kiểm tra:  Trả về *True* nếu đăng nhập thành công, *False* nếu đăng nhập thất bại. | |
| **Notes** | Phương thức này nên được bảo mật để ngăn chặn truy cập trái phép vào hệ thống. | |

#### *TaoTaiKhoanChoNhanVien()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaNV * HoVaTen * Email * SDT * DiaChi * ChucVu * PhongBan * QuyenTruyCap * NgayBatDauLamViec * MatKhau | * string (20) * string (100) * string (100) * number * string (100) * string (100) * string (100) * string (100) * datetime * string (30) |
| **Outputs** | Tạo tài khoản cho nhân viên thành công | Boolean |
| **Algorithm specification** | 1. Kiểm tra thông tin nhập vào:  * Kiểm tra xem mã nhân viên, email đã được sử dụng hay chưa. * Sử dụng truy vấn SQL để kiểm tra xem mã nhân viên, email và đã tồn tại trong cơ sở dữ liệu hay chưa.  1. Tạo tài khoản:  * Nếu chưa được sử dụng, tạo tài khoản mới cho nhân viên trong cơ sở dữ liệu. * Sử dụng truy vấn SQL để tạo tài khoản mới cho nhân viên trong cơ sở dữ liệu. * Lưu trữ thông tin cá nhân, thông tin tài khoản và mật khẩu của nhân viên.  1. Cập nhật thông tin tài khoản và mật khẩu của nhân viên mới:  * Gửi email thông báo cho nhân viên mới về thông tin tài khoản và mật khẩu.  1. Trả về kết quả tạo tài khoản | |
| **Notes** | * Nên đảm bảo thông tin nhân viên được nhập chính xác và đầy đủ. * Nên sử dụng kỹ thuật mã hóa mật khẩu để bảo vệ thông tin người dùng. | |

#### *CapNhatThongTinCaNhan()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * HoVaTen * Email * SDT * DiaChi | * String (20) * string (100) * number * string (100) |
| **Outputs** | Cập nhật thông tin thành công | Boolean |
| **Algorithm specification** | 1. Kiểm tra tính hợp lệ của thông tin cập nhật:  * Xác minh xem thông tin mới có hợp lệ hay không, ví dụ như:   + Họ và tên không được để trống.   + Địa chỉ email phải có định dạng hợp lệ.   + Số điện thoại phải có định dạng hợp lệ.   + Địa chỉ nhà không được để trống. * Nếu thông tin không hợp lệ, hiển thị thông báo lỗi cho nhân viên.  1. Cập nhật thông tin cá nhân trong cơ sở dữ liệu:  * Sử dụng truy vấn SQL để cập nhật thông tin cá nhân của nhân viên trong bảng nhanvien. * Cập nhật các trường HoVaTen, Email, SDT, DiaChi với giá trị mới. * Lưu ý đảm bảo tính nguyên tử của thao tác cập nhật để tránh dữ liệu bị mâu thuẫn.  1. Gửi thông báo cập nhật thông tin thành công:  * Gửi email hoặc thông báo cho nhân viên biết rằng thông tin cá nhân của họ đã được cập nhật thành công.  1. Trả về kết quả:  * Trả về *True* nếu cập nhật thông tin thành công, *False* nếu gặp lỗi trong quá trình cập nhật. | |
| **Notes** | * Nhân viên chỉ nên được phép cập nhật thông tin cá nhân của mình. * Nên ghi chép nhật ký hoạt động để theo dõi việc cập nhật thông tin của nhân viên. | |

#### *VoHieuHoaNguoiDung()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaNV | * String (20) |
| **Outputs** | Tài khoản người dùng đã bị vô hiệu hóa | Boolean |
| **Algorithm specification** | 1. Kiểm tra tính hợp lệ của mã nhân viên:  * Xác minh xem mã nhân viên được cung cấp có tồn tại trong hệ thống hay không. * Nếu mã nhân viên không hợp lệ, hiển thị thông báo lỗi cho người dùng.  1. Cập nhật trạng thái tài khoản của người dùng:  * Sử dụng truy vấn SQL để cập nhật trạng thái tài khoản của người dùng trong bảng nhanvien. * Cập nhật “vô hiệu hóa” cho tài khoản nhân viên có MaNV được cung cấp | |
| **Notes** | * Đảm bảo rằng người dùng đang được vô hiệu hóa không còn quyền truy cập vào hệ thống. * Việc vô hiệu hóa tài khoản không làm ảnh hưởng đến các dữ liệu liên quan đến người dùng đó. * Ghi chép nhật ký hoạt động để theo dõi việc vô hiệu hóa tài khoản người dùng. | |

#### *HienThiThongTinNhanVien()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaNV | * String (20) |
| **Outputs** | * MaNV * HoVaTen * Email * SDT * DiaChi * ChucVu * PhongBan * NgayBatDauLamViec | * String (20) * String (100) * Number * String (100) * String (100) * String (100) * String (100) * Datetime |
| **Algorithm specification** | 1. Kiểm tra tính hợp lệ của mã nhân viên:  * Xác minh xem mã nhân viên được cung cấp có tồn tại trong hệ thống hay không. * Nếu mã nhân viên không hợp lệ, hiển thị thông báo lỗi cho người dùng.  1. Hiển thị thông tin nhân viên qua việc truy vấn MaNV trên cơ sở dữ liệu | |
| **Notes** | 1. Đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể truy cập thông tin chi tiết về nhân viên 2. Đảm bảo thông tin truy xuất chính xác và được cập nhật | |

#### *DoiMatKhau()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaNV * MatKhauMoi | * string (20) * string (30) |
| **Outputs** | Đổi mật khẩu thành công | Boolean |
| **Algorithm specification** | 1. Kiểm tra tính hợp lệ của mã nhân viên:  * Xác minh xem mã nhân viên được cung cấp có tồn tại trong hệ thống hay không. * Nếu mã nhân viên không hợp lệ, hiển thị thông báo lỗi cho người dùng.  1. Kiểm tra tính hợp lệ của mật khẩu mới:  * Sử dụng các quy tắc bảo mật mật khẩu để đảm bảo rằng mật khẩu mới đáp ứng các yêu cầu về độ dài, độ phức tạp và bảo mật. * Nếu mật khẩu không hợp lệ, thông báo “Mật khẩu không hợp lệ”.  1. Mã hóa mật khẩu mới:  * Sử dụng thuật toán mã hóa mật khẩu để mã hóa mật khẩu mới. * Lưu ý rằng mật khẩu được mã hóa không thể được giải mã.  1. Cập nhật mật khẩu mới cho nhân viên:  * Sử dụng truy vấn SQL để cập nhật mật khẩu đã mã hóa cho nhân viên trong bảng NhanVien.  1. Trả về kết quả:  * Trả về *True* nếu đổi mật khẩu thành công. * Trả về *False* nếu đổi mật khẩu thất bại do lỗi truy cập cơ sở dữ liệu hoặc các lỗi khác. | |
| **Notes** | * Mật khẩu mới phải đáp ứng các yêu cầu về độ dài, độ phức tạp và bảo mật. * Việc đổi mật khẩu cần được thực hiện một cách an toàn. | |

#### *TimKiemNhanVien()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * TuKhoa() | * string (100) |
| **Outputs** | List <NhanVien>   * MaNV * HoVaTen * Email * SDT * ChucVu | * string (20) * string (20) * string (100) * number * string (100) |
| **Algorithm specification** | * 1. Xử lý từ khóa: * Chuyển đổi từ khóa sang dạng chữ thường để so sánh không phân biệt chữ hoa chữ thường. * Xóa các ký tự trắng thừa trong từ khóa. * Chia nhỏ từ khóa thành các từ riêng lẻ để tìm kiếm nhiều trường dữ liệu.   1. Tạo truy vấn SQL: * Sử dụng truy vấn SQL động để tìm kiếm nhân viên trong bảng *NhanVien*. * Truy vấn SQL cần bao gồm các điều kiện tìm kiếm dựa trên từ khóa đã được xử lý.   1. Xử lý kết quả: * Lặp qua kết quả truy vấn và lưu trữ thông tin chi tiết về mỗi nhân viên vào một từ điển. * Thêm từ điển chứa thông tin nhân viên vào danh sách kết quả.   1. Trả về danh sách kết quả: * Trả về danh sách List[dict] chứa thông tin chi tiết về các nhân viên phù hợp với từ khóa tìm kiếm. | |
| **Notes** |  | |

### Lớp PhongBan

#### *ThemPhongBan()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaPhong: * TenPhong: | * String(20) * String(100) |
| **Outputs** | Thêm phòng ban thành công | Boolean |
| **Algorithm specification** | 1. Kiểm tra xem mã phòng ban đã tồn tại trong hệ thống hay chưa.  * Nếu mã phòng ban đã tồn tại, hiển thị thông báo lỗi và kết thúc phương thức.  1. Tạo một đối tượng mới của lớp PhongBan với các thuộc tính được truyền vào. 2. Thêm đối tượng phòng ban mới vào danh sách các phòng ban trong hệ thống. 3. Hiển thị thông báo thành công. | |
| **Notes** | Chú ý tính nhất quán của dữ liệu | |

#### *SuaPhongBan()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaPhong | * string (20) |
| **Outputs** | Sửa phòng ban thành công | boolean |
| **Algorithm specification** | * 1. Kiểm tra xem mã phòng ban cần sửa có tồn tại trong hệ thống hay chưa. * Nếu mã phòng ban không tồn tại, hiển thị thông báo lỗi và kết thúc phương thức.   1. Tìm đối tượng phòng ban trong danh sách các phòng ban với mã phòng ban được truyền vào.   2. Cập nhật thông tin của đối tượng phòng ban với các giá trị mới được truyền vào.   3. Hiển thị thông báo thành công. | |
| **Notes** | Đảm bảo tính nhất quán của dữ liệu | |

#### *XoaPhongBan()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaPhong: | * String (20) |
| **Outputs** | Xóa phòng ban thành công | Boolean |
| **Algorithm specification** | * 1. Kiểm tra xem mã phòng ban cần xóa có tồn tại trong hệ thống hay chưa. * Nếu mã phòng ban không tồn tại, hiển thị thông báo lỗi và kết thúc phương thức.   1. Tìm đối tượng phòng ban trong danh sách các phòng ban với mã phòng ban được truyền vào.   2. Kiểm tra xem phòng ban cần xóa có nhân viên hay không. * Nếu phòng ban có nhân viên, hiển thị thông báo lỗi và kết thúc phương thức.   1. Xóa đối tượng phòng ban khỏi danh sách các phòng ban.   2. Hiển thị thông báo thành công. | |
| **Notes** | Đảm bảo tính nhất quán của dữ liệu | |

### Lớp DuAn

#### *ThemDuAn()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaDuAn: * TenDuAn: * MoTaDuAn: * TrangThaiDuAn: * NgayDuAnBatDau: * NgayDuAnKetThuc: * LoaiDuAn: * GhiChuDuAn: | * String (20) * String (100) * String (1000) * String (100) * Datetime * Datetime * String (100) * String (100) |
| **Outputs** | Tạo dự án thành công | Boolean |
| **Algorithm specification** | 1. Kiểm tra xem MaDuAn đã tồn tại trong hệ thống hay chưa.  * Nếu mã dự án đã tồn tại, hiển thị thông báo lỗi và kết thúc phương thức.  1. Tạo một đối tượng mới của lớp Dự Án với các thuộc tính được truyền vào. 2. Thêm đối tượng dự án mới vào danh sách các dự án trong hệ thống. 3. Hiển thị thông báo thành công. | |
| **Notes** | Đảm bảo tính nhất quán dữ liệu | |

#### *SuaDuAn()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaDuAn | * String (20) |
| **Outputs** | Sửa thông tin dự án thành công | Boolean |
| **Algorithm specification** | * 1. Kiểm tra tính hợp lệ MaDuAn: * Xác minh xem thông tin mới có hợp lệ hay không, ví dụ như: * Nếu thông tin không hợp lệ, hiển thị thông báo lỗi cho nhân viên.   1. Cập nhật thông tin dự án trong cơ sở dữ liệu: * Sử dụng truy vấn SQL để sửa đổi dự án. * Cập nhật các trường thuộc lớp dự án * Lưu ý đảm bảo tính nguyên tử của thao tác cập nhật để tránh dữ liệu bị mâu thuẫn.   1. Trả về kết quả: * Trả về True nếu cập nhật thông tin thành công, False nếu gặp lỗi trong quá trình cập nhật. | |
| **Notes** | Đảm bảo tính nhất quán dữ liệu | |

#### *XoaDuAn()*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | - MaDuAn | * String (20) |
| **Outputs** | Xóa dự án thành công | Boolean |
| **Algorithm specification** | * 1. Kiểm tra tính hợp lệ của MaDuAn: * Xác minh xem MaDuAn được cung cấp có tồn tại trong hệ thống hay không. * Nếu mã nhân viên không hợp lệ, hiển thị thông báo lỗi cho người dùng.   1. Cập nhật trạng thái tài khoản của người dùng: * Sử dụng truy vấn SQL để truy vấn dự án cần xóa * Chọn xóa dự án   1. Trả về kết quả: * Trả về *True* nếu xóa thành công, *False* nếu gặp lỗi trong quá trình xóa. | |
| **Notes** | Ghi chép nhật ký hoạt động để theo dõi việc xóa dự án và cần phục hồi khi cần thiết. | |

#### *HienThiThongTinDuAn():*

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * MaDuAn | * String (20) |
| **Outputs** | * MaDuAn: * TenDuAn: * MoTaDuAn: * TrangThaiDuAn: * NgayDuAnBatDau: * NgayDuAnKetThuc: * LoaiDuAn: * GhiChuDuAn: | * String (20) * String (100) * String (1000) * String (100) * Datetime * Datetime * String (100) * String (100) |
| **Algorithm specification** | * 1. Kiểm tra tính hợp lệ của MaDuAn: * Xác minh xem mã dự án được cung cấp có tồn tại trong hệ thống hay không. * Nếu mã không hợp lệ, hiển thị thông báo lỗi cho người dùng.   1. Hiển thị thông tin dự án qua việc truy vấn MaDuAn trên cơ sở dữ liệu | |
| **Notes** | * Đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể truy cập thông tin chi tiết về dự án. * Đảm bảo thông tin truy xuất chính xác và được cập nhật | |

#### *TimKiemDuAn()*

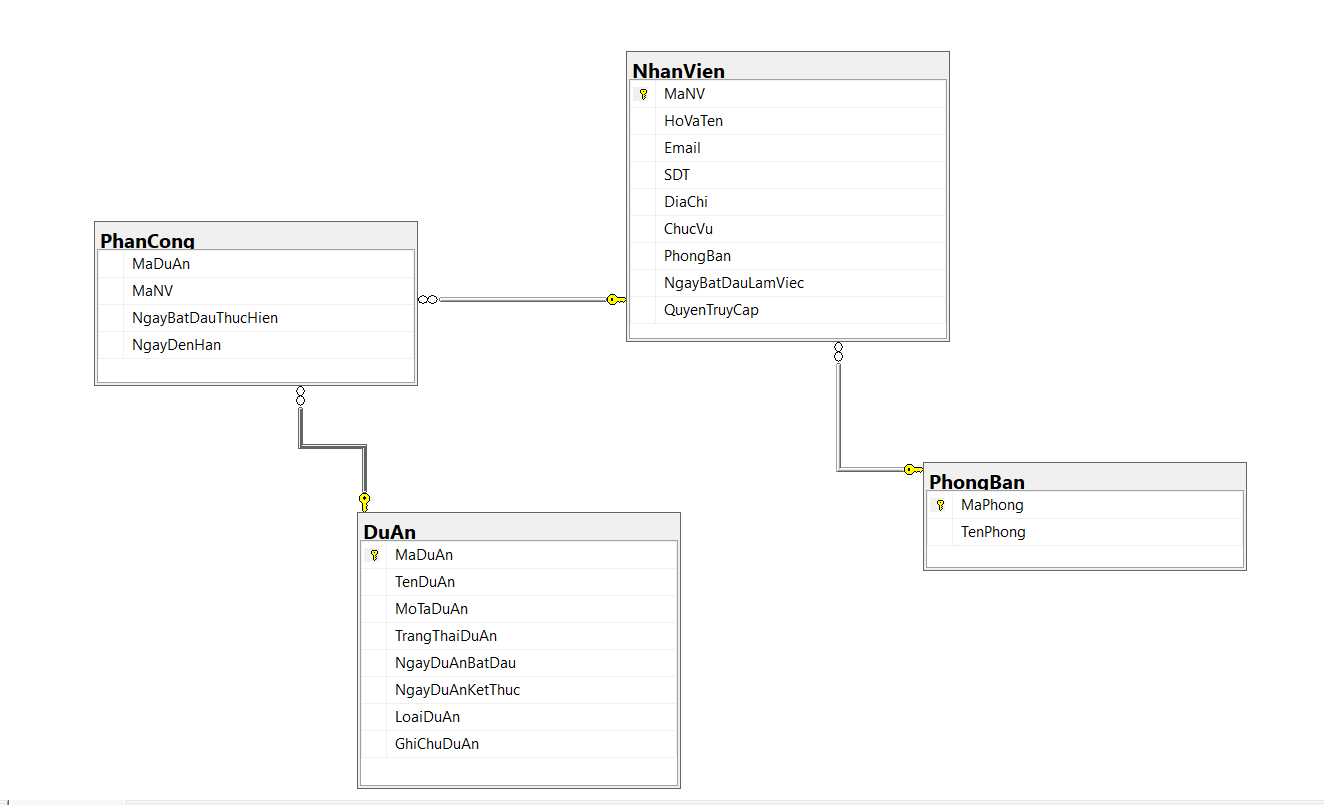
|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * TuKhoa | * string (100) |
| **Outputs** | List <DuAn> |  |
| **Algorithm specification** | * 1. Xử lý từ khóa: * Chuyển đổi từ khóa sang dạng chữ thường để so sánh không phân biệt chữ hoa chữ thường.   1. Tạo truy vấn SQL:   2. Sử dụng truy vấn SQL động để tìm kiếm dự án * Truy vấn SQL cần bao gồm các điều kiện tìm kiếm dựa trên từ khóa đã được xử lý.   1. Xử lý kết quả: * Lặp qua kết quả truy vấn và lưu trữ thông tin chi tiết về mỗi nhân viên vào một từ điển. * Thêm từ điển chứa thông tin nhân viên vào danh sách kết quả.   1. Trả về danh sách kết quả: * Trả về danh sách List[dict] chứa thông tin chi tiết về các dự án phù hợp với từ khóa tìm kiếm. | |
| **Notes** |  | |

#### *ThongKeDuAn****()***

|  |  |  |
| --- | --- | --- |
|  | Name | Data type |
| **Inputs** | * List <DuAn> * TrangThaiDuAn: * LoaiDuAn: * NgayDuAnBatDau: * NgayDuAnKetThuc | * List * String (100) * String (100) * DateTime * Datetime |
| **Outputs** | Tạo báo cáo thống kê | Png/ charts |
| **Algorithm specification** | 1. Tạo danh sách hệ thống các dự án 2. Dựa trên mong muốn của người dùng thống kê theo thời gian/ theo phân loại/ hay tiến độ 3. Truy vấn SQL theo mong muốn của người dùng, sau đó lọc ra danh sách phù hợp để tạo ra các báo cáo, biểu đồ mong muốn 4. Trả kết quả thống kê dự án | |
| **Notes** | Đảm bảo tính nhất quán của dữ liệu | |

# Thiết kế cơ sở dữ liệu

## Sơ đồ quan hệ



## Mô tả chi tiết cơ sở dữ liệu quan hệ

Mô tả chi tiết cơ sở dữ liệu dưới đây được cài đặt trên hệ thống quản lý cơ sở dữ liệu Microsoft SQL Server

### Bảng PhongBan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng: PhongBan | | | | |
| Tên | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| MaPhong | Nchar(20) | Not null | PK | Mã phòng ban, khóa chính của bảng |
| TenPhong | Nvarchar(100) | Not null |  | Tên phòng ban |

### Bảng NhanVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng: NhanVien | | | | |
| Tên | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| MaNV | Nchar(20) | Not null | PK | Mã nhân viên, khóa chính của bảng |
| HoVaTen | Nvarchar(100) | Not null |  | Họ và tên nhân viên |
| Email | Nvarchar(100) | Not null |  | Email của nhân viên |
| SDT | Nvarchar(100) | Not null |  | Số điện thoại nhân viên |
| DiaChi | Nvarchar(100) | Not null |  | Địa chỉ sinh sống của nhân viên |
| ChucVu | Nvarchar(100) | Not null |  | Chức vụ nhân viên |
| PhongBan | Nchar(20) | Not null | FK | Mã phòng ban làm việc (MaPhongBan) , khóa ngoại |
| NgayBatDauLamViec | date | notnull |  | Ngày bắt đầu làm việc tại công ty |
| QuyenTruyCap | Nvarchar(100) |  |  | Quyền truy cập |

### Bảng DuAn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng: NhanVien | | | | |
| Tên | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| MaDuAn | Nchar(20) | Not null | PK | Mã dự án, khóa chính của bảng |
| TenDuAn | Nvarchar(100) | Not null |  | Tên của dự án |
| MoTaDuAn | Nvarchar(200) |  |  | Mô tả dự án |
| TrangThaiDuAn | Nvarchar(100) | Not null |  | Trạng thái (tiến độ) dự án |
| NgayDuAnBatDau | date | Not null |  | Ngày dự án bắt đầu |
| NgayDuAnKetThuc | date |  |  | Ngày dự án kết thúc |
| LoaiDuAn | Nvarchar(200) | Not null |  | Phân loại của dự án |
| GhiChuDuAn | Nvarchar(200) |  |  | Ghi chú về dự án |

### Bảng PhanCong

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bảng: NhanVien | | | | |
| Tên | **Kiểu dữ liệu** | **Ràng buộc** | **Khóa** | **Mô tả** |
| MaDuAn | Varchar(20) | Not null | PK | Mã dự án được phân công |
| MaNV | Varchar(20) | Not null | PK | Mã nhân viên được phân công thực hiện dự án |
| NgayBatDauThucHien | date | Not null |  | Ngày được phân công thực hiện |
| NgayDenHan | date | Not null |  | Ngày đến hạn buộc phải hoàn thành |