LỜI NÓI ĐẦU

Ngày nay, các ứng dụng công nghệ thông tin đã được áp dụng trong mọi lĩnh vực nghiên cứu khoa học kĩ thuật cũng như trong đời sống, máy tính trở thành công cụ trợ giúp đắc lực cho con người trong lưu trữ, phân tích và xử lý thông tin. Ở nước ta, cùng với sự phát triển chung của nền kinh tế, tin học đang ngày được ứng dụng rộng rãi trong công tác quản lý và từng bước khẳng định sức mạnh cũng như vị trí quan trọng của mình.

Với sự hỗ trợ của phần mềm, con người sẽ dễ dàng tiếp cận thêm tới những dịch vụ đa phương tiện mới, nhanh gọn và chi phí thấp, chúng ta có thể dễ dàng xem những bộ phim yêu thích trên Internet, không lo bị lỡ mất những thước phim hay trên truyền hình truyền thống, không phải ra rạp chiếu phim. Cho nên nhóm em quyết định chọn đề tài “website xem phim online” để ứng dụng công nghệ thông tin vào cung cấp dịch vụ xem phim trực tuyến.

Em xin chân thành cảm ơn Thầy đã trang bị cho em những kiến thức cơ bản cần thiết về môn cơ sở dữ liệu nâng cao để em có thể hoàn thành tốt đề tài này.

Mục Lục

[I. Giới thiệu sơ lược về cơ sở dữ liệu quản lý “Website xem phim online” 3](#_Toc70228395)

[II. Các khái niệm cơ bản 3](#_Toc70228396)

[III. Thiết kế ở mức khái niệm 3](#_Toc70228397)

[IV. Xác định RBTV 5](#_Toc70228398)

[1) Định nghĩa 5](#_Toc70228399)

[2) Phân loại RBTV 5](#_Toc70228400)

[3) Xác định RBTV 6](#_Toc70228401)

[V. Xác định phụ thuộc hàm 10](#_Toc70228402)

[VI. Xây dựng mô hình thực thể liên kết 10](#_Toc70228403)

[1) Các khái niệm 10](#_Toc70228404)

[2) Xác định các quan hệ trong CSDL 11](#_Toc70228405)

[ CHƯƠNG II: CHUYỂN TỪ MÔ HÌNH THỰC THỂ LIÊN KẾT SANG MÔ HÌNH QUAN HỆ 12](#_Toc70228406)

[I. Cách chuyển đổi 12](#_Toc70228407)

[1) Quy tắc 1: Xử lý kiểu liên kết 1:1 12](#_Toc70228408)

[2) Quy tắc 2: Xử lý kiểu liên kết 1:n 12](#_Toc70228409)

[3) Quy tắc 3: Xử lý kiểu liên kết n:n 12](#_Toc70228410)

[4) Xử lý thuộc tính đa trị 12](#_Toc70228411)

[VII. Xác định các bảng và thuộc tính của bảng 13](#_Toc70228412)

[VIII. Mô hình dữ liệu quan hệ quản lý dự án: 13](#_Toc70228413)

[ CHƯƠNG III: XÁC ĐỊNH KHÓA 13](#_Toc70228414)

[I. Thuật toán xác định khóa của lược đồ quan hệ 14](#_Toc70228415)

[1) Định nghĩa về khóa 14](#_Toc70228416)

[2) Thuật toán xác định khóa của lược đồ quan hệ 14](#_Toc70228417)

[IX. Xác định khóa cho các quan hệ trong CSDL Quản lý dự án: 14](#_Toc70228418)

[ CHƯƠNG IV: CHUẨN HÓA LƯỢC ĐỒ QUAN HỆ THÀNH DẠNG CHUẨN 3NF HOẶC BCNF 15](#_Toc70228419)

[I. Các dạng chuẩn và thuật toán chuẩn hóa 15](#_Toc70228420)

[1) Các dạng chuẩn 15](#_Toc70228421)

[2) Thuật toán chuẩn hóa 15](#_Toc70228422)

[X. Chuẩn hóa cơ sở dữ liệu quan hệ quản lý dự án 16](#_Toc70228423)

[ CÂU LỆNH SQL 17](#_Toc70228424)

[I. Các câu lệnh tạo bảng 17](#_Toc70228425)

[ Đồ thị quan hệ: 20](#_Toc70228426)

CHƯƠNG I: THIẾT KẾ CƠ SỞ DỮ LIỆU VÀ MÔ HÌNH THỰC THỂ LIÊN KẾT

## Giới thiệu sơ lược về cơ sở dữ liệu quản lý “Website xem phim online”

Cơ sở dữ liệu (CSDL) “Website xem phim online” đáp ứng việc lưu trữ dữ liệu đầy đủ và chính xác. CSDL này lưu trữ thông tin phim, thông tin khách hàng, các bộ phim đang chiếu, các bình luận đánh giá của người dùng và đường dẫn của các bộ phim.

CSDL cho phép truy vấn thông tin liên quan đến những bộ phim.

Với mỗi phim chúng ta sẽ biết được lượt view, số lượng bình luận, phim đã được upload chưa, tình trạng của phim? Dựa theo thể loại, đạo diễn, diễn viên mà chúng ta có thể xác định được phim.

Các thực thể và các thuộc tính của chúng

Các đối tượng thực thể của CSDL “Website xem phim online” bao gồm:

1. **Phim** (**Id**, TenPhim, DaoDien\_Id, TheLoai NamPhatHanh, ThoiLuong, ChiTiet, BannerURL, NgayTao, LuotXem, TrangThai\_Id)
2. **BinhLuan** (**Phim\_Id**, **TaiKhoan\_Email**, **ThoiGian**, NoiDung)
3. **TaiKhoan** (**Email**, Password, Hoten, NgaySinh, AvatarURL)
4. **DanhSachPhat** (**TaiKhoan\_Email**, **Phim\_Id**, NgayTao, ChiTiet)
5. **TaiKhoan\_Role** (**TaiKhoan\_Email**, **Role\_Id**)
6. **Role** (**Id**, Quyen)
7. **NhanVat** (**Id**, TenNhanVat, ChiTiet, AvatarURL)
8. **DienVien\_Phim** (**DienVien\_Id**, **Phim\_Id**)
9. **TheLoai** (**Id**, TenTheLoai)
10. **TrangThai** (**Id**, TenTrangThai)
11. **TapPhim** (**Phim\_Id**, **Tap**, URL)

## Các khái niệm cơ bản

Để tiện cho việc quản lý phim, hệ thống lưu trữ thông tin phim bao gồm: mã phim, tên phim, mã đạo diễn của phim, năm phát hành, thời lượng tính theo phút, giới thiệu chi tiết phim, đường dẫn banner, đường dẫn , ngày upload phim lên website, lượt xem và mã tình trạng của phim.

Hệ thống lưu trữ những dự án và đưa 1 dự án được chỉ định tiến hành để cho 1 team đảm nhận 1 dự án, hệ thống sẽ tiếp nhận dự án qua dự án theo yêu cầu của khách hàng.

Chúng ta sẽ có 1 hệ thống để lưu trữ thông tin các khách hàng bao gồm: mã khách hàng, tên, công ty, địa chỉ và thông tin liên lạc của khách hàng.

Sẽ có 1 hệ thống để tương tác giữa khách hàng và dự án mà khách hàng đã yêu cầu. Mục đích của hệ thống này là để cho khách hàng đánh giá những dự án đã hoàn thành nhằm giúp cho chúng ta có thể đánh giá năng lực của 1 team và những nhân viên.

## Thiết kế ở mức khái niệm

Dưới đây là mô tả chi tiết các thuộc tính:

1. **Phim:** **Phim**:
   1. Id: đây là thuộc tính khóa để phân biệt các bộ phim với nhau.
   2. OrderID: mã dự án được đặt.
   3. TenPhim: tên của bộ phim.
   4. DaoDien\_Id: Id của đạo diễn.
   5. NamPhatHanh: năm phát hành của bộ phim.
   6. ThoiLuong: thời lượng của bộ phim.
   7. ChiTiet: chi tiết giới thiệu của bộ phim.
   8. BannerURL: đường dẫn của banner.
   9. URL: đường dẫn của .
   10. NgayTao: ngày upload phim lên website.
   11. LuotXem: Tổng số lượt xem của bộ phim.
   12. TrangThai\_Id: Id trạng thái của phim.
2. **Bình luận:** **BinhLuan**
3. ID: đây là khóa để phân biệt các bình luận với nhau.
4. Phim\_Id: Id của phim được bình luận.
5. TaiKhoan\_Email: Email của người bình luận xác định TaiKhoan.
6. ThoiGian: thời điểm bình luận này được gửi.
7. NoiDung: nội dung của bình luận.
8. **Người dùng: TaiKhoan** 
   1. Email: email của người dùng, đây là khoá để phân biệt các người dùng với nhau.
   2. Password: mật khẩu của người dùng.
   3. Hoten: họ tên của người dùng.
   4. NgaySinh: ngày tháng năm sinh của người dùng.
   5. AvatarURL: đường dẫn avatar của người dùng.
9. **Danh sách phát: DanhSachPhat**
   1. TaiKhoan\_Email: email của người tạo danh sách, là thuộc tính khoá để phân biệt các danh sách của mỗi người với nhau.
   2. Phim\_Id: Id của phim có trong danh sách, là thuộc tính khoá để phân biệt các phim có trong danh sách của mỗi người dùng.
   3. NgayTao: ngày tạo danh sách.
   4. ChiTiet: chi tiết, ghi chú của danh sách.
10. **Người dùng và quyền: TaiKhoan\_Role**
    1. TaiKhoan\_Email: email của người dùng.
    2. Role\_Id: Id của quyền.
11. **Quyền: Role**
    1. Id: Id của quyền là khoá để phân biệt các quyền.
    2. Quyen: quyền truy cập.
12. **Nhân vật: NhanVat**
    1. Id: đây là khoá để phân biệt các nhân vật với nhau.
    2. TenNhanVat: tên của nhân vật.
    3. ChiTiet: chi tiết, giới thiệu chung nhân vật.
    4. AvatarURL: đường dẫn ảnh đại diện của nhân vật.
13. **Diễn viên\_phim: DienVien\_Phim** 
    1. DienVien\_Id: Id để xác định diễn viên, là thuộc tính khoá để phân biệt các diễn viên trong từng bộ phim.
    2. Phim\_Id:Id của phim để xác định phim, là thuộc tính khoá để phân biệt các phim của từng diễn viên.
14. **Phim\_Thể loại: Phim\_TheLoai** 
    1. TheLoai\_Id: Id thể loại để xác định thể loại, là thuộc tính khoá để phân biệt các thể loại với nhau.
    2. Phim\_Id: Id phim để xác định phim, là thuộc tính khoá để phân biệt các phim với nhau.
15. **Thể loại: TheLoai**
    1. Id: là khoá để phân biệt các thể loại với nhau.
    2. TenTheLoai: tên của thể loại.
16. **Trạng thái:TrangThai**
    1. Id: là khoá để phân biệt các trạng thái với nhau.
    2. TenTrangThai: tên của trạng thái.
17. **Tập phim:TapPhim**
    1. Phim\_Id: Id phim xác định phim, là thuộc tính khoá để phân biệt các phim với nhau.
    2. Tap:
    3. URL: đường dẫn lưu trữ tập phim.

## Xác định RBTV

### Định nghĩa

Trong mỗi CSDL luôn tồn tại nhiều mối liên hệ giữa các thuộc tính, giữa các bộ; sự liên hệ này có thể xảy ra trong cùng một quan hệ hoặc trong các quan hệ của một lược đồ CSDL; các mối liên hệ này là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liên quan trong CSDL đều phải thỏa mãn ở mọi thời điểm. Những điều kiện bất biến đó được gọi là ràng buộc toàn vẹn (integrity constraint)- RBTV. Trong thực tế ràng buộc toàn vẹn là các quy tắc quản lý được áp đặt trên các đối tượng của thế giới thực; chẳng hạn các bộ phim phải có một mã phim khác nhau.

Ràng buộc toàn vẹn được xem như là một công cụ để diễn đạt ngữ nghĩa của CSDL. Công việc kiểm tra ràng buộc toàn vẹn thường được tiến hành vào các thời điểm cập nhật dữ liệu (thêm, sửa, xoá).

Ràng buộc toàn vẹn và kiểm tra sự vi phạm ràng buộc toàn vẹn là hai trong số những vấn đề quan trọng trong quá trình phân tích thiết kế cơ sở dữ liệu; nếu không quan tâm đúng mức đến những vấn đề trên; thì có thể dẫn đến những hậu quả nghiêm trọng về tính an toàn và toàn vẹn dữ liệu, đặc biệt là đối với những cơ sở dữ liệu lớn.

### Phân loại RBTV

Trong quá trình phân tích thiết kế CSDL, người phân tích phải phát hiện tất cả các ràng buộc toàn vẹn tiềm ẩn trong CSDL đó. Việc phân loại các ràng buộc toàn vẹn là rất có ích, nó nhằm giúp cho người phân tích có được một định hướng để phát hiện các ràng buộc toàn vẹn, tránh bỏ sót. Các ràng buộc toàn vẹn có thể được chia làm hai loại chính như sau: Thứ nhất: Ràng buộc toàn vẹn có phạm vi là một quan hệ bao gồm: Ràng buộc toàn vẹn miền giá trị, ràng buộc toàn vẹn liên thuộc tính, ràng buộc toàn vẹn liên bộ. Thứ hai: Ràng buộc toàn vẹn có phạm vi là nhiều quan hệ bao gồm: Ràng buộc toàn vẹn phụ thuộc tồn tại, ràng buộc toàn vẹn liên bộ - liên quan hệ, ràng buộc toàn vẹn liên thuộc tính - liên quan hệ.

#### Ràng buộc toàn vẹn có bối cảnh là một quan hệ

**Ràng buộc toàn vẹn miền giá trị:**

RBTV miền giá trị là ràng buộc quy định các giá trị cho một thuộc tính. Miền giá trị của một thuộc tính có thể liên tục hoặc rời rạc.

**Ràng buộc toàn vẹn liên bộ:**

RBTV liên bộ trên một quan hệ là ràng buộc giữa các bộ trong cùng một quan hệ, nghĩa là sự tồn tại của một hay nhiều bộ phụ thuộc vào sự tồn tại của một hay nhiều

bộ khác trong một quan hệ đó. Trường hợp đặc biệt của RBTV liên bộ trên một quan hệ là ràng buộc khóa chính.

**Ràng buộc toàn vẹn liên thuộc tính:**

Ràng buộc toàn vẹn liên thuộc tính trên một quan hệ là sự ràng buộc giữa các thuộc tính trong cùng một quan hệ.

#### Ràng buộc toàn vẹn trên nhiều quan hệ

**Ràng buộc toàn vẹn tham chiếu**

RBTV tham chiếu là ràng buộc trên giá trị các thuộc tính trong một quan hệ nào đó, các giá trị này phải xuất hiện (ở bộ nào đó) trên tập thuộc tính là khóa chính của một quan hệ khác. RBTV tham chiếu còn được gọi là ràng buộc khóa ngoại hay ràng buộc tồn tại. Thông thường bối cảnh của RBTV này là hai quan hệ, tuy nhiên cũng có trường hợp bối cảnh của RBTV loại này là một quan hệ.

**Ràng buộc toàn vẹn liên thuộc tính**

RBTV liên thuộc tính trên nhiều quan hệ là RBTV xảy ra trên nhiều thuộc tính của nhiều quan hệ khác nhau.

**Ràng buộc toàn vẹn liên bộ**

RBTV liên bộ trên nhiều quan hệ là RBTV giữa các bộ trên các quan hệ khác nhau, nghĩa là sự tồn tại của bộ dữ liệu trên quan hệ này quyết định sự tồn tại của bộ dữ liệu trên quan hệ khác.

**Ràng buộc toàn vẹn trên thuộc tính tổng hợp**

Thuộc tính tổng hợp là thuộc tính mà giá trị của nó được tính toán từ các thuộc tính khác. Khi lược đồ cơ sở dữ liệu có thuộc tính tổng hợp thì RBTV trên thuộc tính tổng hợp đảm bảo sự đúng đắn về mối quan hệ giữa thuộc tính tổng hợp và các thuộc tính nguồn.

### Xác định RBTV

Các RBTV trong cơ sở dữ liệu quản lý dự án gồm:

1. Với r là một quan hệ trên lược đồ quan hệ **Phim** ta có ràng buộc toàn vẹn (RBTV) sau:

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Sửa | Xoá |
| Phim | +(Id) | -(\*) | +(Id) |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **Phim**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
| NhanVat | +(ID) | +( ID) | - |
| Phim | - | - | +() |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **Phim**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
| TheLoai | +(ID) | +( ID) | - |
| Phim | - | - | +() |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **Phim**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
| TrangThai | +(ID) | +( ID) | - |
| Phim | - | - | +() |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **TapPhim**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
| Phim | +(ID) | +( ID) | - |
| TapPhim | - | - | +() |

1. Với r là một quan hệ trên lược đồ quan hệ **BinhLuan** ta có ràng buộc toàn vẹn (RBTV) sau:

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Sửa | Xoá |
| BinhLuan | +(Id) | -(\*) | +(Id) |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **BinhLuan**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
| Phim | +(ID) | +( ID) | - |
| BinhLuan | - | - | +() |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **BinhLuan**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
| TaiKhoan | +(Id) | +( Id) | - |
| BinhLuan | - | - | +() |

1. Với r là một quan hệ trên lược đồ quan hệ **TaiKhoan** ta có ràng buộc toàn vẹn (RBTV) sau:

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Sửa | Xoá |
| TaiKhoan | +(Email) | -(\*) | +(Email) |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **TaiKhoan\_Role**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
| TaiKhoan | +(ID) | +( ID) | - |
| TaiKhoan\_Role | - | - | +() |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **TaiKhoan\_Role**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
| Role | +(ID) | +( ID) | - |
| TaiKhoan\_Role | - | - | +() |

1. Với r là một quan hệ trên lược đồ quan hệ **Role**  ta có ràng buộc toàn vẹn (RBTV) sau:

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Sửa | Xoá |
| Role | +(Id) | -(\*) | +(Id) |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **DanhSachPhat**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
| TaiKhoan | +(ID) | +( ID) | - |
|  | - | - | +() |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **DanhSachPhat**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
|  | +(ID) | +( ID) | - |
|  | - | - | +() |

1. Với r là một quan hệ trên lược đồ quan hệ **TrangThai** ta có ràng buộc toàn vẹn (RBTV) sau:

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Sửa | Xoá |
| TrangThai | +(Id) | -(\*) | +(Id) |

1. Với r là một quan hệ trên lược đồ quan hệ **NhanVat** ta có ràng buộc toàn vẹn (RBTV) sau:

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Sửa | Xoá |
| NhanVat | +(Id) | -(\*) | +(Id) |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **DienVien\_Phim**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
|  | +(ID) | +( ID) | - |
|  | - | - | +() |

1. Với r là một quan hệ bất kỳ trên lược đồ quan hệ **DienVien\_Phim**, ta có RBTV

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xoá |
|  | +(ID) | +( ID) | - |
|  | - | - | +() |

1. Với r là một quan hệ trên lược đồ quan hệ **TheLoai** ta có ràng buộc toàn vẹn (RBTV) sau:

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Sửa | Xoá |
| TheLoai | +(Id) | -(\*) | +(Id) |

## Xác định phụ thuộc hàm

Cơ sở dữ liệu “Website xem phim online” có các phụ thuộc hàm như sau:

1. Id (Phim.Id) -> TenPhim, DaoDien\_Id (NhanVat.Id), TheLoai\_Id (TheLoai.Id), NamPhatHanh, ThoiLuong, ChiTiet, BannerURL, NgayTao, LuotXem, TrangThai\_Id.
2. Phim\_Id (Phim.Id), TaiKhoan\_Email (TaiKhoan.Email), ThoiGian -> NoiDung (BinhLuan.NoiDung).
3. Email (TaiKhoan.Email) -> Password, Hoten, NgaySinh, AvatarURL.
4. Id (Role.Id) -> Quyen.
5. Id (TheLoai.Id) -> TenTheLoai.
6. Id (NhanVat.Id) -> TenNhanVat, ChiTiet, AvatarURL.
7. Id (TrangThai.Id) -> TenTrangThai.
8. Phim\_Id (Phim.Id), Tap (TapPhim.Tap) -> URL (TapPhim.URL).

## Xây dựng mô hình thực thể liên kết

### Các khái niệm

Mô hình thực thể - liên kết xây dựng dựa trên những khái niệm:

* + - * Thuộc tính (Attribute)
      * Tập thực thể (Entity Set)
      * Mối liên kết (Relationship)

#### Thuộc tính (Attribute)

Thuộc tính là thông tin, dữ liệu đặc trưng của các đối tượng cần lưu trữ trong hệ thống thông tin (HTTT). Mỗi thuộc tính có các thành phần như:

* Tên: Mang ý nghĩa của thông tin cần lưu trữ. Tên thuộc tính phải phân biệt trên toàn mô hình.
* Kiểu dữ liệu
* Miền giá trị của thuộc tính: tập giá trị của thuộc tính được phép lưu trữ.

#### Thực thể (Entity)

Thực thể là một hình ảnh tương ứng với một lớp đối tượng có cùng tập thuộc tính cần lưu trữ trong HTTT.

Mỗi thực thể được xác định trên các thành phần:

* Tên gọi: thường là danh từ mang ý nghĩa của lớp đối tượng được mô hình hóa.
* Ý nghiã: Cho biết thực thể phản ảnh thông tin của lớp đối tượng nào. Các qui định liên quan đến dữ liệu cần lưu trữ
* Danh sách thuộc tính: Các thông tin đặc trương cần lưu trữ của lớp đối tượng.
* Khóa của thực thể: Khóa là tập thuộc tính dùng xác định duy nhất một đối tượng. Mỗi thực thể phải có ít nhất 1 khóa.

#### Mối liên kết (Relationship)

Dùng thể hiện sự quan hệ ngữ nghiã giữa các đối tượng ở các thực thể.

Các đặc trưng của mối kết hợp:

* Tên gọi: thường là động từ hay tính từ mang ý nghĩa về mối quan hệ giữa các lớp đối tượng liên quan trong tổ chức.
* Ý nghĩa: Dùng mô tả mối quan hệ ngữ nghĩa mà MKH được thể hiện.
* Bản số mỗi nhánh của mối kết hợp: là một cặp số tự nhiên (Min, Max) thể hiện ràng buộc về số lượng tối thiểu và tối đa của 1 đối tượng trên thực thể của nhánh có quan hệ với các đối tượng của các thực thể khác trong cùng mối liên kết.
* Giá trị Min = 0 nếu có thể có một đối tượng không tham gia vào bất kỳ trường hợp nào của mối liên kết.
* Giá trị Min = 1 nếu tất cả đối tượng đều tham gia vào mối liên kết.
* Giá trị Max = 1 nếu mỗi đối tượng chỉ tham gia tối đa 1 trường hợp của mối liên kết. Trong trường hợp này sẽ xuất hiện một phụ thuộc hàm giữa các khoá của các thực thể.
* Giá trị Max = n nếu mỗi đối tượng có thể tham gia nhiều trường hợp của mối kết hợp với số lượng không giới hạn.

### Xác định các quan hệ trong CSDL

1. Mối quan hệ: Có

Thực thể tham gia: Phim và BinhLuan

Một bộ phim có thể có nhiều nhiều bình luận nhưng mỗi bình luận chỉ thuộc 1 bộ phim.

1. Mối quan hệ: Có

Thực thể tham gia: Phim và NhanVat

Một bộ phim có một đạo diễn nhưng mỗi đạo diễn có thể có nhiều phim.

1. Mối quan hệ: Thuộc

Thực thể tham gia: Phim và TheLoai

Một bộ phim thuộc một thể loại nhưng mỗi thể loại có thể có nhiều phim.

1. Mối quan hệ: Có

Thực thể tham gia: Phim và TrangThai

Một bộ phim có một trạng thái nhưng mỗi trạng thái có thể có nhiều phim.

1. Mối quan hệ: Có

Thực thể tham gia: Phim và TrangThai

Một bộ phim có một trạng thái nhưng mỗi trạng thái có thể có nhiều phim.

1. Mối quan hệ: Có

Thực thể tham gia: TaiKhoan và BinhLuan

Một TaiKhoan có thể có nhiều bình luận nhưng mỗi bình luận chỉ có một phim.

1. Mối quan hệ: Có

Thực thể tham gia: TaiKhoan và Role

Một TaiKhoan có thể có nhiều Role và mỗi role có thể có nhiều TaiKhoan.

1. Mối quan hệ: Có

Thực thể tham gia: DanhSachPhat và Phim

Một danh sách phát có thể có nhiều phim và mỗi phim có thể thuộc nhiều danh sách.

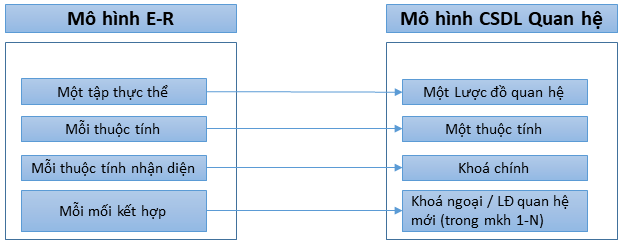
1. Mối quan hệ: Có

Thực thể tham gia: NhanVat và Phim

Mỗi nhân vật (diễn viên) có thể có nhiều bộ phim và mỗi bộ phim có thể có nhiều diễn viên.

# CHƯƠNG II: CHUYỂN TỪ MÔ HÌNH THỰC THỂ LIÊN KẾT SANG MÔ HÌNH QUAN HỆ

## Cách chuyển đổi



### Quy tắc 1: Xử lý kiểu liên kết 1:1

* Cách 1: Chuyển khóa chính của LĐQH này sang làm khóa ngoại của LĐQH khác hoặc ngược lại.
* Cách 2: Nhập 2 kiểu thực thể và mối liên kết thành một LĐQH, chọn khóa chính cho phù hợp.

### Quy tắc 2: Xử lý kiểu liên kết 1:n

Chuyển khóa chính của LĐQH bên 1 (cha) sang làm khóa ngoại của LĐQH bên nhiều (con).

### Quy tắc 3: Xử lý kiểu liên kết n:n

Chuyển mối liên kết thành một LĐQH có thuộc tính là thuộc tính của mối liên kết, thêm các thuộc tính khóa chính của các LĐQH có liên quan, khóa chính của LĐQH mới này là các thuộc tính mới thêm vào.

### Xử lý thuộc tính đa trị

Tách thuộc tính đa trị ra khỏi kiểu thực thể ban đầu, thêm một kiểu thực thể mới chứa thuộc tính đa trị này, xác định lại mối liên kết của kiểu thực thể này với kiểu thực thể ban đầu. Áp dụng các quy tắc từ 1 đến quy tắc 3.

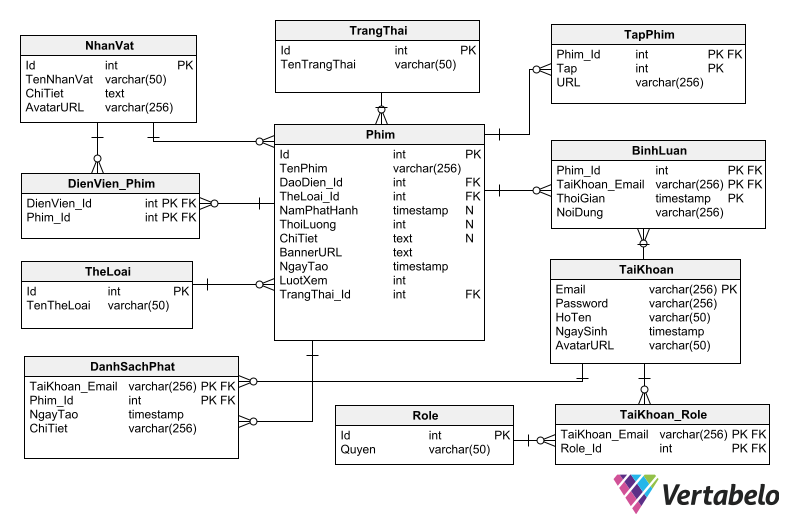
## Xác định các bảng và thuộc tính của bảng

Dựa theo cách chuyển đổi trên, ta được các bảng sau:

* 1. Phim (Id, TenPhim, DaoDien\_Id, TheLoai\_Id, NamPhatHanh, ThoiLuong, ChiTiet, BannerURL, NgayTao, LuotXem, TrangThai\_Id)
  2. BinhLuan (Phim\_Id, TaiKhoan\_Email, ThoiGian, NoiDung)
  3. TaiKhoan (Email, Password, Hoten, NgaySinh, AvatarURL)
  4. DanhSachPhat (TaiKhoan\_Email, Phim\_Id, NgayTao, ChiTiet)
  5. TaiKhoan\_Role (TaiKhoan\_Email, Role\_Id)
  6. Role (Id, Quyen)
  7. NhanVat (Id, TenNhanVat, ChiTiet, AvatarURL)
  8. DienVien\_Phim (DienVien\_Id, Phim\_Id)
  9. TheLoai (Id, TenTheLoai)
  10. TrangThai (Id, TenTrangThai)
  11. TapPhim (Phim\_Id, Tap, URL)

## Mô hình dữ liệu quan hệ quản lý dự án:

Đây là mô hình dữ liệu quan hệ của CSDL Quản lý dự án:



# CHƯƠNG III: XÁC ĐỊNH KHÓA

## Thuật toán xác định khóa của lược đồ quan hệ

### Định nghĩa về khóa

Cho lược đồ quan hệ R, S⊆R+. S được gọi là một siêu khóa (superkey) của lược đồ quan hệ R nếu với hai bộ tùy ý trong quan hệ R thì giá trị của các thuộc tính trong S là khác nhau.

Một lược đồ quan hệ có thể có nhiều siêu khoá. Siêu khoá chứa ít thuộc tính nhất được gọi là khóa chỉ định, trong trường hợp lược đồ quan hệ có nhiều khóa chỉ định, thì khóa được chọn để cài đặt gọi là khóa chính (Primary key) (trong các phần sau khóa chính được gọi tắt là khóa).

Các thuộc tính tham gia vào một khóa được gọi là thuộc tính khóa (prime key), ngược lại được gọi là thuộc tính không khóa (nonprime key).

Một thuộc tính được gọi là khóa ngoại nếu nó là thuộc tính của một lược đồ quan hệ này nhưng lại là khóa chính của lược đồ quan hệ khác.

### Thuật toán xác định khóa của lược đồ quan hệ

**Bước 1**: Tạo tập nguồn TN và tập trung gian TG

**Bước 2**: Nếu TG=0 (rỗng) thì K=TN, kết thúc, ngược lại qua bước 3

**Bước 3**: Tìm tất cả tập con Xi của tập trung gian.

**Bước 4**: Tìm siêu khóa Si bằng cách với mọi Xi,

nếu (TN U Xi)+ = Q+ thì Si = TN U Xi

**Bước 5**: Tìm khóa bằng cách loại bỏ các siêu khóa không tối thiểu - với mọi Si, Sj thuộc S nếu Si chứa trong Sj thì loại bỏ tập Sj ra khỏi siêu khóa (VD: Si=AB, Sj=ABC thì loại bỏ Sj ra khỏi tập siêu khóa) S còn lại chính là tập khóa cần tìm.

## Xác định khóa cho các quan hệ trong CSDL Quản lý dự án:

Cho LĐQH R.

R = { Phim.Id, TenPhim, DaoDien\_Id, TheLoai\_Id, NamPhatHanh, ThoiLuong, ChiTiet, BannerURL, NgayTao, LuotXem, TrangThai\_Id, ThoiGian, NoiDung, Email, Password, Hoten, NgaySinh, AvatarURL, Role.Id, Quyen, DanhSachPhat.NgayTao, DanhSachPhat.ChiTiet, TheLoai. Id, TheLoai. TenTheLoai, TenNhanVat, ChiTiet, AvatarURL, TrangThai. TenTrangThai, TapPhim.Phim\_Id, TapPhim.Tap, TapPhim.URL }

Tập phụ thuộc hàm F = {

f1:Phim.ID -> TenPhim, DaoDien\_Id, TheLoai\_Id, NamPhatHanh, ThoiLuong, ChiTiet, BannerURL, NgayTao, LuotXem, TrangThai\_Id;

f2: Phim\_Id, TaiKhoan\_Email -> ThoiGian, NoiDung;

f3: Email -> Password, Hoten, NgaySinh, AvatarURL;

f4: TaiKhoan\_Email, Phim\_Id -> NgayTao, ChiTiet;

f5: Role.Id-> Quyen;

f6: NhanVat.Id -> TenNhanVat, ChiTiet, AvatarURL;

f7: TheLoai.Id -> TenTheLoai;

f8: TrangThai.Id, TenTrangThai;

f9: Phim\_Id, Tap -> URL;}

# CHƯƠNG IV: CHUẨN HÓA LƯỢC ĐỒ QUAN HỆ THÀNH DẠNG CHUẨN 3NF HOẶC BCNF

Mục đích của chuẩn hoá:

* Loại bỏ dư thừa dữ liệu
* Loại bỏ dữ liệu dị thường xảy ra khi cập nhật CSDL (update anomaly), thêm dữ liệu (insertion anomaly) và khi xoá dữ liệu (deletion anomaly).

## Các dạng chuẩn và thuật toán chuẩn hóa

### Các dạng chuẩn

#### Dạng chuẩn 1NF

Lược đồ quan hệ R được gọi là đạt dạng chuẩn 1 (1NF) nếu và chỉ nếu toàn bộ các thuộc tính của R đều mang giá trị đơn.

#### Dạng chuẩn 2NF

Một lược đồ quan hệ R đạt dạng chuẩn 2 nếu R đạt dạng chuẩn 1 và tất cả các thuộc tính không khoá của R đều phụ thuộc đầy đủ vào khoá.

#### Dạng chuẩn 3NF

Một lược đồ quan hệ R đạt dạng chuẩn 3 nếu với mọi phụ thuộc hàm thì một trong 2 điều kiện sau được thoả: hoặc X là một siêu khoá của R hoặc A là một thuộc tính khoá của R.

#### Dạng chuẩn BC

Một lược đồ quan hệ R ở dạng chuẩn BC nếu với mỗi phụ thuộc hàm thì X là một siêu khoá của R.

### Thuật toán chuẩn hóa

Một số phương pháp phân rã để đưa một lược đồ quan hệ từ dạng chuẩn thấp về một lược đồ cơ sở dữ liệu đạt dạng chuẩn cao hơn.

#### Phân rã bảo toàn thông tin

Phân rã lược đồ R (A1, A2,…, An) là thay nó bằng một lược đồ cơ sở dữ liệu p = {R1, R2,…, Rk} sao cho

p: là một phần rã của R.

Thuật toán:

**Bước 1**: Thiết lập bảng với k+1 dòng, n+1 cột. Cột j ứng với mỗi thuộc tính Aj (j=1…n), hàng I ứng với lược đồ quan hệ Ri (i=1…k). Tại vị trí hàng I, cột j ta điền ký hiệu Aj nếu , nếu không ta đặt ký hiệu bt vào vị trí đó với t đầu tiên bằng 1 và tăng t một đơn vị.

**Bước 2**: Xét lần lượt các phụ thuộc hàm trong F, áp dụng cho bảng vừa mới thành lập ở trên.

Giả sử xét , chúng ta tìm những hàng giống nhau ở tất cả các thuộc tính của X, nếu thấy những hàng như vậy ta sẽ làm cho các ký hiệu của hai hàng này bằng nhau ở tất cả các thuộc tính của Y.

Khi làm cho 2 ký hiệu này bằng nhau, nếu một trong hai ký hiệu là aj thì cho ký hiệu kia thành aj, nếu hai ký hiệu là bk hoặc bl thì có thể cho chúng trở thành bk hoặc bt (với t = min(k, l)).

Bước này được tiếp tục cho các phụ thuộc hàm còn lại của F cho đến khi không còn áp dụng được nữa.

**Bước 3**: Xét bảng kết quả, nếu thấy trong bảng này có một hàng chứa toàn aj (j=1…n) thì kết luận đó là phép kết nối bảo toàn thông tin. Ngược lại là phép kết nối mất mát thông tin.

#### Phân rã bảo toàn phụ thuộc hàm

Cho lược đồ quan hệ R và tập phụ thuộc hàm F. Phân rã của R là p = {R1, R2,…, Rn}.

Mỗi Ri, Fi (chiếu của F lên Ri) p= {<R1, F1>, <R2, F2>,…,<Rn, Fn>}.

Xét :

* Nếu tương đương: phân rã bảo toàn phụ thuộc hàm
* Nếu không tương đương: phân rã không bảo toàn phụ thuộc hàm

#### Phân rã thành dạng chuẩn 3NF vừa bảo toàn thông tin vừa bảo toàn phụ thuộc hàm

Thuật toán:

* Tìm phủ tối thiểu Ftt của F
* Nếu có một phụ thuộc hàm nào của Ftt mà liên quan đến tất cả các thuộc tính của Q thì kết quả phân rã chính là R (R không thể phân rã).
* Nếu có những thuộc tính của R không nằm trong một phụ thuộc nào của Ftt – dù ở vế phải hay vế trái của F thì chúng tạo thành một lược đố cần tìm.
* Cứ mỗi phụ thuộc hàm thì XA là một lược đồ cần tìm
* Nếu có một lược đồ con chứa khoá K của R thì kết thúc thuật toán.
* Ngược lại tạo một lược đồ con K.

## Chuẩn hóa cơ sở dữ liệu quan hệ quản lý dự án

Sau khi chuẩn hoá, ta được CSDL “Website xem phim online” gồm các bảng và thuộc tính như sau:

* Bảng Phim: khóa là Id.

Phim (Id, TenPhim, DaoDien\_Id, TheLoai\_Id, NamPhatHanh, ThoiLuong, ChiTiet, BannerURL, NgayTao, LuotXem, TrangThai\_Id)

* Bảng BinhLuan: khóa là Phim\_Id, TaiKhoan\_Email, ThoiGian.

BinhLuan (Phim\_Id, TaiKhoan\_Email, ThoiGian, NoiDung)

* Bảng TaiKhoan: khóa là Email.

TaiKhoan (Email, Password, Hoten, NgaySinh, AvatarURL)

* Bảng DanhSachPhat: khóa là TaiKhoan\_Email, Phim\_Id.

DanhSachPhat (TaiKhoan\_Email, Phim\_Id, NgayTao, ChiTiet)

* Bảng TaiKhoan\_Role: khóa là TaiKhoan\_Email, Role\_Id.

TaiKhoan\_Role (TaiKhoan\_Email, Role\_Id)

* Bảng Role: khóa là Id.

Role (Id, Quyen)

* Bảng NhanVat: khóa là Id.

NhanVat (Id, TenNhanVat, ChiTiet, AvatarURL)

* Bảng DienVien\_Phim: khóa là DienVien\_Id, Phim\_Id.

DienVien\_Phim (DienVien\_Id, Phim\_Id)

* Bảng TheLoai: khóa là Id.

TheLoai (Id, TenTheLoai)

* Bảng TrangThai: khóa là Id.

TrangThai (Id, TenTrangThai)

* Bảng TapPhim: khóa là Phim\_Id, Tap.

TapPhim (Phim\_Id, Tap, URL)

# CÂU LỆNH SQL

## Các câu lệnh tạo bảng

* + - 1. Phim.

-- Table: Phim

CREATE TABLE Phim (

Id int NOT NULL,

TenPhim varchar(256) NOT NULL,

DaoDien\_Id int NOT NULL,

TheLoai\_Id int NOT NULL,

NamPhatHanh int NULL,

ThoiLuong int NULL,

ChiTiet text NULL,

BannerURL text NOT NULL,

NgayTao timestamp NOT NULL,

LuotXem int NOT NULL,

TrangThai\_Id int NOT NULL,

CONSTRAINT Phim\_pk PRIMARY KEY (Id)

##### );

* + - 1. BinhLuan.

-- Table: BinhLuan

CREATE TABLE BinhLuan (

Phim\_Id int NOT NULL,

TaiKhoan\_Email varchar(256) NOT NULL,

ThoiGian timestamp NOT NULL,

NoiDung varchar(256) NOT NULL,

CONSTRAINT BinhLuan\_pk PRIMARY KEY (ThoiGian,Phim\_Id,TaiKhoan\_Email)

##### );

* + - 1. TaiKhoan.

-- Table: TaiKhoan

CREATE TABLE TaiKhoan (

Email varchar(256) NOT NULL,

Password varchar(256) NOT NULL,

HoTen varchar(50) NOT NULL,

NgaySinh timestamp NOT NULL,

AvatarURL varchar(50) NOT NULL,

CONSTRAINT TaiKhoan\_pk PRIMARY KEY (Email)

);

* + - 1. DanhSachPhat.

-- Table: DanhSachPhat

CREATE TABLE DanhSachPhat (

TaiKhoan\_Email varchar(256) NOT NULL,

Phim\_Id int NOT NULL,

NgayTao timestamp NOT NULL,

ChiTiet varchar(256) NOT NULL,

CONSTRAINT DanhSachPhat\_pk PRIMARY KEY (TaiKhoan\_Email,Phim\_Id)

##### );

* + - 1. TaiKhoan\_Role.

-- Table: TaiKhoan\_Role

CREATE TABLE TaiKhoan\_Role (

TaiKhoan\_Email varchar(256) NOT NULL,

Role\_Id int NOT NULL,

CONSTRAINT TaiKhoan\_Role\_pk PRIMARY KEY (TaiKhoan\_Email,Role\_Id)

##### );

* + - 1. Role.

-- Table: Role

CREATE TABLE Role (

Id int NOT NULL,

Quyen varchar(50) NOT NULL,

CONSTRAINT Role\_pk PRIMARY KEY (Id)

);

* + - 1. NhanVat.

-- Table: NhanVat

CREATE TABLE NhanVat (

Id int NOT NULL,

TenNhanVat varchar(50) NOT NULL,

ChiTiet text NOT NULL,

AvatarURL varchar(256) NOT NULL,

CONSTRAINT NhanVat\_pk PRIMARY KEY (Id)

##### );

* + - 1. DienVienPhim.

-- Table: DienVien\_Phim

CREATE TABLE DienVien\_Phim (

DienVien\_Id int NOT NULL,

Phim\_Id int NOT NULL,

CONSTRAINT DienVien\_Phim\_pk PRIMARY KEY (DienVien\_Id,Phim\_Id)

##### );

* + - 1. TheLoai.

-- Table: TheLoai

CREATE TABLE TheLoai (

Id int NOT NULL,

TenTheLoai varchar(50) NOT NULL,

CONSTRAINT TheLoai\_pk PRIMARY KEY (Id)

);

* + - 1. TrangThai.

-- Table: TrangThai

CREATE TABLE TrangThai (

Id int NOT NULL,

TenTrangThai varchar(50) NOT NULL,

CONSTRAINT TrangThai\_pk PRIMARY KEY (Id)

);

* + - 1. TapPhim

-- Table: TapPhim

CREATE TABLE TapPhim (

Phim\_Id int NOT NULL,

Tap int NOT NULL,

URL varchar(256) NOT NULL,

CONSTRAINT TapPhim\_pk PRIMARY KEY (Phim\_Id,Tap)

);

-- foreign keys

-- Reference: BinhLuan\_Phim (table: BinhLuan)

ALTER TABLE BinhLuan ADD CONSTRAINT BinhLuan\_Phim FOREIGN KEY BinhLuan\_Phim (Phim\_Id)

REFERENCES Phim (Id);

-- Reference: BinhLuan\_User (table: BinhLuan)

ALTER TABLE BinhLuan ADD CONSTRAINT BinhLuan\_User FOREIGN KEY BinhLuan\_User (TaiKhoan\_Email)

REFERENCES TaiKhoan (Email);

-- Reference: DanhSachPhat\_Phim (table: DanhSachPhat)

ALTER TABLE DanhSachPhat ADD CONSTRAINT DanhSachPhat\_Phim FOREIGN KEY DanhSachPhat\_Phim (Phim\_Id)

REFERENCES Phim (Id);

-- Reference: DanhSachPhat\_User (table: DanhSachPhat)

ALTER TABLE DanhSachPhat ADD CONSTRAINT DanhSachPhat\_User FOREIGN KEY DanhSachPhat\_User (TaiKhoan\_Email)

REFERENCES TaiKhoan (Email);

-- Reference: DienVien\_Phim\_DienVien (table: DienVien\_Phim)

ALTER TABLE DienVien\_Phim ADD CONSTRAINT DienVien\_Phim\_DienVien FOREIGN KEY DienVien\_Phim\_DienVien (DienVien\_Id)

REFERENCES NhanVat (Id);

-- Reference: DienVien\_Phim\_Phim (table: DienVien\_Phim)

ALTER TABLE DienVien\_Phim ADD CONSTRAINT DienVien\_Phim\_Phim FOREIGN KEY DienVien\_Phim\_Phim (Phim\_Id)

REFERENCES Phim (Id);

-- Reference: Phim\_NhanVat (table: Phim)

ALTER TABLE Phim ADD CONSTRAINT Phim\_NhanVat FOREIGN KEY Phim\_NhanVat (DaoDien\_Id)

REFERENCES NhanVat (Id);

-- Reference: Phim\_Status (table: Phim)

ALTER TABLE Phim ADD CONSTRAINT Phim\_Status FOREIGN KEY Phim\_Status (TrangThai\_Id)

REFERENCES TrangThai (Id);

-- Reference: Phim\_TheLoai (table: Phim)

ALTER TABLE Phim ADD CONSTRAINT Phim\_TheLoai FOREIGN KEY Phim\_TheLoai (TheLoai\_Id)

REFERENCES TheLoai (Id);

-- Reference: Quyen\_User\_User (table: TaiKhoan\_Role)

ALTER TABLE TaiKhoan\_Role ADD CONSTRAINT Quyen\_User\_User FOREIGN KEY Quyen\_User\_User (TaiKhoan\_Email)

REFERENCES TaiKhoan (Email);

-- Reference: TapPhim\_Phim (table: TapPhim)

ALTER TABLE TapPhim ADD CONSTRAINT TapPhim\_Phim FOREIGN KEY TapPhim\_Phim (Phim\_Id)

REFERENCES Phim (Id);

-- Reference: User\_Role\_Role (table: TaiKhoan\_Role)

ALTER TABLE TaiKhoan\_Role ADD CONSTRAINT User\_Role\_Role FOREIGN KEY User\_Role\_Role (Role\_Id)

REFERENCES Role (Id);

# Đồ thị quan hệ:

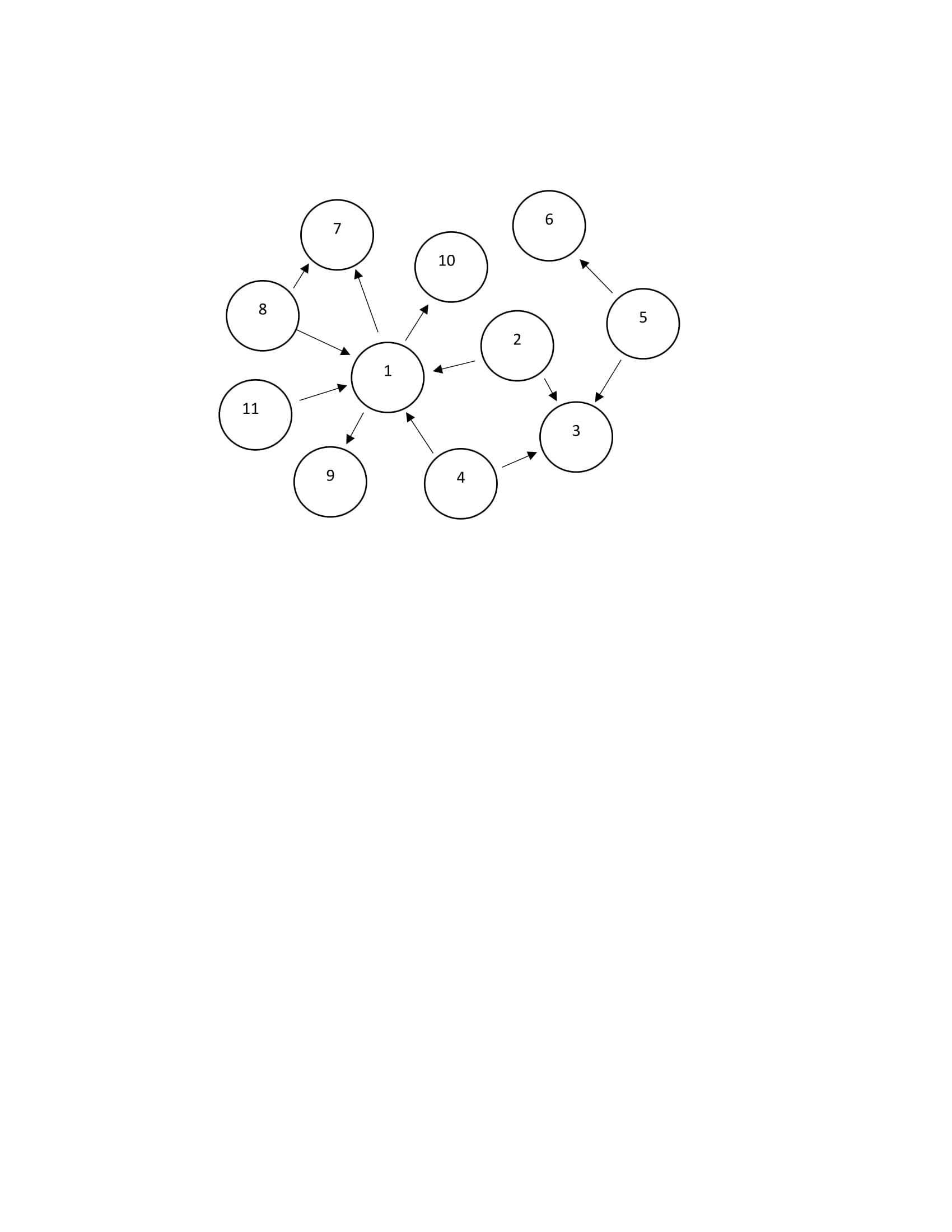
**Đầu tiên chúng ta sẽ tạo nút:**

1. **Phim** (**Id**, TenPhim, DaoDien\_Id, TheLoai\_Id, NamPhatHanh, ThoiLuong, ChiTiet, BannerURL, NgayTao, LuotXem, TrangThai\_Id)
2. **BinhLuan** (**Phim\_Id**, **TaiKhoan\_Email**, **ThoiGian**, NoiDung)
3. **TaiKhoan** (**Email**, Password, Hoten, NgaySinh, AvatarURL)
4. **DanhSachPhat** (**TaiKhoan\_Email**, **Phim\_Id**, NgayTao, ChiTiet)
5. **TaiKhoan\_Role** (**TaiKhoan\_Email**, **Role\_Id**)
6. **Role** (**Id**, Quyen)
7. **NhanVat** (**Id**, TenNhanVat, ChiTiet, AvatarURL)
8. **DienVien\_Phim** (**DienVien\_Id**, **Phim\_Id**)
9. **TheLoai** (**Id**, TenTheLoai)
10. **TrangThai** (**Id**, TenTrangThai)
11. **TapPhim** (**Phim\_Id**, **Tap**, URL)

**Sau đó chúng ta bắt đầu tạo cung:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | PTH | PTH\_Thừa | LK | LK\_Thừa | Cung |
| 1. Phim | 7,9,10 | Ø | Ø | Ø | 7,9,10 |
| 2. BinhLuan | 1,5 | Ø | Ø | Ø | 1,5 |
| 3. TaiKhoan | - | - | - | - | - |
| 4. DanhSachPhat | 1,3 | Ø | Ø | Ø | 1,3 |
| 5. TaiKhoan\_Role | 3,6 | Ø | Ø | Ø | 3,6 |
| 6. Role | - | - | - | - | - |
| 7. NhanVat | 1,5 | Ø | Ø | Ø | 1,5 |
| 8. DienVien\_Phim | 1,7 | Ø | Ø | Ø | 1,7 |
| 9. TheLoai | - | - | - | - | - |
| 10. TrangThai | - | - | - | - | - |
| 11.TapPhim | 1 | Ø | Ø | Ø | 1 |

Mô hình đồ thị quan hệ theo các bảng dữ liệu:



**- Ý nghĩa của đồ thị quan hệ:**

1. 1->7,9,10 Mỗi bộ Phim sẽ xác định được một thể loại, một đạo diễn và một trạng thái. Thống kê được đạo diễn nào làm nhiều phim nhất. Thể loại nào có nhiều phim nhất.
2. 4->1,3,9 Mỗi DanhSachPhat sẽ xác định được một bộ phim, một thể loại của phim đó và một TaiKhoan của DanhSachPhat.
3. 5->6,3 Mỗi TaiKhoan\_Role sẽ xác định được một TaiKhoan và một Role.
4. 8->7,1,9 Mỗi DienVien\_Phim xác định một NhanVat, một bộ phim mà nhân vật đó đóng và thể loại của phim đó. Thống kê được diễn viên nào đóng nhiều bộ phim nhất.
5. 2->3,1,7 Mỗi BinhLuan sẽ xác định một TaiKhoan, một bộ phim và đạo diễn của phim đó. Thống kê được tài khoản nào hoạt động bình luận nhiều nhất.