# ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN KHOA HỆ THỐNG THÔNG TIN

# HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

# ĐỒ ÁN QUẢN LÝ RẠP CHIẾU PHIM

Giảng viên hướng dẫn:

Ths. Đỗ Thị Minh Phụng

Sinh viên thực hiện:

Đồng Xuân Danh - 17520321

Lê Minh Thông - 18521456

Trần Vĩnh Phong - 18521241

## LỜI MỞ ĐẦU

Hiện nay, Công nghệ Thông tin được xem là một ngành mũi nhọn của Quốc gia, đặc biệt là các nước phát triển, tiến hành Công nghiệp hóa – Hiện đại hóa như nước ta. Sự bùng nổ thông tin và sự phát triển mạnh mẽ của công nghệ kỹ thuật số, muốn phát triển thì phải áp dụng tin học hóa và tất cả các ngành lĩnh vực.

Do vậy mà trong việc phát triển phần mềm sự đòi hỏi không chỉ là sự chính xác, xử lí được nhiều nghiệp vụ thực tế mà còn phải đáp ứng những nhu cầu khác như về tốc độ, giao diện thân thiện, mô hình hóa được thực tế vào máy tính để người dùng sử dụng tiện lợi hơn, quen thuộc, tính tương thích cao, bảo mật cao,... Các phần mềm giúp tiết kiệm được 1 lượng lớn thời gian, công sức của con người, tăng độ chính xác và hiệu quả trong công việc.

Qua tìm hiểu, khảo sát nhu cầu và đánh giá xu hướng công nghệ hiện nay, nhóm chúng em quyết định xây dựng một hệ thống phần mềm quản lý rạp chiếu phim với mục đích giúp rạp chiếu phim một cách thông minh, hiệu quả, đáng tin cậy, chất lượng và dễ dàng thao tác. Đáp ứng nhu cầu ngày càng tăng trong mảng dịch vụ.

Nhưng bên cạnh việc xây dựng được một chương trình quản lý hoàn thiện còn phụ thuộc vào rất nhiều yếu tố khác nhau. Việc quản lý dự án phần mềm là bước quan trọng đầu tiên trong xây dựng một đề tài.

Mặc dù nhóm chúng em đã có sự cố gắng tìm hiểu trong khi thực hiện đề tài nhưng do điều kiện và thời gian có hạn, cũng như khả năng quản lý dự án có hạn nên trong bài báo cáo này chúng em không tránh khỏi thiếu sót, chúng em rất mong nhận được sự giúp đỡ từ phía các thầy cô và các bạn.

Đặc biệt chúng em xin chân thành cảm ơn cô Đỗ Thị Minh Phụng đã tận tình giúp đỡ chúng em trong quá trình làm đề tài này.

Nhóm thực hiện.

# NHẬN XÉT CỦA GIẢNG VIÊN

 . <b></b>
 . <b></b>
 • • • •
 • • • •
 • • • •
 • • • •
 • • • •
 • • • •
 • • • •

# MŲC LŲC

1
7
7
7
7
8
9
9
9
12
12
12
13
13
14
15
15
15
16
16
17
18
19
20
21
22
23
24
28
28 29
29
29 29

3.2.3.5. In danh sách phim và ca chiếu theo tháng năm	3
3.2.3.6. Thêm chi tiết sản phẩm	3
3.2.3.7. Thêm chi tiết vé	3
3.3 Function	3
3.3.2 Danh sách Function	3
3.3.3. Mô tả một số Funtion	3
3.3.3.1. Doanh số phim	3
3.3.3.2. Doanh số sản phẩm	3
3.3.3.3. Tổng doanh thu Phim.	4
3.3.3.4. Phim có doanh thu cao nhất	4
3.3.3.5. Phim có doanh thu thấp nhất	4
3.3.3.6. Sản phẩm có doanh thu cao nhất	4
3.3.3.7. Sản phẩm có doanh thu thấp nhất	4
3.4. SEQUENCE	4
3.4.2. Danh sách các Sequence	4
3.4.3 Mô tả một số Sequence	4
3.4.3.1 Tạo mã tự động nhân viên	4
3.4.3.2 Tạo mã tự động sản phẩm	4
3.4.3.3 Tạo mã tự động hóa đơn	4
3.4.3.4 Tạo mã tự động khách hàng	4
3.4.3.5 Tạo mã tự động phim	4
3.4.3.6 Tạo mã tự động sự kiện	4
3.4.3.7 Tạo mã tự động khuyến mãi	4
CHƯƠNG 4. XỬ LÝ TRUY XUẤT ĐỒNG THỜI	4
4.1. CÁC MỨC CÔ LẬP TRONG ORACLE	4
4.1.1. Repeatable read	4
4.1.2. Read committed	4
4.1.3. Read uncommitted	4
4.2 LOST UPDATE	5
4.1.4. Serializable	5
4.1.5. Locking reads	
4.3 DIRTY READ	
4.4 Unrepeatable read	
4.5 PHANTOM	
4.6 DEADLOCK	
4.6.1 Conversion Deadlock	
4.6.2 Cycle Deadlock	7
CHĽƠNG 5: THIẾT KẾ GIAO DIÊN	7

Trang menu	76
Trang quản lý bán vé: Đặt phim và thêm lịch chiếu	77
Trang quản lý bán vé 2: Chọn ghế	78
Trang quản lý bán vé 3: Thêm sản phẩm	79
Hóa đơn	80
Xuất vé	81
Quản lý khách hàng (Tra cứu + Xóa khách hàng)	82
Thêm khách hàng	83
Cập nhật khách hàng	84
Quản lý khuyến mãi (Tra cứu + Xóa khuyến mãi)	85
Thêm khuyến mãi	86
Cập nhật khuyến mãi	87
Quản lý sự kiện (Tra cứu + Xóa sự kiện)	88
Thêm sự kiện	89
Cập nhật sự kiện	90
Quản lý sản phẩm (Tra cứu + Xóa sản phẩm)	91
Thêm sản phẩm	92
Cập nhật sản phẩm	93
Quản lý ca chiếu (Tra cứu + Xóa ca chiếu)	92
Thêm ca chiếu: Step 1	95
Thêm ca chiếu: Step 2	96
Cập nhật ca chiếu: Step 1	97
Cập nhật ca chiếu: Step 2	98
Quản lý phim (Tra cứu + Xóa phim)	99
Thêm phim	99
Cập nhật phim	100
Quản lý phòng chiếu (Tra cứu + Xóa phòng chiếu)	101
Thêm phòng chiếu	102
Cập nhật phòng chiếu	103
Quản lý Cơ sở vật chất (Tra cứu + Xóa Cơ sở vật chất)	104
Thêm Cơ sở vật chất	105
Cập nhật Cơ sở vật chất	106
Quản lý Nhân viên (Tra cứu + Xóa Nhân viên)	107
Thêm nhân viên	108
Cập nhật Nhân viên	109
CHƯƠNG 6: KẾT LUẬN	110
6.1 Bảng Phân công việc nhóm	110
6.2 Môi trường phát triển ứng dụng:	110
6.3 Kết quả đạt được	111
6 4 Han chế	111

# Đề tài: QUẢN LÝ RẠP CHIẾU PHIM

6.5	Hướng phát triển	. 111
ΓÀΙΙΙÊΙΙ	ΤΗΔΜ ΚΗΔΟ	112

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI DỰ ÁN

# 1.1 Đặt vấn đề

Công nghệ thông tin đã được ứng dụng nhằm xây dựng và tổ chức nên các hệ thống này một cách khoa học, chặt chẽ, giúp cho việc quản lý tránh được những sai sót không đáng có và những tối thiểu những khả năng thất thoát dữ liệu thông tin, từ đó giúp việc kinh doanh và hoạt động của công ty, doanh nhiệp có thể trở nên suôn sẻ, tăng năng suất và giảm thiểu chi phí về công tác quản lí, cũng như tăng tính tiện dụng cho người dùng, giúp họ có trải nghiệm tốt hơn.

Trong lĩnh vực giải trí, đây là lĩnh vực có khá nhiều sự canh tranh cũng như có nhu cầu phát triển khá cao. Gắn liền với ngành giải trí các hệ thống rạp chiếu phim ngày càng nhiều phụ vụ nhu cầu giải trí của khách hàng ngày càng cao và càng có nhiều bất cập hơn chính vì thế việc tạo ra hệ thống có thể tương tác giữa khách hàng và rạp chiếu phim là việc cần thiết giúp phục vụ và chăm sóc khách hàng dễ dàng hơn.

Trong phần nội dung nghiên cứu của đồ án thì chỉ tập trung bó hẹp ở việc quản lý các thông tin cơ bản của phần mềm quản lý hệ thông. Các công tác quản lý được tin học hóa với các chức năng cơ bản như quản lý phim, quản lý rạp, quản lý ca chiếu, quản lý khách hàng,.... Tuy nhiên, hệ thống vẫn chưa giải quyết được vấn đề bán vé thủ công. Đối với các chức năng khác của phần mềm thì tuỳ yêu cầu và môi trường phát triển của khách hàng mà thay đổi và phát triển cho phù hợp.

# 1.2 Tổng quan về đồ án

- Tên dự án: Quản lý rạp chiếu phim
- ❖ Thành viên quản lý:
  - 1. Đồng Xuân Danh
  - 2. Lê Minh Thông
  - 3. Trần Vĩnh Phong

Thời gian thực hiện:

Bắt đầu: ngày 25/03/2020
 Kết thúc: ngày 06/07/2020

Mục đích dự án: Xây dựng được phần mềm quản lý rạp chiếu phim thiết thực và mang lại hiệu quả.

# 1.3 Các công nghệ sử dụng:

- Hệ quản trị CSDL: Oracle.
- Ngôn ngữ lập trình Java trên phần mềm NetBean IDE.

- JDBC.

# 1.4 Bố cục báo cáo:

Chương 1: Tổng quan đề tài.

Chương 2: Xác định và phân tích yêu cầu.

Chương 3: Xây dựng và quản lý giao tác.

Chương 4: Xử lý truy xuất đồng thời.

Chương 5: Thiết kế giao diện.

Chương 6: Kết luận.

# CHƯƠNG 2: XÁC ĐỊNH VÀ PHÂN TÍCH YÊU CẦU

# 2.1 Khảo sát hiện trạng

	2.1.1 Phỏng vân Kế hoạch phỏng vấn tổng quan					
Hệ thống: Rạp chiếu phim HD Cinema						
Ngườ	Người lập: Ngày lập: 04/03/2019					
STT	Chủ đề	Yêu cầu	Ngày BĐ	Ngày KT		
1	Quy trình quản lí phim	Nắm rõ quy trình quản lí phim, phân suất chiếu, phòng chiếu và định giá vé.	04/04/2020	04/04/2020		
2	Quy trình quản lý khách hàng	Nắm rõ quy trình quản lí thông tin khách hàng.	04/04/2020	04/04/2020		
3	Quy trình quản lí các phòng chiếu	Nắm rõ quy tắc đối với phòng chiếu, loại phòng chiếu, loại ghế tương ứng mức giá.	04/04/2020	04/04/2020		
4	Quy trình thanh toán	nh Nắm rõ quy trình đặt vé, và 04/04/2020 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04/04/2000 04		04/04/2020		
5	Quy trình báo cáo, thống kê	Dựa vào dữ liệu hệ thống để báo cáo số lượng vé và sản phẩm bán được.	04/04/2020	04/04/2020		

# Bảng kế hoạch phỏng vấn tổng quan

Bảng kế hoạch phỏng vấn Hệ thống: Rạp chiếu phim			
Người được phỏng vấn: Trần Văn Thanh Phân tích viên: Lê Minh Thông			
Vị trí/ phương tiện: Văn phòng đại diện rạp chiếu phim	Thời gian bắt đầu: từ 8g00 ngày 04/04/2020 Thời gian kết thúc: 11g30 ngày 04/04/2020		
Mục tiêu: Thu thập và hiểu được quy trình hoạt động của rạp chiếu phim			
Chi tiết buổi phỏng vấn:	Thời gian ước lượng:		

<ul> <li>Giới thiệu</li> <li>Tổng quan về hệ thống</li> <li>Tổng quan về buổi phỏng vấn</li></ul>		uản lý phim luản lý khách hàng luản lý các phòng hanh toán		
Quan	sát tổng quan			
Phát s	sinh ngoài dự kiến			
Hệ th	<b>Bảng c</b> ống: Rạp chiếu phim	âu hỏi và ghi nhận câu trả lời		
	i được phỏng vấn: Văn Thanh	Ngày: 04/04/2020		
Câu h	női	Ghi nhận trả lời		
Chủ đề 01	Câu hỏi 1: Giá vé được phân hạng như thế nào?			
thường khoảng cách giữa suất chiếu phim A với phim B là bao nhiêu?		<ul> <li>Chúng tôi sẽ cộng thêm 30 phút (thời gian dọn dẹp phòng chiếu)</li> </ul>		
		Thông thường chúng tôi chia suất chiếu theo vào độ mới của phim, theo thời lượng của phim		
	Câu hỏi 4: Phim có bao nhiều loại suất chiếu?	Hiện tại chúng tôi cung cấp 2 loại phim là 2D, 3D.		

Chủ đề 02	Câu hỏi 1: Khách hàng muốn đăng kí làm thành viên thì phải làm sao và cần những gì?	-Khách hàng phải điền đầy đủ các thông tin được cung cấp trên form.
	Câu hỏi 2: Có bao nhiêu loại khách hàng và ưu đãi họ sẽ nhận được là gì?	+Có thể sử dụng điểm tại các LotteMart +Ưu đãi thả ga và quà tặng balo vào ngày thành viên thứ 3.
	Câu hỏi 3: Khách hàng được tích điểm thông qua hình thức nào?	Khách hàng sẽ được tích điểm khi mua vé hoặc bắp nước tại quầy, điểm tích lũy sẽ được cộng tương ứng với số tiền quy đổi ra theo quy định của rạp. Cụ thể 1000 đồng thì khách hàng sẽ được cộng 1 điểm
	Câu hỏi 4: Khi khách hàng thành viên lâu quá không sử dụng thẻ thì thông tin có còn được lưu trữ hay không?	Khách hàng là yếu tố quyết định đến hoạt động của rạp. Vì vậy việc lưu trữ thông tin khách hàng là cần thiết cho dù khách hàng có thường xuyên sử dụng thẻ hay không.
Chủ đề 03	Câu hỏi 1: Các phòng chiếu được phân chia như thế nào?	Các phòng chiếu được sắp xếp bởi người quản lý suất chiếu.
	Câu hỏi 2: Các phòng chiếu có bao nhiều loại ghế và giá của chúng có giống nhau không?	Tuy vào phòng chiếu có nhiều loại ghế khác nhau ví dụ ghế thường (75000), ghế VIP (90000) chúng tôi có thể tùy chỉnh lại ghế cũng như thêm ghế mới.
	Câu hỏi 3: Phòng chiếu có số lượng ghế và cách bố trí có giống nhau hay không?	Rạp có chúng tôi thì có số lượng ghế và cách sắp xếp ở các phòng là khác nhau nhưng các rạp khác thì có thể sắp xếp giống nhau.
Chủ đề 04	Câu hỏi 1: Khi mua vé tại quầy, khách hàng có thể trả tiền bằng hình thức khác hay thay vì là thanh toán trực tiếp với nhân viên bán vé?	Bên chúng tôi có hỗ trợ thanh toán trực tiếp qua ví điện tử như momo, zalopay hoặc bằng thẻ ngân hàng.

	Câu hỏi 2: Khách hàng có thể lấy điểm tích lũy quy thành vé xem phim được không?	Khách hàng có thể đổi đổi tích lũy thành vé xem phim với điều kiện số điểm tích lũy đủ để đổi.
	Câu hỏi 3: Coupon, discount có được đổi sang tiền mặt không?	Coupon và discount không có giá trị đổi thành tiền mặt.
Chủ đề 05	Câu hỏi 1: Rạp cần thống kế những gì?	-Gồm: thống kê vé bán được theo từng sản phẩm, thống kê doanh thu sản phẩm theo thời gian nhất định.
	Câu hỏi 2: Thời gian lập thống kê bán vé/ bắp nước định kỳ là bao lâu?	Thông thường bên chúng tôi sẽ thống kê từng ngày ở từng rạp riêng biệt. Và bắp nước cũng được thống kê theo ngày.

Bảng câu hỏi và ghi nhận câu trả lời

# 2.2 Phân tích yêu cầu

# 2.2.1 Yêu cầu chức năng

- a) Yêu cầu lưu trữ:
- Lưu trữ thông tin luôn là một yêu cầu quan trọng và quan trọng nhất của hầu hết tất cả hệ thống.
- Là cơ sở để thực hiện phần lớn các tính năng của hệ thống.
- b) Yêu cầu về tính năng: Ngoài việc lưu trữ dữ liệu, hệ thống phải đáp ứng được các yêu cầu của khách hàng.
- Chức năng truy vấn.
- Tính bảo mật và phân quyền người sử dụng.
- Chức năng thống kê, báo cáo: giúp nhân viên quản lý theo dõi tình hình chung của rạp chiếu phim.

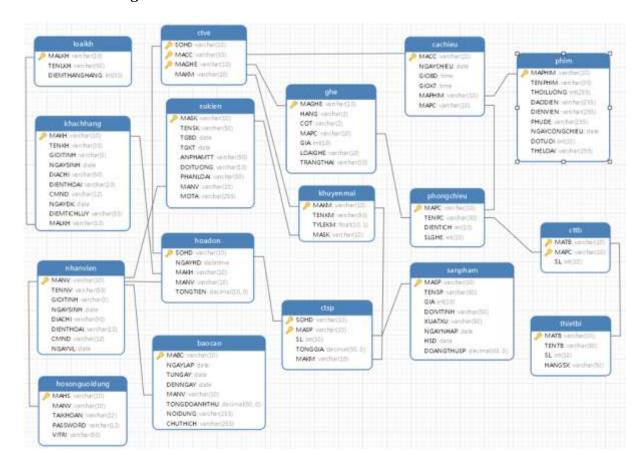
# 2.2.2 Yêu cầu phi chức năng

- Yêu cầu về chất lượng:
  - Tính cập nhật phát triển.
  - Tính tiện dụng.
  - Tính tương thích.
- ❖ Yêu cầu về giao diện:

- Ua nhìn.
- Người dùng dễ sử dụng.
- Hiển thị rõ ràng và chi tiết các thông tin.

# 2.3. Thiết kế cơ sở dữ liệu

# 2.3.1. Sơ đồ Logic



Hình 1. Mô hình dữ liệu quan hệ

#### 2.3.2. Mô hình quan hệ

KHACHHANG (MAKH, TENKH, GIOITINH, NGAYSINH, DIACHI, DIENTHOAI, EMAIL, CMND, NGAYDK, DIEMTICHLUY, MALKH, MANV)

LOAIKH (MALKH, TENLKA, DIEM\_TOITHIEU)

NHANVIEN (<u>MANV</u>, TENNV, GIOITINH, NGAYSINH, DIACHI, DIENTHOAI, EMAIL, CMND, NGAYVL, TONGTIENBAN)

PHIM (MAPHIM, TENPHIM, THOILUONG, DAODIEN, DIENVIEN, HINHANH, NAMSX, PHUDE, NGAYCONGCHIEU, DOTUOIQUYDINH, THELOAI, NGON NGU, DOANHTHUPHIM)

THIETBI (MATB, TENTB, SL, HANGSX)

CTTHIETBI (MATB, MAP, SL)

PHONGCHIEU (MAPC, TENPC, SOGHE, DIENTICH)

HOADON (SOHD, MANV, MAKH, NGHD, TONGTRIGIA)

CACHIEU (MACC, NGAYCHIEU, MAPHIM, GIOBD, GIOKT, MAPC)

GHE (MAGHE, HANG, COT, GIA, LOAIGHE, MAPC)

CTVE (SOHD, MAGHE, MACC, MAKM, THANHTIEN)

SANPHAM (MASP, TENSP, GIA, XUATXU, NGNHAP, DOANHTHUSP)

CTSP (SOHD, MASP, SL, MAKM, THANHTIEN)

SUKIEN (<u>MASK</u>, TENSK, TGBD, TGKT, ANPHAMTT, DOITUONG, PHANLOAI, MOTASK, <u>MANV</u>)

KHUYENMAI (MAKM, TENKM, TYLEKM, MASK)

HOSONGUOIDUNG (MAHS, MANV, TAIKHOAN, PASSWORD, VITRI)

BAOCAO (<u>MABC</u>, NGLAP, TUNGAY, DENNGAY, NOIDUNG, <u>MANV</u>, TONGTIEN)

# CHƯƠNG 3. XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC

# 3.1. Trigger

# 3.1.1. Danh sách các Trigger

STT	TÊN	THAO TÁC	BÅNG	NỘI DUNG
1	HD	Before insert	HoaDon	Mặc định tổng tiền ban đầu của hóa đơn bằng 0.
2	SP	Before insert	SANPHAM	Thứ tự sản phẩm sẽ tự động tăng khi thêm sản phẩm mới
3	INS_CTSP	After insert	CTSP	Cập nhật tổng tiền của hóa đơn. Cập nhật điểm tích lũy của khách hàng Cập nhật lại doanh số sản phẩm
4	INS_CTVE	After insert	CTVE	Cập nhật tổng tiền của hóa đơn. Cập nhật điểm tích lũy của khách hàng Cập nhật doanh thu phim.
5	INS_KH	Before insert	KHACHHANG	Thứ tự khách hàng tăng theo thứ tự khi thêm mới khách hàng
6	MANV_NHANVIEN	Before insert	NHANVIEN	Thứ tự nhân viên tăng khi thêm mới nhân viên
7	MALKH_LOAIKH	Before insert	LOAIKH	Thứ tự loai khách hàng tăng khi thêm mới loại khách hàng
8	BC	Before insert	BAOCAO	Xét tổng doanh thu của hóa đơn trong một khoảng thời gian nhất định tùy thuộc

				vào yêu cầu của nhân viên.
9	MAKM_KHUYENMAI	Before insert	KHUYENMAI	Thứ tự tăng khi thêm mới loại khuyến mãi
10	MACC_CACHIEU	Before insert	CACHIEU	Thứ tự tăng khi thêm mới ca chiếu
11	MAPC_PHONGCHIEU	Before insert	PHONGCHIEU	Thứ tự tăng khi thêm mới phòng chiếu
12	MAPHIM_PHIM	Before insert	PHIM	Thứ tự tăng khi thêm mới phim
13	MASK_SUKIEN	Before insert	SUKIEN	Thứ tự tăng khi thêm mới sự kiện
14	MATB_THIETBI	Before insert	THIETBI	Thứ tự tăng khi thêm mới thiết bị

# 3.1.2. Mô tả một số Trigger

## 3.1.2.1. Thêm vào bảng HOADON

Tên: HD

Thao tác: Before insertTrên bảng: HOADON

- Nội dung: Khi thêm một hóa đơn thì mặc định tổng tiền ban đầu của hóa đơn bằng 0 và mặc định ngày xuất hóa đơn là ngày hiện tại trên hệ thống.
- Mã SQL:
  - create or replace TRIGGER HD
  - BEFORE INSERT ON HOADON
  - FOR EACH ROW
  - declare idhd int;
  - BEGIN
  - :NEW.TONGTIEN := 0;
  - :NEW.NGAYHD := SYSDATE;
  - select HD.NEXTVAL INTO idhd FROM DUAL;
  - END;
- Bước thực hiên:

- [1] Gán tổng tiền ban đầu bằng 0.
- [2] Gán ngày xuất hóa đơn là ngày hiện tại trên hệ thống
- [3] Kiểm tra số hóa đơn tiếp theo:
- [3.1] Nếu số hóa đơn bé hơn 10 thì số hóa đơn bằng HD0 cộng số hóa đơn
  - [3.2] Ngược lại, số hóa đơn bằng HD cộng số hóa đơn.

#### 3.1.2.2. Thêm vào bảng SANPHAM

- Tên: SP
- Thao tác: Before insert
- Trên bảng: SANPHAM
- Nội dung: Thứ tự sản phẩm sẽ tự động tăng khi thêm sản phẩm mới
- Mã SQL:
  - create or replace TRIGGER SP
  - BEFORE INSERT ON SANPHAM
  - FOR EACH ROW
  - WHEN(new.MASP is null)
  - BEGIN
  - select SP.NEXTVAL
  - INTO :new.MASP
  - FROM DUAL;
  - \_
  - END;
- Bước thực hiện:
  - [1] Kiểm tra mã sản phẩm tiếp theo:

#### 3.1.2.3. Thêm vào bảng CTSP

- Tên: INS\_CTSP
- Thao tác: After insert
- Trên bảng: CTSP, HOADON, KHACHHANG
- Nội dung: Khi cập nhật lại thông tin trong bảng HOADON tương ứng với SOHD, thì TONGTIEN được cộng dồn. Khi thanh toán xong, DIEMTICHLUY của khách hàng sẽ tăng lên, khi DIEMTICHLUY đạt đến DIEMTHANGHANG thì LOAIKH sẽ thay đổi.
- Mã SQL:
  - create or replace TRIGGER INS\_CTSP
  - AFTER INSERT ON CTSP
  - FOR EACH ROW
  - BEGIN
  - UPDATE HOADON
  - SET TONGTIEN = TONGTIEN + :NEW.THANHTIEN
  - WHERE SOHD=:NEW.SOHD;
  - UPDATE SANPHAM
  - SET DOANHTHU = DOANHTHU + :NEW.THANHTIEN,
  - SL = SL +:NEW.SL
  - WHERE MASP = :NEW.MASP;
  - END;
- Bước thực hiện:
  - [1] Khi thêm CTSP mới thì cập nhật lại bảng HOADON: TONGTIEN = TONGTIEN + :NEW.THANHTIEN tương ứng với SOHD.
  - SL = SL + : NEW.SL

3.1.2.4. Thêm vào bảng CTVE

- Tên: INS\_CTVE
- Thao tác: After insert
- Trên bảng: CTVE, HOADON, KHACHHANG, PHIM
- Nội dung: Khi cập nhật lại thông tin trong bảng HOADON tương ứng với SOHD, thì TONGTIEN được cộng dồn DOANHTHU cũng được cộng dồn.
- Mã SQL:
  - create or replace TRIGGER INS\_CTVE
  - AFTER INSERT ON CTVE
  - FOR EACH ROW
  - BEGIN
  - UPDATE HOADON
  - SET TONGTIEN = TONGTIEN +: NEW. THANHTIEN
  - WHERE SOHD=:NEW.SOHD;

\_

- UPDATE PHIM
- SET DOANHTHU = DOANHTHU + :NEW.THANHTIEN
- WHERE MAPHIM IN (SELECT MAPHIM FROM CACHIEU WHERE MACC = :NEW.MACC);
- END:
- Bước thực hiện:
  - [1] Khi thêm CTVE mới thì cập nhật lại bảng HOADON: TONGTIEN = TONGTIEN + :NEW.THANHTIEN tương ứng với SOHD.
  - [2] Tương tự, DOANHTHU được cộng dồn.

#### 3.1.2.5. Thêm Khách hàng mơi vao bản KHACHHANG

```
Tên: INS_KHThao tác: Before insert
```

- Trên bảng: KHACHHANG

- Nội dung: Thứ tự khách hàng tăng theo thứ tự khi thêm mới khách hàng Mã SQL:
  - create or replace TRIGGER INS\_KH
  - BEFORE INSERT ON KHACHHANG
  - FOR EACH ROW
  - declare idkh int;
  - BEGIN

```
- :NEW.DIEMTICHLUY :=0;- :NEW.MALKH := 'LKH01';
```

select KH.NEXTVAL INTO idkh FROM DUAL;

- end;

- Bước thực hiện:
  - [1] Kiểm tra ma khách hàng và tăng thứ tự và gán giá trị DIEMTICHLUY=0;

#### 3.1.2.6. Thêm nhân viên mới vao bản NHANVIEN

- Tên: MANV\_NHANVIEN
- Thao tác: Before insert
- Trên bảng: NHANVIEN
- Nội dungThứ tự nhân viên tăng khi thêm mới nhân viên
- Mã SQL:
  - create or replace TRIGGER manv\_nhanvien
  - BEFORE INSERT on nhanvien
  - FOR EACH ROW
  - WHEN (new.MANV IS NULL)
  - BEGIN
  - SELECT NV.NEXTVAL
  - INTO :new.MANV
  - FROM dual;
  - END;
- Bước thực hiện:
  - [1] Kiểm tra ma nhan viên và tự động tăng khi thêm mới nhân viên

#### 3.1.2.7. Thêm mới loại khách hàng vào bản LOAIKH

- Tên: MALKH\_LOAIKH
- Thao tác: Before insert
- Trên bảng: LOAIKH
- Nội dung: Kiểm tra xem loai khách hàng và tự động tăng khi thêm mới loai khách háng
- Mã SQL:
  - create or replace TRIGGER manv\_nhanvien
  - BEFORE INSERT on nhanvien
  - FOR EACH ROW
  - WHEN (new.MANV IS NULL)
  - BEGIN
  - SELECT NV.NEXTVAL
  - INTO :new.MANV
  - FROM dual;
  - END;
- Bước thực hiện:
  - [1] Kiểm tra loại khách hàng và tự động tăng khi thêm mới loại khách hàng

#### 3.1.2.8. Thêm vào bảng BAOCAO

- Tên: BC

Thao tác: Before insertTrên bảng: BAOCAO

 Nội dung: Tính TONGDOANHTHU để báo cáo trong khoảng thời gian nhất đinh.

- Mã SQL:

CREATE or replace TRIGGER BC
BEFORE INSERT ON BAOCAO
FOR EACH ROW
DECLARE TONG number;
BEGIN
SELECT SUM(TONGTIEN) INTO TONG FROM HOADON
WHERE NGAYHD BETWEEN :NEW.TUNGAY AND
:NEW.DENNGAY;
:NEW.TONGDOANHTHU := TONG;
END;

- Bước thực hiên:
  - [1] Khai báo biến TONG
  - [2] Trong bảng HOADON, tính tổng TONGTIEN để gán vào biến TONG trong khoảng thời gian nhất định (TUNGAY DENNGAY)
  - [3] Gán TONGDOANHTHU = TONG

#### 3.1.2.9. Thêm mới khuyến mãi vao bản KHUYENMAI

- Tên: MAKM\_KHUYENMAI
- Thao tác: Before insert
- Trên bảng: KHUYENMAI
- Nội dung: Tự động tăng thư tự khi thêm khuyến mãi
- Mã SQL:
  - create or replace TRIGGER makm\_khuyenmai
  - BEFORE INSERT on khuyenmai
  - FOR EACH ROW
  - WHEN (new.MAKM IS NULL)
  - BEGIN
  - SELECT KM.NEXTVAL
  - INTO :new.MAKM
  - FROM dual:
  - END;
- Bước thực hiện:
  - [1] Kiêm tra khuyến mãi và tự động tăng khi thêm mới khuyến mãi

#### 3.1.2.10 Thếm mới ca chiếu vào bản CACHIEU

- Tên: MACC CACHIEU
- Thao tác: Before insert
- Trên bảng: CACHIEU
- Nội dung: Tự động tăng thư tự khi thêm mới ca chiếu
- Mã SQL:
  - create or replace TRIGGER macc\_cachieu
  - BEFORE INSERT on cachieu
  - FOR EACH ROW
  - WHEN (new.MACC IS NULL)
  - BEGIN
  - SELECT macc.NEXTVAL
  - INTO :new.MACC
  - FROM dual;
  - END;
- Bước thực hiện:
  - [1] Kiêm tra ca chieu và tự động tăng khi thêm mới ca chiếu

3.1.2.11 Thếm mới phòng chiếu vào bản PHONGCHIEU

- Tên: MAPC PHONGCHIEU
- Thao tác: Before insert
- Trên bảng: PHONGCHIEU
- Nội dung: Tự động tăng thư tự khi thêm mới phòng chiếu
- Mã SQL:
  - create or replace TRIGGER mapc\_phongchieu
  - BEFORE INSERT on phongchieu
  - FOR EACH ROW
  - WHEN (new.MAPC IS NULL)
  - BEGIN
  - SELECT mapc.NEXTVAL
  - INTO :new.MAPC
  - FROM dual;
  - END;
- Bước thực hiên:
  - [1] Kiêm tra phong chieu và tự động tăng khi thêm mới phòng chiếu

# 3.1.2.12 Thếm mới phim vao bản PHIM

- Tên: MAPHIM\_PHIM
- Thao tác: Before insert
- Trên bảng: PHIM
- Nội dung: Tự động tăng thư tự khi thêm mới phim
- Mã SQL:
- create or replace TRIGGER maphim\_phim
- BEFORE INSERT on phim
- FOR EACH ROW
- WHEN (new.MAPHIM IS NULL)
- BEGIN
- SELECT PH.NEXTVAL
- INTO :new.MAPHIM
- FROM dual:
- END;
- Bước thực hiện:

[1] Kiêm tra phim và tự động tăng khi thêm mới phim

## 3.1.2.13 Thếm mới sự kiện vao bản SUKIEN

- Tên: MASK SUKIEN

Thao tác: Before insertTrên bảng: SUKIEN

- Nội dung: Tự động tăng thư tự khi thêm mới sự kiện

– Mã SQL:

create or replace TRIGGER mask\_sukien

BEFORE INSERT on sukien

FOR EACH ROW

WHEN (new.MASK IS NULL)

**BEGIN** 

SELECT SK.NEXTVAL

INTO :new.MASK

FROM dual;

END;

- Bước thực hiện:
  - [1] Kiêm tra sự kiện và tự động tăng khi thêm mới sự kiện

# 3.1.2.14 Thêm mới thiết bị vào bản THIETBI

Tên: MASK\_SUKIEN

Thao tác: Before insert

- Trên bảng: SUKIEN

- Nội dung: Tự động tăng thư tự khi thêm mới sự kiện

- Mã SQL:

create or replace TRIGGER matb\_thietbi

- BEFORE INSERT on thietbi
- FOR EACH ROW
- WHEN (new.MATB IS NULL)
- BEGIN
- SELECT matb.NEXTVAL
- INTO :new.MATB
- FROM dual;
- END;
- Bước thực hiện:
  - [1] Kiêm tra thiet bi và tự động tăng khi thêm mới thiết bị

# 3.2 Stored Procedure

# 3.2.3. Danh sách Stored procedure

S T T	Tên	Input Parameter	Output Parameter	Ý nghĩa	Ghi chú
1	TOP1PHIM	NAM		In ra top 10 bộ phim có doanh số cao nhất trong năm nào đó	
2	DSPHIM	MAKH1		In ra những bộ phim	
		NAM		khách hàng đã xem trong năm truyền vào.	
3	DSOPHIM	NAM		In ra doanh số của từng phim trong năm đó.	
4	DSOSP	NAM		In ra tất cả các sản phẩm cùng với số lượng đã bán ra trong năm đó.	
5	LICHCHIEU	THANG NAM		In ra bộ phim cùng với những ca chiếu của bộ phim trong tháng / trong năm đó	
6	INS_CTSP	SOHD		Thêm Chi tiết sản phẩm	
		MASP			
		SL			
		MAKM			
7	INS_CTVE	SOHD		Thêm Chi tiết vé	
		MACC			
		MAGHE			
		MAKM			

## 3.2.3 Mô tả một số Stored procedure

# 3.2.3.1. In danh sách phim top1 doanh thu cao nhất trong năm

- Tên: TOP1PHIM
- Nội dung: In ra những bộ phim top1 doanh thu cao nhất trong năm truyền vào.
- Tham số đầu vào: NAM INT
- Tham số đầu ra: (không có)
- Mã SQL:

```
CREATE OR REPLACE PROCEDURE TOP1PHIM (NAM INT)
AS MAPHIM1 PHIM.MAPHIM%TYPE;
TENPHIM1 PHIM. TENPHIM% TYPE;
CURSOR TOP1 (NAM INT) IS
SELECT DISTINCT PHIM. MAPHIM, TENPHIM
FROM HOADON INNER JOIN (CTVE INNER JOIN (CACHIEU INNER JOIN
PHIM ON CACHIEU. MAPHIM = PHIM. MAPHIM) ON
CTVE.MACC=CACHIEU.MACC) ON HOADON.SOHD = CTVE.SOHD
WHERE (extract (year from NGAYHD)) = NAM
GROUP BY PHIM. MAPHIM, TENPHIM
ORDER BY SUM (THANHTIEN) DESC
FETCH FIRST 1 ROWS ONLY;
BEGIN
  dbms output.put line ('TOP 1 BO PHIM CO DOANH SO CAO NHAT
TRONG NAM ' || NAM || ':');
OPEN TOP1 (NAM);
LOOP
fetch TOP1 into MAPHIM1, TENPHIM1;
 exit when TOP1%notfound;
  dbms_output.put_line (MAPHIM1 || ' ' || TENPHIM1);
end loop;
close TOP1;
END;
```

#### 3.2.3.2. In danh sách phim kèm doanh số theo năm

Tên: DSOPHIM

- Nội dung: In ra những bộ phim kèm doanh số theo năm truyền vào.

Tham số đầu vào: NAM INT
Tham số đầu ra: (không có)

- Mã SQL:

```
CREATE OR REPLACE PROCEDURE DSOPHIM (NAM INT)
AS MAPHIM1 PHIM.MAPHIM%TYPE;
TENPHIM1 PHIM.TENPHIM%TYPE;
DOANHSO NUMBER;
CURSOR DSO (NAM INT) IS
SELECT DISTINCT PHIM. MAPHIM, TENPHIM, SUM (THANHTIEN)
FROM CTVE INNER JOIN (CACHIEU INNER JOIN PHIM ON
CACHIEU.MAPHIM = PHIM.MAPHIM) ON CTVE.MACC=CACHIEU.MACC
WHERE (extract (year from NGAYCONGCHIEU)) = NAM
GROUP BY PHIM. MAPHIM, TENPHIM;
BEGIN
OPEN DSO (NAM);
LOOP
fetch DSO into MAPHIM1, TENPHIM1, DOANHSO;
  exit when DSO%notfound;
 dbms output.put line (MAPHIM1 || ' ' || TENPHIM1|| ' '
|| DOANHSO);
end loop;
close DSO;
END;
```

#### 3.2.3.3. In danh sách phim kèm doanh số theo năm

Tên: DSOPHIM

- Nội dung: In ra những bộ phim kèm doanh số theo năm truyền vào.

Tham số đầu vào: NAM INT
Tham số đầu ra: (không có)

- Mã SQL:

```
CREATE OR REPLACE PROCEDURE DSOPHIM (NAM INT)
AS MAPHIM1 PHIM.MAPHIM%TYPE;
TENPHIM1 PHIM.TENPHIM%TYPE;
DOANHSO NUMBER;
CURSOR DSO (NAM INT) IS
SELECT DISTINCT PHIM. MAPHIM, TENPHIM, SUM (THANHTIEN)
FROM CTVE INNER JOIN (CACHIEU INNER JOIN PHIM ON
CACHIEU.MAPHIM = PHIM.MAPHIM) ON CTVE.MACC=CACHIEU.MACC
WHERE (extract (year from NGAYCONGCHIEU)) = NAM
GROUP BY PHIM. MAPHIM, TENPHIM;
BEGIN
OPEN DSO (NAM);
LOOP
fetch DSO into MAPHIM1, TENPHIM1, DOANHSO;
  exit when DSO%notfound;
 dbms output.put line (MAPHIM1 || ' ' || TENPHIM1|| ' '
|| DOANHSO);
end loop;
close DSO;
END;
```

#### 3.2.3.4. In danh sách sản phẩm kèm doanh số theo năm

- Tên: DSOSP

- Nội dung: In ra những sản phẩm kèm doanh số theo năm truyền vào.

Tham số đầu vào: NAM INT
Tham số đầu ra: (không có)

- Mã SQL:

```
CREATE OR REPLACE PROCEDURE DSOSP (NAM INT)
IS MASP1 SANPHAM.MASP%TYPE;
TENSP1 SANPHAM.TENSP%TYPE;
SL INT;
CURSOR SLBAN (NAM INT) IS
SELECT DISTINCT SANPHAM.MASP, TENSP, SUM(SL)
FROM HOADON INNER JOIN (CTSP INNER JOIN SANPHAM ON
CTSP.MASP=SANPHAM.MASP) ON HOADON.SOHD=CTSP.SOHD
WHERE (extract (year from NGAYHD)) = NAM
GROUP BY SANPHAM. MASP, TENSP;
BEGIN
OPEN SLBAN (NAM);
LOOP
fetch SLBAN into MASP1, TENSP1, SL;
 exit when SLBAN%notfound;
 dbms output.put line (MASP1 || ' ' || TENSP1 || ' ' ||
SL);
end loop;
close SLBAN;
END;
```

#### 3.2.3.5. In danh sách phim và ca chiếu theo tháng năm

- Tên: LICHCHIEU
- Nội dung: In ra danh sách phim và ca chiếu theo tháng năm truyền vào.
- Tham số đầu vào: THANG INT, NAM INT
- Tham số đầu ra: (không có)
- Mã SQL:

```
CREATE OR REPLACE PROCEDURE LICHCHIEU (THANG INT, NAM INT)
AS MAPHIM1 PHIM.MAPHIM%TYPE;
TENPHIM1 PHIM.TENPHIM%TYPE;
CC CACHIEU%ROWTYPE;
CURSOR CCHIEU (MAPHIM1 PHIM.MAPHIM%TYPE) IS
SELECT NGAYCHIEU, GIOBD, GIOKT FROM CACHIEU WHERE MAPHIM =
MAPHIM1 AND (extract (year from NGAYCHIEU)) = NAM AND
(extract (MONTH from NGAYCHIEU)) = THANG;
CURSOR DSPHIM (THANG INT, NAM INT) IS
SELECT MAPHIM, TENPHIM FROM PHIM WHERE (extract (year from
NGAYCONGCHIEU)) = NAM AND (extract (MONTH from
NGAYCONGCHIEU)) = THANG;
open DSPHIM (THANG, NAM);
       loop
       fetch DSPHIM into MAPHIM1, TENPHIM1;
       exit when DSPHIM%notfound;
       dbms output.put_line (' - ' || MAPHIM1 || ' ' ||
TENPHIM1 || ':');
       open CCHIEU (MAPHIM1);
       loop
       fetch CCHIEU into CC.NGAYCHIEU, CC.GIOBD, CC.GIOKT;
       exit when CCHIEU%notfound;
       CC.GIOBD || ' ' || CC.GIOKT);
       end loop;
       close CCHIEU;
       end loop;
       close DSPHIM;
END;
```

#### 3.2.3.6. Thêm chi tiết sản phẩm

- Tên: INS\_CTSP
- Nội dung: In ra danh sách phim và ca chiếu theo tháng năm truyền vào.
- Tham số đầu vào: SOHD, MASP, SL, MAKM
- Tham số đầu ra: (không có)
- Mã SQL:
- create or replace PROCEDURE INS\_CTSP (SOHD1 hoadon.sohd%type, MASP1 sanpham.masp%type, SL1 sanpham.sl%type, MAKM1 khuyenmai.makm%type)
- IS GIA1 NUMBER;
- TYLEKM1 FLOAT;
- THANHTIEN1 NUMBER;
- BEGIN
- SELECT GIA INTO GIA1 FROM SANPHAM WHERE MASP = MASP1;
- IF (MAKM1 IS NULL)
- THEN
- THANHTIEN1 := SL1 \* GIA1;
- ELSE
- BEGIN
- SELECT TYLEKM INTO TYLEKM1 FROM KHUYENMAI
   WHERE MAKM = MAKM1;
- THANHTIEN1 := SL1\*GIA1\*(1-TYLEKM1);
- END;
- END IF;
- INSERT INTO CTSP VALUES(SOHD1,MASP1,SL1,MAKM1,THANHTIEN1);
- END:

Giảng viên hướng dẫn: Đỗ Thị Minh Phụng

#### 3.2.3.7. Thêm chi tiết vé

- Tên: INS\_CTVE
- Nội dung: In ra danh sách phim và ca chiếu theo tháng năm truyền vào.
- Tham số đầu vào: SOHD, MACC, MAGHE, MAKM
- Tham số đầu ra: (không có)
- Mã SQL:

\_

- CREATE OR REPLACE PROCEDURE INS\_CTVE (SOHD1 hoadon.sohd%type, MASP1 sanpham.masp%type, SL1 sanpham.sl%type, MAKM1 khuyenmai.makm%type)
- IS GIA1 NUMBER;
- TYLEKM1 FLOAT;
- THANHTIEN1 NUMBER;
- MAGHE1 ghe.maghe%type;
- MACC1 cachieu.macc%type;
- BEGIN
- SELECT GIA INTO GIA1 FROM GHE WHERE MAGHE = MAGHE1;
- IF (MAKM1 IS NULL)
- THEN
- THANHTIEN1 := GIA1;
- ELSE
- BEGIN
- SELECT TYLEKM INTO TYLEKM1 FROM KHUYENMAI
   WHERE MAKM = MAKM1;
- THANHTIEN1 := GIA1 \* (1-TYLEKM1);
- END:
- END IF;
- INSERT INTO CTVE VALUES (SOHD1,MACC1,MAGHE1,MAKM1,THANHTIEN1);
- END;

## 3.3 Function

## 3.3.2 Danh sách Function

S T T	Tên	Input Parameter	Output Parameter	Ý nghĩa	Ghi chú
1	THU	NAM	THU	Trả về Thứ có hóa đơn thanh toán nhiều nhất.	
2	DOANHSOPHIM	MAPHIM	DOANHSO	Trả về tổng doanh thu của bộ phim.	
3	DOANHSOSP	MASP	DOANHSO	Trả về tổng doanh thu của sản phẩm.	
4	TONGDTPHIM	THANG NAM	TONG	Trả về tổng doanh thu theo từng tháng.	
5	PHIMCAONHAT	NAM	PHIMMAX	Trả về một bộ phim có doanh thu cao nhất trong năm.	
6	PHIMTHAPNHAT	NAM	PHIMMIN	Trả về một bộ phim có doanh thu thấp nhất trong năm.	
7	SANPHAMCAONHAT	NAM	SPMAX	Trả về một sản phẩm có doanh thu cao nhất.	
8	SANPHAMTHAPNHAT	NAM	SPMIN	Trả về một sản phẩm có doanh thu thấp nhất.	

## 3.3.3. Mô tả một số Funtion

## 3.3.3.1. Doanh số phim.

➤ Tên: DOANHSOPHIM

Nội dung: Trả về tổng doanh thu của bộ phim

> Tham số đầu vào: MAPHIM

➤ Tham số đầu ra: DOANHSO

```
CREATE OR REPLACE FUNCTION DOANHSOPHIM (MAPHIM1
NVARCHAR2)
RETURN NUMBER
IS DOANHSO NUMBER;
BEGIN

SELECT DOANHTHU INTO DOANHSO
FROM PHIM
WHERE MAPHIM = MAPHIM1;
RETURN DOANHSO;
END;
```

- > Bước thực hiện:
- [1] Chọn doanh thu từ Phim.
- [2] Trả về doanh số Phim.

## 3.3.3.2. Doanh số sản phẩm.

➤ Tên: DOANHSOSP

➤ Nội dung: Trả về tổng doanh thu của sản phẩm

> Tham số đầu vào: MASP

➤ Tham số đầu ra: DOANHSO

```
CREATE OR REPLACE FUNCTION DOANHSOSP (MASP1 NVARCHAR2)
RETURN NUMBER
IS DOANHSO NUMBER;
BEGIN

SELECT DOANHTHU INTO DOANHSO
FROM SANPHAM
WHERE MASP = MASP1;
RETURN DOANHSO;
END;
```

- > Bước thực hiện:
- [1] Chọn doanh thu từ Sản phẩm.
- [2] Trả về doanh số Sản phẩm.

## 3.3.3.3. Tổng doanh thu Phim.

➤ Tên: TONGDTPHIM

Nội dung: Trả về tổng doanh thu theo từng tháng.

> Tham số đầu vào: THANG, NAM

➤ Tham số đầu ra: TONG

```
CREATE OR REPLACE FUNCTION TONGDTPHIM (THANG INT, NAM INT)

RETURN NUMBER
IS TONG NUMBER;

BEGIN

SELECT SUM(THANHTIEN) INTO TONG
FROM HOADON INNER JOIN CTVE ON HOADON.SOHD

=CTVE.SOHD

WHERE (extract (year from NGAYHD)) = NAM AND
(extract (MONTH from NGAYHD)) = THANG;
RETURN TONG;

END;
```

- > Bước thực hiện:
- [1] Chọn tổng thành tiền vé từ các chi tiết vé theo tháng và theo năm nhập vào.
- [2] Trả về tổng doanh thu sản phẩm.

### 3.3.3.4. Phim có doanh thu cao nhất.

➤ Tên: PHIMCAONHAT

Nội dung: Trả về một bộ phim có doanh thu cao nhất trong năm.

➤ Tham số đầu vào: NAM

➤ Tham số đầu ra: PHIMMAX

➤ Mã SOL:

```
CREATE OR REPLACE FUNCTION PHIMCAONHAT (NAM INT)
RETURN PHIM%ROWTYPE
IS PHIMMAX PHIM%ROWTYPE;
BEGIN
          SELECT PHIM. MAPHIM, TENPHIM INTO
PHIMMAX.MAPHIM, PHIMMAX.TENPHIM
          FROM HOADON INNER JOIN (CTVE INNER JOIN
(CACHIEU INNER JOIN PHIM ON CACHIEU.MAPHIM =
PHIM.MAPHIM) ON CTVE.MACC = CACHIEU.MACC) ON HOADON.SOHD
=CTVE.SOHD
          WHERE (extract (year from NGAYHD)) = NAM
          GROUP BY PHIM. MAPHIM, TENPHIM
          ORDER BY SUM (THANHTIEN) DESC
          FETCH FIRST 1 ROW ONLY;
          RETURN PHIMMAX;
END;
```

- > Bước thực hiện:
- [1] Khai báo biến PHIMMAX
- [2] Từ các bảng HOADON, CTVE, CACHIEU và PHIM được kết với nhau, chiếu 2 thuộc tính MAPHIM và TENPHIM vào biến tương ứng với điều kiện năm của NGAYHD bằng NAM nhập vào.
- [3] Nhóm lại và sắp xếp giảm dần theo doanh thu từng phim.
- [4] Chọn hàng đầu tiên (phim có doanh thu cao nhất) để trả về giá trị của biến PHIMMAX

## 3.3.3.5. Phim có doanh thu thấp nhất.

➤ Tên: PHIMTHAPNHAT

Nội dung: Trả về một bộ phim có doanh thu thấp nhất trong năm.

➤ Tham số đầu vào: NAM

➤ Tham số đầu ra: PHIMMIN

➤ Mã SQL:

```
CREATE OR REPLACE FUNCTION PHIMTHAPNHAT (NAM INT)
RETURN PHIM%ROWTYPE
IS PHIMMIN PHIM%ROWTYPE;
BEGIN
          SELECT PHIM. MAPHIM, TENPHIM INTO
PHIMMIN.MAPHIM, PHIMMIN.TENPHIM
          FROM HOADON INNER JOIN (CTVE INNER JOIN GHE ON
CTVE.MAGHE = GHE.MAGHE
INNER JOIN (CACHIEU INNER JOIN PHIM ON CACHIEU.MAPHIM =
PHIM.MAPHIM) ON CTVE.MACC = CACHIEU.MACC) ON HOADON.SOHD
=CTVE.SOHD
          WHERE (extract (year from NGAYHD)) = NAM
          GROUP BY PHIM. MAPHIM, TENPHIM
          ORDER BY SUM(GIA) ASC
          FETCH FIRST 1 ROW ONLY;
          RETURN PHIMMIN;
END;
```

#### > Bước thực hiện:

- [1] Khai báo biến PHIMMIN
- [2] Từ các bảng HOADON, CTVE, CACHIEU và PHIM được kết với nhau, chiếu 2 thuộc tính MAPHIM và TENPHIM vào biến tương ứng với điều kiện năm của NGAYHD bằng NAM nhập vào.
- [3] Nhóm lại và sắp xếp tăng dần theo doanh thu từng phim.
- [4] Chọn hàng đầu tiên (phim có doanh thu thấp nhất) để trả về giá trị của biến PHIMMIN

3.3.3.6. Sản phẩm có doanh thu cao nhất.

➤ Tên: SPCAONHAT

Nội dung: Trả về một sản phẩm có doanh thu cao nhất.

➤ Tham số đầu vào: NAM

➤ Tham số đầu ra: SPMAX

```
CREATE OR REPLACE FUNCTION SPCAONHAT (NAM INT)

RETURN SANPHAM%ROWTYPE

IS SPMAX SANPHAM%ROWTYPE;

BEGIN

SELECT SANPHAM.MASP, TENSP INTO

SPMAX.MASP, SPMAX.TENSP

FROM HOADON INNER JOIN (CTSP INNER JOIN

SANPHAM ON CTSP.MASP=SANPHAM.MASP) ON HOADON.SOHD

=CTSP.SOHD

WHERE (extract (year from NGAYHD)) = NAM

GROUP BY SANPHAM.MASP, TENSP

ORDER BY SUM(THANHTIEN) DESC

FETCH FIRST 1 ROW ONLY;

RETURN SPMAX;

END;
```

- > Bước thực hiện:
- [1] Khai báo biến SPMAX
- [2] Từ các bảng HOADON, CTSP và SANPHAM được kết với nhau, chiếu 2 thuộc tính MASP và TENSP vào biến tương ứng với điều kiện năm của NGAYHD bằng NAM nhập vào.
- [3] Nhóm lại và sắp xếp giảm dần theo doanh thu từng sản phẩm.
- [4] Chọn hàng đầu tiên (sản phẩm có doanh thu cao nhất) để trả về giá trị của biến SPMAX

## 3.3.3.7. Sản phẩm có doanh thu thấp nhất.

➤ Tên: SPTHAPNHAT

Nội dung: Trả về một sản phẩm có doanh thu thấp nhất.

➤ Tham số đầu vào: NAM

➤ Tham số đầu ra: SPMIN

```
CREATE OR REPLACE FUNCTION SPTHAPNHAT (NAM INT)
RETURN SANPHAM%ROWTYPE
IS SPMIN SANPHAM%ROWTYPE;
BEGIN

SELECT SANPHAM.MASP, TENSP INTO

SPMIN.MASP, SPMIN.TENSP

FROM HOADON INNER JOIN (CTSP INNER JOIN
SANPHAM ON CTSP.MASP=SANPHAM.MASP) ON HOADON.SOHD
=CTSP.SOHD

WHERE (extract (year from NGAYHD)) = NAM
GROUP BY SANPHAM.MASP, TENSP
ORDER BY SUM(THANHTIEN) ASC
FETCH FIRST 1 ROW ONLY;
RETURN SPMIN;
END;
```

- > Bước thực hiện:
- [1] Khai báo biến SPMIN
- [2] Từ các bảng HOADON, CTSP và SANPHAM được kết với nhau, chiếu 2 thuộc tính MASP và TENSP vào biến tương ứng với điều kiện năm của NGAYHD bằng NAM nhập vào.
- [3] Nhóm lại và sắp xếp tang dần theo doanh thu từng sản phẩm.
- [4] Chọn hàng đầu tiên (sản phẩm có doanh thu thấp nhất) để trả về giá trị của biến SPMIN

## 3.4. Sequence

### 3.4.2. Danh sách các Sequence

STT	Tên	Nội dung	Ghi chú
1	NV	Tạo mã số tự động cho nhân viên.	
2	SP	Tạo mã số tự động cho sản phẩm	
3	HD	Tạo mã số tự động cho hóa đơn	
4	KH	Tạo mã số tự động cho khách hàng	
5	PH	Tạo mã số tự động cho phim	
6	SK	Tạo mã số tự động cho sự kiện	
7	KM	Tạo mã số tự động cho khuyến mãi	

# 3.4.3 Mô tả một số Sequence

## 3.4.3.1 Tạo mã tự động nhân viên

➤ Tên: NV

> Nội dung: Tạo mã số tự động cho nhân viên

> SQL:

create sequence NV
nominvalue
nomaxvalue
increment by 1
start with 1
nocache
order
nocycle;

## 3.4.3.2 Tạo mã tự động sản phẩm

➤ Tên: SP

Nội dung: Tạo mã số tự động cho sản phẩm

> SQL:

create sequence SP
nominvalue
nomaxvalue
increment by 1
start with 1
nocache
order
nocycle;

## 3.4.3.3 Tạo mã tự động hóa đơn

➤ Tên: HD

Nội dung: Tạo mã số tự động cho hóa đơn

> SQL:

create sequence HD
nominvalue
nomaxvalue
increment by 1
start with 1
nocache
order
nocycle;

## 3.4.3.4 Tạo mã tự động khách hàng

Tên: KH

Nội dung: Tạo mã số tự động cho khách hàng

> SQL:

create sequence KH nominvalue nomaxvalue increment by 1 start with 1 nocache order nocycle;

## 3.4.3.5 Tạo mã tự động phim

➤ Tên: PH

> Nội dung: Tạo mã số tự động cho phim

➤ SQL:

create sequence PH
nominvalue
nomaxvalue
increment by 1
start with 1
nocache
order
nocycle;

## 3.4.3.6 Tạo mã tự động sự kiện

➤ Tên: SK

Nội dung: Tạo mã số tự động cho sự kiện

> SQL:

create sequence SK nominvalue nomaxvalue increment by 1 start with 1 nocache order nocycle;

# 3.4.3.7 Tạo mã tự động khuyến mãi

Tên: KM

Nội dung: Tạo mã số tự động cho khuyến mãi

> SQL:

create sequence KM nominvalue nomaxvalue increment by 1 start with 1 nocache order nocycle;

## CHƯƠNG 4. XỬ LÝ TRUY XUẤT ĐỒNG THỜI

### 4.1. Các mức cô lập trong Oracle

### 4.1.1. Repeatable read

Đây là mức cô lập mặc định của InnoDB. Consistent read trong cùng một transaction đọc snapshot được lập bởi lần SELECT đọc đầu tiên. Điều đó có nghĩa là nếu thực thi các câu lệnh SELECT giống nhau trong cùng một transaction thì dữ liệu được truy vấn bởi những câu lệnh SELECT này là nhất quán với nhau. Locking read (tức lệnh SELECT có FOR UPDATE hoặc LOCK IN SHARE MODE), UPDATE và DELETE, chế độ khóa phụ thuộc vào việc câu lệnh sử dụng điều kiện tìm kiếm độc nhất hay điều kiện tìm kiếm trong khoảng. Đối với điều kiện tìm kiếm độc nhất, InnoDB chỉ khóa đối với đơn vị dữ liệu đó, không khóa khoảng đơn vị dữ liệu trước nó. Đối với những điều kiện tìm kiếm khác, InnoDB khóa khoảng đơn vị dữ liệu mà nó quét qua bằng cách sử dụng gap locks hoặc next-key locks để tránh những session khác ghi dữ liệu mới vào trong khoảng đơn vị dữ liệu đó.

#### 4.1.2. Read committed

Mỗi consistent read thiết lập và đọc snapshot của chính nó thậm chí là trong cùng một transaction. Locking read (tức lệnh SELECT có FOR UPDATE hoặc LOCK IN SHARE MODE), UPDATE và DELETE, InnoDB chỉ khóa dòng dữ liệu, không khóa khoảng đơn vị dữ liệu trước nó. Vì vậy nó cho phép ghi dữ liệu mới cạnh dòng dữ liệu bị khóa. Gap locking chỉ được sử dụng cho ràng buộc khóa ngoại và khóa chính. Bởi vì Gap locking đã bị vô hiệu hóa nên vấn đề phantom có thể xảy ra khi những sessions khác có thể ghi dữ liệu mới vào những khoảng trống. Nếu sử dụng read committed, bạn phải sử dụng khóa nhị phân trên dòng. Sử dụng read committed có những tác dụng phụ sau: - Đối với lệnh UPDATE và DELETE, InnoDB chỉ giữ khóa trên dòng mà nó cập nhật hoặc xóa. Điều này làm giảm khả năng xảy ra deadlock. - Đối với lệnh UPDATE, nếu một dòng bị khóa, InnoDB thực thi semiconsistent read trả về phiên bản committed mới nhất đến Oracle để Oracle có thể xác định liệu dòng dữ liệu có phù hợp với điều kiện WHERE của lệnh UPDATE hay không. Nếu dòng thỏa điều kiện, Oracle đọc dòng đó lại một lần nữa và lúc này InnoDB vừa khóa nó và đợi một khóa trên nó

#### 4.1.3. Read uncommitted

Lệnh SELECT được thực thi mà không sinh ra khóa, nhưng phiên bản cũ hơn của dòng dữ liệu có thể được sử dụng. Vì vậy việc đọc dữ liệu là không nhất quán và sinh ra dirty read

## 4.2 Lost update

*Tình huống:* Trong hệ thống, hai nhân viên bán vé cho khách hàng cho cùng một phim trong cùng một thời điểm, trong lúc đang thực hiện bán vé đó thì hệ thống chưa commit dữ liệu đó thì người thứ hai cũng đang thao tác tương tự thì có thể xảy ra mất dữ liệu khi cập nhật doanh thu phim.

Thêm vé xem phim khi nhân viên bán vé	Thêm vé xem phim khi nhân viên bán vé	
T1 bán vé	T2 bán vé	
Create or replace procedure	Create or replace procedure	
ThemVeXemPhim(v_hoadon varchar2,	ThemVeXemPhim(v_hoadon varchar2,	
v_macc varchar2, v_maghe varchar2)	v_macc varchar2, v_maghe varchar2)	
AS	AS	
pMaphim varchar2(4);	pMaphim varchar2(4);	
pGia int;	pGia int;	
pDoanhThu phim.doanhthu%type;	pDoanhThu phim.doanhthu%type;	
begin	begin	
select cc.maphim into pMaphim	select cc.maphim into pMaphim	
from cachieu cc	from cachieu cc	
where cc.macc= v_macc;	where cc.macc= v_macc;	
select ghe.gia into pGia	select ghe.gia into pGia	
from ghe	from ghe	
where maghe= v_maghe;	where maghe= v_maghe;	
insert into CTVE (sohd, macc, maghe,	insert into CTVE (sohd, macc, maghe,	
makm) values (v_hoadon,	makm) values (v_hoadon,	
v_macc,v_maghe,null);	v_macc,v_maghe,null);	
select doanhthu into pDoanhThu	select doanhthu into pDoanhThu	
from phim	from phim	
where maphim= pMaphim;	where maphim= pMaphim;	

dbms_lock.sleep(10);	dbms_lock.sleep(10);
Update doanh thu phim tương ứng	Update doanh thu phim tương ứng
update phim  set doanhthu = pDoanhThu + pGia  where maphim= pMaphim  commit;	<pre>update phim set doanhthu = pDoanhThu + pGia where maphim= pMaphim; commit;</pre>
end;	end;

*Vấn đề xảy ra:* Việc cập nhật doanh thu phim ở T2 sẽ ghi đè lên cập nhật doanh thu phim ở T1. Dữ liệu doanh thu không đúng.

*Nguyên nhân:* T1 và T2 đọc giá trị doanh thu ban đầu. Tuy nhiên T2 không nhận biết được T1 đã cập nhật doanh thu mới trong khi T2 vẫn sử dụng giá trị doanh thu ban đầu. Vì vậy, dữ liệu doanh thu của phim bị cập nhật sai.

#### Giải pháp:

Hệ quản trị Oracle có hỗ trợ câu lệnh SELECT ... FOR, vì vậy sử dụng câu lệnh SELECT...FOR UPDATE thay cho câu lệnh SELECT lấy giá trị doanh thu ban đầu.

Kết quả: T2 đợi T1 thực hiện xong thì T2 mới vào đọc giá trị doanh thu phim và tiến hành cập nhật doanh thu mới nên dữ liệu không bị mất. Nguyên nhân là vì câu lệnh SELECT...FOR UPDATE sẽ đặt một exclusive lock trên dòng đã được chọn, nó giống như khi đang UPDATE và nó cũng chạy ngầm ở mức cô lập READ COMMITED cho dù mức thiết lập ban đầu là gì.

Thêm vé xem phim khi nhân viên bán vé T1 bán vé	Thêm vé xem phim khi nhân viên bán vé T2 bán vé
Create or replace procedure ThemVeXemPhim(v_hoadon varchar2, v_macc varchar2, v_maghe varchar2)	Create or replace procedure ThemVeXemPhim(v_hoadon varchar2, v_macc varchar2, v_maghe varchar2)
AS	AS
pMaphim varchar2(4);	pMaphim varchar2(4);
pGia int;	pGia int;
pDoanhThu phim.doanhthu%type;	pDoanhThu phim.doanhthu%type;
begin	begin
select cc.maphim into pMaphim	select cc.maphim into pMaphim
from cachieu cc	from cachieu cc
where cc.macc= v_macc;	where cc.macc= v_macc;
select ghe.gia into pGia	select ghe.gia into pGia
from ghe	from ghe
where maghe= v_maghe;	where maghe= v_maghe;
insert into CTVE (sohd, macc, maghe, makm) values (v_hoadon, v_macc,v_maghe,null);	insert into CTVE (sohd, macc, maghe, makm) values (v_hoadon, v_macc,v_maghe,null);
select doanhthu into pDoanhThu	select doanhthu into pDoanhThu
from phim	from phim
where maphim= pMaphim FOR UPDATE;	where maphim= pMaphim FOR UPDATE;
dbms_lock.sleep(10);	dbms_lock.sleep(10);
Update doanh thu phim tương ứng	Update doanh thu phim tương ứng

```
update phimupdate phimset doanhthu = pDoanhThu + pGiaset doanhthu = pDoanhThu + pGiawhere maphim= pMaphim;where maphim= pMaphim;commit;commit;end;end;
```

#### 4.1.4. Serializable

Mức cô lập này giống với repeatable read nhưng InnoDB ngầm định chuyển tất cả câu lệnh SELECT thành SELECT .. LOCK IN SHARE MODE nếu chế độ autocommit bị vô hiệu hóa. Nếu ở chế độ autocommit, lệnh SELECT không bị chuyển đổi. Vì vậy đây là chế độ chỉ cho phép đọc tuần tự nếu thực thi như một consistent read và không khóa transaction khác.

### 4.1.5. Locking reads

Nếu truy vấn dữ liệu và sau đó ghi mới hoặc cập nhật dữ liệu liên quan trong cùng một transaction thì lệnh SELECT thông thường không đủ bảo vệ nhất quán dữ liệu. Những transaction khác có thể cập nhật hoặc xóa những dòng dữ liệu mà vừa truy vấn. InnoDB hỗ trợ 2 loại locking reads đảm bảo sự an toàn dữ liệu cao hơn: SELECT ... LOCK IN SHARE MODE: thiết lập khóa đọc trên bất cứ dòng nào mà nó đọc. Những session khác có thể đọc những dòng dữ liệu đó nhưng không thể cập nhật chúng cho đến khi transaction đang thực thi commit. Nếu có bất cứ dòng dữ liệu nào đang được thay đổi bởi một transaction khác, lệnh truy vấn của bạn phải đọi cho đến khi transaction đó kết thức và sử dụng giá trị mới nhất của chúng. SELECT ... FOR UPDATE khóa những dòng dữ liệu và bất cứ những dòng liên quan giống như lệnh UPDATE. Những transaction khác bị cấm trong việc cập nhật, SELECT ... LOCK IN SHARE MODE hoặc đọc những dữ liệu này ở bất kỳ mức cô lập nào. Consistent read bỏ qua các khóa được thiết lập.

Tất cả các khóa được thiết lập bởi LOCK IN SHARE MODE và FOR UPDATE sẽ được giải phóng khi transaction commit hoặc roll back.

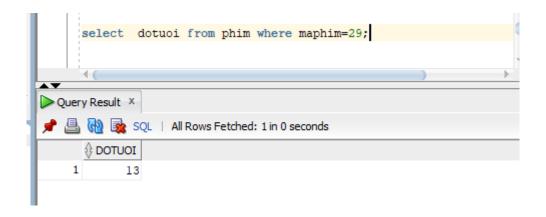
## 4.3 Dirty read

#### a. Dirty read thực thi trên ORACLE

**Tình huống:** Trong hệ thống, khi hệ quản trị đang cập nhật độ tuổi của phim thì nhân viên xem thông tin phim.

Ví dụ với phim có maphim=29

- Dữ liệu tuổi ban đầu là:



- Ở Transaction T1 là xem độ tuổi ở maphim=29
- Ở Transaction T2 là cập nhật lai độ tuổi nhưng sau đó ta cố tình roll back lại transaction này

```
create or replace procedure updatetuoil( v_maphim phim.maphimis)

begin

update phim
set dotuoi=34
where maphim= v_maphim;
DBMS_LOCK.SLEEP(20);
rollback;
end;
execute updatetuoil(29);

Dbms Output

Buffer Size: 20000

demo2_x
```

Sau khi thực hiện hai giao tác đồng thời thì ta thấy độ tuổi của mã phim
 29 vẫn không đổi

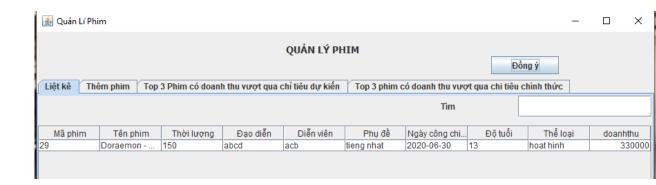
```
□ create or replace PROCEDURE XEMDOTUOI1(V MAPHIM PHIM
          V DOTUOI
                         PHIM. DOTUOI%TYPE;
      BEGIN
      -- SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
      SELECT DOTUOI INTO V DOTUOI
       FROM PHIM
       WHERE MAPHIM=V MAPHIM;
         DBMS LOCK.SLEEP(10);
      DBMS_OUTPUT.PUT_LINE('DO TUOI:'||V_DOTUOI);
          COMMIT;
      END;
      execute xemdotuoil(29);
Dbms Output
💠 🥟 🔡 🖺 | Buffer Size: 20000
DO TUOI:13
DO TUOI:13
DO TUOI:13
```

Từ đó suy ra là: Do mức cô lập READ COMMITTED là mức cô lập mặc định trong Oracle và mức cô lập này chỉ cho phép transaction đọc những giá trị của transaction đã commit. Do đó, T2 sẽ đợi cho đến khi T1 thực hiện xong thì T2 mới vào đọc giá trị. Trường hợp Dirty Read sẽ không xảy ra trong hệ quản trị Oracle.

# b . Dirty read thực thi trên CHƯƠNG TRÌNH

Ví dụ với phim có maphim=29

- Dữ liệu tuổi ban đầu là:



- Sau khi thực hiện 2 transaction T1 và T2 thì độ tuổi ở maphim=29 vẫn không đổi:

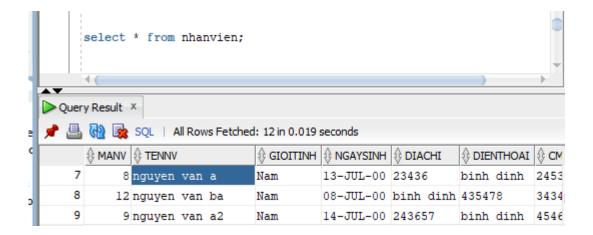


## 4.4 Unrepeatable read

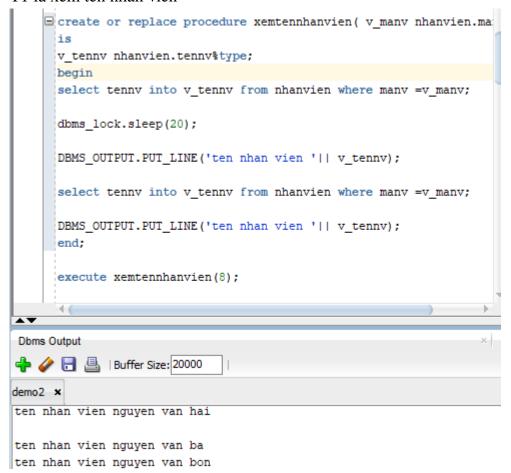
### a. Unrepeatable read thực thi trên ORACLE

**Tình huống:** Trong hệ thống, khi một quản trị viên đang xem tên của nhân viên thì quản trị viên khác lại cập nhật tên của chính nhân viên đó:

Ví dụ nhân viên có manv=8 có tên ban đầu là:



- Khi thực hiện 2 transaction T1 và T2
- T1 là xem tên nhân viên



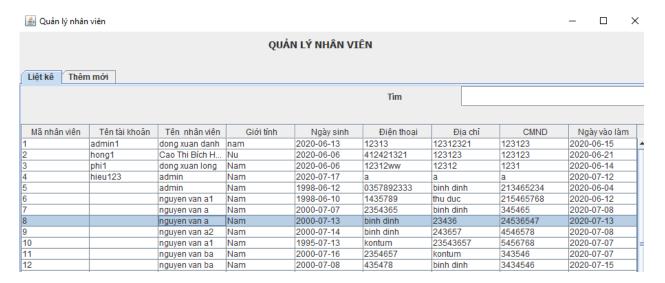
- T2 là cập nhật lại tên của nhân viên

- Hai giao tác này xảy ra đồng thời .Mức cô lập của Oracle mặc định là Commited Read nên nó cho phép transaction đọc những giá trị của transaction đã commit do đó ta đã có 2 lần **SELECT** có kết quả khác nhau:
- Vấn đề xảy ra: Kết quả 2 lần SELECT khác nhau mặc dù đang ở trong cùng một transaction.
- Nguyên nhân: T2 tiến hành cập nhật giá trị thành công vì với Read Committed thì Shared Lock chỉ tồn tại cho đến khi hoàn thành câu lệnh SELECT nên việc đặt một Exclusive Lock vào giữa khoảng thời gian đó là khả thi.
- *Giải pháp:* Chuyển mức cô lập của T1 thành SERIALIZABLE READ để đảm bảo nhiều tác vụ đọc dữ liệu trong cùng một transaction có kết quả giống nhau.
- **Kết quả:** T2 đợi T1 thực hiện xong thì T2 mới có thể cập nhật lại tên nhân viên và 2 kết quả trả về ở T1 vẫn đồng nhất.
- Lưu ý: SERIALIZABLE READ thì Shared Lock được đặt cho đến hết giao tác, làm cho việc thiết lập một Exclusive Lock lên cùng một đơn vị dữ liệu trở nên bất khả thi và phải đợi.

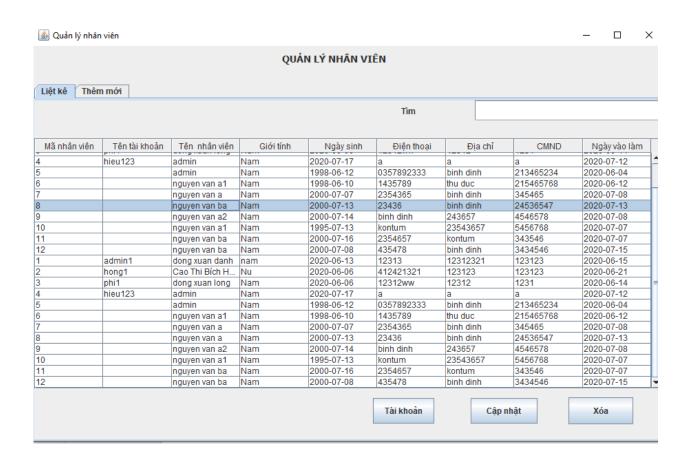
```
create or replace procedure xemtennhanvien( v_manv nhanvien.ma
      v_tennv nhanvien.tennv%type;
     begin
     SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
     select tennv into v_tennv from nhanvien where manv =v_manv;
     dbms lock.sleep(20);
     DBMS_OUTPUT.PUT_LINE('ten nhan vien '|| v_tennv);
     select tennv into v_tennv from nhanvien where manv =v_manv;
      DBMS_OUTPUT.PUT_LINE('ten nhan vien '|| v_tennv);
      execute xemtennhanvien(8);
Dbms Output
👍 🥢 🔡 🖺 | Buffer Size: 20000
demo2 x
ten nhan vien nguyen van bon
ten nhan vien nguyen van bon
ten nhan vien nguyen van bon
```

# b.Unrepeatable read thực thi trên CHƯƠNG TRÌNH

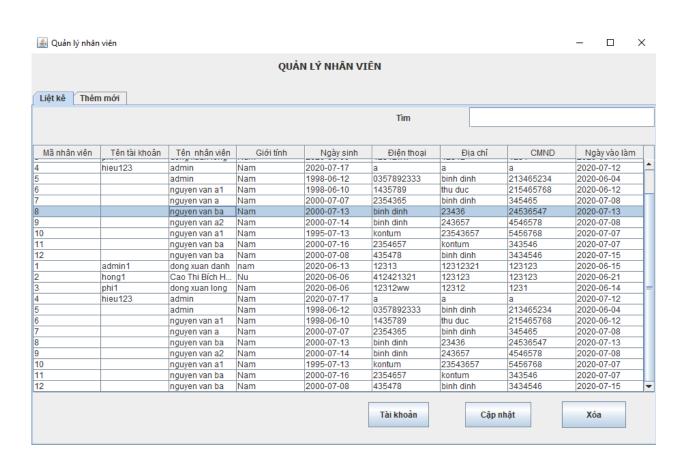
Tên nhân viên ban đầu với manv =8 la:



Sau khi cập nhật tên nhân viên với manv=8 vơi tennv=" nguyen van ba" thì ta thấy 2 lần select cho kết quả khác nhau:



# Giải pháp:

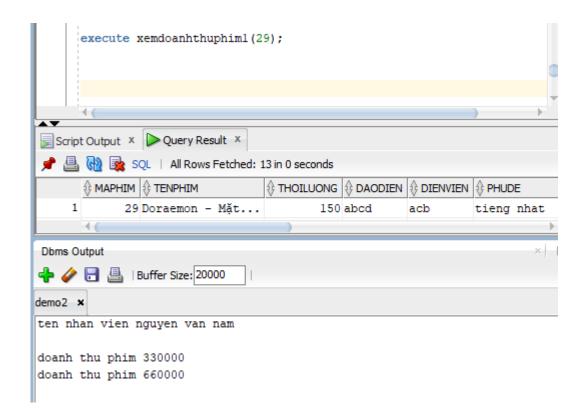


Sau khi cho về mức cô lập SERIALIZABLE thì kết quả ở hai lần select đã giống nhau

#### 4.5 Phantom

a.Phantom read thực hiện ở mức ORACLE

Đầu tiên khi thục hiện hai giao tác ở mức cô lập commited read thì ta thấy doanh thu phim của maphim=29 bi thay đổi:

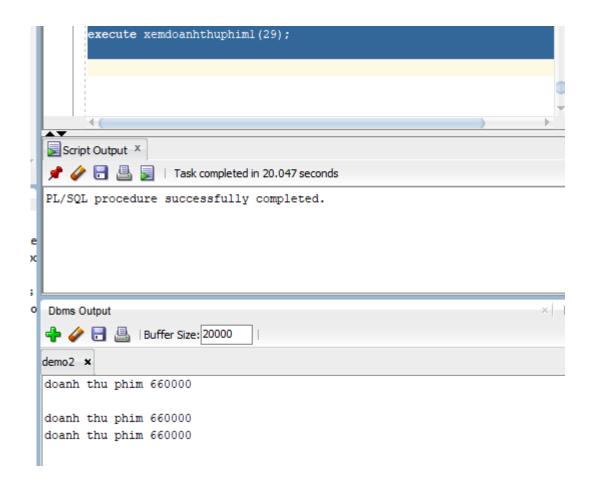


Giải pháp sử dụng mức cô lập cao nhất là serializable sau đó ta sẽ select ra hai giá trị giống nhau:

Doanh thu của maphim=29 ở giao tác T1 là

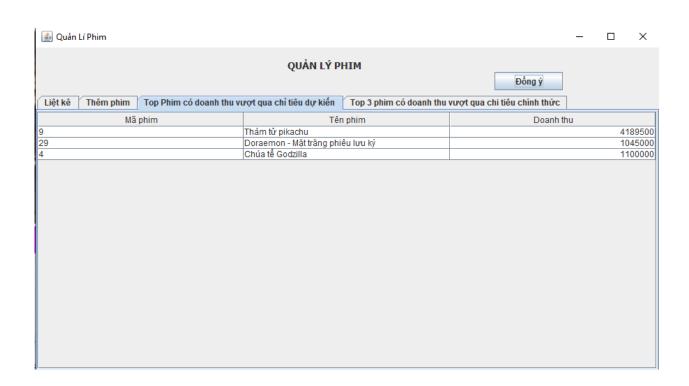


Nhưng sau khi sử dụng mức cô lập cao nhất là serializable sau đó ta sẽ select ra hai giá trị cũ giống nhau:



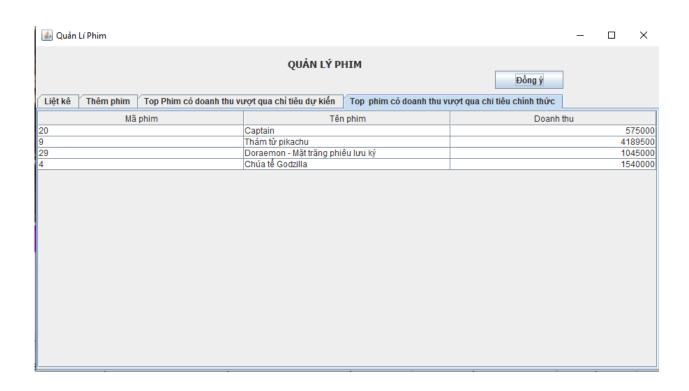
# b.Phantom read thực hiện trên CHƯƠNG TRÌNH

Lúc đầu danh sach phim co doanh thu > 500000 là:



Đong thoi lúc đó co nguoi mua them vé và có phim có doanh thu >500000 và nó được đưa vao DS du kiến

Sau khi quan trị viên xem kỹ càng và bấm nút đồng ý thì thấy dữ liệu ban đầu đã thay đổi:



#### 4.6 Deadlock

#### 4.6.1 Conversion Deadlock

*Tình huống:* Một nhân viên quản lý phim xem thông tin phim sau đó cập nhật thông tin phim đó. Đồng thời một nhân viên quản lý phim khác cũng vào xem rồi cập nhật thông tin của cùng một phim . Giao tác thực hiện được đặt ở mức SERIALIZABLE gây ra Deadlock.

Nhân viên quản lý phim xem và cập nhật	Nhân viên quản lý phim xem và cập nhật
thông tin T1	thông tin T2

create or replace PROCEDURE create or replace PROCEDURE XEMVASUATTPHIM1 **XEMVASUATTPHIM2** (V\_MAPHIM phim.maphim%type) (V\_MAPHIM phim.maphim%type) AS AS pTHOILUONG pDOTUOI PHIM.THOILUONG% TYPE; PHIM.DOTUOI%TYPE; BEGIN SET TRANSACTION **BEGIN SET TRANSACTION** ISOLATION LEVEL **ISOLATION LEVEL** SERIALIZABLE; **SERIALIZABLE**; SELECT THOILUONG INTO SELECT DOTUOI INTO pTHOILUONG FROM PHIM pDOTUOI FROM PHIM WHERE WHERE MAPHIM=V\_MAPHIM; MAPHIM=V\_MAPHIM; DBMS\_OUTPUT.PUT\_LINE('THOI DBMS\_OUTPUT.PUT\_LINE('DO LUONG PHIM: | | pTHOILUONG); TUOI:'||pDOTUOI); DBMS LOCK.SLEEP(10); DBMS LOCK.SLEEP(10); UPDATE PHIM SET UPDATE PHIM SET DOTUOI = THOILUONG =140 WHERE 14 WHERE MAPHIM=V\_MAPHIM; MAPHIM=V\_MAPHIM; COMMIT; END; COMMIT: END;

**♣** Demo:

- Thuc thi Procedure XEMVASUATTPHIM1:

```
Command Prompt - sqlplus
SOL>
SQL> execute XEMVASUATTPHIM1(10);
PL/SQL procedure successfully completed.
SQL> create or replace PROCEDURE XEMVASUATTPHIM11 (V MAPHIM phim.maphim%type)
          pTHOILUONG PHIM.THOILUONG%TYPE;
     BEGIN
         SELECT THOILUONG INTO pTHOILUONG FROM PHIM WHERE MAPHIM=V_MAPHIM; DBMS_OUTPUT.PUT_LINE('THOI LUONG PHIM:'||pTHOILUONG);
  8
          DBMS_LOCK.SLEEP(20);
  9
          UPDATE PHIM SET THOILUONG =150 WHERE MAPHIM=V_MAPHIM;
 10
          COMMIT;
 11
     END;
Procedure created.
SQL> execute XEMVASUATTPHIM11(10);
PL/SQL procedure successfully completed.
SQL>
```

- Thuc thi Procedure XEMVASUATTPHIM2:

```
SQL Plus
         pDOTUOI PHIM.DOTUOI%TYPE;
    BEGIN
 6
         SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
 7
         SELECT DOTUOI INTO pDOTUOI FROM PHIM WHERE MAPHIM=V MAPHIM;
 8
         DBMS_OUTPUT.PUT_LINE('DO TUOI:'||pDOTUOI);
         DBMS LOCK.SLEEP(20);
10
        UPDATE PHIM SET DOTUOI = 17 WHERE MAPHIM=V_MAPHIM;
11
        COMMIT;
12
    END;
13
Procedure created.
SQL>
SQL> execute XEMVASUATTPHIM2(10);
BEGIN XEMVASUATTPHIM2(10); END;
ERROR at line 1:
ORA-08177: can't serialize access for this transaction
ORA-06512: at "DANH2.XEMVASUATTPHIM2", line 10
ORA-06512: at line 1
SQL>
SQL>
```

Khi đó Deadlock xảy ra và T2 rollback. Giá trị cập nhật ở T1 được cập nhật thành công.

Vấn đề xảy ra: T2 chờ T1 giải phóng khóa đọc để ghi trên Phim, T1 cũng chờ T2 giải phóng khóa đọc để ghi trên Phim. Kết quả dẫn đến hệ quản trị báo lỗi và hủy T2.

**Nguyên nhân**: Do mức cô lập là mức SERIALIZABLE nghĩa là các giao tác khác phải chờ đến khi giao tác này kết thúc nếu muốn cập nhật hay thay đổi giá trị trên đơn vị dữ liệu đó. Trong trường hợp trên, lệnh update ở T1 yêu cầu khóa ghi và khóa đó đang được T2 giữ do đó phải chờ khóa ở T2 giải phóng. Nhưng khóa ở T2 chỉ giải phóng khi và chỉ khi T2 commit transaction. Nhưng để T2 commit thì T2 phải thực hiện được lệnh update, phải xin được khóa ghi nhưng không được vì yêu cầu xin khóa đọc của T1 trước đó chưa được giải phóng.

*Giải pháp:* Hệ quản trị tự giải quyết deadlock bằng cách rollback 1 transaction. Để giải quyết deadlock ta giảm mức cô lập thành read committed hoặc đặt Xlock cho bảng Phim.

*Kết quả:* Ở mức lệnh Read Committed khóa đọc sẽ được giải phóng ngay sau khi đọc xong dữ liệu nên sẽ không gây ra tắc nghẽn hoặc xử lý SELECT... FOR UPDATE sẽ khóa bảng Phim lại làm cho cho T2 không giữ khóa nên T1 không cần chờ T2 nhả khóa để thực hiện.

## Chạy lại:

Nhân viên quản lý phim xem và cập nhật thông tin T1	Nhân viên quản lý phim xem và cập nhật thông tin T2
	. 5
create or replace PROCEDURE	create or replace PROCEDURE
XEMVASUATTPHIM11 (V_MAPHIM	XEMVASUATTPHIM22 (V_MAPHIM
phim.maphim%type)	phim.maphim%type)
AS	AS
pTHOILUONG	pDOTUOI PHIM.DOTUOI%TYPE;
PHIM.THOILUONG%TYPE;	
BEGIN	BEGIN
SELECT THOILUONG INTO	SELECT DOTUOI INTO pDOTUOI
pTHOILUONG FROM PHIM WHERE	FROM PHIM WHERE
MAPHIM=V_MAPHIM;	MAPHIM=V_MAPHIM;
DBMS_OUTPUT.PUT_LINE("THOI	DBMS_OUTPUT.PUT_LINE('DO
LUONG PHIM:'  pTHOILUONG);	TUOI:'  pDOTUOI);
DBMS_LOCK.SLEEP(20);	DBMS_LOCK.SLEEP(20);
UPDATE PHIM SET THOILUONG	UPDATE PHIM SET DOTUOI = 17
=150 WHERE MAPHIM=V_MAPHIM;	WHERE MAPHIM=V_MAPHIM;
DBMS_OUTPUT.PUT_LINE('DO	DBMS_OUTPUT.PUT_LINE('DO
TUOI:'  pTHOILUONG );	TUOI:'  pDOTUOI);
COMMIT;	COMMIT;
END;	END;

♣ Demo:

- Thực thi Procedure XEMVASUATTPHIM1, thời lượng phim hiện tại là 182:

```
Command Prompt - sqlplus
SQL>
SQL> execute XEMVASUATTPHIM1(10);
PL/SQL procedure successfully completed.
SQL> create or replace PROCEDURE XEMVASUATTPHIM11 (V MAPHIM phim.maphim%type)
  2 AS
          pTHOILUONG PHIM.THOILUONG%TYPE;
     BEGIN
         SELECT THOILUONG INTO pTHOILUONG FROM PHIM WHERE MAPHIM=V_MAPHIM; DBMS_OUTPUT.PUT_LINE('THOI LUONG PHIM:'||pTHOILUONG);
  8
          DBMS_LOCK.SLEEP(20);
          UPDATE PHIM SET THOILUONG =150 WHERE MAPHIM=V_MAPHIM;
  9
 10
          COMMIT;
     END;
 11
Procedure created.
SQL> execute XEMVASUATTPHIM11(10);
PL/SQL procedure successfully completed.
SQL>
```

- Thực thi Procedure XEMVASUATTPHIM2, độ tuổi dành cho phim là 17:

```
SQL Plus
                                                                                SQL>
SQL>
SQL> create or replace PROCEDURE XEMVASUATTPHIM22 (V_MAPHIM phim.maphim%type)
         pDOTUOI PHIM.DOTUOI%TYPE;
    BEGIN
 5
 6
         SELECT DOTUOI INTO pDOTUOI FROM PHIM WHERE MAPHIM=V MAPHIM;
 8
         DBMS_OUTPUT.PUT_LINE('DO TUOI:'||pDOTUOI);
 9
         DBMS_LOCK.SLEEP(20);
 10
         UPDATE PHIM SET DOTUOI = 17 WHERE MAPHIM=V_MAPHIM;
11
         COMMIT;
 12
    END;
13
Procedure created.
SQL>
SQL> execute XEMVASUATTPHIM22(10);
PL/SQL procedure successfully completed.
SQL>
```

Kết quả, sau khi hạ mức cô lập xuống mức READ COMMITED (mức cô lập mặc định trong Oracle) thì vấn đề deadlock trên được giải quyết.



#### 4.6.2 Cycle Deadlock

*Tình huống:* Khi hai quản trị viên cùng cập nhật cho 2 bảng Phim và Sản phẩm, một người thì cập nhật Phim rồi sau đó là Sản phẩm và trong cùng lúc đó người còn lại cập nhật Sản phẩm và Phim → gây ra Deadlock.

Quản trị viên cập nhật T1	Quản trị viên cập nhật T2		
create or replace PROCEDURE	create or replace PROCEDURE		
CAPNHATSPVAPHIM1 (V_MAPHIM	CAPNHATSPVAPHIM2 (V_MAPHIM		
VARCHAR2, V_MASP VARCHAR2)	VARCHAR2, V_MASP VARCHAR2)		
AS	AS		
BEGIN	BEGIN		
UPDATE PHIM SET THOILUONG	UPDATE SANPHAM SET GIA =		
= 150 WHERE	6000 WHERE MASP=V_MASP;		
MAPHIM=V_MAPHIM;	DBMS_LOCK.SLEEP(5);		
DBMS_LOCK.SLEEP(5);	UPDATE PHIM SET DOTUOI=16		
UPDATE SANPHAM SET GIA =	WHERE MAPHIM=V_MAPHIM;		
6000 WHERE MASP=V_MASP;	COMMIT;		
COMMIT;	END;		
END;			

Vấn đề: T1 chờ T2 trả khóa ghi trên Sản phẩm và T2 chờ T1 trả khóa ghi trên Phim

Nguyên nhân: Do mức cô lập mặc định của Oracle là READ COMMITTED nghĩa là các giao tác khác phải chờ đến khi giao tác này kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này. Trong trường hợp này lệnh update ở T1 yêu cầu khóa ghi và phải chờ khóa ghi ở T2 giải phóng mà khóa ghi ở T2 được giải phóng khi và chỉ khi T2 commit. Nhưng để T2 commit được thì T2 phải thực hiện được lệnh update, phải xin khóa ghi trước đó của T1 chưa được giải phóng

# CHƯƠNG 5: THIẾT KẾ GIAO DIỆN

STT	Tên giao diện
1	Trang menu
2	Trang quản lý bán vé 1: Chọn phim và lịch chiếu
3	Trang quản lý bán vé 2: Chọn ghế
4	Trang quản lý bán vé 3: Thêm sản phẩm
5	Hóa đơn
6	Xuất vé
7	Quản lý khách hàng (Tra cứu + Xóa khách hàng)
8	Thêm khách hàng
9	Cập nhật khách hàng
10	Quản lý khuyến mãi (Tra cứu + Xóa khuyến mãi)
11	Thêm khuyến mãi
12	Cập nhật khuyến mãi
13	Quản lý sự kiện (Tra cứu + Xóa sự kiện)
14	Thêm sự kiện
15	Cập nhật sự kiện
16	Quản lý sản phẩm (Tra cứu + Xóa sản phẩm)
17	Thêm sản phẩm
18	Cập nhật sản phẩm
19	Quản lý ca chiếu (Tra cứu + Xóa ca chiếu)
20	Thêm ca chiếu: Step 1
21	Thêm ca chiếu: Step 2
22	Cập nhật ca chiếu: Step 1
23	Cập nhật ca chiếu: Step 2
24	Quản lý phim (Tra cứu + Xóa phim)
25	Thêm phim
26	Cập nhật phim
27	Quản lý phòng chiếu (Tra cứu + Xóa phòng chiếu)
28	Thêm phòng chiếu
29	Cập nhật phòng chiếu
30	Quản lý Cơ sở vật chất (Tra cứu + Xóa Cơ sở vật chất)
31	Thêm Cơ sở vật chất
32	Cập nhật Cơ sở vật chất
33	Quản lý Nhân viên (Tra cứu + Xóa Nhân viên)
34	Thêm Nhân viên
35	Cập nhật Nhân viên
36	Phân quyền truy cập
37	Lập báo cáo doanh thu

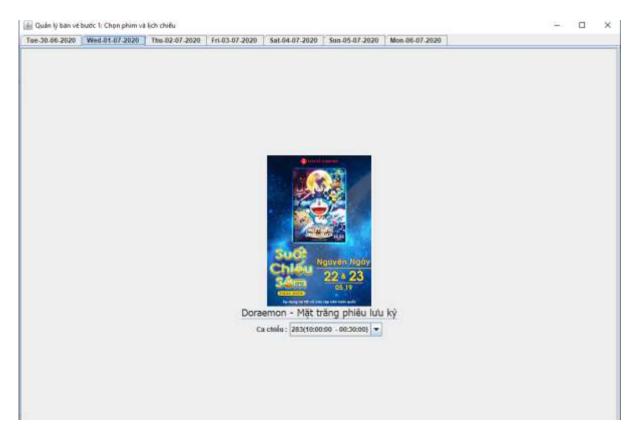
Bảng Giao diện

## Trang menu



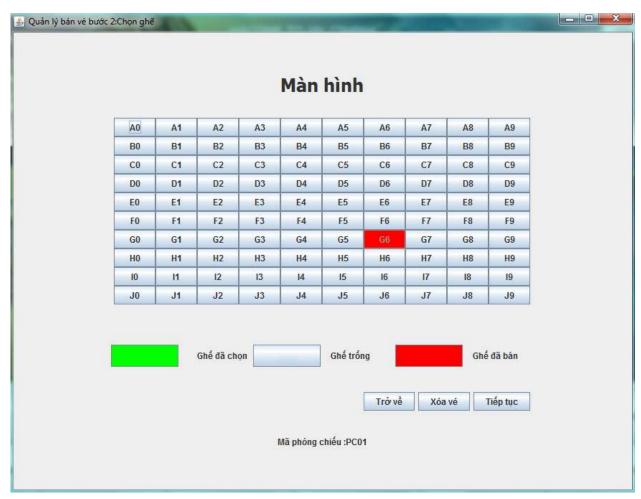
Hình. Giao diện Trang Menu

## Trang quản lý bán vé: Đặt phim và thêm lịch chiếu



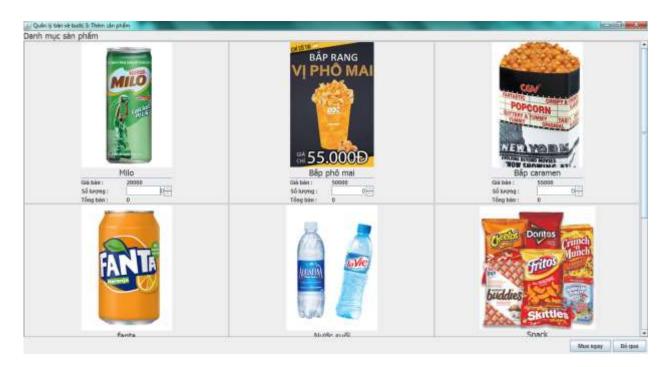
Hình. Giao diện trang quản lý bán vé

## Trang quản lý bán vé 2: Chọn ghế



Hình. Giao diện trang quản lý bán vé 2

## Trang quản lý bán vé 3: Thêm sản phẩm



Trang. Giao diện trang quản lý bán vé 3

#### Hóa đơn



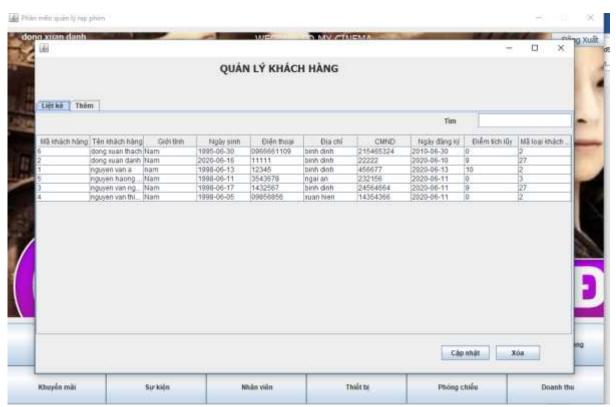
Hình. Giao diện hóa đơn

## Xuất vé



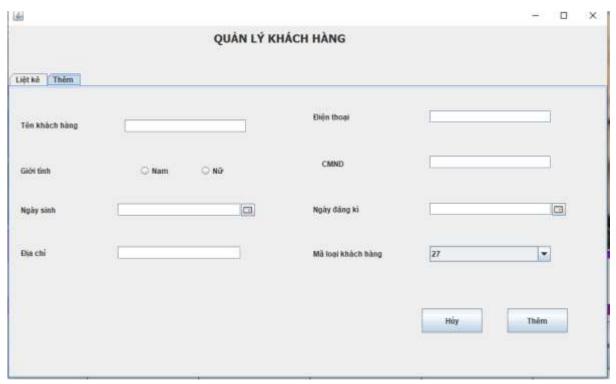
Hình. Giao diện xuất vé

### Quản lý khách hàng (Tra cứu + Xóa khách hàng)



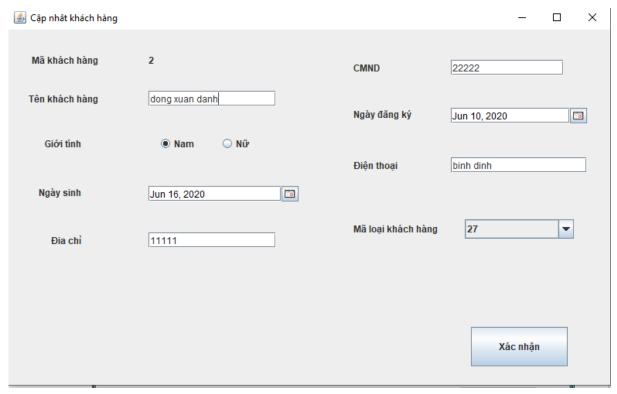
Hình.  $Giao\ diện\ quản\ lý\ khách\ hàng\ (Tra\ cứu + xóa)$ 

### Thêm khách hàng



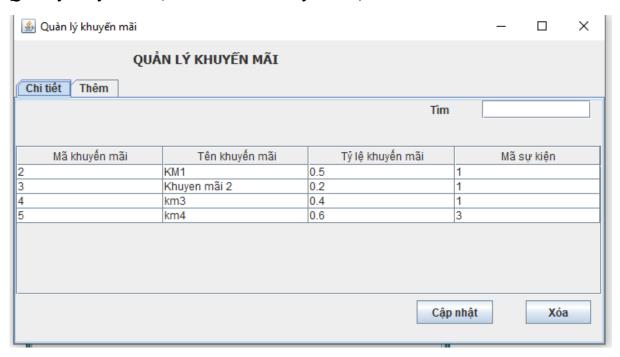
Hình. Giao diện quản lý khách hàng (Thêm khách hàng)

### Cập nhật khách hàng



Hình . Giao diện Cập nhật khách hàng

## Quản lý khuyến mãi (Tra cứu + Xóa khuyến mãi)



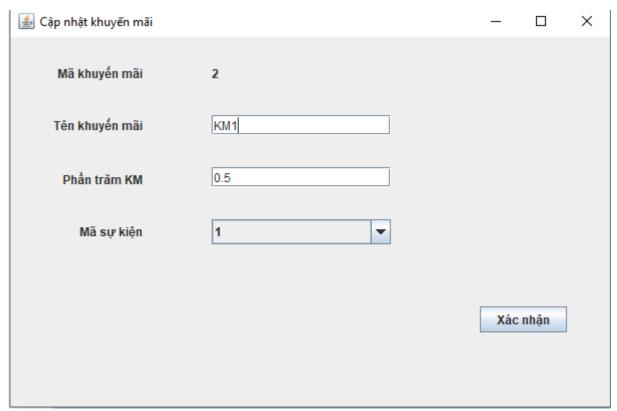
Hình. Giao diện Quản lý khuyến mãi (Tra cứu + Xóa)

## Thêm khuyến mãi



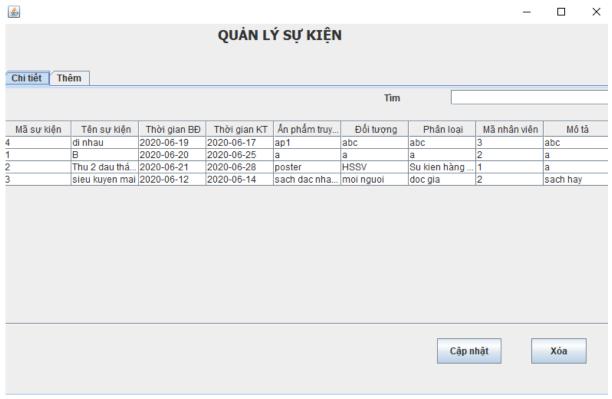
Hình. Giao diện Thêm khuyến mãi

# Cập nhật khuyến mãi



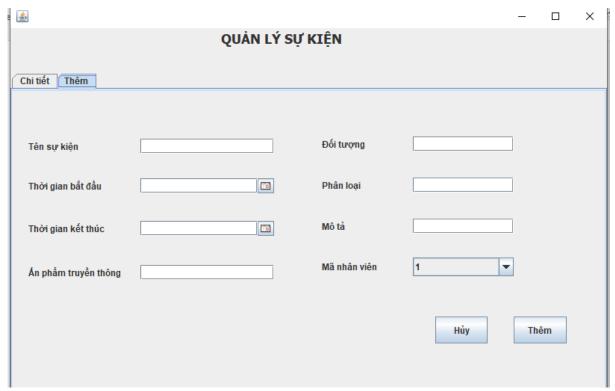
Hình. Giao diện cập nhật khuyến mãi

### Quản lý sự kiện (Tra cứu + Xóa sự kiện)



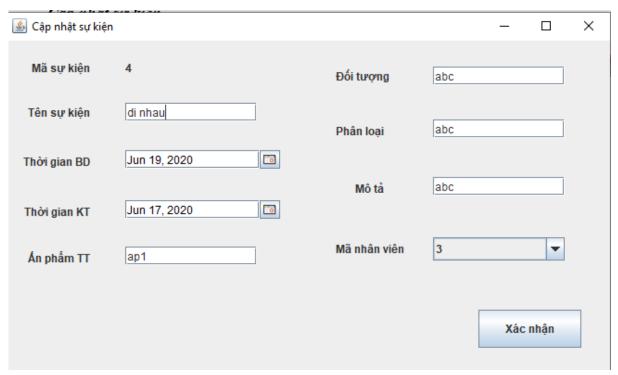
Hình. Giao diện Quản lý sự kiện (Tra cứu + Xóa sự kiện)

## Thêm sự kiện



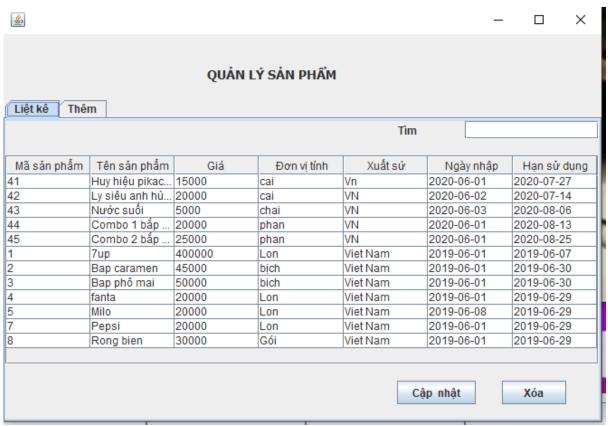
Hình. Giao diện Thêm sự kiện

### Cập nhật sự kiện



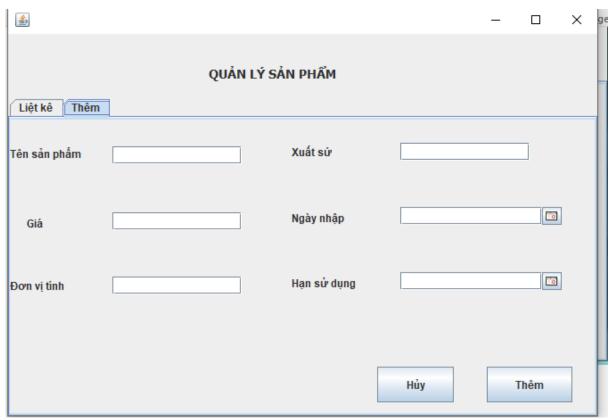
Hình. Giao diện Cập nhật sự kiện

## Quản lý sản phẩm (Tra cứu + Xóa sản phẩm)



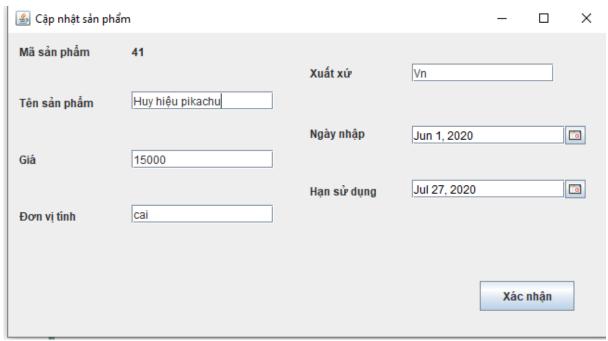
Hình. Giao diện quản lý sản phẩm (xóa + tra cứu)

## Thêm sản phẩm



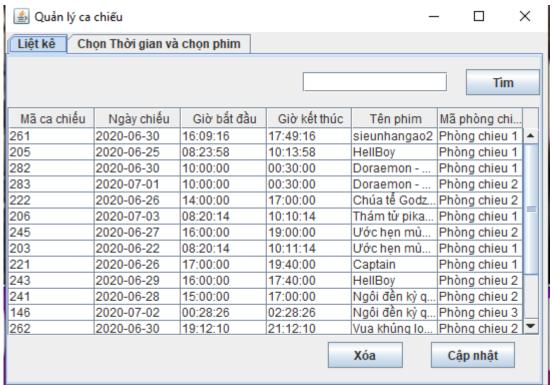
Hình. Giao diện Thêm sản phẩm

## Cập nhật sản phẩm



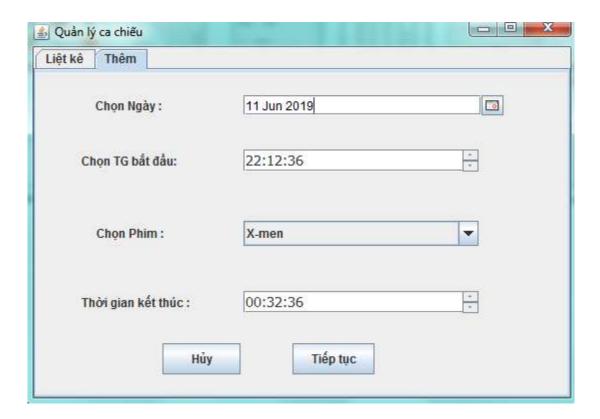
Hình. Giao diện Cập nhật sản phẩm

### Quản lý ca chiếu (Tra cứu + Xóa ca chiếu)



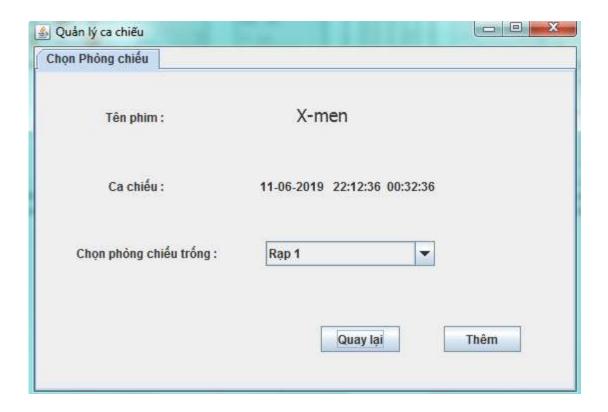
Hình. Giao diện quản lý ca chiếu (Tra cứu + Xóa ca chiếu)

## Thêm ca chiếu: Step 1



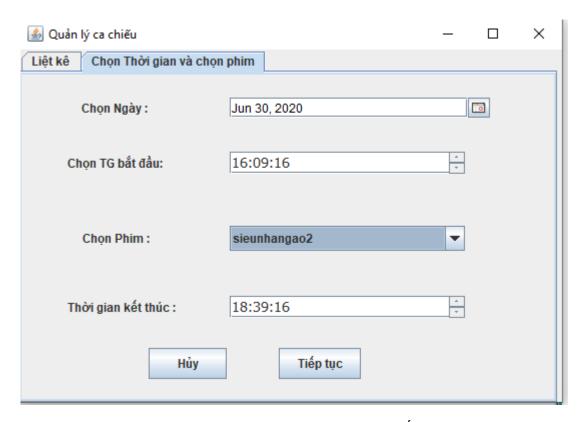
Hình. Giao diện Thêm ca chiếu Step 1

## Thêm ca chiếu: Step 2



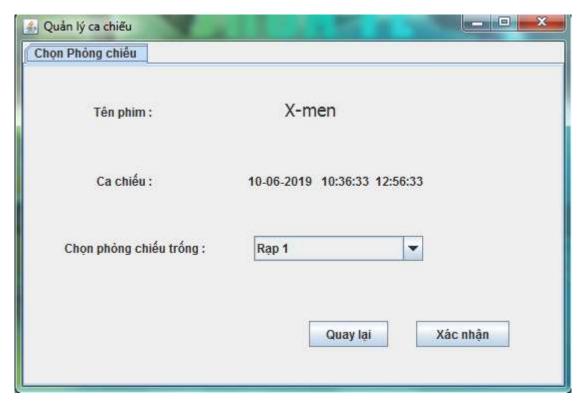
Hình. Giao diện thêm ca chiếu Step 2

## Cập nhật ca chiếu: Step 1



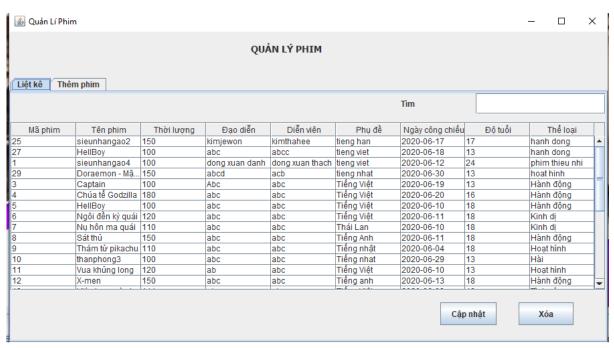
Hình. Giao diện cập nhật ca chiếu Step 1

## Cập nhật ca chiếu: Step 2



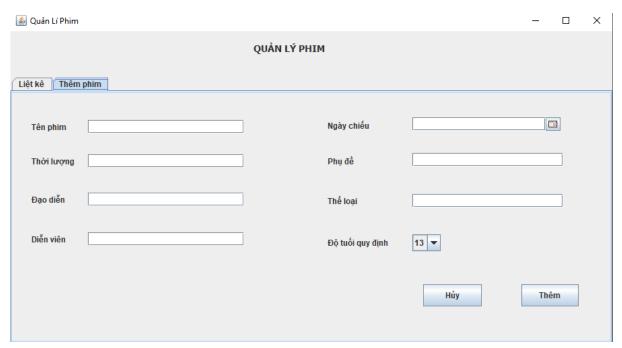
Hình. Giao diện cập nhật ca chiếu Step 2

#### Quản lý phim (Tra cứu + Xóa phim)



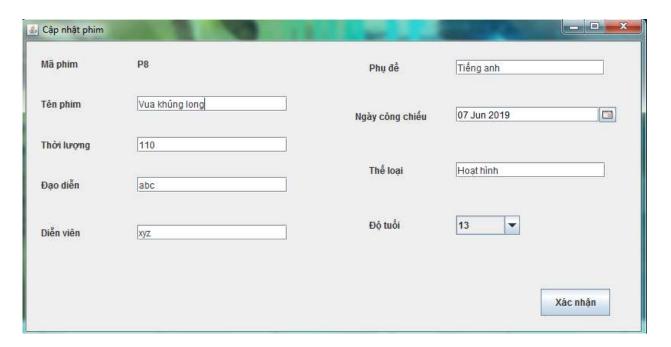
Hình. Giao diện Quản lý phim (Tra cứu + xóa)

#### Thêm phim



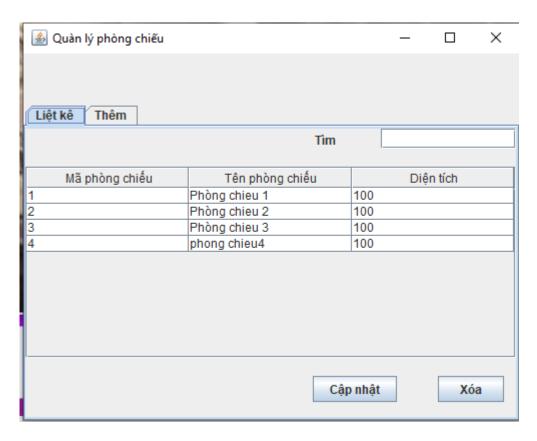
Hình. Giao diện Thêm phim

### Cập nhật phim



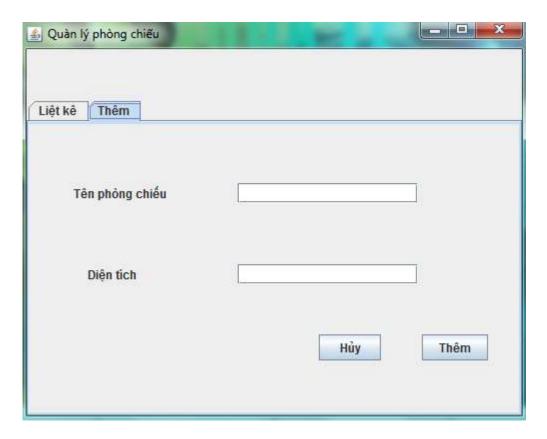
Hình. Giao diện Cập nhật phim

## Quản lý phòng chiếu (Tra cứu + Xóa phòng chiếu)



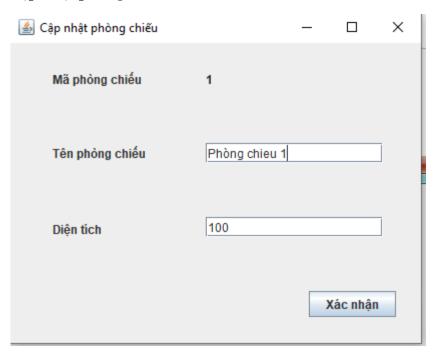
Hình. Giao diện Quản lý phòng chiếu (Tra cứu+Xóa phòng chiếu)

## Thêm phòng chiếu



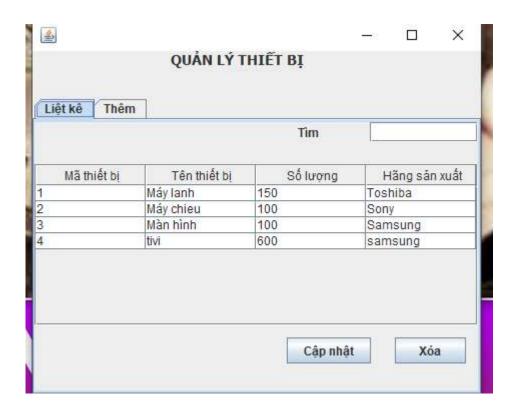
Hình . Giao diện thêm phòng chiếu

## Cập nhật phòng chiếu



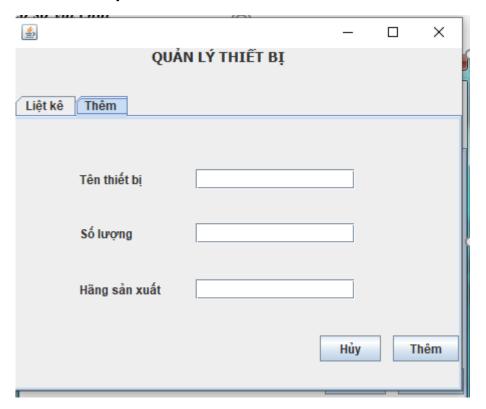
Hình . Giao diện cập nhật phòng chiếu

Quản lý Cơ sở vật chất (Tra cứu + Xóa Cơ sở vật chất)



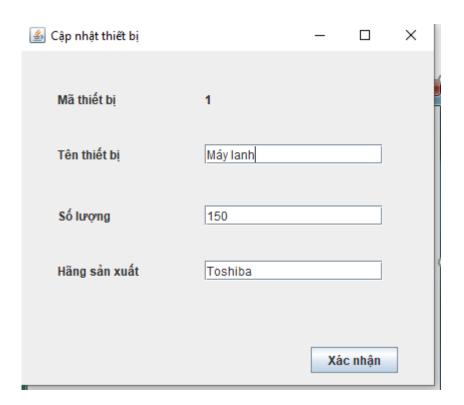
Hình .  $Giao\ diện\ Quản\ lý\ Cơ\ sở\ vật\ chất\ (Tra\ cứu + Xóa\ Cơ\ sở\ vật\ chất)$ 

## Thêm Cơ sở vật chất



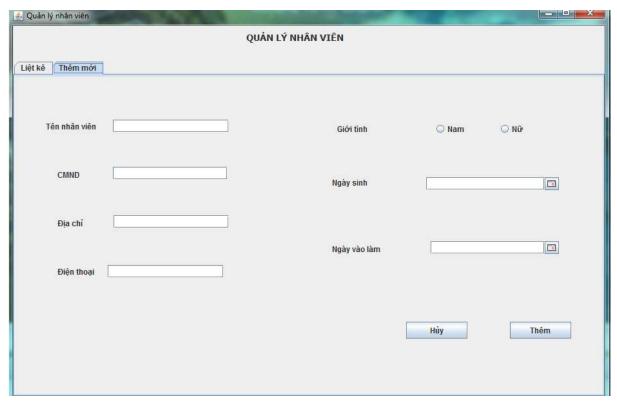
Hình . Giao diện thêm cơ sở vật chất

## Cập nhật Cơ sở vật chất



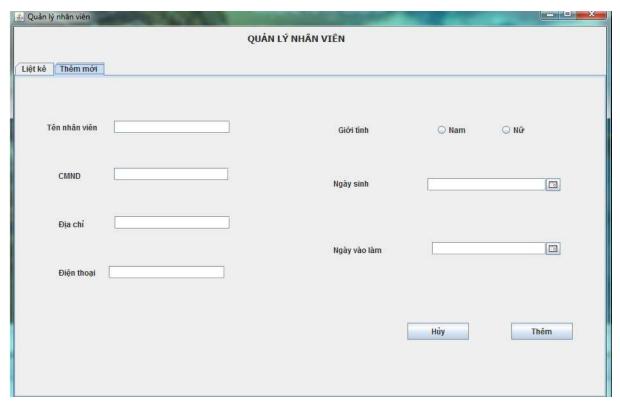
Hình . Giao diện cập nhật cơ sở vật chất

### Quản lý Nhân viên (Tra cứu + Xóa Nhân viên)



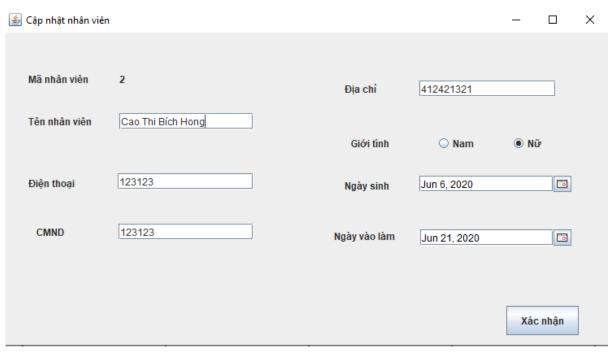
Hình . Giao diện Nhân viên (Tra cứu + Xóa Nhân viên)

#### Thêm nhân viên



Hình. Giao diện thêm nhân viên

### Cập nhật Nhân viên



Hình . Giao diện Cập nhật thông tin Nhân viên

# CHƯƠNG 6: KẾT LUẬN

# 6.1 Bảng Phân công việc nhóm

Công việc	Trần	Lê	Đồng
	Vĩnh	Minh	Xuân
	Phong	Thông	Danh
Thiết kế CSDL		X	X
Cài đặt CSDL		X	X
Viết báo cáo	X	X	
Trigger		X	X
Function			X
Stored Procedure	X	X	X
Lost Update	X	X	
Dirty Read		X	
Non Repeatable Read		X	
Phantom	X		
Conversion Deadlock	X	X	
Cycle Deadlock		X	X
Thiết kế giao diện			X
Form và Report			X

# $6.2~\mbox{Môi}$ trường phát triển ứng dụng:

Hệ điều hành Microsoft Windows 10

- Hệ quản trị cơ sở dữ liệu Oracle
- Ngôn ngữ lập trình: Java

## 6.3 Kết quả đạt được

Trong kỳ học vừa qua, nhóm đã tìm hiểu vận dụng kiến thức đã học về môn Hệ quản trị cơ sở dữ liệu, từ đó xây dựng phần mềm quản lý rạp chiếu phim. Về cơ bản, phần mềm bao gồm đầy đủ các chức năng như mục tiêu đề ra. Các chức năng hoàn thành như sau:

- Quản lý phim
- Quản lý vé
- Quản lý ghế ngồi
- Quản lý hóa đơn
- Báo cáo thống kê

### 6.4 Hạn chế

Lần đầu tiếp xúc với hệ quản trị Oracle, tuy đã cố gắng hoàn thiện các sơ đồ thiết kế dữ liệu, nhưng vì thời gian có hạn và kinh nghiệm chưa được tích lũy nhiều nên đồ án vẫn còn những hạn chế sau:

- Chưa vận dụng tốt các sơ đồ thiết kế phục vụ cho quá trình cài đặt phần mềm.
- Tốc độ xử lý còn chậm.
- Khả năng bảo mật chưa cao.
- Chưa thực hiện được chức năng Quên mật khẩu.
- Chưa hoàn thiện tất cả các chức năng ban đầu đưa ra.

## 6.5 Hướng phát triển

Để khắc phục những hạn chế nêu trên, nhóm đề xuất hướng phát triển như sau:

- Bổ sung và Tối ứu hóa các chức năng cho rạp phim
- Đưa ra doanh thu trước cho rạp phim ở một thời gian nhất định.
- Tối ưu chương trình và xây dựng hệ thống cung đặt vé qua mạng.

- Hoàn thiện chức năng sao lưu phục hồi trực tiếp không cần xuống mức CSDL.
- Tăng cường bảo mật cho hệ thống.
- Ứng dụng công nghệ để tin học hóa nhiều hơn các nghiệp vụ khác trong quy trình quản lý rạp phim.

# TÀI LIỆU THAM KHẢO

- 1. Giáo trình Hệ quản trị Cơ sở dữ liệu Oracle. Biên soạn Nguyễn Thị Trà Linh
- 2. Slide môn học Hệ quản trị Cơ sở dữ liệu. Khoa HTTT trường ĐH Công nghệ Thông Tin ĐHQGHCM.
- 3. Trang web về quản trị SQL:

http://www.sqlviet.com/blog/

4. Các bài viết về quản lý giao tác: <a href="https://viblo.asia/newest">https://viblo.asia/newest</a>