

**ĐẠI HỌC QUỐC GIA TP.HCM**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

🙞🙜🕮🙞🙜

****

ĐỒ ÁN MÔN HỌC

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

Đề tài:

**QUẢN LÝ HỆ THỐNG BÁN VÉ**

**RẠP CHIẾU PHIM**

*Lớp****:*** **IS210.K21**

*GVHD:* **Cô Đỗ Thị Minh Phụng**

*Các thành viên****:*** **1. Lê Thị Mỹ Diệu - MSSV 18520019**

**2. Trần Quốc Phong - MSSV 18520332**

**2. Phan Thành Long - MSSV 18521054**

**3. Đỗ Thị Thảo Quỳnh - MSSV 18521325**

TP.HCM, Tháng 4 - 2020

**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.

Cùng với sự phát triển của phần cứng máy tính, các phần mềm càng trở nên đa dạng, phong phú hỗ trợ hiệu quả hơn cho người sử dụng. Các phần mềm hiện nay ngày càng hỗ trợ cho người dung thuận tiện sử dụng, thời gian xử lí nhanh chóng và các nghiệp vụ được tự động hóa cao

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 chưa có 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 GVHD**

**MỤC LỤC**

**LỜI MỞ ĐẦU2**

**NHẬN XÉT GVHD3**

**MỤC LỤC4**

**PHẦN I: TỔNG QUAN ĐỀ TÀI DỰ ÁN7**

1.1 Phát biểu bài toán7

1.1.1 Vấn đề7

1.1.2 Giải pháp7

1.2 Tổng quan về đồ án8

1.2.1 Tổng quan8

1.2.2 Phạm vi, mục tiêu dự án8

1.2.3 Bố cục báo cáo10

1.3 Xác định và phân tích yêu cầu10

1.3.1 Phân tích yêu cầu10

1.3.2 Yêu cầu chức năng10

1.3.3 Yêu cầu phi chức năng11

1.4 Mô hình quan hệ12

1.4.1 Thiết kế mô hình quan hệ12

1.4.2 Bảng thuyết minh quan hệ và thuộc tính12

1.4.3 Mô tả ràng buộc toàn vẹn15

**PHẦN II: XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC18**

2.1 Trigger18

2.1.1 Trigger trong Oracle18

2.1.2 Danh sách các Trigger18

2.1.3 Mô tả một số Trigger18

2.2 Stored Procedure18

2.2.1 Stored Procedure trong Oracle18

2.2.2 Transaction trong Oracle18

2.2.3 Danh sách các Stored Procedure 18

2.2.4 Mô tả một số Stored Procedure 18

**PHẦN III: XỬ LÝ TRUY XUẤT ĐỒNG THỜI1**

3.1 Các mức cô lập trong Oracle2

3.1.1 Repeatable read3

3.1.2 Read committed3

3.1.3 Read uncommitted 3

3.1.4 Repeatable read3

3.1.5 Repeatable read3

3.2 Lost update2

3.3 Dirty read2

3.4 Unrepeatacle read2

3.5 Phantom2

3.6 Deadlock2

3.6.1 Conversion Deadlock3

3.6.2 Cycle Deadlock3

**PHẦN IV: THIẾT KẾ GIAO DIỆN1**

Type chapter title (level 2)2

Type chapter title (level 3)3

**PHẦN V: KẾT LUẬN1**

5.1 Bảng phân công công việc nhóm2

5.2 Môi trường phát triển ứng dụng2

5.3 Kết quả đạt được2

5.4 Hạn chế2

5.5 Hướng phát triển2

**TÀI LIỆU THAM KHẢO3**

**PHẦN I: TỔNG QUAN ĐỀ TÀI DỰ ÁN**

* 1. **Phát biểu bài toán**

***1.1.1 Vấn đề***

Những năm gần đây xu thế phát triển của thế giới đang dần phát triển lên thành một xã hội thông tin. Với sự phát triển đó, cũng kéo theo nhu cầu tăng cao và yêu cầu khắt khe của mọi người đối với các ngành nghề, bởi vậy, các dự án kinh doanh, sản xuất với nhiều qui mô khác nhau đang được tiến hành liên tục. Đặc biệt là các lĩnh vực gần gũi và không thể thiếu đối với đời sống con người như sức khoẻ, thực phẩm, giáo dục,...

Bên cạnh đó là sự bùng nổ về công nghệ với, đặc biệt với cuộc cách công nghiệp 4.0, khiến tầm quan trọng của công nghệ thông tin được đề cao hơn bao giờ hết, đã thúc đẩy việc ứng dụng các công nghệ, ứng dụng và phần mềm vào kinh doanh nhằm nâng cao chất lượng công việc, dịch vụ, tăng sức cạnh tranh với các dự án, doanh nghiệp khác trong nền kinh tế thị trường hiện nay.

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.

***1.1.2 Giải pháp***

Qua tìm hiểu và khảo sát tình hình thực tế, nhóm chúng em quyết định xây dựng một hệ thống quản lý bán vé chiếu phim với sự phát triển phần mềm đòi hỏi không chỉ là mức độ chính xác, xử lý 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.

Quy trình bán vé của rạp như sau: Hệ thống hỗ trợ bán vé cho khách hàng khi khách hàng cần mua vé có thể đến trực tiếp rạp bán vé hoặc đặt vé qua mạng hoặc qua điện thoại. Khách hàng sẽ được cung cấp đầy đủ thông tin về lịch chiếu và phim, sau đó sẽ lựa chọn suất chiếu mà mình muốn xem. Nếu suất chiếu đó còn ghế (chỗ) thì khách hàng sẽ được nhân viên tiến hành lập hoá đơn và lựa chọn các sản phẩm tại rạp sau đó in vé kiểm soát cho khách hàng. Khách hàng đã đặt vé phải đến mua vé chậm nhất một giờ trước khi giờ chiếu bắt đầu. Nếu trong thời gian ấy mà khách hàng không xác nhận thì coi như đã hủy việc đặt vé và phòng vé sẽ xóa tên trong danh sách đặt vé nếu cần. Nếu những suất chiếu không thể đáp ứng được thì thông báo cho khách hàng để họ có thể lựa chọn suất chiếu sau. Cuối mỗi ngày bộ phận kế toán in bảng tổng hợp tình hình bán vé trong ngày theo mẫu đã định.

**1.2 Tổng quan về đồ án**

***1.2.1 Tổng quan***

❖ Tên dự án: Quản lý hệ thống bán vé rạp chiếu phim Universal Cinemas

❖ Đơn vị thực hiện: Nhóm Universal

❖ Thành viên quản lý:

1. Trần Quốc Phong

2. Phan Thành Long

3. Đỗ Thị Thảo Quỳnh

4. Lê Thị Mỹ Diệu

❖ Thời gian thực hiện:

➢ *Bắt đầu:* ngày 29/03/2020

➢ *Kết thúc:* ngày 30/06/2020

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

***1.2.2 Phạm vi, mục tiêu dự án***

|  |  |
| --- | --- |
| *Mục tiêu dự án* | − Xây dựng phần mềm quản lý bán vé rạp chiếu phim đáp ứng yêucầu, phục vụ, hỗ trợ các thao tác nghiệp vụ của nhân viên,ban quản lý và khách hàng của Rạp chiếu phim **Universal Cinemas**.  − Cải tiến, nâng cao phần mềm với giao diện đẹp mắt, hiệu quảcao và đáp ứng tính bảo mật.  − Hoàn thành các công việc dự án trong phạm vi ngân sáchđược duyệt.  − Xây dựng và chuyển giao sản phầm trong vòng 3 tháng. |
| *Phạm vi dự án* | Xây dựng được hệ thống bán vé chiếu phim với đầy đủ các chức năng sau:  -Thành viên: cung cấp các thông tin đăng kí thành viên và các thông tin đặt vé, hoá đơn.  -Nhân viên: thêm, xoá, sửa, tra cứu nhân viên, quản lý các thông tin phim, rạp, hỗ trợ lập hoá đơn, báo cáo... giúp cho việc quản lý nhân viên được dễ dàng hơn.  -Tài khoản: lưu trữ thông tin tài khoản của nhân viên hay NQL hỗ trợ cho việc phân quyền các chức năng trong hệ thống, khi mỗi cá nhân được phân quyền đều có một tài khoản duy nhất.  -Sản phẩm: ngoài mua vé, rạp còn cung cấp các sản phẩm: thức ăn, đồ uống tại rạp cho khách hàng.  -Báo cáo: hỗ trợ báo cáo thống kê doanh thu theo ngày, tháng, thống kê doanh thu theo năm.  -Hoá đơn: hỗ trợ lưu thông tin các hoá đơn giúp khách hàng có thể có đầy đủ thông tin về các dịch vụ mình đã đặt mua và nhân viên có thể lưu lại ở hệ thống.  -Rạp chiếu: lưu trữ các thông tin phòng chiếu giúp nhân viên sắp xếp phòng chiếu một cách thông minh để không bị trùng lịch chiếu.  -Ghế: khách hàng ngồi tại nhà vẫn có thể đặt chỗ và giữ chỗ, không cần đến rạp trực tiếp.  -Loại phim, Phim: nhân viên có nhiệm vụ điền đầy đủ thông tin cần thiết theo yêu cầu của hệ thống để giúp khách hàng có thể dễ lựa chọn những thể loại, tên phim yêu thích.  -Suất chiếu: nhân viên có nhiệm vụ điền đầy đủ thông tin cần thiết theo yêu cầu của hệ thống để giúp khách hàng lựa chọn cho phù hợp với giờ xem của mình. |
| *Work – product* | − Hồ sơ khảo sát hiện trạng  − Tài liệu YCND  − Waterfall Model  − Hợp đồng kinh tế  − Source code  − Tài liệu kỹ thuật  − Testcase  − Test plan  − Báo cáo kiểm thử  − Bảng chi phí |
| *Phương pháp**phát triển* | − Waterfall Model |
| *Ngôn ngữ lập**trình* | − Sử dụng ngôn ngữ Java.  − Cơ sở dữ liệu Oracle. |
| *Chính sách* | − Khách hàng có quyền sửa yêu cầu sản phẩm trong quá trình thực hiện dự án nếu được sự đồng ý của giám đốc dự án.  − Không được sửa sản phẩm khi đã hoàn thành từ 70% trở lên.  − Nếu khách hàng yêu cầu thêm chức năng khác trong quán trình đang thực hiện sẽ có mức chi phí cụ thể riêng cho mỗi chức năng thêm đó.  − Sản phẩm được bàn giao cho khách hàng đúng tiến độ đề ra từ đầu dự án và có sự hỗ trợ khách hàng trong bước đầu sử dụng. |

***1.2.3 Bố cục báo cáo***

***Phần I:*** Tổng quan đề tài.

***Phần II:*** Xây dựng và quản lý giao tác.

***Phần III:*** Xử lý truy xuất đồng thời.

***Phần IV:*** Thiết kế giao diện.

***Phần V:*** Kết luận

* 1. **Xác định và phân tích yêu cầu**

***1.3.1 Phân tích yêu cầu***

-Thành viên: cung cấp các thông tin đăng kí thành viên và các thông tin đặt vé, hoá đơn.

-Nhân viên: thêm, xoá, sửa, tra cứu nhân viên, quản lý các thông tin phim, rạp, hỗ trợ lập hoá đơn, báo cáo.. giúp cho việc quản lý nhân viên được dễ dàng hơn.

-Tài khoản: lưu trữ thông tin tài khoản của nhân viên hay NQL hỗ trợ cho việc phân quyền các chức năng trong hệ thống, khi mỗi cá nhân được phân quyền đều có một tài khoản duy nhất.

-Sản phẩm: ngoài mua vé, rạp còn cung cấp các sản phẩm: thức ăn, đồ uống tại rạp cho khách hàng.

-Báo cáo: hỗ trợ báo cáo thống kê doanh thu theo ngày, tháng, thống kê doanh thu theo năm.

-Hoá đơn: hỗ trợ lưu thông tin các hoá đơn giúp khách hàng có thể có đầy đủ thông tin về các dịch vụ mình đã đặt mua và nhân viên có thể lưu lại ở hệ thống.

-Rạp chiếu: lưu trữ các thông tin phòng chiếu giúp nhân viên sắp xếp phòng chiếu một cách thông minh để không bị trùng lịch chiếu.

-Ghế: khách hàng ngồi tại nhà vẫn có thể đặt chỗ và giữ chỗ, không cần đến rạp trực tiếp.

-Loại phim, Phim: nhân viên có nhiệm vụ điền đầy đủ thông tin cần thiết theo yêu cầu của hệ thống để giúp khách hàng có thể dễ lựa chọn những thể loại, tên phim yêu thích.

-Suất chiếu: nhân viên có nhiệm vụ điền đầy đủ thông tin cần thiết theo yêu cầu của hệ thống để giúp khách hàng lựa chọn cho phù hợp với giờ xem của mình.

***1.3.2 Yêu cầu chức năng***

❖ 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.

− Để đảm bảo rằng việc quản lý rạp chiếu phim có hiệu quả hệ thống cần lưu trữ những thông tin:

+ Thông tin Rạp: số rạp, số ghế trong rạp...

+ Thông tin Khách hàng: họ tên khách hàng, ngày tháng năm sinh, CMND, số lần đã xem...

+ Thông tin Sản phẩm: số lượng sản phẩm bắp, nước còn lại.

+ Thông tin về vé và hóa đơn: mã vẽ, mã hoá đơn, các sản phẩm sử dụng…

❖ Yêu cầu về tính năng: Ngoài việc lưu trữ dữ liệu, hệ thống phải đáp ứng được các yêu cầu của khách hàng, nhân viên bán vé. Cụ thể như sau:

− Chức năng truy vấn: hệ thống phải luôn đáp ứng các yêu cầu truy vấn từ người dùng như:

-Tìm kiếm

+ Tra cứu thông tin vé theo mã vé.

+ Tra cứu thông tin vé, rạp còn trống.

+ Tra cứu thông tin khách hàng theo tên, ngày tháng năm sinh, CMND.

+ Tra cứu thông tin hóa đơn.

- Tính toán: Tính toán hoá đơn.

− Tính bảo mật và phân quyền người sử dụng:

• Mỗi nhân viên được cấp tài khoản và mật khẩu riêng để đăng

nhập vào hệ thống.

• Bộ phận quản lý hệ thống phân quyền cho từng nhân viên ở các

bộ phận khác nhau.

• Bộ phận quản lý và giám đốc được toàn quyền xử lý và giám sát

hệ thống

− Chức năng thống kê, báo cáo: giúp nhân viên quản lý theo dõi tình hình

chung của rạp chiếu phim:

• Thống kê doanh thu hằng ngày, hằng tuần, hằng tháng.

• Báo cáo số lượng, tỉ lệ các suất chiếu.

− Chức năng cập nhật:

• Hệ thống phải đảm bảo chức năng cập nhật được thuận tiện.

• Tránh tối đa các lỗi do sơ xuất của người sử dụng.

• Cập nhật số lượng vé còn lại và các suất ghế trống của suất chiếu.

***1.3.3 Yêu cầu phi chức năng***

❖ Yêu cầu về chất lượng:

- Tính tiến hóa: dễ dàng trong lúc nâng cấp

- Tính hiệu quả:

* Ứng dụng hoạt động ổn định và đáng tin cậy.
* Ứng dụng thể hiện tốt các loại vé, rạp, giống với mô tả.

- Tính tiện dụng:

* Hệ thống có giao diện trực quan, thân thiện và dễ sử dụng.
* Hướng dẫn sử dụng rõ ràng.

- Tính tương thích:

* Hoạt động tốt với các loại hệ điều hành phổ biến như (Android, iOS,…).
* Có Responsive theo từng kích thước màn hình khác nhau.

❖ Yêu cầu về giao diện:

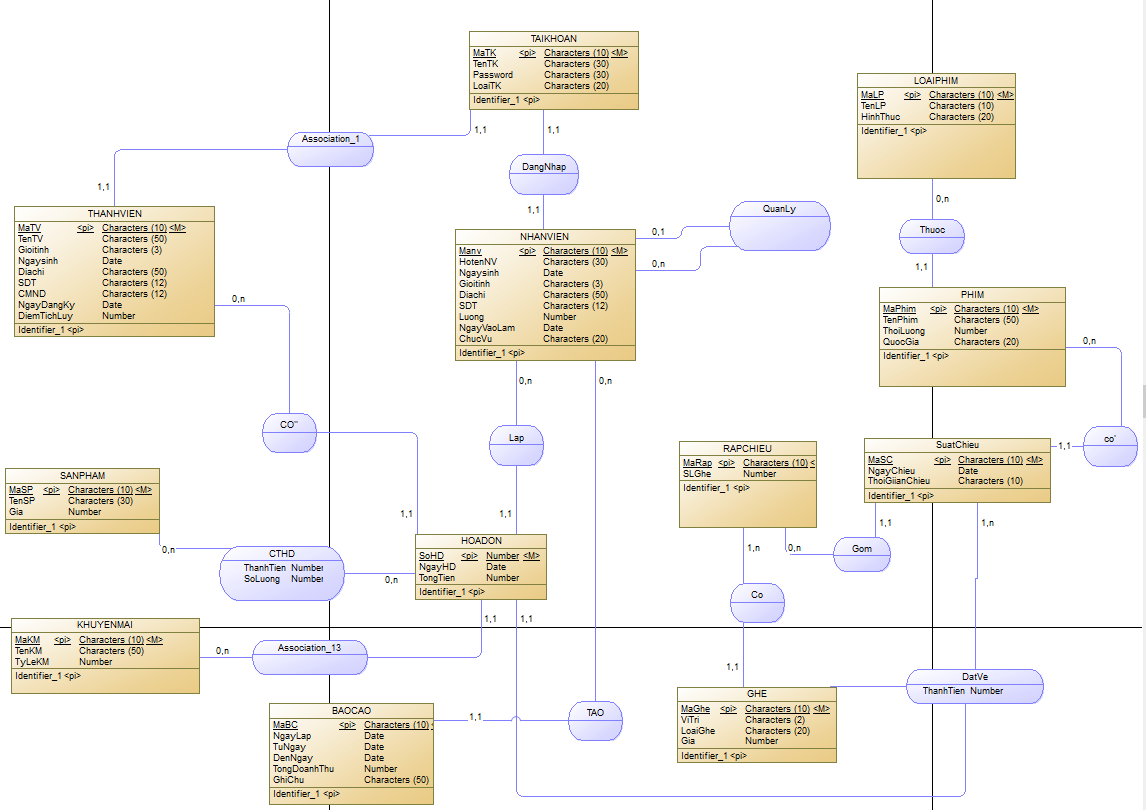
- Ưa nhìn, gần gũi, đơn giản.

- Hiển thị rõ ràng và chi tiết, nhất quán các thông tin.

- Sử dụng đa ngôn ngữ.

* 1. **Mô hình quan hệ**

***1.4.1 Thiết kế mô hình quan hệ***



***1.4.2 Bảng thuyết minh quan hệ và thuộc tí*nh**

1. Table: THANHVIEN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MaTV | Mã thành viên | varchar2(10) | 10 |  |  |
| 2 | Manv | Mã nhân viên | varchar2(10) | 10 |  |  |
| 3 | TenTV | Tên thành viên | varchar2(50) | 50 |  |  |
| 4 | GioiTinh | Giới tính | varchar2(3) | 3 |  |  |
| 5 | NgaySinh | Ngày sinh | date |  |  |  |
| 6 | DiaChi | Địa chỉ | varchar2(50) | 50 |  |  |
| 7 | SDT | Số điện thoại | varchar2(12) | 12 |  |  |
| 8 | CMND | CMND | varchar2(12) | 12 |  |  |
| 9 | NgayDangKy | Ngày đăng ký | date |  |  |  |
| 10 | DiemTichLuy | Điểm tích lũy | number |  |  |  |

1. Table: SANPHAM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MaSP | Mã sản phẩm | varchar2(10) | 10 |  |  |
| 2 | TenSP | Tên sản phẩm | varchar2(30) | 30 |  |  |
| 3 | Gia | Giá | number |  |  |  |

1. Table: TAIKHOAN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MaTK | Mã tài khoản | varchar2(10) | 10 |  |  |
| 2 | TenTK | Tên tài khoản | varchar2(30) | 30 |  |  |
| 3 | Password | Mật khẩu | varchar2(30) | 30 |  |  |
| 4 | LoaiTK | Loại tài khoản | varchar2(20) | 20 |  |  |

1. Table: NHANVIEN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | Manv | Mã nhân viên | varchar2(10) | 10 |  |  |
| 2 | HotenNV | Họ tên nhân viên | varchar2(30) | 30 |  |  |
| 3 | NgaySinh | Ngày sinh | date |  |  |  |
| 4 | GioiTinh | Giới tính | varchar2(3) | 3 |  |  |
| 5 | DiaChi | Địa chỉ | varchar2(50) | 50 |  |  |
| 6 | SDT | Số điện thoại | varchar2(12) | 12 |  |  |
| 7 | Luong | Lương | number |  |  |  |
| 8 | NgayVaoLam | Ngày vào làm | date |  |  |  |
| 9 | ChucVu | Chức vụ | varchar2(20) | 20 |  |  |

1. Table: HOADON

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | SoHD | Số hóa đơn | number |  |  |  |
| 2 | NgayHD | Ngày tạo hóa đơn | date |  |  |  |
| 3 | TongTien | Mã nhân viên | varchar2(10) | 10 |  |  |

1. Table: BAOCAO

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MaBC | Mã báo cáo | varchar2(10) | 10 |  |  |
| 2 | NgayLap | Ngày lập | date |  |  |  |
| 3 | TuNgay | Từ ngày | date |  |  |  |
| 4 | DenNgay | Đến ngày | date |  |  |  |
| 5 | TongDoanhThu | Tổng doanh thu | number |  |  |  |
| 6 | GhiChu | Ghi chú | varchar2(50) | 50 |  |  |

1. Table: RAPCHIEU

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MaRap | Mã rạp chiếu | varchar2(10) | 10 |  |  |
| 2 | SLGhe | Số lượng ghế | number |  |  |  |

1. Table: GHE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MaGhe | Mã ghế | varchar2(10) | 10 |  |  |
| 2 | ViTri | Vị trí | varchar2(2) | 2 |  |  |
| 3 | LoaiGhe | Loại ghế | varchar2(20) | 20 |  |  |
| 4 | GiaTien | Giá tiền | number |  |  |  |

1. Table: LOAIPHIM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MaLP | Mã loại phòng | varchar2(10) | 10 |  |  |
| 2 | TenLP | Tên loại phòng | varchar2(10) | 10 |  |  |
| 3 | HinhThuc | Hình thức | varchar2(20) | 20 |  |  |

1. Table: PHIM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MaPhim | Mã phim | varchar2(10) | 10 |  |  |
| 2 | TenPhim | Tên phim | varchar2(50) | 50 |  |  |
| 3 | ThoiLuong | Thời lượng chiếu | number |  |  |  |
| 4 | QuocGia | Quốc gia | varchar2(20) | 20 |  |  |

1. Table: SuatChieu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MaSC | Mã suất chiếu | varchar2(10) | 10 |  |  |
| 2 | NgayChieu | Ngày chiếu | date |  |  |  |
| 3 | ThoiGianChieu | Thời gian chiếu | date |  |  |  |

1. Table: KHUYENMAI

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | MaKM | Mã khuyến mãi | varchar2(10) | 10 |  |  |
| 2 | TenKM | Tên khuyến mãi | varchar2(50) | 50 |  |  |
| 3 | TyLeKM | Tỷ lệ khuyến mãi | number |  |  |  |

***1.4.3 Mô tả ràng buộc toàn* vẹn**

1. *R1: “Giới tính của nhân viên chỉ là nam hoặc nữ”*

Bối cảnh: NHANVIEN

RBTV miền giá trị

∀ nv ∈ NHANVIEN: nv.Gioitinh ∈ {‘Nam’,’Nữ’}

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Xóa | Sửa |
| NHANVIEN | + | - | +(Gioitinh) |

1. *R2: “Giới tính của thành viên chỉ là nam hoặc nữ”*

RBTV miền giá trị

Bối cảnh: THANHVIEN

∀ tv ∈ THANHVIEN: tv.Gioitinh ∈ {‘Nam’,’Nữ’}

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | Thêm | Xóa | Sửa |
| THANHVIEN | + | - | +(Gioitinh) |

1. *R3: “Lương nhân viên tăng chứ không giảm*”

Bối cảnh: NHANVIEN

RBTV miền giá trị

∀ nv ∈ NHANVIEN: nv.Luong mới > nv.Luong cũ

|  |  |  |  |
| --- | --- | --- | --- |
| R3 | Thêm | Xóa | Sửa |
| NHANVIEN | + | - | +(Luong) |

1. *R4: “Ghế chỉ có 2 trạng thái: trống, đã đặt”*

RBTV miền giá trị

Bối cảnh: GHE

∀ gh ∈ GHE: gh.TrangThai ∈ {“Trống”, “Đã đặt”}

|  |  |  |  |
| --- | --- | --- | --- |
| R4 | Thêm | Xóa | Sửa |
| GHE | + | - | +(Trangthai) |

1. *R5: “Hạn sử dụng của sản phẩm phải lớn hơn Ngày sản xuất”*

RBTV liên thuộc tính

Bối cảnh: SANPHAM

∀ sp ∈ SANPHAM: sp.HSDsp.NSX

|  |  |  |  |
| --- | --- | --- | --- |
| R6 | Thêm | Xóa | Sửa |
| SANPHAM | + | - | +(HSD, NSX) |

1. *R6: “TuNgay phải nhỏ hơn DenNgay và NgayLap phải lớn hơn hoặc bằng DenNgay”*

RBTV liên thuộc tính

Bối cảnh: BAOCAO

∀ bc ∈ BAOCAO: bc.TuNgay<bc.DenNgaybc.NgayLap

|  |  |  |  |
| --- | --- | --- | --- |
| R7 | Thêm | Xóa | Sửa |
| BAOCAO | + | - | +(NgayLap, TuNgay, DenNgay) |

1. *R7:”Ngày chiếu phải lớn hơn hoặc bằng ngày công chiếu”*

RBTV liên thuộc tính, liên quan hệ

Bối cảnh: PHIM, SUATCHIEU

∀ p ∈ PHIM, ∀ sc ∈ SC: p.MaPhim=sc.MaPhim ^ sc.NgayChieup.NgayCongChieu

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Xóa | Sửa |
| PHIM | - | + | +(NgayCongChieu) |
| SUATCHIEU | + | - | +(MaPhim, NgayChieu) |

1. *R8: “Khi một thành viên mua hàng, ngày hóa đơn phải lớn hơn hoặc bằng ngày đăng kí thành viên”*

RBTV liên thuộc tính, liên quan hệ

Bối cảnh: THANHVIEN, HOADON

∃tv ∈ THANHVIEN, ∃ hd ∈ HD: tv.MaTV=hd.MaTV ^ tv.NgayDangKyhd.NgayHD

|  |  |  |  |
| --- | --- | --- | --- |
| R9 | Thêm | Xóa | Sửa |
| THANHVIEN | - | - | +(NgayDangKy) |
| HOADON | + | - | +(NgayHD) |

1. *R9: “Ngày sinh của thành viên phải nhỏ hơn ngày đăng kí thành viên”*

RBTV liên thuộc tính

Bối cảnh: THANHVIEN

∀ tv ∈ THANHVIEN: tv.Ngaysinh<tv.NgayDangKy

|  |  |  |  |
| --- | --- | --- | --- |
| R10 | Thêm | Xóa | Sửa |
| THANHVIEN | + | - | +(Ngaysinh, NgayDangKy) |

1. *R10: “Ngày sinh của nhân viên phải nhỏ hơn vào làm của nhân viên và ngày vào làm phải nhỏ hơn hoặc bằng ngày lập báo cáo”*

RBTV liên thuộc tính, liên quan hệ

Bối cảnh: NHANVIEN, BAOCAO

∀ bc ∈ BC, ∃ nv ∈ NHANVIEN: bc.Manv=nv.Manv ^ bc.NgayLapnv.NgayVL > nv.NgaySinh

|  |  |  |  |
| --- | --- | --- | --- |
| R11 | Thêm | Xóa | Sửa |
| NHANVIEN | - | - | +(NgayVL, NgaySinh) |
| BAOCAO | + | - | +(Manv, TuNgay) |

1. *R11: “Tất cả các phim phải có tên phim phân biệt với nhau”*

RBTV liên bộ

Bối cảnh: PHIM

∀ p1, p2 ∈ PHIM: p1p2 p1.TenPhimp2.TenPhim

|  |  |  |  |
| --- | --- | --- | --- |
| R12 | Thêm | Xóa | Sửa |
| PHIM | + | - | +(TenPhim) |

1. *R12: “Mã người quản lí là mã của một nhân viên trong công ty”*

RBTV tham chiếu

Bối cảnh: NHANVIEN

*∀* nv ∈ NHANVIEN: (nv.MaNQL = NULL) hoặc (∃ nv’ ∈ NHANVIEN: nv.MaNQL = nv’.MaNV)

|  |  |  |  |
| --- | --- | --- | --- |
| R12 | Thêm | Xóa | Sửa |
| NHANVIEN | + | + | +(MaNQL) |

1. *R13: “Chỉ có nhân viên là người quản lí mới được lập báo cáo”*

RBTV chu trình

Bối cảnh: NHANVIEN, BAOCAO

∀ bc ∈ BAOCAO, ∃ nv1, nv2 ∈ NHANVIEN: (bc.MaNV = nv1.Manv) và (nv1.Manv=nv2.MaNQL)

|  |  |  |  |
| --- | --- | --- | --- |
| R13 | Thêm | Xóa | Sửa |
| NHANVIEN | - | - | +(MaNQL) |
| BAOCAO | + | - | +(Manv) |

**PHẦN II. XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC**

**2.1. Trigger**

***2.1.1. Trigger trong Oracle***

− Trigger là một đối tượng được định danh trong cơ sở dữ liệu và được gắn chặt với một sự kiện xảy ra trên một bảng nào đó (điều này có nghĩa là nó sẽ được tự động thực thi khi xảy ra một sự kiện trên một bảng). Các sự kiện này bao gồm: chèn (Insert), xóa (Delete) hay cập nhật (Update) một bảng.

− Trigger được sử dụng để tạo ra các công việc tự động thay cho việc phải làm thủ công bằng tay như: kiểm tra dữ liệu, đồng bộ hóa dữ liệu, đảm bảo các mối quan hệ giữa các bảng, ...

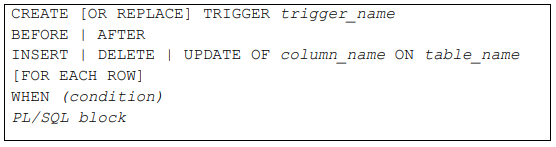
− Lý do chính cho việc dùng Trigger là để khai báo các ràng buộc toàn vẹn phức tạp, đó có thể là những ràng buộc mà không thể khai báo ở cấp Table như ràng buộc NOT NULL, UNIQUE KEY, PRIMARY KEY, FOREIGN KEY, CHECK.

− Lưu ý:

+Bạn không thể tạo Before Trigger cho View

+Bạn không thể cập nhật dữ liệu cũ, chỉ co thể cập nhật dữ liệu mới.

− Cú pháp của Trigger trong Oracle:



*Trong đó:*

+ Từ khóa **OR REPLACE** để tự động xóa và tạo mới Trigger nếu Trigger đó đã tồn tại.

+ **table\_name** để chỉ đến tên của table muốn tạo trigger.

+ **INSERT | DELETE | UPDATE** ứng với sự kiện tác động lên table để trigger tự động thi hành khi sự kiện đó xảy ra.

+ **AFTER** chỉ rằng database trigger sẽ thi hành sau khi đã thực hiện sự kiện và **BEFORE** là để khia báo trigger sẽ thi hành trước khi thi hành sự kiện.

+ Chọn lựa **FOR EACH ROW** để chỉ rằng trigger sẽ thi hành khi câu lệnh SQL tác động lên từng dòng.

− Những chú ý khi tạo Trigger:

+ Phần thân trigger có thể chứa các lệnh DML, nhưng lệnh SELECT phải là SELECT INTO ngoại trừ lệnh SELECT khi khai báo cursor.

+ DDL không được dùng trong phần thân của Trigger.

+ Không cho phép các lệnh quản lý giao tác (COMMIT, ROLLBACK, SAVEPOINT) trong phần thân của Trigger.

+ Nếu Trigger gọi một chương trình con thì chương trình con đó không được chứa các lệnh quản lý giao tác.

− Thao tác Trigger:

+ **Lệnh disable trigger**: ALTER TRIGGER *trigger\_name* DISABLE; Để disable tất cả các trigger liên quan đến một table cụ thể, dùng lệnh: ALTER TABLE *table\_name* DISABLE ALL TRIGGERS;

+ **Lệnh enable trigger**: ALTER TRIGGER *trigger\_name* ENABLE; Để enable tất cả các trigger liên quan đến một table cụ thể, dùng lệnh: ALTER TABLE *table\_name* ENABLE ALL TRIGGERS;

− Xóa Trigger:

*Cú pháp:*



***2.1.2. Danh sách các Trigger***

***2.1.3. Mô tả một số Trigger***

**2.2 Stored Procedure**

***2.2.1 Stored Procedure trong Oracle***

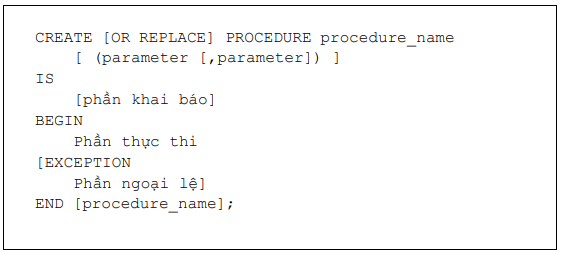
− Stored procedure là một hoặc một tập các lệnh SQL được biên dịch để thực hiện cùng lúc mỗi khi được gọi đến. Về bản chất giống với hàm, thủ tục trong các ngôn ngữ lập trình khác.

− Stored procedure giúp giảm thời gian giao tiếp giữa các ứng dụng với hệ quản trị Oracle, bởi vì thay vì gửi nhiều câu lệnh dài thì ta chỉ cần gọi tới một thủ tục và trong thủ tục này sẽ thực hiện nhiều câu lệnh SQL.

− Stored procudure sẽ giúp các ứng dụng nhìn minh bạch hơn, nghĩa là khi ta định nghĩa các thao tác xử lý vào một Stored thì công việc của các ngôn ngữ lập trình khác chỉ quan tâm đến tên thủ tục, các tham số truyền vào chứ không cần biết nó thực hiện như thế nào. Mỗi thủ tục sẽ có các mức độ truy cập, nghĩa là ta có thể cấp quyền sử dụng cho một User nào đó trong hệ quản trị.

− Stored procedure cũng có một số nhược điểm. Nếu tạo ra quá nhiều Procedure thì hệ quản trị sẽ sử dụng bộ nhớ để lưu trữ các thủ tục này khá nhiều. Ngoài ra nếu thực hiện quá nhiều xử lý trong mỗi thủ tục thì đồng nghĩa với việc CPU sẽ làm việc nặng hơn.

− Cú pháp khai báo Stored procedure:



*Trong đó:*

+ Từ khóa OR REPLACE sẽ tự động xóa procedure và tạo mới nếu nó đã tồn tại.

+ parameter được thay thế bới:

variable\_name [IN | OUT | IN OUT] datatype [{ := | DEFAULT} value]

− Cấu trúc của một thủ tục:

Một thủ tục có hai phần: Phần đặc tả và phần thân thủ tục.

➢ Phần đặc tả bắt đầu từ khóa PROCEDURE và kết thúc bởi tên của thủ tục hoặc danh sách tham số.

➢ Phần thân của thủ tục bắt đầu từ từ khóa IS và kết thúc bởi từ khóa END. Phần thân của thủ tục được chia thành ba phần sau:

✓ Phần khai báo

✓ Phần thi hành lệnh

✓ Phần xử lý ngoại lệ

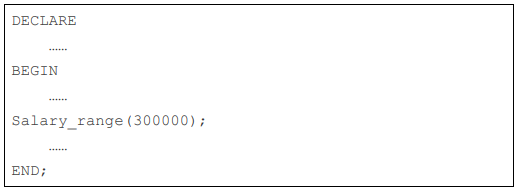
➢ Phần khai báo chứa các khai báo cục bộ, nằm giữa từ khóa IS và BEGIN. Từ khóa DECLARE không được dùng để khai báo trong thủ tục, nó chỉ dùng để khai báo trong khối PL/SQL mà thôi.

➢ Phần thi hành lệnh chứa các câu lệnh nằm trong khoảng giữa BEGIN và EXCEPTION (hoặc END). Tối thiểu phải có một câu lệnh trong phần này.

➢ Phần xử lý ngoại lệ chứa các xử lý ngoại lệ.

➢ Procedure trong Oracle được gọi là Stored procedure vì procedure được stored trong cơ sở dữ liệu và được gọi bởi bất cứ một khối PL/ SQL nào. Trước khi thủ tục được tạo, Oracle sẽ phân tích cú pháp lệnh và biên dịch thủ tục đó. Sau đó, phần mã lệnh được biên dịch sẽ được stored trong cơ sở dữ liệu.

− Gọi thủ tục:



− Gọi một thủ tục từ SQL \* Plus



− Xóa thủ tục:



***2.2.2 Transaction trong Orcale***

*2.2.2.1 Transaction*

- Một giao tác trên cơ sở dữ liệu bao gồm một chuỗi các thay đổi trên một hay nhiều table.

- Một giao tác chứa một trong những lệnh sau:

+ Các lệnh DML (Data Manipulation Language) mà tạo ra sự thay đổi dữ liệu.

+ Một lệnh DDL (Data Definition Language).

+ Một lệnh DCL (Data Control Language).

+ Một giao tác bắt đầu khi câu lệnh SQL đầu tiên được thi hành.

* Một giao tác kết thúc khi xảy ra một trong những sự kiện sau:

+ Khi có lệnh COMMIT hay ROLLBACK phát ra.

+ Khi thi hành một lệnh ĐL hoặc DCL (các lệnh này tự động COMMIT)

+ Khi có lỗi thi hành lệnh, lỗi hệ thống.

+ Khi người dùng kết thúc SQL \* Plus

*2.2.2.2. Điều khiển giao tác bằng lệnh SQL*

**❖** Lệnh **COMMIT**

Cú pháp: COMMIT

Tình trạng dữ liệu trước khi COMMIT:

− User hiện hành nhìn thấy kết quả thay đổi trong câu lệnh DML nhưng các user khác thì không nhìn thấy sự thay đổi đó.

− Các dòng bị thay đổi sẽ bị khóa.

Tình trạng dữ liệu sau khi COMMIT:

− Lưu các thay đổi dữ liệu vào database.

− Hủy tất cả các savepoints.

− Giải phóng cơ chế khóa của giao tác.

− Kết thúc giao tác thành công.

❖ Lệnh **ROLLBACK**

Cú pháp: ROLLBACK [TO SAVEPOINT name]

Tình trạng dữ liệu trước khi ROLLBACK:

− User hiện hành nhìn thấy kết quả thay đổi trong câu lệnh DML nhưng các user khác thì không nhìn thấy sự thay đổi đó.

− Các dòng bị thay đổi sẽ bị khóa.

Tình trạng dữ liệu sau khi ROLLBACK:

− Hủy bỏ sự thay đổi giao tác hiện hành.

− Hủy tất cả các savepoints.

− Giải phóng cơ chế khóa của giao tác.

❖ Lệnh **SAVEPOINT**

Cú pháp: SAVEPOINT savepoint\_name

Mục đích tạo savepoint:

− Để chia nhỏ giao tác.

− Chủ động điều khiển điểm ROLLBACK

Nếu tạo savepoint thứ hai trùng tên với savepoint thứ nhất thì savepoint thứ nhất sẽ bị loại bỏ. Số lượng savepoint tối đa là 5.

❖ Chế độ **AUTOCOMMIT**

− Lệnh SET AUTOCOMMIT ON: Tự động COMMIT khi mỗi lệnh INSERT, UPDATE, DELETE phát ra.

− Lệnh SET AUTOCOMMIT OFF: Không tự động commit. Commit chỉ phát ra khi một lệnh ĐL, DCL phát ra hoặc khi thoát khỏi SQL \* Plus.

***2.2.3 Danh sách Stored procedure***

***2.2.4 Mô tả một số Stored procedure***

**PHẦN III: XỬ LÝ TRUY XUẤT ĐỒNG THỜI**

**3.1. Các mức cô lập trong Oracle**

SQL Transaction isolation là một nền tảng trong tiến trình cơ sở dữ liệu. Isolation thể hiện từ I trong cụm từ viết vắt ACID. Mức cô lập là sự cài đặt tinh chỉnh sự cân bằng giữa hiệu suất và độ xác thực, nhất quán khi nhiều transaction thực hiện các truy cấn trong cùng một thời điểm. InnoDB cung cấp 4 mức cô lập: read uncommitted, read committed, repeatable read and serializable. Mức cô lập mặc định của InnoDB là repeatable read. Người dùng có thể thay đổi mức cô lập cho một session hoặc cho tất cả các connection tiếp theo thông qua lệnh SET TRANSACTION. Để thiết lập mức cô lập mặc định cho tất cả các connection, sử dụng transaction-isolation. InnoDB hỗ trợ mỗi mức cô lập sử dụng cơ chế khóa khác nhau. Bạn có thể thiết lập mức cô lập cao hơn mặc định repeatable read cho những tiến trình mà ACID là yêu cầu quan trọng. Hoặc bạn có thể thoải mái hơn trong việc đảm bảo sự nhất quán dữ liệu thông qua việc sử dụng read commited hoặc thậm chí là read uncommitted trong trường hợp số lượng dữ liệu quá lớn. Serializable thi hành cơ chế khóa nghiêm ngặt hơn repeatable read và được sử dụng trong những trường hợp đặc biệt như transaction XA và giải quyết sự cố tranh chấp đồng thời và deadlock. Sau đây sẽ là mô tả về những mức cô lập mà Oracle hỗ trợ. Mô tả sẽ theo thứ tự từ mức cô lập thường được sử dụng nhất cho đến ít sử dụng nhất.

***3.1.1. 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.

***3.1.2. 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 môt lần nữa và lúc này InnoDB vừa khóa nó và đợi một khóa trên nó.

***3.1.3. 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.

***3.1.4. 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.

***3.1.5.*** ***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 đợi 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.

**3.2 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  T1 bán vé | Thêm vé xem phim khi nhân viên bán vé  T2 bán vé |
|  |  |

**3.3 Dirty read**

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

|  |  |
| --- | --- |
| Quản trị viên cập nhật thông tin phim T1 | Nhân viên xem thông tin phim T2 |
| create or replace procedure  SuaTTPhim1(v\_dotuoi NUMBER,  v\_maphim varchar2)  as  begin  update phim  set dotuoi=v\_dotuoi  where maphim= v\_maphim;  DBMS\_LOCK.SLEEP(5);  ROLLBACK;  end; | create or replace PROCEDURE  XEMTTPHIM1 (V\_MAPHIM  VARCHAR2)  AS  V\_DOTUOI  PHIM.DOTUOI%TYPE;  BEGIN  SELECT DOTUOI INTO  V\_DOTUOI  FROM PHIM  WHERE MAPHIM=V\_MAPHIM;  DBMS\_OUTPUT.PUT\_LINE('DO  TUOI:'||V\_DOTUOI);  COMMIT;  END; |

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.

**3.4 Unrepeatable read**

***Tình huống:*** Trong hệ thống, khi khách hàng đang xem thông tin phim thì nhân viên quản lý phim cập nhật lại tên phim.

|  |  |
| --- | --- |
| Xem thông tin phim T1 | Cập nhật thông tin phim T2 |
|  |  |

**3.5 Phantom**

***Tình huống 1***: Trong hệ thống, tại cùng một thời điểm, hai nhân viên bán vé cho hai khách hàng cùng một suất chiếu và cùng một ghế. Một trục trặc hệ thống khiến cho việc đặt vé của nhân viên 1 chưa được commit khiến cho nhân viên 2 đặt vé thành công, và sau đó nhân viên 1 vẫn đặt được vé => xung đột trong dữ liệu.

|  |  |  |
| --- | --- | --- |
| Nhân viên 1 (T1) | Nhân viên 2 (T2) | Giải thích |
| exec SP\_DATVE\_XULY('e5r1','1sc','NV1',null, 65000,null,0); |  | T1 tiến hành đặt vé |
| V\_GHEDADAT:= F\_CHECK\_SEAT(‘e5r1’, ‘1sc’); |  | T1 kiểm tra xem ghế ‘e5r1’ thuộc suất chiếu ‘1sc’ đã được đặt chưa. |
| F\_CREATE\_HOADON\_DATVE(‘NV1’, null,65000,null,0); |  | Vì ghế ‘e5r1’ thuộc suất chiếu ‘1sc’ chưa được ai đặt nên T1 tiến hành tạo hóa đơn 207 cho lần đặt vé này. |
| INSERT INTO DATVE VALUES(‘e5r1’, ‘1sc’, 207); |  | T1 tiến hành đặt vé cho khách hàng với số hóa đơn 207 |
|  | exec SP\_DATVE\_XULY('e5r1','1sc','NV2',null,65000,null,0); | T2 tiến hành đặt vé |
|  | V\_GHEDADAT:= F\_CHECK\_SEAT(‘e5r1’, ‘1sc’); | T2 kiểm tra ghế đã được đặt hay chưa. Lúc này T1 vẫn chưa commit dữ liệu nên ghế vẫn chưa có trong bảng đặt vé. |
|  | F\_CREATE\_HOADON\_DATVE(‘NV2’, null,65000,null,0); | T2 tiến hành tạo hóa đơn 208 cho lần đặt vé này. |
|  | INSERT INTO DATVE VALUES(‘e5r1’, ‘1sc’, 207); | T2 tiến hành đặt vé cho hóa đơn 208 |
|  | Commit; | T2 commit dữ liệu, kết thúc transaction |
|  | select \* from DATVE; | Ghế e5r1 đã được đặt thành công. |
| Commit; |  | T1 commit dữ liệu, kết thúc transacion. |
| select \* from DATVE; |  | Ghế e5r1 đã được đặt thành công. |

Cả hai transaction đều đặt ghế thành công dù không được điều này không được phép do procedure đã kiểm tra xem ghế đã được đặt trước đó hay chưa.

Giải pháp:

***Tình huống 1***: Trong hệ thống, khi quản trị viên đang cập nhật tỷ lệ khuyến mãi thì tại thời điểm đó nhân viên quản lý khuyến mãi thêm khuyến mãi mới.

|  |  |
| --- | --- |
| Cập nhật tỷ lệ khuyến mãi T1 | Thêm thông tin khuyến mãi T2 |
|  |  |

***Tình huống 2:*** Khi quản trị viên đang xem đang xem điểm tích lũy của khách hàng thì cùng lúc đó khách hàng mua vé và được cộng thêm điểm tích lũy

|  |  |
| --- | --- |
| Quản trị viên xem điểm tích lũy T1 | Nhân viên thu ngân bán vé T2 |
|  |  |

**3.6 Deadlock**

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

***3.6.2 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 |
|  |  |

**PHẦN IV: THIẾT KẾ GIAO DIỆN**

|  |  |
| --- | --- |
| **STT** | **TÊN GIAO DIỆN** |
|  |  |

*Bảng giao diện*

**PHẦN V: KẾT LUẬN**

**5.1 Bảng Phân công việc nhóm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CÔNG VIỆC** | **Trần Quốc Phong** | **Lê Thị Mỹ Diệu** | **Phan Thành Long** | **Đỗ Thị Thảo Quỳnh** |
| Thiết kế CSDL |  |  |  |  |
| Cài đặt CSDL |  |  |  |  |
| Viết báo cáo |  |  |  |  |
| Trigger |  |  |  |  |
| Function |  |  |  |  |
| Stored Procedure |  |  |  |  |
| Lost Update |  |  |  |  |
| Dirty Read |  |  |  |  |
| Non-Repeatable Read |  |  |  |  |
| Phantom |  |  |  |  |
| Conversion Deadlock |  |  |  |  |
| Cycle Deadlock |  |  |  |  |
| **Thiết kế giao diện** | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**5.2 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

**5.3 Kết quả đạt được**

− Quản lý rạp chiếu phim sử dụng hệ thống thông tin không phải là một hướng đi mới.

− Tuy nhiên, hiện tại vẫn còn rất nhiều cơ sở kinh doanh rạp chiếu phim nhỏ chỉ quản lý bằng cách thủ công. Nhận thấy đây là một đề tài hay và thiết thực có thể ứng dụng rộng rãi. Chính vì vậy nhóm tác giả đã quyết định chọn đề tài nhằm tạo ra hệ thống quản lý rạp chiếu phim giúp ích cho việc quản lý đơn giản hơn, tiện ích hơn và giảm tải lượng công việc thủ công.

− Dưới đây là những chức năng cơ bản nhóm đã đạt được:

• Bán vé và thanh toán

• Quản lý khuyến mãi

• Quản lý sự kiện

• Quản lý sản phẩm

• Quản lý ca chiếu

• Quản lý phim

• Quản lý phòng chiếu

• Quản lý cơ sở vật chất

• Phân quyền truy cập

• Lập báo cáo doanh thu

**5.4 Hạn chế**

Vì thời gian nghiên cứu và hiện thực đề tài còn giới hạn nên đề tài còn có một số hạn chế, như sau:

− Chức năng Sao lưu – Phục hồi còn thủ công trên hệ quản trị CSDL, mà chưa trở thành một chức năng cụ thể của phần quản lý.

− Chưa thực hiện được chức năng Quên mật khẩu.

− Chưa hoàn thiện tất cả các chức năng ban đầu đưa ra.

Chính vì vậy, đề tài chỉ được dừng lại ở mức đồ án môn học và chưa thể đạt được một số mục tiêu đã đề ra lúc đầu.

**5.5 Hướng phát triển**

Như đã trình bày ở các phần trước, nhóm tác giả nhận định còn rất nhiều việc cần phải làm để hoàn thiện hệ thống quản lý rạp chiếu phim. Chính vì vậy nhóm đặt ra mục tiêu để phát triển đề tài trong tương lai như sau:

− 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**

**Tiếng anh:**

1. *[Undergraduate Topics in Computer Science] Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel (auth.) - UML @ Classroom\_ An Introduction to Object-Oriented Modeling (2015, Springer International Publishing)*

**Tiếng việt:**

*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.*