

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN: CƠ SỞ DỮ LIỆU



BÁO CÁO BÀI TẬP LỚN

ĐỀ TÀI: HỆ THỐNG BÁN VÉ XEM PHIM ONLINE

Nhóm sinh viên thực hiện : Nhóm 13

Lớp : CSDL 08

Giảng viên hướng dẫn : ThS. Nguyễn Quỳnh Chi

Hà Nội - 2025

PHÂN CÔNG NHIỆM VỤ NHÓM THỰC HIỆN

TT	SV thực hiện	MSV	Công việc / Nhiệm vụ
1	Nguyễn Hữu Huy	B23DCCN395	Hoàn thiện mô tả chi tiết, ERD
2	Lê Quốc Tuấn	B23DCCN885	
3	Nguyễn Đức Anh	B23DCCN031	Hoàn thiện bảng, giao diện
4	Nguyễn Thành Nam	B23DCCN587	
5	Trần Hoàng Nam	B23DCCN591	Kiểm tra tiến độ, hoàn thiện sản phẩm (Nhóm trưởng)

Mục Lục

Phần 1. Xác định yêu cầu về dữ liệu & chức năng.....5

1.1 Mô tả chung về hệ thống.....	5
1.2 Yêu cầu về dữ liệu của hệ thống	6
1.3 Một số chức năng chính	7
a, Khách hàng.....	7
b, Nhân viên	7
c, Ban quản lý.....	9

Phần 2. Phân tích yêu cầu dữ liệu – Vẽ ERD mở rộng11

2.1. Liệt kê các thực thể	11
1. Người dùng.....	11
2. Lịch cụ thể.....	11
3. Ca làm việc.....	11
4. Vé xem phim	12
5. Phòng chiếu.....	12
6. Phim	12
7. Thẻ loại.....	12
8. Suất chiếu	12
9. Ghế ngồi	13
10. Hóa đơn	13
11. Phương thức thanh toán	13
2.2 Ràng buộc giữa các thực thể	13
1.Cụm đặt vé	13
2.Cụm phim.....	13
3.Cụm vé	14
4.Cụm rạp	14
2.3 Vẽ mô hình ER mở rộng	16

Phần 3. Thiết kế mô hình quan hệ17

1. Bảng NguoiDung	17
2. Bảng CaLamViec	18
3. Bảng LichCuThe	19
4. Bảng PhongChieu.....	20
5. Bảng Phim.....	20
6. Bảng SuatChieu.....	21
7. Bảng PhuongThucThanhToan	22

8. Bảng HoaDon.....	23
9. Bảng Ghe.....	23
10.Bảng VeXemPhim	24
11.Bảng TheLoai.....	25
12.Bảng TheLoaiPhim	26
Phần 4. Kiểm tra chuẩn hóa các bảng về chuẩn 3.....	27
Phần 5. Demo 2 biểu mẫu nhập liệu & SQL tương ứng.....	28
5.1 Biểu mẫu quản lý Phim	28
5.2 Biểu mẫu quản lý Suất chiếu.....	33
Phần 6. Demo 2 biểu mẫu báo cáo + SQL.....	37
6.1 Biểu mẫu báo cáo doanh thu	37
6.2 Biểu mẫu báo cáo hóa đơn vé	41

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc tới cô Chi, giảng viên môn Cơ sở dữ liệu của Học viện Công nghệ Bưu chính Viễn thông, người đã tận tình truyền đạt cho chúng em những kiến thức nền tảng và quan trọng về mô hình dữ liệu, phân tích thiết kế hệ thống, chuẩn hóa dữ liệu, cũng như các kỹ thuật xây dựng và quản trị cơ sở dữ liệu hiện đại. Những kiến thức mà cô cung cấp không chỉ giúp chúng em hiểu rõ về lý thuyết mà còn tạo nền tảng vững chắc để nhóm em có thể hoàn thành bài tập lớn này một cách hiệu quả và khoa học.

Thông qua môn học, chúng em đã có cơ hội tiếp cận và vận dụng các khái niệm cốt lõi như mô hình thực thể liên kết, chuyển đổi mô hình quan hệ, chuẩn hóa dữ liệu, cũng như thao tác với các câu lệnh SQL trong thực tế. Việc xây dựng hệ thống quản lý rạp chiếu phim online đã giúp chúng em rèn luyện khả năng phân tích yêu cầu, thiết kế dữ liệu và triển khai các chức năng quản trị cơ bản, từ đó củng cố và mở rộng khối kiến thức đã được học.

Trong quá trình thực hiện bài tập lớn, mặc dù đã nỗ lực hoàn thiện một cách chi tiết và đầy đủ nhất, nhưng do hạn chế về kiến thức và kinh nghiệm thực tế, bài báo cáo khó tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý của cô để có thể cải thiện kỹ năng và hoàn thiện hơn trong tương lai.

Cuối cùng, chúng em xin kính chúc cô thật nhiều sức khỏe, hạnh phúc và thành công trong sự nghiệp giáo dục.

Chúng em xin trân trọng cảm ơn!

Phần 1. Xác định yêu cầu về dữ liệu & chức năng

1.1 Mô tả chung về hệ thống

Hệ thống bán vé xem phim online được xây dựng với mục tiêu hỗ trợ quá trình vận hành rạp chiếu phim một cách tự động, hiệu quả và hiện đại. Hệ thống cho phép người dùng tra cứu thông tin phim, lựa chọn suất chiếu, đặt vé, thanh toán và theo dõi giao dịch trực tuyến. Đồng thời, hệ thống cũng cung cấp các công cụ quản trị dành cho nhân viên và ban quản lý nhằm đảm bảo quy trình hoạt động của rạp được đồng bộ, chính xác và tiện lợi.

Hệ thống phục vụ ba nhóm người dùng chính: Khách hàng, nhân viên rạp, và ban quản lý.

- **Khách hàng** sử dụng để tra cứu phim, lịch chiếu, chọn ghế, đặt và thanh toán vé trực tuyến, đồng thời quản lý tài khoản cá nhân và lịch sử giao dịch.
- **Nhân viên** đảm nhiệm việc chăm sóc khách hàng online. Họ giúp giải đáp thắc mắc, tiếp nhận phản hồi và theo dõi hoạt động hệ thống để đảm bảo trải nghiệm người dùng được ổn định. Tiếp nhận thông tin hóa đơn và vé, phục vụ khách hàng tại quầy
- **Ban quản lý** khai thác hệ thống để quản lý danh mục phim, lịch chiếu, giá vé, đồng thời quản lý thông tin nhân viên, xuất báo cáo doanh thu để xem tiến trình hoạt động của rạp.

Hệ thống giúp tối ưu hóa quy trình bán vé, quản lý rạp hiệu quả và nâng cao trải nghiệm khách hàng.

1.2 Yêu cầu về dữ liệu của hệ thống

Hệ thống cần lưu và quản lý các nhóm dữ liệu sau:

- Nhóm quản lý người dùng:

- **Bao gồm:** NguoiDung (Khách hàng, Nhân viên, Quản lý).

- **Dữ liệu:** Thông tin cá nhân, tài khoản, phân quyền, người quản lý trực tiếp.

- Nhóm quản lý rạp và phim:

- **Bao gồm:** Phim, TheLoai, PhongChieu, Ghe, SuatChieu.

- **Dữ liệu:** Thông tin phim, lịch chiếu, suất chiếu, thể loại phim, cấu trúc phòng chiếu và sơ đồ ghế.

- Nhóm giao dịch và vé:

- **Bao gồm:** HoaDon, VeXemPhim, PhuongThucThanhToan.

- **Dữ liệu:** Trạng thái thanh toán, chi tiết vé đặt (ghế nào, suất nào), lịch sử mua hàng.

- Nhóm quản lý nhân sự:

- **Bao gồm:** CaLamViec, LichCuThe.

- **Dữ liệu:** Ca làm việc cho nhân viên, chấm công (lịch làm việc cụ thể).

1.3 Một số chức năng chính

a, Khách hàng

1. Đăng ký tài khoản

- **Bảng liên quan:** NguoiDung
- **Thao tác:**
 - Thêm bản ghi mới vào bảng NguoiDung khi người dùng hoàn tất biểu mẫu đăng ký.
 - Vai trò lưu mặc định là khách hàng
 - Kiểm tra trùng lặp tài khoản, email, số điện thoại trước khi lưu.
- **Dữ liệu ghi:** tài khoản, mật khẩu, họ tên, ngày sinh, SĐT, email, vai trò (tạo tự động).

2. Đăng nhập / Đăng xuất

- **Bảng liên quan:** NguoiDung
- **Thao tác:**
 - Đối chiếu tài khoản và mật khẩu với bản ghi trong NguoiDung với vai trò là khách hàng.

3. Cập nhật thông tin cá nhân

- **Bảng liên quan:** NguoiDung
- **Thao tác:**
 - Cập nhật họ tên, SĐT, email, mật khẩu trong bản ghi tương ứng với ID khách hàng.
 - Có kiểm tra định dạng và trùng lặp dữ liệu.

4. Tra cứu phim, suất chiếu

- **Bảng liên quan:** Phim, TheLoai, TheLoaiPhim, SuatChieu

- **Thao tác:**

- Truy vấn dữ liệu để hiển thị danh sách phim, suất chiếu
- Tìm kiếm phim theo tên phim, thể loại, độ dài phim, ngôn ngữ chính

5. Đặt vé xem phim

- **Bảng liên quan:**

- Phim
- PhongChieu
- SuatChieu
- Ghe
- HoaDon
- VeXemPhim
- TheLoai, TheLoaiPhim

- **Thao tác:**

- Khách hàng chọn phim
- Tạo hóa đơn rồng lưu thông tin khách hàng
- Khách hàng chọn suất chiếu, chọn ghế trống
- Tạo vé bao gồm các thông tin (Tên phim, phòng chiếu, suất chiếu, ghế ngồi, giá vé, trạng thái vé)
- Hóa đơn cập nhật theo số lượng vé khách đặt để đưa ra tổng tiền và cho phép khách hàng thực hiện thanh toán.
- Nếu thanh toán thất bại hoặc không thanh toán, vé và hóa đơn bị hủy bỏ

6. Xem lịch sử giao dịch

- **Bảng liên quan:** HoaDon, Ve, SuatChieu, Phim, PhongChieu, Ghe

- **Thao tác:**

- Truy vấn dữ liệu để xem chi tiết hóa đơn

- Truy vấn dữ liệu liên kết các bảng để hiển thị danh sách chi tiết vé đã mua, phim, thời điểm chiếu phim, phòng, ghế, giá vé, trạng thái của vé thông qua hóa đơn

b, Nhân viên

1. Đăng nhập / Đăng xuất

- **Bảng liên quan:** NguoiDung
- **Thao tác:**
 - Kiểm tra tài khoản và mật khẩu trong bảng NguoiDung với vai trò nhân viên

2. Tìm kiếm thông tin khách hàng và xuất hóa đơn, in vé

- **Bảng liên quan:** NguoiDung, Ve, HoaDon, PhongChieu, SuatChieu, Phim, Ghe
- **Thao tác:**
 - Truy vấn dữ liệu từ bảng KhachHang để tìm kiếm theo tên, số điện thoại hoặc email nhằm hỗ trợ khách hàng
 - Truy vấn dữ liệu từ bảng HoaDon và Ve để xem thông tin hóa đơn và vé của khách hàng. Xuất hóa đơn và in vé

c, Ban quản lý

1. Đăng nhập / Đăng xuất

- **Bảng liên quan:** NguoiDung
- **Thao tác:**
 - Kiểm tra thông tin tài khoản, xác thực quản lý.

2. Quản lý nhân viên

- **Bảng liên quan:** NguoiDung

- **Thao tác:**

- Tạo tài khoản nhân viên: thêm bản ghi mới vào NhanVien (gồm TaiKhoan, MatKhau, HoTen, email, SĐT, NgayBatDau, TrangThai...)
- Cập nhật thông tin nhân viên
- Xem danh sách nhân viên: truy vấn toàn bộ bảng NguoiDung với vai trò là nhân viên và quản lý.

3. Quản lý phim và thể loại

- **Bảng liên quan:** Phim, TheLoai

- **Thao tác:**

- Thêm phim, thể loại mới chỉnh sửa thông tin phim, hoặc ngừng chiếu phim.
- Có thể có 2 phim cùng tên (sẽ có cảnh báo trùng tên, nếu vẫn muốn đặt trùng tên có thể tiếp tục)
- Cho phép tìm kiếm phim theo tên phim, thể loại, thời lượng phim, ngôn ngữ chính

4. Quản lý suất chiếu

- **Bảng liên quan:** SuatChieu, Phim, PhongChieu

- **Thao tác:**

- Tạo bản ghi mới trong SuatChieu (phim, phòng, thời điểm chiếu, giá mỗi vé trong suất chiếu đó).
- Kiểm tra tránh trùng thời điểm chiếu và phòng.
- Cho phép cập nhật hoặc xóa suất chiếu.
- Cho phép tìm kiếm suất chiếu theo tên phim hoặc phòng chiếu

Phần 2. Phân tích yêu cầu dữ liệu – Vẽ ERD mở rộng

2.1. Liệt kê các thực thể

1. Người dùng

- Mã người dùng: định danh duy nhất (tự cấp).
- Tài khoản: Duy nhất, tối thiểu 8 ký tự, không có dấu cách
- Mật khẩu: Tối thiểu 8 ký tự, có ít nhất 1 chữ viết hoa, 1 ký tự đặc biệt, không chứa dấu cách
- Họ tên: không được để trống.
- Ngày sinh: phải hợp lệ (mm/dd/yyyy), nhỏ hơn ngày hiện tại.
- Số điện thoại: bắt buộc, duy nhất, đúng định dạng.
- Email: bắt buộc, duy nhất, đúng định dạng.
- Ngày bắt đầu làm việc: không được lớn hơn ngày hiện tại. (chỉ dùng cho nhân viên có thể để trống)
- Trạng thái: “Đang làm”, “Nghỉ phép” (chỉ dùng cho nhân viên có thể để trống)
- Vai trò: “Khách hàng”, “Nhân viên”, “Quản lý”

2. Lịch cụ thể

- Mã lịch cụ thể: định danh duy nhất
- Ngày làm việc: ngày trong năm

3. Ca làm việc

- Mã ca làm việc: định danh duy nhất (tăng từ 1 đại diện cho ca 1, ca 2, ...)
- Nội dung: “ca sáng”, “ca chiều”, “ca tối”
- Giờ bắt đầu: giờ trong ngày (≥ 0 , ≤ 24)
- Giờ kết thúc: giờ trong ngày (≥ 0 , ≤ 24), sau giờ bắt đầu

4. Vé xem phim

- Mã vé: định danh duy nhất.
- Trạng thái: “Chưa sử dụng”, “Đã sử dụng”, “Hết hạn”.

5. Phòng chiếu

- Mã phòng chiếu (Số phòng): duy nhất trong rạp
- Sức chứa: một số nguyên dương
- Trạng thái: “Đang hoạt động”, “Bảo trì”

6. Phim

- Mã phim: mã phim duy nhất trong hệ thống.
- Tên: rõ ràng, dễ nhận diện (có thể trùng nhau nhưng quản lý nên đặt tên cẩn thận để dễ quản lý)
- Ngày phát hành: để xác định thời điểm công chiếu.
- Thời lượng: độ dài phim tính bằng phút
- Ngôn ngữ chính: ngôn ngữ ban đầu của phim
- Mô tả nội dung: tóm tắt cốt truyện.

7. Thể loại

- Mã thể loại: định danh duy nhất tăng từ 1
- Nội dung: mô tả thể loại (duy nhất)

8. Suất chiếu

- Mã suất chiếu: mã suất chiếu duy nhất.
- Giá mỗi vé: Số nguyên dương
- Thời gian bắt đầu(cùng 1 phòng thì không được chồng chéo lên nhau), phải lớn hơn thời điểm hiện tại

9. Ghế ngồi

- Mã ghế: mã ghế duy nhất
- Hàng: Ký tự ('A', 'B', 'C' ..., 'Z')
- Cột: Số tự nhiên
- Trạng thái: "Tốt", "Bảo trì"

10. Hóa đơn

- Mã hóa đơn: mã hóa đơn duy nhất.
- Ngày thanh toán: thời điểm thanh toán.
- Trạng thái: "Đã thanh toán", "Đã hủy"

11. Phương thức thanh toán

- Mã phương thức thanh toán: định danh duy nhất, tăng dần từ 1
- Nội dung: nội dung thanh toán (Thẻ ngân hàng, Visa, ...) (duy nhất)

2.2 Ràng buộc giữa các thực thể

1. Cụm đặt vé

- Người dùng (khách hàng) và hóa đơn: 1 - nhiều
 - 1 khách hàng có thể thực hiện tạo nhiều hóa đơn giao dịch nhưng 1 hóa đơn chỉ thuộc 1 khách hàng duy nhất
- Hóa đơn và vé: 1 - nhiều
 - 1 hóa đơn có thể chứa nhiều vé, 1 vé chỉ thuộc 1 hóa đơn duy nhất
- Hóa đơn và phương thức thanh toán: nhiều - 1
 - Nhiều hóa đơn có thể có chung phương thức thanh toán, 1 hóa đơn phải có 1 phương thức thanh toán

2. Cụm phim

- Phim và thể loại: nhiều - nhiều
 - 1 phim có thể có nhiều thể loại, 1 thể loại có thể là của nhiều phim

- 1 phim bắt buộc phải có ít nhất 1 thể loại
- Phim và suất chiếu: 1 - nhiều
 - 1 phim có thể có nhiều suất chiếu, 1 suất chiếu khi được tạo bắt buộc chỉ được chiếu 1 phim
- Phòng và suất chiếu: 1 - nhiều
 - 1 phòng có thể có nhiều suất chiếu khác nhau, 1 suất chiếu phải nằm ở 1 phòng nhất định
 - Đảm bảo 2 suất chiếu không bắt đầu cùng 1 thời điểm tại 1 phòng
- Phòng và ghế: 1 - nhiều
 - 1 phòng có thể có nhiều ghế, 1 ghế bắt buộc phải thuộc về 1 phòng
 - Đảm bảo không có 2 ghế trong cùng 1 phòng có cùng vị trí

3. Cụm vé

- Vé và suất chiếu: nhiều - 1
 - 1 vé bắt buộc có 1 suất chiếu, 1 suất chiếu có thể bán ra nhiều vé
- Vé và ghế: nhiều - 1
 - 1 vé phải được bán cho 1 ghế ngồi, 1 ghế ngồi có thể bán cho nhiều vé khác nhau miễn là khác suất chiếu

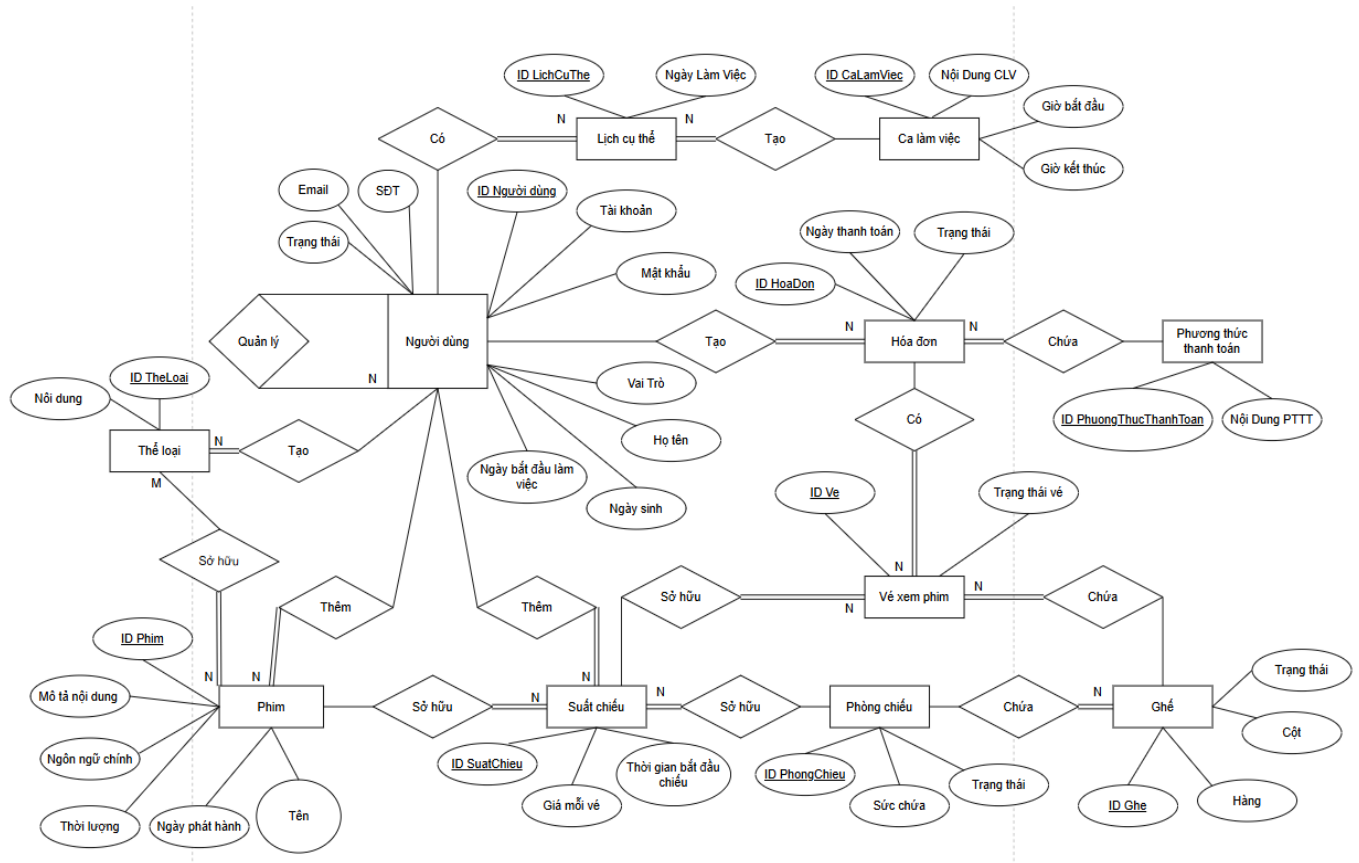
4. Cụm rạp

- Người dùng (quản lý) và người dùng (nhân viên): 1 - nhiều
 - 1 quản lý có thể quản lý nhiều nhân viên, 1 nhân viên phải được quản lý bởi 1 quản lý
- Người dùng (quản lý) và suất chiếu: 1 - nhiều
 - 1 quản lý có thể quản lý nhiều suất chiếu. 1 suất chiếu phải được quản lý bởi 1 quản lý
- Người dùng (quản lý) và phim: 1 - nhiều
 - 1 quản lý có thể quản lý nhiều bộ phim. 1 bộ phim phải được quản lý bởi 1 quản lý
- Người dùng (quản lý) và thể loại: 1 - nhiều

- 1 quản lý có thể tạo nhiều thể loại. 1 thể loại phải được tạo bởi 1 quản lý
- Người dùng (nhân viên) và lịch cụ thể: 1 - nhiều
 - 1 nhân viên có thể có nhiều lịch làm việc cụ thể, 1 lịch làm việc cụ thể mô tả chi tiết 1 nhân viên làm ca nào, ngày nào
- Lịch làm việc cụ thể và ca làm việc: nhiều - 1
 - 1 lịch làm việc chỉ có 1 ca làm, 1 ca làm có thể là của nhiều lịch làm việc cụ thể

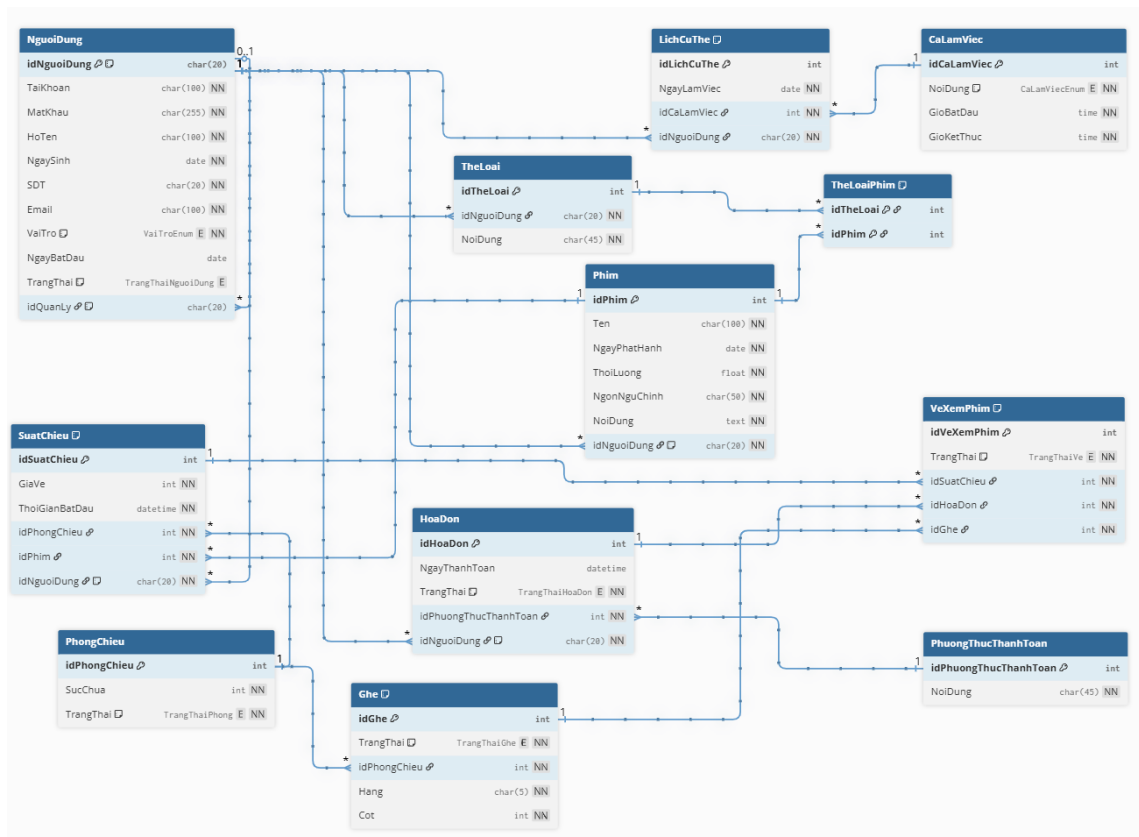
2.3 Vẽ mô hình ER mở rộng

<https://drive.google.com/file/d/17RNc105btrFBgrqwqGMuQKRFRGOcfFs1/view>



Phần 3. Thiết kế mô hình quan hệ

<https://dbdiagram.io/d/691f240c228c5bbc1ac26b21>



1. Bảng Ngoidung

- Các cột của bảng

- idNgoidung: char(20), PrimaryKey, Not Null
- TaiKhoan: char(20), Unique, Not Null
- MatKhau: char(255), Not Null
- Hoten: char(100), Not Null
- NgaySinh: date, Not Null
- SDT: char(20), Not Null, Unique
- Email: char(100), Not Null, Unique
- VaiTro: char(45), Not Null
- NgayBatDau: date
- TrangThai: enum('DANGLAM', 'NGHIPHEP')
- idQuanly: char(20)

- Câu lệnh tạo bảng

```
CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`NguoiDung` (  
  `idNguoiDung` CHAR(20) NOT NULL,  
  `TaiKhoan` CHAR(100) NOT NULL,  
  `MatKau` CHAR(255) NOT NULL,  
  `HoTen` CHAR(100) NOT NULL,  
  `NgaySinh` DATE NOT NULL,  
  `SDT` CHAR(20) NOT NULL,  
  `Email` CHAR(100) NOT NULL,  
  `VaiTro` ENUM('KHACHHANG', 'NHANVIEN', 'QUANLY') NOT NULL,  
  `NgayBatDau` DATE NULL DEFAULT NULL,  
  `TrangThai` ENUM('DANGLAM', 'NGHIPHEP') NULL DEFAULT NULL,  
  `idQuanLy` CHAR(20) NULL DEFAULT NULL,  
  PRIMARY KEY (`idNguoiDung`),  
  UNIQUE INDEX `TaiKhoan_UNIQUE` (`TaiKhoan` ASC) VISIBLE,  
  UNIQUE INDEX `Email_UNIQUE` (`Email` ASC) VISIBLE,  
  UNIQUE INDEX `SDT_UNIQUE` (`SDT` ASC) VISIBLE,  
  INDEX `fk_NguoiDung_NguoiQuanLy_idx` (`idQuanLy` ASC) VISIBLE,  
  CONSTRAINT `fk_NguoiDung_NguoiQuanLy`  
    FOREIGN KEY (`idQuanLy`)  
    REFERENCES `QuanLyBanVeOnline`.`NguoiDung` (`idNguoiDung`)  
    ON DELETE SET NULL)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_bin;
```

2. Bảng CaLamViec

- Các cột của bảng

- idCaLamViec: int, PrimaryKey, Not Null, AutoIncrement
- NoiDung: enum('CASANG', 'CACHIEU', 'CATOI'), Not Null
- GioBatDau: time, Not Null
- GioKetThuc: time, Not Null

- Câu lệnh tạo bảng

```

CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`CaLamViec` (
  `idCaLamViec` INT NOT NULL AUTO_INCREMENT,
  `NoiDung` ENUM('CASANG', 'CACHIEU', 'CATOI') NOT NULL,
  `GioBatDau` TIME NOT NULL,
  `GioKetThuc` TIME NOT NULL,
  PRIMARY KEY (`idCaLamViec`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_bin;

```

3. Bảng LịchCuThe

- Các cột của bảng

- idLichCuThe: int, PrimaryKey, Not Null, AutoIncrement, Unique
- NgayLamViec: date, Not Null
- idCaLamViec: int, Not Null, ForeignKey
- idNguoiDung: char(20), Not Null, ForeignKey
- Ràng buộc: Unique(NgayLamViec, idCaLamViec, idNguoiDung) -

Một nhân viên không làm trùng ca trong một ngày

- Câu lệnh tạo bảng

```

CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`LichCuThe` (
  `idLichCuThe` INT NOT NULL AUTO_INCREMENT,
  `NgayLamViec` DATE NOT NULL,
  `idCaLamViec` INT NOT NULL,
  `idNguoiDung` CHAR(20) NOT NULL,
  PRIMARY KEY (`idLichCuThe`),
  UNIQUE INDEX `idLichCuThe_UNIQUE` (`idLichCuThe` ASC) VISIBLE,
  UNIQUE INDEX `PhanCong_UNIQUE` (`NgayLamViec` ASC, `idCaLamViec` ASC, `idNguoiDung` ASC) VISIBLE,
  INDEX `fk_LichCuThe_CaLamViec1_idx` (`idCaLamViec` ASC) VISIBLE,
  INDEX `fk_LichCuThe_NguoiDung1_idx` (`idNguoiDung` ASC) VISIBLE,
  CONSTRAINT `fk_LichCuThe_CaLamViec1`
    FOREIGN KEY (`idCaLamViec`)
      REFERENCES `QuanLyBanVeOnline`.`CaLamViec` (`idCaLamViec`),
  CONSTRAINT `fk_LichCuThe_NguoiDung1`
    FOREIGN KEY (`idNguoiDung`)
      REFERENCES `QuanLyBanVeOnline`.`NguoiDung` (`idNguoiDung`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_bin;

```

4. Bảng PhongChieu

- Các cột của bảng

- idPhongChieu: int, PrimaryKey, Not Null, AutoIncrement
- SucChua: int, Not Null
- TrangThai: enum('DANGHOATDONG', 'BAOTRI'), Not Null

- Câu lệnh tạo bảng

```
CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`PhongChieu` (  
  `idPhongChieu` INT NOT NULL AUTO_INCREMENT,  
  `SucChua` INT NOT NULL,  
  `TrangThai` ENUM('DANGHOATDONG', 'BAOTRI') NOT NULL,  
  PRIMARY KEY (`idPhongChieu`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 4  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_bin;
```

5. Bảng Phim

- Các cột của bảng

- idPhim: int, PrimaryKey, Not Null, AutoIncrement
- Ten: char(100), Not Null
- NgayPhatHanh: date, Not Null
- ThoiLuong: float, Not Null
- NgonNguChinh: char(50), Not Null
- NoiDung: text, Not Null
- idNguoiDung: char(20), Not Null, ForeignKey

- Câu lệnh tạo bảng

```

CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`Phim` (
  `idPhim` INT NOT NULL AUTO_INCREMENT,
  `Ten` CHAR(100) NOT NULL,
  `NgayPhatHanh` DATE NOT NULL,
  `ThoiLuong` FLOAT NOT NULL,
  `NgonNguChinh` CHAR(50) NOT NULL,
  `NoiDung` TEXT NOT NULL,
  `idNguoiDung` CHAR(20) NOT NULL,
  PRIMARY KEY (`idPhim`),
  INDEX `fk_Phim_NguoiDung1_idx` (`idNguoiDung` ASC) VISIBLE,
  CONSTRAINT `fk_Phim_NguoiDung1`
    FOREIGN KEY (`idNguoiDung`)
      REFERENCES `QuanLyBanVeOnline`.`NguoiDung` (`idNguoiDung`))
ENGINE = InnoDB
AUTO_INCREMENT = 11
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_bin;

```

6. Bảng SuatChieu

- Các cột của bảng

- idSuatChieu: int, PrimaryKey, Not Null, AutoIncrement
- GiaVe: int, Not Null
- ThoiGianBatDau: datetime, Not Null
- idPhongChieu: int, Not Null, ForeignKey
- idPhim: int, Not Null, ForeignKey
- idNguoiDung: char(20), Not Null, ForeignKey
- Ràng buộc: Unique(idPhongChieu, ThoiGianBatDau) - Một phòng không thể có 2 suất chiếu bắt đầu cùng thời điểm

- Câu lệnh tạo bảng

```

CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`SuatChieu` (
  `idSuatChieu` INT NOT NULL AUTO_INCREMENT,
  `GiaVe` INT NOT NULL,
  `ThoiGianBatDau` DATETIME NOT NULL,
  `idPhongChieu` INT NOT NULL,
  `idPhim` INT NOT NULL,
  `idNguoiDung` CHAR(20) NOT NULL,
  PRIMARY KEY (`idSuatChieu`),
  UNIQUE INDEX `TranhTrungGio_UNIQUE` (`idPhongChieu` ASC, `ThoiGianBatDau` ASC) VISIBLE,
  INDEX `fk_SuatChieu_PhongChieu1_idx` (`idPhongChieu` ASC) VISIBLE,
  INDEX `fk_SuatChieu_Phim1_idx` (`idPhim` ASC) VISIBLE,
  INDEX `fk_SuatChieu_NguoiDung1_idx` (`idNguoiDung` ASC) VISIBLE,
  CONSTRAINT `fk_SuatChieu_NguoiDung1`
    FOREIGN KEY (`idNguoiDung`)
    REFERENCES `QuanLyBanVeOnline`.`NguoiDung` (`idNguoiDung`),
  CONSTRAINT `fk_SuatChieu_Phim1`
    FOREIGN KEY (`idPhim`)
    REFERENCES `QuanLyBanVeOnline`.`Phim` (`idPhim`),
  CONSTRAINT `fk_SuatChieu_PhongChieu1`
    FOREIGN KEY (`idPhongChieu`)
    REFERENCES `QuanLyBanVeOnline`.`PhongChieu` (`idPhongChieu`))
ENGINE = InnoDB
AUTO_INCREMENT = 23
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_bin;

```

7. Bảng PhuongThucThanhToan

- Các cột của bảng

- idPhuongThucThanhToan: int, PrimaryKey, Not Null, AutoIncrement
- NoiDung: char(45), Unique, Not Null

- Câu lệnh tạo bảng

```

CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`PhuongThucThanhToan` (
  `idPhuongThucThanhToan` INT NOT NULL AUTO_INCREMENT,
  `NoiDung` CHAR(45) NOT NULL,
  PRIMARY KEY (`idPhuongThucThanhToan`),
  UNIQUE INDEX `NoiDung_UNIQUE` (`NoiDung` ASC) VISIBLE)
ENGINE = InnoDB
AUTO_INCREMENT = 5
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_bin;

```

8. Bảng HoaDon

- Các cột của bảng

1. Các cột của bảng

- idHoaDon: int, PrimaryKey, Not Null, AutoIncrement
- NgayThanhToan: datetime, Not Null
- TrangThai: enum('DATHANHTOAN', 'DAHUY'), Not Null
- idPhuongThucThanhToan: int, Not Null, ForeignKey
- idNguoiDung: char(20), Not Null, ForeignKey

- Câu lệnh tạo bảng

```
CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`HoaDon` (  
  `idHoaDon` INT NOT NULL AUTO_INCREMENT,  
  `NgayThanhToan` DATETIME NULL DEFAULT NULL,  
  `TrangThai` ENUM('DATHANHTOAN', 'DAHUY') NOT NULL,  
  `idPhuongThucThanhToan` INT NOT NULL,  
  `idNguoiDung` CHAR(20) NOT NULL,  
  PRIMARY KEY (`idHoaDon`),  
  INDEX `fk_HoaDon_PhuongThucThanhToan1_idx` (`idPhuongThucThanhToan` ASC) VISIBLE,  
  INDEX `fk_HoaDon_NguoiDung1_idx` (`idNguoiDung` ASC) VISIBLE,  
  CONSTRAINT `fk_HoaDon_NguoiDung1`  
    FOREIGN KEY (`idNguoiDung`)  
    REFERENCES `QuanLyBanVeOnline`.`NguoiDung` (`idNguoiDung`),  
  CONSTRAINT `fk_HoaDon_PhuongThucThanhToan1`  
    FOREIGN KEY (`idPhuongThucThanhToan`)  
    REFERENCES `QuanLyBanVeOnline`.`PhuongThucThanhToan` (`idPhuongThucThanhToan`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 8  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_bin;
```

9. Bảng Ghe

- Các cột của bảng

- idGhe: int, PrimaryKey, Not Null, AutoIncrement, Unique
- TrangThai: enum('TOT', 'BAOTRI'), Not Null
- idPhongChieu: int, Not Null, ForeignKey
- Hang: char(5), Not Null
- Cot: int, Not Null
- Ràng buộc: Unique(Cot, Hang, idPhongChieu) - Không được trùng vị trí ghế trong cùng một phòng.

- Câu lệnh tạo bảng

```
CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`Ghe` (  
  `idGhe` INT NOT NULL AUTO_INCREMENT,  
  `TrangThai` ENUM('TOT', 'BAOTRI') NOT NULL,  
  `idPhongChieu` INT NOT NULL,  
  `Hang` CHAR(5) NOT NULL,  
  `Cot` INT NOT NULL,  
  PRIMARY KEY (`idGhe`),  
  UNIQUE INDEX `idGhe_UNIQUE` (`idGhe` ASC) VISIBLE,  
  UNIQUE INDEX `Vi_Tri_Ghe` (`Cot` ASC, `Hang` ASC, `idPhongChieu` ASC) VISIBLE,  
  INDEX `fk_Ghe_PhongChieu1_idx` (`idPhongChieu` ASC) VISIBLE,  
  CONSTRAINT `fk_Ghe_PhongChieu1`  
    FOREIGN KEY (`idPhongChieu`)  
      REFERENCES `QuanLyBanVeOnline`.`PhongChieu` (`idPhongChieu`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 81  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_bin;
```

•

10. Bảng VeXemPhim

- Các cột của bảng

- idVeXemPhim: int, PrimaryKey, Not Null, AutoIncrement
- TrangThai: enum('CHUASUDUNG', 'DASUDUNG', 'HETHAN'), Not Null
- idSuatChieu: int, Not Null, ForeignKey
- idHoaDon: int, Not Null, ForeignKey
- idGhe: int, Not Null, ForeignKey
- Ràng buộc: Unique(idSuatChieu, idGhe) - Một ghế trong một suất chiếu chỉ được bán 1 lần.

- Câu lệnh tạo bảng

```

CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`VeXemPhim` (
  `idVeXemPhim` INT NOT NULL AUTO_INCREMENT,
  `TrangThai` ENUM('CHUASUDUNG', 'DASUDUNG', 'HETHAN') NOT NULL,
  `idSuatChieu` INT NOT NULL,
  `idHoaDon` INT NOT NULL,
  `idGhe` INT NOT NULL,
  PRIMARY KEY (`idVeXemPhim`),
  UNIQUE INDEX `TrungVe_UNIQUE` (`idSuatChieu` ASC, `idGhe` ASC) INVISIBLE,
  INDEX `fk_VeXemPhim_SuatChieu1_idx` (`idSuatChieu` ASC) VISIBLE,
  INDEX `fk_VeXemPhim_HoaDon1_idx` (`idHoaDon` ASC) VISIBLE,
  INDEX `fk_VeXemPhim_Ghe1_idx` (`idGhe` ASC) VISIBLE,
  CONSTRAINT `fk_VeXemPhim_Ghe1`
    FOREIGN KEY (`idGhe`)
      REFERENCES `QuanLyBanVeOnline`.`Ghe` (`idGhe`),
  CONSTRAINT `fk_VeXemPhim_HoaDon1`
    FOREIGN KEY (`idHoaDon`)
      REFERENCES `QuanLyBanVeOnline`.`HoaDon` (`idHoaDon`),
  CONSTRAINT `fk_VeXemPhim_SuatChieu1`
    FOREIGN KEY (`idSuatChieu`)
      REFERENCES `QuanLyBanVeOnline`.`SuatChieu` (`idSuatChieu`))
ENGINE = InnoDB
AUTO_INCREMENT = 30
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_bin;

```

11. Bảng TheLoai

- Các cột của bảng

- idTheLoai: int, PrimaryKey, Not Null, AutoIncrement
- idNguoiDung: char(20), Not Null, ForeignKey
- NoiDung: char(45), Unique, Not Null

- Câu lệnh tạo bảng

```

CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`TheLoai` (
  `idTheLoai` INT NOT NULL AUTO_INCREMENT,
  `NoiDung` CHAR(45) NOT NULL,
  `idNguoiDung` CHAR(20) NOT NULL,
  PRIMARY KEY (`idTheLoai`),
  UNIQUE INDEX `NoiDung_UNIQUE` (`NoiDung` ASC) VISIBLE,
  INDEX `fk_TheLoai_NguoiDung1_idx` (`idNguoiDung` ASC) VISIBLE,
  CONSTRAINT `fk_TheLoai_NguoiDung1`
    FOREIGN KEY (`idNguoiDung`)
      REFERENCES `QuanLyBanVeOnline`.`NguoiDung` (`idNguoiDung`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 9
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_bin;

```

12. Bảng TheLoaiPhim

- Các cột của bảng

- idTheLoai: int, PrimaryKey, Not Null, ForeignKey
- idPhim: int, PrimaryKey, Not Null, ForeignKey

- Câu lệnh tạo bảng

```
CREATE TABLE IF NOT EXISTS `QuanLyBanVeOnline`.`TheLoaiPhim` (  
  `idTheLoai` INT NOT NULL,  
  `idPhim` INT NOT NULL,  
  PRIMARY KEY (`idTheLoai`, `idPhim`),  
  INDEX `fk_TheLoai_has_Phim_Phim1_idx` (`idPhim` ASC) VISIBLE,  
  INDEX `fk_TheLoai_has_Phim_TheLoai1_idx` (`idTheLoai` ASC) VISIBLE,  
  CONSTRAINT `fk_TheLoai_has_Phim_Phim1`  
    FOREIGN KEY (`idPhim`)  
      REFERENCES `QuanLyBanVeOnline`.`Phim` (`idPhim`),  
  CONSTRAINT `fk_TheLoai_has_Phim_TheLoai1`  
    FOREIGN KEY (`idTheLoai`)  
      REFERENCES `QuanLyBanVeOnline`.`TheLoai` (`idTheLoai`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_bin;
```

Phần 4. Kiểm tra chuẩn hóa các bảng về chuẩn 3

TT	Tên bảng	Khóa chính	1NF	2NF	3NF	BCNF
1	CaLamViec	idCaLamViec	Đạt	Đạt	Đạt	Đạt
2	PhongChieu	idPhongChieu	Đạt	Đạt	Đạt	Đạt
3	Ghe	idGhe	Đạt	Đạt	Đạt	Đạt
4	NguoiDung	idNguoiDung	Đạt	Đạt	Đạt	Đạt
5	PhuongThucThanhToan	idPhuongThucThanhToan	Đạt	Đạt	Đạt	Đạt
6	HoaDon	idHoaDon	Đạt	Đạt	Đạt	Đạt
7	LichCuThe	idLichCuThe	Đạt	Đạt	Đạt	Đạt
8	Phim	idPhim	Đạt	Đạt	Đạt	Đạt
9	SuatChieu	idSuatChieu	Đạt	Đạt	Đạt	Đạt
10	TheLoai	idTheLoai	Đạt	Đạt	Đạt	Đạt
11	TheLoaiPhim	idTheLoai, idPhim	Đạt	Đạt	Đạt	Đạt
12	VeXemPhim	idVeXemPhim	Đạt	Đạt	Đạt	Đạt

Phần 5. Demo 2 biểu mẫu nhập liệu & SQL tương ứng

5.1 Biểu mẫu quản lý Phim

Hệ Thống Bán Vé Xem Phim Online - Nhóm 13

QUẢN LÝ HỆ THỐNG Xin chào, Quản Lý Rạp Đăng xuất

Quản lý Phim Quản lý Suất chiếu Quản lý Thể loại

Thông tin Phim

Tên Phim:

Ngày PH:

Thời lượng (p...):

Ngôn ngữ:

Nội dung:

Lưu Phim Làm mới form

Bộ lọc & Gán Thể loại

Tìm kiếm theo tên, NN, TL...

Lọc:

Danh sách Thể loại (Chọn để gán):

Đã tải lại toàn bộ danh sách.

ID ▲	Tên Phim	Ngày PH	TL	Ngôn ngữ	Thể loại	SL SC	Thao tác
1	Đào, Phở và Piano	2024-02-10	100.0	Tiếng Việt	Tình cảm, Tình cảm, Tình ...	6	<input type="button" value="Sửa"/>
2	Mai	2024-02-10	131.0	Tiếng Việt	Tình cảm, Tình cảm, Hài ...	4	<input type="button" value="Sửa"/>
3	Kung Fu Panda 4	2024-03-08	94.0	Tiếng Anh	Hành động, Hoạt hình	0	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
4	Dune: Part Two	2024-03-01	166.0	Tiếng Anh	Hành động, Hành động, ...	4	<input type="button" value="Sửa"/>
5	1111111	2025-11-13	11.0	111	Hoạt hìna, 111	0	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
6	123231	2025-10-30	11.0	11	Hoạt hình	0	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
7	1	2025-11-04	1.0	11	111	0	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>

1. Nút Lưu Phim (btnSavePhim) - savePhim()

Trường hợp 1: THÊM PHIM MỚI (isNew = true)

1a. INSERT Phim

- Câu lệnh SQL

```
INSERT INTO Phim (Ten, NgayPhatHanh, ThoiLuong, NgonNguChinh,
NoiDung, idNguoiDung)
```

```
VALUES (?, ?, ?, ?, ?, ?)
```

- Ten: Tên phim từ txtTenPhim

- NgayPhatHanh: Ngày từ dpNgayPhatHanh

- ThoiLuong: Thời lượng từ txtThoiLuong (float - phút)

- NgonNguChinh: Ngôn ngữ từ txtNgonNguChinh

- NoiDung: Nội dung từ txtNoiDung
- idNguoiDung: ID quản lý hiện tại từ UserSession

1b. INSERT Liên kết Thẻ loại

- Câu lệnh SQL

```
INSERT INTO TheLoaiPhim (idTheLoai, idPhim)
VALUES (?, ?)
```

- Thực hiện theo batch cho mỗi thẻ loại được chọn (từ selectedGenreIds)
- idTheLoai: ID của thẻ loại được chọn
- idPhim: ID phim vừa được tạo (lấy từ GENERATED KEYS)

Trường hợp 2: CẬP NHẬT PHIM (isNew = false)

2a. UPDATE Phim

- Câu lệnh SQL

```
UPDATE Phim
SET Ten = ?, NgayPhatHanh = ?, ThoiLuong = ?, NgonNguChinh = ?,
NoiDung = ?
WHERE idPhim = ?
```

- Ten, NgayPhatHanh, ThoiLuong, NgonNguChinh, NoiDung: Dữ liệu mới từ form
- idPhim: ID phim được chọn

2b. DELETE liên kết thẻ loại cũ

- Câu lệnh SQL

```
DELETE FROM TheLoaiPhim
WHERE idPhim = ?
```

- Xóa tất cả liên kết cũ trước khi thêm mới

2c. INSERT liên kết thẻ loại mới

- Câu lệnh SQL

```
INSERT INTO TheLoaiPhim (idTheLoai, idPhim) VALUES (?, ?)
```

- Thực hiện theo batch cho mỗi thẻ loại được chọn (từ selectedGenreIds)
- idTheLoai: ID của thẻ loại được chọn

- idPhim: ID phim vừa được tạo (lấy từ GENERATED KEYS)

Ghi chú: Tất cả 3 câu lệnh trên thực hiện trong một TRANSACTION (bắt đầu bằng conn.setAutoCommit(false) và kết thúc bằng conn.commit())

2. NÚT KIỂM TRA TRÙNG TÊN PHIM - checkDuplicateTen()

- Câu lệnh SQL

```
SELECT COUNT()
```

```
FROM Phim
```

```
WHERE Ten = ?
```

```
AND idPhim != ?
```

- Ten: Tên phim cần kiểm tra

- idPhim: ID phim hiện tại (0 nếu là phim mới)

- Nếu kết quả > 0 thì có trùng, hiển thị cảnh báo xác nhận

3. NÚT XÓA PHIM (deleteButton) - handleDeletePhim()

Điều kiện: Chỉ xóa được nếu phim KHÔNG CÓ suất chiếu (showtimeCount == 0)

3a. DELETE Liên kết Thẻ loại

- Câu lệnh SQL

```
DELETE FROM TheLoaiPhim
```

```
WHERE idPhim = ?
```

3b. DELETE Phim

- Câu lệnh SQL

```
DELETE FROM Phim
```

```
WHERE idPhim = ?
```

Ghi chú:

- Nếu phim có suất chiếu, nút "Xóa" sẽ bị vô hiệu hóa (disabled)

- Lỗi FK từ bảng SuatChieu hoặc HoaDon sẽ gây ra rollback

4. NÚT TÌM KIẾM PHIM - handleSearchPhim() / handleCombinedSearch()

Thực hiện tìm kiếm kết hợp bằng STREAM API trên dữ liệu đã load (không gọi SQL trực tiếp)

Nhưng dữ liệu gốc được load bằng:

- Câu lệnh SQL

```
SELECT p.,  
       GROUP_CONCAT(tl.NoiDung SEPARATOR ', ') as DanhSachTheLoai,  
       COUNT(sc.idSuatChieu) as ShowtimeCount  
FROM Phim p  
LEFT JOIN TheLoaiPhim tlp ON p.idPhim = tlp.idPhim  
LEFT JOIN TheLoai tl ON tlp.idTheLoai = tl.idTheLoai  
LEFT JOIN SuatChieu sc ON p.idPhim = sc.idPhim  
GROUP BY p.idPhim  
ORDER BY p.idPhim DESC
```

Các bộ lọc được áp dụng:

- Tìm kiếm Text: Tìm trong Ten, NgonNguChinh, TheLoai (case-insensitive)
- Lọc Thẻ loại: So khớp với chuỗi thẻ loại
- Lọc Thời lượng:
 - "Dưới 90 phút": ThoiLuong < 90
 - "90 - 120 phút": 90 <= ThoiLuong <= 120
 - "Trên 120 phút": ThoiLuong > 120
- Lọc Ngôn ngữ: So khớp chính xác NgonNguChinh

5. NÚT LẤY ID THẺ LOẠI CỦA PHIM - getGenreIdsByPhimId()

- Câu lệnh SQL

```
SELECT idTheLoai FROM TheLoaiPhim WHERE idPhim = ?
```

- Dùng khi click chọn phim để hiển thị các thẻ loại được gán

6. NÚT TẢI TOÀN BỘ PHIM - loadPhimData() / getAllPhim()

- Câu lệnh SQL

```
SELECT p.,  
       GROUP_CONCAT(tl.NoiDung SEPARATOR ', ') as DanhSachTheLoai,  
       COUNT(sc.idSuatChieu) as ShowtimeCount  
FROM Phim p  
LEFT JOIN TheLoaiPhim tlp ON p.idPhim = tlp.idPhim
```



```
LEFT JOIN TheLoai tl ON tlp.idTheLoai = tl.idTheLoai
LEFT JOIN SuatChieu sc ON p.idPhim = sc.idPhim
GROUP BY p.idPhim
ORDER BY p.idPhim DESC
```

- Gọi khi khởi tạo hoặc refresh danh sách
- ShowtimeCount dùng để kiểm tra khóa nút "Xóa"

5.2 Biểu mẫu quản lý Suất chiếu

[illegible]

1. NÚT LƯU SUẤT CHIẾU (btnSaveShowtime) - handleSaveShowtime()

Trường hợp 1: THÊM SUẤT CHIẾU MỚI

- Câu lệnh SQL

```
INSERT INTO SuatChieu (GiaVe, ThoiGianBatDau, idPhongChieu, idPhim,
idNguoiDung)
VALUES (?, ?, ?, ?, ?)
```

- GiaVe: Giá vé từ txtGiaVe (int - VND)

- ThoiGianBatDau: Ngày giờ từ dpNgayChieu + spHour + spMinute

- idPhongChieu: ID phòng chiếu từ cboPhongChieu (int)

- idPhim: ID phim từ cboPhim (int)

- idNguoiDung: ID quản lý hiện tại từ UserSession

Trường hợp 2: CẬP NHẬT SUẤT CHIỀU

- Câu lệnh SQL

```
UPDATE SuatChieu
```

```
SET GiaVe = ?, ThoiGianBatDau = ?, idPhongChieu = ?, idPhim = ?
```

```
WHERE idSuatChieu = ?
```

- GiaVe, ThoiGianBatDau, idPhongChieu, idPhim: Dữ liệu mới từ form

- idSuatChieu: ID suất chiều được chọn

2. NÚT KIỂM TRA TRÙNG LỊCH CHIỀU - isOverlap()

- Câu lệnh SQL

```
SELECT sc.ThoiGianBatDau, p.ThoiLuong
```

```
FROM SuatChieu sc
```

```
JOIN Phim p ON sc.idPhim = p.idPhim
```

```
WHERE sc.idPhongChieu = ? AND sc.idSuatChieu != ?
```

- idPhongChieu: ID phòng cần kiểm tra

- idSuatChieu: ID suất chiều hiện tại (0 nếu mới)

Logic kiểm tra:

- Tính $ThoiGianKetThuc = ThoiGianBatDau + ThoiLuong$ (phút)

- Kiểm tra: $(ExistingStart < NewEnd) \text{ AND } (NewStart < ExistingEnd)$

- Nếu TRUE \rightarrow Trùng lịch, từ chối lưu

3. NÚT TẢI SUẤT CHIỀU CHI TIẾT -loadSuatChieuData(), getAllSuatChieuDetails()

- Câu lệnh SQL

```
SELECT sc., p.Ten as TenPhim
```

```
FROM SuatChieu sc
```

```
JOIN Phim p ON sc.idPhim = p.idPhim
```

```
ORDER BY sc.ThoiGianBatDau DESC
```

- Join với Phim để lấy TenPhim (hiển thị trong table)

- Sắp xếp theo thời gian bắt đầu (mới nhất trước)

4. NÚT LOAD SUẤT CHIỀU THEO PHIM (khách hàng xem) getSuatChieuByPhim()

- Câu lệnh SQL

```
SELECT FROM SuatChieu
WHERE idPhim = ? AND ThoiGianBatDau > NOW()
ORDER BY ThoiGianBatDau ASC
```

- idPhim: ID phim được chọn
- Điều kiện: Chỉ lấy suất chiếu chưa diễn ra (> hiện tại)
- Sắp xếp theo thời gian sắp tới

5. NÚT XÓA SUẤT CHIẾU (deleteButton) - handleDeleteShowtime()

- Câu lệnh SQL

```
DELETE FROM SuatChieu
WHERE idSuatChieu = ?
```

Điều kiện: Chỉ xóa được nếu suất chiếu KHÔNG CÓ vé nào liên quan

- Nếu FK lỗi (có vé), quá trình DELETE sẽ thất bại
- Hiển thị thông báo: "Suất chiếu đã có vé hoặc lỗi CSDL"

6. NÚT TÌM KIẾM SUẤT CHIẾU - handleSearchShowtime()

Thực hiện tìm kiếm kết hợp bằng STREAM API trên dữ liệu đã load (không gọi SQL trực tiếp)

Dữ liệu được lọc theo:

- Phim: So khớp idPhim với cboSearchPhim.getValue()
- Phòng chiếu: So khớp idPhongChieu với cboSearchPhong.getValue()
- Cả hai bộ lọc là optional (NULL → không lọc)

```
List<SuatChieu> filteredList = masterSuatChieuList.stream()
    .filter(sc -> {
        boolean matchPhim = (searchPhim == null) || (sc.getIdPhim() ==
searchPhim.getIdPhim());
        boolean matchPhong = (searchPhong == null) || (sc.getIdPhongChieu() ==
searchPhong.getIdPhongChieu());
        return matchPhim && matchPhong;
    })
    .collect(Collectors.toList());
```

7. NÚT LẤY PHIM/PHÒNG CHO COMBOBOX - Helper Methods

Lấy Phim theo ID:

- Câu lệnh SQL

(Được lọc từ danh sách đã load bằng getAllPhim())

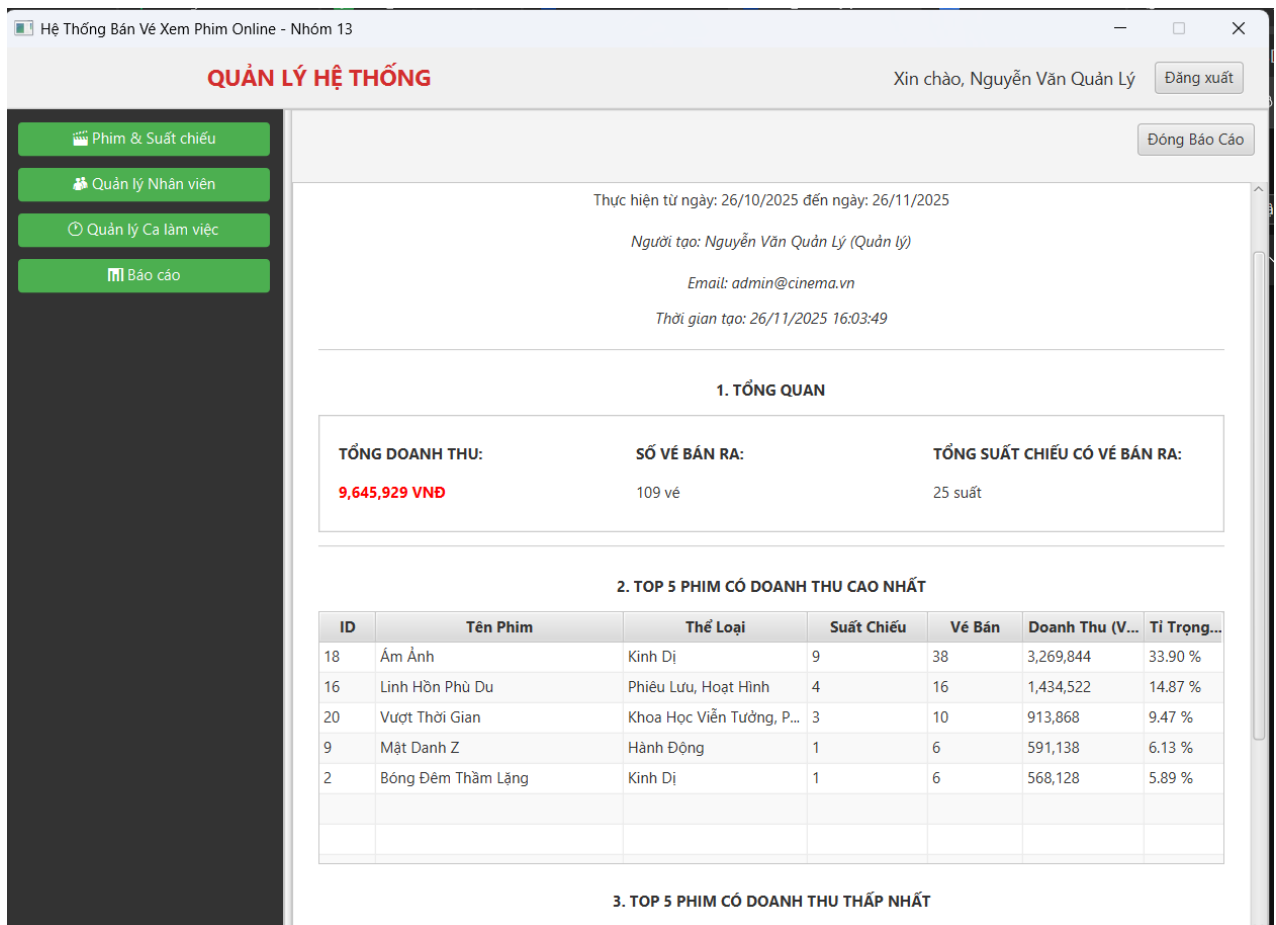
Lấy Phòng theo ID:

- Câu lệnh SQL

(Được lọc từ danh sách đã load bằng getAllPhongChieu())

Phần 6. Demo 2 biểu mẫu báo cáo + SQL

6.1 Biểu mẫu báo cáo doanh thu



1. NÚT TẠO BÁO CÁO - handleGenerateReport() / fetchFullReport()

- Câu lệnh SQL chính - Lấy báo cáo doanh thu theo khoảng thời gian:

SELECT

p.idPhim,

p.Ten,

(

SELECT GROUP_CONCAT(tl.NoiDung SEPARATOR ',')

FROM TheLoaiPhim tlp

JOIN TheLoai tl ON tlp.idTheLoai = tl.idTheLoai

WHERE tlp.idPhim = p.idPhim

) AS TheLoaiList,

```

COUNT(DISTINCT CASE WHEN h.idHoaDon IS NOT NULL THEN
sc.idSuatChieu END) AS TongSuatChieu,
COUNT(DISTINCT CASE WHEN h.idHoaDon IS NOT NULL THEN
v.idVeXemPhim END) AS SoLuongVe,
COALESCE(SUM(CASE WHEN h.idHoaDon IS NOT NULL THEN
sc.GiaVe ELSE 0 END), 0) AS TongDoanhThu
FROM Phim p
LEFT JOIN SuatChieu sc ON p.idPhim = sc.idPhim
LEFT JOIN VeXemPhim v ON sc.idSuatChieu = v.idSuatChieu
LEFT JOIN HoaDon h ON v.idHoaDon = h.idHoaDon
AND h.TrangThai = 'DATHANHTOAN'
AND h.NgayThanhToan BETWEEN ? AND ?
GROUP BY p.idPhim, p.Ten
HAVING TongSuatChieu > 0
ORDER BY TongDoanhThu DESC;

```

Tham số:

- ? (1): startDate.atStartOfDay() - Ngày bắt đầu (00:00:00)
- ? (2): endDate.plusDays(1).atStartOfDay() - Ngày kết thúc + 1 ngày (00:00:00)

Kết quả trả về:

- idPhim: ID duy nhất của phim
- Ten: Tên phim
- TongSuatChieu: Tổng số suất chiếu có ít nhất 1 vé bán ra, COUNT DISTINCT để tránh nhân bản dữ liệu
- SoLuongVe: Tổng số vé đã bán và đã thanh toán, COUNT DISTINCT để tránh đếm trùng vé
- TongDoanhThu: Tổng doanh thu thực tế từ vé bán ra, tính bằng SUM(GiaVe), nếu không có vé thì trả về 0 (COALESCE)
- TheLoaiList: Danh sách thể loại phim, gom thành chuỗi phân cách bằng dấu , (dùng GROUP_CONCAT trong SQL)

Logic xử lý sau khi lấy dữ liệu:

1. Tính tổng hợp (Summary):

summary.tongSoVe = SUM(SoLuongVe)

summary.tongDoanhThu = SUM(TongDoanhThu)

summary.tongSuatChieu = SUM(TongSuatChieu)

2. Tính tỉ trọng doanh thu:

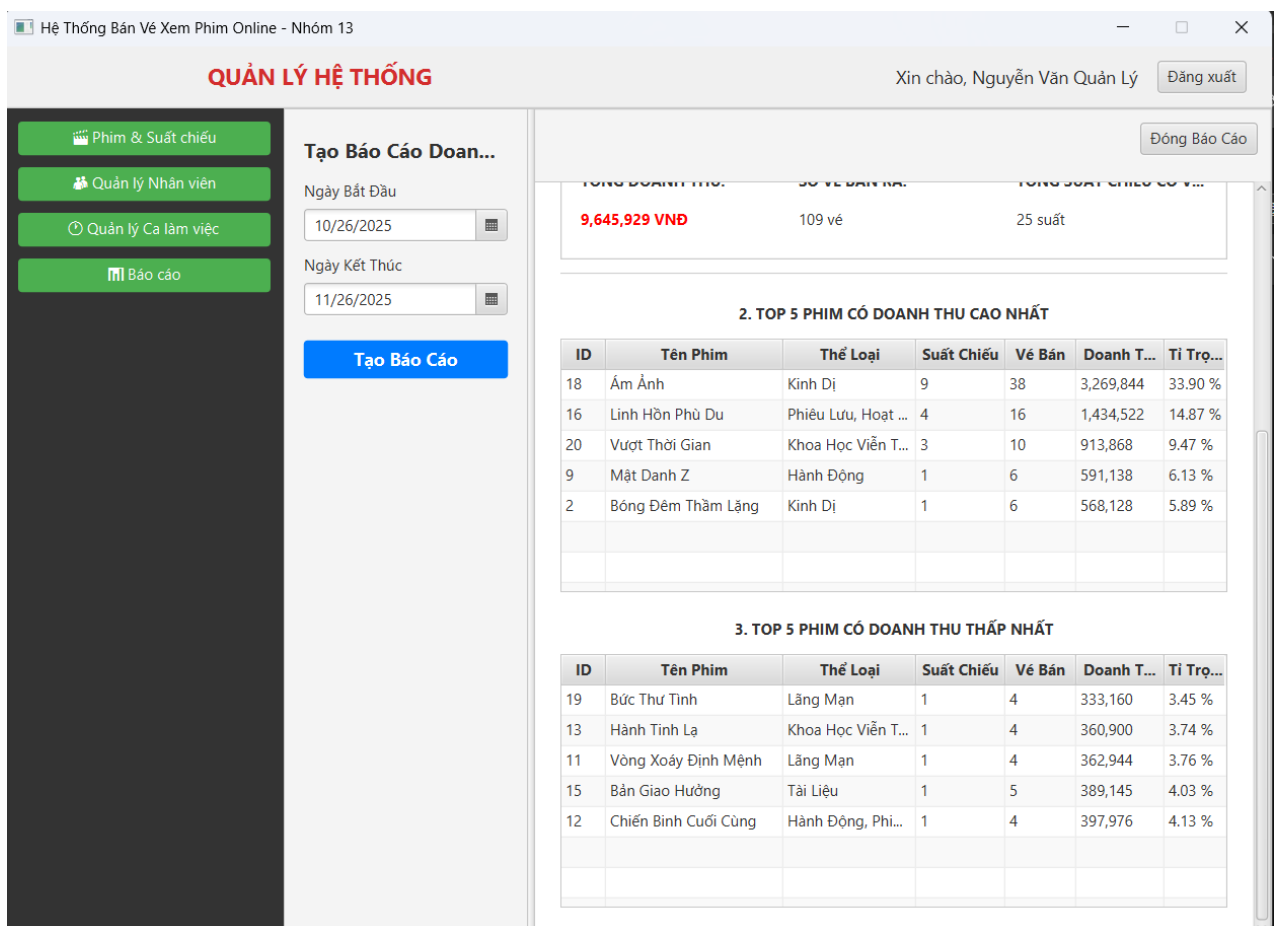
tiTrong = (TongDoanhThu / TotalRevenue) 100 (%)

3. Phân loại Top 5 & Bottom 5:

Top 5: sắp xếp giảm dần theo TongDoanhThu, lấy 5 phim đầu

Bottom 5: sắp xếp tăng dần theo TongDoanhThu, lấy 5 phim đầu

2. CÁC THÀNH PHẦN HIỂN THỊ TRONG BÁO CÁO



Thông tin báo cáo:

- Khoảng thời gian: Từ startDate đến endDate
- Người tạo: Lấy từ UserSession.getInstance().getCurrentUser()

- Họ tên
- Vai trò (QUANLY → "Quản lý")
- Email
- **Thời gian tạo:** LocalDateTime.now()

Tóm tắt:

- **Tổng doanh thu:** SUM(TongDoanhThu) (VNĐ)
- **Tổng số vé:** SUM(SoLuongVe) (vé)
- **Tổng số suất chiếu có vé bán ra:** SUM(TongSuatChieu) (suất)

Chi tiết:

- **Top 5 phim có doanh thu cao nhất**
- **Bottom 5 phim có doanh thu thấp nhất**

6.2 Biểu mẫu báo cáo hóa đơn thanh toán

Hệ Thống Bán Vé Xem Phim Online - Nhóm 13

< Trở về lịch sử

PHIẾU THANH TOÁN (HÓA ĐƠN)

Hệ thống Bán vé xem phim Online - Nhóm 13

THÔNG TIN KHÁCH HÀNG
Họ tên: Nguyễn Văn Khách
Số điện thoại: 0909000002 Email: khachtest@gmail.com

Mã HD: 2 Ngày TT: 22:45 25/11/2025

CHI TIẾT ĐẶT VÉ
Phòng: 2 Thời điểm: 14:00 01/12/2025
Giá/Vé: 60,000 VNĐ Số lượng: 1
Phim: Đảo, Phở và Piano

GHẾ ĐÃ CHỌN:
E3

TỔNG KẾT
Phương thức: Thẻ Quốc tế (Visa/Master) **DATHANHTOAN**
TỔNG CỘNG: **60,000 VNĐ**

1, Lấy danh sách hóa đơn của khách hàng

```
SELECT h.*, ptt.NoiDung as TenPTTT, SUM(s.GiaVe) as TongTien
FROM HoaDon h
JOIN PhuongThucThanhToan ptt ON h.idPhuongThucThanhToan =
ptt.idPhuongThucThanhToan
JOIN VeXemPhim v ON h.idHoaDon = v.idHoaDon
JOIN SuatChieu s ON v.idSuatChieu = s.idSuatChieu
WHERE h.idNguoiDung = ?
GROUP BY h.idHoaDon
ORDER BY h.NgayThanhToan DESC
```

Mục đích: Lấy danh sách tất cả hóa đơn của khách hàng kèm tên phương thức thanh toán và tổng tiền

2, Lấy chi tiết vé theo mã hóa đơn

```

SELECT v.TrangThai AS TrangThaiVe, s.ThoiGianBatDau, s.GiaVe,
s.idPhongChieu, p.Ten, g.Hang, g.Cot
FROM VeXemPhim v
JOIN SuatChieu s ON v.idSuatChieu = s.idSuatChieu
JOIN Phim p ON s.idPhim = p.idPhim
JOIN Ghe g ON v.idGhe = g.idGhe
WHERE v.idHoaDon = ?

```

Mục đích: Lấy chi tiết từng vé trong hóa đơn (tên phim, suất chiếu, phòng, giá vé, ghế đã chọn)

3, Lấy danh sách phương thức thanh toán

```
SELECT * FROM PhuongThucThanhToan
```

Mục đích: Hiển thị các phương thức thanh toán có sẵn trong hệ thống

-Thông tin hiển thị trong báo cáo được tổng hợp từ:

- **Thông tin khách hàng:** Lấy từ UserSession (đã đăng nhập)
- **Mã hóa đơn, ngày thanh toán, trạng thái:** Từ bảng HoaDon
- **Tên phim, suất chiếu, phòng:** Từ JOIN VeXemPhim → SuatChieu → Phim
- **Giá vé, số lượng:** Từ bảng SuatChieu và đếm số vé
- **Ghế đã chọn:** Từ bảng Ghe (Hang + Cot)
- **Phương thức thanh toán:** Từ bảng PhuongThucThanhToan
- **Tổng tiền:** Tính bằng SUM(s.GiaVe) theo từng hóa đơn

Tài liệu tham khảo

1. NamThanh102. *BTL_CSDL Repository*. GitHub, 2025.
URL: https://github.com/NamThanh102/BTL_CSDL
2. Giáo trình CSDL: “Database System Concepts” (7th ed.), Silberschatz, Korth, Sudarshan — nền tảng mô hình quan hệ, chuẩn hóa, chỉ mục, giao dịch.
3. Thiết kế CSDL: “Fundamentals of Database Systems” (7th ed.), Elmasri & Navathe — ERD, ràng buộc toàn vẹn, chuẩn hóa 3NF/BCNF.
4. Oracle. *MySQL 8.0 Reference Manual*.
5. Slide bài giảng PTIT