

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
ĐẠI HỌC QUỐC GIA TP.HCM



BÁO CÁO ĐỒ ÁN

Đề tài: THIẾT KẾ HỆ THỐNG QUẢN LÝ VÉ MÁY BAY

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

Sinh viên thực hiện : **Nhóm 7**

Lê Văn Pa – 15520596

Lã Ngô Mỹ Linh – 17520683

Đặng Văn Công Hoàng – 17520503

Mai Trung Tín – 15520894

TP. Hồ Chí Minh, ngày 28 tháng 6 năm 2019

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
ĐẠI HỌC QUỐC GIA TP.HCM



BÁO CÁO ĐỒ ÁN

Đề tài: THIẾT KẾ HỆ THỐNG QUẢN LÝ VÉ MÁY BAY

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

Sinh viên thực hiện : **Nhóm 7**

Lê Văn Pa – 15520596

Lã Ngô Mỹ Linh – 17520683

Đặng Văn Công Hoàng – 17520503

Mai Trung Tín – 15520894

TP. Hồ Chí Minh, ngày 28 tháng 6 năm 2019

MỞ ĐẦU

Trên thế giới hiện nay, tin học là một ngành phát triển không ngừng. Thời kỳ công nghiệp đòi hỏi đòi hỏi thông tin nhanh chóng chính xác. Có thể nói tin học đã chiếm một vị trí quan trọng trong cuộc sống hàng ngày, trong các ngành khoa học kỹ thuật. Trong những năm gần đây cùng với sự phát triển chung của ngành tin học thế giới và khu vực, ngành tin học nước ta đó cú bước phát triển nhất định.

Việc đưa tin học vào công việc quản lý, nhằm giảm bớt sức lao động của con người, tiết kiệm thời gian, độ chính xác cao và tiện lợi rất nhiều so với việc làm quản lý giắt tờ bằng thủ công như trước đây. Tin học hóa nhằm thu hẹp không gian lưu trữ, tránh được thất lạc dữ liệu, tự động hóa và cụ thể hóa các thông tin theo nhu cầu của con người. Còn ở Việt Nam ngành tin học tuy còn non trẻ nhưng cũng đang từng bước bứt phá được những thành công trong việc áp dụng tin học vào việc phát triển kinh tế – xã hội, góp phần vào việc phát triển xã hội và khẳng định mình ở thị trường quốc tế.

Chính vì vậy khi em được nhận đề tài xây dựng chương trình quản lý vé máy bay dùng hệ quản trị CSDL Oracle, với những kiến thức đã được học và quá trình tìm hiểu, đặc biệt là sự hướng dẫn rất đổi nhiệt tình của cô giáo, Ths. Đỗ Thị Minh Phụng, chúng em đã hoàn thành được đề tài của mình với một bài toán quản lý. Trong quá trình làm bài chắc chắn em sẽ mắc phải những thiếu sót nên rất mong nhận được sự đóng góp ý kiến của cô để em có cơ hội sửa sai, phát triển trong thực tế và từng bước hoàn thiện mình.

LỜI CẢM ƠN

Lời đầu tiên, nhóm xin đặc biệt gửi lời cảm ơn đến Cô Đỗ Thị Minh Phụng (Giáo viên môn Hệ quản trị cơ sở dữ liệu). Cô đã cung cấp kiến thức, chỉ bảo và đóng góp những ý kiến quý báu giúp nhóm hoàn thành được đề án môn học của mình.

Xuất phát từ mục đích học tập, tìm hiểu sâu hơn các kiến thức về các hệ quản trị cơ sở dữ liệu, cũng như tìm hiểu thêm về quy trình nghiệp vụ của các rạp chiếu phim, nhóm chúng em đã thực hiện đề án “Thiết kế hệ thống quản lý vé máy bay”. Trong quá trình thực hiện đề án, dựa trên những kiến thức được Cô cung cấp trên trường kết hợp với việc tự tìm hiểu những công cụ và kiến thức mới, nhóm đã cố gắng thực hiện đề án một cách tốt nhất. Tuy nhiên, đề án chưa hoàn thiện và còn nhiều sai sót nhưng nó là kết quả của sự nỗ lực của các thành viên trong nhóm, sự giúp đỡ của tất cả bạn bè và Cô.

Nhóm rất mong nhận sự góp ý từ phía Cô nhằm rút ra những kinh nghiệm quý báu và hoàn thiện vốn kiến thức để nhóm có thể tiếp tục hoàn thành những đề án khác trong tương lai. Xin chân thành cảm ơn quý Cô!

Nhóm sinh viên thực hiện

TÓM TẮT BÁO CÁO

Nội dung bài báo cáo này là kết quả quá trình thực hiện đồ án của các thành viên trong nhóm trong vòng 4 tháng. Thời gian từ ngày 28 tháng 02 năm 2019 đến ngày 28 tháng 06 năm 2019

Bài báo cáo bao gồm các chương sau:

- Chương 1: Tổng quan đề tài
- Chương 2: Phân tích, thiết kế chức năng
- Chương 3: Thiết kế CSDL
- Chương 4: Xây dựng và quản lý giao tác
- Chương 5: Xử lý truy xuất đồng thời
- Chương 6: Thiết kế giao diện
- Chương 7: Tổng kết

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the entire width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Giảng viên hướng dẫn

MỤC LỤC

DANH MỤC HÌNH ẢNH.....	1
DANH MỤC BẢNG BIỂU VÀ TỪ VIẾT TẮT	2
Chương 1. TỔNG QUAN ĐỀ TÀI.....	3
1.1. Đặt vấn đề	3
1.2. Giải pháp	3
1.3. Công nghệ sử dụng	3
Chương 2. PHÂN TÍCH, THIẾT KẾ CHỨC NĂNG	4
2.1. Phân tích yêu cầu.....	4
2.1.1. Yêu cầu chức năng.....	4
2.1.2. Yêu cầu phi chức năng.....	4
2.2. Thiết kế chức năng.....	5
Chương 3. THIẾT KẾ CƠ SỞ DỮ LIỆU	6
3.1. Mô hình thực thể - mối kết hợp (ERD)	6
3.2. Mô hình quan hệ.....	6
3.3. Mô tả các bảng	7
3.4. Ràng buộc toàn vẹn	10
Chương 4. XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC	11
4.1. Triggers.....	11
4.1.1. Cơ sở lý thuyết.....	11
4.1.2. Danh sách các trigger	11
4.1.3. Mô tả một số trigger	12
4.2. Stored procedures.....	14
4.2.1. Cơ sở lý thuyết.....	14
4.2.2. Danh sách các stored procedure	14
4.2.3. Mô tả một số stored procedure.....	16
4.3. Functions	19
4.3.1. Cơ sở lý thuyết.....	19
4.3.2. Danh sách các function.....	20
4.3.3. Mô tả một số function	20
4.4. Tối ưu hóa một số câu truy vấn	21

Chương 5.	XỬ LÝ TRUY XUẤT ĐỒNG THỜI	23
5.1.	Các mức cô lập.....	23
5.1.1.	Read Uncommitted	23
5.1.2.	Read Committed	23
5.1.3.	Repeatable Read.....	24
5.1.4.	Serializable.....	25
5.2.	Lost update	25
5.3.	Dirty read	28
5.4.	Unrepeatable read	30
5.5.	Phantom.....	32
5.6.	Deadlock.....	33
Chương 6.	THIẾT KẾ GIAO DIỆN	36
6.1.	Danh sách các màn hình	36
6.2.	Sơ đồ liên kết màn hình	37
6.3.	Mô tả chức năng từng màn hình	37
Chương 7.	TỔNG KẾT	44
7.1.	Kết quả đạt được.....	44
7.2.	Hạn chế.....	44
7.3.	Hướng phát triển.....	45
BẢNG PHÂN CÔNG CÔNG VIỆC.....		46
TÀI LIỆU THAM KHẢO.....		47

DANH MỤC HÌNH ẢNH

Hình 1: Mô hình ERD.....	6
Hình 2: Cấu trúc của trigger trong Oracle	11
Hình 3: Cấu trúc của stored procedure trong Oracle	14
Hình 4: Cấu trúc của function trong Oracle.....	19
Hình 5: Nguyên nhân gây deadlock.....	Error! Bookmark not defined.
Hình 6: Sơ đồ liên kết màn hình	37

DANH MỤC BẢNG BIỂU VÀ TỪ VIẾT TẮT

Danh mục từ viết tắt:

CSDL – Cơ sở dữ liệu

HQT CSDL – Hệ quản trị cơ sở dữ liệu

RBTV – Ràng buộc toàn vẹn

Danh mục bảng biểu:

Bảng 1: Yêu cầu CSDL.....	4
Bảng 2: Thiết kế chức năng	5
Bảng 3: mô tả bảng DANGNHAP.....	7
Bảng 4: Mô tả bảng HANGMAYBAY	7
Bảng 5: Mô tả bảng SANBAY	7
Bảng 6: Mô tả bảng CHUYENBAY.....	8
Bảng 7: Mô tả bảng KHACHHANG.....	8
Bảng 8: Mô tả bảng VE	9
Bảng 9: Mô tả bảng CHITIETVE.....	9
Bảng 10: Mô tả bảng KHUYENMAI	10
Bảng 11: Các ràng buộc toàn vẹn	10
Bảng 12: Danh sách trigger	12
Bảng 13: Danh sách stored procedure	16
Bảng 14: Danh sách function.....	20
Bảng 15: Danh sách các màn hình.....	36

Chương 1. TỔNG QUAN ĐỀ TÀI

1.1. Đặt vấn đề

Việc tin học hóa quy trình, công cụ quản lý vé máy bay cũng như các chuyến bay, khách hàng ngày càng trở thành mối quan tâm của nhiều doanh nghiệp, cơ quan, tổ chức. Lợi ích của các hệ thống quản lý này đem lại là vô cùng to lớn. Nó khắc phục được những yếu điểm của phương pháp quản lý truyền thống bằng thủ công và mở ra một môi trường làm việc hiện đại, tự động cho doanh nghiệp, tổ chức. Từ đó, phương pháp này đem lại nhiều hiệu quả về mặt thời gian, chi phí và nhân lực cho doanh nghiệp.

Ngày nay nhu cầu du lịch của mọi người ngày càng cao, nhu cầu sử dụng dịch vụ hàng không theo đó cũng tăng vọt. Điều đó dẫn tới quy mô CSDL cần lưu trữ của hãng hàng không ngày càng lớn làm cho việc quản lý CSDL truyền thống không còn hiệu quả, gây tiêu tốn thời gian, tiền bạc, công sức.

1.2. Giải pháp

Để giải quyết vấn đề trên, nhóm chúng em quyết định thiết kế phần mềm Quản lý vé máy bay, sử dụng HQT CSDL Oracle và các công nghệ của nó. Kèm theo việc áp dụng các kỹ thuật xử lý các giao tác xảy ra đồng thời và giải quyết các vấn đề khác của HQT CSDL, chúng em mong muốn phần mềm này sẽ được đưa vào sử dụng trong thực tiễn.

1.3. Công nghệ sử dụng

- HQT CSDL: Oracle SQL Developer
- Ngôn ngữ xây dựng phần mềm: Java
- IDE xây dựng phần mềm: Netbean
- Công cụ quản lý mã nguồn: Github

Chương 2. PHÂN TÍCH, THIẾT KẾ CHỨC NĂNG

2.1. Phân tích yêu cầu

2.1.1. Yêu cầu chức năng

- Yêu cầu CSDL phải quản lý được các đối tượng sau:

Đối tượng	Mô tả
Khách hàng	Lưu các thông tin cơ bản của khách hàng, để phân biệt từng khách hàng với nhau
Nhân viên hỗ trợ	Mỗi khi khách hàng cần tra vé, tra chuyến bay hay đặt vé chuyến bay thì có nhân viên hỗ trợ cho khách hàng. Phần mềm cho phép nhiều nhân viên đăng nhập vào cùng một lúc
Chuyến bay	Lưu thông tin chuyến bay
Vé máy bay	Thông tin vé có chưa thông tin chuyến bay và thông tin khách hàng cũng như nhân viên hỗ trợ
Khuyến mãi	Lưu thông tin chương trình khuyến mãi hoặc mã khuyến mãi (coupon, voucher,...)
Sân bay	Lưu thông tin sân bay

Bảng 1: Yêu cầu CSDL

- Yêu cầu phần mềm cho phép người dùng tạo chuyến bay, tạo khách hàng, và các đối tượng khác theo như yêu cầu.
- Có tính năng tìm kiếm chuyến bay, đặt vé / hủy vé máy bay.
- Cho phép truy xuất lại thông tin chi tiết vé đã đặt, xem lịch sử chuyến bay và lịch sử vé.
- Tạo thống kê, báo cáo theo năm, tháng, quý.
- Có thể sao lưu và phục hồi dữ liệu.

2.1.2. Yêu cầu phi chức năng

- Yêu cầu giao diện: tiện dụng, thân thiện, sinh động và tương tác cao.
- Yêu cầu hiệu suất:

- Thời gian cho phép để hệ thống phản hồi lại thông tin đã tiếp nhận yêu cầu xử lý từ phía người sử dụng là 5 giây.
 - Thời gian cho phép để trả kết quả tìm kiếm thông tin là dưới 7 giây.
 - Phần mềm sử dụng hợp lý tài nguyên hệ thống để tránh treo máy, tràn bộ nhớ ảnh hưởng các chương trình khác.
 - Cho phép nhiều người sử dụng cùng lúc mà không bị xung đột dữ liệu.
- Yêu cầu an toàn:
- Hệ thống có thể sao lưu, phục hồi lại cơ sở dữ liệu khi có sự cố.
 - Phần mềm chạy trên hệ thống không xung đột với các phần mềm khác.

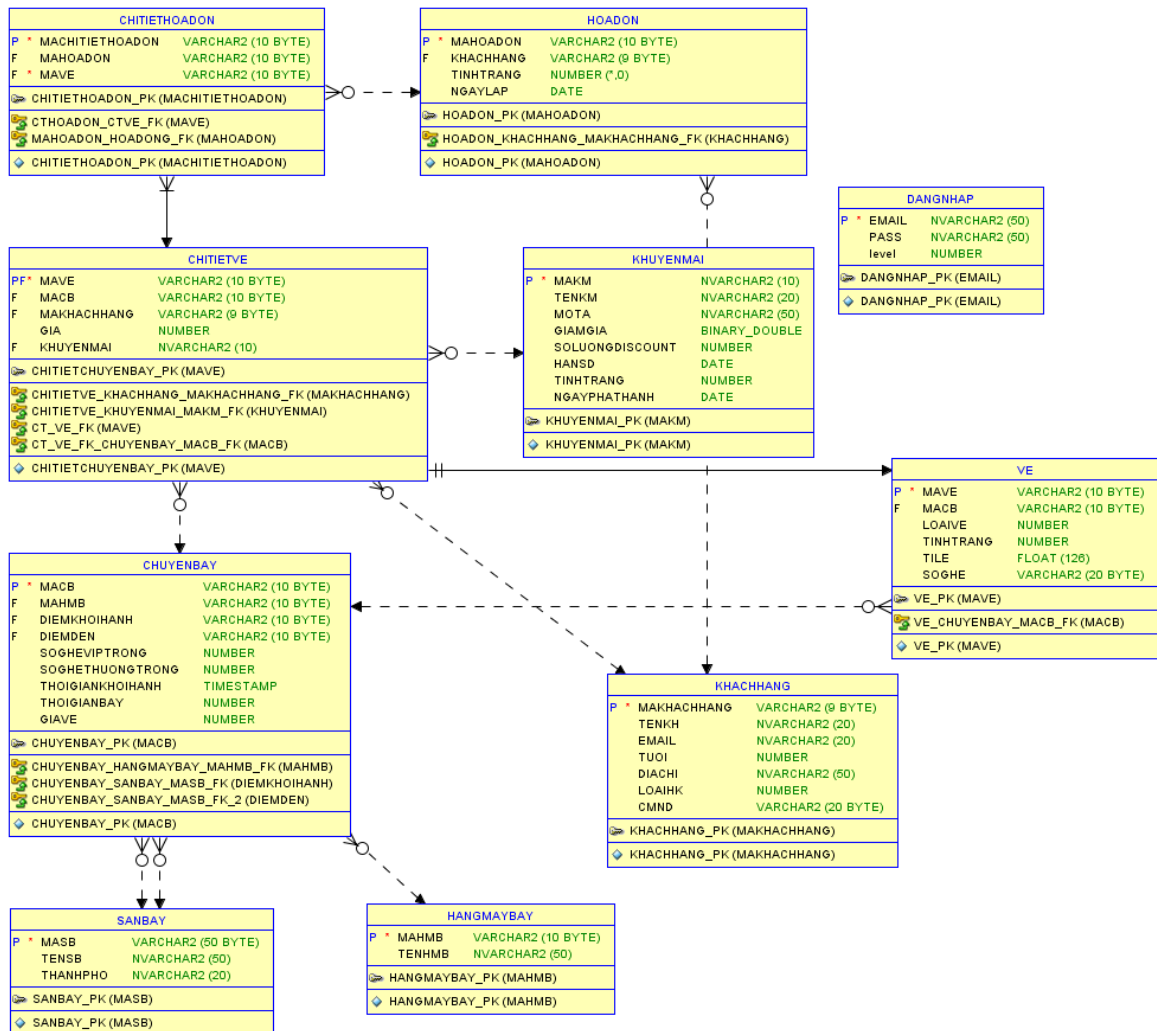
2.2. Thiết kế chức năng

Tên chức năng	Mô tả
Tạo chuyến bay	Tạo chuyến bay mới với thời gian khởi hành, địa điểm đi, địa điểm đến, thời gian bay, số ghế và một số thông tin khác
Đặt vé	Đặt vé khi khách hàng yêu cầu, thông tin vé cần có thông tin chuyến bay, loại vé và giá trị để tính giá vé
Tra cứu vé	Sau khi đặt vé, nếu khách hàng có nhu cầu kiểm tra lại thì dùng chức năng này để truy xuất thông tin vé. Trong thông tin vé có đầy đủ thông tin khách hàng, thông tin chuyến bay và thông tin vé
Thống kê	Thống kê số lượng khách/chuyến bay trong tuần/tháng/năm hoặc trong một khoảng thời gian nhất định.
Khuyến mãi	Áp dụng khuyến mãi khi mua vé, có 2 loại là khuyến mãi theo khoảng thời gian và giảm giá theo voucher.

Bảng 2: Thiết kế chức năng

Chương 3. THIẾT KẾ CƠ SỞ DỮ LIỆU

3.1. Mô hình thực thể - mối kết hợp (ERD)



Hình 1: Mô hình ERD

3.2. Mô hình quan hệ

- **DANGNHAP**: EMAIL, PASS, LEVEL
- **HANGMAYBAY**: MAHMB, TENHMB
- **SANBAY**: MASB, TENSB, THANHPHO
- **CHUYENBAY**: MACB, MAHMB, DIEMKHOIHANH, DIEMDEN, SOGHEVIPTRONG, SOGHETHUONGTRONG, THOIGIANKHOIHANH, THOIGIANBAY, GIAVE

- **KHACHHANG:** MAKHACHHANG, TENKH, EMAIL, TUOI, DIACHI, LOAIKH, CMND
- **VE:** MAVE, MACB, LOAIVE, TINHTRANG, TILE, SOGHE
- **CHITIETVE:** MAVE, MACB, MAKHACHHANG, GIA, KHUYENMAI
- **KHUYENMAI:** MAKM, TENKM, MOTA, GIAMGIA, SOLUONGDISCOUNT, HANS, TINHTRANG, NGAYPHATHANH
- **HOADON:** MAHOADON, KHACHHANG, TINHTRANG, NGAYLAP
- **CHITIETHOADON:** MACHITIETHOADON, MAHOADON, MAVE,

3.3. Mô tả các bảng

- Bảng DANGNHAP

STT	Thuộc tính	Kiểu dữ liệu	null	Mô tả
1	EMAIL	NVARCHAR2(50)	x	Email nhân viên, dùng như là username
2	PASS	NVARCHAR2(50)		Mật khẩu truy cập
3	LEVEL	NUMBER		Cấp độ nhân viên

Bảng 3: mô tả bảng DANGNHAP

- Bảng HANGMAYBAY

STT	Thuộc tính	Kiểu dữ liệu	null	Mô tả
1	MAHMB	VARCHAR2(10)	x	Mã hãng máy bay
2	TENHMB	NVARCHAR2(50)		Tên hãng máy bay

Bảng 4: Mô tả bảng HANGMAYBAY

- Bảng SANBAY

STT	Thuộc tính	Kiểu dữ liệu	null	Mô tả
1	MASB	NVARCHAR2(50)	x	Mã sân bay
2	TENSB	NVARCHAR2(50)		Tên sân bay
3	THANHPHO	NVARCHAR2(20)		Tên thành phố

Bảng 5: Mô tả bảng SANBAY

- Bảng CHUYENBAY

STT	Thuộc tính	Kiểu dữ liệu	null	Mô tả
1	MACB	VARCHAR2(10)	x	Mã chuyến bay
2	MAHMB	VARCHAR2(10)	x	Mã hãng máy bay
3	DIEMKHOIHANH	VARCHAR2(10)		Điểm khởi hành / điểm đi
4	DIEMDEN	VARCHAR2(10)		Điểm đến / điểm dừng cuối cùng
5	SOGHEVIPTRONG	NUMBER		Số ghế vip còn trống trên chuyến bay
6	SOGHETHUONGTRONG	NUMBER		Số ghế thường còn trống trên chuyến bay
7	THOIGIANKHOIHANH	TIMESTAMP (6)		Ngày khởi hành
8	THOIGIANBAY	NUMBER		Thời gian bay
9	GIAVE	NUMBER		Giá vé gốc loại thường

Bảng 6: Mô tả bảng CHUYENBAY

- Bảng KHACHHANG

STT	Thuộc tính	Kiểu dữ liệu	null	Mô tả
1	MAKHACHHANG	VARCHAR2(9)	x	Mã khách hàng
2	TENKH	NVARCHAR2(20)	x	Tên khách hàng
3	EMAIL	NVARCHAR2(20)		Email khách hàng
4	TUOI	NUMBER		Tuổi khách hàng
5	DIACHI	NVARCHAR2(50)		Địa chỉ
6	LOAIHK	NUMBER		Loại khách hàng (trẻ em, người lớn)
7	CMND	VARCHAR2(20)		CMND

Bảng 7: Mô tả bảng KHACHHANG

- Bảng VE

STT	Thuộc tính	Kiểu dữ liệu	null	Mô tả
1	MAVE	NVARCHAR2(10)	x	Mã vé
2	MACB	NVARCHAR2(10)	x	Mã chuyến bay
3	LOAIVE	NUMBER		Loại vé (thường, vip)
4	TINHTRANG	NUMBER		Tình trạng (còn trống hoặc không)
5	TILE	FLOAT(126)		Tỉ lệ giá vé
6	SOGHE	VARCHAR2(20)		Số ghế tương ứng với vé

Bảng 8: Mô tả bảng VE

- Bảng CHITIETVE

STT	Thuộc tính	Kiểu dữ liệu	null	Mô tả
1	MAVE	VARCHAR2(10)	x	Mã vé
2	MACB	VARCHAR2(10)	x	Mã chuyến bay
3	MAKHACHHANG	VARCHAR2(9)	x	Mã khách hàng
4	GIA	NUMBER		Giá vé cuối cùng
5	KHUYENMAI	NVARCHAR2(10)	x	Mã khuyến mãi

Bảng 9: Mô tả bảng CHITIETVE

- Bảng KHUYENMAI

STT	Thuộc tính	Kiểu dữ liệu	null	Mô tả
1	MAKM	NVARCHAR2(10)	x	Mã khuyến mãi
2	TENKM	NVARCHAR2(20)		Tên khuyến mãi
3	MOTA	NVARCHAR2(50)		Mô tả
4	GIAMGIA	BINARY_DOUBLE		Số tiền giảm
5	SOLUONGDISCOUNT	NUMBER		Số lần giảm giá
6	HANSĐ	DATE		Hạn sử dụng
7	TINHTRANG	NUMBER		Tình trạng (hợp lệ hay không)

8	NGAYPHATHANH	DATE		Ngày phát hành khuyến mãi
---	--------------	------	--	------------------------------

Bảng 10: Mô tả bảng KHUYENMAI

3.4. Ràng buộc toàn vẹn

STT	Nội dung RBTV
1	Khách hàng đặt vé phải trên 18 tuổi
2	Thời gian mỗi chuyến bay kéo dài ít nhất là 30 phút
3	Khách hàng loại “trẻ em” phải đảm bảo dưới 18 tuổi và nếu dưới 14 tuổi thì không có CMND
4	Một chuyến bay có điểm đến khác với điểm đi, ngày về phải lớn hơn ngày khởi hành

Bảng 11: Các ràng buộc toàn vẹn

Chương 4. XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC

4.1. Triggers

4.1.1. Cơ sở lý thuyết

Trigger có thể hiểu đơn giản là một thủ tục SQL được thực thi ở phía server khi có một sự kiện như Insert, Delete, hay Update. Tuy nhiên khác với Stored Procedure, Trigger hoàn toàn không có tham số. Chúng ta không thể gọi thực hiện trực tiếp Trigger bằng lệnh EXECUTE như Store Procedure hoặc bằng bất kỳ một lệnh nào khác. Thay vào đó Trigger sẽ được thực hiện một cách tự động. Trigger được lưu trữ và quản lý trong Server Database, được dùng trong trường hợp ta muốn kiểm tra các ràng buộc toàn vẹn trong Database.

```
1  CREATE [ OR REPLACE ] TRIGGER trigger_name
2  AFTER/BEFORE INSERT OR UPDATE OR DELETE
3  ON table_name
4  [ FOR EACH ROW ]
5
6  DECLARE
7      -- variable declarations
8
9  BEGIN
10     -- trigger code
11
12  EXCEPTION
13      WHEN ...
14      -- exception handling
15  END;
```

Hình 2: Cấu trúc của trigger trong Oracle

4.1.2. Danh sách các trigger

STT	Tên trigger	Kích hoạt	Mô tả
1	UD_GHE	Sau khi update trên bảng VE	Cập nhật tình trạng ghế sau khi khách hàng hủy vé

2	KT_VE	Trước khi insert hoặc update trên bảng CHITIETVE	Kiểm tra và tính giá vé
3	GHE_CB	Sau khi insert trên bảng VE	Cập nhật số lượng ghế sau khi thêm vé
4	CK_TG_CB	Sau khi insert trên bảng chuyển bay	Kiểm tra RBTV : Thời gian bay tối thiểu là 30 phút
5	CK_TUOI_KH	Sau khi inseret hoặc update trên bảng KHACHHANG	Kiểm tra RBTV : khách hàng đặt vé phải trên 18 tuổi

Bảng 12: Danh sách trigger

4.1.3. Mô tả một số trigger

Tên Trigger: KT_VE

Thao tác: Insert or Update

Trên bảng: CHITIETVE

Nội dung: Khi thêm mới hoặc cập nhật bảng CHITIETVE thì cập nhật giá vé nếu tình trạng vé khác 0.

Code:

```
CREATE OR REPLACE FUNCTION UP_GIA (V_MAVE CHITIETVE.MAVE%TYPE)
RETURN CHITIETVE.GIA%TYPE
IS
    V_GIA CHITIETVE.GIA%TYPE;
BEGIN
    SELECT CB.GIAVE * ((100 - KM.GIAMGIA) / 100) * (VE.TYLE / 100)
    NUMBERO V_GIA
    FROM CHITIETVE CT, VE, CHUYENBAY CB, KHUYENMAI KM
    WHERE (CT.MAVE = V_MAVE) AND
           (CT.MAVE = VE.MAVE) AND
           (VE.MACB = CB.MACB) AND
           (CT.MAKM = KM.MAKM);

    RETURN V_GIA;
```

```

END;
CREATE OR REPLACE TRIGGER KT_VE
BEFORE INSERT OR UPDATE
ON CHITIETVE
FOR EACH ROW
DECLARE
  V_TINHTRANG VE.TINHTRANG%TYPE;
BEGIN
  SELECT VE.TINHTRANG FROM CHITIETVE CT JOIN VE ON CT.MAVE = VE.MAVE
  NUMBERO V_TINHTRANG
  WHERE CT.MAVE = :NEW.MAVE;

  IF (V_TINHTRANG != 0)
  THEN
    BEGIN
      UPDATE CHITIETVE CT
      SET CT.GIA = UP_GIA(:NEW.MAVE)
      WHERE CT.MAVE = :NEW.MAVE;

      DBMS_OUTPUT.PUT('THANH CONG!');
    END;
  ELSE
    BEGIN
      DBMS_OUTPUT.PUT('LOI: TINH TRANG VE = 0');
      ROLLBACK;
    END;
  END IF;
END;

```

Thực hiện:

[B1] Kiểm tra tình trạng vé.

[1.1] Nếu tình trạng vé khác 0, sang bước 2.

[1.2] Ngược lại, bỏ qua.

[B2] Cập nhật giá vé, sử dụng hàm UP_GIA().

4.2. Stored procedures

4.2.1. Cơ sở lý thuyết

Procedure hay còn gọi là thủ tục, nó dùng để gom một nhóm lệnh SQL cùng xử lý một mục đích cụ thể nào đó, sau đó đặt cho nó một cái tên và khai báo tham số truyền vào để sau này mỗi khi cần sử dụng thì ta chỉ cần gọi tên và truyền tham số là được.

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

```
1 CREATE [OR REPLACE] PROCEDURE procedure_name
2   [ (parameter [,parameter]) ]
3 IS
4   [declaration_section]
5 BEGIN
6   executable_section
7 [EXCEPTION
8   exception_section]
9 END [procedure_name];
```

Hình 3: Cấu trúc của stored procedure trong Oracle

4.2.2. Danh sách các stored procedure

STT	Tên procedure	Tham số	Mô tả
1	danhsachchuyenbay	Điểm đi, điểm đến, số vé, loại vé, ngày khởi hành	Truy vấn danh sách chuyến bay
2	danhsachchuyenbaykh	Điểm đi, điểm đến, hãng máy bay, số vé, loại vé, ngày khởi hành	Truy vấn danh sách chuyến bay khứ hồi

3	capnhattienve	Mã vé, số tiền mới	Cập nhật số tiền mới cho một mã vé cụ thể
4	danhsachchuyenbay	Điểm đi, điểm đến, số vé, loại vé, ngày khởi hành	Truy vấn danh sách chuyến bay (tính năng tìm chuyến bay trong chức năng đặt vé)
5	danhsachchuyenbaykh	Điểm đi, điểm đến, số vé, loại vé, ngày khởi hành, hãng máy bay	Truy vấn danh sách chuyến bay khứ hồi
6	danhsachvecua1cb	Mã chuyến bay	Truy vấn danh sách vé của 1 chuyến bay cụ thể
7	doanhthutheocb	Mã chuyến bay	Tính tổng doanh thu của một chuyến bay cụ thể
8	doanhthutheonam	Năm bao nhiêu	Tính tổng doanh thu của một năm cụ thể
9	huyve	Mã vé	Hủy đặt vé
10	insert_khuyen_mai	Tên khuyến mãi, mô tả, giảm giá, hạn sử dụng, s1, tình trạng	Tạo mã khuyến mãi mới
11	khachhangphothong		
12	Khachhangvip		
13	laychuyenbaybangid	Mã chuyến bay	Truy vấn chuyến bay
14	Soluongvecuahangmbtheongay	Ngày, mã hãng máy bay	Truy vấn số lượng vé của một hãng máy bay trong ngày cụ thể
15	suachuyenbay	Mã chuyến bay, mã hãng máy bay, điểm đi, điểm đến, ngày khởi hành, thời gian bay, giá vé	Cập nhật thông tin cho chuyến bay

16	suagiakhuyenmai	Mã khuyến mãi, giảm giá	Cập nhật thông tin cho khuyến mãi
17	themchuyenbay	Mã chuyến bay, mã hãng máy bay, điểm đi, điểm đến, ngày khởi hành, thời gian bay, giá vé	Thêm chuyến bay mới
18	Thongtinkhachhang motchuyenbay	Mã chuyến bay	Truy vấn thông tin của tất cả hành khách trong một chuyến bay cụ thể

Bảng 13: Danh sách stored procedure

4.2.3. Mô tả một số stored procedure

- Mua vé:

Tên: insert_ve()

Nội dung: Thực hiện khi khách hàng mua vé.

Tham số đầu vào: p_macb in VARCHAR2, p_sove in number, p_makh in varchar2, p_loaive in number, p_km in NVARCHAR22

Tham số đầu ra: không có.

Script:

```

procedure insert_ve(p_macb in VARCHAR2,
                   p_sove in number,
                   p_makh in varchar2,
                   p_loaive in number,
                   p_km in NVARCHAR22)
as
    p_loaihk number;
    v_dem NUMBEReger := 0;
    v_thanhtien number;
    type arr_type IS VARRAY (5) OF Ve.mave%
    type;
    mave_arr arr_type := arr_type();
    cursor c1 is select ve.mave
                   from ve
                   where MACB = p_macb
                   and LOAIVE = p_loaive

```



```

        and TINHTRANG = 0;
        counter NUMBEReger := 0;
begin
    select LOAIHK NUMBERo p_loaihk from KHACHHANG where CMND = p_makh;

    for n in c1
        loop
            counter := counter + 1;
            mave_arr.extend;
            mave_arr(counter) := n.MAVE;
            v_thanhtien := fn_thanhtien(p_macb, p_loaihk, p_km, mave_arr(counter));
--            DBMS_OUTPUT.put_line(v_thanhtien);
            insert NUMBERo CHITIETVE(MAVE, MACB, CMND, GIA, KHUYENMAI)
            values (mave_arr(counter), p_macb, p_makh, v_thanhtien, p_km);

            update ve
            set tinhtrang = 1
            where mave = mave_arr(counter);
        end loop;
    commit;
exception
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.put_line('trung khoa chinh');
    WHEN OTHERS THEN
        DBMS_OUTPUT.put_line('loi gi do');
        ROLLBACK;
end insert_ve;

```

Bước thực hiện:

1. Lấy loại khách hàng của khách hàng.
2. Thực hiện lặp.
3. Thêm mã vé vào mảng vé đã được khai báo trước đó.
4. Gọi hàm tính tiền để tính tổng tiền vé.
5. Thêm những vé trong mảng vào bảng CHITIETVE.
6. Cập nhật tình trạng vé thành 1(đã đặt) tương ứng.

- Thêm chuyến bay

Tên: themchuyenbay()

Nội dung: Tạo chuyến bay vô hệ thống.

Tham số đầu vào: p_macb varchar2, p_mahmb varchar2, p_diemkh NVARCHAR22, p_diemden NVARCHAR22, p_ngaykh timestamp, p_tgbay number, p_giave number

Tham số đầu ra: không có.

Script:

```
create or replace procedure themchuyenbay
(
  p_macb varchar2,
  p_mahmb varchar2,
  p_diemkh NVARCHAR22,
  p_diemden NVARCHAR22,
  p_ngaykh timestamp,
  p_tgbay number,
  p_giave number
) as
  v_dem number := 50;
  v_mave NVARCHAR22(10);
  v_soghe varchar2(2) := 0;
  ex_notfound exception ;
begin
  insert NUMBERo CHUYENBAY(macb, mahmb, diemkhoiinh, diemden, ngaykhoiinh,
thoigianbay, giave)
    values(p_macb,p_mahmb,p_diemkh,p_diemden,p_ngaykh,p_tgbay,p_giave);
  while v_dem >= 1
  loop
    v_mave := p_macb||v_dem;
    if v_dem <= 10 then
      insert NUMBERo VE(mave, macb, loaive, tinhtrang, tile,SOGHE) values (v_mave,
p_macb,1,0,2,v_soghe);
    end if;
    if v_dem > 10 then
      insert NUMBERo VE(mave, macb, loaive, tinhtrang, tile,SOGHE) values (v_mave,
p_macb,0,0,1,v_soghe);
    end if;
    v_soghe := v_soghe + 1;
    v_dem := v_dem - 1;
  end loop;
  commit;
exception
  when others then
    DBMS_OUTPUT.put_line('1 khong them chuyen bay duoc');
    raise_application_error(-20343,'khong them chuyen bay duoc',true);
    rollback ;
end;
```

Bước thực hiện:

1. Thêm 1 chuyến bay mới vào bảng CHUYENBAY.
2. Thực hiện vòng lặp.
3. Gán mã vé là mã chuyến bay và số thứ tự.
4. 10 vé đầu tiên sẽ được thêm vào bảng VE với loại vé là vé VIP, còn lại là loại phổ thông.

4.3. Functions

4.3.1. Cơ sở lý thuyết

Function là một kiểu chương trình con, nó bao gồm một nhóm các lệnh PL/SQL. Mỗi thủ tục trong Oracle có tên riêng của nó mà nó có thể được gọi. Đơn vị chương trình con này được lưu trữ như là một đối tượng cơ sở dữ liệu.

Các hàm sẽ giúp đơn giản hóa chương trình và có thể sử dụng nhiều lần. Ví dụ bạn cần viết một loạt các câu lệnh SQL phức tạp lên đến hàng ngàn dòng thì có thể phân tích và chia nhỏ thành nhiều hàm. Nếu nói đến gom nhóm và sử dụng nhiều lần thì stored procedure cũng làm được, tuy nhiên procedure sẽ không có giá trị trả về, còn function thì sẽ có giá trị trả về.

```
1  CREATE [OR REPLACE] FUNCTION function_name
2      [ (parameter [,parameter]) ]
3  RETURN return_datatype
4  IS | AS
5      [declaration_section]
6  BEGIN
7      executable_section
8  [EXCEPTION
9      exception_section]
10 END [function_name];
```

Hình 4: Cấu trúc của function trong Oracle

4.3.2. Danh sách các function

STT	Tên function	Tham số	Mô tả
1	soluongvecon	Mã chuyến bay, loại vé	Trả về số lượng vé còn lại của một chuyến bay

Bảng 14: Danh sách function

4.3.3. Mô tả một số function

- Tên: soluongvecon()
- Nội dung: kiểm tra số lượng vé chưa được mua của một chuyến bay cụ thể
- Tham số đầu vào: mã chuyến bay, loại vé
- Kiểu dữ liệu trả về: NUMBER
- Code SQL:

```
-- function tra ve so luong ve con cua mot chuyen bay
create or replace function soluongvecon
(
    p_loaive in int,
    p_macb in varchar
)
return int
as
    v_sove int;
begin
    select count(*) into v_sove from Ve where MaCB = p_macb and
        TinhTrang = 0 and LoaiVe = p_loaive ;
    return v_sove;

EXCEPTION
WHEN OTHERS THEN
    raise_application_error(-20001,'An error was encountered
        - '||SQLCODE||' -ERROR- '||SQLERRM);
end;
```

- Thực hiện:

Truy vấn số lượng vé trong bảng vé với mã chuyến bay và loại vé trùng với tham số đầu vào và tình trạng vé = 0 (tức chưa đặt)

4.4. Tối ưu hóa một số câu truy vấn

- Trigger GHE_CB

Câu truy vấn ban đầu:

```
-- trigger GHE_CB
-- original query:
SELECT CB.SOGHEVIPTRONG FROM CHUYENBAY CB, VE
WHERE VE.MAVE=:NEW.MAVE AND VE.MACB=CB.MACB;
```

Câu truy vấn sau khi tối ưu hóa:

```
-- trigger GHE_CB
-- optimized query:
select SOGHEVIPTRONG from(
    (select SOGHEVIPTRONG, macb from CHUYENBAY CB)
    inner join
    (select MACB, MAVE from VE where MAVE=:new.MAVE)
    on cb.MACB = ve.MACB
)
```

- Procedure danh sach vecua1cb

Câu truy vấn ban đầu:

```
-- procedure danh sach vecua1cb
-- original query:
select ct.MAVE,ct.MACB,ct.CMND, ct.GIA, kh.TENKH, ve.LOAIVE ,ve.SOGHE
from CHITIETVE ct, KHACHHANG kh, VE
where ct.MAVE = ve.MAVE and kh.CMND = ct.CMND
and ct.MACB = p_macb;
```

Câu truy vấn sau khi tối ưu hóa:

```
-- procedure danhsachvecua1cb
-- optimized query:
select ct.MAVE,ct.MACB,ct.CMND, ct.GIA, kh.TENKH, ve.LOAIVE ,ve.SOGHE
from CHITIETVE ct, KHACHHANG kh, VE
where ct.MAVE = ve.MAVE and kh.CMND = ct.CMND and ct.MACB = p_macb;
select CTV.MAVE, CTV.macb, CTV.makh, gia, TENKH, LOAIVE, SOGHE from(
    (select MAVE, makh, macb, gia from CHITIETVE CTV where macb = p_macb)
    inner join
    (select TENKH, makh from KHACHHANG KH)
    on CTV.makh = kh.makh
    inner join
    (select LOAIVE, SOGHE, MAVE from VE)
    on CTV.MAVE = VE.MAVE
)
```

- Procedure soluongvecuahangmbtheongay

Câu truy vấn ban đầu:

```
-- procedure soluongvecuahangmbtheongay
-- original query:
select cb.DIEMKHOIHANH, cb.DIEMDEN, count(ct.MAVE) as sl
from CHITIETVE ct, CHUYENBAY cb, HANGMAYBAY hmb
where cb.MAHMB = hmb.MAHMB
    and ct.MACB = cb.MACB
    and hmb.TENHMB = 'VietnamAirline'
    and cb.NGAYKHOIHANH = p_ngay
group by cb.DIEMKHOIHANH, cb.DIEMDEN;
```

Câu truy vấn sau khi tối ưu hóa:

```
-- procedure soluongvecuahangmbtheongay
-- optimized query:
select CB.DIEMKHOIHANH, cb.DIEMDEN, count(CTV.MAVE) as sl from(
    (select CTV.MACB, CTV.MAVE from CHITIETVE CTV)
    inner join
    ( select CB.MACB, CB.MAHMB, CN.DIEMKHOIHANH, CB.DIEMDEN
      from CHUYENBAY CB where CB.NGAYKHOIHANH = p_ngay)
    on CTV.MACB = CB.MACB
    inner join
    (select HMB.MAHMB from HANGMAYBAY HMB where HMB.TENHMB = "VietnamAirline")
    on CB.MAHMB = HMB.MAHMB
)
group by cb.DIEMKHOIHANH, cb.DIEMDEN;
```

Chương 5. XỬ LÝ TRUY XUẤT ĐỒNG THỜI

5.1. Các mức cô lập

5.1.1. Read Uncommitted

Khi transaction thực hiện ở mức này, các truy vấn vẫn có thể truy nhập vào các bản ghi đang được cập nhật bởi một transaction khác và nhận được dữ liệu tại thời điểm đó mặc dù dữ liệu đó chưa được commit (uncommitted data). Nếu vì lý do nào đó transaction ban đầu rollback lại những cập nhật, dữ liệu sẽ trở lại giá trị cũ. Khi đó transaction thứ hai nhận được dữ liệu sai.

- Đặc điểm:
 - Không thiết lập Shared Lock trên những đơn vị dữ liệu cần đọc. Do đó không phải chờ khi đọc dữ liệu (kể cả khi dữ liệu đang bị lock bởi giao tác khác)
 - Vẫn tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác
- Ưu điểm
 - Tốc độ xử lý rất nhanh
 - Không cản trở những giao tác khác thực hiện việc cập nhật dữ liệu
- Nhược điểm
 - Có khả năng xảy ra mọi vấn đề khi xử lý đồng thời : Dirty Reads, Unrepeatable Reads, Phantoms, Lost Updates

5.1.2. Read Committed

Đây là mức isolation mặc định, nếu bạn không đặt gì cả thì transaction sẽ hoạt động ở mức này. Transaction sẽ không đọc được dữ liệu đang được cập nhật mà phải đợi đến khi việc cập nhật thực hiện xong. Vì thế nó tránh được dirty read như ở mức trên.

- Đặc điểm

Đây là mức độ cô lập mặc định của SQL Server

- Tạo Shared Lock trên đơn vị dữ liệu được đọc, Shared Lock được giải phóng ngay sau khi đọc xong dữ liệu

- Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác
- Ưu điểm
 - Giải quyết vấn đề Dirty Reads.
 - Shared Lock được giải phóng ngay, không cần phải giữ cho đến hết giao tác nên không cản trở nhiều đến thao tác cập nhật của các giao tác khác.
- Nhược điểm
 - Chưa giải quyết được vấn đề Unrepeatable Reads, Phantoms, Lost Updates
 - Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock)

5.1.3. Repeatable Read

Mức isolation này hoạt động như mức read commit nhưng nâng thêm một bậc nữa bằng cách ngăn không cho transaction ghi vào dữ liệu đang được đọc bởi một transaction khác cho đến khi transaction khác đó hoàn tất.

- Đặc điểm
 - Tạo Shared Lock trên đơn vị dữ liệu được đọc và giữ shared lock này đến hết giao tác => Các giao tác khác phải chờ đến khi giao tác này kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này .
(Repeatable Read = Read Committed + Giải quyết Unrepeatable Reads).
 - Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác.
- Ưu điểm
 - Giải quyết vấn đề Dirty Reads và Unrepeatable Reads
- Nhược điểm
 - Chưa giải quyết được vấn đề Phantoms, do vẫn cho phép insert những dòng dữ liệu thỏa điều kiện thiết lập shared lock
 - Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock)
 - Shared lock được giữ đến hết giao tác ==> cản trở việc cập nhật dữ liệu của các giao tác khác

5.1.4. Serializable

Mức isolation này tăng thêm một cấp nữa và khóa toàn bộ dải các bản ghi có thể bị ảnh hưởng bởi một transaction khác, dù là UPDATE/DELETE bản ghi đã có hay INSERT bản ghi mới.

- Đặc điểm
 - Tạo Shared Lock trên đơn vị dữ liệu được đọc và giữ shared lock này đến hết giao tác => Các giao tác khác phải chờ đến khi giao tác này kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này.
 - Không cho phép Insert những dòng dữ liệu thỏa mãn điều kiện thiết lập Shared Lock (sử dụng Key Range Lock) ==> Serializable = Repeatable Read + Giải quyết Phantoms
 - Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác.
- Ưu điểm
 - Giải quyết thêm được vấn đề Phantoms
- Nhược điểm
 - Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock)
 - Cản trở nhiều đến việc cập nhật dữ liệu của các giao tác khác

5.2. Lost update

- Tình huống: 2 transaction T1 và T2 cùng hủy vé có mã chuyến bay '31'.

T1	T2	Giải thích
SELECT MACB, SUM(MAVE) AS SOLUONGVE FROM VE WHERE MACB='31';		T1 đếm số vé của chuyến bay '31', có tổng cộng 50 vé.

VE									
MACB	SOLUONGVE								
31	50								
DELETE FROM VE WHERE (MACB='31') AND (MAVE='1367');			T1 xóa 1 vé của chuyến bay '31'.						
		DELETE FROM VE WHERE (MACB='31') AND (MAVE IN ('1351','1352','1353'));	T2 xóa 3 vé của chuyến bay '31'.						
		SELECT MACB, SUM(MAVE) AS SOLUONGVE FROM VE WHERE MACB='31'; <table><tr><td colspan="2">VE</td></tr><tr><td>MACB</td><td>SOLUONGVE</td></tr><tr><td>31</td><td>47</td></tr></table>	VE		MACB	SOLUONGVE	31	47	T2 đếm số vé của chuyến bay '31', có tổng cộng 47 vé. Mất dữ liệu cập nhật của T1.
VE									
MACB	SOLUONGVE								
31	47								
SELECT MACB, SUM(MAVE) AS SOLUONGVE FROM VE WHERE MACB='31'; <table><tr><td colspan="2">VE</td></tr><tr><td>MACB</td><td>SOLUONGVE</td></tr><tr><td>31</td><td>49</td></tr></table>		VE		MACB	SOLUONGVE	31	49		T1 đếm số vé của chuyến bay '31', có tổng cộng 49 vé. Mất dữ liệu cập nhật của T2.
VE									
MACB	SOLUONGVE								
31	49								

- Nguyên nhân: T1 và T2 đều hủy vé của cùng một chuyến bay nhưng vẫn chưa được cập nhật cho nhau biết nên khi đếm lại số vé của chuyến bay thì ra kết

quả cập nhật khác nhau. Transaction này làm mất dữ liệu cập nhật của transaction kia. Đây là Lost update.

- Hướng giải quyết: Dùng mức cô lập Serializable. Khi T1 thực hiện hủy vé của chuyến bay '31' sẽ tạo Share lock và giữ Share lock này đến hết giao tác. T2 muốn thực hiện hủy vé của chuyến bay này phải đợi cho đến khi T1 thực hiện xong.
- Kết quả: Không bị mất dữ liệu cập nhật, dữ liệu đúng.

T1	T2	Giải thích						
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;	T1 và T2 đặt mức cô lập Serializable.						
SELECT MACB, SUM(MAVE) AS SOLUONGVE FROM VE WHERE MACB='31'; <table><tr><th colspan="2">VE</th></tr><tr><th>MACB</th><th>SOLUONGVE</th></tr><tr><td>31</td><td>50</td></tr></table>	VE		MACB	SOLUONGVE	31	50		T1 đếm số vé của chuyến bay ‘31’, có tổng cộng 50 vé.
VE								
MACB	SOLUONGVE							
31	50							
DELETE FROM VE WHERE (MACB='31') AND (MAVE='1367');		T1 xóa 1 vé của chuyến bay ‘31’.						
	DELETE FROM VE WHERE (MACB='31') AND (MAVE IN (‘1351’,‘1352’, ‘1353’));	T2 xóa 3 vé của chuyến bay ‘31’.						
SELECT MACB, SUM(MAVE) AS	SELECT MACB, SUM(MAVE) AS	T1 và T2 thực hiện đếm lại số vé của						

SOLUONGVE FROM VE WHERE MACB='31';	SOLUONGVE FROM VE WHERE MACB='31';	chuyến bay '31'. Kết quả giống nhau, dữ liệu đúng.												
<table><tr><th colspan="2">VE</th></tr><tr><th>MACB</th><th>SOLUONGVE</th></tr><tr><td>31</td><td>46</td></tr></table>	VE		MACB	SOLUONGVE	31	46	<table><tr><th colspan="2">VE</th></tr><tr><th>MACB</th><th>SOLUONGVE</th></tr><tr><td>31</td><td>46</td></tr></table>	VE		MACB	SOLUONGVE	31	46	
VE														
MACB	SOLUONGVE													
31	46													
VE														
MACB	SOLUONGVE													
31	46													

5.3. Dirty read

- Tình huống:

T1	T2	Giải thích										
<div>SELECT MaCB, GiaVe FROM ChuyenBay WHERE MaCB IN (‘100’, ‘200’, ‘300’);</div> <div><table><tr><th colspan="2">ChuyenBay</th></tr><tr><th>MaCB</th><th>GiaVe</th></tr><tr><td>100</td><td>1200000</td></tr><tr><td>200</td><td>1000000</td></tr></table></div>	ChuyenBay		MaCB	GiaVe	100	1200000	200	1000000		T1 thực hiện câu truy vấn cho chuyến bay ‘100’, ‘200’, ‘300’. Không tìm thấy chuyến bay ‘300’.		
ChuyenBay												
MaCB	GiaVe											
100	1200000											
200	1000000											
<div>INSERT NUMBERO ChuyenBay (MaCB, GiaVe) VALUES (300, 1300000);</div>		T1 thêm 1 dòng dữ liệu vào bảng ChuyenBay, nhưng chưa commit.										
	<div>SELECT MaCB, GiaVe FROM ChuyenBay WHERE MaCB IN (‘100’, ‘200’, ‘300’);</div> <div><table><tr><th colspan="2">ChuyenBay</th></tr><tr><th>MaCB</th><th>GiaVe</th></tr><tr><td>100</td><td>1200000</td></tr><tr><td>200</td><td>1000000</td></tr><tr><td>300</td><td>1300000</td></tr></table></div>	ChuyenBay		MaCB	GiaVe	100	1200000	200	1000000	300	1300000	T2 thực hiện câu truy vấn cho chuyến bay ‘100’, ‘200’, ‘300’.
ChuyenBay												
MaCB	GiaVe											
100	1200000											
200	1000000											
300	1300000											

	UPDATE ChuyenBay SET GiaVe=990000 WHERE MaCB='100';	T2 cập nhật giá vé của chuyến bay '100'.								
ROLLBACK;		T1 hủy giao tác.								
	COMMIT;	T2 commit dữ liệu.								
	SELECT MaCB, GiaVe FROM ChuyenBay WHERE MaCB IN ('100', '200', '300'); <table><tr><th colspan="2">ChuyenBay</th></tr><tr><th>MaCB</th><th>GiaVe</th></tr><tr><td>100</td><td>990000</td></tr><tr><td>200</td><td>1000000</td></tr></table>	ChuyenBay		MaCB	GiaVe	100	990000	200	1000000	T2 thực hiện lại truy vấn cho chuyến bay '100', '200', '300', nhưng không tìm thấy '300'. Dữ liệu ban đầu T2 đọc chưa được commit mà bị hủy bỏ nên ra kết quả dữ liệu không chính xác.
ChuyenBay										
MaCB	GiaVe									
100	990000									
200	1000000									

- Nguyên nhân: Khi T1 cập nhật dữ liệu, T2 có thể đọc được dữ liệu vừa cập nhật của T1, tuy nhiên dữ liệu đó chưa được T1 commit (chế độ Read Uncommitted). Sau đó T1 hủy giao tác (rollback) nên dữ liệu T2 đọc được là vô nghĩa (dữ liệu rác). -> Xảy ra hiện tượng Dirty Read.
- Hướng giải quyết: Sử dụng mức cô lập Read Committed, Repeatable Read hoặc Serializable để giải quyết vấn đề này.
- Kết quả:

T1	T2	Giải thích								
<div>SELECT MaCB, GiaVe FROM ChuyenBay WHERE MaCB IN (‘100’, ‘200’, ‘300’);</div> <div><table><tr><th colspan="2">ChuyenBay</th></tr><tr><th>MaCB</th><th>GiaVe</th></tr><tr><td>100</td><td>1200000</td></tr><tr><td>200</td><td>1000000</td></tr></table></div>	ChuyenBay		MaCB	GiaVe	100	1200000	200	1000000		T1 thực hiện câu truy vấn cho chuyến bay ‘100’, ‘200’, ‘300’. Không tìm thấy chuyến bay ‘300’.
ChuyenBay										
MaCB	GiaVe									
100	1200000									
200	1000000									
<div>INSERT NUMBERO ChuyenBay (MaCB, GiaVe) VALUES (300, 1300000);</div>		T1 thêm 1 dòng dữ liệu vào bảng ChuyenBay, nhưng chưa commit.								

	<div>SET TRANSACTION ISOLATION LEVEL READ COMMITTED; --SET TRANSACTION ISOLATION LEVEL REPEATABLE READ/ SERIALIZABLE;</div> <div>SELECT MaCB, GiaVe FROM ChuyenBay WHERE MaCB IN (‘100’, ‘200’, ‘300’);</div> <table><tr><th colspan="2">ChuyenBay</th></tr><tr><th>MaCB</th><th>GiaVe</th></tr><tr><td>100</td><td>1200000</td></tr><tr><td>200</td><td>1000000</td></tr></table>	ChuyenBay		MaCB	GiaVe	100	1200000	200	1000000	<div>T2 đặt mức cô lập Read Committed (hoặc Repeatable Read hoặc Serializable).</div> <div>T2 thực hiện câu truy vấn cho chuyến bay ‘100’, ‘200’, ‘300’, không tìm thấy chuyến bay ‘300’ vì dữ liệu chưa được T1 commit.</div>
ChuyenBay										
MaCB	GiaVe									
100	1200000									
200	1000000									

- Tuy nhiên, Oracle sử dụng cơ chế Read Committed làm cơ chế mặc định (default) nên đã giải quyết được vấn đề Dirty Read.

5.4. Unrepeatable read

- Tình huống: Khi thay đổi hãng máy bay của một chuyến bay. Ban đầu T1 đọc thông tin chuyến bay, sau đó delay. T2 vào thay đổi thông tin chuyến bay, sau đó commit lên CSDL. T1 đọc lại dữ liệu nhưng không thể nhìn thấy dữ liệu ban đầu.

```

1  commit;
2  set transaction isolation level READ COMMITTED ;
3  declare
4      p_macb varchar2(10) := 1;
5      p_ngkh timestamp;
6      p_tgbay number;
7      p_giave number;
8      p_hmb varchar2(20);
9  begin
10     SELECT  cb.MaCB,cb.NGAYKHOIHANH,cb.ThoiGianBay,cb.GiaVe,hmb.T
11     into p_macb, p_ngkh, p_tgbay,p_giave,p_hmb
12     from ChuyenBay cb, HangMayBay hmb
13     where cb.MaHMB = hmb.MaHMB
14     and cb.MACB = p_macb;
15     dbms_output.put_line('MaCB: ' || p_macb
16     || ', NGÀYKHOIHANH:' || p_ngkh
17     || ', ThoiGianBay:' || p_tgbay
18     || ', GiaVe:' || p_giave
19     || ', TenHMB:' || p_hmb );
20
21     DBMS_LOCK.SLEEP(10);
22
23     SELECT  cb.MaCB,cb.NGAYKHOIHANH,cb.ThoiGianBay,cb.GiaVe,hmb.T
24     into p_macb, p_ngkh, p_tgbay,p_giave,p_hmb
25     from ChuyenBay cb, HangMayBay hmb
26     where cb.MaHMB = hmb.MaHMB
27     and cb.MACB = p_macb;
28     dbms_output.put_line('MaCB: ' || p_macb
29     || ', NGÀYKHOIHANH:' || p_ngkh
30     || ', ThoiGianBay:' || p_tgbay
31     || ', GiaVe:' || p_giave
32     || ', TenHMB:' || p_hmb );
33     COMMIT ;
34 end;

```

```

1  commit;
2  set transaction isolation level READ COMMITTED ;
3  declare
4      p_macb varchar2(10) := 1;
5      p_ngkh timestamp;
6      p_tgbay number;
7      p_giave number;
8      p_hmb varchar2(20);
9  begin
10     SELECT  cb.MaCB,cb.NGAYKHOIHANH,cb.ThoiGianBay,cb.GiaVe,hmb.T
11     into p_macb, p_ngkh, p_tgbay,p_giave,p_hmb
12     from ChuyenBay cb, HangMayBay hmb
13     where cb.MaHMB = hmb.MaHMB
14     and cb.MACB = p_macb;
15     dbms_output.put_line('MaCB: ' || p_macb
16     || ', NGÀYKHOIHANH:' || p_ngkh
17     || ', ThoiGianBay:' || p_tgbay
18     || ', GiaVe:' || p_giave
19     || ', TenHMB:' || p_hmb );
20
21     update CHUYENBAY
22     set MAHMB = 2
23     where MACB = p_macb;
24     COMMIT;
25 end;

```

```

[2019-06-12 18:03:24] completed in 10 s 26 ms
[2019-06-12 18:03:24] MaCB: 1, NGÀYKHOIHANH:15-THÁNG SÁU    -19 12.00.00.000000 SÁNG, ThoiGianBay:1, GiaVe:200, TenHMB:VietnamAirline
[2019-06-12 18:03:24] MaCB: 1, NGÀYKHOIHANH:15-THÁNG SÁU    -19 12.00.00.000000 SÁNG, ThoiGianBay:1, GiaVe:200, TenHMB:VietnamAirline

```

- Nguyên nhân: T1 thiết lập mức cô lập read committed nên mỗi câu lệnh SELECT trong cùng 1 thao tác sẽ thiết lập snapshot của chính nó dù những câu lệnh này đọc trên những đơn vị dữ liệu giống nhau.
- Hướng giải quyết : Dùng mức cô lập Serializable. Khi chạy dòng lệnh select sẽ tạo Share lock và giữ Share lock này đến hết giao tác. Khi T2 vào update thông tin. T2 phải đợi T1 thực hiện xong mới update. → Dữ liệu đúng.
- Kết quả:

```

1  commit;
2  set transaction isolation level serializable ;
3  declare
4      p_macb varchar2(10) := 1;
5      p_ngkh timestamp;
6      p_tgbay number;
7      p_giave number;
8      p_hmb varchar2(20);
9  begin
10     SELECT  cb.MaCB,cb.NGAYKHOIHANH,cb.ThoiGianBay,cb.GiaVe,hmb.T
11     into p_macb, p_ngkh, p_tgbay,p_giave,p_hmb
12     from ChuyenBay cb, HangMayBay hmb
13     where cb.MaHMB = hmb.MaHMB
14     and cb.MACB = p_macb;
15     dbms_output.put_line('MaCB: ' || p_macb
16     || ', NGÀYKHOIHANH:' || p_ngkh
17     || ', ThoiGianBay:' || p_tgbay
18     || ', GiaVe:' || p_giave
19     || ', TenHMB:' || p_hmb );
20
21     DBMS_LOCK.SLEEP(10);
22
23     SELECT  cb.MaCB,cb.NGAYKHOIHANH,cb.ThoiGianBay,cb.GiaVe,hmb.T
24     into p_macb, p_ngkh, p_tgbay,p_giave,p_hmb
25     from ChuyenBay cb, HangMayBay hmb
26     where cb.MaHMB = hmb.MaHMB
27     and cb.MACB = p_macb;
28     dbms_output.put_line('MaCB: ' || p_macb
29     || ', NGÀYKHOIHANH:' || p_ngkh
30     || ', ThoiGianBay:' || p_tgbay
31     || ', GiaVe:' || p_giave
32     || ', TenHMB:' || p_hmb );
33     COMMIT ;
34 end;

```

```

1  commit;
2  set transaction isolation level serializable ;
3  declare
4      p_macb varchar2(10) := 1;
5      p_ngkh timestamp;
6      p_tgbay number;
7      p_giave number;
8      p_hmb varchar2(20);
9  begin
10     SELECT  cb.MaCB,cb.NGAYKHOIHANH,cb.ThoiGianBay,cb.GiaVe,hmb.T
11     into p_macb, p_ngkh, p_tgbay,p_giave,p_hmb
12     from ChuyenBay cb, HangMayBay hmb
13     where cb.MaHMB = hmb.MaHMB
14     and cb.MACB = p_macb;
15     dbms_output.put_line('MaCB: ' || p_macb
16     || ', NGÀYKHOIHANH:' || p_ngkh
17     || ', ThoiGianBay:' || p_tgbay
18     || ', GiaVe:' || p_giave
19     || ', TenHMB:' || p_hmb );
20
21     update CHUYENBAY
22     set MAHMB = 2
23     where MACB = p_macb;
24     COMMIT;
25 end;

```

[2019-06-12 17:48:29] completed in 10 s 53 ms

[2019-06-12 17:48:29] MaCB: 1, NGÀYKHOIHANH:15-THÁNG SÁU

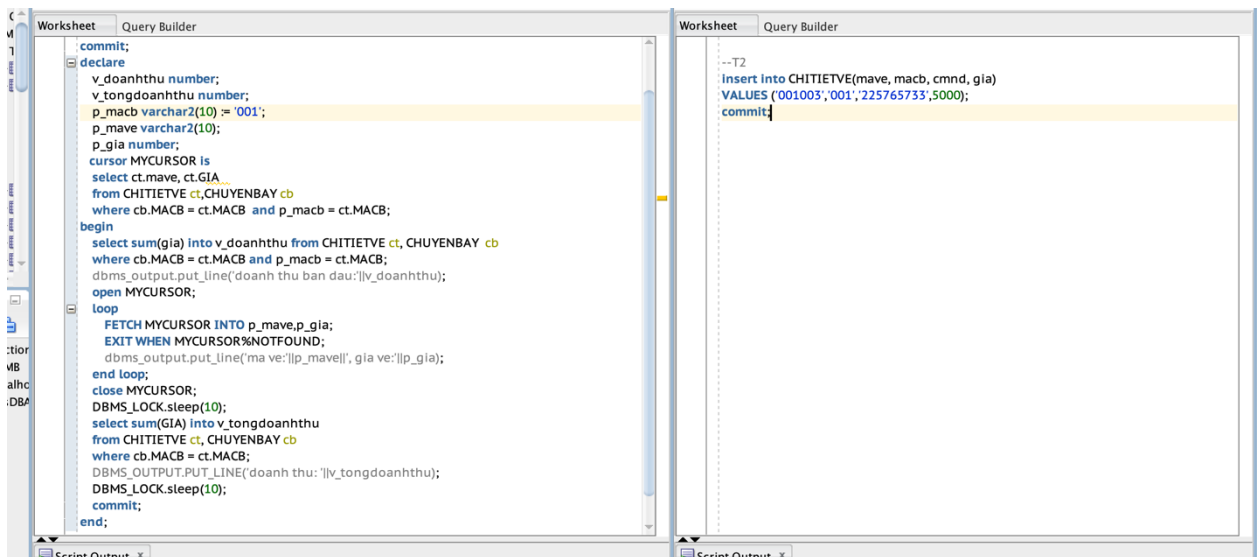
-19 12.00.00.000000 SÁNG, ThoigianBay:1, GiaVe:200, TenHMB:VietnamAirline

[2019-06-12 17:48:29] MaCB: 1, NGÀYKHOIHANH:15-THÁNG SÁU

-19 12.00.00.000000 SÁNG, ThoigianBay:1, GiaVe:200, TenHMB:Jetstar Pacific

5.5. Phantom

- Tình huống: Khi một nhân viên đang xem báo cáo bán vé của chuyến bay số “001” thì tại thời điểm đó một nhân viên khác thực hiện bán vé.



- Vấn đề xảy ra: T1 thực hiện hiển thị danh sách vé của chuyến bay số “001” được bán ra. Sau đó T2 vào thực hiện bán vé. Sau đó T1 thực hiện tính tổng doanh thu vé được bán. Khi đó tổng doanh thu sẽ tính cả doanh thu của vé vừa bán trong khi trên danh sách hiển thị thì không có vé vừa bán ở T2.
- Nguyên nhân: T1 thiết lập mức cô lập read committed nên khi tính tổng doanh thu thì sẽ tính cả số tiền vừa thu được từ việc bán vé ở T2.
- Giải pháp: Chuyển mức cô lập của T1 thành Serializable. Khi thiết lập mức cô lập này thì T2 phải chờ đến khi T1 thực hiện xong rồi mới thực hiện.
- Kết quả: Tổng doanh thu sẽ không tính số tiền vừa thu được của việc bán thức ăn ở T2. Vì lúc này T1 dùng dữ liệu trước khi T2 và thực hiện để tính tổng doanh thu.

5.6. Deadlock

- Tình huống: Ba transactions T1, T2 và T3 đồng thời hủy vé bằng cách cập nhật tình trạng vé bằng 0 và đặt một vé khác bằng cách cập nhật tình trạng vé đó bằng 1. Tuy nhiên ba transactions này đã gây deadlocks khi đặt vé vì chúng giữ khóa của nhau khi hủy vé và phải chờ nhau vô thời hạn để được nhả khóa.

```

commit;
set transaction isolation level Serializable;
begin
  UPDATE ve
  SET TINHTRANG = 0
  WHERE MAVE = 1418;

  DBMS_LOCK.SLEEP(10);

  UPDATE ve
  SET TINHTRANG = 1
  WHERE MAVE = 1420;
  ROLLBACK;
end;

```

```

commit;
set transaction isolation level Serializable;
begin
  UPDATE ve
  SET TINHTRANG = 0
  WHERE MAVE = 1418;

  DBMS_LOCK.SLEEP(10);

  UPDATE ve
  SET TINHTRANG = 1
  WHERE MAVE = 1419;
  ROLLBACK;
end;

```

```

commit;
set transaction isolation level Serializable;
begin
  UPDATE ve
  SET TINHTRANG = 0
  WHERE MAVE = 1420;

  DBMS_LOCK.SLEEP(10);

  UPDATE ve
  SET TINHTRANG = 1
  WHERE MAVE = 1418;
  ROLLBACK;
end;

```

T1	T2	T3	Giải thích
UPDATE ve SET TINHTRANG = 0 WHERE MAVE = 1418;			T1 cập nhật tình trạng vé '1418' bằng 0 thành công
	UPDATE ve SET TINHTRANG = 0 WHERE MAVE = 1419;		T2 cập nhật tình trạng vé '1419' bằng 0 thành công
		UPDATE ve SET TINHTRANG = 0 WHERE MAVE = 1420;	T2 cập nhật tình trạng vé '1420' bằng 0 thành công
UPDATE ve			T1 cập nhật tình trạng vé '1419'

SET TINHTRANG = 1 WHERE MAVE = 1419;			bằng 1. Lúc này T2 đang giữ khóa của vé '1419' nên T1 phải chờ T2 giải phóng khóa.
	UPDATE ve SET TINHTRANG = 1 WHERE MAVE = 1420;		T2 cập nhật tình trạng vé '1420' bằng 0. Lúc này T3 đang giữ khóa của vé '1420' nên T2 phải chờ T3 giải phóng khóa.
		UPDATE ve SET TINHTRANG = 1 WHERE MAVE = 1418;	T3 cập nhật tình trạng vé '1418' bằng 1. Lúc này T1 đang giữ khóa của vé '1418' nên T3 phải chờ T1 giải phóng khóa.

- Phát hiện deadlock:

1. Oracle sẽ tự động thông báo có deadlocks tại transaction T2:

```
Error report -
ORA-00060: deadlock detected while waiting for resource
ORA-06512: at line 8
00060. 00000 - "deadlock detected while waiting for resource"
*Cause:   Transactions deadlocked one another while waiting for resources.
*Action:  Look at the trace file to see the transactions and resources
          involved. Retry if necessary.
```

2. Xem được đồ thị chờ, câu lệnh SQL đã gây thông báo deadlocks (câu lệnh update ở transaction T2) và các thông tin khác về deadlocks tại trace file.

```
orcl_ora_1640 - Notepad
File Edit Format View Help
*** 2019-07-10T17:31:08.568684+07:00 (CDB$ROOT(1))
*** SESSION ID:(258.19796) 2019-07-10T17:31:08.569629+07:00
*** CLIENT ID:() 2019-07-10T17:31:08.569629+07:00
*** SERVICE NAME:(orcl) 2019-07-10T17:31:08.570621+07:00
*** MODULE NAME:(SQL Developer) 2019-07-10T17:31:08.570621+07:00
*** ACTION NAME:() 2019-07-10T17:31:08.570621+07:00
*** CLIENT DRIVER:(jdbcthin : 12.2.0.1.0) 2019-07-10T17:31:08.570621+07:00
*** CONTAINER ID:(1) 2019-07-10T17:31:08.570621+07:00

2019-07-10 17:31:08.556*:ksq.c@12954:ksqldl_hdr_dump():
DEADLOCK DETECTED ( ORA-00060 )
See Note 60.1 at My Oracle Support for Troubleshooting ORA-60 Errors

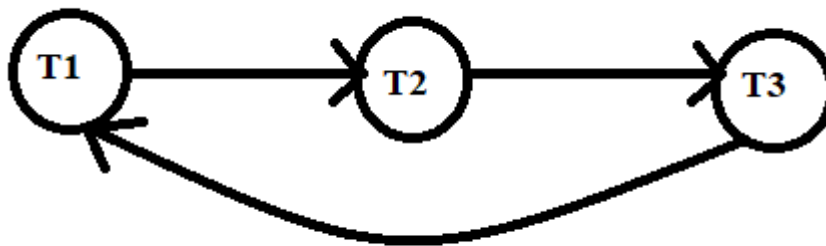
[Transaction Deadlock]

The following deadlock is not an ORACLE error. It is a
deadlock due to user error in the design of an application
or from issuing incorrect ad-hoc SQL. The following
information may aid in determining the deadlock:

Deadlock graph:
-----Blocker(s)-----
Resource Name      process session holds waits serial
TX-00010021-00001FF8-00000001-00000000    22      258      X      19796
TX-00090005-00001F03-00000001-00000000    43      387      X      2992
TX-00080014-00001EFF-00000001-00000000    24         6      X      63264
-----Waiter(s)-----
process session holds waits serial
43      387      X      2992
24         6      X      63264
22      258      X      19796

*** 2019-07-10T17:31:08.571617+07:00 (CDB$ROOT(1))
dbkedDefDump(): Starting a non-incident diagnostic dump (flags=0x0, level=1, mask=0x0)
----- Error Stack Dump -----
----- Current SQL Statement for this session (sql_id=g5f9p5fhs4r4a) -----
UPDATE VE SET TINHTRANG = 0 WHERE MAVE = 1420
----- PL/SQL Stack -----
----- PL/SQL Call Stack -----
```

Ta có đồ thị chờ như sau:



3. Ta có thể giải quyết deadlocks này bằng cách rollback transaction đã gây thông báo deadlocks (transaction T2).
- Kết quả: Lúc này cơ sở dữ liệu sẽ quay lại trạng thái ban đầu lúc chưa thực thi T1, T2 và T3.

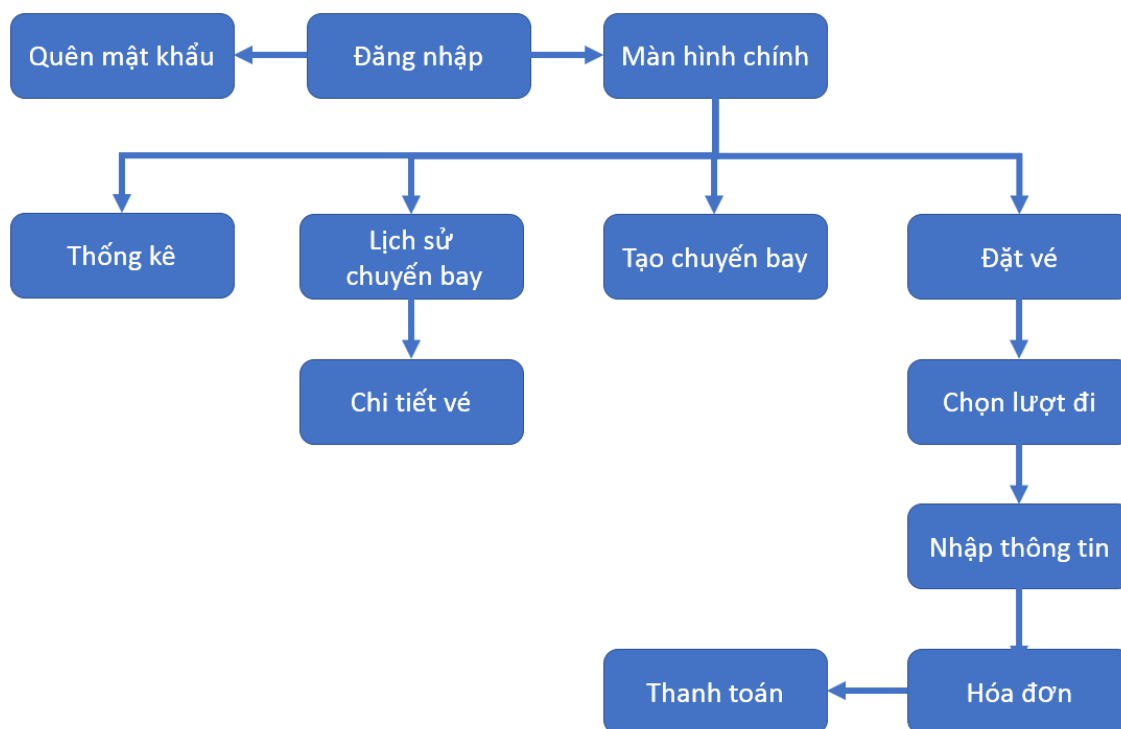
Chương 6. THIẾT KẾ GIAO DIỆN

6.1. Danh sách các màn hình

STT	Tên màn hình	Mô tả chức năng
1	Đăng nhập	Đăng nhập vào màn hình chính
2	Quên mật khẩu	Lấy lại mật khẩu cho tài khoản
3	Man hình chính	Màn hình chính để truy cập các tính năng chính của phần mềm
4	Đặt vé	Nhập thông tin để tìm kiếm chuyến bay
5	Chọn lượt đi	Chọn chuyến bay phù hợp
6	Nhập thông tin	Nhập thông tin từng khách hàng
7	Hóa đơn	Xuất hóa đơn sau khi xác định được chuyến bay và thông tin khách hàng
8	Thanh toán	In ra thông tin thanh toán
9	Tạo chuyến bay	Dùng cho nhân viên tạo chuyến bay mới
10	Thống kê	Thống kê theo từng năm, tháng,...
11	Lịch sử chuyến bay	Nhập mã chuyến bay để xem lịch sử chuyến bay
12	Chi tiết vé	Hiện thông tin chi tiết vé đã đặt

Bảng 15: Danh sách các màn hình

6.2. Sơ đồ liên kết màn hình

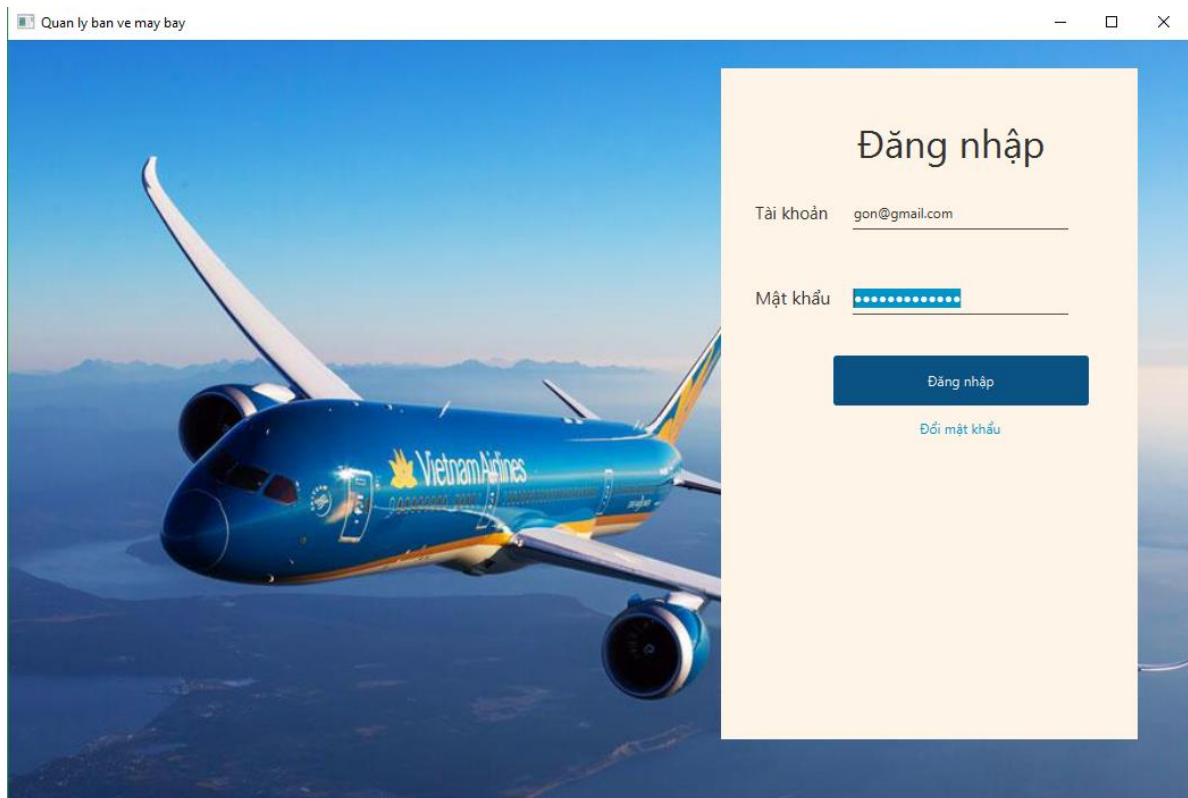


Hình 5: Sơ đồ liên kết màn hình

6.3. Mô tả chức năng từng màn hình

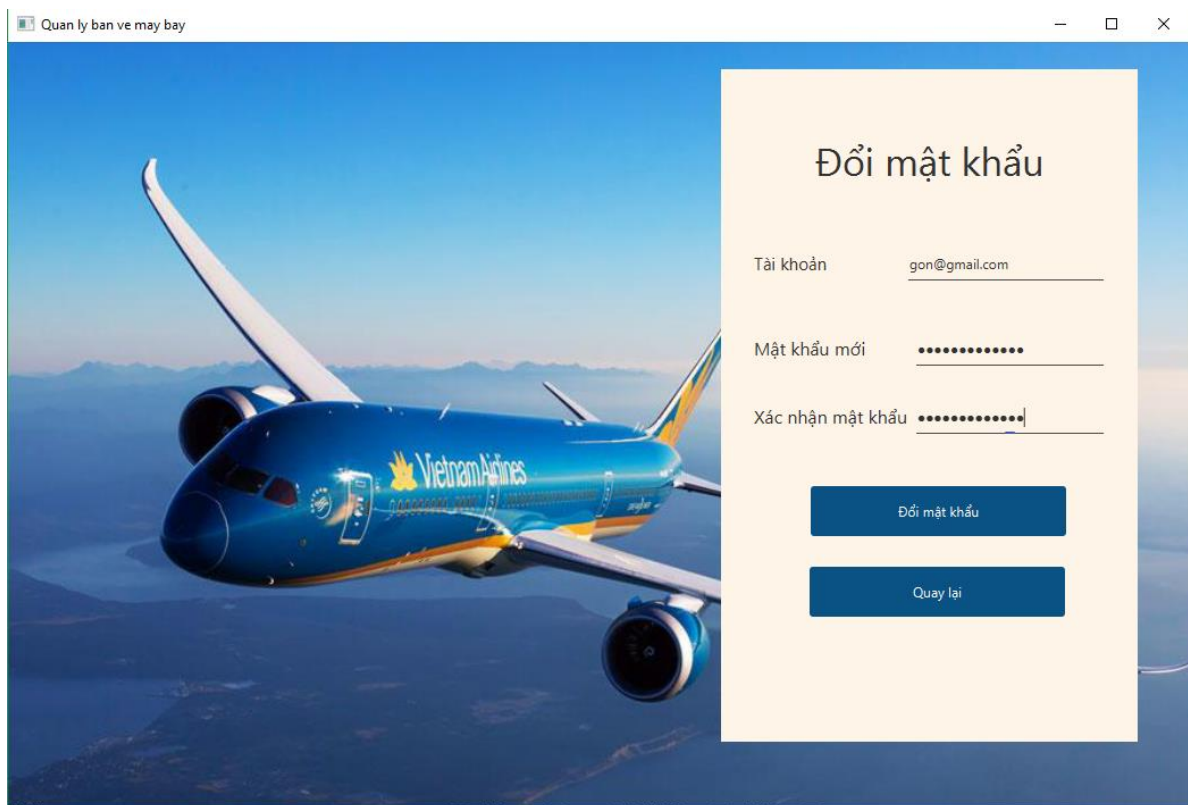
- Màn hình đăng nhập

Chức năng: Đăng nhập bằng tài khoản nhân viên. Bấm nút đăng nhập để vào màn hình chính, click vào “Đổi mật khẩu” nếu nhân viên quên mật khẩu.



- Màn hình đổi mật khẩu

Chức năng: đổi mật khẩu nếu nhân viên quên mật khẩu. Bấm vào nút đổi mật khẩu để thực hiện, bấm nút quay lại để trở về màn hình đăng nhập.



- Màn hình chính

Chức năng: trên màn hình có các tùy chọn Đặt vé, Tạo chuyến bay, Thống kê, Lịch sử chuyến bay, tương ứng với các chức năng của phần mềm.

- Màn hình tạo chuyển bay

Chức năng: tạo chuyến bay mới, tìm kiếm và chỉnh sửa các chuyến bay đã tạo. nhập mã chuyến bay và nhấn nút tìm kiếm để tìm kiếm chuyến bay, nhấn nút thêm sau khi điền thông tin để tạo chuyến bay mới, ấn nút xóa để xóa chuyến bay đã tạo.

[illegible]

- Màn hình tìm kiếm chuyển bay

Chức năng: nhập đầy đủ các thông tin chuyến bay mình cần và ấn nút Tìm kiếm, phần mềm sẽ truy vấn ra các chuyến bay phù hợp để ta lựa chọn. Sau khi tìm kiếm sẽ chuyển tới màn hình chọn chuyến bay.

Tìm chuyến bay

Điểm khởi hành:

Ho Chi Minh

Điểm đến:

Bình Định

Ngày đi:

6/25/2019

Ngày về:

Số hành khách:

1

1

Tìm kiếm

- Màn hình đặt vé

Chức năng: nhập thông tin từng khách hàng sau khi đã chọn chuyến bay. Nếu ở màn hình tìm kiếm chuyến bay, người dùng nhập số lượng khách hàng nhiều hơn 1, thì ở bước này ta nhập lần lượt thông tin từng khách hàng. Sau mỗi khách hàng, nhấn nút Tiếp theo để nhập thông tin khách hàng kế tiếp. Bảng thông tin vé kế bên cho thấy ghế nào đã được đặt.

Chi tiết vé

[<- Quay lại trang trước](#)

Thông tin vé:
Mã vé: 1
Mã chuyến bay: 1
Hãng máy bay: Jetstar
Điểm khởi hành: TP.HCM
Điểm đến: Bình Định
Sân bay đi: Sân bay Tân Sơn Nhất
Sân bay đến: Sân bay Phú Cát
Ngày khởi hành: 6/13/2019
Thời gian bay: 0 tiếng
Giờ khởi hành: 8:03 PM

Thông tin hành khách:
Tên hành khách: Tín
CMND: 1
Email: gon@gmail.com
Loại HK: Trẻ em
Loại vé: Thương gia
Giá vé: 1000000 đồng

[In vé](#) [Hủy Vé](#) [Sửa](#)

- Màn hình lịch sử bán vé

Chức năng: xem lịch sử bán vé của một chuyến bay cụ thể. Nhập mã chuyến bay vào và nhấn Tìm để tìm kiếm, nhấn Xem tất cả để xem tất cả vé. Nhấn vào Xem chi tiết để chuyển tới màn hình chi tiết vé.



- Màn hình hóa đơn

Hiển thị thông tin chi tiết hóa đơn vừa đặt, bấm vào Tiếp theo để chuyển đến màn hình thanh toán

Chương 7. TỔNG KẾT

7.1. Kết quả đạt được

Trong học kỳ vừa qua, nhóm đã tìm hiểu vận dụng kiến thức về hệ quản trị cơ sở dữ liệu và đã đạt được một số kết quả như sau.

- Nắm vững kiến thức và có thể vận dụng, cài đặt stored procedure, trigger, function, transaction trong nhiều hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle.
- Trang bị thêm kiến thức về các chế độ khóa và mức cô lập trong Oracle.
- Giải quyết được các trường hợp xử lý đồng thời trên hệ quản trị trong môi trường giả lập.
- Xây dựng được phần mềm “Quản lý rạp vé máy bay” với các chức năng:
 - Tạo chuyến bay
 - Đặt vé
 - Tra cứu vé
 - Thống kê
 - Khuyến mãi

Các chức năng hoạt động như ý muốn và có thể hỗ trợ nhiều người dùng đồng thời mà không gặp vấn đề về xung đột dữ liệu.

7.2. Hạn chế

Do thời gian hạn chế, lượng công việc lớn, cùng với năng lực nhân lực vẫn chưa đủ kinh nghiệm, nhóm vẫn chưa thiết kế được một giao diện đẹp và mang lại trải nghiệm tốt nhất cho người dùng. Các câu lệnh P/L SQL vẫn chưa thực sự tối ưu.

Về vấn đề bảo mật, phần mềm cũng chưa có các cơ chế bảo mật tránh mối nguy hiểm như chèn mã độc vào câu lệnh SQL (SQL Injection). Hơn nữa mật khẩu của nhân viên vẫn lưu dưới dạng text, chưa được mã hóa tránh việc ăn cắp thông tin, cướp quyền truy cập tài khoản. Vấn đề sao lưu, phục hồi dữ liệu cũng chưa thể hiện được trong đồ án.

Ngoài ra, nhóm chưa xây dựng được mobile app và phiên bản web của hệ thống quản lý vé máy bay. Nếu làm được điều này thì hệ thống này sẽ tiếp cận tới nhiều người dùng hơn, và khiến nó trở nên tiện dụng hơn khi người dùng không cần phải có một chiếc máy tính thì mới có thể sử dụng được.

7.3. Hướng phát triển

Để khắc phục các nhược điểm kể trên, ta cần giải quyết các vấn đề sau:

- Tối ưu hóa câu lệnh P/L SQL
- Mã hóa mật khẩu nhân viên
- Dùng các kỹ thuật phòng tránh SQL Injection
- Thiết kế lại UI/UX
- Xây dựng tính năng sao lưu, phục hồi
- Xây dựng phiên bản trên thiết bị di động
- Xây dựng phiên bản web

BẢNG PHÂN CÔNG CÔNG VIỆC

Công việc	Hoàng	Tín	Linh	Pa
Tìm hiểu tổng quan đề tài	x	x	x	x
Phân tích yêu cầu	x	x	x	x
Thiết kế chức năng	x	x	x	x
Thiết kế CSDL	x	x	x	x
Xây dựng ràng buộc toàn vẹn	x		x	x
Xây dựng trigger			x	
Xây dựng stored procedure	x		x	
Xây dựng function	x		x	
Tối ưu hóa câu truy vấn				x
Các mức cô lập				x
Giải quyết lost update	x		x	
Giải quyết dirty read			x	
Giải quyết unrepeatable read	x			
Giải quyết phantom	x			
Giải quyết deadlock	x			
Thiết kế giao diện		x		
Xây dựng giao diện phần mềm		x		
Xây dựng chức năng phần mềm		x		
Lập trình báo biểu	x	x		
Kết nối CSDL		x		
Chuẩn bị dữ liệu ban đầu	x	x	x	x
Tổng kết				x
Tổng hợp và viết báo cáo				x

TÀI LIỆU THAM KHẢO

- Documentation của Oracle: <https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/data-concurrency-and-consistency.html>
- Tối ưu hóa câu truy vấn : <https://sqlvietnam.wordpress.com/2016/01/09/thuat-toi-uu-hoa-sql/>
- Isolation levels : <http://www.sqlviet.com/blog/cac-mc-isolation-level>