**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH  TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**https://lh6.googleusercontent.com/zea-7LUSL2I-GtMsKco5l-ljlMQJdIzdhX6i8u9WPiCg7GlHWS7zTbqJ-eeHIrwBzuGDrZVzWmwCe9ZPWJy1d447j6DrjmI5aV-qeKdOToeRw6qxig9gF9TkVEM2CvItWNUQOOvS**

KHOA HỆ THỐNG THÔNG TIN

**ĐỒ ÁN MÔN HỌC**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

Đề tài : **QUẢN LÝ KHÁCH SẠN**

Giảng viên hướng dẫn : **Đỗ Thị Minh Phụng**

|  |  |  |
| --- | --- | --- |
| **Sinh viên thực hiện:** | | |
| STT | Họ tên | MSSV |
| 1 | Đặng Vũ Phương Uyên | 19520345 |
| 2 | Nguyễn Thị Thu Phương | 19522066 |
| 3 | Lê Hữu Thắng | 19522200 |
| 4 | Nguyễn Thái Toàn | 19622366 |

**TP. HỒ CHÍ MINH – 03/2021**

**MỤC LỤC**

[CHƯƠNG 1 : TỔNG QUAN ĐỀ TÀI 1](#_Toc76678441)

[1. Đặt vấn đề: 1](#_Toc76678442)

[2. Mục tiêu: 2](#_Toc76678443)

[CHƯƠNG 2 : PHÂN TÍCH VÀ YÊU CẦU DỮ LIỆU 3](#_Toc76678444)

[I. Đặc tả bài toán: 3](#_Toc76678445)

[II. Phân tích yêu cầu: 3](#_Toc76678446)

[**1.** **Yêu cầu phi chức năng:** 3](#_Toc76678447)

[**2.** **Thiết kế mô hình quan hệ:** 4](#_Toc76678448)

[a. Mô hình thực thể mối kết hợp: 4](#_Toc76678449)

[b. Thiết kế cơ sở dữ liệu : 5](#_Toc76678450)

[i. Mô hình vật lý : 5](#_Toc76678451)

[ii. Mô hình dữ liệu quan hệ: 6](#_Toc76678452)

[c. Mô tả thành phần dữ liệu: 7](#_Toc76678453)

[CHƯƠNG 3 : CÁC CÔNG NGHỆ SỬ DỤNG 28](#_Toc76678454)

[1. Oracle Database 28](#_Toc76678455)

[2. Sql Developer: 28](#_Toc76678456)

[3. Eclipse: 28](#_Toc76678457)

[4. JDBC 29](#_Toc76678458)

[5. Draw.io. 29](#_Toc76678459)

[6. Google Drive: 29](#_Toc76678460)

[CHƯƠNG 4 : XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC 30](#_Toc76678461)

[I. Trigger : 30](#_Toc76678462)

[1. Định nghĩa: 30](#_Toc76678463)

[2. Danh sách các Trigger : 32](#_Toc76678464)

[II. Store Procedure : 43](#_Toc76678465)

[**1.** **Store Procedure trong Oracle :** 43](#_Toc76678466)

[1.1. Định nghĩa: 43](#_Toc76678467)

[1.2 . Danh sách Store Procedure: 44](#_Toc76678468)

[**2.** **Function trong Oracle :** 71](#_Toc76678469)

[2.1. Định nghĩa : 71](#_Toc76678470)

[2.2 Danh sách các Function : 72](#_Toc76678471)

[**3.** **Transaction trong Oracle:** 73](#_Toc76678472)

[3.1. Định nghĩa : 73](#_Toc76678473)

[3.2. Cấu trúc transaction: 73](#_Toc76678474)

[3.3. Transaction control: 74](#_Toc76678475)

[CHƯƠNG 5 : XỬ LÝ TRUY XUẤT ĐỒNG THỜI 82](#_Toc76678476)

[I. Các mức cô lập trong Oracle : 82](#_Toc76678477)

[**1.** **Read Committed:** 82](#_Toc76678478)

[**2.** **Serializable:** 82](#_Toc76678479)

[**3.** **Read-only:** 83](#_Toc76678480)

[II. Cơ chế khóa: 83](#_Toc76678481)

[III. Deadlock: 84](#_Toc76678482)

[IV. Mô tả đồ án môn học: 92](#_Toc76678483)

[**1.** **Lost Update:** 92](#_Toc76678484)

[2. **Non – repeatable Read:** 106](#_Toc76678485)

[3. **Phantom Read:** 114](#_Toc76678486)

[CHƯƠNG 6: THIẾT KẾ GIAO DIỆN 123](#_Toc76678487)

[I. Danh sách các màn hình: 123](#_Toc76678488)

[II. Mô tả các màn hình: 125](#_Toc76678489)

[1. Đăng nhập : 125](#_Toc76678490)

[2. Màn hình chính của quản lý: 126](#_Toc76678491)

[3. Màn hình quản lý quy định: 126](#_Toc76678492)

[4. Màn hình thêm quy định 127](#_Toc76678493)

[5. Màn hình thống kê nhân viên 127](#_Toc76678494)

[6. Màn hình tra cứu nhân viên 128](#_Toc76678495)

[7. Màn hình thêm nhân viên 128](#_Toc76678496)

[8. Màn hình sửa nhân viên 129](#_Toc76678497)

[9. Màn hình quản lý dịch vụ 130](#_Toc76678498)

[10. Màn hình sửa dịch vụ 130](#_Toc76678499)

[11. Màn hình thêm dịch vụ 131](#_Toc76678500)

[12. Màn hình chi tiết phiếu dịch vụ 131](#_Toc76678501)

[13. Màn hình thêm dịch vụ vào phiếu dịch vụ 132](#_Toc76678502)

[14. Màn hình chính của nhân viên 132](#_Toc76678503)

[15. Màn hình quản lý khách hàng 133](#_Toc76678504)

[16. Màn hình thêm khách hàng 133](#_Toc76678505)

[17. Màn hình sửa khách hàng 134](#_Toc76678506)

[18. Màn hình quản lý hóa đơn 135](#_Toc76678507)

[19. Màn hình thêm hóa đơn 135](#_Toc76678508)

[20. Màn hình quản lý phiếu thuê phòng 136](#_Toc76678509)

[21. Màn hình thêm phiếu thuê phòng 136](#_Toc76678510)

[22. Màn hình lập phiếu thuê phòng 137](#_Toc76678511)

[23. Màn hình chi tiết phiếu thuê phòng 137](#_Toc76678512)

[24. Màn hình quản lý phiếu đặt phòng 138](#_Toc76678513)

[25. Màn hình thêm phiếu đặt phòng 139](#_Toc76678514)

[26. Màn hình chi tiết phiếu đặt phòng 140](#_Toc76678515)

[27. Màn hình thống kê theo tháng 141](#_Toc76678516)

[28. Màn hình xuất phiếu dịch vụ 142](#_Toc76678517)

[29. Màn hình xuất hóa đơn 143](#_Toc76678518)

[CHƯƠNG 7: KẾT LUẬN 145](#_Toc76678519)

[1. Kết quả đạt được: 145](#_Toc76678520)

[2. Hạn chế: 145](#_Toc76678521)

[3. Hướng phát triển: 145](#_Toc76678522)

[PHỤ LỤC 1: BẢNG PHÂN CHIA CÔNG VIỆC 146](#_Toc76678523)

[PHỤ LỤC 2 : TÀI LIỆU THAM KHẢO 147](#_Toc76678524)

**LỜI CẢM ƠN**

Lời đầu tiên nhóm Lucifer chúng em xin gửi lời cảm ơn chân thành tới cô Đỗ Thị Minh Phụng – giảng viên lý thuyết và thực hành môn Hệ quản trị cơ sở dữ liệu đã tận tình giúp đỡ, chỉ bảo, cung cấp các kiến thức căn cơ nhất trong bộ môn này, đồng thời giúp chúng em có những định hướng rõ ràng trong suốt quá trình thực hiện đề tài. Cô cũng đã tạo điều kiện thuận lợi, giải đáp các thắc mắc cho chúng em ngay tại lớp cũng như thông qua các kênh thông tin khác nhau.

Mặc dù chúng em luôn cố gắng chu toàn trong mọi công đoạn thực hiện đồ án nhưng vẫn không thể tránh khỏi những sai sót, khuyết điểm, những vấn đề tồn động chưa thể giải quyết. Vậy nên, chúng em rất mong nhận được nhận xét từ Cô, lời nhận xét của cô chính là động lực phát triển giúp chúng em có thể hoàn thiện bản thân trong tương lại.

Lời cuối, nhóm Lucifer xin kính chức cô thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp là truyền đạt kiến thức cho các bạn sinh viên.

Xin chân thành cảm ơn cô !

**Lucifer Team**

# 

# **CHƯƠNG 1 : TỔNG QUAN ĐỀ TÀI**

## Đặt vấn đề:

Ngày nay, thế giới không ngừng phát triển, đòi hỏi mọi người phải có những bước tiến bộ để theo kịp luồng quay của xã hội, sự đổi mới về mọi mặt của đời sống từ văn hóa đến giáo dục, từ kinh tế đến chính trị, … Điều gì đã tác động sự phát triển không ngừng của lực lượng sản xuất, đã thúc đẩy sự tăng trưởng về trí não cũng như tinh thần và thể lực của con người? Phải chăng, sự chuyển biến mạnh mẽ của ngành công nghệ thông tin là chìa khóa của mọi thay đổi, là đòn bẩy cho sự tiến bộ của cơ cấu nền kinh tế - xã hội trong và ngoài nước? Nhận thức sâu sắc về sự thống trị của công nghệ và sự bùng nổ của thông tin, đòi hỏi mọi người cần áp dụng tin học hóa vào tất cả các ngành từ công nghiệp, giải trí đến dịch vụ, ... Bên cạnh đó, phần mềm cũng có những bước chuyển đổi đáng kể, ngày càng đa dạng, phong phú và hỗ trợ hiệu quả cho người dùng.

Song song đó, việc phát triển phần mềm hiện nay không chỉ đòi hỏi khả năng chính xác cao, xử lý đồng thời nhiều nghiệp vụ mà còn giải quyết các vấn đề khác như mô phỏng và tự động được rất nhiều nghiệp vụ khó khăn, cải thiện tốc độ truy xuất, tăng khả năng lưu trữ, bảo mật dữ liệu cao, … Các phần mềm giúp đơn giản hóa các hình thức kinh doanh (nhất là việc sửa lỗi và tự động hóa), tiết kiệm thời gian và nhân lực, thực hiện chiến dịch marketing hiệu quả, … Điều gì sẽ xảy ra, khi một quy trình quản lý lại sử dụng những phương thức truyền thống để vận hành? Một cách đơn giản, hãy thử tưởng tượng một hệ thống quản lý khách sạn, nơi mà quản trị một lượng lớn và liên tục về dữ liệu liên quan đến khách hàng, các loại dịch vụ và các bản báo cáo thống kê về doanh thu. Việc quản lý này cần khá nhiều nhân lực, chia thành nhiều khâu xử lý mới có thể quản lý tốt các bộ hồ sơ, khách hàng, … nhưng rất dễ bị thất lạc và tốn kém mà độ chính xác và hiệu quả lại không cao dẫn đến việc các quy trình trở nên phức tạp, trì trệ trong khi, các nghiệp vụ này hoàn toàn có thể tin học hóa dễ dàng. Với việc áp dụng công nghệ, mọi khó khăn gần như được giải quyết, quy trình quản lý khách sạn sẽ trở nên đơn giản, thuận tiện và hiệu quả hơn rất nhiều.

Phần mềm quản lý khách sạn đang không ngừng phủ sóng trong thị trường quốc tế nhưng lại chưa được phổ biến ở các nước phát triển nói chung và ở Việt Nam nói riêng. Nhận thấy được những lợi ích mà phần mềm quản lý khách sạn có thể mang lại, khách sạn Lucifer’s Hotel đã thuê một đội kỹ sư để thiết kế một hệ thống quản lý vừa có thể đáp ứng các yêu cầu đa dạng của khách sạn, vừa cung cấp tất cả các công cụ mà nhân viên khách sạn cần để thực hiện các công việc như xử lý đặt phòng, kiểm tra phòng, ... nhằm hỗ trợ cho quá trình quản lý cũng như nâng cao chất lượng dịch vụ.

Bên cạnh việc xử lý các công việc hàng ngày, hệ thống còn giải quyết các vấn đề khi giao tác và xử lý truy xuất đồng thời giữa các nhân viên trong quá trình tra cứu, đặt phòng hay khả năng tích hợp quá trình tương tác giữa nhân viên và quản lý, giữa khách hàng và lễ tân.

1. **Mục tiêu:**

* Thiết kế và xây dựng hệ thống quản lý khách sạn hoàn thiện và đáng tin cậy, giúp giám đốc quản lý khách hàng, nhân viên và các dịch vụ khác thuận tiện hơn và cải thiện quá trình trao đổi giữa nhân viên và giám đốc, giữa khách hàng và tiếp tân. Phần mềm đem lại những trải nghiệm tốt nhất và đơn giản nhất khi người dùng tương tác với hệ thống. Với một bộ phận người dùng nhất định, hệ thống sẽ cung cấp các quyền và chức năng riêng theo yêu cầu của chủ hệ thống như nhân viên lễ tân có chức năng tra cứu thông tin nhân viên, thêm và tra cứu thông tin khách hàng, … Giám đốc có chức năng quản lý nhân viên, quản lý phòng, quản lý tài chính, …
* Hệ thống sẽ đáp ứng các yêu cầu sau:
* Quản lý thông tin nhân viên, thông tin khách hàng, thông tin dịch vụ, thông tin phòng, kế hoạch tài chính và một số rủi ro.
* Quản lý các hoạt động của nhân viên: đăng nhập vào ứng dụng; tra cứu thông tin của nhân viên, khách hàng, dịch vụ hay phòng thuê; tạo và tra cứu các thông tin hóa đơn, thêm khách hàng và quản lý các chương trình khuyến mãi.
* Khi khách hàng muốn đặt phòng trước, hệ thống sẽ hiển thị số phòng, tình trạng phòng, giá phòng và các đặc điểm khác để thuận tiện trong quá trình trao đổi giữa khách hàng và nhân viên.
* Quản lý các hoạt động báo cáo thống kê và báo cáo doanh thu theo ngày, tháng và năm.
* Hỗ trợ lưu trữ các thông tin nhanh chóng và bảo mật.
* Hỗ trợ phân quyền người dùng, tăng tính an ninh thông tin trong hệ thống.

# **CHƯƠNG 2 : PHÂN TÍCH VÀ YÊU CẦU DỮ LIỆU**

## Đặc tả bài toán:

Lucifer's Hotel đang tiến hành xây dựng ứng dụng phần mềm phục vụ việc quản lý cho khách sạn của mình với những chức năng sau:

Hoạt động của nhân viên: khi du khách đến khách sạn để thuê phòng, nhân viên lễ tân phải kiểm tra khách hàng có trong danh sách phiếu đặt phòng đã lập trước hay không? Trường hợp khách hàng không đặt trước, nhân viên kiểm tra còn phòng phù hợp với yêu cầu của khách hay không? Nếu có phòng đáp ứng với nhu cầu khách hàng hoặc khi khách hàng có trong danh sách đặt phòng thì tiến hành lập phiếu thuê phòng. Ngoài ra, nhân viên còn có chức năng tra cứu thông tin dịch vụ khi khách hàng hỏi thăm và lập phiếu dịch vụ khi khách hàng lựa chọn và sử dụng dịch vụ tại đây. Khi khách hàng trả phòng, nhân viên sẽ tạo hóa đơn và tiếp nhận tiền thanh toán của khách hàng trước khi in hóa đơn và ký xác nhận giữa khách hàng và nhân viên để hoàn thành thủ tục.

Hoạt động của giám đốc: giám đốc là người điều hành và quản lý khách sạn nên giữ nhiều chức năng nhất trong hệ thống. Bên cạnh các chức năng của nhân viên, người quản lý còn có chức năng thêm, xóa, cập nhất các thông tin về phòng, dịch vụ, nhân sự. Trong quản lý tài chính, giám đốc sẽ theo dõi công nợ, đóng thuế doanh nghiệp, … Hệ thống sẽ hỗ trợ thống kê số lượng khách hàng, số lượng phòng theo từng ngày và hỗ trợ báo cáo doanh thu theo ngày, tuần, tháng hoặc năm.

Nhân viên lễ tân và giám đốc đều phải đăng nhập thành công trước khi sử dụng các chức năng của hệ thống ứng với tên tài khoản và mật khẩu đã được cung cấp trước đó.

## Phân tích yêu cầu:

### **Yêu cầu phi chức năng:**

* Yêu cầu về giao diện:
* Giao diện thân thiện, phù hợp với nhiều người dùng.
* Bố cục rõ ràng, màu sắc hài hòa.
* Thao tác nhanh gọn và dễ sử dụng.
* Có tính tương tác cao.
* Yêu cầu về chất lượng:
* *Tính tiến hóa*:
* Dễ dàng trong việc cập nhật hoặc nâng cấp cơ sở dữ liệu
* Dễ dàng trong lúc nâng cấp các modules khác trong tương lai.
* *Tính tiện dụng*:
* Hệ thống có giao diện trực quan, sinh động và dễ sử dụng.
* Các đặc tả và hướng dẫn sử dụng rõ ràng.
* Thao tác đơn giản và nhanh gọn, đáp ứng nhu cầu của nhân viên khách sạn.
* *Tính hiệu quả*:
* Hệ thống hỗ trợ xử lý sự cố trong vòng 24h sau khi được liên hệ.
* Hệ thống có tính bảo mật dữ liệu cao, an ninh tốt.
* Hệ thống hoạt động ổn định và đáng tin cậy với lượng dữ liệu lớn.
* Ứng dụng thể hiện tốt thông tin nhân viên, khách hàng, dịch vụ, hoá đơn, thống kê.
* *Tính tương thích*:
* Hoạt động tốt trên nhiều nền tảng và ứng dụng, mobile, tablet.
* Ứng dụng phần mềm hoạt động ổn định, đáng tin cây, có thể truy cập và xử lý nhiều tác vụ cùng lúc và ghi lại quá trình thao tác trên cơ sở dữ liệu.

### **Thiết kế mô hình quan hệ:**

1. **Mô hình thực thể mối kết hợp:**

**Diagram

Description automatically generated**

*Hình 1: Mô hình ERD.*

1. **Thiết kế cơ sở dữ liệu:**

##### Diagram Description automatically generatedi. Mô hình vật lý:

*Hình 2: Mô hình vật lý.*

1. **Mô hình dữ liệu quan hệ:**

NhanVien (**maNV**, tenNV, ngaySinhNV, gioiTinhNV, chucVu, cccdNV, ngayVL, tenND, password, emailNV, maQL)

KhachHang (**maKH**, tenKH, cccdKH, sdtKH, gioiTinhKH, ngaySinhKH, quocTich, doanhSo, maLKH)

LoaiKhachHang (**maLKH**, tenLKH)

KhuyenMai (**maKM**, tenKM, tuNgay, denNgay)

ChiTietKM(**maKM, maLKH**, phanTramKM)

Phong (**maPhong**,tenPhong, ghiChu, maLP)

LoaiPhong (**maLP**, tenLP, giaPhong, slToiDa)

TinhTrangPhong (**maTTP**, tenTTP)

ChiTietTTP (**maPhong, maTTP, ngayBD, ngayKT**)

DichVu (**maDV**, tenDV, giaDV, maDonVi)

DonVi (**maDonVi**, tenDonVi)

TinhTrangPhieu (**maTT,** tenTT)

PhieuDichVu (**maPDV**, tienDV, maKH, maNV, maTT)

ChiTietPDV (**maPDV, maDV**, **ngaySD,** slDV)

PhieuDatTruoc (**maPDT**, ngayDT, maKH, maNV , maTT)

ChiTietPDT (**maPDT, maPhong**, ngayNP, ngayTPDD)

PhieuThuePhong (**maPTP**, tienPhong, maKH, maNV, maPDT , maTT)

ChiTietPTP (**maPTP, maPhong**, ngayNP, ngayTPTT, slNguoi, phuThu)

HoaDon (**maHD**, tongTien, ngayTT, maNV, maKH, maPTP, maPDV)

QuyDinh **(maQD**, tenQD, moTa)

1. **Mô tả thành phần dữ liệu:**

Thực thể : Người dùng

1. Thực thể : Nhân viên

* Tên bảng quan hệ: NhanVien
* Mô tả thực thể: là một thực thể đại diện cho nhân viên đang làm việc tại khách sạn Lucifer’s Hotel.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maNV | Mã nhân viên. | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | tenNV | Tên nhân viên. | Kiểu ký tự: varchar2 | 50 |  |  |
| 3 | cccdNV | Căn cước công nhân của nhân viên. | Kiểu ký tự: varchar2 | 12 |  | Các trường giá trị phải là duy nhất. |
| 4 | ngayVL | Ngày vào làm nhân viên. | Kiểu ngày: date |  |  |  |
| 5 | gioiTinhNV | Giới tính nhân viên. | Kiểu ký tự: varchar2 | 3 | 2 | Giới tính chỉ có một trong hai giá trị là nam hoặc nữ. |
| 6 | ngaySinhNV | Ngày sinh nhân viên. | Kiểu ngày: date |  |  | Nhân viên phải từ đủ 18 tuổi trở lên mới được làm việc tại khách sạn. |
| 7 | chucVu | Chức vụ hiện tại của nhân viên. | Kiểu ký tự: varchar2 | 50 | 2 | Có hai loại chức vụ chính trong hệ thống là nhân viên lễ tân và quản lý. |
| 8 | tenND | Tên người dùng. | Kiểu ký tự: varchar2 | 25 | 100 | Do người dùng đặt. |
| 9 | Password | Mật khẩu người .dùng | Kiểu ký tự: varchar2 | 25 | 100 |  |
| 10 | emailNV | Email người dùng. | Kiểu ký tự: varchar2 | 30 | 100 | Mỗi người dùng khi truy cập hệ thống sẽ có một email riêng để có thể thay đổi mật khẩu trong trường hợp quên mật khẩu hoặc các trường hợp khác. |
| 11 | maQL | Mã quản lý. | Kiểu số: int |  | 1 | Mã quản lý của khách sạn là duy nhất. |

1. Thực thể : Khách hàng

* Tên bảng quan hệ: KhachHang
* Mô tả thực thể: là một thực thể đại diện cho khách hàng đến thuê phòng hoặc đặt phòng tại khách sạn.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maKH | Mã khách hàng. | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | tenKH | Tên khách hàng. | Kiểu ký tự: varchar2 | 50 |  |  |
| 3 | cccdKH | Căn cước công dân của khách hàng. | Kiểu ký tự: varchar2 | 12 |  | Các trường giá trị phải là duy nhất. |
| 4 | sdtKH | Số điện thoại khách hàng. | Kiểu ký tự: varchar2 | 10 |  |  |
| 5 | gioiTinhKH | Giới tính khách hàng. | Kiểu ký tự: varchar2 | 3 | 2 | Giới tính chỉ có một trong hai giá trị là nam hoặc nữ. |
| 6 | ngaySinhKH | Ngày sinh khách hàng. | Kiểu ngày: date |  |  | Những khách hàng từ đủ 18 tuổi trở lên mới có thể đặt phòng và thuê phòng tại khách sạn. |
| 7 | quocTich | Quốc tịch khách hàng. | Kiểu ký tự: varchar2 | 25 |  |  |
| 8 | doanhSo | Doanh số khách hàng | Kiểu số:  number | 15 |  | Tổng số tiền khách hàng đã chi cho khách sạn. |
| 9 | maLKH | Mã loại khách hàng. | Kiểu số: int |  | 3 | Đây là khóa ngoại của thực thể. |

1. Thực thể : Loại khách hàng.

* Tên bảng quan hệ: LoaiKhachHang
* Mô tả thực thể: là một thực thể đại diện cho các loại khách hàng tồn tại trong khách sạn.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maLKH | Mã lọai khách hàng. | Kiểu số: int |  | 3 | Đây là khóa chính của thực thể. |
| 2 | tenLKH | Tên loại khách hàng. | Kiểu ký tự: varchar2 | 25 |  |  |

1. Thực thể : Khuyến mãi.

* Tên bảng quan hệ: KhuyenMai
* Mô tả thực thể: là một thực thể đại diện các chương trình khuyến mãi được áp dụng với loại khách hàng tương ứng.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maKM | Mã khuyến mãi. | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | tenKM | Tên khuyến mãi. | Kiểu ký tự: varchar2 | 50 |  |  |
| 4 | tuNgay | Từ ngày – ngày bắt đầu khuyến mãi. | Kiểu ngày: date |  |  |  |
| 5 | denNgay | Đến ngày – ngày kết thúc khuyến mãi. | Kiểu ngày: date |  |  |  |

1. Thực thể : Chi tiết khuyến mãi

- Tên bảng quan hệ: ChiTietKM

- Mô tả thực thể: là một thực thể đại diện từng ưu đãi riêng đối với từng loại khách hàng trong khách sạn.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maKM | Mã khuyến mãi. | Kiểu số: int |  |  | Đây là thành phần  khóa chính của thực thể. |
| 2 | maLKH | Mã loại khách hàng. | Kiểu số: int |  | 3 | Đây là thành phần khóa chính của thực thể. |
| 3 | phanTramKM | Phần trăm khuyến mãi. | Kiểu số: number | 3 | [0, 1] | Phần trăm khuyến mãi chỉ có thể là số thực với tối đa hai chữ số sau phần thập phân. |

1. Thực thể: Phòng

- Tên bảng quan hệ: Phong

- Mô tả thực thể: là một thực thể đại diện các phòng trong khách sạn Lucifer’s Hotel.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maPhong | Mã phòng. | Kiểu số: int | 3 | 100 | Đây là khóa chính của thực thể. |
| 2 | tenPhong | Tên phòng | Kiêu ký tự: varchar2 | 25 |  |  |
| 3 | ghiChu | Ghi chú. | Kiểu ký tự: varchar2 | 100 |  | Ghi chú sẽ chứa các thông tin về các chi tiết và thiết bị trong phòng như phòng có một giường và một bồn tắm, một vòi sen, ... |
| 4 | maLP | Mã loại phòng. | Kiểu số: int |  | 5 | Đây là khóa ngoại của thực thể. |

1. Thực thể loại phòng

* Tên bảng quan hệ: LoaiPhong
* Mô tả thực thể: là một thực thể đại diện loại phòng tồn tại trong khách sạn sẽ phục vụ khách hàng như phòng standard, phòng deluxe, ...

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maLP | Mã loại phòng. | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | tenLP | Tên loại phòng. | Kiểu ký tự: varchar2 | 25 |  |  |
| 3 | giaPhong | Giá phòng. | Kiểu số: number | 15 |  |  |
| 4 | slToiD | Số lượng người tối đa trong phòng. | Kiểu: int |  | [1, 4] |  |

1. Thực thể : Tình trạng phòng

* Tên bảng quan hệ: TinhTrangPhong
* Mô tả thực thể: Đây là thực thể đại diện cho hai thực thể là tình trạng phòng và phòng. Trong đó, thực thể này sẽ liệt kê các tình trạng phòng trong khoảng thời gian cụ thể.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maTTP | Mã tình trạng phòng. | Kiểu số: int |  | 3 | Đây là khóa chính của thực thể. |
| 2 | tenTTP | Tên tình trạng phòng. | Kiểu ký tự: varchar2 | 25 | 3 |  |

1. Thực thể : Chi tiết tình trạng phòng

* Tên bảng quan hệ: ChiTietTTP
* Mô tả thực thể: là một thực thể đại diện tình trạng phòng cụ thể của phòng tại mỗi thời điểm khác nhau.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maTTP | Mã tình trạng phòng. | Kiểu số: int |  | 3 | Đây là thành phần khóa chính của thực thể. |
| 2 | maPhong | Mã phòng. | Kiểu số: int | 3 | 100 | Đây là thành phần khóa chính của thực thể. |
| 3 | ngayBD | Ngày bắt đầu của tình trạng đó. | Kiểu ngày: date. |  |  | Đây là thành phần khóa chính của thực thể. |
| 4 | ngayKT | Ngày kết thúc của tình trạng đó. | Kiểu ngày: date. |  |  | Đây là thành phần khóa chính của thực thể. |

1. Thực thể: Dịch vụ

* Tên bảng quan hệ: DichVu
* Mô tả thực thể: là một thực thể đại diện các hình thức dịch vụ bên trong và bên ngoài khách sạn.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maDV | Mã dịch vụ. | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | tenDV | Tên dịch vụ. | Kiểu ký tự: varchar2 | 50 |  |  |
| 3 | maDonVi | Mã đơn vị. | Kiểu số: int |  | 2 | Đây là khóa ngoại của thực thể. |
| 4 | giaDV | Giá dịch vụ. | Kiểu số: number | 15 |  |  |

1. Thực thể: Đơn vị

* Tên bảng quan hệ: DonVi
* Mô tả thực thể: là một thực thể đại diện các đối tác khi lập danh sách các dịch vụ đã liên kết với Lucifer’s Hotel để phục vụ khách hàng trong khách sạn.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maDonVi | Mã đơn vị. | Kiểu số: int |  | 2 | Đây là khóa chính của thực thể. |
| 2 | tenDonVi | Tên đơn vị. | Kiểu ký tự: varchar2 | 50 |  |  |

1. Thực thể: Tình trạng phiếu

* Tên bảng quan hệ: TinhTrangPhieu
* Mô tả thực thể: là một thực thể thể hiện tình trạng sử dụng của các phiếu như phiếu thuê phòng, phiếu dịch vụ như đã được đã được thanh toán hay chưa được thanh toán.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maTT | Mã tình trạng phiếu | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | tenTT | Tên tình trạng phiếu | Kiểu ký tự: var char2 | 25 |  |  |

1. Thực thể: Phiếu dịch vụ

* Tên bảng quan hệ: PhieuDichVu
* Mô tả thực thể: là một thực thể đại diện cho phiếu dịch vụ khi khách hàng sử dụng dịch vụ trong khách sạn. (Có thể thuê phòng tại khách sạn hoặc không)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maPDV | Mã phiếu dịch vụ. | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | tongTienDV | Tổng tiền dịch vụ. | Kiểu số: number | 15 |  |  |
| 3 | maKH | Mã khách hàng. | Kiểu số: int |  |  | Đây là khóa ngoại của thực thể. |
| 4 | maNV | Mã nhân viên. | Kiểu số: int |  | 100 | Đây là khóa ngoại của thực thể. |

1. Thực thể : Chi tiết phiếu dịch vụ

* Tên bảng quan hệ: ChiTietPDV
* Mô tả thực thể: là thực thể đại diện cho hai thực thể là phiếu dịch vụ và dịch vụ. Trong đó, thực thể này sẽ liệt kê các dịch vụ sử dụng trong một khoảng thời gian nhất định.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maPDV | Mã phiếu dịch vụ. | Kiểu số: int |  |  | Đây là thành phần khóa chính của thực thể. |
| 2 | maDV | Mã dịch vụ. | Kiểu số: int |  |  | Đây là thành phần khóa chính của thực thể. |
| 4 | ngaySD | Ngày sử dụng dịch vụ. | Kiểu ngày: date |  |  | Đây là thành phần khóa chính của thực thể. |
| 4 | slDV | Số lượng dịch vụ. | Kiểu số: int |  | 100 |  |

1. Thực thể: Phiếu đặt trước

* Tên bảng quan hệ: PhieuDatTruoc.
* Mô tả thực thể: là một thực thể được thiết lập khi khách hàng đặt phòng trước khi đến khách sạn.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maPDT | Mã phiếu đặt trước. | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | ngayDT | Ngày đặt trước. | Kiểu ngày: date |  |  |  |
| 3 | maKH | Mã khách hàng. | Kiểu số: int |  |  | Đây là khóa ngoại của thực thể. |
| 4 | maNV | Mã nhân viên. | Kiểu số: int |  | 100 | Đây là khóa ngoại của thực thể. |
| 5 | maTT | Mã tình trạng | Kiểu số : int |  | 1 hoặc 2 | Đây là khóa ngoại của thực thể |

1. Thực thể: Chi tiết phiếu đặt trước

* Tên bảng quan hệ: ChiTietPDT
* Mô tả thực thể: là thực thể đại diện cho hai thực thể là phiếu đặt trước và phòng. Trong đó, thực thể này sẽ liệt kê các phòng sẽ có thể được sử dụng trong một khoảng thời gian nhất định.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maPDT | Mã phiếu đặt trước. | Kiểu số: int |  |  | Đây là thành phần khóa chính của thực thể. |
| 2 | maPhong | Mã phòng. | Kiểu số: int |  | 100 | Đây là thành phần khóa chính của thực thể. |
| 3 | ngayNP | Ngày nhận phòng. | Kiểu ngày: date |  |  |  |
| 4 | ngayTPDD | Ngày trả phòng dự kiến | Kiểu ngày: date |  |  | Thế hiện ngày dự kiến khách hàng đặt phòng sẽ trả phòng, khách hàng có thể thực hiện trả phòng trước, trong, hoặc sau ngày này. |

1. Thực thể : Phiếu thuê phòng

* Tên bảng quan hệ: PhieuThuePhong
* Mô tả thực thể: là thực thể được thiết lập khi khách hàng thuê phòng tại khách sạn.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maPTP | Mã phiếu thuê phòng. | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | tongTienPhong | Tổng tiền phòng. | Kiểu số: number | 15 |  |  |
| 3 | maKH | Mã khách hàng. | Kiểu số: int |  |  | Đây là khóa ngoại của thực thể. |
| 4 | maNV | Mã nhân viên. | Kiểu số: int |  | 100 | Đây là khóa ngoại của thực thể. |
| 5 | maPDT | Mã phiếu đặt trước. | Kiểu số: int |  |  | Đây là khóa ngoại của thực thể. |
| 6 | maTT | Mã tình trạng | Kiểu số : int |  | [1,2] | Đây là khóa ngoại của thực thể |

1. Thực thể : Chi tiết phiếu thuê phòng

* Tên bảng quan hệ: ChiTietPTP
* Mô tả thực thể: là thực thể đại diện cho hai thực thể là phiếu đặt phòng và phòng. Trong đó, thực thể này sẽ liệt kê các phòng được sử dụng trong một khoảng thời gian nhất định.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maPTP | Mã phiếu thuê phòng. | Kiểu số: int |  |  | Đây là thành phần  khóa chính của thực thể. |
| 2 | maPhong | Mã phòng. | Kiểu số: int |  |  | Đây là thành phần khóa chính của thực thể. |
| 3 | ngayNP | Ngày nhận phòng. | Kiểu ngày: date |  |  |  |
| 4 | ngayTPTT | Ngày trả phòng thực tế. | Kiểu ngày: date |  |  | Thể hiện ngày thực tế mà khách hàng đặt phòng thực hiện trả phòng. |
| 5 | slNguoi | Số lượng khách trong phòng. | Kiểu số: int. |  | [1, 5] | Các trường giá trị này là thành phần quyết định giá trị của thuộc tính phuThu |
| 7 | phuThu | Phụ thu. | Kiểu số: number | 15 | [0, 1] | Phụ thu sẽ dành trong trường hợp kê thêm giường đối với phòng 2 người hoặc 4 người. Riêng đối với phòng có khách là người nước ngoài cũng phụ thu thêm. Trường phụ thu chỉ có thể là số thực với tối đa hai chữ số sau phần thập phân. |

1. Thực thể: Hóa đơn

* Tên bảng quan hệ: HoaDon
* Mô tả thực thể: là thực thể đại diện cho hóa đơn khi khách hàng tiến hành thanh toán về phiếu dịch vụ hay phiếu thuê phòng hay cả hai.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maHD | Mã hóa đơn. | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | tongTien | Tổng tiền. | Kiểu số: number | 15 |  |  |
| 3 | ngayTT | Ngày thanh toán. | Kiểu ngày: date |  |  |  |
| 4 | maKH | Mã khách hàng. | Kiểu số: int |  |  | Đây là khóa ngoại của thực thể. |
| 5 | maNV | Mã nhân viên. | Kiểu số: int |  | 100 | Đây là khóa ngoại của thực thể. |
| 6 | maPTP | Mã phiếu thuê phòng. | Kiểu số: int |  |  | Đây là khóa ngoại của thực thể. |
| 7 | maPDV | Mã phiếu dịch vụ. | Kiểu số: int |  |  | Đây là khóa ngoại của thực thể. |

1. Thực thể : Quy định

* Tên bảng quan hệ: QuyDinh
* Mô tả thực thể: là thực thể đại diện cho các quy định được áp dụng vận hành trong quá trình hoạt dộng của khách sạn, được thể hiện ra để người dùng có thể tra cứu để áp dụng thực hiện hoặc thay đổi các quy định.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Chiều dài** | **Miền giá trị** | **Ghi chú** |
| 1 | maQD | Mã quy định. | Kiểu số: int |  |  | Đây là khóa chính của thực thể. |
| 2 | tenQD | Tên quy định. | Kiểu ký tự: varchar2 | 25 |  |  |
| 3 | moTa | Mô tả. | Kiểu ký tự: varchar2 | 500 |  | Sẽ mô tả các đặc điểm hay chi tiết bổ sung vào cho người dùng biết. |

1. **Mô tả ràng buộc toàn vẹn**

**1**. Thực thể: Nhân viên

1.1 . R1: “Ngày vào làm của nhân viên phải lớn hơn ngày sinh.”

RBTV liên bộ.

Bối cảnh : NHANVIEN.

∀ nv ∈ NHANVIEN : nv.ngayvaolam > nv.ngaysinh

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Xóa | Sửa |
| NHANVIEN | + | - | +(ngayvaolam,  ngaySinhNV) |

1.2. R2: “Giới tính chỉ có thể là Nam hoặc Nữ.”

RBTV miền giá trị.

Bối cảnh : NHANVIEN

∀ nv ∈ NHANVIEN: nv.gioiTinhNV = ‘Nam’ ∨ nv.gioiTinhNV = ‘Nữ’

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | Thêm | Xóa | Sửa |
| NHANVIEN | + | - | +( gioiTinhNV) |

1.3. R3: “Số chứng minh nhân dân của nhân viên là duy nhất”.

RBTV miền liên bộ.

Bối cảnh : NHANVIEN

∀ nv1, nv2 ∈ NHANVIEN: nv1 ≠ nv2 → nv1. cccdNV ≠ nv2. cccdNV;

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

2. Thực thể: khách hàng

2.1 R1: “Số chứng minh nhân dân của khách hàng là duy nhất”.

RBTV miền liên bộ.

Bối cảnh: KHACHHANG

∀ kh1, kh2 ∈ KHACHHANG: kh1 ≠ kh2 → kh1. cccdKH ≠ kh2. cccdKH

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Xóa | Sửa |
| KHACHHANG | + | - | +(cccdKH) |

2.2. R2: “Khách hàng ít nhất từ 18 tuổi trở lên”.

RBTV miền giá trị.

Bối cảnh: KHACHHANG

∀ kh ∈ KHACHHANG: year(sysdate()) - year(kh.ngaySinhKH) >= 18

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | Thêm | Xóa | Sửa |
| KHACHHANG | + | - | +(ngaySinhKH) |

2.3. R2: “Giới tính chỉ có thể là Nam hoặc Nữ.”

RBTV miền giá trị.

Bối cảnh : KHACHHANG

∀ kh ∈ KHACHHANG: kh.gioiTinhKH = ‘Nam’ kh.gioiTinhKH = ‘Nu’

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | Thêm | Xóa | Sửa |
| KHACHHANG | + | - | +( gioiTinhKH) |

3. Thực thể: Phòng

3.1. R1: “Tình trạng phòng có ba trạng thái: phòng trống (1), đang thuê (3)  hoặc đặt trước (2)”.

RBTV miền liên thuộc tính

Bối cảnh:, CHITIETTINHTRANGPHONG,

∀ p ∈ PHONG, ∀ ctttp ∈ CHITIETTINHTRANGPHONG: p.maPhong=ctttp.maPhong∧ (ctttp.maTTP = ctttp.maTTP = 2 ctttp.maTTP = 3)

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Xóa | Sửa |
| PHONG | + | - | - |
| CHITIETTINHTRANGPHONG | + | - | +(maTTP) |

**4**. Thực thể: phiếu đặt trước

4.1. R1:“Ngày đặt phòng phải nhỏ hơn hoặc bằng ngày nhận phòng”

RBTV miền liên thuộc tính.

Bối cảnh: PHIEUDATRUOC, CHITIEPHIEUDATTRUOC

∀ dp ∈ PHIEUDATTRUOC,

ctdp ∈ CHITIEPHIEUDATTRUOC,

ctdp.maPDT= dp.maPDT → dp.ngayDT < = ctdp.ngayNP

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Xóa | Sửa |
| PHIEUDATPHONG | + | - | +(ngayDT) |
| CHITIEPHIEUDATTRUOC | + | - | +(ngayNP) |

4.2. R2 :”Tình trạng phiếu đặt trước có hai trạng thái: Phiếu chưa được sử dụng trong phiếu thuê phòng (1), phiếu đã được sử dụng trong phiếu thuê phòng(2).”

RBTV miền liên thuộc tính

Bối cảnh: PHIEUDATRUOC, PHIEUTHUEPHONG , TINHTRANGPHIEU

∀ dp ∈ PHIEUDATTRUOC, tp ∈ PHIEUTHUEPHONG,

ttp ∈ TINHTRANGPHIEU: dp.maPDT=tp.PDT ttp.maTT = 1 ttp.maTT= 2

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Xóa | Sửa |
| PHIEUDATRUOC | + | - | +(maPDT, maTT) |
| PHIEUTHUEPHONG | + | - | - |
| TINHTRANGPHIEU | - | - | +(maTT) |

**5.** Thực thể : Phiếu thuê phòng

5.1. R1: “Ngày trả phòng thực tế phải lớn hơn hoặc bằng ngày nhận phòng”.

RBTV miền liên thuộc tính .

Bối cảnh: CHITIETPHIEUTHUEPHONG

,∀ c ∈ CHITIETPHIEUTHUEPHONG : c.ngayTPTT >= c.ngayNP

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Xóa | Sửa |
| CHITIEPHIEUTHUEPHONG | + | - | +(ngayTPTT, ngayNP) |

5.2. R2 :”Tình trạng phiếu thuê phòng có hai trạng thái: Phiếu đã chưa được lập hóa đơn (1), phiếu đã được lập hóa đơn(2).”

RBTV miền liên thuộc tính

Bối cảnh:HOADON, PHIEUTHUEPHONG , TINHTRANGPHIEU

∀ hd ∈HOADON, tp ∈ PHIEUTHUEPHONG,

ttp ∈ TINHTRANGPHIEU: hd.maPTP=tp.PTP∧( ttp.maTT = 1 ttp.maTT= 2)

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | Thêm | Xóa | Sửa |
| HOADON | + | - | - |
| PHIEUTHUEPHONG | + |  | +( maTT) |
| TINHTRANGPHIEU | - | - | +(maTT) |

**6** .Thực thể: Phiếu dịch vụ

6.1. R1 :”Tình trạng phiếu dịch vụ chỉ có một trong hai trạng thái: Phiếu đã chưa được lập phiếu hóa đơn (1), phiếu đã được lập hóa đơn(2).”

RBTV miền liên thuộc tính

Bối cảnh:HOADON, PHIEUDICHVU , TINHTRANGPHIEU

∀ hd ∈HOADON, dv ∈ PHIEUDICHVU,

ttp ∈ TINHTRANGPHIEU: hd.maPDV=dv.maPDV∧ (ttp.maTT = 1 ttp.maTT= 2)

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | Thêm | Xóa | Sửa |
| HOADON | + | - | - |
| PHIEUDICHVU | + | - | +(maTT) |
| TINHTRANGPHIEU | - | - | +(maTT) |

7.Thực thể: Hóa đơn

7.1. R2:“Tổng tiền thanh toán phải bằng tổng tiền phòng và tiền dịch vụ nhân với thuế giá trị gia tăng”.  ( trường hợp có phiếu dịch vụ và phiếu thuê phòng)

RBTV miền liên thuộc tính.

Bối cảnh: PHIEUTHUEPHONG, HOADON, PHIEUDICHVU

∀ p ∈ PHIEUTHUEPHONG, h ∈ HOADON, p1 ∈ PHIEUDICHVU: p.maPTP = h.maPTP ∧ p1.maPDV =  h.maPDV→h.tongTien = (p.tienPhong + p1.tienDV)\*1.1

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Xóa | Sửa |
| PHIEUTHUEPHONG | + | - | +( tienPhong) |
| HOADON | + | + | - |
| PHIEUDICHVU | + | - | (tienDV) |

# **CHƯƠNG 3 : CÁC CÔNG NGHỆ SỬ DỤNG**

## Oracle Database

Oracle Database được giới thiệu và phát triển vào năm 1979 bởi Oracle Corporation, đây là một hệ quản trị cơ sở dữ liệu (RDBMS) được xây dựng dựa trên SQL – một trong các ngôn ngữ lập trình chuẩn hóa, được xây dựng với mục đích hỗ trợ quản trị cơ sở dữ liệu của doanh nghiệp. Đa phần các nhà phân tích đều sử dụng công cụ này nhằm quản lý cơ sở dữ liệu và truy vấn dữ liệu của mình.

Kiến trúc của Oracle Database sử dụng cấu trúc bảng phân theo hàng và phân theo cột nhằm kết nối các phần tử, tạo sự liên kết trong nhiều bảng khác nhau. Từ đó, giúp cho người dùng không phải tốn nhiều công sức lưu trữ dữ liệu qua các bảng khác nhau để xử lý nữa..Cơ sở dữ liệu Oracle cũng hỗ trợ lập trình bằng Java và các chương trình được viết trên PL/SQL.

So với các hệ quản trị cơ sở dữ liệu khác như SQL Server, MySQL,.. thì Oracle Database chiếm không gian lưu trữ ít hơn, có sự ổn định cao, tạo ra hệ thống quản trị dữ liệu quy mô lớn với tốc độ truy vấn nhanh, chính xác. Tuy nhiên , Oracle Database có chi phí bản quyền khá lớn, nên mức độ tiếp cận của các doanh nghiệp nhỏ với hệ quản trị này khá khó khăn.

* + - 1. **Sql Developer:**

SQL Developer là công cụ quản trị và phát triển cơ sở dữ liệu đồ họa miễn phí của Oracle , SQL Developer cung cấp các trình soạn thảo mạnh mẽ để làm việc với SQL, PL / SQL, giúp thực hiện các câu truy vấn, kiểm tra, gỡ lỗi cho các chương trình cơ sở dữ liệu.

SQL Developer hỗ trợ cho cơ sở dữ liệu Oracle và sẽ chạy trên bất kỳ hệ điều hành nào hỗ trợ Java.

* + - 1. **Eclipse:**

Eclipse IDE là một môi trường phát triển tích hợp (IDE) cho Java và các ngôn ngữ lập trình khác như C, C ++, PHP, và Ruby … Môi trường phát triển được cung cấp bởi Eclipse bao gồm các công cụ phát triển Java Eclipse (JDT) cho Java, Eclipse CDT cho C/C ++, và Eclipse PDT cho PHP, và một số thứ khác

Eclipse IDE được viết chủ yếu bằng Java và một số phần khác viết bằng C / C ++. Eclipse có thể chạy trên các hệ điều hành chính như Windows, Mac và Linux.

* + - 1. **JDBC**

JDBC (Java Database Connectivity) là một API tiêu chuẩn (Application Program Interface) giúp cho Java thực hiện kết nối với nhiều cơ sở dữ liệu được cài đặt trên các hệ quản trị cơ sở dữ liệu như SQL Server, Oracle,.., để thực hiện các tác vụ xử lý bằng câu lệnh SQL (truy vấn, thêm, xóa, sửa).

Có 4 loại JDBC driver:

+ Loại 1: JDBC – ODBC Bridge

+ Loại 2: Native API Driver

+ Loại 3: Network Protocol

+ Loại 4: Native-Protocol

Thông dụng nhất và hiệu quả nhất là loại 4. Driver loại 4 còn được gọi là Native - Protocol được viết thuần túy bằng Java và là loại hiệu quả nhất. Chúng cho phép kết nối trực tiếp vào cơ sở dữ liệu, cung cấp kết quả tối ưu và cho phép lập trình viên thực hiện các chức năng tùy thuộc vào cơ sở dữ liệu cụ thể. Điều này đã tạo ra tính cơ động cao nhất là khi bạn cần thay đổi cơ sở dữ liệu bên dưới một ứng dụng. Loại driver này thường được dùng cho các ứng dụng phân tán cao.

* + - 1. **Draw.io.**

Được thiết kế bởi Seibert Media, draw.io là một giải pháp lưu đồ được thiết kế để giúp các nhà phát triển, quản trị viên mạng, nhà phân tích CNTT và nhà thiết kế sử dụng chức năng kéo và thả để tạo và xuất bản các sơ đồ. Phần mềm cho phép bạn chọn các thiết kế từ các template có sẵn hoặc bạn có thể tự tạo ra các mô hình sơ đồ tự chọn. Tính năng kéo và thả được tích hợp giúp việc tạo một sơ đồ hoặc biểu đồ đẹp mắt trở nên rất đơn giản.

* + - 1. **Google Drive:**

Google Drive là dịch vụ lưu trữ và đồng bộ hóa tập tin được tạo bởi Google. Dịch vụ lưu trữ này giúp người dùng có thể lưu trữ tập tin trên đám mây. Điều này giúp người sử dụng có thể chia sẻ tập tin và chỉnh sửa tài liệu, chia sẻ file, chia sẻ hình ảnh… một cách dễ dàng.

Trên Google Drive người dùng dễ dàng chia sẻ tập tin như tài liệu, PDF, hình ảnh, video và thư mục với người khác, tìm kiếm các tập tin theo tên và nội dung, đính kèm tập tin có dung lượng lớn trong Gmail,

# **CHƯƠNG 4 : XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC**

1. **Trigger:**

**Định nghĩa:**

* Trigger là một loại stored procedure đặc biệt (không có tham số) được thực thi một cách tự động khi có một sự kiện thay đổi dữ liệu. Trigger được phân loại bởi các thông số sau:
* Phân loại dựa trên thời điểm:

+ BEFORE Trigger: Được kích hoạt trước khi sự kiện được chỉ định xảy ra.

+ AFTER Trigger: Được kích hoạt sau khi sự kiện được chỉ định đã xảy ra.

* Phân loại dựa trên cấp độ:

+ Trigger cấp STATEMENT: Được kích hoạt một lần cho câu lệnh sự kiện được chỉ định.

+ Trigger cấp ROW: Được kích hoạt cho từng bản ghi bị ảnh hưởng trong sự kiện được chỉ định. (chỉ dành cho DML)

* Phân loại dự trên sự kiện:

+ DML Trigger: Được kích hoạt khi câu lệnh DML (INSERT, UPDATE, DELETE) được thi thi trên một bảng hoặc view cụ thể.

+ DDL Trigger: Được kích hoạt khi câu lệnh DDL (CREATE, ALTER) được chỉ định.

+ DATABASE Trigger: Được kích hoạt khi các lệnh hoạt động dữ liệu như đăng nhập/ đăng xuất, khởi động/tắt hoặc thông báo lỗi (LOGON, LOGOFF, STARTUP, SHUTDOWN, SERVERERROR) được chỉ định.

* Công dụng của Trigger:
* Kiểm tra ràng buộc trên nhiều quan hệ hoặc trên nhiều dòng của bảng.
* Ngăn chặn các thao tác không hợp lệ.
* Sao chép các đồng bộ của bảng.
* Cú pháp tạo Trigger:

CREATE [ OR REPLACE] TRIGGER <trigger\_name>

{BEFORE | AFTER | INSTEAD OF}

{INSERT [OR] | UPDATE [OR] | DELETE}

[OF col\_name]

ON table\_name

[REFERENCING OLD AS o NEW AS n]

[FOR EACH ROW]

[WHEN<condition for trigger to get execute>]

DECLARE

<Declaration-statements>

BEGIN

<Executable-statements>

EXCEPTION

<Exception-handling-statements>

END;

Trong đó:

* CREATE [OR REPLACE] TRIGGER : Mệnh đề này tạo một trình kích hoạt với tên đã cho hoặc ghi đè một trình kích hoạt hiện có có cùng tên.
* {BEFORE | AFTER | INSTEAD OF} Mệnh đề này chỉ ra thời điểm mà trình kích hoạt sẽ được kích hoạt. INSTEAD OF được sử dụng để tạo một trình kích hoạt trên một khung nhìn.
* {INSERT [OR] | UPDATE [OR] | DELETE}: Mệnh đề này xác định sự kiện kích hoạt. Nhiều sự kiện kích hoạt có thể được sử dụng cùng nhau được phân tách bằng từ khóa OR.
* [OF col\_name] - Mệnh đề này được sử dụng với các trình kích hoạt cập nhật. Mệnh đề này được sử dụng khi bạn chỉ muốn kích hoạt một sự kiện khi một cột cụ thể được cập nhật.
* [ON table\_name] - Mệnh đề này xác định tên của bảng hoặc view mà trình kích hoạt được liên kết tới.
* [REFERENCING OLD AS o NEW AS n] : Mệnh đề này được sử dụng để tham chiếu các giá trị cũ và mới của dữ liệu đang được thay đổi.
* [FOR EACH ROW] : Mệnh đề này được sử dụng để xác định liệu trình kích hoạt có phải kích hoạt khi mỗi hàng bị ảnh hưởng (tức là trình kích hoạt mức hàng) hay chỉ một lần khi toàn bộ câu lệnh sql được thực thi (tức là trình kích hoạt mức câu lệnh).
* WHEN (điều kiện) : Mệnh đề này chỉ có hiệu lực đối với trình kích hoạt cấp hàng. Trình kích hoạt chỉ được kích hoạt cho các hàng thỏa mãn điều kiện được chỉ định.

**Danh sách các Trigger:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Thao tác** | **Bảng** | **Nội dung** |
| 1 | BO\_SUNG\_THONG\_TIN\_KHACHHANG | Insert | KHACHHANG | Truyền vào các tham số mặc định khi vừa tạo khách hàng. |
| 2 | BO\_SUNG\_THONG\_TIN\_NHANVIEN | Insert | NHANVIEN | Bổ sung các giá trị mặc định cho nhân viên. |
| 3 | BO\_SUNG\_THONG\_TIN\_PHIEUDATTRUOC | Insert | PHIEUDATTRUOC | Bổ sung các giá trị mặc định chi phiếu đặt trước. |
| 4 | HUY\_PHIEUDATTRUOC | Delete | PHIEUDATTRUOC | Xóa phiếu đặt trước đồng thời cập nhật thông tin trong chi tiết phiếu thuê phòng và chi tiết tình trạng phòng. |
| 5 | KIEM\_TRA\_CCCDKH\_KHACHHANG | Insert or update | KHACHHANG | Các điều kiện khi thêm và cập nhật căn cước công dân khách hàng. |
| 6 | KIEM\_TRA\_CCCDNV\_NHANVIEN | Insert or update | NHANVIEN | Các điều kiện khi thêm và cập nhật căn cước công dân nhan viên. |
| 7 | KIEM\_TRA\_DIEU\_CAP\_NHAT\_NGAYNP\_CHITIETPDT | Update | CHITIETPDT | Kiểm tra điều kiện ngày nhập phòng khi cập nhật phiếu đặt phòng và chỉnh sửa tình trạng phòng. |
| 8 | KIEM\_TRA\_DIEU\_CAP\_NHAT\_NGAYTPDD\_CHITIETPDT | Update | CHITIETPDT | Kiểm tra điều kiện ngày trả phòng dự đoán khi cập nhật phiếu đặt phòng và chỉnh sửa tình trạng phòng. |
| 9 | KIEM\_TRA\_DIEU\_CAP\_NHAT\_NGAYTPTT\_CHITIETPTP | Update | CHITIETPTP | Kiểm tra điều kiện ngày trả phòng thực tế khi cập nhật phiếu thuê phòng với chỉnh sửa tình trạng phòng. |
| 10 | KIEM\_TRA\_DIEU\_KIEN\_CAP\_NHAT\_HOADON | Update | HOADON | Cập nhật thông tin bao gồm doanh số và mã loại khách hàng khi khách hàng lập xong hóa đơn. |
| 11 | KIEM\_TRA\_DIEU\_KIEN\_CAP\_NHAT\_NGAY\_CHITIETTTP | Update | CHITIETTTP | Kiểm tra các điều kiện khi cập nhật ngày bắt đầu và ngày kết thúc của tình trạng. |
| 12 | KIEM\_TRA\_DIEU\_KIEN\_CAP\_NHAT\_NGAYTPTT\_CHITIETPTP | Update | CHITIETPTP | Kiểm tra điều kiện ngày trả phòng thực tế cới ngày nhận phòng khi cập nhật phiếu thuê phòng. |
| 14 | KIEM\_TRA\_DIEU\_KIEN\_CAP\_NHAT\_SLNGUOI\_CHITIETPTP | Update | CHITIETPTP | Kiểm tra và cập nhật phụ thu khi cập nhật số lượng khách hàng. |
| 15 | KIEM\_TRA\_DIEU\_KIEN\_CTPDT | Insert. | CHITIETPDT | Kiểm tra điều kiện khi thêm chi tiết phiếu đặt trước. |
| 16 | KIEM\_TRA\_DIEU\_KIEN\_CTPDV | Insert. | CHITIETPDV | Kiểm tra các điều kiện khi thêm chi tiết phiếu dịch vụ. |
| 17 | KIEM\_TRA\_DIEU\_KIEN\_CTPTP | Insert. | CHITIETPTP | Kiểm tra điều kiện khi thêm chi tiết phiếu thuê phòng. |
| 19 | KIEM\_TRA\_DIEU\_KIEN\_PHIEUDICHVU | Insert. | PHIEUDICHVU | Kiểm tra các điều kiện khi lập phiếu dịch vụ. |
| 20 | KIEM\_TRA\_DIEU\_KIEN\_PHIEUTHUEPHONG\_TRUOC | Insert. | PHIEUTHUEPHONG | Kiểm tra thông tin khi thêm phiếu thuê phòng. |
| 21 | KIEM\_TRA\_DIEU\_KIEN\_QD | Insert. |  | Kiểm tra thông tin khi thêm quy định. |
| 22 | KIEM\_TRA\_DIEU\_KIEN\_THEM\_HOADON | Insert. |  | Kiểm tra thông tin khi thêm hóa đơn. |
| 23 | KIEM\_TRA\_EMAILNV\_NHANVIEN | Insert or update | NHANVIEN | Kiểm tra các điều kiện khi thêm và cập nhật email của nhân viên. |
| 24 | KIEM\_TRA\_GHICHU\_PHONG | Insert or update | PHONG | Kiểm tra điều kiện thông tin ghi chú của phòng. |
| 25 | KIEM\_TRA\_GIADV\_DICHVU | Insert or update | DICHVU | Kiểm tra điều kiện khi thêm hoặc cập nhật giá dịch vụ. |
| 26 | KIEM\_TRA\_GIOITINHKH\_KHACHHANG | Insert | KHACHHANG | Kiếm tra điều kiện giới tính khách hàng. |
| 27 | KIEM\_TRA\_GIOITINHNV\_NHANVIEN | Insert | NHANVIEN | Kiếm tra điều kiện giới tính nhân viên. |
| 28 | KIEM\_TRA\_MADONVI\_DICHVU | Insert or update | DICHVU | Kiểm tra điều kiện tồn tại của mã đơn vị. |
| 29 | KIEM\_TRA\_MAKH\_PHIEUDATTRUOC | Insert | PHIEUDATTRUOC | Kiểm tra điều kiện thông tin khách hàng đã có sẵn trong hệ thống. |
| 30 | KIEM\_TRA\_MAKM\_CHITIETKM | Insert or update | CHITIETKM | Kiểm tra điều kiện chương trình khuyến mãi tồn tại. |
| 31 | KIEM\_TRA\_MALKH\_CHITIETKM | Insert or update | CHITIETKM | Kiểm tra điều kiện loại khách hàng tồn tại. |
| 32 | KIEM\_TRA\_MALKH\_MAKM\_CHITIETKM | Insert or update | CHITIETKM | Kiếm tra điều kiện phần trăm khuyến mãi hợp lệ. |
| 33 | KIEM\_TRA\_MALP\_PHONG | Insert or update | PHONG | Kiểm tra điều kiện tồn tại loại phòng. |
| 34 | KIEM\_TRA\_MANV\_PHIEUDATTRUOC | Insert or update | PHIEUDATTRUOC | Kiểm ra điều kiện có thông tin nhân viên trong hệ thống khi lập phiếu đặt trước. |
| 35 | KIEM\_TRA\_MAPHONG\_CHITIETTTP | Insert or update | CHITIETTTP | Kiểm tra điều kiện tồn tại mã phòng khi thêm hoặc chỉnh sửa chi tiết tình trạng phòng. |
| 36 | KIEM\_TRA\_MATTP\_CHITIETTTP | Insert or update | CHITIETTTP | Kiểm tra mã tình trạng phù hợp khi thêm hoặc cập nhât chi tiết tình trạng phòng. |
| 37 | KIEM\_TRA\_NGAY\_KHUYENMAI | Insert or update | KHUYENMAI | Kiểm tra ngày điều kiện giữa ngày bắt đầu và ngày kết thúc khuyến mãi. |
| 38 | KIEM\_TRA\_NGAYNV\_NHANVIEN | Insert | NHANVIEN | Ràng buộc giá trị ngày vào làm của nhân viên. |
| 39 | KIEM\_TRA\_NGAYSINHKH\_KHACHHANG | Insert | KHACHHANG | Kiểm tra điều kiện ngày sinh của khách hàng tại khách sạn. |
| 40 | KIEM\_TRA\_QUOCTICH\_KHACHHANG | Insert or update | KHACHHANG | Kiểm tra điều kiện quốc tịc của khách hàng tại khách san. |
| 41 | KIEM\_TRA\_SDTKH\_KHACHHANG | Insert or update. | KHACHHANG | Kiểm tra điều kiên ràng buộc số điện thoại của khách hàng tại khách sạn. |
| 42 | KIEM\_TRA\_TENDONVI\_DONVI | Insert or update | DONVI | Kiểm tra điều kiện ràng buộc tên đơn vị. |
| 43 | KIEM\_TRA\_TENDV\_DICHVU | Insert | DICHVU | Kiểm tra điều kiện ràng buộc khi thêm hoặc cập nhật tên dịch vụ. |
| 44 | KIEM\_TRA\_TENKH\_KHACHHANG | Insert or update | KHACHHANG | Kiểm tra điều kiên ràng buộc về số lượng ký tự trong tên khách hàng. |
| 45 | KIEM\_TRA\_TENKM\_KHUYENMAI | Insert | KHUYENMAI | Kiểm tra điều kiện ràng buộc về tên khuyến mãi trong chương trình khuyến mãi. |
| 46 | KIEM\_TRA\_TENNV\_NHANVIEN | Insert or update | NHANVIEN | Kiểm tra điều kiện ràng buộc số lượng ký tự trong tên nhân viên. |
| 47 | KIEM\_TRA\_TENPHONG\_PHONG | Insert or update | PHONG | Gán giá trị tên phòng với mã loại phòng và mã phòng tương ứng. |
| 48 | KIEM\_TRA\_XOA\_NHANVIEN | Delete | NHANVIEN | Kiểm tra điều kiện xóa nhan viên. |
| 49 | TRANH\_TRUNG\_LAP | Insert | CHITIETTTP | Kiểm tra điều kiện tráng trùng lập khi thêm chi tiết tình trạng phòng. |
| 50 | TRANH\_TRUNG\_LAP\_CTPTP | Insert | CHITIETPTP | Kiểm tra điều kiện tránh trùng lập khi thêm chi tiết phiếu thuê phòng. |

* Mô tả trigger:

1. Cập nhật ngày nhận phòng trong chi tiết phiếu đặt phòng.

- Tên: KIEM\_TRA\_DIEU\_CAP\_NHAT\_NGAYNP\_CHITIETPDT.

- Nội dung: kiểm tra điều kiện ngày cập nhật có còn trống hay không? Nếu còn thì tiến hành cập nhật tình trạng phòng và ngược lại thì thông báo lỗi.

- Các bước thực hiện:

Box and whisker chart

Description automatically generated with low confidence

*Hình 3: Mô tả trigger KIEM\_TRA\_DIEU\_CAP\_NHAT\_NGAYNP\_CTPDT*

[1]: Đếm số lượng của tình trạng phòng thỏa mãn yêu cầu sau:

* Mã phòng bằng với lại mã phòng cần cập nhật trong chi tiết phiếu đặt trước.
* Ngày nhận phòng mới phải lớn hơn hoặc bằng ngày bắt đầu của tình trạng phòng trống liền trước.
* Ngày nhận phòng cũ trừ một phải bé hơn hoặc bằng với ngày kết thúc của tình trạng trống trước.

[2]: Gán số lượng vừa đếm được vào biến KIEM\_TRA.

[3]: Nếu giá trị ngày nhận phòng mới bé hơn giá trị ngày nhận phòng cũ và giá trị KIEM\_TRA bằng 0 tức là không tồn tại khoảng thời gian liền trước còn trống để tiến hành cập nhật ngày nhận phòng thì thông báo cập nhật chi tiết phiếu đặt trước thất bại và kết thúc trigger.

[4]: Ngược lại, tiến hành thực hiện hàm XU\_LY với chức năng cập nhật tình trạng phòng của chi tiết phiếu đặt phòng cũ là trống sau đó tiếp tục tạo chi tiết tình trạng phòng với ngày nhận phòng là giá trị mới và ngày trả phòng là giá trị trả phòng cũ và kết thúc trigger.

CREATE OR REPLACE TRIGGER KIEM\_TRA\_DIEU\_CAP\_NHAT\_NGAYNP\_CHITIETPDT

AFTER UPDATE OF NGAYNP ON CHITIETPDT

FOR EACH ROW

DECLARE

KIEM\_TRA INT;

XU\_LY EXCEPTION;

BEGIN

SELECT COUNT(\*) INTO KIEM\_TRA

FROM CHITIETTTP

WHERE MAPHONG = :OLD.MAPHONG AND MATTP = 1 AND

TRUNC(NGAYBD) <= TRUNC(:NEW.NGAYNP) AND

TRUNC(NGAYKT) >= TRUNC(:OLD.NGAYNP - 1);

IF TRUNC(:NEW.NGAYNP) < TRUNC(:OLD.NGAYNP) AND KIEM\_TRA = 0 THEN

BEGIN

RAISE\_APPLICATION\_ERROR(-20001, 'Cập nhật chi tiết phiếu đặt phòng thất bại');

END;

ELSE

BEGIN

RAISE XU\_LY;

END;

END IF;

EXCEPTION WHEN XU\_LY THEN

BEGIN

SUA\_MATTP\_CTTTP(:OLD.MAPHONG, 1, :OLD.NGAYNP, :OLD.NGAYTPDD);

SUA\_MATTP\_CTTTP(:OLD.MAPHONG, 2, :NEW.NGAYNP, :OLD.NGAYTPDD);

END;

END;

1. Xóa phiếu đặt trước

- Tên: HUY\_PHIEUDATTRUOC

- Nội dung: xóa các chi tiết phiếu đặt trước tồn tại trong phiếu đó.

- Các bước thực hiện:

[1]: Đếm số lượng mã phiếu đặt phòng với điều kiện mã phiếu đặt phòng bằng với mã phiếu muốn xóa và mã tình trạng của phiếu sẽ bằng một tức là chưa được sử dụng trong phiếu thuê phòng.

[2]: Sau đó gán giá trị vừa đếm được vào biến KIEM\_TRA.

[3]: Nếu biến KIEM\_TRA có giá trị bằng không tức là phiếu đặt trước này không tồn tại hoặc đã được sử dụng trong phiếu thue phòng thì sẽ thông báo ra màn hình “Phiếu đặt trước này không tồn tại, vui lòng kiểm tra lại”.

[4]: Ngược lại sẽ cập nhât tình trạng của các chi tiết phiếu đặt phòng sang còn trống và xóa chi tiết phiếu đặt phòng đó.

CREATE OR REPLACE TRIGGER HUY\_PHIEUDATTRUOC

BEFORE DELETE ON PHIEUDATTRUOC

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

KIEM\_TRA INT;

CURSOR CUR\_PHONG IS SELECT MAPHONG, NGAYNP, NGAYTPDD

FROM CHITIETPDT CTPDT JOIN PHIEUDATTRUOC PDT ON PDT.MAPDT = CTPDT.MAPDT

WHERE PDT.MAPDT = :OLD.MAPDT AND MATT = 1;

V\_PHONG PHONG.MAPHONG%TYPE;

V\_NGAYNP CHITIETPDT.NGAYNP%TYPE;

V\_NGAYTPDD CHITIETPDT.NGAYTPDD%TYPE;

BEGIN

SELECT COUNT(\*) INTO KIEM\_TRA

FROM PHIEUDATTRUOC

WHERE MAPDT = :OLD.MAPDT AND MATT = 1;

IF KIEM\_TRA = 0 THEN

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Phiếu đặt trước này không tồn tại, vui lòng kiểm tra lai');

END;

ELSE

BEGIN

OPEN CUR\_PHONG;

LOOP

FETCH CUR\_PHONG INTO V\_PHONG, V\_NGAYNP, V\_NGAYTPDD;

EXIT WHEN CUR\_PHONG%NOTFOUND;

SUA\_MATTP\_CTTTP(V\_PHONG, 1, V\_NGAYNP, V\_NGAYTPDD);

END LOOP;

CLOSE CUR\_PHONG;

DELETE FROM CHITIETPDT WHERE MAPDT = :OLD.MAPDT;

COMMIT;

END;

END IF;

END;

1. Đổi tên phòng.

Tên: KIEM\_TRA\_TENPHONG\_PHONG

Nội dung: thêm hoặc cập nhật tên phòng với mã phòng và mã loại phòng tương ứng.

Các bước thực hiện:

[1]: Dựa vào mã loại phòng để xét ký hiệu của phòng đó.

[2]: Gán tên phòng với dạng: <ký hiệu của mã loại phòng> + <giá trị làm tròn khi lấy mã phòng chia 2 > + <mã phòng> .

CREATE OR REPLACE TRIGGER KIEM\_TRA\_TENPHONG\_PHONG

BEFORE INSERT OR UPDATE OF MALP ON PHONG

FOR EACH ROW

BEGIN

CASE

WHEN :NEW.MALP = 1 THEN :NEW.TENPHONG :='SD';

WHEN :NEW.MALP = 2 THEN :NEW.TENPHONG :='SP';

WHEN :NEW.MALP = 3 THEN :NEW.TENPHONG :='DL';

WHEN :NEW.MALP = 4 THEN :NEW.TENPHONG :='ST';

WHEN :NEW.MALP = 5 THEN :NEW.TENPHONG :='CN';

END CASE;

CASE

WHEN :NEW.MAPHONG < 10 THEN :NEW.TENPHONG := :NEW.TENPHONG ||ROUND(:NEW.MAPHONG/2)||0||:NEW.MAPHONG;

ELSE

:NEW.TENPHONG := :NEW.TENPHONG ||ROUND(:NEW.MAPHONG/2)||:NEW.MAPHONG;

END CASE;

IF UPDATING THEN

DBMS\_OUTPUT.PUT\_LINE('Cập nhật tên phòng thành công');

END IF;

END;

1. **Store Procedure:**
   * + 1. **Store Procedure trong Oracle:**
   1. Định nghĩa:

* Stored procedure là tập hợp một hoặc nhiều câu lệnh PL/QL thành một nhóm đơn vị xử lý logic và được biên dịch để thực hiện cùng lúc mỗi khi được gọi đến, nhằm làm tăng khả năng xử lý, khả năng sử dụng chung, tăng tính bảo mật và an toàn dữ liệu, tiện ích trong phát triển.
* Stored Procedure cho tốc độ thực thi nhanh hơn so với các câu lệnh PL/SQL thông thường, Stored Procedure đã được lưu sẵn tại database server, do vậy chương trình không cần gởi hàng ngàn dòng lệnh đến database server. Nó chỉ cần gọi một dòng lệnh để thực thi Stored Procedure, điều này giúp tăng tốc độ thực thi.
* Cú pháp tạo Store Procedure:

CREATE [OR REPLACE] PROCEDURE procedure\_name

[ (parameter [, parameter])]

IS

[declaration\_section]

BEGIN

executable\_section

[EXCEPTION

exception\_section]

END [procedure\_name];

Trong đó:

* procedure\_name là tên của procedure.
* [OR REPLACE]: Khai báo tham số này thì nó sẽ xóa procedure có tên trùng với tên của procedure đang chạy.
* [ (parameter [, parameter])] là các tham số truyền vào procedure.
* Mỗi tham số truyền vào được xác định bởi ba loại như sau:
* IN: Đây là kiểu mặc định, tham số này sẽ là dữ liệu đầu vào
* OUT: Tham số này sẽ là dữ liệu đầu ra.
* IN OUT: Là tham số đặc biệt, vừa là đầu vào vừa là đầu ra, và thường thì giá trị tại đầu ra sẽ bị thay đổi.
  1. . Danh sách Store Procedure:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Tham số đầu vào** | **Bảng** | **Nội dung** |
| 1 | HIEN\_THI\_THEO\_MATTP | V\_TTP int,  V\_NGAYBD date,  V\_NGAYKT date,  V\_PHIEU int,  V\_TT int. | CHITIETTTP,  PHONG, LOAIPHONG, PHIEUDATTRUOC, CHITIETPDT, PHIEUTHUEPHONG, CHITIETPTP | Thao tác hiển thị các thông tin khi nhập vào tình trạng phòng tương ứng. |
| 2 | SUA\_CCCDKH\_KH | V\_KH int,  V\_CCCDKH int. | KHACHHANG | Sửa căn cước công dân của khách hàng. |
| 3 | SUA\_CCCDNV\_NV | V\_NV int,  V\_CCCDNV int | NHANVIEN | Sửa căn cước công dân của nhân viên. |
| 4 | SUA\_DENNGAY\_KM | V\_KM int,  V\_DN date. | KHUYENMAI | Sửa ngày kết thúc trong chương trình khuyến mãi. |
| 5 | SUA\_EMAIL\_NV | V\_NV int,  V\_E varchar2(30). | NHANVIEN | Sửa email của nhân viên. |
| 6 | SUA\_GHICHU\_PHONG | V\_P int,  V\_GHICHU varchar2(100). | PHONG | Sửa ghi chú của phòng. |
| 7 | SUA\_MADONVI\_DICHVU | V\_DV int,  V\_GIADV int. | DICHVU | Thay đổi mã đơn vị của dịch vụ. |
| 8 | SUA\_MADV\_CTPDV | V\_PDV int,  V\_DV int,  V\_NGAYSD date. | CHITIETPDV | Sửa dịch vụ trong chi tiết phiếu dịch vụ. |
| 9 | SUA\_MALP\_PHONG | V\_P int,  V\_LP int. | PHONG | Sửa mã loại phòng của phòng. |
| 10 | SUA\_MATT\_PDT | V\_PDT int,  V\_TT int. | PHIEUDATTRUOC | Sửa mã tình trạng của phiếu đặt trước. |
| 11 | SUA\_MATT\_PDV | V\_PDV int,  V\_TT int. | PHIEUDICHVU | Sửa mã tình trạng của phiếu dịch vụ. |
| 12 | SUA\_MATT\_PTP | V\_PTP int,  V\_TT int. | PHIEUTHUEPHONG | Sửa mã tình trạng của phiếu thuê phòng. |
| 13 | SUA\_MATTP\_CTTTP | V\_PHONG int,  V\_TTP int  V\_NGAYBD date,  V\_NGAYKT date. | CHITIETTTP | Sửa mã tình trạng phòng của phòng. |
| 14 | SUA\_PASSWORD\_NV | V\_NV int,  V\_PASSWORD varchar2(30).. | NHANVIEN | Sửa mật khẩu của nhân viên. |
| 15 | SUA\_PHANTRAMKM\_CTKM | V\_KM int,  V\_LKH int,  V\_PTKM number(3, 2). | KHUYENMAI | Sửa phần trăm khuyến mãi của chi tiết khuyến mãi. |
| 16 | SUA\_SDTKH\_KH | V\_KH int.  V\_SDTKH varchar2(10). | KHACHHANG | Sửa số điện thoại của khách hàng. |
| 17 | SUA\_SLDV\_CTPDV | V\_PDV int,  V\_DV int,  V\_SLDV int. | CHITIETPDV | Sửa số lượng dịch vụ của chi tiết phiếu dịch vụ. |
| 18 | SUA\_TENDONVI\_DONVI | V\_DONVI int,  V\_TENDONVI varchar2(30). | DONVI | Sửa tên đơn vị trong đơn vị. |
| 19 | SUA\_TENDV\_DICHVU | V\_DV int,  V\_TENDV varchar2(30). | DICHVU | Sửa tên dịch vụ. |
| 20 | SUA\_TENND\_NV | V\_NV int,  V\_TENND varcahr2(30). | NHANVIEN | Sửa tên tài khoản của nhân viên. |
| 21 | SUA\_TENND\_QD | V\_QD int,  V\_TENQD | QUYDINH | Sửa tên quy định. |
| 22 | SUA\_TUNGAY\_KM | V\_KM int,  V\_TN date. | KHUYENMAI | Sửa ngày bắt đầu của chương trình khuyến mãi. |
| 23 | THEM\_CTKM | V\_KM int,  V\_LKH int,  V\_PTKM number(3,2). | CHITIETKM | Thêm chi tiết khuyến mãi. |
| 24 | THEM\_CTPDT | V\_PDT int,  V\_PHONG int,  V\_NGAYNP date,  V\_NGAYTPDD date. | CHITIETPDT | Thêm chi tiết phiếu đặt trước. |
| 25 | THEM\_CTPDV | V\_PDV int,  V\_DV int,  V\_SLDV int. | CHITIETPDV | Thêm chi tiết phiếu dịch vụ. |
| 26 | THEM\_CTPTP | V\_PTP int,  V\_PHONG int,  V\_NGAYNP date,  V\_NGAYTPTT date,  V\_SLNGUOI int. | CHITIETPTP | Thêm chi tiết phiếu thuê phòng. |
| 27 | THEM\_CTTTP | V\_P int,  V\_TTP int,  V\_BD date,  V\_KT date. | CHITIETTTP | Thêm chi tiết tình trạng phòng. |
| 28 | THEM\_DICHVU | V\_TENDV  V\_GIADV int,  V\_MADONVI int. | DICHVU | Thêm dịch vụ. |
| 29 | THEM\_DONVI | V\_TENDONVI | DONVI | Thêm đơn vị. |
| 30 | THEM\_HOADON | V\_KH int,  V\_NV int. | PHIEUTHUEPHONG, CHITIETPTP, PHIEUDICHVU, CHITIETPDV, KHUYENMAI, CHITIETKM, KHACHHANG,  DICHVU, PHONG, LOAIPHONG | Thêm hóa đơn. |
| 31 | THEM\_KH | V\_TEN varchar2(30),  V\_CCCD varchar2(12),  C\_NS date,  V\_GT varchar2(4),  V\_SDT varchar2(10),  V\_QT varchar2(30). | KHACHHANG | Thêm khách hàng. |
| 32 | THEM\_KM | V\_TEN varchar2(30),  V\_TUNGAY date,  V\_DENNGAY date. | KHUYENMAI | Thêm khuyến mãi. |
| 33 | THEM\_NV | V\_TEN varchar2(30),  V\_NS date,  V\_GT varchar2(4),  V\_EMAIL varchar2(30),  V\_CCCD varchar2(12),  V\_CHUCVU varchar2(30),  V\_MAQL int. | NHANVIEN | Thêm nhân viên. |
| 34 | THEM\_PDT | V\_KH int,  V\_NV int. | PHIEUDATTRUOC | Thêm phiếu đặt trước. |
| 35 | THEM\_PDV | V\_KH int,  V\_NV int. | PHIEUDICHVU | Thêm phiếu dịch vụ. |
| 36 | THEM\_PHONG | V\_GC varchar2(100),  V\_LP int. | PHONG | Thêm phòng. |
| 37 | THEM\_PTP | V\_KH int,  V\_NV int. | PHIEUTHUEPHONG | Thêm phếu thuê phòng. |
| 38 | THEM\_QD | V\_TENQD varchar2(30),  V\_MOTA varchar2(500). | QUYDINH | Thêm quy định. |
| 39 | XOA\_CCCD\_NV | V\_CCCDNV varchar2(12). | NHANVIEN | Xóa nhân viên dựa vào căn cước công dân. |
| 40 | XOA\_EMAIL\_NV | V\_EMAILNV varchar2(30). | NHANVIEN | Xóa nhân viên dựa vào email. |
| 41 | XOA\_MAPDT\_PDT | V\_KH int. | PHIEUDATTRUOC | Xóa phiếu đặt trước dựa vào mã phiếu đặt trước. |
| 42 | XOA\_MAKM\_KM | V\_KM int. | KHUYENMAI | Xóa khuyến mãi dựa vào mã khuyến mãi. |
| 43 | XOA\_MAKM\_MALKH\_CTKM | V\_KM int,  V\_LKH int. | CHITIETKM | Xóa chi tiết khuyến mãi dựa vào mã khuyến mãi và mã loại khách hàng. |
| 44 | XOA\_MANV\_NV | V\_NV int. | NHANVIEN | Xóa nhân viên dựa vào mã nhân viên. |
| 45 | XOA\_MAPDT\_PDT | V\_PDT int. | PHIEUDATTRUOC | Xóa phiếu đặt trước dựa vào mã phiếu đặt trước. |
| 46 | XOA\_MAPDV\_MADV\_CTPDV | V\_PDV int,  V\_DV int. | CHITIETPDV | Xóa chi tiết phiếu dịch vụ dựa vào mã phiếu dịch vụ và mã dịch vụ. |
| 47 | XOA\_MAQD\_QD | V\_QD int. | QUYDINH | Xóa quy định dựa vào mã quy định. |
| 48 | XOA\_TENQD\_QD | V\_TENQD varchar2(30). | QUYDINH | Xóa quy định dựa cào tên quy định. |
| 49 | XUAT\_HOADON | V\_MAHD int. | HOADON,  NHANVIEN, KAHCHHANG, CHITIETPTP, PHONG, CHITIETPDV, DICHVU | Xuất hóa đơn. |

* Mô tả Procedure:

1. Xem thông tin phòng, phiếu thuê phòng, phiếu đặt phòng.

- Tên: hien\_thi\_theo\_mattp.

- Nội dung: Liệt kê thông tin các phòng tương ứng với mã tình trạng phòng.

TH1: Nếu mã tình trạng phòng là còn trống thì hiển thị danh sách phòng bao gồm mã phòng, tên phòng và giá phòng. Đồng thời liệt kê các tình trạng của phòng tương ứng như sau:

* Nếu có thông tin ngày bắt đầu và ngày kết thúc tình trạng phòng thì chỉ thể hiện tình trạng những phòng còn trống tại ngày đó.
* Ngược lại thì hiển thị tất cả thông tin các ngày còn trống tồn tại trong danh sách.

TH2: Nếu nhập mã tình trạng phòng là đã đặt thì sẽ hiển thị danh sách các phiếu đặt phòng bao gồm mã phiếu, ngày lập phiếu và mã khách hàng tương ứng. Đồng thời thông tin sẽ được hiển thị dựa trên các tình huống sau:

* Nếu không nhập mã tình trạng phiếu:
  + Và không nhập thông tin bao gồm ngày bắt đầu, ngảy kết thúc của tình trạng phòng và cả thông tin phiếu đặt trước thì hiển thị tất cả thông tin phiếu đặt phòng có trong cơ sở dữ liệu.
  + Và khi nhập ngày bắt đầu và ngày kết thúc của tình trạng phòng thì hệ thống sẽ liệt kê các thông tin phiếu có phòng được đặt trong khoảng thời gian đó.
  + Đối với trường hợp chỉ có thông tin phiếu thì sẽ hiển thị cụ thể thông tin chi tiết của chính phiếu đó.
* Nếu có nhập mã tình trạng phiếu, thông tin sẽ được rút ngắn lại làm hai trường hợp là phiếu đặt trước chưa được áp dụng vào trong phiếu thuê phòng và phiếu đặt trước đã được thêm vào phiếu thuê phòng. Trong tình huống này cũng xét ba trường hợp tương tự như khi không nhập mã tình trạng phiếu.

TH3: Nếu nhập mã tình trạng phòng là thuê thì sẽ hiển thị danh sách các phiếu thuê phòng bao gồm mã phiếu, tổng tiền phòng, mã khách hàng tương ứng và mã phiếu đặt trước nếu có. Thông tin cũng được hiển thị dựa vào sáu trường hợp như phiếu đặt trước.

- Các bước thực thi:

[1]: Kiểm tra thông tin tình trạng phòng nhập vào thuộc loại tình trạng nào?

[1.1]: Nếu tình trạng phòng là trống thì nhảy sang bước [2].

[1.2]: Nếu tình trạng phòng là đã đặt thì nhảy sang bước [4].

[1.3]: Nếu tình trạng phòng là thuê thì nhảy sang bước [6].

[1.4]: Nếu tình trang phòng không thuộc một trong ba trường hợp trên thì thông báo lỗi “Không tồn tại tình trạng này, vui lòng kiểm tra lại”.

[2]: Sử dụng hàm KIEM\_TRA để xét tình huống danh sách muốn xem:

[2.1]: Nếu ngày bắt đầu và ngày kết thúc là NULL thì sẽ lấy thông tin tất cả các phòng có tình trạng là một xuyên suốt trong dữ liệu.

[2.2]: Nếu thông tin ngày bắt đầu và ngày kết thúc được nhập thì sẽ giới hạn thông tin các phòng trống trong khoảng thời gian đó.

[2.3]: Ngược lại, sẽ không lấy dữ liệu và kết thúc quy trình xử lý hàm.

[3]: Thực hiện hàm HIEN\_THI\_CHI\_TIET\_1 để lấy thông tin trong danh sách phòng bao gồm thông tin chính chính là mã phòng, tên phòng và giá phòng, kèm theo các thông tin chi tiết như ngày bắt đầu và ngày kết thúc của tình trạng là trống. Sau khi xuất dữ liệu thành công thì hoàn tất thủ tục.

[4]: Sử dụng hàm KIEM\_TRA để xét tình huống danh sách muốn xem:

[4.1]: Nếu mã tình trạng phiếu là NULL thì:

[4.1.1]: Nếu các thông tin như ngày bắt đầu tình trạng phòng, ngày kết thúc tình trạng phòng và mã phiếu là NULL thì: sẽ lấy tất cả thông tin phiếu đặt trước hiện có của hệ thống.

[4.1.2]: Nếu thông tin ngày bắt đầu và ngày kết thúc khác NULL nhưng mã phiếu là NULL thì: sẽ giới hạn lại khoảng thời gian sẽ nhận phòng và trả phòng dự đoán trong chi tiết phiếu đặt trước.

[4.1.3]: Nếu thông tin ngày bắt đầu và ngày kết thúc là NULL nhưng thông tin mã phiếu khác NULL thì chỉ hiển thị chính xác thông tin phiếu đó.

[4.1.4]: Ngược lại, sẽ không lấy dữ liệu và kết thúc quy trình xử lý hàm.

[4.2]: Nếu tình trạng phiếu có giá trị là chưa sử dụng hoặc sử dụng thì truy xuất các thông tin tương ứng với tình trạng như đã xét ở trường hợp [4.1].

[4.3]: Ngược lại, sẽ không lấy dữ liệu và kết thúc quy trình xử lý hàm.

[5]: Thực hiện hàm HIEN\_THI\_CHI\_TIET\_2 để lấy thông tin trong danh sách phiếu đặt phòng bao gồm thông tin chính chính là mã phiếu, ngày đặt phòng và mã khách hàng, kèm theo các thông tin chi tiết như mã phòng đặt, ngày nhận phòng và ngày trả phòng dự đoán của tình trạng là đã đặt. Sau khi xuất dữ liệu thành công thì hoàn tất thủ tục.

[6]: Sử dụng hàm KIEM\_TRA để xét tình huống danh sách phiếu thue phòng muốn xem giống như các tình huống trong trường hợp [4].

[7]: Thực hiện hàm HIEN\_THI\_CHI\_TIET\_3 để lấy thông tin trong danh sách phiếu thuê phòng bao gồm thông tin chính chính là mã phiếu, tổng tiền phòng, mã khách hàng và mã phiếu đặt trước nếu có, kèm theo các thông tin chi tiết như mã phòng đặt, ngày nhận phòng và ngày trả phòng dự đoán củatình trạng là thuê. Sau khi xuất dữ liệu thành công thì hoàn tất thủ tục.

create or replace PROCEDURE HIEN\_THI\_THEO\_MATTP (V\_TTP TINHTRANGPHONG.MATTP%TYPE,

V\_NGAYBD CHITIETTTP.NGAYBD%TYPE,

V\_NGAYKT CHITIETTTP.NGAYKT%TYPE,

V\_PHIEU PHIEUTHUEPHONG.MAPTP%TYPE,

V\_TT TINHTRANGPHIEU.MATT%TYPE

)

IS

BO\_VALUE VARCHAR2(5);

TYPE T\_PHONG IS RECORD ( MPHONG PHONG.MAPHONG%TYPE,

TPHONG PHONG.TENPHONG%TYPE,

GPHONG LOAIPHONG.GIAPHONG%TYPE

);

TYPE T\_HH IS TABLE OF T\_PHONG INDEX BY PLS\_INTEGER;

TYPE T\_CTTTP IS TABLE OF CHITIETTTP%ROWTYPE INDEX BY PLS\_INTEGER;

TYPE T\_PDT IS TABLE OF PHIEUDATTRUOC%ROWTYPE INDEX BY PLS\_INTEGER;

TYPE T\_CTPDT IS TABLE OF CHITIETPDT%ROWTYPE INDEX BY PLS\_INTEGER;

TYPE T\_PTP IS TABLE OF PHIEUTHUEPHONG%ROWTYPE INDEX BY PLS\_INTEGER;

TYPE T\_CTPTP IS TABLE OF CHITIETPTP%ROWTYPE INDEX BY PLS\_INTEGER;

V\_HH T\_HH;

V\_CTTTP T\_CTTTP;

V\_PDT T\_PDT;

V\_CTPDT T\_CTPDT;

V\_PTP T\_PTP;

V\_CTPTP T\_CTPTP;

PROCEDURE HIEN\_THI\_CHI\_TIET\_1 IS

HH\_INDEX PLS\_INTEGER;

CTTTP\_INDEX PLS\_INTEGER;

BEGIN

HH\_INDEX := V\_HH.FIRST;

LOOP

EXIT WHEN HH\_INDEX IS NULL;

DBMS\_OUTPUT.PUT\_LINE(V\_HH(HH\_INDEX).MPHONG ||' '|| V\_HH(HH\_INDEX).TPHONG||' '|| V\_HH(HH\_INDEX).GPHONG);

SELECT DISTINCT \* BULK COLLECT INTO V\_CTTTP

FROM CHITIETTTP

WHERE MATTP = 1 AND MAPHONG = V\_HH(HH\_INDEX).MPHONG AND ( ( (BO\_VALUE = 'TRUE') AND TRUNC(NGAYBD) <= TRUNC(V\_NGAYBD) AND TRUNC(V\_NGAYKT) <= TRUNC(NGAYKT))

OR (BO\_VALUE = 'FALSE')

)

ORDER BY NGAYBD;

FOR CTTTP\_INDEX IN 1 .. V\_CTTTP.COUNT

LOOP

DBMS\_OUTPUT.PUT\_LINE(' '||V\_CTTTP(CTTTP\_INDEX).NGAYBD||' '|| V\_CTTTP(CTTTP\_INDEX).NGAYKT);

END LOOP;

HH\_INDEX := V\_HH.NEXT(HH\_INDEX);

END LOOP;

END;

PROCEDURE HIEN\_THI\_CHI\_TIET\_2 IS

PDT\_INDEX PLS\_INTEGER;

CTPDT\_INDEX PLS\_INTEGER;

BEGIN

PDT\_INDEX := V\_PDT.FIRST;

LOOP

EXIT WHEN PDT\_INDEX IS NULL;

DBMS\_OUTPUT.PUT\_LINE(V\_PDT(PDT\_INDEX).MAPDT ||' '|| V\_PDT(PDT\_INDEX).NGAYDT||' '|| V\_PDT(PDT\_INDEX).MAKH);

SELECT DISTINCT \* BULK COLLECT INTO V\_CTPDT

FROM CHITIETPDT

WHERE MAPDT = V\_PDT(PDT\_INDEX).MAPDT AND ( (BO\_VALUE = 'TH1' OR BO\_VALUE = 'TH4')

OR ((BO\_VALUE = 'TH2' OR BO\_VALUE = 'TH5')AND TRUNC(NGAYNP) <= TRUNC(V\_NGAYBD) AND TRUNC(V\_NGAYKT) <= TRUNC(NGAYTPDD))

OR ((BO\_VALUE = 'TH3' OR BO\_VALUE = 'TH6')AND MAPDT = V\_PDT(PDT\_INDEX).MAPDT)

)

ORDER BY MAPHONG, NGAYNP;

FOR CTPDT\_INDEX IN 1 .. V\_CTPDT.COUNT

LOOP

DBMS\_OUTPUT.PUT\_LINE(' '||V\_CTPDT(CTPDT\_INDEX).MAPHONG||' '|| V\_CTPDT(CTPDT\_INDEX).NGAYNP||' '|| V\_CTPDT(CTPDT\_INDEX).NGAYTPDD);

END LOOP;

PDT\_INDEX := V\_PDT.NEXT(PDT\_INDEX);

END LOOP;

END;

PROCEDURE HIEN\_THI\_CHI\_TIET\_3 IS

PTP\_INDEX PLS\_INTEGER;

CTPTP\_INDEX PLS\_INTEGER;

BEGIN

PTP\_INDEX := V\_PTP.FIRST;

LOOP

EXIT WHEN PTP\_INDEX IS NULL;

DBMS\_OUTPUT.PUT\_LINE(V\_PTP(PTP\_INDEX).MAPTP ||' '|| V\_PTP(PTP\_INDEX).TIENPHONG||' '|| V\_PTP(PTP\_INDEX).MAKH||' '|| V\_PTP(PTP\_INDEX).MAPDT);

SELECT DISTINCT \* BULK COLLECT INTO V\_CTPTP

FROM CHITIETPTP

WHERE MAPTP = V\_PTP(PTP\_INDEX).MAPTP AND ( (BO\_VALUE = 'TH1' OR BO\_VALUE = 'TH4')

OR ((BO\_VALUE = 'TH2' OR BO\_VALUE = 'TH5')AND TRUNC(NGAYNP) <= TRUNC(V\_NGAYBD) AND TRUNC(V\_NGAYKT) <= TRUNC(NGAYTPTT))

OR ((BO\_VALUE = 'TH3' OR BO\_VALUE = 'TH6')AND MAPTP = V\_PTP(PTP\_INDEX).MAPTP)

)

ORDER BY MAPHONG, NGAYNP;

FOR CTPTP\_INDEX IN 1 .. V\_CTPTP.COUNT

LOOP

DBMS\_OUTPUT.PUT\_LINE(' '||V\_CTPTP(CTPTP\_INDEX).MAPHONG||' '|| V\_CTPTP(CTPTP\_INDEX).NGAYNP||' '|| V\_CTPTP(CTPTP\_INDEX).NGAYTPTT

||' '|| V\_CTPTP(CTPTP\_INDEX).SLNGUOI||' '|| V\_CTPTP(CTPTP\_INDEX).PHUTHU);

END LOOP;

PTP\_INDEX := V\_PTP.NEXT(PTP\_INDEX);

END LOOP;

END;

PROCEDURE KIEMTRA IS

BEGIN

IF V\_TTP = 1 THEN

BO\_VALUE := CASE

WHEN V\_NGAYBD IS NOT NULL AND V\_NGAYKT IS NOT NULL THEN 'TRUE'

ELSE 'FALSE'

END;

ELSE

BO\_VALUE := CASE

WHEN V\_TT IS NULL THEN( CASE

WHEN V\_NGAYBD IS NULL AND V\_NGAYKT IS NULL AND V\_PHIEU IS NULL THEN 'TH1'

WHEN V\_NGAYBD IS NOT NULL AND V\_NGAYKT IS NOT NULL AND V\_PHIEU IS NULL THEN 'TH2'

WHEN V\_NGAYBD IS NULL AND V\_NGAYKT IS NULL AND V\_PHIEU IS NOT NULL THEN 'TH3'

END

)

WHEN V\_TT IS NOT NULL THEN ( CASE

WHEN V\_NGAYBD IS NULL AND V\_NGAYKT IS NULL AND V\_PHIEU IS NULL THEN 'TH4'

WHEN V\_NGAYBD IS NOT NULL AND V\_NGAYKT IS NOT NULL AND V\_PHIEU IS NULL THEN 'TH5'

WHEN V\_NGAYBD IS NULL AND V\_NGAYKT IS NULL AND V\_PHIEU IS NOT NULL THEN 'TH6'

END

)

END;

END IF;

END;

BEGIN

LOCK TABLE PHONG IN ROW SHARE MODE;

LOCK TABLE CHITIETTTP IN ROW SHARE MODE;

LOCK TABLE PHIEUDATTRUOC IN ROW SHARE MODE;

LOCK TABLE CHITIETPDT IN ROW SHARE MODE;

LOCK TABLE PHIEUTHUEPHONG IN ROW SHARE MODE;

LOCK TABLE CHITIETPTP IN ROW SHARE MODE;

IF V\_TTP = 1 THEN

BEGIN

KIEMTRA;

SELECT DISTINCT P.MAPHONG, TENPHONG, GIAPHONG

BULK COLLECT INTO V\_HH

FROM PHONG P JOIN LOAIPHONG LP ON P.MALP = LP.MALP JOIN CHITIETTTP CTTTP ON CTTTP.MAPHONG = P.MAPHONG

WHERE ( (BO\_VALUE = 'TRUE') AND TRUNC(NGAYBD) <= TRUNC(V\_NGAYBD) AND TRUNC(V\_NGAYKT) <= TRUNC(NGAYKT) )

OR (BO\_VALUE = 'FALSE')

ORDER BY P.MAPHONG;

HIEN\_THI\_CHI\_TIET\_1;

END;

ELSIF V\_TTP = 2 THEN

BEGIN

KIEMTRA;

SELECT DISTINCT PDT.MAPDT, PDT.NGAYDT, PDT.MAKH, PDT.MANV, PDT.MATT BULK COLLECT INTO V\_PDT

FROM PHIEUDATTRUOC PDT JOIN CHITIETPDT CTPDT ON CTPDT.MAPDT = PDT.MAPDT

WHERE ( (BO\_VALUE = 'TH1' ) OR

(BO\_VALUE = 'TH2' AND TRUNC(NGAYNP) <= TRUNC(V\_NGAYBD) AND TRUNC(V\_NGAYKT) <= TRUNC(NGAYTPDD)) OR

(BO\_VALUE = 'TH3' AND PDT.MAPDT = V\_PHIEU)

)

OR (

MATT = V\_TT AND

( (BO\_VALUE = 'TH4' ) OR

(BO\_VALUE = 'TH5' AND TRUNC(NGAYNP) <= TRUNC(V\_NGAYBD) AND TRUNC(V\_NGAYKT) <= TRUNC(NGAYTPDD)) OR

(BO\_VALUE = 'TH6' AND PDT.MAPDT = V\_PHIEU)

)

)

ORDER BY MAPDT;

HIEN\_THI\_CHI\_TIET\_2;

END;

ELSIF V\_TTP = 3 THEN

BEGIN

KIEMTRA;

SELECT DISTINCT PTP.MAPTP, PTP.TIENPHONG, PTP.MAKH, PTP.MANV, PTP.MAPDT, PTP.MATT BULK COLLECT INTO V\_PTP

FROM PHIEUTHUEPHONG PTP JOIN CHITIETPTP CTPTP ON CTPTP.MAPTP = PTP.MAPTP

WHERE ( (BO\_VALUE = 'TH1' ) OR

(BO\_VALUE = 'TH2' AND TRUNC(NGAYNP) <= TRUNC(V\_NGAYBD) AND TRUNC(V\_NGAYKT) <= TRUNC(NGAYTPTT)) OR

(BO\_VALUE = 'TH3' AND PTP.MAPTP = V\_PHIEU)

)

OR (

MATT = V\_TT AND

( (BO\_VALUE = 'TH4' ) OR

(BO\_VALUE = 'TH5' AND TRUNC(NGAYNP) <= TRUNC(V\_NGAYBD) AND TRUNC(V\_NGAYKT) <= TRUNC(NGAYTPTT)) OR

(BO\_VALUE = 'TH6' AND PTP.MAPTP = V\_PHIEU)

)

)

ORDER BY MAPTP;

HIEN\_THI\_CHI\_TIET\_3;

END;

ELSE

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Không tồn tại tình trạng phòng này, vui lòng kiểm tra lại');

END;

END IF;

END;

1. Cập nhật mã tình trạng phòng từ còn trống sang đã đặt hoặc đang thuê và ngược lại.

- Tên: sua\_mattp\_ctttp.

- Nội dung: Cập nhật trạng thái của một phòng cụ thể ứng với tham số v\_phong. Nếu khách hàng tiến hành hủy phiếu đặt phòng hoặc cập nhật thông tin ngày nhạp phòng hoặc ngày trả phòng trong phiếu thuê phòng hoặc đặt trước thì tình trạng sẽ chuyển sang còn trống là tình trạng 1. Nếu khách hàng có nhu cầu đặt phòng tại khách sạn trước khi đến thì đã đặt là 2 và nếu khách hàng muốn thuê phòng trực tiếp tại khách sạn thì phòng sẽ có tình trạng là còn trống sang đang được thuê và tình trạng là 3.

- Các bước thực hiện:

[1]: Kiểm tra mã tình trạng phòng sẽ được nhập vào:

[1.1]: Nếu tình trạng phòng được nhập vào là “1” tức là chuyển sang tình trạng còn trống thì sẽ nhảy sang bước [2].

[1.2]: Nếu tình trạng phòng được nhập vào là “2” hoặc “3” tức là chuyển sang tình trạng đã đặt hoặc đang thuê thì sẽ nhảy sang bước [6].

[1.3]: Ngược lại, sẽ thông báo lỗi “Cập nhật chi tiết tình trạng phòng thất bại” và kết thúc quy trình xử lý hàm.

[2]: Cập nhật chi tiết tình trạng phòng theo ngày bắt đầu đến ngày kết thúc là “1” thông qua hàm XU\_LY\_MATTP\_1.

Diagram

Description automatically generated

*Hình 4: Mô tả procedure SUA\_MATTP\_CTTTP trường hợp XU\_LY\_MATTP\_1.*

[3]: Gán giá trị ngày bắt đầu đã nhập vào một biến tạm thời gọi là TEMP. (TEMP = E)

[4]: Kiểm tra khoảng thời gian liền trước ngày bắt đầu (A), tình trạng phòng có phải là trống hay không?

[4.1]: Nếu tồn tại khoảng thời gian liền trước có tình trạng là còn trống:

[4.1.1]: Gán các giá trị ngày bắt đầu và ngày kết thúc của tình trạng liền trước đó lần lượt vào biến V\_BD1, V\_KT1. (V\_BD1 = A; V\_KT1 = B)

[4.1.2]: Xóa tình trạng của khoảng thời gian liền trước. [A, B]

[4.1.3]: Cập nhật ngày bắt đầu tình trạng của chi tiết phòng vừa cập nhật là ngày bắt đầu của tình trạng liền trước. [E, F] → [A, F]

[4.1.4]: Cập nhật giá trị của TEMP là ngày bắt đầu của tình trạng trước. (TEMP = A).

[4.2]: Nếu không có thì nhảy sang bước [5].

[5]: Kiểm tra khoảng thời gian liền sau ngày kết thúc, tình trạng phòng có phải là trống hay không?

[5.1]: Nếu tồn tại khoảng thời gian liền sau có tình trạng là còn trống:

[5.1.1]: Lấy giá trị lấy các giá trị ngày bắt đầu và ngày kết thúc của tình trạng liền sau đó. (V\_BD2 = C; V\_KT2 = D)

[5.1.2]: Xóa tình trạng của khoảng thời gian liền sau. [C, D]

[5.1.3]: Cập nhật ngày kết thúc tình trạng của chi tiết phòng vừa nhập là ngày kết thúc của tình trạng liền sau. [TEMP, F] → [TEMP, D]

[5.2]: Nếu không có hoặc thực hiện xong bước [5.1] thì sẽ kết thúc quy trình xử lý hàm.

[6]: Lấy ra các giá trị lần lượt là ngày bắt đầu và ngày kết thúc của phòng phù hợp với khoảng thời gian khách hàng muốn thuê phòng hoặc đặt phòng.

[7]: Cập nhật chi tiết tình trạng phòng từ còn trống sang đã đặt hoặc đang thuê, cập nhật ngày bắt đầu là ngày nhận phòng và ngày két thúc là ngày trả phòng.

[8]: Kiểm tra thông tin ngày:

Diagram, timeline

Description automatically generated

*Hình 5: Mô tả procedure SUA\_MATTP\_CTTTP trường hợp XU\_LY\_MATTP\_2\_HOAC\_3.*

[8.1]: Nếu ngày bắt đầu tình trạng bằng ngày nhận phòng thì thêm chi tiết tình trạng phòng từ ngày trả phòng cộng 1 đến ngày kết thúc ban đầu là có tình trạng còn trống.

[8.2]: Nếu ngày kết thúc tình trạng bằng với ngày trả phòng và ngày nhận phòng sẽ lớn hơn ngày ngày bắt đầu của tình trạng thì thêm một chi tiết tình trang phòng từ ngày bắt đầu ban đầu đến ngày nhận phòng trừ một là tình trạng trống.

[8.3]: Nếu khoảng thời gian tiến hành đặt phòng hoặc thuê phòng nằm trong khoảng thời gian còn trống thì tiến hành thêm hai thông tin chi tiết tình trạng phòng lần lượt là ngày bắt đầu của tình trạng trống là ngày trả phòng cộng 1 đến ngày kết thúc ban đầu và ngày bắt đầu của tình trạng trống thứ hai là ngày bắt đầu ban đầu đến ngày nhận phòng trừ một.

[8.4]: Kết thúc quy trình xử lý hàm.

create or replace PROCEDURE SUA\_MATTP\_CTTTP ( V\_PHONG PHONG.MAPHONG%TYPE,

V\_TTP TINHTRANGPHONG.MATTP%TYPE,

V\_NGAYBD CHITIETTTP.NGAYBD%TYPE,

V\_NGAYKT CHITIETTTP.NGAYKT%TYPE

)

IS

XU\_LY\_MATTP\_1 EXCEPTION;

XU\_LY\_MATTP\_2\_HOAC\_3 EXCEPTION;

----------------XU\_LY\_MATTP\_1--------------------

V\_BD CHITIETTTP.NGAYBD%TYPE;--CU

V\_KT CHITIETTTP.NGAYKT%TYPE;--CU

V\_BD1 CHITIETTTP.NGAYBD%TYPE;--CU

V\_KT1 CHITIETTTP.NGAYKT%TYPE;--CU

TEMP CHITIETTTP.NGAYBD%TYPE;

DEM\_DANG\_TRUOC INT;

DEM\_DANG\_SAU INT;

----------------XU\_LY\_MATTP\_2\_HOAC\_3---------------

V\_BD3 CHITIETTTP.NGAYBD%TYPE;

V\_KT3 CHITIETTTP.NGAYKT%TYPE;

V\_TP CHITIETTTP.MATTP%TYPE;

BEGIN

LOCK TABLE CHITIETTTP IN ROW EXCLUSIVE MODE;

IF V\_TTP = 1 THEN

BEGIN

RAISE XU\_LY\_MATTP\_1;

END;

ELSIF V\_TTP = 2 OR V\_TTP = 3 THEN

BEGIN

RAISE XU\_LY\_MATTP\_2\_HOAC\_3;

END;

END IF;

EXCEPTION

WHEN XU\_LY\_MATTP\_1 THEN

BEGIN

UPDATE CHITIETTTP

SET MATTP = V\_TTP

WHERE MAPHONG = V\_PHONG AND TRUNC(NGAYBD) = TRUNC(V\_NGAYBD) AND TRUNC(NGAYKT) = TRUNC(V\_NGAYKT);

TEMP := V\_NGAYBD;

SELECT COUNT(\*) INTO DEM\_DANG\_TRUOC

FROM CHITIETTTP

WHERE MAPHONG = V\_PHONG AND MATTP = 1 AND TRUNC(NGAYKT + 1) = TRUNC(V\_NGAYBD);

SELECT COUNT(\*) INTO DEM\_DANG\_SAU

FROM CHITIETTTP

WHERE MAPHONG = V\_PHONG AND MATTP = 1

AND TRUNC(NGAYBD - 1) = TRUNC(V\_NGAYKT);

IF DEM\_DANG\_TRUOC = 1 THEN

BEGIN

SELECT NGAYBD, NGAYKT INTO V\_BD, V\_KT

FROM CHITIETTTP

WHERE MAPHONG = V\_PHONG AND MATTP = 1 AND TRUNC(NGAYKT + 1) = TRUNC(V\_NGAYBD);

DELETE FROM CHITIETTTP

WHERE MAPHONG = V\_PHONG AND MATTP = 1

AND TRUNC(NGAYKT)=TRUNC(V\_KT) AND TRUNC(V\_BD) = TRUNC(NGAYBD);

UPDATE CHITIETTTP

SET NGAYBD = V\_BD

WHERE MAPHONG = V\_PHONG AND MATTP = 1 AND

TRUNC(NGAYBD) = TRUNC(TEMP) AND

TRUNC(NGAYKT) = TRUNC(V\_NGAYKT);

TEMP :=V\_BD;

END;

END IF;

IF DEM\_DANG\_SAU = 1 THEN

BEGIN

SELECT NGAYBD, NGAYKT INTO V\_BD1, V\_KT1

FROM CHITIETTTP

WHERE MAPHONG = V\_PHONG AND MATTP = 1

AND TRUNC(NGAYBD - 1) = TRUNC(V\_NGAYKT);

DELETE FROM CHITIETTTP

WHERE MAPHONG = V\_PHONG AND MATTP = 1 AND TRUNC(NGAYKT)=TRUNC(V\_KT1) AND TRUNC(V\_BD1) = TRUNC(NGAYBD);

UPDATE CHITIETTTP

SET NGAYKT = V\_KT1

WHERE MAPHONG = V\_PHONG AND MATTP = 1 AND

TRUNC(NGAYBD) = TRUNC(TEMP) AND

TRUNC(NGAYKT) = TRUNC(V\_NGAYKT);

END;

END IF;

END;

WHEN XU\_LY\_MATTP\_2\_HOAC\_3 THEN

BEGIN

SELECT NGAYBD, NGAYKT INTO V\_BD3, V\_KT3

FROM CHITIETTTP

WHERE MAPHONG = V\_PHONG AND MATTP = 1 AND

TRUNC(NGAYBD) <= TRUNC(V\_NGAYBD) AND

TRUNC(NGAYKT) >= TRUNC(V\_NGAYKT);

UPDATE CHITIETTTP

SET MATTP = V\_TTP, NGAYBD = V\_NGAYBD , NGAYKT = V\_NGAYKT

WHERE MAPHONG = V\_PHONG AND

TRUNC(NGAYBD) <= TRUNC(V\_NGAYBD) AND

TRUNC(NGAYKT) >= TRUNC(V\_NGAYKT);

IF TRUNC(V\_NGAYBD) = TRUNC(V\_BD3) AND TRUNC(V\_NGAYKT) < TRUNC(V\_KT3) THEN

BEGIN

INSERT INTO CHITIETTTP(MAPHONG, MATTP, NGAYBD, NGAYKT)

VALUES(V\_PHONG, 1, V\_NGAYKT + 1, V\_KT3);

END;

ELSIF TRUNC(V\_NGAYKT) = TRUNC(V\_KT3) AND TRUNC(V\_NGAYBD) > TRUNC(V\_BD3) THEN

BEGIN

INSERT INTO CHITIETTTP(MAPHONG, MATTP, NGAYBD, NGAYKT)

VALUES(V\_PHONG, 1, V\_BD3, V\_NGAYBD - 1);

END;

ELSIF TRUNC(V\_NGAYBD) > TRUNC(V\_BD3) AND TRUNC(V\_NGAYKT) < TRUNC(V\_KT3) THEN

BEGIN

INSERT INTO CHITIETTTP(MAPHONG, MATTP, NGAYBD, NGAYKT)

VALUES(V\_PHONG, 1, V\_NGAYKT + 1, V\_KT3);

INSERT INTO CHITIETTTP(MAPHONG, MATTP, NGAYBD, NGAYKT)

VALUES(V\_PHONG, 1, V\_BD3, V\_NGAYBD - 1);

END;

END IF;

END;

WHEN NO\_DATA\_FOUND THEN

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Cập nhật chi tiết tình trạng phòng thất bại');

END;

END;

1. Thêm hóa đơn khi khách hàng tiến hành thanh toán.

- Tên: them\_hoadon.

- Nội dung: lập hóa đơn khi khách hàng đã lập phiếu thuê phòng hoặc phiếu dịch vụ tại khách sạn và muốn thành toán.

- Các bước thực hiện:

[1]: Tạo ba con trỏ lần lượt là CUR\_PHONG, CUR\_KHUYENMAI, CUR\_DICHVU. Trong đó:

[1.1]: CUR\_PHONG sẽ lấy thông tin bao gồm ngày nhập phòng, ngày trả phòng thực tế, giá phòng và phụ thu theo chi tiết từng phòngcủa phiếu thuê phòng.

[1.2]: CUR\_KHUYENMAI: sẽ lấy thông tin bao gồm phần trăm khuyến mãi kèm theo khoảng thời gian tổ chức chương trình.

[1.3]: CUR\_DICHVU: sẽ lấy thông tin ngày sử dụng, số lượng dịch vụ và giá dịch vụ tương ứng với dịch vụ được sử dụng trong chi tiết của phiếu.

[2]: Kiểm tra thông tin khách hàng nhập vào có tồn tại hay không?

[2.1]: Nếu mã khách hàng có trong dữ liệu thì tiến hành lập hóa đơn ở bước [3].

[2.2]: Ngược lại, kết thúc quy trình xử lý hàm.

[3]: Lấy thông tin mã loại khách hàng của khách hàng tương ứng và gán vào biến V\_LKH.

[4]: Gán giá trị tien\_phong = 0.

[5]: Kiểm tra khách hàng đó có đang sử dụng phiếu thuê phòng hay không?

[5.1]: Nếu có thì tiến hành tính tiền phòng như sau:

[5.1.1]: Lấy ra mã phiếu thuê phòng tương ứng.

[5.1.2]: Thực hiện tính tổng tiền phiếu thuê phòng theo từng ngày và xét giá trị bắt đầu từ ngày nhận phòng cho đến hết ngày trả phòng thực tế.

[5.1.3]: Tại ngày nhận phòng, xét xem ngày đó có chương trình khuyến mãi nào không?

[5.1.3.1]: Nếu có thì chọn phần trắm khuyến mãi lớn nhất trong ngày đó.

[5.1.3.2]: Ngược lại thì không cần tính.

[5.1.4]: Sau đó tính tiền phòng như sau: lấy giá trị tien\_phong + giá phòng \* (1 + phụ thu - phần trăm khuyến mãi lớn nhất).

[5.1.5]: Giá trị ngày nhận phòng tăng thêm một.

[5.1.6]: Kiếm tra nếu ngày nhận phòng nhỏ hơn hoặc bằng ngày trả phòng thực tế thì quay lại bước [5.1.3]. Nếu không thì nhảy sang bước [6].

[5.2]: Nếu không thì nhảy sang bước [6].

[6]: Gán giá trị tien\_dv = 0.

[7]: Kiểm tra khách hàng đó có đang sử dụng phiếu dịch vụ hay không?

[7.1]: Nếu có thì tiến hành tính tiền phòng như sau:

[7.1.1]: Lấy ra mã phiếu thuê dịch vụ tương ứng.

[7.1.2]: Thực hiện tính tổng tiền phiếu dịch vụ theo từng ngày sử dụng.

[7.1.3]: Trong từng ngày sử dụng, xét xem ngày đó có chương trình khuyến mãi nào không?

[7.1.3.1]: Nếu có thì chọn phần trắm khuyến mãi lớn nhất trong ngày đó.

[7.1.3.2]: Ngược lại thì không cần tính.

[7.1.4]: Sau đó tính tiền dịch vụ như sau: lấy giá trị tien\_dv + giá dịch vụ \* (1 - phần trăm khuyến mãi lớn nhất).

[7.1.5]: Và sẽ quay lại bước [7.1.3] khi vần cong chi tiết phiếu dịch vụ chưa xét. Ngược lại thì nhảy sang bước [7.1.8].

[7.2]: Nếu không thì nhảy sang bước [8].

[7]: Tính tổng tiền hóa đơn theo công thức sau: tong\_tien = (tien\_phong + tien\_dv) \* 1.1.

[8]: Thêm thông tin háo đơn vào dữ liệu hóa đơn.

[9]: Kiểm tra đã thêm hóa đơn thành công không?

[9.1]: Nếu có thì xuất thông báo thành công.

[9.2]: Ngược lại thì kết thúc quy trình xử lý hàm.

create or replace PROCEDURE "THEM\_HOADON" ( V\_KH IN KHACHHANG.MAKH%TYPE,

V\_NV IN NHANVIEN.MANV%TYPE

)

IS

V\_HD HOADON.MAHD%TYPE;

KIEM\_TRA INT;

V\_PTP PHIEUTHUEPHONG.MAPTP%TYPE;

V\_TIENPHONG PHIEUTHUEPHONG.TIENPHONG%TYPE;

V\_GIAPHONG LOAIPHONG.GIAPHONG%TYPE;

V\_NGAYNP CHITIETPTP.NGAYNP%TYPE;

V\_NGAYTP CHITIETPTP.NGAYTPTT%TYPE;

V\_PHUTHU CHITIETPTP.PHUTHU%TYPE;

CURSOR CUR\_PHONG IS SELECT NGAYNP, NGAYTPTT, PHUTHU, GIAPHONG

FROM CHITIETPTP CTPTP JOIN PHONG P ON CTPTP.MAPHONG = P.MAPHONG

JOIN LOAIPHONG LP ON P.MALP = LP.MALP

WHERE CTPTP.MAPTP = V\_PTP;

V\_TUNGAY KHUYENMAI.TUNGAY%TYPE;

V\_DENNGAY KHUYENMAI.DENNGAY%TYPE;

V\_PHANTRAMKM CHITIETKM.PHANTRAMKM%TYPE;

V\_MAXKM CHITIETKM.PHANTRAMKM%TYPE;

V\_LKH LOAIKHACHHANG.MALKH%TYPE;

CURSOR CUR\_KHUYENMAI IS SELECT TUNGAY, DENNGAY, PHANTRAMKM

FROM CHITIETKM CTKM JOIN KHUYENMAI KM ON CTKM.MAKM = KM.MAKM

WHERE MALKH = V\_LKH;

V\_PDV PHIEUDICHVU.MAPDV%TYPE;

V\_TIENDV PHIEUDICHVU.TIENDV%TYPE;

V\_DV DICHVU.MADV%TYPE;

V\_GIADV DICHVU.GIADV%TYPE;

V\_SLDV CHITIETPDV.SLDV%TYPE;

V\_NGAYSD CHITIETPDV.NGAYSD%TYPE;

CURSOR CUR\_DICHVU IS SELECT NGAYSD, SLDV, GIADV

FROM CHITIETPDV CTPDV JOIN DICHVU DV ON CTPDV.MADV =DV.MADV

WHERE CTPDV.MAPDV = V\_PDV;

V\_TONGTIEN HOADON.TONGTIEN%TYPE;

BEGIN

V\_HD := HD\_S.NEXTVAL;

SELECT COUNT(\*) INTO KIEM\_TRA

FROM KHACHHANG

WHERE MAKH = V\_KH;

IF KIEM\_TRA = 1 THEN

BEGIN

--TIM MALKH

SELECT MALKH INTO V\_LKH

FROM KHACHHANG

WHERE MAKH = V\_KH;

--TINH TIEN PHONG

SELECT COUNT(MAPTP) INTO KIEM\_TRA

FROM PHIEUTHUEPHONG

WHERE MAKH = V\_KH AND MATT = 1;

V\_TIENPHONG := 0;

IF KIEM\_TRA <> 0 THEN

BEGIN

SELECT MAPTP INTO V\_PTP

FROM PHIEUTHUEPHONG

WHERE MAKH = V\_KH AND MATT = 1;

OPEN CUR\_PHONG;

LOOP

FETCH CUR\_PHONG INTO V\_NGAYNP, V\_NGAYTP, V\_PHUTHU, V\_GIAPHONG;

EXIT WHEN CUR\_PHONG%NOTFOUND;

WHILE TRUNC(V\_NGAYNP) < TRUNC(V\_NGAYTP)

LOOP

V\_MAXKM := 0;

OPEN CUR\_KHUYENMAI;

LOOP

FETCH CUR\_KHUYENMAI INTO V\_TUNGAY, V\_DENNGAY, V\_PHANTRAMKM;

EXIT WHEN CUR\_KHUYENMAI%NOTFOUND;

IF TRUNC(V\_NGAYNP) >= TRUNC(V\_TUNGAY) AND TRUNC(V\_NGAYTP) <= TRUNC(V\_DENNGAY) THEN

IF V\_MAXKM = 0 OR V\_MAXKM < V\_PHANTRAMKM THEN

V\_MAXKM := V\_PHANTRAMKM;

END IF;

END IF;

END LOOP;

CLOSE CUR\_KHUYENMAI;

V\_TIENPHONG := V\_TIENPHONG + (V\_GIAPHONG \* (1 + V\_PHUTHU - V\_MAXKM));

V\_NGAYNP := V\_NGAYNP + 1;

END LOOP;

END LOOP;

CLOSE CUR\_PHONG;

--CAP NHAT TIENPHONG VAO PHIETHUEPHONG

UPDATE PHIEUTHUEPHONG

SET TIENPHONG = V\_TIENPHONG

WHERE MAPTP = V\_PTP;

END;

END IF;

--TINH TIEN DICHVU

SELECT COUNT(\*) INTO KIEM\_TRA

FROM PHIEUDICHVU

WHERE MAKH = V\_KH AND MATT = 1;

V\_TIENDV := 0;

IF KIEM\_TRA <> 0 THEN

BEGIN

SELECT MAPDV INTO V\_PDV

FROM PHIEUDICHVU

WHERE MAKH = V\_KH AND MATT = 1;

OPEN CUR\_DICHVU;

LOOP

FETCH CUR\_DICHVU INTO V\_NGAYSD, V\_SLDV, V\_GIADV;

EXIT WHEN CUR\_DICHVU%NOTFOUND;

V\_MAXKM := 0;

OPEN CUR\_KHUYENMAI;

LOOP

FETCH CUR\_KHUYENMAI INTO V\_TUNGAY, V\_DENNGAY, V\_PHANTRAMKM;

EXIT WHEN CUR\_KHUYENMAI%NOTFOUND;

IF TRUNC(V\_NGAYSD) BETWEEN TRUNC(V\_TUNGAY) AND TRUNC(V\_DENNGAY) THEN

IF V\_MAXKM = 0 OR V\_MAXKM < V\_PHANTRAMKM THEN

V\_MAXKM := V\_PHANTRAMKM;

END IF;

END IF;

END LOOP;

CLOSE CUR\_KHUYENMAI;

V\_TIENDV := V\_TIENDV + (V\_SLDV \* V\_GIADV) \* (1 - V\_MAXKM);

END LOOP;

CLOSE CUR\_DICHVU;

--CAP NHAT TIENDV VAO PHIEUDIHVU

UPDATE PHIEUDICHVU

SET TIENDV = V\_TIENDV

WHERE MAPDV = V\_PDV;

END;

END IF;

--TINH TIEN HOADON

V\_TONGTIEN := (V\_TIENPHONG + V\_TIENDV) \* 1.1;

END;

END IF;

INSERT INTO HoaDon (maHD, tongTien, maKH, maNV, maPTP, maPDV)

VALUES(V\_HD, V\_TONGTIEN, V\_KH, V\_NV, V\_PTP, V\_PDV);

SELECT COUNT(\*) INTO KIEM\_TRA

FROM HOADON

WHERE MAHD = V\_HD;

IF KIEM\_TRA > 0 THEN

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Them hoa don thanh cong');

END;

END IF;

END;

* + - 1. **Function trong Oracle:**
  1. Định nghĩa:
* Trong Oracle, một function là một hàm dùng để xử lý một chức năng nào đó và nó sẽ trả về một giá trị duy nhất. Có một số function có sẵn như hàm MIN, MAX, COUNT, ... đó là những function mà Oracle đã cung cấp sẵn cho chúng ta.
* Cú pháp tạo Function:

CREATE [OR REPLACE] FUNCTION function\_name

[ (parameter [, parameter])]

RETURN return\_datatype

IS | AS

[declaration\_section]

BEGIN

executable\_section

[EXCEPTION

exception\_section]

END [function\_name];

Trong đó:

* function\_name là tên của function.
* [OR REPLACE]: Khai báo tham số này thì nó sẽ xóa function có tên trùng với tên của function đang chạy.
* [ (parameter [, parameter])] là các tham số truyền vào function.
* Mỗi tham số truyền vào được xác định bởi ba loại như sau:
* IN: Đây là kiểu mặc định, tham số này sẽ là dữ liệu đầu vào
* OUT: Tham số này sẽ là dữ liệu đầu ra.
* IN OUT: Là tham số đặc biệt, vừa là đầu vào vừa là đầu ra, và thường thì giá trị tại đầu ra sẽ bị thay đổi.

2.2 Danh sách các Function:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Tham số đầu vào** | **Bảng** | **Nội dung** |
| 1 | LAY\_MAPDT | V\_KH int,  V\_NV int. | PHIEUDATTRUOC. | Lấy mã phiếu đặt trước có tình trạng là chưa sử dụng với thông tin đầu vào là mã khách hàng và mã nhân viên lập phiếu. |
| 2 | LAY\_MAPDV | V\_KH int,  V\_NV int. | PHIEUDICHVU. | Lấy mã phiếu dịch vụ có tình trạng là chưa sử dụng với thông tin đầu vào là mã khách hàng và mã nhân viên lập phiếu. |
| 3 | LAY\_MAPTP | V\_KH int,  V\_NV int. | PHIEUTHUEPHONG. | Lấy mã thuê phòng có tình trạng là chưa sử dụng với thông tin đầu vào là mã khách hàng và mã nhân viên lập phiếu. |
| 4 | TINH\_PHU\_THU\_KH | V\_PHIEU int. | KHACHHANG, LOAIKHACHHANG, CHITITIETPTP. | Tính chi phí phụ thu của khách hàng khi tiến hành thuê phòng. |
| 5 | TINH\_TIEN\_DICHVU | V\_PDV int. | KHACHHANG, LOAIKHACHHANG, KHUYENMAI, CHITIETKM, DICHVU, CHITIETPDV, PHIEUDICHVU. | Tính tổng tiền dịch vụ của phiếu dịch vụ. |
| 6 | TINH\_TIEN\_PHONG | V\_PTP int. | KHACHHANG, LOAIKHACHHANG, KHUYENMAI, CHITIETKM, PHONG, LOAIPHONG, CHITIETPTP, PHIEUTHUEPHONG. | Tính tổng tiền phòng của phiếu thuê phòng. |

* + - 1. **Transaction trong Oracle:**
  1. Định nghĩa:
* Transaction là một đơn vị công việc logic có chứa một hoặc nhiều câu lệnh SQL. Các tác động của tất cả các câu lệnh SQL trong một transaction có thể được tất cả commit tức là được áp dụng vào cơ sở dữ liệu hoặc tất cả được khôi phục (hoàn tác) khỏi cơ sở dữ liệu.
  1. Cấu trúc transaction:
* Một transaction đều có bắt đầu và kết thúc.
* Bắt đầu transaction:
  + Transaction sẽ bắt đầu khi câu lệnh SQL đầu tiên được thực thi, bao gồm câu lệnh DDL, DML hoặc SET TRANSACTION.
  + Khi một transaction bắt đầu, Oracle sẽ gán giao dịch vào undo data segment, vùng nhớ này sẽ giúp hệ quản trị duy trì các bản ghi về các hành động của các transaction, từ đó có thể thực hiện các việc như:

+ Rollback một transaction đang hoạt động.

+ Rollback một transaction đã chấm dứt.

+ Cung cấp cơ chế đọc nhất quán.

* Kết thúc transaction:
  + Transaction có thể kết thúc bằng nhiều trường hợp như :

+ Khi có lệnh COMMIT hay ROLLBACK được phát ra mà không có savepoint.

+ Khi thi hành một lệnh DDL như CREATE, DROP, hoặc ALTER…, các lệnh này sẽ tự động commit.

+ Kết nối của người dùng đến hệ quản trị bị ngắt đột ngột, transaction sẽ tự động commit.

+ Các ứng dụng đang kết nối đến hệ quản trị bị dừng đột ngột,

transaction sẽ tự động rollback.

* 1. Transaction control:
* Transaction control dùng để quản lý sự ảnh hưởng khi các câu lệnh DDL được thực hiện trên cơ sở dữ liệu, có một số câu lệnh chính như:

+ COMMIT: câu lệnh này sẽ kết thúc transaction và lưu những thay đổi dữ liệu vào database đồng thời xóa tất cả các savepoints và giải phóng khóa của transaction.

+ ROLLBACK: câu lệnh này sẽ hủy bỏ sự thay đổi dữ liệu trong giao tác hiện hành và các savepoint đồng thời giải phóng khóa của transaction. Tuy nhiên nếu câu lệnh là ROLLBACK TO SAVEPOINT thì nó sẽ chỉ hoàn tác các thay đổi kể từ savepoints cuối cùng mà không kết thúc toàn bộ transaction.

+ SAVEPOINT: Xác định 1 điểm trong transaction để rollback về khi có sự cố.

Ví dụ:

|  |  |
| --- | --- |
| **Session 1** | **Explanation** |
| commit;  Commit complete. | Session 1 thực hiện câu lệnh commit để kết thúc mọi giao tác thực hiện trước đó. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| set transaction name 'update\_room';  Transaction name succeeded.. | Session 1 bắt đầu thực hiện giao tác và đặt tên là ‘update\_room’. |
| update loaiphong  set giaphong = 990000  where malp = 1;  1 row updated. | Câu lệnh này thực hiện cập nhật giá phòng của mã loại phòng bằng 1 từ 800000 lên 990000. |
| **Session 1** | **Explanation** |
| savepoint after\_malp\_1;  Savepoint created. | Session 1 tiếp tục thực hiện câu lệnh tạo ‘SAVEPOINT’ và đặt tên là ‘after\_malp\_1’.  Câu lệnh này cho phép hoàn tác tất cả các công việc được thực hiện dưới ‘SAVEPOINT’, chỉ giữ lại các thao tác trước nó. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 990000  2 1500000  3 2000000  4 1700000  5 3000000 | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| update loaiphong  set giaphong = 2500000  where malp = 5;  1 row updated. | Sau đó tiếp tục thực hiện lệnh cập nhật giá phòng của mã loại phòng là 5. |
| savepoint after\_malp\_5;  Savepoint created. | Câu lệnh này tạo ‘SAVEPOINT’ có tên là’ after\_malp\_5’, cho phép thay đổi các giao dịch và hoàn tác về vị trí này. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 990000  2 1500000  3 2000000  4 1700000  5 2500000 | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| rollback to savepoint after\_malp\_1;  Rollback complete. | Session 1 thực hiện câu lệnh ‘ROLLBACK TO SAVEPOINT’ với điểm lưu có tên là ‘after\_malp\_1’. Sau khi câu lệnh thực thi, giao tác sẽ lùi về giao dịch thứ 4 và hoàn tác công việc cập nhật giá phòng của mã loại phòng là 5. Tuy nhiên, giao dịch ‘update\_room’ vẫn chưa kết thúc do chưa xuất hiên câu lệnh ‘COMMIT’ hoặc ‘ROLLBACK’. |
| **Session 1** | **Explanation** |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 990000  2 1500000  3 2000000  4 1700000  5 3000000 | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| rollback to savepoint after\_malp\_5;  ORA-01086: savepoint ‘after\_malp\_5’ never established in thí session or is invalid. | Sau khi thực hiện câu lệnh ‘ROLLBACK TO SAVEPOINT after\_malp\_1’, toàn độ các savepoint trong khoảng từ ‘SAVEPOINT after\_malp\_1’ đến ‘ROLLBACK TO SAVEPOINT after\_malp\_1’ đã bị hủy. Chính vì vậy, session 1 không thể quay về vị trí ‘SAVEPOINT after\_malp\_5’. |
| update loaiphong  set giaphong = 2900000  where malp = 5;  1 row updated. | Session 1 tiếp tục thực hiện câu lệnh cập nhật giá phòng cho loại phòng số 5. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 990000  2 1500000  3 2000000  4 1700000  5 2900000 | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| rollback;  Rollback complete. | Câu lệnh này sẽ khôi phục lại tất cả các câu lệnh trong transaction ‘update\_room’ và kết thúc transaction. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| **Session 1** | **Explanation** |
| update loaiphong  set giaphong = 900000  where malp = 1;  1 row updated. | Câu lệnh này thực hiện cập nhật giá phòng của mã loại phòng bằng 1 từ 800000 lên 900000. |
| update loaiphong  set giaphong = 2800000  where malp = 5;  1 row updated. | Sau đó tiếp tục thực hiện lệnh cập nhật giá phòng của mã loại phòng là 5. |
| commit;  Commit complete. | Session 1 thực hiện câu lệnh ‘COMMIT’ để lưu thay đổi vĩnh viễn, xóa tất cả các ‘SACEPOINT’ và kết thúc transaction. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 900000  2 1500000  3 2000000  4 1700000  5 2800000 | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| rollback;  Rollback complete. | Session 1 tiếp tục thực hiện câu lệnh ‘ROLLBACK’. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 900000  2 1500000  3 2000000  4 1700000  5 2800000 | Lúc này, thông tin trong cơ sở dữ liệu LOAIPHONG không bị thay đổi. Vì trước đó, session 1 đã thực hiện câu lệnh ‘COMMIT’ nên khi sử dụng câu lệnh ‘ROLLBACK’, session 1 không thể hoàn tác lại dữ liệu ban đầu. |

* 1. Hàng đợi Transaction:
  + Khi một transaction phải chờ tài nguyên từ blocking transaction thì nó sẽ phải xếp vào hàng đợi của blocking transaction đang giữ tài nguyên đó vì vậy blocking transaction phải đảm bảo là sẽ commit hoặc rollback để transaction bị chặn có thể tiếp tục.
  + Tuy nhiên trong một số tình huống, transaction đang chờ tài nguyên vẫn bị chặn sau khi blocking transaction rollback về savepoint như trong tình huống hiển thị trong bản sau, session 1 quay trở về savepoint được tạo trước khi nó thực hiện câu lệnh DML nhưng session 2 vẫn bị chặn vì đang chờ session 1 hoàn tất.

|  |  |  |  |
| --- | --- | --- | --- |
| **Session 1** | **Session 2** | **Session 3** | **Explanation** |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | No action. | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| update loaiphong  set giaphong = 990000  where malp = 1;  1 row updated. | No action. | No action. | Câu lệnh này thực hiện cập nhật giá phòng của mã loại phòng bằng 1 từ 800000 lên 990000. |
| savepoint after\_malp\_1;  Savepoint created. | No action. | No action. | Session 1 tiếp tục thực hiện câu lệnh tạo ‘SAVEPOINT’ và đặt tên là ‘after\_malp\_1’.  Câu lệnh này cho phép lưu trữ tất cả các công việc được thực hiện trước ‘SAVEPOINT’, và hoàn tác cacsc\ công việc đằng sau nó. |
| update loaiphong  set giaphong = 2700000  where malp = 5;  1 row updated. | No action. | No action. | Sau đó tiếp tục thực hiện lệnh cập nhật giá phòng của mã loại phòng là 5. |
| **Session 1** | **Session 2** | **Session 3** | **Explanation** |
| savepoint after\_malp\_5;  Savepoint created. | No action. | No action. | Câu lệnh này tạo ‘SAVEPOINT’ có tên là’ after\_malp\_5’, cho phép thay đổi các giao dịch và hoàn tác về vị trí này. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 990000  2 1500000  3 2000000  4 1700000  5 2700000 | No action. | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| No action. | select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | No action. | Session 2 truy vấn để kiểm tra thông tin. |
| No action. | update loaiphong  set giaphong = 2400000  where malp = 5;  Prompt does not return | No action. | Session 2 thực hiện câu lệnh cập nhật giá phòng với mã loại phòng là 5.  Tuy nhiên, do session 1 đang giữa khóa malp = 5 nên session 2 sẽ chờ sesion 1 thực hiện xong và kết thúc. |
| rollback to savepoint after\_malp\_1;  Rollback complete. | No action. | No action. | Session 1 tiến hành câu lệnh ‘ROLLBACK TO SAVEPOINT after\_malp\_1’ và nhả khóa cho malp = 5. Tuy nhiên, session 2 vẫn chờ khóa bởi session 1 do sesion 1 vẫn chưa thực hiện câu lệnh ‘ROLLBACK’ hoặc ‘COMMIT’. |
| **Session 1** | **Session 2** | **Session 3** | **Explanation** |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 990000  2 1500000  3 2000000  4 1700000  5 3000000 | No action. | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| No action. | No action. | select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | Session 3 truy vấn để kiểm tra thông tin. |
| No action. | No action. | update loaiphong  set giaphong = 2900000  where malp = 5;  1 row updated. | Sau đó session 3 tiếp tục thực hiện lệnh cập nhật giá phòng của mã loại phòng là 5.  Câu lệnh lúc này thành công và kết thúc.  Nguyên nhân là do session 1 đã trả khóa trên đơn vị dữ liệu mã loại phòng là 5. |
| No action. | No action. | commit;  Commit complete. | Session 3 thực hiện câu lệnh ‘COMMIT’, lưu lại thông tin và kết thúc phiên làm việc của mình. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 990000  2 1500000  3 2000000  4 1700000  5 2900000 | No action. | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật và nhận thấy thông tin đã thay đổi. |
| commit;  Commit complete. | No action. | No action. | Session 1 tiến hành ‘COMMIT’ và xóa tất cả các ‘SAVEPOINT’ đồng thởi mở transaction locks. |
| **Session 1** | **Session 2** | **Session 3** | **Explanation** |
| No action. | 1 row updated.  commit;  Commit complete. | No action. | Session 2 nhận được khóa, cập nhật thành công và kết thúc transaction. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 990000  2 1500000  3 2000000  4 1700000  5 2400000 | No action. | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật sau khi session 2 hoàn tất. |

# **CHƯƠNG 5: XỬ LÝ TRUY XUẤT ĐỒNG THỜI**

1. **Các mức cô lập trong Oracle:**

* Transaction isolation level là các cấp độ bảo vệ tính toàn vẹn dữ liệu trong transaction. Trong một môi trường hoàn hảo, dữ liệu chính xác nhất khi tất cả các thay đổi xảy ra một cách tuần tự, thực hiện xong query này rồi mới đến cái khác, tuy nhiên trong thực tế nếu làm như vậy sẽ giảm hiệu suất của hệ thống nên ta có các cấp độ isolation để cân bằng giữa việc đảm bảo tính chính xác của dữ liệu và khả năng xử lý song song.

**Read Committed:**

* Đây là mức isolation mặc định của Oracle, nếu bạn không thiết lập gì cả thì transaction sẽ hoạt động ở mức này. Transaction ở mức cô lập này sẽ không đọc được dữ liệu đang được cập nhật bởi transaction khác mà phải đợi đến khi việc cập nhật thực hiện xong và cũng nhờ cơ chế này mà Oracle không gặp phải trường hợp Dirty- read. Tuy nhiên, mức cô lập này không ngăn transaction sửa đổi dữ liệu trên đơn vị dữ liệu mà đang được đọc bởi transaction khác (xung đột ghi), làm cho dữ liệu bị thay đổi giữa các lần truy vấn nên có thể dẫn đến vấn đề non-repeatable read hoặc Phantoms.
* Xung đột ghi trong Read Committed:
  + Transaction có mức cô lập là Read Committed sẽ xảy ra trường hợp xung đột ghi khi transaction cố gắng thực hiện cập nhật trên một hàng cũng đang được cập nhật đồng thời bởi transaction khác (blocking transaction). Transaction Read Commited sẽ đợi đến khi blocking transaction hoàn tất và giải phóng khóa trên hàng đó. Trường hợp:

+ Nếu blocking transaction thực hiện rollback, thì transaction đang chờ sẽ tiến hành thay đổi trên khóa trước đó như thể transaction khác chưa bao giờ tồn tại.

+ Nếu blocking transaction thực hiện commit và giải phóng các khóa của nó, thì transaction đang chờ sẽ tiến hành cập nhật cho hàng mới được thay đổi.

**Serializable:**

* Với mức cô lập Serializable, transaction chỉ thấy những thay đổi đã được commit trước thời điểm transaction có mức cô lập này bắt đầu và các thay đổi được thực hiện bởi chính transaction này. Không giống như Read Committed, transaction ở mức cô lập này đảm bảo rằng bất kỳ dòng nào được đọc bởi transaction tuần tự được đảm bảo giống nhau khi đọc lại. và ất kỳ truy vấn nào cũng được đảm bảo trả về cùng kết quả trong suất thời gian transaction đó hoạt động, do đó sẽ không gặp phải hiện tượng phantom reads hoặc dirty read.
* Khi một Serializable transaction cố gắng cập nhật hoặc xóa dữ liệu được thay đổi bởi một giao dịch khác và đã được commit sau khi Serializable transaction bắt đầu, cơ sở dữ liệu sẽ thông báo lỗi:

ORA-08177: Cannot serialize access for this transaction.

Nếu gặp lỗi này, ứng dụng có thể thực hiện các hành động sau:

+ Commit các công việc được thực hiện đến thời điểm đó.

+ Rollback toàn bộ giao dịch.

+ Thực hiện câu lệnh bổ sung (nhưng khác), có thể sau khi quay trở savepoint được thiết lập trước đó trong giao dịch.

**Read-only:**

* Mức cô lập Read-only tương tự như mức cô lập Serializable, nhưng các transaction chỉ được phép đọc và không cho phép sửa đổi dữ liệu trong transaction trừ khi người dùng là SYS, do đó lỗi ORA-08177 sẽ ít xảy ra. Mức cô lập Read-only rất hữu ích để tạo báo cáo trong đó nội dung phải khớp với thời điểm giao dịch bắt đầu.

1. **Cơ chế khóa:**

* Khóa là cơ chế ngăn chặn các tương tác phá hoại, là các tương tác cập nhật dữ liệu không chính xác hoặc thay đổi sai cấu trúc dữ liệu cơ bản giữa các transactiom truy cập dữ liệu được chia sẻ. Khóa đóng một vai trò quan trọng trong việc duy trì tính nhất quán và đồng thời của cơ sở dữ liệu.
* Có 3 loại Lock chính đó là: Share lock, exclusive lock và update lock
* Share Lock: hay còn gọi là read-only lock (khóa chỉ đọc) là lock mà một transaction chiếm hữu khi muốn đọc một dữ liệu. G transaction giữ Share lock được phép đọc dữ liệu, nhưng không được phép ghi. Nhiều transaction có thể đồng thời giữ Share lock trên cùng một đơn vị dữ liệu.
* Exclusive Lock: hay còn gọi là write lock là lock mà một transaction chiếm hữu khi muốn đọc và ghi dữ liệu. Tại một thời điểm chỉ có tối đa một transaction được quyền giữ Exclusive lock trên một đơn dữ liệu. Không thể thiết lập Share lock trên đơn vị dữ liệu đang có Exclusive lock.
* Update lock: Khóa dự định ghi. Update lock sử dụng khi đọc dữ liệu với dự định ghi trở lại trên dữ liệu này. Update lock là chế độ khoá trung gian giữa Share lock và Exclusive lock. Khi thực hiện thao tác ghi lên dữ liệu thì bắt buộc Update lock phải tự động chuyển thành Exclusive lock. Transaction giữ Update lock được phép ghi và đọc dữ liệu. Tại 1 thời điểm chỉ có tối đa một transaction được quyền giữ Update lock trên một đơn dữ liệu. Có thể thiết lập Share lock trên đơn vị dữ liệu đang có Update lock.
* Khóa ảnh hưởng đến sự tương tác của người đọc và người viết. Người đọc là một câu lệnh truy vấn tài nguyên, trong khi người viết là một câu lệnh sửa đổi tài nguyên. Các quy tắc sau đây tóm tắt hành vi khóa của Oracle cho người đọc và người viết:
* Một hàng chỉ bị khóa khi người viết sửa đổi.
* Khi một câu lệnh cập nhật một hàng, giao dịch chỉ có được một khóa cho hàng này. Bằng cách khóa dữ liệu bảng ở cấp độ hàng, cơ sở dữ liệu giảm thiểu sự tranh chấp cho cùng một dữ liệu.
* Nếu một giao dịch đang sửa đổi một hàng, thì khóa hàng sẽ ngăn một giao dịch khác sửa đổi cùng một hàng đồng thời.
* Bởi vì người đọc của một hàng không khóa nó, người viết có thể sửa đổi hàng này. Ngoại lệ duy nhất là câu lệnh SELECT ... FOR UPDATE, là một loại câu lệnh SELECT đặc biệt để khóa hàng mà nó đang đọc.

1. **Deadlock:**

* Deadlock xảy ra khi hai hay nhiều phiên đang chờ dữ liệu mà bị khóa lẫn nhau, dẫn đến tất cả các phiên đều bị khóa. Oracle có thể tự động phát hiện và giải quyết deadlock bằng cách rollback câu lệnh được liên kết với giao tác gây ra deadlock.
* Tình huống deadlock được thể trong trường hợp dưới đây:

+ Mô tả: Những ngày tết đến xuân về, khách sạn thu hút một lượng lớn du khách đến nơi đây. Để có thể mang lại nhiều lợi nhuận trong quá trình hoạt động, quản lý khách sạn tiến hành tăng giá phòng lên. Do quá trình tăng giá phòng của hai quản lý diễn ra tại hai địa điểm khác nhau nên họ không thẻ biết được đối tương đang cập nhật và dẫn đến tình hướng deadlock.

+ Tình huống:

|  |  |  |
| --- | --- | --- |
| **Session 1** | **Session 2** | **Explanation** |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| update loaiphong  set giaphong = giaphong \*(1.1)  where malp in (1, 3, 5);  1 row updated. | No action. | Câu lệnh này thực hiện cập nhật giá phòng của mã loại phòng lẻ thêm 10%. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 880000  2 1500000  3 2200000  4 1700000  5 3300000 | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| No action. | select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | Session 2 truy vấn để kiểm tra thông tin cập nhật. |
| No action. | update loaiphong  set giaphong = giaphong + 10000  where malp = 2;  1 row updated. | Câu lệnh này thực hiện cập nhật giá phòng của mã loại phòng là 2 thêm 10000 đồng. |
| No action. | select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1510000  3 2000000  4 1700000  5 3000000 | Session 2 truy vấn để kiểm tra thông tin cập nhật. |
| **Session 1** | **Session 2** | **Explanation** |
| update loaiphong  set giaphong = 2000000  where malp = 2;  Prompt does not return | No action. | Session 1 thực hiện câu lệnh cập nhật mã loại phòng là 2 lên 2000000.  Tuy nhiên, khóa của malp = 2 đang được giữ bởi session 2 nên session 1 sẽ chờ session 2 kết thúc transaction. |
| No action. | update loaiphong  set giaphong = giaphong + 10000  where malp = 1;  Prompt does not return | Session 2 tiến hành cập nhật giá phòng của malp = 1 thêm 10000 đồng.  Tuy nhiên, khóa của malp = 1 cũng đang được giữ bởi session 1 nên session 2 sẽ chờ session 2 kết thúc transaction |
| Lúc này, cả hai session đều chờ nhau và không có session nào được thực thi. => Deadlock. | | |
| Cách khắc khục thứ nhất: phát hiện  create or replace procedure gia\_phong(v\_lp loaiphong.malp%type,  v\_gia loaiphong.giaphong%type)  is  pragma autonomous\_transaction;  begin  update loaiphong  set giaphong = v\_gia  where malp = v\_lp;  commit;  end; | | |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| **Session 1** | **Session 2** | **Explanation** |
| update loaiphong  set giaphong = giaphong \*(1.1)  where malp in (1, 3, 5);  1 row updated. | No action. | Câu lệnh này thực hiện cập nhật giá phòng của mã loại phòng lẻ thêm 10%. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 880000  2 1500000  3 2200000  4 1700000  5 3300000 | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| No action. | select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | Session 2 truy vấn để kiểm tra thông tin cập nhật. |
| No action. | update loaiphong  set giaphong = giaphong + 10000  where malp = 2;  1 row updated. | Câu lệnh này thực hiện cập nhật giá phòng của mã loại phòng là 2 thêm 10000 đồng. |
| No action. | select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1510000  3 2000000  4 1700000  5 3000000 | Session 2 truy vấn để kiểm tra thông tin cập nhật. |
| begin  gia\_phong(2, 2000000);  end;  Prompt does not return | No action. | Trong hàm *gia\_phong* , session 1 thực hiện cập nhật mã loại phòng là 2 lên 2000000 và có thiết lập Autonomous\_transaction. |
| **Session 1** | **Session 2** | **Explanation** |
| Prompt does not return | update loaiphong  set giaphong = giaphong + 10000  where malp = 1;  Prompt does not return | Session 2 tiến hành cập nhật giá phòng của malp = 1 thêm 10000 đồng.  Kết quả của quá trình này là session 1 và session 2 bị deadlock. |
| ORA-00060: deadlock detected while waiting for resource. | Prompt does not return | Khi session 1 gọi hàm *gia\_phong* thì pragma autonomous\_transaction có tác dụng giới hạn phạm vi ‘COMMIT’ hoặc ‘ROLLBACK’ với các câu lệnh trong thủ tục gọi chính nó.  Khi session 1 có gắng truy cập tài nguyên do transaction chính khác chính là session 2 nắm giữa thì sẽ xảy ra tình huống deadlock.  Tuy nhiên, khi sử dụng autonomous\_transaction, Oracle đã đưa ra một ngoại lệ để xử lý chính là thực hiện câu lệnh ‘ROLLBACK’ trong trường hợp không xử lý được câu lệnh.  Trong trường hợp thủ tục thiết lập autonomous \_transaction mà không có câu lệnh ‘COMMIT’ hoặc ‘ROLLBACK’ thì Oracle cũng tự động thực hiện ‘ROLLBACK’.  Tuy session 1 đã được rollback bởi thủ tục gia\_phong nhưng transaction chính là session 1 vẫn chưa thực hiện câu lệnh ‘COMMIT’ hoặc ‘ROLLBACK’ nên session 2 vẫn tiếp tục chờ khóa của session 1. |
| **Session 1** | **Session 2** | **Explanation** |
| commit;  Commit complete. | Prompt does not return | Session 1 tiến hành ‘COMMIT’ và mở transaction locks. |
| No action. | 1 row updated. | Session 2 cập nhật thành công. |
| No action. | select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 890000  2 1510000  3 2200000  4 1700000  5 3300000 | Session 2 truy vấn để kiểm tra thông tin cập nhật. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 880000  2 1500000  3 2200000  4 1700000  5 3300000 | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| No action. | commit;  Commit complete. | Session 2 tiến hành ‘COMMIT’ và mở transaction locks |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 890000  2 1510000  3 2200000  4 1700000  5 3300000 | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| **Session 1** | **Session 2** | **Explanation** |
| Cách khắc khục thứ hai: ngăn ngừa | | |
| LOCK TABLE LOAIPHONG IN EXCLUSIVE MODE;  Lock succeeds. |  | Session 1 khóa dữ liệu đọc quyền của toàn bảng LOAIPHONG và lúc này các session khác chỉ có thể tra cứu chứ không thể ghi. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| update loaiphong  set giaphong = giaphong \*(1.1)  where malp in (1, 3, 5);  1 row updated. | No action. | Câu lệnh này thực hiện cập nhật giá phòng của mã loại phòng lẻ thêm 10%. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 880000  2 1500000  3 2200000  4 1700000  5 3300000 | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| No action. | select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 800000  2 1500000  3 2000000  4 1700000  5 3000000 | Session 2 truy vấn để kiểm tra thông tin cập nhật. |
| No action. | update loaiphong  set giaphong = giaphong + 10000  where malp = 2;  Prompt does not return | Câu lệnh này thực hiện cập nhật giá phòng của mã loại phòng là 2 thêm 10000 đồng.  Tuy nhiên, do session 1 đã thiết lập khóa độc quyền nên sesion 2 không thể thực hiện câu lệnh cập nhật và chờ session 1 kết thúc phiên làm việc. |
| **Session 1** | **Session 2** | **Explanation** |
| update loaiphong  set giaphong = giaphong \*(1.1)  where malp in 2;  1 row updated. | Prompt does not return | Khi thực hiện câu lệnh cập nhật, session 1 thực hiện cập nhật mã loại phòng là 2 lên 1650000. Và bởi vì Session 1 đang giữ khóa của các dòng dữ liệu của bảng LOAIPHONG nên sẽ thực hiện thành công. |
| commit;  Commit complete. | Prompt does not return | Session 1 tiến hành ‘COMMIT’ và mở transaction locks. |
| No action. | 1 row updated. | Session 2 cập nhật thành công. |
| No action. | select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 880000  2 1660000  3 2200000  4 1700000  5 3300000 | Session 2 truy vấn để kiểm tra thông tin cập nhật và nhận được thông tin chỉnh sửa từ sesion 1. |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 880000  2 1650000  3 2200000  4 1700000  5 3300000 | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật. |
| No action. | commit;  Commit complete. | Session 2 tiến hành ‘COMMIT’ và mở transaction locks |
| select malp, giaphong  from loaiphong;  MALP GIAPHONG  ------- --------  1 880000  2 1660000  3 2200000  4 1700000  5 3300000 | No action. | Session 1 truy vấn để kiểm tra thông tin cập nhật và nhận thấy thông tin phòng số 2 đã bị thay đổi. |

1. **Mô tả đồ án môn học:**
2. **Lost Update:**

* Tình trạng này xảy ra khi có nhiều hơn một giao tác cùng thực hiện cập nhật trên 1 đơn vị dữ liệu. Khi đó, tác dụng của giao tác cập nhật thực hiện sau sẽ đè lên tác dụng của thao tác cập nhật trước.
  1. Tình huống: Vào ngày 29/06/2021, khách hàng Velasquez Carmen (maKh = 3) có một chuyến công tác đột xuất ở Việt Nam, cô vừa xuống sân bay và muốn đặt một phòng khách sạn để nghỉ ngơi. Cô đến một trong những khách sạn gần nhất và yêu cầu đặt phòng từ ngày 29/06/2021 đến ngày 01/07/2021. Lúc này, khách sạn đã chọn một phòng ở tầng thượng còn trống đáp ứng được yêu cầu của cô và đã tiến hành lập thủ tục nhận phòng này cho cô. Cùng lúc đó, Mr. Audry nhận được cuộc họp gấp ở Việt Nam vào ngày 30/06/2021, nên anh đã đặt phòng trước tại khách sạn từ ngày 30/06/2021 đến 01/07/2021 và trùng với mã phòng của Mrs. Carmen.
  2. Mô tả:

**Bảng mô tả Lost Update:**

|  |  |  |
| --- | --- | --- |
| **Session 1** | **Session 2** | **Explanation** |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, '29/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 15/09/2024  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | No action. | Session 1 thực hiện thủ tục HIEN\_THI\_THEO\_MATTP với tham số đầu vào lần lượt là mã tình trạng phòng, ngày bắt đầu, ngày kết thúc, mã phiếu và mã tình trạng phiếu.  Với mã tình trạng phòng bằng 1, người dùng đang tìm kiếm phòng trống trong khoảng thời gian từ ngày 29/06/2021 đến hết ngày 01/07/2021. |
| BEGIN  THEM\_PTP(3, 3);  END;  PL/SQL procedure successfully completed. | No action. | Session 1 tiến hành thủ tục nhận phòng cho khách hàng có hai tham số truyền vào lần lượt là mã khách hàng đăng ký nhận phòng và mã nhân viên của nhân viên đang tiến hành thủ tục cho khách hàng đó. |
| **Session 1** | **Session 2** | **Explanation** |
| SELECT MAPTP  FROM PHIEUTHUEPHONG  WHERE MAPTP = LAY\_MAPTP(3,3);  MAPTP  -----  144 | No action. | Session 1 chạy câu lệnh truy vấn vào cơ sở dữ liệu PHIEUTHUEPHONG để kiểm tra thao tác lập phiếu thuê phòng thành công. |
| BEGIN  THEM\_CTPTP (  LAY\_MAPTP (3,3), 19, '29/06/2021',  '01/07/2021', 2);  END;  PL/SQL procedure successfully completed. | No action. | Session 1 bắt đầu giao tác thêm vào cơ sở dữ liệu ChiTietPTP thông tin phòng được thuê với tham số lần lượt là mã phiếu thuê phòng, số phòng, ngày nhận phòng, ngày trả phòng thực tế và số lượng khách trong phòng. |
| SELECT MAPTP, MAPHONG, NGAYNP, NGAYTPTT  FROM CHITIETPTP  WHERE MAPTP = LAY\_MAPTP(3, 3);  MAPTP MAPHONG NGAYNP NGAYTPTT  ----- ------- -------- -------  144 19 29/06/2021 01/07/2021 | No action. | Session 1 chạy câu lệnh truy vấn để kiểm tra thêm chi tiết phiếu thuê phòng thành công. |
| No action. | SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  Transaction ISOLATION succeeded | Session 2 bắt đầu thực hiện giao tác và thiết lập mức cô lập mặc định cho Session 2. |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, '29/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | BEGIN  HIEN\_THI\_THEO\_MATTP (1, '30/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 15/09/2024  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | Session 1 và Sesstion 2 cùng thực hiện thủ tục HIEN\_THI\_THEO\_MATT.    Trong đó, session 2 chưa nhận được thông tin được cập nhật bởi session 1. Vì Oracle sử dụng mức cô lập mặc định là *‘SET TRANSACTION ISOLATION LEVEL READ COMMITTED’* nên khi session 1 chưa thực hiện *‘COMMIT’* hoặc *‘ROLLBACK’* vẫn chưa thay đổi cơ sở dữ liệu. |
| **Session 1** | **Session 2** | **Explanation** |
| No action. | BEGIN  THEM\_PDT(4,4);  END;  PL/SQL procedure successfully completed. | Sau khi truy vấn cơ sở dữ liệu và thấy tình trạng phòng số 19 còn trống nên Session 2 tiến hành lập phiếu đặt phòng. |
| No action. | SELECT MAPDT  FROM PHIEUDATTRUOC  WHERE MAPDT = LAY\_MAPDT(4,4);  MAPDT  -----  81 | Session 2 chạy câu lệnh truy vấn để kiểm tra thêm phiếu đặt phòng thành công. |
| No action. | BEGIN  THEM\_CTPDT(LAY\_MAPDT(4,4), 19, '30/06/2021', '01/07/2021');  END;  Prompt does not return | Sau đó, session 2 tiến hành giao tác thêm thông tin vào ChiTietPDT.  Tuy nhiên, session 1 đang giữ khóa dòng dữ liệu này và dẫn đến xung đột trong quá trình ghi dữ liệu. |
| SELECT \*  FROM CHITIETTTP  WHERE MAPHONG = 19  ORDER BY NGAYBD;  MAPHONG MATTP NGAYBD NGAYKT  ------- ----- ------ ------  19 1 01/01/2021 08/01/2021  .................................  19 3 29/06/2021 01/07/2021  ................................. | Prompt does not return | Session 1 chạy câu lệnh truy vấn để kiểm tra chi tiết tình trạng phòng đã thay đổi thành công. |
| COMMIT;  Commit completed. | Prompt does not return | Sau một khoảng thời gian làm việc, Session 1 thực hiện xong và kết thúc công việc của mình bằng câu lệnh *‘COMMIT’.* |
| No action. | PL/SQL procedure successfully completed. | Session 1 giải phóng khóa trên dòng dữ liệu có maphong = 19 và thực thi câu lệnh trong Session 2. |
| No action. | SELECT MAPDT, MAPHONG, NGAYNP, NGAYTPDD  FROM CHITIETPDT  WHERE MAPDT = LAY\_MAPDT(4, 4); | Session 2 chạy câu lệnh truy vấn để kiểm tra thêm chi tiết phiếu đặt phòng thành công. |
| **Session 1** | **Session 2** | **Explanation** |
|  | MAPDT MAPHONG NGAYNP NGAYTPTT  ----- ------- ------ --------  81 19 30/06/2021 01/07/2021 |  |
| No action. | BEGIN  HIEN\_THI\_THEO\_MATTP (1, '29/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  30/06/2021 01/07/2024  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | Session 2 chạy thủ thục HIEN\_THI\_THEO\_MATTP để kiểm tra tại khoảng thời gian này, phòng 19 đã không còn trống. |
| No action. | SELECT \*  FROM CHITIETTTP  WHERE MAPHONG = 19  ORDER BY NGAYBD;  MAPHONG MATTP NGAYBD NGAYKT  ------- ----- ------ ------  19 1 01/01/2021 08/01/2021  .................................  19 2 30/06/2021 01/07/2021  ................................. | Session 2 chạy câu lệnh truy vấn để kiểm tra chi tiết tình trạng phòng đã thay đổi thành công. |
| No action. | COMMIT;  Commit completed. | Session 2 hoàn thành công việc và kết thúc bằng lệnh *‘COMMIT’.* |
| BEGIN  HIEN\_THI\_THEO\_MATTP(3, NULL, NULL, LAY\_MAPTP(3,3), 1);  END;  MAPTP TIENPHONG MAKH  ----- -------- --------  144 0 3  MAPHONG NGAYNP NGAYTPTT  ------ ------- -------- | BEGIN  HIEN\_THI\_THEO\_MATTP(2, NULL, NULL, LAY\_MAPDT(4,4), 1);  END;  MAPDT NGAYDT MAKH  ----- -------- --------  81 29/06/2021 4  MAPHONG NGAYNP NGAYTPDD  ------ ------- -------- | Session 1 và Session 2 đồng thời kiểm tra thông tin vừa đăng ký của mình bằng thủ tục HIEN\_THI\_THEO\_MATT. |
| **Session 1** | **Session 2** | **Explanation** |
| 19 29/06/2021 01/07/2021  SLNGUOI PHUTHU  ------ ------  2 0.15 | 19 30/06/2021 01/07/2021 |  |
| SELECT \*  FROM CHITIETTTP  WHERE MAPHONG = 19  ORDER BY NGAYBD;  MAPHONG MATTP NGAYBD NGAYKT  ------- ----- ------ ------  19 1 01/01/2021 08/01/2021  .................................  19 2 30/06/2021 01/07/2021  ................................. | SELECT \*  FROM CHITIETTTP  WHERE MAPHONG = 19  ORDER BY NGAYBD;  MAPHONG MATTP NGAYBD NGAYKT  ------- ----- ------ ------  19 1 01/01/2021 08/01/2021  .................................  19 2 30/06/2021 01/07/2021  ................................. | Session 1 và Session 2 tiến hành truy vấn cơ sở dữ liệu ChiTietTTP và thấy tình trạng phòng, ngày bắt đầu và ngày kết thúc tình trạng đã giống nhau.  Tuy nhiên, thông tin đã xảy ra sai sót. Mã tình trạng phòng 19 là 2 từ ngày 30/06/2021 đến ngày 01/07/2021 được cập nhật bởi Session 2. Và thông tin thuê phòng số 19 từ ngày 29/06/2021 đến ngày 30/07/2021 của Session 1 đã bị mất. |

**→ Lost Update đã xảy ra**

Nguyên nhân và giải pháp:

* Vấn đề xảy ra: Khi session 2 thực thi đã ghi đè dữ liệu trong session 1.
* Nguyên nhân: Session 2 cập nhật dữ liệu trước khi session 1 tiến hành ***‘COMMIT’* hoặc *‘ROLLBACK’***vì vậy, dữ liệu trên hiển thị không phù hợp.
* Giải pháp: Sử dụng câu lệnh ***‘LOCK TABLE CHITIETTTP IN ROW EXCLUSIVE MODE’*** trong stored procedure **SUA\_MATTP\_CTTTP** hoặc chuyển đổi mức cô lập mặc định READ COMMITTED thành SERIALIZABLE.
* Xử lý trường hợp Lost Update bằng Serializable:

|  |  |  |
| --- | --- | --- |
| **Session 1** | **Session 2** | **Explanation** |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, '29/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 15/09/2024  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | No action. | Session 1 thực hiện thủ tục HIEN\_THI\_THEO\_MATTP với tham số đầu vào lần lượt là mã tình trạng phòng, ngày bắt đầu, ngày kết thúc, mã phiếu và mã tình trạng phiếu.  Với mã tình trạng phòng bằng 1, người dùng đang tìm kiếm phòng trống trong khoảng thời gian từ ngày 29/06/2021 đến hết ngày 01/07/2021. |
| BEGIN  THEM\_PTP(3, 3);  END;  PL/SQL procedure successfully completed. | No action. | Session 1 tiến hành thủ tục nhận phòng cho khách hàng có hai tham số truyền vào lần lượt là mã khách hàng đăng ký nhận phòng và mã nhân viên của nhân viên đang tiến hành thủ tục cho khách hàng đó. |
| SELECT MAPTP  FROM PHIEUTHUEPHONG  WHERE MAPTP = LAY\_MAPTP(3,3);  MAPTP  -----  144 | No action. | Session 1 chạy câu lệnh truy vấn vào cơ sở dữ liệu PHIEUTHUEPHONG để kiểm tra thao tác lập phiếu thuê phòng thành công. |
| BEGIN  THEM\_CTPTP (  LAY\_MAPTP (3,3), 19, '29/06/2021',  '01/07/2021', 2);  END;  PL/SQL procedure successfully completed. | No action. | Session 1 bắt đầu giao tác thêm vào cơ sở dữ liệu ChiTietPTP thông tin phòng được thuê với tham số lần lượt là mã phiếu thuê phòng, số phòng, ngày nhập phòng, ngày trả phòng thực tế và số lượng khách trong phòng. |
| SELECT MAPTP, MAPHONG, NGAYNP, NGAYTPTT  FROM CHITIETPTP  WHERE MAPTP = LAY\_MAPTP(3, 3);  MAPTP MAPHONG NGAYNP NGAYTPTT  ----- ------- -------- -------  144 19 29/06/2021 01/07/2021 | No action. | Session 1 chạy câu lệnh truy vấn để kiểm tra thêm chi tiết phiếu thuê phòng thành công. |
| **Session 1** | **Session 2** | **Explanation** |
| No action. | SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  Transaction ISOLATION succeeded | Session 2 bắt đầu thực hiện giao tác và thiết lập mức cô lập Serializable cho Session 2. |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, '29/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | BEGIN  HIEN\_THI\_THEO\_MATTP (1, '30/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 15/09/2024  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | Session 1 và Sesstion 2 cùng thực hiện thủ tục HIEN\_THI\_THEO\_MATTP.  Trong đó, session 2 chưa nhận được thông tin được cập nhật bởi session 1. Vì Oracle sử dụng mức cô lập mặc định là *‘SET TRANSACTION ISOLATION LEVEL READ COMMITTED’* nên khi session 1 chưa thực hiện *‘COMMIT’* hoặc *‘ROLLBACK’* vẫn chưa thay đổi cơ sở dữ liệu. |
| No action. | BEGIN  THEM\_PDT(4,4);  END;  PL/SQL procedure successfully completed. | Sau khi truy vấn cơ sở dữ liệu và thấy tình trạng phòng số 19 còn trống nên Session 2 tiến hành lập phiếu đặt phòng. |
| No action. | SELECT MAPDT  FROM PHIEUDATTRUOC  WHERE MAPDT = LAY\_MAPDT(4,4);  MAPDT  -----  81 | Session 2 chạy câu lệnh truy vấn để kiểm tra thêm phiếu đặt phòng thành công. |
| No action. | BEGIN  THEM\_CTPDT(LAY\_MAPDT(4,4), 19, '30/06/2021', '01/07/2021');  END;  Prompt does not return | Sau đó, session 2 tiến hành giao tác thêm thông tin vào ChiTietPDT.  Tuy nhiên, session 1 đang giữ khóa dòng dữ liệu này và dẫn đến xung đột trong quá trình ghi dữ liệu. Session 2 chờ Session 1 thực hiện công việc của mình. |
| **Session 1** | **Session 2** | **Explanation** |
| SELECT \*  FROM CHITIETTTP  WHERE MAPHONG = 19  ORDER BY NGAYBD;  MAPHONG MATTP NGAYBD NGAYKT  ------- ----- ------ ------  19 1 01/01/2021 08/01/2021  .................................  19 3 29/06/2021 01/07/2021  ................................. | Prompt does not return | Session 1 chạy câu lệnh truy vấn để kiểm tra chi tiết tình trạng phòng đã thay đổi thành công. |
| COMMIT;  Commit completed. | Prompt does not return | Sau một khoảng thời gian làm việc, Session 1 thực hiện xong và kết thúc công việc của mình bằng câu lệnh *‘COMMIT’.* |
| No action. | ORA-08177: can't serialize access for this transaction. | Câu lệnh ‘COMMIT’ đã kết thúc phiên làm việc của Session 1 và thay đổi dữ liệu trong ChiTietTTP nên Session 2 không thực hiện thành công với lỗi ‘ORA-08177’.  Nguyên nhân vì Session 1 đã cập nhật chi tiết phòng 19 sau khi Session 2 bắt đầu. |
| No action. | COMMIT;  Commit completed. | Session 2 tiếp tục thực hiện và kết thúc công việc của mình bằng cau lệnh ‘COMMIT’. |
| No action. | SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  Transaction ISOLATION succeeded | Session 2 bắt đầu thực hiện giao tác và thiết lập mức cô lập Serializable cho Session 2. |
| No action. | BEGIN  HIEN\_THI\_THEO\_MATTP (1, '30/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  17 ST917 1700000  NGAYBD NGAYKT  ------ -------  04/06/2021 18/08/2021  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | Session 2 thực hiện lại thủ tục HIEN\_THI\_THEO\_MATTP để tìm kiếm phòng trống. Thì lúc này không tìm thấy thông tin phòng 19 nữa.  Nguyên nhân vì: Session 1 kết thúc công việc bằng câu lệnh*‘COMMIT’* sau khi Session 2 bắt đầu nên trước đó Session 2 không nhận được thông tin thay đổi của Session 1. |
| **Session 1** | **Session 2** | **Explanation** |
| No action. | BEGIN  THEM\_CTPDT(LAY\_MAPDT(4,4), 17, '30/06/2021', '01/07/2021');  END;  PL/SQL procedure successfully completed. | Bởi vì phòng số 19 đã được thuê bởi khách hàng có makh là 3 trước khi Session 2 bắt đầu thực hiện đặt phòng nên nhân viên nhận thấy phòng không còn trống và thông báo với khách hàng. Sau quá trình trao đổi, khách hàng đã quyết định đặt phòng số 17 với mức giá cao hơn. |
| No action. | SELECT MAPDT, MAPHONG, NGAYNP, NGAYTPDD  FROM CHITIETPDT  WHERE MAPDT = LAY\_MAPDT(4, 4);  MAPDT MAPHONG NGAYNP NGAYTPTT  ----- ------- ------ --------  81 17 30/06/2021 01/07/2021 | Session 2 thực hiện câu lệnh truy vấn dể kiểm tra thêm thông tin chi tiết phiếu đặt trước thành công. |
| No action. | SELECT \*  FROM CHITIETTTP  WHERE MAPHONG = 17  ORDER BY NGAYBD;  MAPHONG MATTP NGAYBD NGAYKT  ------- ----- ------ ------  17 1 01/01/2021 28/01/2021  .................................  17 2 30/06/2021 01/07/2021  ................................. | Session 2 chạy câu lệnh truy vấn để kiểm tra chi tiết tình trạng phòng đã thay đổi thành công. |
| No action. | COMMIT;  Commit completed. | Session 2 hoàn thành công việc và kết thúc bằng lệnh *‘COMMIT’* mà không có bất kỳ vấn đề về khóa nữa. |
| BEGIN  HIEN\_THI\_THEO\_MATTP(3, NULL, NULL, LAY\_MAPTP(3,3), 1);  END;  MAPTP TIENPHONG MAKH  ----- -------- --------  144 0 3  MAPHONG NGAYNP NGAYTPTT  ------ ------- --------  19 29/06/2021 01/07/2021  SLNGUOI PHUTHU  ------ ------  2 0.15 | BEGIN  HIEN\_THI\_THEO\_MATTP(2, NULL, NULL, LAY\_MAPDT(4,4), 1);  END;  MAPDT NGAYDT MAKH  ----- -------- --------  81 29/06/2021 4  MAPHONG NGAYNP NGAYTPDD  ------ ------- --------  17 30/06/2021 01/07/2021 | Session 1 và Session 2 đồng thời kiểm tra thông tin vừa đăng ký của mình bằng thủ tục HIEN\_THI\_THEO\_MATTP.  Giao dịch kết thúc. |

* Xử lý Lost Update bằng Lock:

|  |  |  |
| --- | --- | --- |
| **Session 1** | **Session 2** | **Explanation** |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, '29/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 15/09/2024  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | No action. | Session 1 thực hiện thủ tục HIEN\_THI\_THEO\_MATTP với tham số đầu vào lần lượt là mã tình trạng phòng, ngày bắt đầu, ngày kết thúc, mã phiếu và mã tình trạng phiếu.  Với mã tình trạng phòng bằng 1, người dùng đang tìm kiếm phòng trống trong khoảng thời gian từ ngày 29/06/2021 đến hết ngày 01/07/2021. |
| BEGIN  THEM\_PTP(3, 3);  END;  PL/SQL procedure successfully completed. | No action. | Session 1 tiến hành thủ tục nhận phòng cho khách hàng có hai tham số truyền vào lần lượt là mã khách hàng đăng ký nhận phòng và mã nhân viên của nhân viên đang tiến hành thủ tục cho khách hàng đó. |
| SELECT MAPTP  FROM PHIEUTHUEPHONG  WHERE MAPTP = LAY\_MAPTP(3,3);  MAPTP  -----  144 | No action. | Session 1 chạy câu lệnh truy vấn vào cơ sở dữ liệu PHIEUTHUEPHONG để kiểm tra thao tác lập phiếu thuê phòng thành công. |
| BEGIN  THEM\_CTPTP (  LAY\_MAPTP (3,3), 19, '29/06/2021',  '01/07/2021', 2);  END;  PL/SQL procedure successfully completed. | No action. | Session 1 bắt đầu giao tác thêm vào cơ sở dữ liệu ChiTietPTP thông tin phòng được thuê với tham số lần lượt là mã phiếu thuê phòng, số phòng, ngày nhập phòng, ngày trả phòng thực tế và số lượng khách trong phòng. |
| **Session 1** | **Session 2** | **Explanation** |
| SELECT MAPTP, MAPHONG, NGAYNP, NGAYTPTT  FROM CHITIETPTP  WHERE MAPTP = LAY\_MAPTP(3, 3);  MAPTP MAPHONG NGAYNP NGAYTPTT  ----- ------- -------- -------  144 19 29/06/2021 01/07/2021 | No action. | Session 1 chạy câu lệnh truy vấn để kiểm tra thêm chi tiết phiếu thuê phòng thành công. |
| No action. | SET TRANSACTION ISOLATION LEVEL READ COMMITTED; | Session 2 bắt đầu thực hiện giao tác và thiết lập mức cô lập mặc định cho Session 2. |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, '29/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | BEGIN  HIEN\_THI\_THEO\_MATTP (1, '30/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 15/09/2024  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | Session 1 và Sesstion 2 cùng thực hiện thủ tục HIEN\_THI\_THEO\_MATTP.  Trong đó, session 2 chưa nhận được thông tin được cập nhật bởi session 1. Vì Oracle sử dụng mức cô lập mặc định là *‘SET TRANSACTION ISOLATION LEVEL READ COMMITTED’* nên khi session 1 chưa thực hiện *‘COMMIT’* hoặc *‘ROLLBACK’* vẫn chưa thay đổi cơ sở dữ liệu. |
| No action. | BEGIN  THEM\_PDT(4,4);  END;  PL/SQL procedure successfully completed. | Sau khi truy vấn cơ sở dữ liệu và thấy tình trạng phòng số 19 còn trống nên Session 2 tiến hành lập phiếu đặt phòng. |
| No action. | SELECT MAPDT  FROM PHIEUDATTRUOC  WHERE MAPDT = LAY\_MAPDT(4,4);  MAPDT  -----  81 | Session 2 chạy câu lệnh truy vấn để kiểm tra thêm phiếu đặt phòng thành công. |
| **Session 1** | **Session 2** | **Explanation** |
| No action. | BEGIN  THEM\_CTPDT(LAY\_MAPDT(4,4), 19, '30/06/2021', '01/07/2021');  END;  Prompt does not return | Sau đó, session 2 tiến hành giao tác thêm thông tin vào ChiTietPDT.  Tuy nhiên, session 1 đang giữ khóa dòng dữ liệu này và dẫn đến xung đột trong quá trình ghi dữ liệu. |
| SELECT \*  FROM CHITIETTTP  WHERE MAPHONG = 19  ORDER BY NGAYBD;  MAPHONG MATTP NGAYBD NGAYKT  ------- ----- ------ ------  19 1 01/01/2021 08/01/2021  .................................  19 3 29/06/2021 01/07/2021  ................................. | Prompt does not return | Session 1 chạy câu lệnh truy vấn để kiểm tra chi tiết tình trạng phòng đã thay đổi thành công. |
| COMMIT;  Commit completed. | Prompt does not return | Sau một khoảng thời gian làm việc, Session 1 thực hiện xong và kết thúc công việc của mình bằng câu lệnh *‘COMMIT’.* |
| No action. | Không có phòng phù hợp.  PL/SQL procedure successfully completed. | Session 1 giải phóng khóa trên dòng dữ liệu có maphong = 19 và thực thi câu lệnh trong Session 2. |
| No action. | BEGIN  HIEN\_THI\_THEO\_MATTP (1, '30/06/2021’, '01/07/2021', NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  29/06/2021 26/06/2024  17 ST917 1700000  NGAYBD NGAYKT  ------ -------  04/06/2021 18/08/2021  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  20 SD101 800000  NGAYBD NGAYKT  ------ -------  04/06/2021 12/10/2024 | Session 2 kiểm tra lại thì không tìm thấy thông tin phòng 19 nữa.    Nguyên nhân vì: Session 1 kết thúc công việc bằng câu lệnh*‘COMMIT’* sau khi Session 2 bắt đầu nên trước đó Session 2 không nhận được thông tin thay đổi của Session 1. |
| **Session 1** | **Session 2** | **Explanation** |
| No action. | BEGIN  THEM\_CTPDT(LAY\_MAPDT(4,4), 17, '30/06/2021', '01/07/2021');  END;  PL/SQL procedure successfully completed. | Bởi vì phòng số 19 đã được thuê bởi khách hàng có makh là 3 trước khi Session 2 bắt đầu thực hiện đặt phòng nên nhân viên nhận thấy phòng không còn trống và thông báo với khách hàng. Sau quá trình trao đổi, khách hàng đã quyết định đặt phòng số 17 với mức giá cao hơn. |
| No action. | SELECT \*  FROM CHITIETTTP  WHERE MAPHONG = 17  ORDER BY NGAYBD;  MAPHONG MATTP NGAYBD NGAYKT  ------- ----- ------ ------  17 1 01/01/2021 28/01/2021  .................................  17 2 30/06/2021 01/07/2021  ................................. | Session 2 chạy câu lệnh truy vấn để kiểm tra chi tiết tình trạng phòng đã thay đổi thành công. |
| No action. | COMMIT;  Commit completed. | Session 2 hoàn thành công việc và kết thúc bằng lệnh *‘COMMIT’* mà không có bất kỳ vấn đề về khóa nữa. |
| BEGIN  HIEN\_THI\_THEO\_MATTP(3, NULL, NULL, LAY\_MAPTP(3,3), 1);  END;  MAPTP TIENPHONG MAKH  ----- -------- --------  144 0 3  MAPHONG NGAYNP NGAYTPTT  ------ ------- --------  19 29/06/2021 01/07/2021  SLNGUOI PHUTHU  ------ ------  2 0.15 | BEGIN  HIEN\_THI\_THEO\_MATTP(2, NULL, NULL, LAY\_MAPDT(4,4), 1);  END;  MAPDT NGAYDT MAKH  ----- -------- --------  81 29/06/2021 4  MAPHONG NGAYNP NGAYTPDD  ------ ------- --------  17 30/06/2021 01/07/2021 | Session 1 và Session 2 đồng thời kiểm tra thông tin vừa đăng ký của mình bằng thủ tục HIEN\_THI\_THEO\_MATTP.  Giao dịch kết thúc. |

* Code:

Text

Description automatically generated with medium confidence

*Hình 6: Cập nhật giá trị cccd trong bảng khachhang lần 1.*

Text

Description automatically generated with medium confidence

*Hình 7: Cập nhật giá trị cccd trong bảng khachhang lần 2.*

Graphical user interface, text, application

Description automatically generated

*Hình 8: Giải quyết lost update bằng mức cô lập Serializable.*

1. **Non – repeatable Read:**

* Tình trạng này xảy ra khi một giao tác T1 vừa thực hiện xong thao tác đọc trên một đơn vị dữ liệu (nhưng chưa commit) thì giao tác khác (T2) lại thay đổi (ghi) trên đơn vị dữ liệu này. Điều này làm cho lần đọc sau đó của T1 không còn nhìn thấy dữ liệu ban đầu nữa.

1. Tình huống: Khách hàng Nguyễn Ngọc Thiện (makh = 18) muốn đặt vé DamSen’s Park tại Lucifer’s Hotel vì giá vé rẻ hơn bên ngoài 10%. Khách hàng gọi điện cho nhân viên và kiểm tra vé thì thấy giá vé lúc này là 320.000, Thiện quyết định đặt vé tại đây. Thì ngay lúc này, khách sạn nhận được cuộc gọi bởi đơn vị DamSen’sPark và tăng mức giá của vé thêm 5%. Khi xuất hóa đơn cho khách hàng, thì nhân viên và Thiện đã nhận thấy sự thay đổi trong giá vé.
2. Mô tả:

|  |  |  |
| --- | --- | --- |
| **Session 1** | **Session 2** | **Explanation** |
| SELECT \*  FROM DICHVU  WHERE MADONVI = 5;  MADV TENDV GIADV MADONVI  ---- -------- ----- -------  23 Đặt vé khu 320000 5  vui chơi | No action. | Session 1 thực hiện câu lệnh truy vấn tìm kiếm giá dịch vụ tại đơn vị DamSen’Park. Thì thấy giá chương trình “Đặt vé khu vui chơi” có giá là 320000. |
|  | UPDATE DICHVU  SET GIADV = GIADV \*1.05  WHERE MADONVI = 5;  1 row updated. | Session 2 sử dụng mức cô lập mặc định trong Oracle là ‘SET TRANSACTION ISOLATION LEVEL READ COMMITTED’ và thực hiện lệnh cập nhật theo yêu cầu của đơn vị là tăng giá thêm 5%. |
|  | COMMIT;  Commit completed. | Session 2 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh ‘COMMIT’. |
| BEGIN  THEM\_PDV(18, 5);  END;  PL/SQL procedure successfully completed. |  | Session 1 sau khi tra cứu giá vé đã tiến hành lập phiếu dịch vụ cho khách hàng có mã khách hàng là 18 bởi nhân viên có mã là 5. |
| **Session 1** | **Session 2** | **Explanation** |
| SELECT \*  FROM PHIEUDICHVU  WHERE MAPDV = LAY\_MAPDV(18,5);  MAPDV TIENDV MAKH MANV MATT  ---- ------ ----- ---- ----  33 0 18 5 1 |  | Session 1 kiểm tra lập phiếu thành công. |
| BEGIN  THEM\_CTPDV(LAY\_MAPDV(18, 5), 23, 2);  END;  PL/SQL procedure successfully completed. |  | Session 1 tiếp tục thực hiện thao tác thêm vào chi tiết phiếu dịch vụ có mã là mã phiếu của khách hàng vừa đăng ký, mã dịch vụ là 23 và số lượng dịch vụ là 2. |
| SELECT \*  FROM PHIEUDICHVU  WHERE MAPDV = LAY\_MAPDV(18,5);  MAPDV MADV SLDV NGAYSD  ---- ---- ----- ------  33 23 2 30/06/2021 |  | Session 1 kiểm tra thêm chi tiết phiếu dịch vụ thành công. |
| BEGIN  THEM\_HOADON(18, 5);  END;  PL/SQL procedure successfully completed. |  | Session 1 tiến hành lập hóa đơn cho khách hàng với mã khách hàng là 18 và mã nhân viên là 5; |
| SELECT \*  FROM HOADON  WHERE MAPDV = LAY\_MAPDV(18,5);  MAHD TONGTIEN NGAYTT MANV MAKH MAPT MAPDV  ---- -------- ------ ---- ---- ---- -----  33 739200 30/06/2021 5 18 NULL 33 |  | Session 1 thực hiện câu lệnh truy vấn thông tin hóa đơn của mã phiếu dịch vụ khách hàng Thiện đang sử dụng. |
| BEGIN  XUAT\_HOADON(76);  END;  PL/SQL procedure successfully completed.  Mã hóa đơn: 76 Tên nhân viên: Phạm Thu Huyền Tên khách hàng: Nguyễn Ngọc Thiện  ===> Phiếu dịch vụ: 33  Tên dịch vụ: Đặt vé khu vui chơi Gía dịch vụ: 336000 Ngày sử dụng: 30/06/2021 Số lượng dịch vụ: 2 Tên đơn vị: DamSen’sPark Tiền dịch vụ: 672000  Tiền dịch vụ thực tế: 672000 Tiền dịch vụ sau khi sử dụng khuyến mãi: 672000  ===> Tổng tiền: 672000 Tổng tiền sau khi tính thuế: 739200 |  | Session 1 tiến hành xuất hóa đơn với mã hóa đơn là 76. Và nhận thấy tiền dịch vụ của chương trình “Đặt vé khu vui chơi” đã tăng từ 320000 lên 336000 so với ban đầu dẫn đến chi phí của khách hàng Thiện phải trả cao hơn so với dự tính. |

**→ Unrepeatable Read đã xảy ra**

1. Nguyên nhân và giải pháp:

* Vấn đề: Session 1 thực hiện truy vấn thông tin dịch vụ, ngay lúc này session 2 cập nhật giá dịch vụ tương ứng, sau đó session 1 thực hiện truy vấn lại thì phát hiện dữ liệu đã bị thay đổi.
* Nguyên nhân: vì mức cô lập của Session 1 là **READ COMMITTED** nên mỗi lần truy vấn trên cùng một đơn vị dữ liệu sẽ đọc lại từ cơ sở dữ liệu (cơ sở dữ liệu lúc này có thể đã bị thay đổi bởi session khác) mặc dù những câu lệnh này đọc trên đơn vị dữ liệu giống nhau.
* Giải pháp: đổi mức cô lập ở Session 1 thành **SERIALIZABLE** thay vì **READ COMMITTED** hoặc sử dụng câu lệnh ***‘LOCK TABLE CHITIETTTP IN SHARE ROW EXCLUSIVE MODE’*** trước khi thực thi.
* Xử lý Non – repeatable Read bằng Serializable:

|  |  |  |
| --- | --- | --- |
| **Session 1** | **Session 2** | **Explanation** |
| SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  Transaction ISOLATION succeeded | No action. | Session 1 thiết lập lức cô lập Serializable. |
| SELECT \*  FROM DICHVU  WHERE MADONVI = 5;  MADV TENDV GIADV MADONVI  ---- -------- ----- -------  23 Đặt vé khu 320000 5  vui chơi | No action. | Session 1 thực hiện câu lệnh truy vấn tìm kiếm giá dịch vụ tại đơn vị DamSen’Park. Thì thấy giá chương trình “Đặt vé khu vui chơi” có giá là 320000. |
|  | UPDATE DICHVU  SET GIADV = GIADV \*1.05  WHERE MADONVI = 5;  1 row updated. | Session 2 sử dụng mức cô lập mặc định trong Oracle là ‘SET TRANSACTION ISOLATION LEVEL READ COMMITTED’ và thực hiện lệnh cập nhật theo yêu cầu của đơn vị là tăng giá thêm 5%. |
|  | COMMIT;  Commit completed. | Session 2 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh ‘COMMIT’. |
| **Session 1** | **Session 2** | **Explanation** |
| BEGIN  THEM\_PDV(18, 5);  END;  PL/SQL procedure successfully completed. |  | Session 1 sau khi tra cứu giá vé đã tiến hành lập phiếu dịch vụ cho khách hàng có mã khách hàng là 18 bởi nhân viên có mã là 5. |
| SELECT \*  FROM PHIEUDICHVU  WHERE MAPDV = LAY\_MAPDV(18,5);  MAPDV TIENDV MAKH MANV MATT  ---- ------ ----- ---- ----  33 0 18 5 1 |  | Session 1 kiểm tra lập phiếu thành công. |
| BEGIN  THEM\_CTPDV(LAY\_MAPDV(18, 5), 23, 2);  END;  PL/SQL procedure successfully completed. |  | Session 1 tiếp tục thực hiện thao tác thêm vào chi tiết phiếu dịch vụ có mã là mã phiếu của khàng vừa đăng ký, mã dịch vụ là 23 và số lượng dịch vụ là 2. |
| SELECT \*  FROM CHITIETPDV  WHERE MAPDV = LAY\_MAPDV(18,5);  MAPDV MADV SLDV NGAYSD  ---- ---- ----- ------  33 23 2 30/06/2021 |  | Session 1 kiểm tra thêm chi tiết phiếu dịch vụ thành công. |
| BEGIN  THEM\_HOADON(18, 5);  END;  PL/SQL procedure successfully completed. |  | Session 1 tiến hành lập hóa đơn cho khách hàng với mã khách hàng là 18 và mã nhân viên là 5; |
| SELECT \*  FROM HOADON  WHERE MAPDV = 33;  MAHD TONGTIEN NGAYTT MANV MAKH MAPT MAPDV  ---- -------- ------ ---- ---- ---- -----  33 704000 30/06/2021 5 18 NULL 33 |  | Session 1 thực hiện câu lệnh truy vấn thông tin hóa đơn của mã phiếu dịch vụ khách hàng Thiện đang sử dụng. |
| **Session 1** | **Session 2** | **Explanation** |
| BEGIN  XUAT\_HOADON(76);  END;  PL/SQL procedure successfully completed.  Mã hóa đơn: 76 Tên nhân viên: Phạm Thu Huyền Tên khách hàng: Nguyễn Ngọc Thiện  ===> Phiếu dịch vụ: 33  Tên dịch vụ: Đặt vé khu vui chơi Giá dịch vụ: 320000 Ngày sử dụng: 30/06/2021 Số lượng dịch vụ: 2 Tên đơn vị: DamSen’sPark Tiền dịch vụ: 640000  Tiền dịch vụ thực tế: 640000 Tiền dịch vụ sau khi sử dụng khuyến mãi: 6400000  ===> Tổng tiền: 640000 Tổng tiền sau khi tính thuế: 704000 |  | Session 1 tiến hành xuất hóa đơn với mã hóa đơn là 76. Và nhận thấy tiền dịch vụ của chương trình “Đặt vé khu vui chơi” không đổi, khách hàng Thiện đăng ký dịch vụ và thanh toán thành công. |
| COMMIT;  Commit completed. |  | Session 1 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh ‘COMMIT’. |

* Xử lý Non – repeatable Read bằng Locks:

|  |  |  |
| --- | --- | --- |
| **Session 1** | **Session 2** | **Explanation** |
| LOCK TABLE DICHVU IN SHARE ROW EXCLUSIVE MODE;  Transaction ISOLATION succeeded | No action. | Session 1 thiết lập khóa trong bảng DICHVU trên dòng nó đang tương tác. |
| SELECT \*  FROM DICHVU  WHERE MADONVI = 5;  MADV TENDV GIADV MADONVI  ---- -------- ----- -------  23 Đặt vé khu 320000 5  vui chơi | No action. | Session 1 thực hiện câu lệnh truy vấn tìm kiếm giá dịch vụ tại đơn vị DamSen’Park. Thì thấy giá chương trình “Đặt vé khu vui chơi” có giá là 320000. |
|  | UPDATE DICHVU  SET GIADV = GIADV \*1.05  WHERE MADONVI = 5;  Prompt does not return. | Session 2 sử dụng mức cô lập mặc định trong Oracle là ‘SET TRANSACTION ISOLATION LEVEL READ COMMITTED’ và thực hiện lệnh cập nhật theo yêu cầu của đơn vị là tăng giá thêm 5%.  Tuy nhiên, session 1 đang giữ khóa dòng dữ liệu này và dẫn đến xung đột trong quá trình ghi dữ liệu. Session 2 chờ Session 1 thực thi. |
| BEGIN  THEM\_PDV(18, 5);  END;  PL/SQL procedure successfully completed. |  | Session 1 sau khi tra cứu giá vé đã tiến hành lập phiếu dịch vụ cho khách hàng có mã khách hàng là 18 bởi nhân viên có mã là 5. |
| SELECT \*  FROM PHIEUDICHVU  WHERE MAPDV = LAY\_MAPDV(18,5);  MAPDV TIENDV MAKH MANV MATT  ---- ------ ----- ---- ----  33 0 18 5 1 |  | Session 1 kiểm tra lập phiếu thành công. |
| **Session 1** | **Session 2** | **Explanation** |
| BEGIN  THEM\_CTPDV(LAY\_MAPDV(18, 5), 23, 2);  END;  PL/SQL procedure successfully completed. |  | Session 1 tiếp tục thực hiện thao tác thêm vào chi tiết phiếu dịch vụ có mã là mã phiếu của khàng vừa đăng ký, mã dịch vụ là 23 và số lượng dịch vụ là 2. |
| SELECT \*  FROM PHIEUDICHVU  WHERE MAPDV = LAY\_MAPDV(18,5);  MAPDV MADV SLDV NGAYSD  ---- ---- ----- ------  33 23 2 30/06/2021 |  | Session 1 kiểm tra thêm chi tiết phiếu dịch vụ thành công. |
| BEGIN  THEM\_HOADON(18, 5);  END;  PL/SQL procedure successfully completed. |  | Session 1 tiến hành lập hóa đơn cho khách hàng với mã khách hàng là 18 và mã nhân viên là 5; |
| SELECT \*  FROM HOADON  WHERE MAPDV = LAY\_MAPDV(18,5);  MAHD TONGTIEN NGAYTT MANV MAKH MAPT MAPDV  ---- -------- ------ ---- ---- ---- -----  76 704000 30/06/2021 5 18 NULL 33 |  | Session 1 thực hiện câu lệnh truy vấn thông tin hóa đơn của mã phiếu dịch vụ khách hàng Thiện đang sử dụng. |
| BEGIN  XUAT\_HOADON(76);  END;  PL/SQL procedure successfully completed.  Mã hóa đơn: 76 Tên nhân viên: Phạm Thu Huyền Tên khách hàng: Nguyễn Ngọc Thiện  ===> Phiếu dịch vụ: 33  Tên dịch vụ: Đặt vé khu vui chơi Gía dịch vụ: 320000 Ngày sử dụng: 30/06/2021 Số lượng dịch vụ: 2 Tên đơn vị: DamSen’sPark Tiền dịch vụ: 640000  Tiền dịch vụ thực tế: 640000 Tiền dịch vụ sau khi sử dụng khuyến mãi: 6740000  ===> Tổng tiền: 640000 Tổng tiền sau khi tính thuế: 704000 |  | Session 1 tiến hành xuất hóa đơn với mã hóa đơn là 76. Và nhận thấy tiền dịch vụ của chương trình “Đặt vé khu vui chơi” đã không đổi, khách hàng Thiện đăng ký dịch vụ và thanh toán thành công. |
| COMMIT;  Commit completed. |  | Session 1 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh ‘COMMIT’. |
| **Session 1** | **Session 2** | **Explanation** |
|  | 1 row updated. | Session 1 thực hiện ‘COMMIT’ đã nhả khóa cho Session 2.  Session 2 cập nhật thành công dòng dữ liệu thành công. |
|  | COMMIT;  Commit completed. | Session 2 thực hiện thành công và kết thúc phiên làm việc của mình bằng câu lệnh ‘COMMIT’. |

* Code:

**A picture containing table

Description automatically generated**

*Hình 9: Lấy thông tin trong bảng khachhang.*

**Graphical user interface, text, application

Description automatically generated**

*Hình 10: Cập nhật giá trị cccd trong bảng khachhang.*

**Text

Description automatically generated**

*Hình 11: Giải quyết Non-Repeatable Read bằng mức cô lập Serializable..*

1. **Phantom Read:**

- Là tình trạng mà một giao tác đang thao tác trên một tập dữ liệu nhưng giao tác khác lại chèn thêm các dòng dữ liệu vào tập dữ liệu mà giao tác kia quan tâm dẫn đến 2 queries giống hệt nhau được thực hiện nhưng kết quả trả về lại khác nhau.

1. Tình huống: Quản lý tiến hành tra cứu thông tin chi tiết tình trạng phòng để thống kê số lượng phòng trống để tìm hiểu và đưa ra chiến lược về tình hình doanh thu sắp tới. Cùng lúc đó, khách hàng Robeburn Audry đã liên hệ và tiến hành hủy phiếu đặt phòng đã thực hiện trước đó vì tình hình dịch bệnh ngày càng căng thẳng. Lúc này, quản lý tra cứu lại thông tin và nhận thấy dữ liệu bên trong đã thay đổi.
2. Mô tả:

|  |  |  |
| --- | --- | --- |
| **Session 1** | **Session 2** | **Explanation** |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, NULL, NULL, NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 03/01/2021  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 08/01/2021  ...................................  18/04/2021 27/05/2021  04/06/2021 29/06/2021  02/07/2021 15/09/2021  ................................... | No action. | Session 1 thực hiện thủ tục HIEN\_THI\_THEO\_MATTP và tìm kiếm các phòng có tình trạng là 1 tức là phòng trống. |
| No action. | BEGIN  XOA\_MAPDT\_PDT(LAY\_MAPDT(4, 4));  END;  PL/SQL procedure successfully completed. | Session 2 thực hiện giao tác xóa phiếu đặt phòng của khách hàng có makh = 4. Hệ thống tự động cập nhật lại tình trạng phòng khi xóa dữ liệu trong PhieuDatTruoc.  Trong thủ tục XOA\_PDT, khi tiến hành xóa phiếu đặt phòng, thì đồng thời cũng sẽ xóa các chi tiết trong phiếu đó và thực hiện câu lệnh *‘COMMIT’* .  Lúc này, Session 2 thực hiện thành công và kết thúc. |
| **Session 1** | **Session 2** | **Explanation** |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, NULL, NULL, NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 03/01/2021  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 08/01/2021  ...................................  18/04/2021 27/05/2021  04/06/2021 15/09/2021  ................................... | No action. | Session 1 thực hiện lại giao tác tra cứu và hiển thị thông tin trong ChiTietTTP. Lúc này, dữ liệu với maPhong = 19 đã bị thay đổi.  Mã tình trạng phòng ngày 18/04/2021 đến ngày 29/06/2021 và ngày 02/07/2021 đến ngày 15/09/2021 đã bị mất và xuất hiện dòng dữ liệu với ngày bắt đầu là 18/04/2021 và ngày kết thúc là 15/09/2021. |

**→ Phantom Read đã xảy ra.**

1. Nguyên nhân và giải pháp:

* Vấn đề: Session 1 thực hiện truy vấn thông tin chi tiết tình trạng phòng trống, ngay lúc này session 2 xóa phiếu đặt trước và thay đổi tình trạng phòng tương ứng, sau đó session 1 thực hiện truy vấn lại thì phát hiện dữ liệu đã bị thay đổi.
* Nguyên nhân: vì mức cô lập của session 1 là **READ COMMITTED** nên mỗi lần truy vấn trên cùng một đơn vị dữ liệu sẽ đọc lại từ cơ sở dữ liệu (cơ sở dữ liệu lúc này có thể đã bị thay đổi bởi session khác) mặc dù những câu lệnh này đọc trên đơn vị dữ liệu giống nhau.

Giải pháp: đổi mức cô lập ở session1 thành **SERIALIZABLE** thay vì **READ COMMITTED** hoặc sử dụng câu lệnh ***‘LOCK TABLE CHITIETTTP IN ROW SHARE MODE’*** trong stored procedure **HIEN\_THI\_THEO\_MATTP**.

* Xử lý Phantom Read bằng Serializable:

|  |  |  |
| --- | --- | --- |
| **Session 1** | **Session 2** | **Explanation** |
| SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  Transaction ISOLATION succeeded | No action. | Session 1 thiết lập lức cô lập Serializable. |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, NULL, NULL, NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 03/01/2021  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 08/01/2021  ...................................  18/04/2021 27/05/2021  04/06/2021 29/06/2021  02/07/2021 15/09/2021  ................................... | No action. | Session 1 thực hiện thủ tục HIEN\_THI\_THEO\_MATTP và tìm kiếm các phòng có tình trạng là 1 tức là phòng trống. |
| No action. | BEGIN  XOA\_MAPDT\_PDT(LAY\_MAPDT(4, 4));  END;  PL/SQL procedure successfully completed. | Session 2 thực hiện giao tác xóa phiếu đặt phòng của khách hàng có makh = 4. Hệ thống tự động cập nhật lại tình trạng phòng khi xóa dữ liệu trong PhieuDatTruoc.  Trong thủ tục XOA\_PDT, khi tiến hành xóa phiếu đặt phòng, thì đồng thời cũng sẽ xóa các chi tiết trong phiếu đó và thực hiện câu lệnh *‘COMMIT’* .  Lúc này, Session 2 thực hiện thành công và kết thúc. |
| **Session 1** | **Session 2** | **Explanation** |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, NULL, NULL, NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 03/01/2021  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 08/01/2021  ...................................  18/04/2021 27/05/2021  04/06/2021 29/06/2021  02/07/2021 15/09/2021  ................................... |  | Session 1 thực hiện thủ tục HIEN\_THI\_THEO\_MATTP và tìm kiếm các phòng có tình trạng phòng trống một lần nữa. Lúc này, thông tin không bị thay đổi. Quản lý xác nhận thông tin chính xác và tiến hành lập báo cáo, xây dựng chiến lược cho khách sạn. |
| COMMIT;  Commit completed. |  | Session 1 hoàn thành công việc và kết thúc bằng lệnh *‘COMMIT’* quá trình truy xuất thông tin. |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, NULL, NULL, NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 03/01/2021  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 08/01/2021  ...................................  18/04/2021 27/05/2021  04/06/2021 15/09/2021  ................................... | No action. | Session 1 thực hiện lại giao tác tra cứu và hiển thị thông tin trong ChiTietTTP. Lúc này, dữ liệu với maPhong = 19 đã bị thay đổi.  Mã tình trạng phòng ngày 18/04/2021 đến ngày 29/06/2021 và ngày 02/07/2021 đến ngày 15/09/2021 đã được cập nhật thành 18/04/2021 đến ngày 15/09/2021. |

* Xử lý Phantom Read bằng Lock:

|  |  |  |
| --- | --- | --- |
| **Session 1** | **Session 2** | **Explanation** |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, NULL, NULL, NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 03/01/2021  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 08/01/2021  ...................................  18/04/2021 27/05/2021  04/06/2021 29/06/2021  02/07/2021 15/09/2021  ................................... | No action. | Session 1 thực hiện thủ tục HIEN\_THI\_THEO\_MATTP và tìm kiếm các phòng có tình trạng là 1 tức là phòng trống. |
| No action. | BEGIN  XOA\_MAPDT\_PDT(LAY\_MAPDT(4, 4));  END;  Prompt does not return. | Sau đó, session 2 tiến hành giao tác xóa thông tin phiếu đặt phòng của khách hàng Audry.  Tuy nhiên, session 1 đang giữ khóa dòng dữ liệu này và dẫn đến xung đột trong quá trình ghi dữ liệu. |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, NULL, NULL, NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 03/01/2021  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 08/01/2021  ...................................  18/04/2021 27/05/2021  04/06/2021 29/06/2021  02/07/2021 15/09/2021  ................................... | Prompt does not return. | Session 1 thực hiện thủ tục HIEN\_THI\_THEO\_MATTP và tìm kiếm các phòng có tình trạng phòng trống một lần nữa. Lúc này, thông tin không bị thay đổi. Quản lý xác nhận thông tin chính xác và tiến hành lập báo cáo, xây dựng chiến lược cho khách sạn. |
| **Session 1** | **Session 2** | **Explanation** |
| COMMIT;  Commit completed. | Prompt does not return. | Session 1 hoàn thành công việc và kết thúc bằng lệnh *‘COMMIT’* quá trình truy xuất thông tin. |
| No action. | PL/SQL procedure successfully completed. | Session 2 thực hiện giao tác xóa phiếu đặt phòng của khách hàng có makh = 4. Hệ thống tự động cập nhật lại tình trạng phòng khi xóa dữ liệu trong PhieuDatTruoc.  Trong thủ tục XOA\_PDT, khi tiến hành xóa phiếu đặt phòng, thì đồng thời cũng sẽ xóa các chi tiết trong phiếu đó và thực hiện câu lệnh *‘COMMIT’* .  Lúc này, Session 2 thực hiện thành công và kết thúc. |
| BEGIN  HIEN\_THI\_THEO\_MATTP (1, NULL, NULL, NULL, NULL);  END;  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  1 SD101 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 03/01/2021  **……………………………..**  MAPHONG TENPHONG GIAPHONG  ------- -------- --------  19 SD1019 800000  NGAYBD NGAYKT  ------ -------  01/01/2021 08/01/2021  ...................................  18/04/2021 27/05/2021  04/06/2021 15/09/2021  ................................... | No action. | Session 1 thực hiện lại giao tác tra cứu và hiển thị thông tin trong ChiTietTTP. Lúc này, dữ liệu với maPhong = 19 đã bị thay đổi.  Mã tình trạng phòng ngày 18/04/2021 đến ngày 29/06/2021 và ngày 02/07/2021 đến ngày 15/09/2021 đã được cập nhật thành 18/04/2021 đến ngày 15/09/2021. |

* Code:

A picture containing application

Description automatically generated

*Hình 12: Lấy thông tin trong bảng khachhang.*

Text

Description automatically generated with low confidence

*Hình 13: Thêm thông tin khách hàng mới vào bảng khachhang.*

Graphical user interface, text, application, email

Description automatically generated

*Hình 14: Giải quyết Phantom Read bằng mức cô lập Serializable.*

# **CHƯƠNG 6: THIẾT KẾ GIAO DIỆN**

## Danh sách các màn hình:

|  |  |  |
| --- | --- | --- |
| **STT** | **Màn hình** | **Chức năng** |
| 1 | Đăng nhập | Đăng nhập vào hệ thống |
| 2 | Màn hình chính của quản lý | Hiển thị các chức năng mà quản lý có thể truy cập được |
| 2 | Màn hình quản lý quy định | Hiển thị các quy định và mô tả quy định |
| 3 | Màn hình thêm quy định | Hiển thị các trường cần thiết để thêm quy định và lưu thông tin vào cơ sở dữ liệu |
| 4 | Màn hình thống kê nhân viên | Hiển thị thống kê nhân viên – quản lý |
| 5 | Màn hình tra cứu nhân viên | Hiển thị chức năng tra cứu nhân viên và danh sách nhân viên tra cứu, các nút xóa, sửa nhân viên |
| 6 | Màn hình thêm nhân viên | Hiển thị các trường cần thiết để thêm nhân viên và lưu lại thông tin vào cơ sở dữ liệu |
| 7 | Màn hình sửa nhân viên | Hiển thị các trường cần thiết cho phép sửa thông tin nhân viên và lưu lại dưới cơ sở dữ liệu |
| 8 | Màn hình quản lý dịch vụ | Hiển thị các trường để tra cứu dịch vụ, các nút chức năng thêm, xóa, sửa dịch vụ. Các trường để tra cứu phiếu dịch vụ và các nút chức năng để thêm, xóa, xem chi tiết phiếu dịch vụ |
| 9 | Màn hình sửa dịch vụ | Hiển thị các trường cần thiết cho phép sửa thông tin dịch vụ và lưu lại dưới cơ sở dữ liệu |
| 10 | Màn hình thêm dịch vụ | Hiển thị các trường cần thiết để thêm dịch vụ và lưu vào cơ sở dữ liệu |
| 11 | Màn hình chi tiết dịch vụ | Hiển thị chi tiết các phiếu dịch vụ và nút chức năng xóa phiếu dịch vụ |
| 12 | Màn hình thêm dịch vụ vào phiếu dịch vụ | Hiển thị các trường để thêm dịch vụ vào phiếu dịch vụ và lưu vào cơ sở dữ liệu |
| 13 | Màn hình chi tiết dịch vụ | Hiển thị chi tiết thông tin dịch vụ |
| 14 | Màn hình chính của nhân viên | Hiển thị các chức năng mà nhân viên có quyền truy cập |
| 15 | Màn hình quản lý khách hàng | Hiển thị các trường để tra cứu thông tin khách hàng, thống kê loại khách hàng và các nút chức năng tra cứu, thêm, xóa, sửa khách hàng |
| 16 | Màn hình thêm khách hàng | Hiển thị các trường cần thiết để thêm thông tin khách hàng và lưu vào cơ sở dữ liệu |
| 17 | Màn hình sửa khách hàng | Hiển thị các trường cần thiết cho phép sửa thông tin khách hàng và lưu lại dưới cơ cở dữ liệu |
| 18 | Màn hình quản lý hóa đơn | Hiển thị thông tin hóa đơn và chức năng tra cứu hóa đơn |
| 19 | Màn hình thêm hóa đơn | Hiển thị trường để tra cứu khách hàng, dánh sách phiếu thuê phòng tương ứng và phiếu dịch vụ tương ứng, thông tin hóa đơn của khách hàng đó |
| 20 | Màn hình quản lý phiếu thuê phòng | Hiển thị danh sách thông tin phiếu thuê phòng và các nút chức năng để thêm/tra cứu phiếu thuê phòng |
| 21 | Màn hình thêm phiếu thuê phòng | Hiển thị các trường cần thiết để thêm phiếu thuê phòng và lưu vào cơ sở dữ liệu |
| 22 | Màn hình lập phiếu thuê phòng | Hiển thị các trường để nhập thông tin khách hàng và lưu vào cơ sở dữ liệu, hiển thị thông tin thuê phòng của khách hàng |
| 23 | Màn hình chi tiết phiếu thuê phòng | Hiển thị chi tiết các phiếu thuê phòng |
| 24 | Màn hình quản lý phiếu đặt phòng | Hiển thị danh sách các phiếu đặt phòng và các nút chức năng để thêm, tra cứu, xóa phiếu thuê phòng và lưu lại dưới cơ sở dữu liệu |
| 25 | Màn hình thêm phiếu đặt phòng | Hiển thị các trường cần thiết để thêm phiếu đặt phòng và lưu vào cơ sở dữ liệu |
| 26 | Màn hình chi tiết phiếu đặt phòng | Hiển thị chi tiết các phiếu đặt phòng và nút chức năng xóa phiếu đặt phòng |
| 27 | Màn hình xuất phiếu dịch vụ | Hiển thị thông tin dịch vụ mà khách hàng đã sủ dụng |
| 28 | Màn hình xuất hóa đơn | Hiển thị thông tin hóa đơn |
| 29 | Màn hình thống kê theo tháng | Hiển thị thống kê tài chính theo tháng |
| 30 | Màn hình thống kê dịch vụ sử dụng | Hiển thị thống kê các dịch vụ được sử dụng nhiều trong tháng |

## II. Mô tả các màn hình:

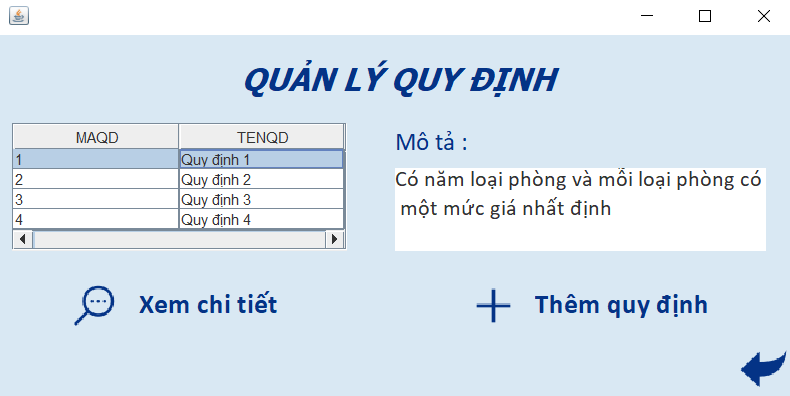
1. Đăng nhập:



1. Màn hình chính của quản lý:



### Màn hình quản lý quy định:



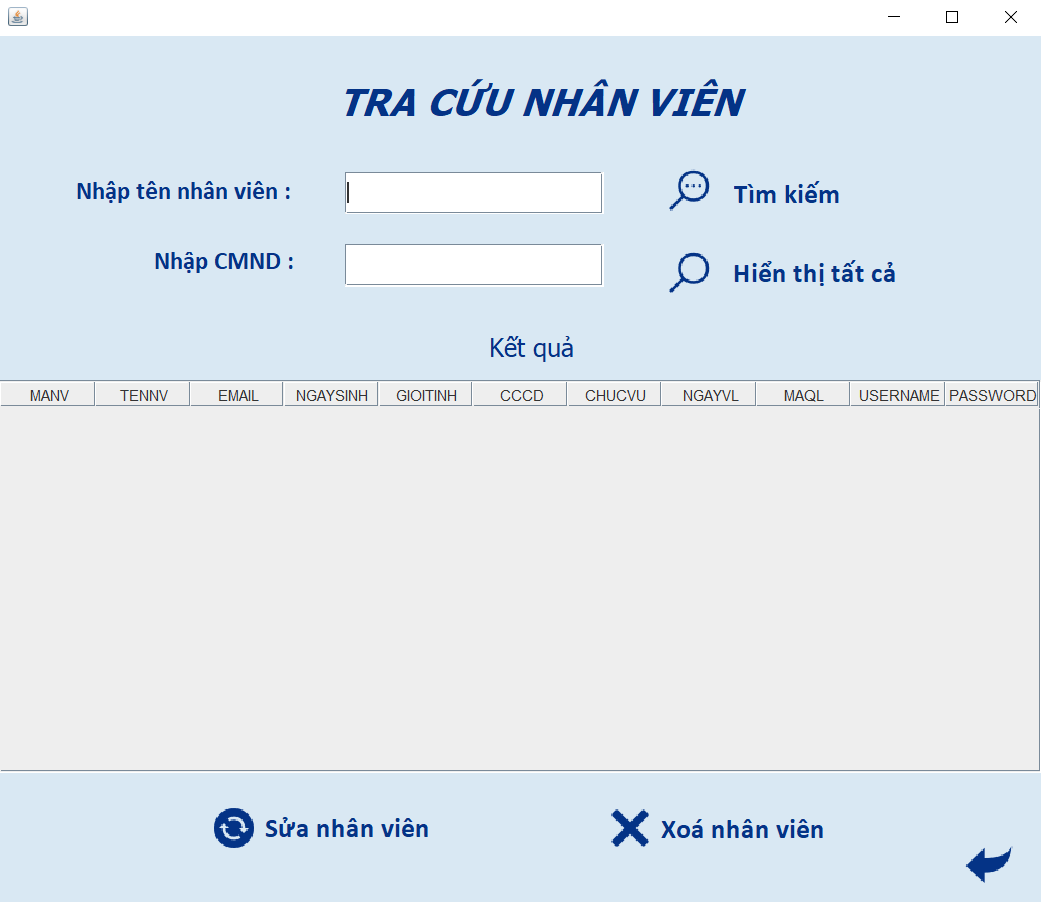
1. Màn hình thêm quy định



1. Màn hình thống kê nhân viên



1. Màn hình tra cứu nhân viên



1. Màn hình thêm nhân viên



1. Màn hình sửa nhân viên



1. Màn hình quản lý dịch vụ



1. Màn hình sửa dịch vụ



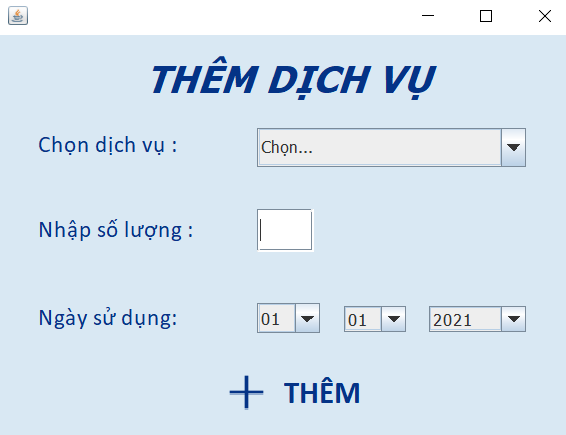
1. Màn hình thêm dịch vụ



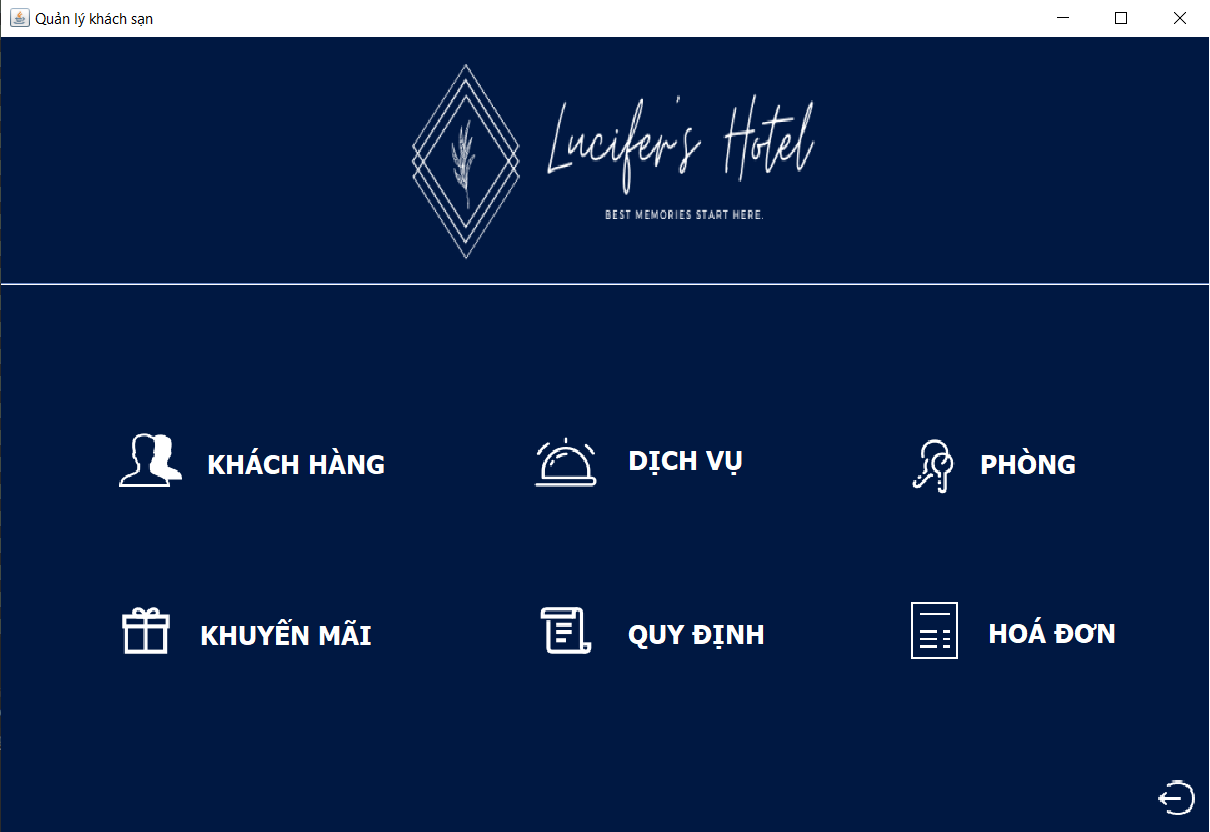
1. Màn hình chi tiết phiếu dịch vụ



1. Màn hình thêm dịch vụ vào phiếu dịch vụ



1. Màn hình chính của nhân viên



1. Màn hình quản lý khách hàng



1. Màn hình thêm khách hàng



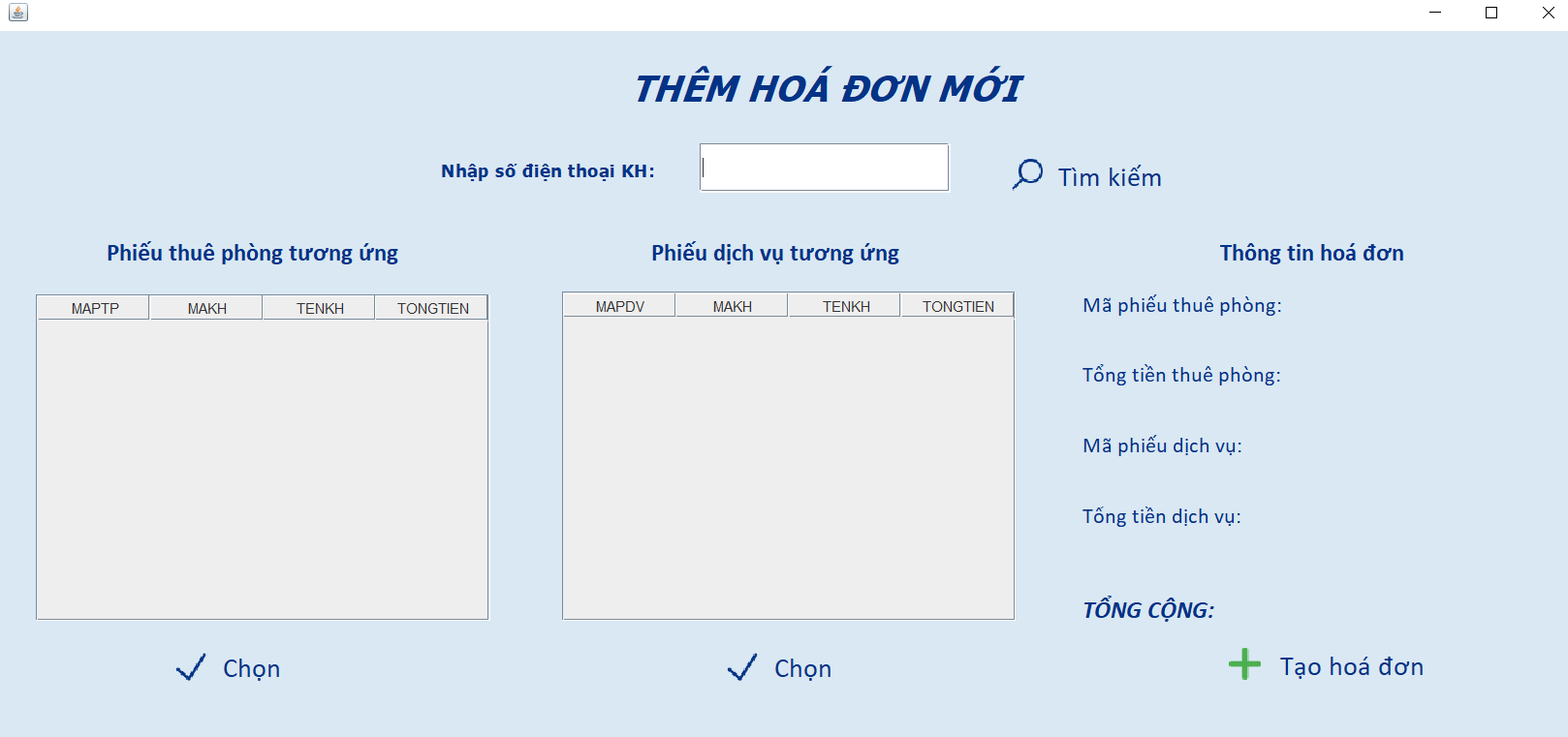
1. Màn hình sửa khách hàng



1. Màn hình quản lý hóa đơn



1. Màn hình thêm hóa đơn



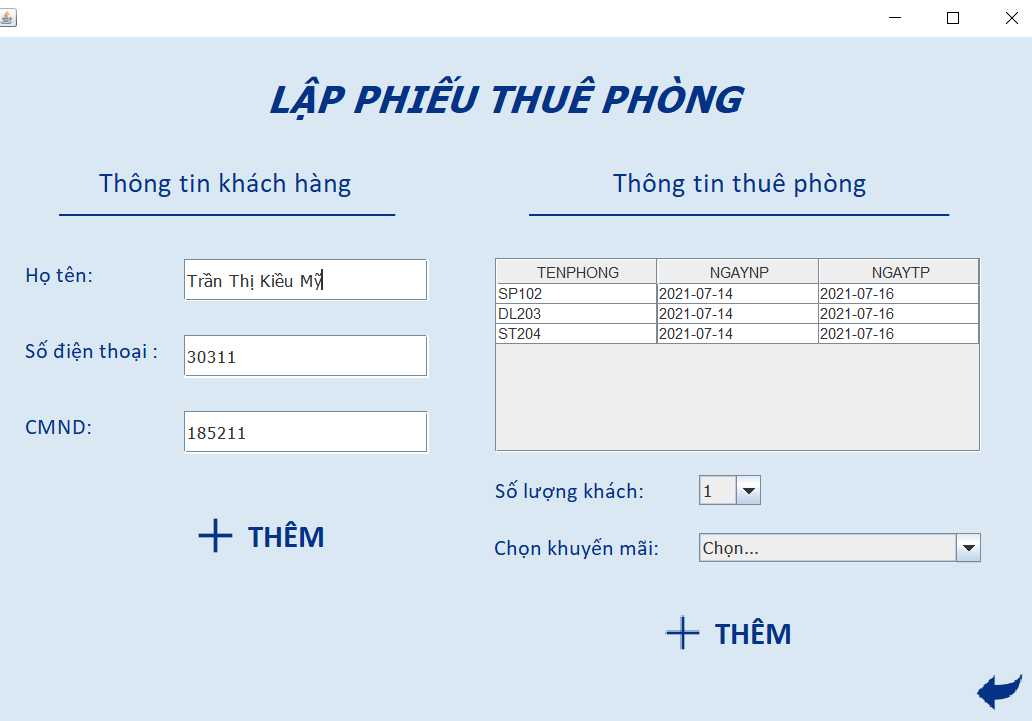
1. Màn hình quản lý phiếu thuê phòng



1. Màn hình thêm phiếu thuê phòng



1. Màn hình lập phiếu thuê phòng



1. Màn hình chi tiết phiếu thuê phòng



1. Màn hình quản lý phiếu đặt phòng



1. Màn hình thêm phiếu đặt phòng



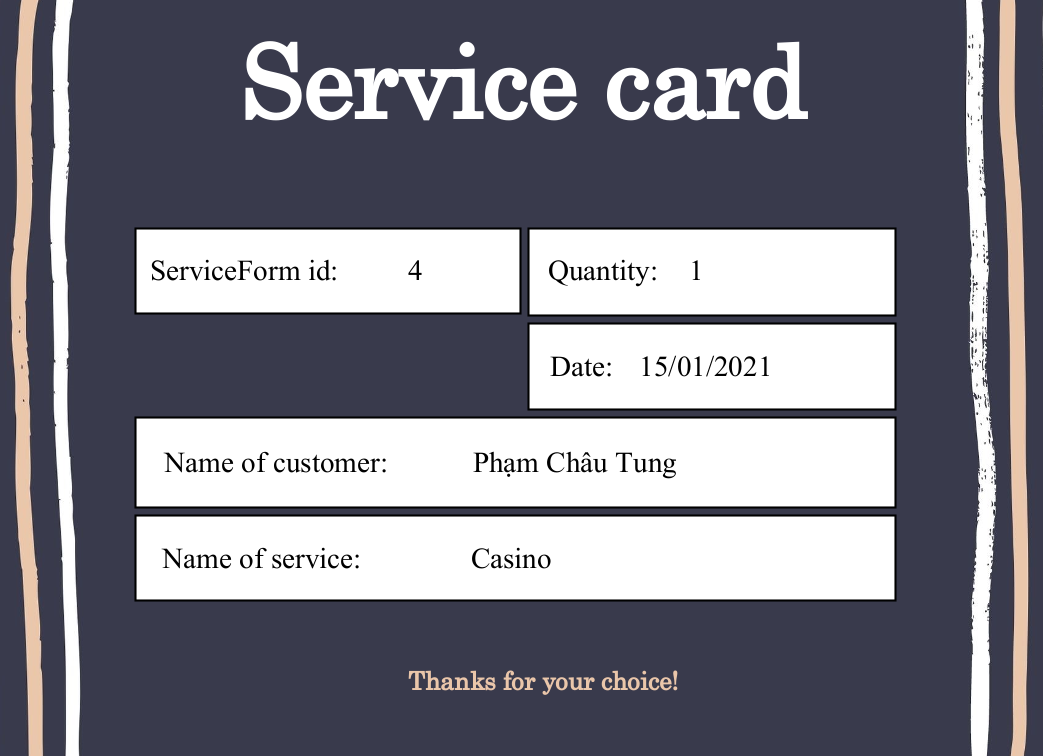
1. Màn hình chi tiết phiếu đặt phòng



1. Màn hình thống kê theo tháng



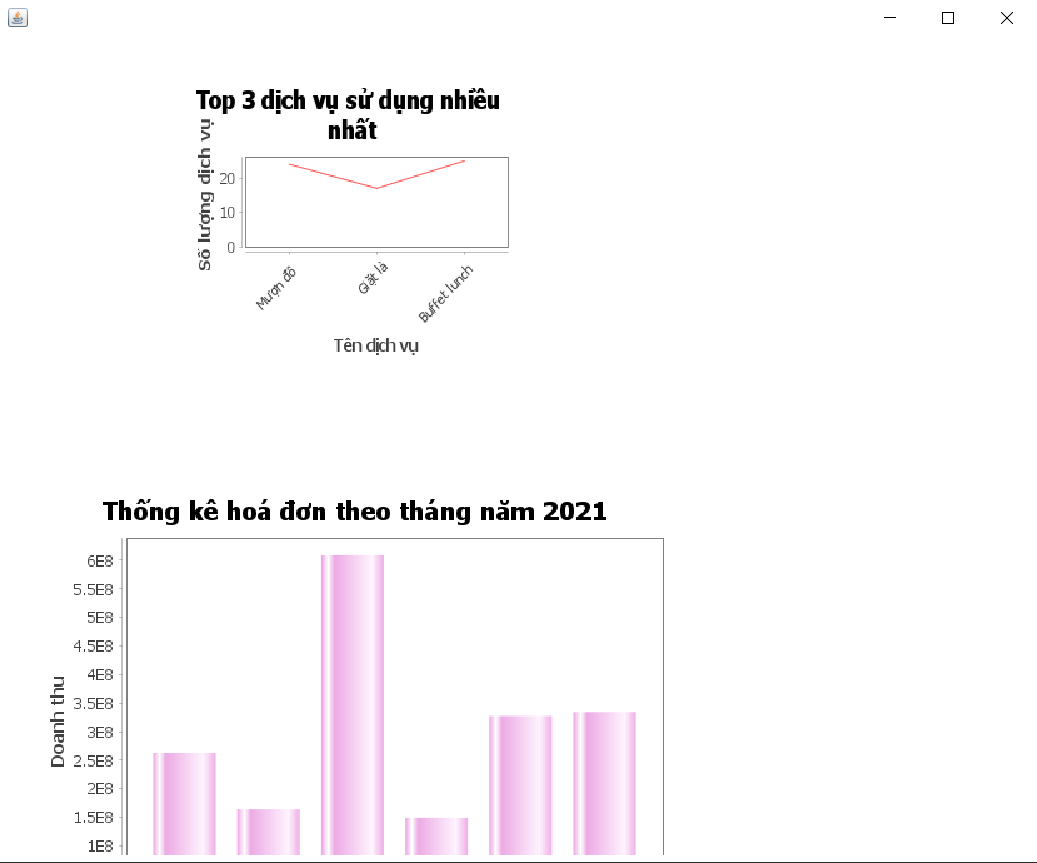
1. Màn hình xuất phiếu dịch vụ



1. Màn hình xuất hóa đơn



1. Màn hình thống kê dịch vụ được sử dụng



# **CHƯƠNG 7: KẾT LUẬN**

1. **Kết quả đạt được:**

Trong suốt quá trình làm đồ án vừa qua, nhóm đã đạt được những điều sau:

* Biết cách thiết kế, hoàn thiện một dự án nhỏ.
* Nắm được kĩ năng phân chia công việc, sắp xếp thời gian, làm việc nhóm.
* Hiểu và vận dụng được kiến thức của môn Hệ quản trị cơ sở dữ liệu, cụ thể là làm việc trên Oracle như:

+ Các cứ pháp và cách tư duy của ngôn ngữ PL/SQL.

+ Hiểu được sự khác nhau giữa ngôn ngữ PL/SQL và T-SQL.

+ Cách thực hiện Trigger, Function, Stored Procedure.

+ Nắm được các kiến thức về Khóa và các mức cô lập, mô phỏng được các trường hợp khóa và cô lập của Oracle.

+ Giải quyết được các trường hợp truy xuất đồng thời.

1. **Hạn chế:**

Mặc dù nhóm đã cố gắng hết sức, nhận được sự góp ý của Cô cũng như tham khảo các nguồn tại liệu nhưng phần mềm của nhóm vẫn có những hạn chế nhất định như sau:

* Giao diện chưa thu hút, thuận tiện cho người dùng.
* Các trường hợp về trigger, stored procedure vẫn còn đơn giản chưa có độ phức tạp khó.
* Các tình huống truy xuất đồng thời và khóa chưa có độ phức tạp cao, có thể chưa hợp lý.

1. **Hướng phát triển:**

Với những hạn chế đã nêu trên, nhóm cũng mong muốn có thể phát triển trong phiên bản kế tiếp của phần mềm. Đồng thời, nhóm cũng muốn tìm hiểu và thay đổi giao diện sao cho dễ tương tác và thân thiện với người dùng hơn. Các xử lý của phần mềm được trơn tru, có độ phức tạp cao hơn và mang lại cho người dùng những tiện ích tốt hơn. Tóm lại, với phiên bản kế tiếp, nhóm mong muốn xây dựng một phần mềm tiện ích, dễ dùng và khả năng vận hành trong thực tế cao hơn.

# **PHỤ LỤC 1: BẢNG PHÂN CHIA CÔNG VIỆC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Công việc | Phương Uyên | Thu Phương | Hữu Thắng | Thái Toàn |
| Phân tích bài toán, xác định các chức năng | X | X | X | X |
| Thiết kế mô hình dữ liệu | X | X | X | X |
| Thiết kế CSDL | X | X | X | X |
| Thiết kế các Trigger, Function, Stored Procedure | X | X |  |  |
| Xử lý các vấn đề liên quan đến truy xuất đồng thời | X |  | X |  |
| Thiết kế giao diện | X | X | X | X |
| Xử lý giao diện | X |  |  | X |
| Lập trình | X | X | X | X |
| Test chức năng | X |  |  | X |
| Soạn và chỉnh sửa báo cáo |  | X | X |  |

# **PHỤ LỤC 2: TÀI LIỆU THAM KHẢO**

1. Các slide môn của khoa Hệ thống thông tin.

2. Giáo tình môn Hệ quản trị cơ sở dữ liệu Oracle– Biên soạn: (TS.Nguyễn Thị Trà Linh).

3. [Oracle Database Concepts, 18c, August 2018.](https://courses.uit.edu.vn/pluginfile.php/267880/mod_resource/content/1/database-concepts.pdf)

4. [Dirty read, Phantom read and non-Repeatable read.](http://javadata.blogspot.com/2011/05/dirty-readphantom-read-and-non.html)

Link file nộp: <https://drive.google.com/drive/u/1/folders/1tqmaRBn5Go8i8mlqmDrabDmDkThXp5cy>