**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

— & œ–

Logo

Description automatically generated

**MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**GVHD:** *TS. Nguyễn Thành Sơn*

**ĐỀ TÀI:**

**QUẢN LÍ CHUYẾN BAY**

**SVTH:** Nguyễn Lê Huy 20110495

Nguyễn Văn Giang 20110463

Lã Quý Quân 20110551

Nguyễn Thành Đạt 20110121

**TP. Hồ Chí Minh, ngày tháng năm 2021**

**MỤC LỤC**

Contents

[**CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG** 3](#_Toc103553446)

[**1.** **Một hãng hàng không cần 2 chương trình quản lí chính:** 3](#_Toc103553447)

[**2.** **Phần mềm quản lí các thông tin sau :** 3](#_Toc103553448)

[**3.** **Các chức năng cần có của hệ thống:** 3](#_Toc103553449)

[**CHƯƠNG 2: THIẾT KẾ CƠ SỞ DỮ LIỆU** 4](#_Toc103553450)

[**1.** **Thiết kế cơ sở dữ liệu mức quan niệm: Xây dựng ERD** 4](#_Toc103553451)

[**2.** **Thiết kế cơ sở dữ liệu mức logic:** 5](#_Toc103553452)

[**3.** **Tạo cơ sở dữ liệu và các ràng buộc:** 7](#_Toc103553453)

[**CHƯƠNG 3: TÍNH NĂNG** 13](#_Toc103553454)

[**1.** **Thủ tục:** 13](#_Toc103553455)

[**2.** **Hàm:** 19](#_Toc103553456)

[**CHƯƠNG 4: tạo user + phân quyền ( USER+ ADMIN )** 20](#_Toc103553457)

[**1.** **Quản lí tài khỏan ⇒ admin** 20](#_Toc103553458)

[**2.** **Kiểm tra login + thay đổi password của user** 24](#_Toc103553459)

[**3.** **Trigger : kiểm tra đảm bảo các tài khoản không trùng nhau** 26](#_Toc103553460)

[**CHƯƠNG 5: Hướng dẫn sử dụng phần mềm** 26](#_Toc103553462)

# **CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG**

1. **Một hãng hàng không cần 2 chương trình quản lí chính:**
   1. Chương trình quản lí đơn hàng đối với khách hàng
   2. Chương trình quản lí chuyến bay phục vụ cho admin
2. **Phần mềm quản lí các thông tin sau :**

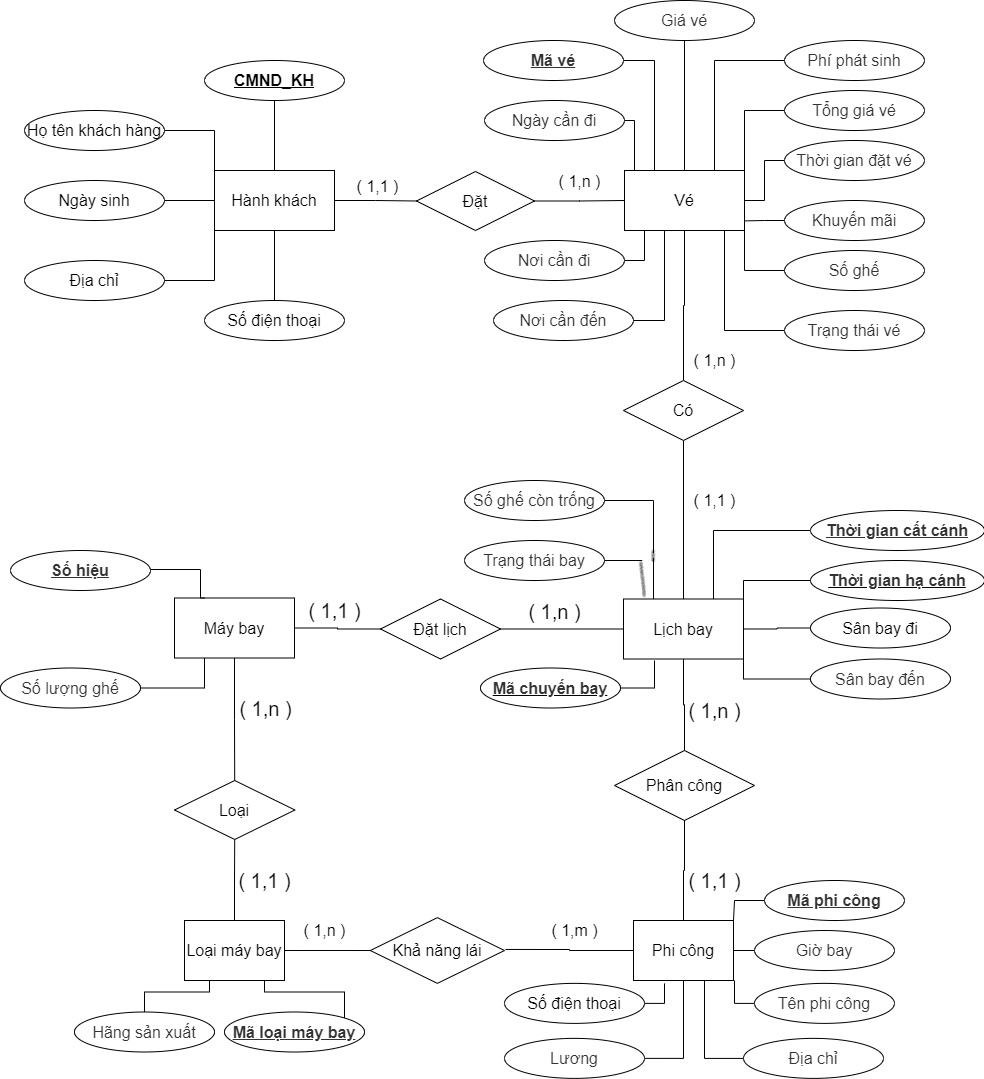
* Hành khách
  + Quản lí các thông tin sau liên quan đến khách hàng:
    - CMND\_KH, Họ và tên Khách Hàng , Ngày sinh, Địa chỉ , số điện thoại
    - Khách hàng chỉ được đặt 1 Vé trong 1 lần giao dịch và có thể đặt nhiều vé khác nhau trong từng lần giao dịch
* Vé
  + Quản lí các thông tin sau liên quan đến Vé:
    - Mã vé, giá vé, Phí phát sinh, tổng giá vé, Thời gian đặt vé,Khuyến Mãi, Số ghế, Ngày cần đi , Nơi cần đi , Nơi Cần đến
    - Mỗi lần giao dịch khách hàng chỉ được phép đặt 1 vé .
    - Một khách hàng có thể đặt nhiều vé trong các lần giao dịch khác nhau.
    - Mỗi vé chỉ cần lưu thông tin người đặt vé - không cần định danh thông tin của chủ sở hữu vé
    - Thời gian đặt vé , Phí phát sinh, Khuyến Mãi ,Trạng thái vé ,Tổng giá vé sẽ được hệ thống tự cập nhật
* Lịch bay
  + Quản lí các thông tin sau liên quan đến lịch bay:
    - Mã chuyến bay,Thời gian cất cánh, thời gian hạ cánh,Trạng thái bay, Số ghế còn trống, Sân bay đi, Sân bay đến.
    - Mỗi lịch bay sẽ có sự tham gia trực tiếp của 1 phi công
* Phi Công
  + Quản lí các thông tin sau liên quan đến Phi Công Mã PC, họ\_ten PC, Số điện thoại , địa chỉ ,mức lương, số giờ bay
    - Mỗi phi công sẽ tham gia bay duy nhất 1 lần trong ngày .
    - Mỗi phi công có thể có khả năng lái nhiều loại máy bay.
* Loại máy bay:
  + Quản lí các thông tin sau liên quan đến Loại máy bay : Mã Loại Máy Bay, Hãng sản xuất
    - Mỗi loại bay có thể được lái bởi nhiêu phi công .
* Máy Bay
  + Quản lí các thông tin sau liên quan đến Máy Bay: số Hiệu, Số lượng ghế.

1. **Các chức năng cần có của hệ thống:**

* Đặt vé :
  + Thêm / xóa/sửa bảng vé
  + Gợi ý lịch bay phù hợp dựa vào ( ngày muốn đi , nơi đi, nơi đến)
* Quản lí vé đã mua:
  + Quản lí chuyến bay dành cho admin:
    - Cơ bản:
      * Thêm / sửa / xóa HÀNH KHÁCH
      * Thêm / sửa / xóa VÉ
      * Thêm / sửa / xóa LỊCH BAY
      * Thêm / sửa / xóa MÁY BAY
      * Thêm / sửa / xóa PHI CÔNG
      * Thêm / sửa / xóa LOẠI MÁY BAY
      * Thêm / sửa / xóa KHẢ NĂNG
    - Thống kê:
      * *Thống kê thông tin phi công sắp xếp theo mức lương*
      * Thống kê : số lượng vé bán được mỗi tháng ( sắp xếp theo năm ⇒ tháng)
      * Thống kê : Phi công có tham gia nhiều chuyến bay trong mỗi tháng
      * Thống kê: Mỗi loại máy bay có bao nhiêu máy bay
    - Tìm kiếm :
      * Từ mã chuyến bay , thời gian cất cánh, thời gian hạ cánh ⇒ trả về ds khách hàng có trên chuyến bay
      * Tình trạng bay của các chuyến bay từ ngày x → y
    - Xử lí dữ liệu:
      * Kiểm tra có phải ngày cuối tháng ( có -1 , ko -0)
      * Tính thời gian còn hiệu lực sửa vé : nếu < 24h thì return 0 , nếu ≥24 return 1
      * Tính số ghế tối đa của mỗi chuyến bay dựa vào vé đã đặt.
      * Tính số ghế còn lại của mỗi chuyến bay dựa vào mã vé đã đặt.
    - Quản lí tài khoản:
      * Thêm / xóa / sửa user

# **CHƯƠNG 2: THIẾT KẾ CƠ SỞ DỮ LIỆU**

1. **Thiết kế cơ sở dữ liệu mức quan niệm: Xây dựng ERD**



**Diagram

Description automatically generated**

1. **Thiết kế cơ sở dữ liệu mức logic:**

Chuyển ERD sang mô hình quan hệ vì hệ thống sẽ được cài đặt trên hệ quản trị CSDL SQL Server

1. **HanhKhach** ( **CMND\_KH**, HoTen, NgaySinh, DiaChi, SoDienThoai)
2. **Ve** ( **MaVe**, GiaVe, NgayDi, NoiDi, NoiDen, PhiPhatSinh, TongGiaVe, ThoiGianDatVe, KhuyenMai, SoGhe, TrangThaiVe, **CMND\_KH**, **MaChuyenBay, ThoiGianCatCanh, ThoiGianHaCanh** )
3. **LichBay** ( **MaChuyenBay**, **ThoiGianCatCanh**, **ThoiGianHaCanh**, SanBayDi, SanBayDen, SoGheConTrong, TrangThaiBay, **SoHieu**, **MaPhiCong** )
4. **MayBay** ( **SoHieu**, SoLuongGhe, **MaLoaiMayBay** )
5. **PhiCong** ( **MaPhiCong**, HoTen, GioBay, DiaChi, Luong, SoDienThoai)
6. **LoaiMayBav**(**MaLoaiMayBay**,Hang san xuat)
7. **KhaNang** ( **MaPhiCong**, **MaLoaiMayBay**)

Mô tả ý nghĩa các quan hệ trên và các ràng buộc:

|  |  |  |
| --- | --- | --- |
| STT | Tên Table | Mục đích |
| 1 | Hành khách | Mỗi khách hàng có một mã khách hàng để phân biệt với những người khác, có tên, ngày sinh, địa chỉ và số điện thoại để liên lạc.  -Khóa chính là : **CMND\_KH** |
| 2 | Vé | -Vé thể hiện các thông tin sau :    Mã Vé dùng để phân biệt các vé khác nhau, Gía Vé,Ngày đi ,Ngày đến,Phí phát sinh, Tổng Gía Vé , Thời gian Đặt Vé,Khuyến mãi ,Số Ghế, Trạng Thái Vé  -Khóa chinh: **MaVe**  -Khóa ngoại:  +**MaCB, ThoiGianCatCanh, ThoiGianHaCanh** : tham chiếu đế bảng **Lịch bay**  +**CMND\_HK** :  tham chiếu đế bảng **HÀNH KHÁCH** |
| 3 | Lịch bay | - Một lịch bay có một ***mã chuyến bay ,thời gian cất cánh , thời gian hạ cánh*** để phân biệt với các chuyến bay khác. Mỗi chuyến bay xuất phát từ một sân bay và hạ cánh tại một sân bay khác; khởi hành vào một giờ và hạ cánh vào một giờ khác.  -Khóa chính:  **MaCB, ThoiGianCatCanh, ThoiGianHaCanh**  -Khóa ngoại:  **SoHieu**  tham chiếu đến bảng **MAYBAY**  **MaPhiCong** tham chiếu đến bảng **PhiCong** |
| 4 | Máy Bay | -Máy bay thể hiện các thông tin sau : Số Hiệu ,Số lượng ghế  -Một máy bay có thể có nhiều lịch bay.  Khóa chính:  **SoHieu**  Khóa ngoại:  **MaLoaiMayBay**  tham chiếu đế bảng  **LOAIMAYBAY** |
| 5 | Loại máy bay | - Mỗi loại máy bay có một mã loại để phân biệt với các loại máy bay khác và do một hãng sản xuất.  -Khóa chính :**MaLoaiMayBay** |
| 6 | Phi công | -Mỗi Phi công của hãng hàng không có một mã phi công duy nhất để phân biệt với các Phi công khác,có  họ tên, lương, địa chỉ, điện thoại và Tổng giờ bay đã bay .  Khóa chính :**MaPC** |
| 7 | Khả năng | -Khóa chính:  **MaPhiCông,MaLoaiMayBay** |
| Các ràng buộc khác | | 1/ Khách hàng chỉ được đổi vé nếu thời gian cất cánh của chuyến bay -thời gian đặt vé >24h.  2/ Mỗi Phi công chỉ tham gia lái máy bay duy nhất một lần trong ngày.  3/Mỗi khách hàng trong 1 lần giao dịch chỉ được phép đặt 1 vé . Muốn đặt nhiều vé khác nhau khách hàng phải thực hiện đặt từng vé một.  4/  **Tổng giá vé** = giá vé ban đầu + khuyến mãi +phí phát sinh  ***với điều kiện :***   khuyến mãi : khuyến mãi +giá vé ban đầu\* 0.1( nếu cuối tháng)  ( khuyến mãi mặc định luôn =0)   Phí  phát sinh :Phí phát sinh+ giá vé ban đầu \*0.3 nếu đổi vé  ( Phí phát sinh luôn mặc định =0 , mỗi lần cập nhật phí phát sinh sẽ được cộng dồn vào phí phát sinh hiện tại ) |

1. **Tạo cơ sở dữ liệu và các ràng buộc:**
   * 1. **Code tạo bảng:**

CREATE DATABASE QUAN\_LI\_CHUYEN\_BAY

GO

USE QUAN\_LI\_CHUYEN\_BAY

GO

* **BẢNG HÀNH KHÁCH**

CREATE TABLE HANHKHACH

(

CMND\_HK varchar(10) ,

HotenHK varchar(50) ,

NgaySinhHK DATE,

DiachiHK varchar(50),

SdtHK varchar(11) ,

CONSTRAINT pk\_HANHKHACH PRIMARY KEY(CMND\_HK)

);

* **BẢNG PHI CÔNG**

CREATE TABLE PHICONG

(

MaPC varchar(10) ,

TenPC varchar(30) ,

DiaChiPC varchar(50),

SDT\_Phi\_cong varchar(11) ,

Luong\_Phi\_cong Float,

Gio\_bay float,

CONSTRAINT pk\_PHICONG PRIMARY KEY (MaPC)

);

* **BẢNG LOẠI MÁY BAY**

CREATE TABLE LOAIMAYBAY

(

MaLoaiMayBay varchar(10) ,

HangSanXuat varchar(50) ,

CONSTRAINT pk\_LOAIMAYBAY PRIMARY KEY (MaLoaiMayBay)

);

* **BẢNG KHẢ NĂNG**

CREATE TABLE KHANANG

(

MAPC varchar(10) ,

MaLoaiMayBay varchar(10) ,

CONSTRAINT pk\_KHANANG PRIMARY KEY(MaPC,MaLoaiMayBay),

CONSTRAINT fk\_KHANANG\_PHICONG FOREIGN KEY(MAPC)

REFERENCES PHICONG(MAPC),

CONSTRAINT fk\_KHANANG\_LOAIMAYBAY FOREIGN KEY(MaLoaiMayBay)

REFERENCES LOAIMAYBAY(MaLoaiMayBay)

);

* **BẢNG MÁY BAY**

CREATE TABLE MAYBAY

(

SoHieu Varchar(10),

MaLoaiMayBay varchar(10),

SoLuongGheToiDa INT ,

CONSTRAINT pk\_MAYBAY PRIMARY KEY(SoHieu),

CONSTRAINT fk\_MAYBAY\_LOAIMAYBAY FOREIGN KEY(MaLoaiMayBay)

REFERENCES LOAIMAYBAY(MaLoaiMayBay)

);

* **BẢNG LỊCH BAY**

CREATE TABLE LICHBAY

(

MaCB varchar(10) ,

SoHieu Varchar(10),

ThoiGianCatCanh DATETIME ,

ThoiGianHaCanh DATETIME ,

SanBayDi varchar(50) CHECK ( SanBayDi IN('HCM','Ha Noi','Da Nang')),

SanBayDen varchar(50)CHECK ( SanBayDen IN('HCM','Ha Noi','Da Nang')),

TrangThaiMayBay varchar(30) CHECK ( TrangThaiMayBay IN ('landing','flying','take off','delay','cancel',NULL)), /\* 4

Trang thai : cat canh , ha canh, dang bay,delay,cancel,NULL chua hoat dong \*/

So\_ghe\_con\_trong int,

MaPC varchar(10) ,

CONSTRAINT pk\_LICHBAY PRIMARY KEY(MaCB, ThoiGianCatCanh, ThoiGianHaCanh),

CONSTRAINT fK\_LICHBAY\_MAYBAY FOREIGN KEY(SoHieu )

REFERENCES MAYBAY (SoHieu),

CONSTRAINT fK\_LICHBAY\_PC FOREIGN KEY(MaPC)

REFERENCES PHICONG (MaPC)

);

* **BẢNG VÉ**

CREATE TABLE VE

(

MaVe int IDENTITY(1,1) ,

CMND\_HK varchar(10) ,

MaCB varchar(10) ,

ThoiGianDatVe DATETIME,

ThoiGianCatCanh DATETIME ,

ThoiGianHaCanh DATETIME ,

Ngay\_can\_bay DATE,/\* bs\*/

Noi\_Di varchar(50)CHECK ( Noi\_Di IN('HCM','Ha Noi','Da Nang')),

Noi\_Den varchar(50) CHECK ( Noi\_Den IN('HCM','Ha Noi','Da Nang')),

TrangThaiVe varchar(30) CHECK (TrangThaiVe IN('exchanged','Paid') ), /\* trang thai : da mua , doi tra\*/

SoGhe int,

GiaVe float check (Giave>0),

PhiPhatsinh float check(PhiPhatsinh>=0),

KhuyenMai float check(KhuyenMai <=0),/\* khuyen mai luon am \*/

TongGiaVe Float check (TongGiaVe >= 0),

CONSTRAINT pk\_VE PRIMARY KEY (MaVe),

CONSTRAINT fk\_VE\_LICHBAY FOREIGN KEY(MaCB, ThoiGianCatCanh, ThoiGianHaCanh)

REFERENCES LICHBAY(MaCB, ThoiGianCatCanh, ThoiGianHaCanh),

CONSTRAINT fk\_VE\_HANHKHACH FOREIGN KEY(CMND\_HK)

REFERENCES HANHKHACH(CMND\_HK)

);

* + 1. **Code trigger và transaction**
* **Trigger:**
  + **Tính giờ bay**

**\*THÊM:**

CREATE TRIGGER Tinh\_Gio\_Bay\_Insert

ON LICHBAY

FOR INSERT

AS

BEGIN

PRINT N'Tinh Gio Bay Insert';

UPDATE PHICONG

SET Gio\_bay = Gio\_bay + (

SELECT DATEDIFF(hour, ThoiGianCatCanh, ThoiGianHaCanh

FROM inserted

WHERE inserted.MaPC = PHICONG.MaPC)

FROM PHICONG

JOIN inserted ON PHICONG.MaPC = inserted.MaPC

END

GO

**\*Sửa**

CREATE TRIGGER Tinh\_Gio\_Bay\_Update

ON LICHBAY

FOR UPDATE

AS

BEGIN

PRINT N'Tinh Gio Bay Update';

UPDATE PHICONG

SET Gio\_bay = Gio\_bay -

( SELECT DATEDIFF(hour, ThoiGianCatCanh, ThoiGianHaCanh)

FROM deleted

WHERE deleted.MaPC = PHICONG.MaPC) +

( SELECT DATEDIFF(hour, ThoiGianCatCanh, ThoiGianHaCanh)

FROM inserted

WHERE inserted.MaPC = PHICONG.MaPC)

FROM PHICONG

JOIN inserted ON PHICONG.MaPC = inserted.MaPC;

END

GO

**\*Xóa**

CREATE TRIGGER Tinh\_Gio\_Bay\_Delete

ON LICHBAY

FOR DELETE

AS

BEGIN

PRINT N'Tinh Gio Bay Delete';

UPDATE PHICONG

SET Gio\_bay = Gio\_bay - (

SELECT DATEDIFF(hour, ThoiGianCatCanh, ThoiGianHaCanh)

FROM deleted

WHERE deleted.MaPC = PHICONG.MaPC)

FROM PHICONG

JOIN deleted ON PHICONG.MaPC = deleted.MaPC

END

GO

* + Tính lương của phi côngdựa trên số giờ bay ( 1h bay = .. $ ) + số giờ bay ( tổng giờ bay của tất cả chuyến bay có PC tham gia)

**\*Thêm**

CREATE TRIGGER Luong\_Giobay\_Bandau

ON PHICONG

AFTER INSERT

AS

BEGIN

DECLARE @MaPC varchar(10)

SELECT @MaPC = i.MaPC

FROM inserted i

IF EXISTS (SELECT \* FROM PHICONG WHERE MaPC = @MaPC)

BEGIN

UPDATE PHICONG

SET Luong\_Phi\_cong = 0,

Gio\_bay = 0

WHERE MaPC = @MaPC

PRINT N'Thanh cong'

END

ELSE

ROLLBACK TRANSACTION

END

GO

**\*Sửa**

CREATE TRIGGER Tinh\_Luong

ON PHICONG

AFTER UPDATE, DELETE

AS

BEGIN

PRINT N'Tinh Luong';

UPDATE PHICONG

SET Luong\_Phi\_cong = 1000 \* PHICONG.Gio\_bay

END

GO

* + Tính hợp lệ của vé( tổng hợp tất cả trigger liên quan vé) +update data vé ( giá vé +thời gian đặt vé ,..)+ trạng thái vé ( nếu insert => Paid , nếu update thì => “exchage (đã đổi trả )”

CREATE TRIGGER [dbo].[UpdateBangVe] on [dbo].[VE]

FOR INSERT, UPDATE

AS

BEGIN

DECLARE @Action as char(1);

SET @Action = 'I'; -- Set Action to Insert by default.

IF EXISTS(SELECT \* FROM DELETED)

BEGIN

SET @Action =

CASE

WHEN EXISTS(SELECT \* FROM INSERTED) THEN 'U' -- Set Action to Updated.

END

END

END

UPDATE VE SET VE.ThoiGianDatVe=GETDATE(), Ve.KhuyenMai = 0, Ve.PhiPhatsinh = 0, Ve.TrangThaiVe = 'Paid' FROM VE inner join inserted on VE.MaVe=inserted.MaVe

DECLARE @newVe INT, @ThoiGianCatCanh DATE, @CheckGhe INT, @CheckGio INT, @CheckTrangThaiVe varchar(30), @SoGheToiDa INT

SELECT @newVe = ne.MaVe, @CheckTrangThaiVe = ne.TrangThaiVe,

@ThoiGianCatCanh = ne.ThoiGianCatCanh, @CheckGhe = ne.SoGhe, @SoGheToiDa = MAYBAY.SoLuongGheToiDa

FROM (inserted ne inner join LICHBAY on ne.MaCB = LICHBAY.MaCB AND ne.ThoiGianCatCanh = LICHBAY.ThoiGianCatCanh AND ne.ThoiGianHaCanh = LICHBAY.ThoiGianHaCanh) inner join MAYBAY on LICHBAY.SoHieu = MAYBAY.SoHieu

IF dbo.NgayCuoiThang(@ThoiGianCatCanh) = 1

BEGIN

UPDATE VE

SET KhuyenMai = KhuyenMai - 0.1 \* GiaVe

WHERE VE.MaVe = @newVe

END

IF @Action = 'U' AND dbo.hieu\_luc\_doi\_ve(@newVe) = 1

BEGIN

UPDATE VE

SET TrangThaiVe = 'exchanged',

PhiPhatsinh = PhiPhatsinh + 0.3 \* GiaVe

WHERE VE.MaVe = @newVe

END

ELSE IF dbo.hieu\_luc\_doi\_ve(@newVe) = 0

BEGIN

print 'Ve da het hieu luc doi'

ROLLBACK TRANSACTION

END

select @SoGheToiDa=[dbo].[funtion\_ghe\_toi\_da] (@newVe)

IF @Action = 'I'

BEGIN

UPDATE VE

SET TrangThaiVe = 'Paid'

WHERE VE.MaVe = @newVe

END

IF ( SELECT COUNT(SoGhe) as SLGT FROM VE WHERE SoGhe = @CheckGhe ) > 1

OR (@CheckGhe > @SoGheToiDa )

BEGIN

print('Loi dat ghe')

ROLLBACK

END

UPDATE VE SET TongGiaVe = GiaVe + PhiPhatsinh + KhuyenMai WHERE VE.MaVe = @newVe

GO

* + Mỗi phi công chỉ bay 1 chuyến trong 1 ngày

CREATE TRIGGER [dbo].[trigger\_Phan\_cong\_PC] on [dbo].[LICHBAY]

AFTER INSERT , UPDATE

AS

DECLARE @sochuyenbay\_CB int

select @sochuyenbay\_CB =count(\*) from LICHBAY inner join inserted on

inserted.MaPC=LICHBAY.MaPC

where FORMAT(LICHBAY.ThoiGianCatCanh, 'yyyy-MM-dd') =FORMAT(inserted.ThoiGianCatCanh, 'yyyy-MM-dd') --Kiem tra PC co bao nhieu chuyen trong ngay

```

IF(@sochuyenbay\_CB >1 )

BEGIN

print'Pc da bay du chuyen trong ngay\n'

ROLLBACK TRANSACTION

END

GO

* + Tính số ghế còn lại của mõi chuyến bay

**\*Thêm /sửa**

CEARTE TRIGGER [dbo].[so\_ghe\_trong\_update\_inserted] ON [dbo].[VE]

AFTER INSERT

AS

DECLARE @Thoigiancatcanh datetime

DECLARE @thoigianhacanh datetime

DECLARE @ID\_ve int

DECLARE @MaCB varchar(10)

select @MaCB=inserted.MaCB,@Thoigiancatcanh=inserted.ThoiGianCatCanh,@thoigianhacanh=inserted.thoigianhacanh,@ID\_ve=inserted.MaVe from inserted

UPDATE LICHBAY SET

So\_ghe\_con\_trong= [dbo].funtion\_ghe\_toi\_da (@ID\_ve)-[dbo].funtion\_so\_ghe\_da\_dat(@ID\_ve)

where LICHBAY.MaCB=@MaCB

and DATEDIFF(MINUTE,ThoiGianCatCanh,@Thoigiancatcanh) =0

and DATEDIFF(MINUTE,ThoiGianHaCanh,@thoigianhacanh)=0

GO

**\*Xóa**

CREATE TRIGGER [dbo].[ghe\_trong\_delete] on [dbo].[VE]

AFTER DELETE

AS

DECLARE @Thoigiancatcanh datetime

DECLARE @thoigianhacanh datetime

DECLARE @MaCB varchar(10)

DECLARE @sohieu varchar(10)

DECLARE @so\_ghe\_trong int

DECLARE @so\_ghe\_toi\_da int

BEGIN

select @MaCB=deleted.MaCB,@Thoigiancatcanh=deleted.ThoiGianCatCanh,@thoigianhacanh=deleted.ThoiGianHaCanh

from deleted

UPDATE LICHBAY SET

LICHBAY.So\_ghe\_con\_trong=LICHBAY.So\_ghe\_con\_trong+1

WHERE LICHBAY.MaCB=@MaCB

and DATEDIFF(MINUTE,ThoiGianCatCanh,@Thoigiancatcanh) =0

and DATEDIFF(MINUTE,ThoiGianHaCanh,@thoigianhacanh)=0

select @so\_ghe\_trong=So\_ghe\_con\_trong ,@sohieu = SoHieu FROM LICHBAY

WHERE LICHBAY.MaCB=@MaCB

and DATEDIFF(MINUTE,ThoiGianCatCanh,@Thoigiancatcanh) =0

and DATEDIFF(MINUTE,ThoiGianHaCanh,@thoigianhacanh)=0

select @so\_ghe\_toi\_da= MAYBAY.SoLuongGheToiDa FROM MAYBAY WHERE MAYBAY.SoHieu=@sohieu

if( @so\_ghe\_toi\_da <@so\_ghe\_trong )

BEGIN

print'Khong the xoa ve do ve ko ton tai !'

ROLLBACK TRANSACTION

END

END

GO

* **Transaction:**
* **View:**
  + Thống kê thông tin phi công sắp xếp theo mức lương

CREATE VIEW Thong\_Ke\_Luong\_PC AS

SELECT TOP(99.99) PERCENT MaPC,TenPC,Luong\_Phi\_cong

from PHICONG

ORDER BY Luong\_Phi\_cong

* + Thống kê : số lượng vé bán được mỗi tháng ( sắp xếp theo năm ⇒ tháng)

CREATE VIEW Thong\_Ke\_SL\_Ve AS

SELECT TOP(99.99) PERCENT YEAR(ThoiGianDatVe) AS Nam, MONTH(ThoiGianDatVe) AS Thang, COUNT(\*) AS SoLuongVe

FROM VE

GROUP BY MONTH(ThoiGianDatVe), YEAR(ThoiGianDatVe)

ORDER BY MONTH(ThoiGianDatVe), YEAR(ThoiGianDatVe)

* + Thống kê Tổng số chuyến bay của mỗi phi công trong tháng

CREATE VIEW Thong\_ke\_PC\_bay\_nhieu\_nhat\_moi\_thang

AS

select TOP(99.99) PERCENT YEAR(ThoiGianCatCanh) AS Nam, MONTH(ThoiGianCatCanh) AS Thang,MaPC ,COUNT(MaPC) AS SoLuongchuyenbay from LICHBAY

group by MaPC,

MONTH(ThoiGianCatCanh),

YEAR(ThoiGianCatCanh)

ORDER BY MONTH(ThoiGianCatCanh), YEAR(ThoiGianCatCanh)

* + Thống kê: Mỗi loại máy bay có bao nhiêu máy bay

CREATE VIEW SoLuongMayBay\_View AS

SELECT HangSanXuat as HangMayBay, COUNT(\*) as SoLuong

FROM dbo.MAYBAY, dbo.LOAIMAYBAY

WHERE LOAIMAYBAY.MaLoaiMayBay = MAYBAY.MaLoaiMayBay

GROUP BY HangSanXuat

GO

# **CHƯƠNG 3: TÍNH NĂNG**

1. **Thủ tục:**
   1. **Thêm/ sửa/ xóa bảng máy bay**

* Thêm

CREATE PROC INSERT\_MAY\_BAY

(

@So\_hieu varchar (10),

@MaLoaiMayBay varchar (10),

@Soghetoida int

)

AS

BEGIN

INSERT INTO MAYBAY VALUES (@So\_hieu,@MaLoaiMayBay,@Soghetoida)

END

* Sửa

CREATE PROC UPDATE\_MAY\_BAY

(

@So\_hieu varchar (10),

@MaLoaiMayBay varchar (10),

@Soghetoida int

)

AS

BEGIN

UPDATE MAYBAY

SET SoHieu=@So\_hieu, MaLoaiMayBay=@MaLoaiMayBay,SoLuongGheToiDa=@Soghetoida

WHERE SoHieu= @So\_hieu

END

* Xóa

CREATE PROC DELETE\_MAY\_BAY

(@So\_hieu varchar (10))

AS

BEGIN

DELETE MAYBAY WHERE SoHieu= @So\_hieu

END

* 1. **Gợi ý chuyến bay**

CREATE PROC proc\_goi\_y\_chuyen\_bay

(

@Ngay\_muon\_bay date,

@Noi\_di varchar(10),

@Noi\_den varchar(10)

)

AS

BEGIN

SELECT MaCB,ThoiGianCatCanh,ThoiGianHaCanh from LICHBAY

Where LICHBAY.SanBayDi=@Noi\_di

and LICHBAY.SanBayDen=@Noi\_den

and FORMAT(LICHBAY.ThoiGianCatCanh,'yyyy-MM-dd')=@Ngay\_muon\_bay

and So\_ghe\_con\_trong>0

END

GO

* 1. **Thêm/ sửa/ xóa loại máy bay**
* Thêm

CREATE PROC INSERT\_LOAIMAYBAY

(

@Maloaimaybay varchar(10),

@hangsanxuat varchar(50)

)

AS

BEGIN

INSERT INTO LOAIMAYBAY VALUES (@Maloaimaybay,@hangsanxuat)

END

* Sửa

CREATE PROC UPDATE\_LOAIMAYBAY

(

@Maloaimaybay varchar(10),

@hangsanxuat varchar(50)

)

AS

BEGIN

UPDATE LOAIMAYBAY

SET MaLoaiMayBay=@Maloaimaybay,HangSanXuat=@hangsanxuat

WHERE MaLoaiMayBay = @Maloaimaybay

END

* Xóa

CREATE PROC DELETE\_LOAIMAYBAY

(@MaLoaiMayBay varchar (10))

AS

BEGIN

DELETE LOAIMAYBAY WHERE MaLoaiMayBay=@MaLoaiMayBay

END

* 1. **Thống kê : Tình trạng bay các chuyến bay cất cánh từ ngày x → y**

CREATE PROC proc\_Thong\_ke\_tinh\_trang\_chuyen\_bay\_catcanh\_tu\_ngay\_xy

(

@Ngay\_bat\_dau date,

@Ngay\_ket\_thuc date

)

AS

BEGIN

Select MaCB,ThoiGianCatCanh,ThoiGianHaCanh,SanBayDi,SanBayDen,TrangThaiMayBay From LICHBAY

Where FORMAT(LICHBAY.ThoiGianCatCanh, 'yyyy-MM-dd') between @Ngay\_bat\_dau and @Ngay\_ket\_thuc

END

* 1. **Thêm/ sửa / xóa bảng hành khách**
* Thêm

CREATE PROCEDURE Insert\_HanhKhach

(

@CMND\_HK varchar(10) ,

@HotenHK varchar(50) ,

@NgaySinhHK DATE,

@DiachiHK varchar(50),

@SdtHK varchar(11)

)

AS

BEGIN

INSERT INTO [dbo].[HANHKHACH]

(

[CMND\_HK],

[HotenHK],

[NgaySinhHK],

[DiachiHK],

[SdtHK]

)

VALUES

(

@CMND\_HK,

@HotenHK,

@NgaySinhHK,

@DiachiHK,

@SdtHK

)

SELECT ERROR\_MESSAGE = N'Thêm thành công'

END

GO

* Sửa

CREATE PROCEDURE Update\_HanhKhach

(

@CMND\_HK varchar(10) ,

@HotenHK varchar(50) ,

@NgaySinhHK DATE,

@DiachiHK varchar(50),

@SdtHK varchar(11)

)

AS

BEGIN

UPDATE [dbo].[HANHKHACH]

SET

[HotenHK] = @HotenHK,

[NgaySinhHK] = @NgaySinhHK,

[DiachiHK] = @DiachiHK,

[SdtHK] = @SdtHK

WHERE

[CMND\_HK] = @CMND\_HK;

SELECT ERROR\_MESSAGE = N'Sửa thành công'

END

GO

* Xóa

CREATE PROCEDURE Delete\_HanhKhach

(

@CMND\_HK varchar(10)

)

AS

BEGIN

DELETE FROM [dbo].[HANHKHACH]

WHERE [CMND\_HK] = @CMND\_HK

SELECT ERROR\_MESSAGE = N'Xóa thành công'

END

GO

* 1. **Từ mã chuyến bay , thời gian cất cánh, thời gian hạ cánh ⇒ trả về danh sách khách hàng có trên chuyến bay**

CREATE PROCEDURE DS\_HANHKHACH

(

@MaCB nchar(10),

@ThoiGianCatCanh DATETIME,

@ThoiGianHaCanh DATETIME

)

AS

BEGIN

SELECT

HANHKHACH.CMND\_HK as CMND,

HANHKHACH.HotenHK as HoTen,

HANHKHACH.NgaySinhHK as NgaySinh,

HANHKHACH.DiachiHK as DiaChi,

HANHKHACH.SdtHK as SDT

FROM ((LICHBAY

INNER JOIN VE ON LICHBAY.MaCB = VE.MaCB)

INNER JOIN HANHKHACH ON VE.CMND\_HK = HANHKHACH.CMND\_HK)

WHERE LICHBAY.MaCB = @MaCB

AND LICHBAY.ThoiGianCatCanh = @ThoiGianCatCanh

AND LICHBAY.ThoiGianHaCanh = @ThoiGianHaCanh

END;

* 1. **Thêm/ sửa/ xóa bảng lịch bay**
* Thêm

CREATE PROCEDURE AddLichBay

@MaCB varchar(10),

@SoHieu Varchar(10),

@ThoiGianCatCanh DATETIME,

@ThoiGianHaCanh DATETIME,

@SanBayDi varchar(50),

@SanBayDen varchar(50),

@TrangThaiMayBay varchar(30),

@MaPC varchar(10)

AS

INSERT INTO LICHBAY (MaCB,SoHieu,ThoiGianCatCanh,ThoiGianHaCanh,SanBayDi,SanBayDen,TrangTha iMayBay,MaPC)

VALUES ( @MaCB, @SoHieu, @ThoiGianCatCanh, @ThoiGianHaCanh, @SanBayDi, @SanBayDen, @TrangThaiMayBay, @MaPC)

GO

* Sửa

CREATE PROCEDURE UpdateLichBay

@MaCB varchar(10),

@SoHieu Varchar(10),

@ThoiGianCatCanh DATETIME,

@ThoiGianHaCanh DATETIME,

@SanBayDi varchar(50),

@SanBayDen varchar(50),

@TrangThaiMayBay varchar(30),

@MaPC varchar(10)

AS

UPDATE LICHBAY

SET MaCB = @MaCB,

SoHieu = @SoHieu,

ThoiGianCatCanh = @ThoiGianCatCanh,

ThoiGianHaCanh = @ThoiGianHaCanh,

SanBayDi = @SanBayDi,

SanBayDen = @SanBayDen,

TrangThaiMayBay = @TrangThaiMayBay,

MaPC = @MaPC

WHERE MaCB = @MaCB

AND ThoiGianCatCanh = @ThoiGianCatCanh

AND ThoiGianHaCanh = @ThoiGianHaCanh

GO

* Xóa

CREATE PROCEDURE DeleteLichBay @MaCBCheck varchar(10), @ThoiGianCatCanhCheck DATETIME, @ThoiGianHaCanhCheck DATETIME

AS

DELETE LICHBAY

WHERE MaCB = @MaCBCheck

AND ThoiGianCatCanh = @ThoiGianCatCanhCheck

AND ThoiGianHaCanh = @ThoiGianHaCanhCheck

GO

* 1. **Xem/ thêm/ sửa/ xóa bảng vé**
* Xem

CREATE PROC proc\_quan\_li\_don\_hang

@IDVE int

AS

Select MaVe,CMND\_HK,ThoiGianDatVe,ThoiGianCatCanh,

ThoiGianHaCanh,ThoiGianHaCanh,Noi\_Di,

Noi\_Den,SoGhe,TongGiaVe from VE

where VE.MaVe=@IDVE

GO

* Thêm

CREATE PROC proc\_insert\_Ve

(

@CMND\_HK varchar(10) ,

@MaCB varchar(10) ,

@ThoiGianCatCanh DATETIME ,

@ThoiGianHaCanh DATETIME ,

@Ngay\_can\_bay DATE,

@Noi\_Di varchar(50),

@Noi\_Den varchar(50) ,

@SoGhe int,

@GiaVe float

)

AS

BEGIN

INSERT VE (CMND\_HK,MaCB,ThoiGianCatCanh,ThoiGianHaCanh,Ngay\_can\_bay,Noi\_Di,Noi\_De n,SoGhe,GiaVe)

VALUES (@CMND\_HK,@MaCB,@ThoiGianCatCanh,@ThoiGianHaCanh,@Ngay\_can\_bay,@Noi\_Di, @Noi\_Den,@SoGhe,@GiaVe)

END

* Sửa

CREATE PROC proc\_update\_Ve

(

@MaVe int,

@CMND\_HK varchar(10) ,

@MaCB varchar(10) ,

@ThoiGianCatCanh DATETIME ,

@ThoiGianHaCanh DATETIME ,

@Ngay\_can\_bay DATE,

@Noi\_Di varchar(50),

@Noi\_Den varchar(50) ,

@SoGhe int,

@GiaVe float

)

AS

BEGIN

UPDATE VE

SET

CMND\_HK=@CMND\_HK,

MaCB=@MaCB ,

ThoiGianCatCanh=@ThoiGianCatCanh,

ThoiGianHaCanh=@ThoiGianHaCanh,

Ngay\_can\_bay=@Ngay\_can\_bay,

Noi\_Di=@Noi\_Di,

Noi\_Den=@Noi\_Den,

SoGhe=@SoGhe,

GiaVe=@GiaVe

WHERE MaVe=@MaVe

END

* Xóa

CREATE PROC proc\_delete\_Ve

@MaVe int

AS

BEGIN

DELETE FROM VE

WHERE MaVe=@MaVe

END

**i.Thêm/ sửa/ xóa bảng Phi Công**

-Thêm

CREATE PROC proc\_insert\_PC

(

@MaPC varchar(10) ,

@TenPC varchar(30) ,

@DiaChiPC varchar(50),

@SDT\_Phi\_cong varchar(11)

)

AS

INSERT INTO [QUAN\_LI\_CHUYEN\_BAY].[dbo].[PHICONG]

(MaPC,TenPC,DiaChiPC,SDT\_Phi\_cong)

VALUES (@MaPC,@TenPC,@DiaChiPC,@SDT\_Phi\_cong)

-Sửa

CREATE PROC proc\_update\_PC

(

@MaPC varchar(10) ,

@TenPC varchar(30) ,

@DiaChiPC varchar(50),

@SDT\_Phi\_cong varchar(11)

)

AS

UPDATE [QUAN\_LI\_CHUYEN\_BAY].[dbo].[PHICONG]

SET

MaPC =@MaPC,

TenPC=@TenPC,

DiaChiPC=@DiaChiPC,

SDT\_Phi\_cong=@SDT\_Phi\_cong

Where MaPC=@MaPC

-Xóa

CREATE PROC proc\_delete\_PC

@MaPC varchar(10)

AS

DELETE FROM [QUAN\_LI\_CHUYEN\_BAY].[dbo].[PHICONG]

WHERE MaPC=@MaPC

j. **Thêm / xóa bảng Khả Năng**

-Thêm

CREATE PROC proc\_insert\_KhaNang

(

@MAPC varchar(10) ,

@MaLoaiMayBay varchar(10)

)

AS

INSERT INTO [QUAN\_LI\_CHUYEN\_BAY].[dbo].[KHANANG]

VALUES (@MAPC,@MaLoaiMayBay)

-**Xóa**

CREATE PROC proc\_delete\_KhaNang

(

@MAPC varchar(10) ,

@MaLoaiMayBay varchar(10)

)

AS

DELETE FROM [QUAN\_LI\_CHUYEN\_BAY].[dbo].[KHANANG]

Where MAPC=@MAPC

and MaLoaiMayBay=@MaLoaiMayBay

1. **Hàm:**
   1. **Tính thời gian còn hiệu lực sửa vé : nếu < 24h thì return 0 , nếu ≥24 return 1**

CREATE FUNCTION hieu\_luc\_doi\_ve

(@ma\_ve int)

RETURNS INT

AS

BEGIN

DECLARE @Khoang\_cach\_dat\_ve float

select @Khoang\_cach\_dat\_ve=DATEDIFF(HOUR,VE.ThoiGianDatVe,VE.ThoiGianCatCanh) from VE

where VE.MaVe=@ma\_ve

IF(@Khoang\_cach\_dat\_ve <=24)

BEGIN

RETURN 0

END

RETURN 1

END

b.**Kiểm tra có phải ngày cuối tháng ( có -1 , ko - 0)**

CREATE FUNCTION NgayCuoiThang ( @ThoiGianCatCanh DATE )

RETURNS INT

AS

BEGIN

DECLARE @CheckThang INT, @CheckNgay INT

SELECT @CheckThang = MONTH(@ThoiGianCatCanh), @CheckNgay = DAY(@ThoiGianCatCanh)

IF (@CheckNgay >= 30 and @CheckThang != 2) or (@CheckNgay >= 28 and @CheckThang = 2)

RETURN 1

RETURN 0

END

GO

c.**Tính số ghế tối đa của chuyến bay vé vừa đặt**

CREATE FUNCTION funtion\_ghe\_toi\_da

(@ID\_VE int )

RETURNS INT

AS

BEGIN

DECLARE @So\_ghe\_toi\_da int;

select @So\_ghe\_toi\_da=MAYBAY.SoLuongGheToiDa from (VE inner join LICHBAY

on Ve.MaCB=LICHBAY.MaCB

and VE.ThoiGianCatCanh=LICHBAY.ThoiGianCatCanh

and VE.ThoiGianHaCanh=LICHBAY.ThoiGianHaCanh)

inner join MAYBAY

on MAYBAY.SoHieu=LICHBAY.SoHieu

Where VE.MaVe=@ID\_VE

RETURN @So\_ghe\_toi\_da

END

GO

d.**Tính số ghế đã đặt trên chuyến bay của vé vừa đặt**

CREATE FUNCTION [funtion\_so\_ghe\_da\_dat]

(@ID\_VE int ) RETURNS INT AS BEGIN DECLARE @MaCB varchar(10) DECLARE @Thoigiancatcanh datetime DECLARE @thoigianhacanh datetime

select @MaCB=MaCB,@Thoigiancatcanh=VE.ThoiGianCatCanh,@thoigianhacanh=thoigianhacanh from VE

Where VE.MaVe=@ID\_VE

DECLARE @soghedadat int

select @soghedadat=count(\*) from VE

where VE.MaCB=@MaCB

and format(VE.ThoiGianCatCanh,'YYYY-MM-DD hh:mm:ss')

=format(@Thoigiancatcanh,'YYYY-MM-DD hh:mm:ss')

and format(VE.ThoiGianHaCanh,'YYYY-MM-DD hh:mm:ss')

=format(@thoigianhacanh,'YYYY-MM-DD hh:mm:ss')

RETURN @soghedadat

END

# **CHƯƠNG 4: tạo user + phân quyền ( USER+ ADMIN )**

1. **Quản lí tài khỏan ⇒ admin**

* Tạo bảng User

CREATE TABLE tbUser

(

UserID int IDENTITY(1,1) NOT NULL,

Username varchar(10) NOT NULL,

Passwords varchar(15) NOT NULL,

Actions varchar(10) NOT NULL,

CONSTRAINT pk\_user PRIMARY KEY (UserID)

)

* Ràng buộc Check

ALTER TABLE tbUser ADD CONSTRAINT check\_Action

CHECK (Actions LIKE 'QLNS' OR Actions LIKE 'QLDV')

GO

* Tạo tài khoản quản lí 2 nhóm quyền QLNS và QLDV

CREATE LOGIN GIAMDOC WITH PASSWORD = '123'

CREATE USER GIAMDOC FOR LOGIN GIAMDOC

EXEC sp\_addrolemember 'db\_owner', 'GIAMDOC'

EXEC sp\_addrole 'QLNS', 'GIAMDOC'

EXEC sp\_addrole 'QLDV', 'GIAMDOC'

GO

* Thay đổi quyền

CREATE PROCEDURE ChinhSuaQuyen

(

@Username varchar(15),

@Actions varchar(10)

)

AS

BEGIN

IF EXISTS (SELECT \* FROM tbUser WHERE Username = @Username)

BEGIN

UPDATE tbUser

SET Actions = @Actions

WHERE Username = @Username

PRINT N'Thay đổi quyền thành công'

END

ELSE

BEGIN

SELECT ERROR\_MESSAGE = N'Tài khoản không tồn tại'

END

END

GO

* Thêm tài khoản

CREATE PROCEDURE ThemTaiKhoan

(

@Username varchar(10),

@Passwords varchar(15),

@Actions varchar(10)

)

AS

BEGIN

IF NOT EXISTS (SELECT \* FROM tbUser WHERE Username = @Username)

BEGIN

INSERT INTO tbUser

VALUES(@Username, @Passwords, @Actions)

PRINT N'Thêm tài khoản thành công'

END

ELSE

BEGIN

SELECT ERROR\_MESSAGE = N'Thêm tài khoản thất bại'

END

END

GO

* Xóa Tài Khoản

CREATE PROCEDURE XoaTaiKhoan

(

@Username varchar(10)

)

AS

BEGIN

IF EXISTS (SELECT \* FROM tbUser WHERE Username = @Username)

BEGIN

DELETE FROM dbo.tbUser

WHERE Username = @Username;

PRINT N'Xóa tài khoản thành công'

END

ELSE

BEGIN

PRINT N'Xóa tài khoản thất bại'

END

END

GO

* Trigger khi thêm tài khoản vào table tbUser thành công thì phân quyền cho tạo tài khoản và phân quyền cho tài khoản đó

CREATE TRIGGER PQ\_ThemTaiKhoan

ON tbUser

FOR INSERT

AS

BEGIN

DECLARE @sql varchar(max)

DECLARE @Username varchar(10), @Passwords varchar(15), @Actions varchar(10)

SELECT @Username = i.Username, @Passwords = i.Passwords, @Actions = i.Actions

FROM inserted i

-- Tài khoản là quản lí nhân sự --

IF (@Actions = 'QLNS')

BEGIN

PRINT N'Cấp quyền cho quản lí nhân sự'

SET @sql = '

USE [QUAN\_LI\_CHUYEN\_BAY]

CREATE LOGIN ['+@Username+'] WITH PASSWORD = '''+@Passwords+'''

CREATE USER ['+@Username+'] FOR LOGIN ['+@Username+']

GRANT SELECT, INSERT, UPDATE, DELETE ON PHICONG TO QLNS

GRANT SELECT, INSERT, UPDATE, DELETE ON KHANANG TO QLNS

GRANT SELECT, INSERT, UPDATE, DELETE ON LOAIMAYBAY TO QLNS

GRANT SELECT, INSERT, UPDATE, DELETE ON MAYBAY TO QLNS

GRANT SELECT ON HANHKHACH TO QLNS

GRANT SELECT ON VE TO QLNS

GRANT SELECT ON LICHBAY TO QLNS

EXEC sp\_addrolemember ''QLNS'', ['+@Username+']'

EXEC (@sql)

PRINT N'Thành công'

END

ELSE IF (@Actions = 'QLDV')

BEGIN

PRINT N'Cấp quyền cho quản lí dịch vụ'

SET @sql = '

USE [QUAN\_LI\_CHUYEN\_BAY]

CREATE LOGIN ['+@Username+'] WITH PASSWORD = '''+@Passwords+'''

CREATE USER ['+@Username+'] FOR LOGIN ['+@Username+']

GRANT SELECT, INSERT, UPDATE, DELETE ON HANHKHACH TO QLDV

GRANT SELECT, INSERT, UPDATE, DELETE ON VE TO QLDV

GRANT SELECT, INSERT, UPDATE, DELETE ON LICHBAY TO QLDV

GRANT SELECT ON PHICONG TO QLDV

GRANT SELECT ON KHANANG TO QLDV

GRANT SELECT ON LOAIMAYBAY TO QLDV

GRANT SELECT ON MAYBAY TO QLDV

EXEC sp\_addrolemember ''QLDV'', ['+@Username+']'

EXEC (@sql)

PRINT N'Thành công'

END

ELSE

ROLLBACK TRANSACTION

END

GO

* Trigger khi thay đổi quyền vào table tbUser sẽ đổi quyền thành công

CREATE TRIGGER PQ\_ChinhSuaQuyen

ON tbUser

FOR UPDATE

AS

BEGIN

DECLARE @Username1 varchar(10), @Username2 varchar(10), @Actions varchar(10)

SELECT @Username1 = i.Username, @Actions = i.Actions

FROM inserted i

SELECT @Username2 = d.Username

FROM deleted d

DECLARE @sql varchar(max)

IF (@Username1 = @Username2)

-- Tài khoản là quản lí nhân sự --

IF (@Actions = 'QLNS')

BEGIN

PRINT N'Cấp quyền cho quản lí nhân sự'

SET @sql = '

USE [QUAN\_LI\_CHUYEN\_BAY]

GRANT SELECT, INSERT, UPDATE, DELETE ON PHICONG TO QLNS

GRANT SELECT, INSERT, UPDATE, DELETE ON KHANANG TO QLNS

GRANT SELECT, INSERT, UPDATE, DELETE ON LOAIMAYBAY TO QLNS

GRANT SELECT, INSERT, UPDATE, DELETE ON MAYBAY TO QLNS

GRANT SELECT ON HANHKHACH TO QLNS

GRANT SELECT ON VE TO QLNS

GRANT SELECT ON LICHBAY TO QLNS

EXEC sp\_addrolemember ''QLNS'', ['+@Username1+']'

EXEC (@sql)

PRINT N'Thành công'

END

ELSE IF (@Actions = 'QLDV')

BEGIN

PRINT N'Cấp quyền cho quản lí dịch vụ'

SET @sql = '

USE [QUAN\_LI\_CHUYEN\_BAY]

GRANT SELECT, INSERT, UPDATE, DELETE ON HANHKHACH TO QLDV

GRANT SELECT, INSERT, UPDATE, DELETE ON VE TO QLDV

GRANT SELECT, INSERT, UPDATE, DELETE ON LICHBAY TO QLDV

GRANT SELECT ON PHICONG TO QLDV

GRANT SELECT ON KHANANG TO QLDV

GRANT SELECT ON LOAIMAYBAY TO QLDV

GRANT SELECT ON MAYBAY TO QLDV

EXEC sp\_addrolemember ''QLDV'', ['+@Username1+']'

EXEC (@sql)

PRINT N'Thành công'

END

ELSE

ROLLBACK TRANSACTION

END

GO

* Trigger khi xóa tài khoản ở table tbUser thì tài khoản sẽ xóa khỏi hệ thống và không thể login

CREATE TRIGGER PQ\_XoaTaiKhoan

ON tbUser

FOR DELETE

AS

BEGIN

DECLARE @Username varchar(10)

SELECT @Username = d.Username

FROM deleted d

DECLARE @sql varchar(max)

SET @sql = '

DROP USER ['+@Username+']

DROP LOGIN ['+@Username+']'

EXEC(@sql)

PRINT N'thành công'

END

GO

1. **Kiểm tra login + thay đổi password của user**

* Kiểm tra đăng nhập

CREATE PROCEDURE KiemTraDangNhap

(

@Username varchar(10),

@Passwords varchar(15)

)

AS

BEGIN

IF EXISTS (SELECT \* FROM tbUser WHERE Username LIKE @Username AND Passwords LIKE @Passwords)

BEGIN

SELECT \* FROM tbUser

WHERE Username LIKE @Username AND Passwords LIKE @Passwords

PRINT N'Tìm thấy tài khoản'

END

ELSE

BEGIN

SELECT ERROR\_MESSAGE = N'Tài khoản không tồn tại'

END

END

GO

* Kiểm tra quyền

CREATE PROCEDURE KiemTraQuyen

@TenQuyen nvarchar(50)

AS

BEGIN

IF EXISTS (SELECT \* FROM tbUser WHERE Actions = @TenQuyen)

BEGIN

SELECT \* FROM tbUser

WHERE Actions = @TenQuyen

PRINT N'Tìm thấy các tài khoản thuộc quyền đã nhập'

END

ELSE

BEGIN

SELECT ERROR\_MESSAGE = N'Sai tên quyền'

END

END

GO

* Đổi Mật Khẩu

CREATE PROCEDURE DoiMatKhau

(

@Username varchar(50),

@NewPassword varchar(50)

)

AS

BEGIN

IF EXISTS (SELECT \* FROM tbUser WHERE Username = @Username)

BEGIN

UPDATE tbUser

SET Passwords = @NewPassword

WHERE Username = @Username

PRINT N'Đổi mật khẩu thành công'

END

ELSE

BEGIN

SELECT ERROR\_MESSAGE = N'Đổi mật khẩu thất bại'

END

END

GO

* Trigger khi đổi mật khẩu vào table tbUser sẽ đổi mật khẩu login thành công

CREATE TRIGGER PQ\_DoiMatKhau

ON tbUser

FOR UPDATE

AS

BEGIN

DECLARE @Username1 varchar(10), @Username2 varchar(10), @Newpassword varchar(15)

SELECT @Username1 = i.Username, @Newpassword = i.Passwords

FROM inserted i

SELECT @Username2 = d.Username

FROM deleted d

DECLARE @sql varchar(max)

IF (@Username1 = @Username2)

BEGIN

SET @sql = '

ALTER LOGIN ['+@Username1+']

WITH PASSWORD = '''+@NewPassword+''' '

EXEC (@sql)

PRINT N'Đổi mật khẩu thành công'

END

ELSE

ROLLBACK TRANSACTION

END

GO

1. **Trigger : kiểm tra đảm bảo các tài khoản không trùng nhau**

* Trigger quản lí để không trùng tài khoản

CREATE TRIGGER tg\_CheckTaiKhoan

ON tbUser FOR INSERT, UPDATE

AS

BEGIN

DECLARE @Username varchar(50), @Temp int

SELECT @Username = inserted.Username FROM inserted

SELECT @Temp = COUNT(\*) FROM tbUser

WHERE Username = @Username

IF (@Temp > 1)

BEGIN

PRINT N'Tài khoản đã tồn tại'

ROLLBACK TRANSACTION

END

END

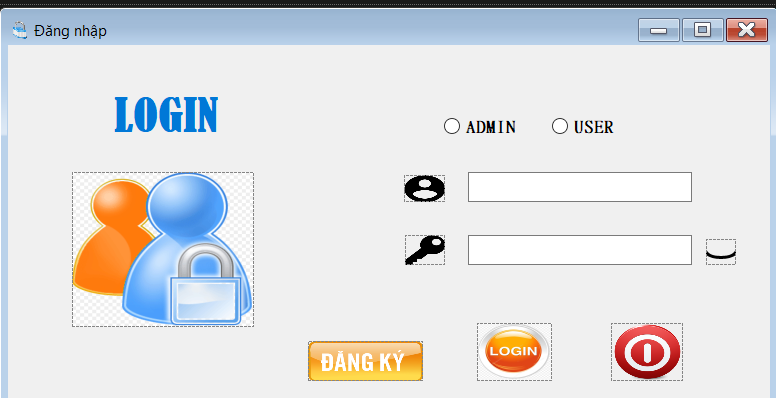
GO

# **CHƯƠNG 5: Hướng dẫn sử dụng phần mềm**

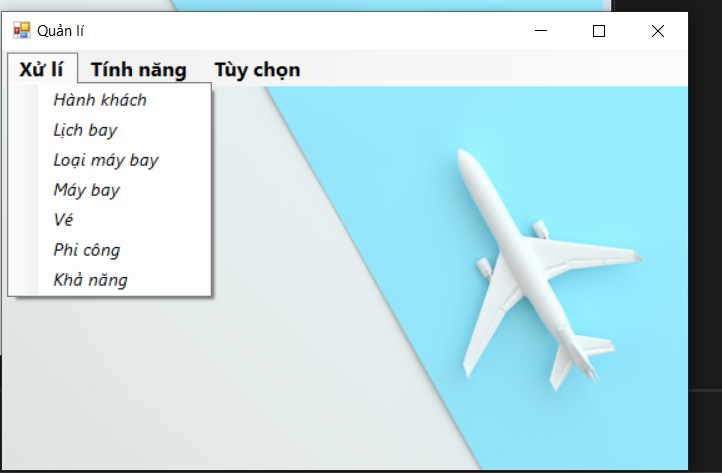
*1/ĐĂNG NHẬP VÀO HỆ THỐNG*

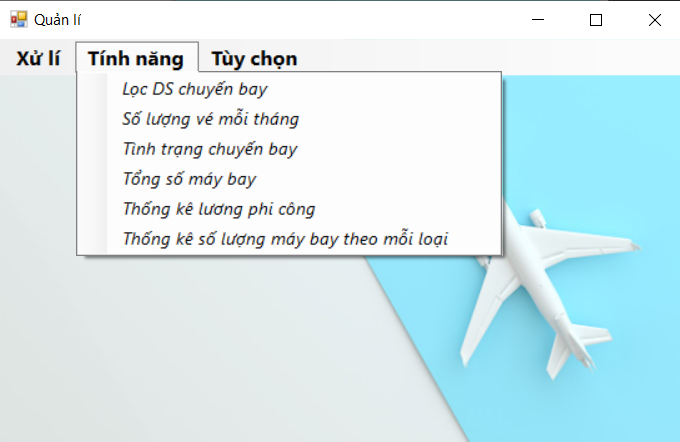
*Tài khoản test*

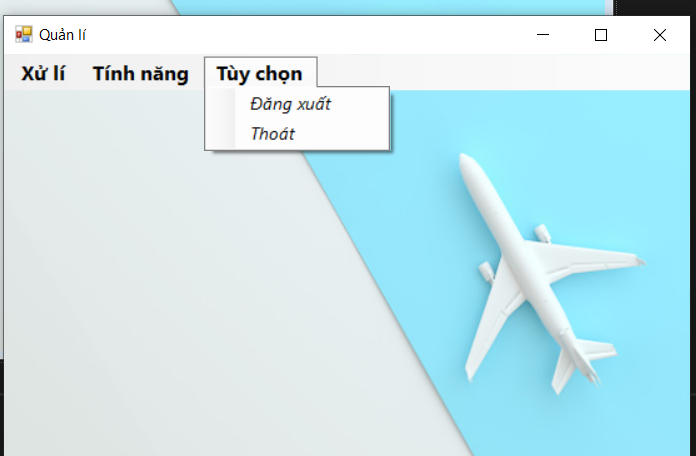
|  |  |  |
| --- | --- | --- |
| *Tài khoản* | *Tên đăng nhập* | *Mật Khẩu* |
| *ADMIN* | admin | admin |
| *USER* | user | user |



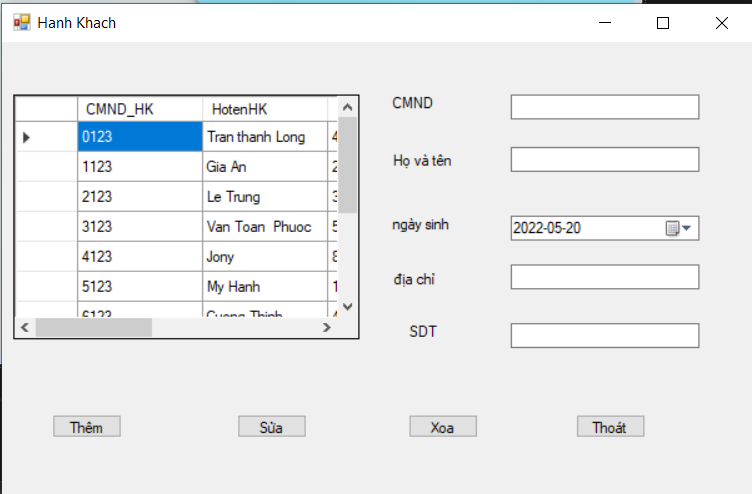
*2/ Đối với tài khoản ADMIN*

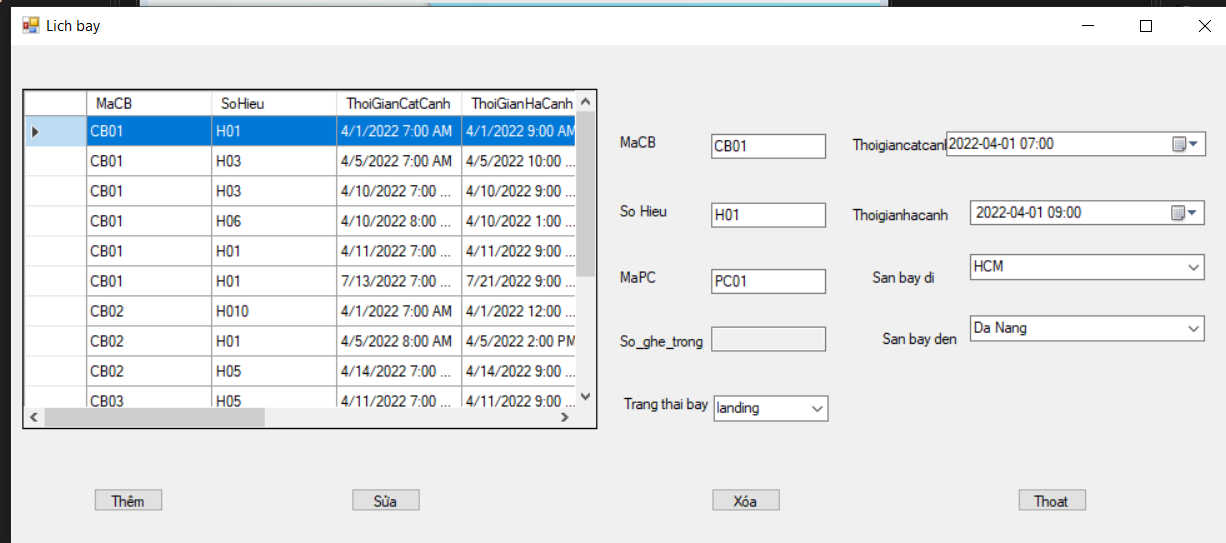




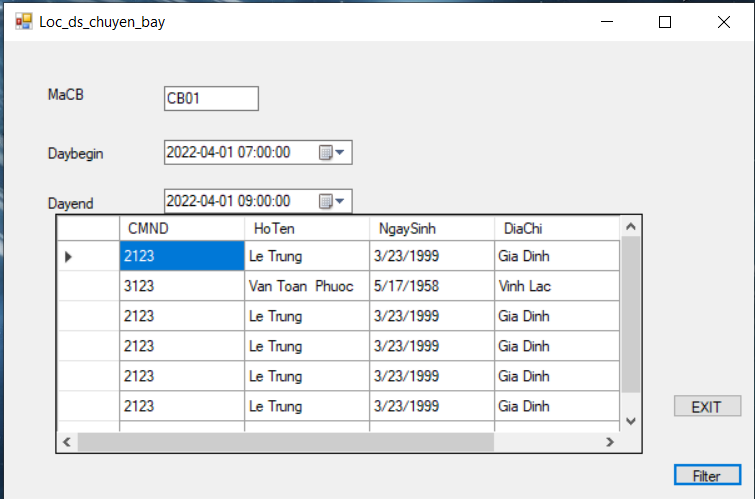


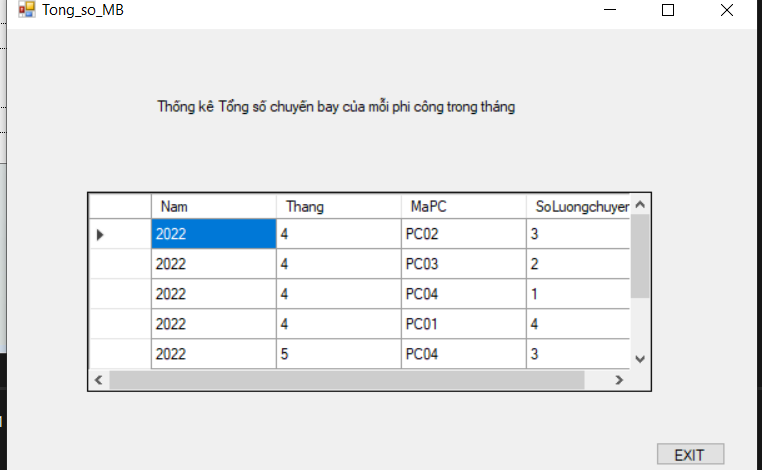
\*Một số tính năng **xử lí**



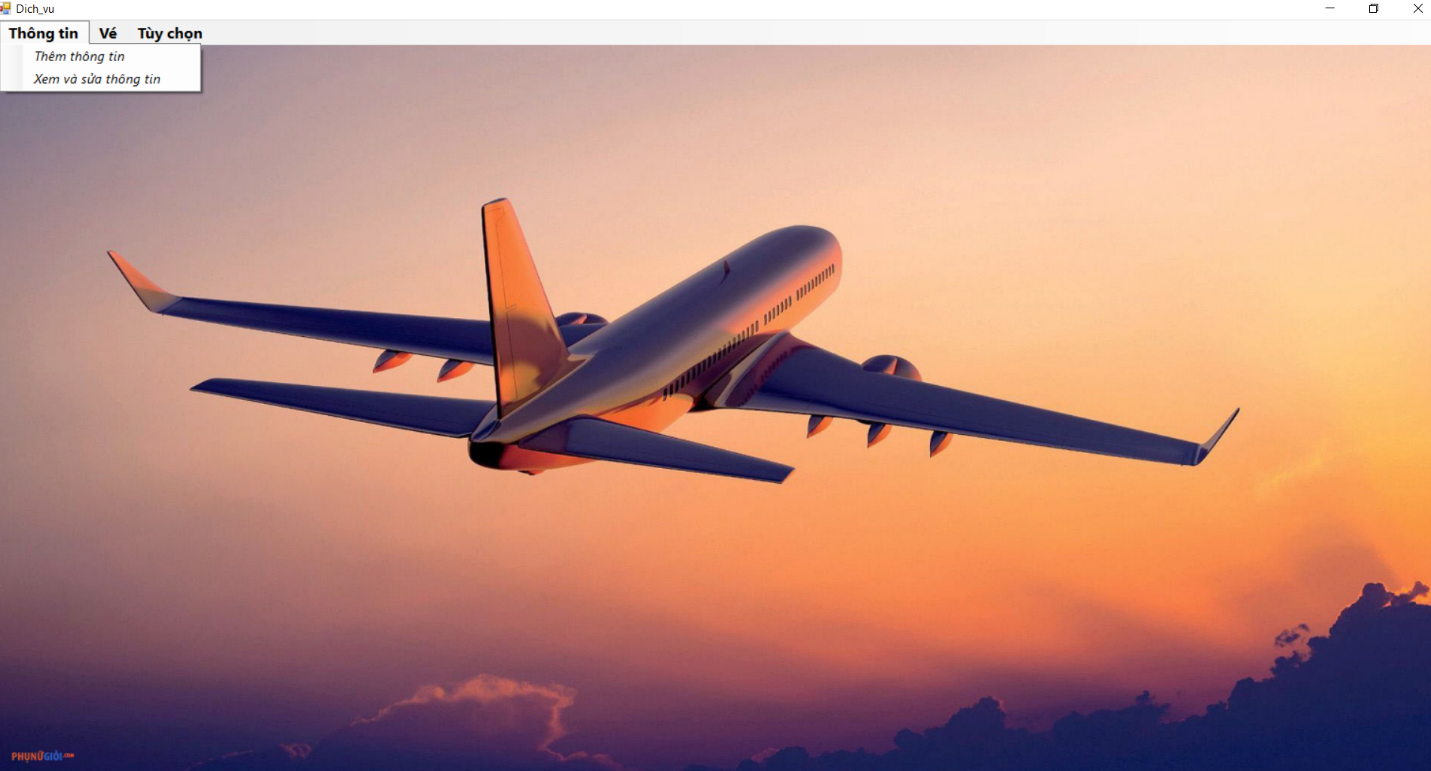


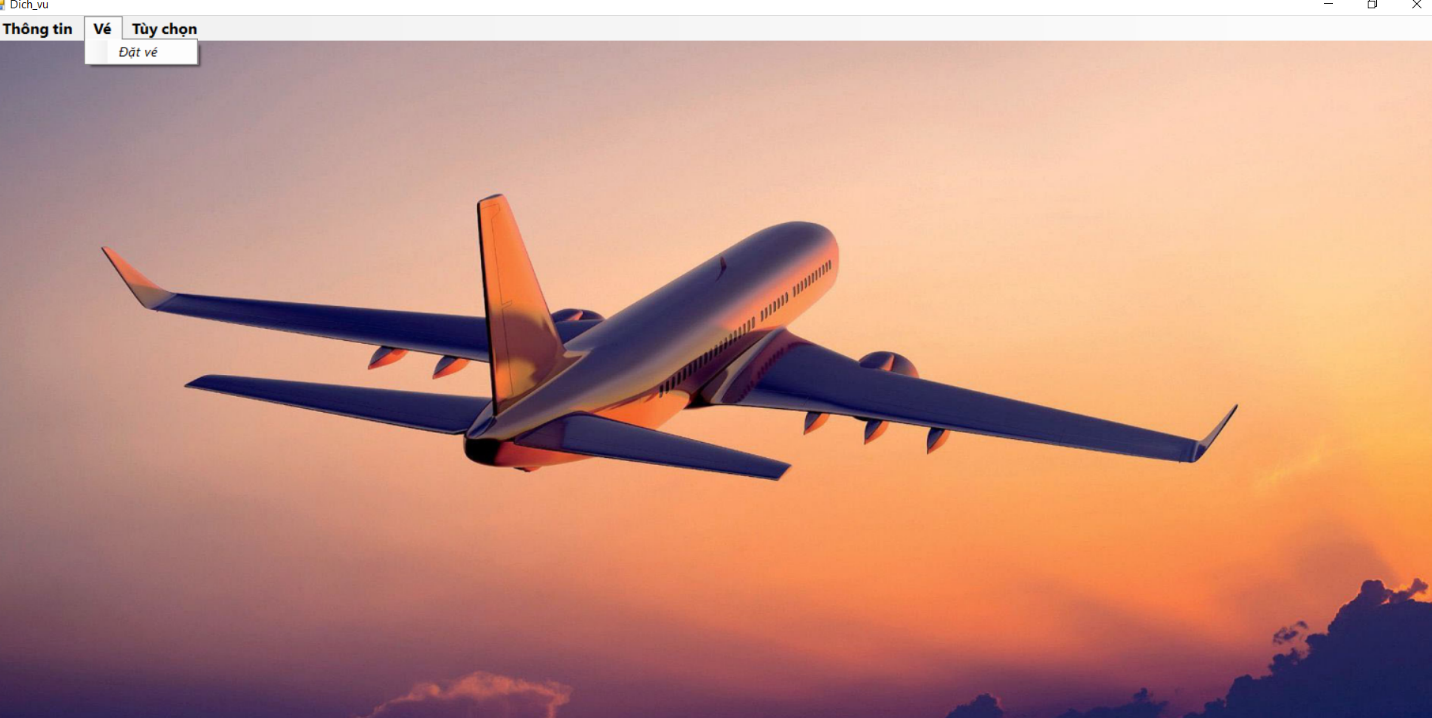
**\***Một số tính năng **“ Tính năng”**

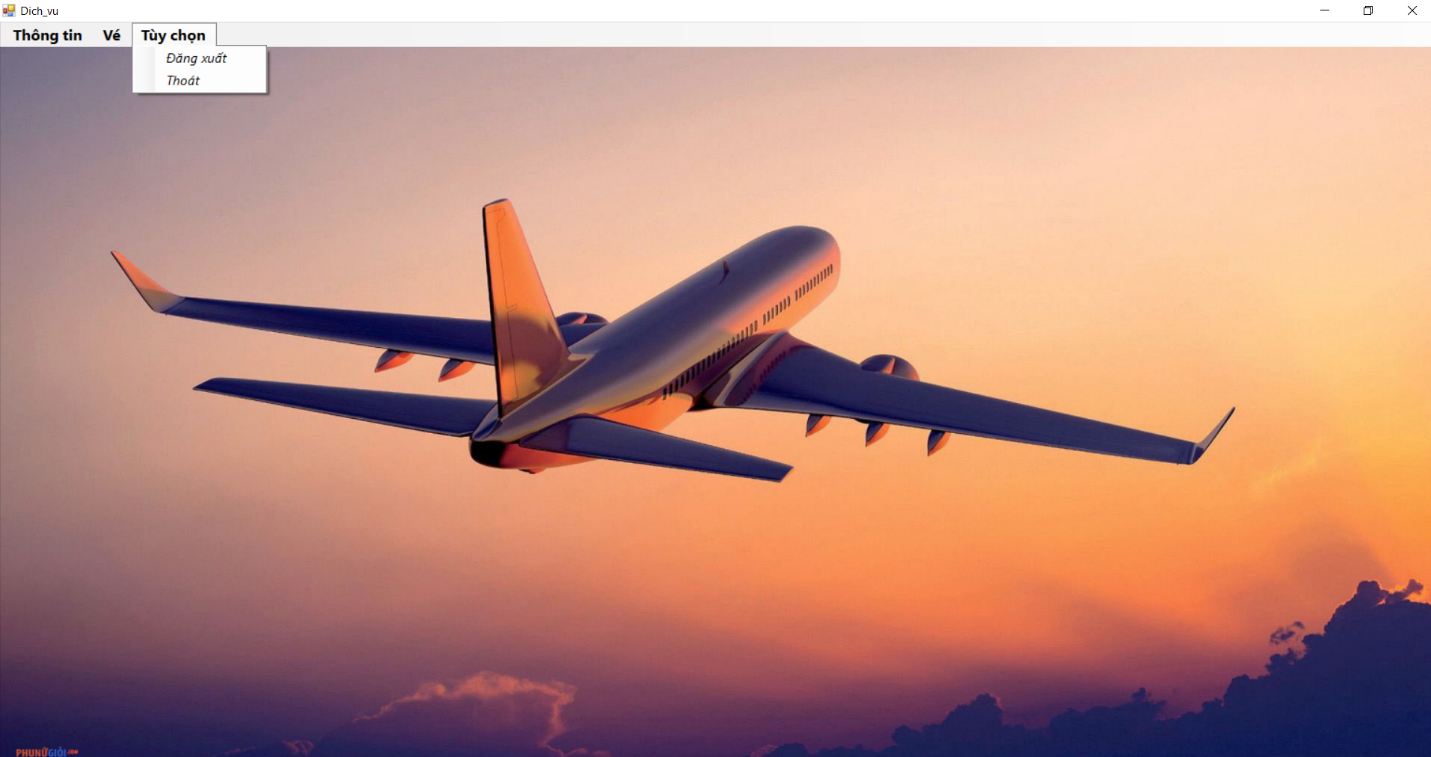




**3/ Đối với tài khoản user**







## **Phụ lục** : Thống kê số lượng trigger ,view,hàm, thủ tục, transactions

|  |  |
| --- | --- |
| TRIGGER | 12 |
| FUNCTION | 4 |
| VIEW | 4 |
| PROCEDURE | 30 |
| TRAN | 8 |

TRIGGER 12

CREATE TRIGGER Tinh\_Gio\_Bay\_Insert

CREATE TRIGGER Tinh\_Gio\_Bay\_Delete

CREATE TRIGGER Tinh\_Luong

CREATE TRIGGER Luong\_Giobay\_Bandau

CREATE TRIGGER ValidBangVe on VE

CREATE TRIGGER [dbo].[trigger\_Phan\_cong\_PC] on [dbo].[LICHBAY]

CREATE TRIGGER [dbo].[ghe\_trong\_delete] on [dbo].[VE]

CREATE TRIGGER PQ\_ThemTaiKhoan

CREATE TRIGGER PQ\_ChinhSuaQuyen

CREATE TRIGGER PQ\_XoaTaiKhoan

CREATE TRIGGER PQ\_DoiMatKhau

CREATE TRIGGER tg\_CheckTaiKhoan

FUNCTION 4

CREATE FUNCTION hieu\_luc\_doi\_ve

CREATE FUNCTION funtion\_ghe\_toi\_da

CREATE FUNCTION [funtion\_so\_ghe\_da\_dat]

CREATE FUNCTION NgayCuoiThang ( @ThoiGianCatCanh DATE )

VIEW 4

CREATE VIEW Thong\_Ke\_Luong\_PC AS

CREATE VIEW Thong\_Ke\_SL\_Ve AS

CREATE VIEW Thong\_ke\_PC\_bay\_nhieu\_nhat\_moi\_thang

CREATE VIEW SoLuongMayBay\_View

PROCEDURE 30

CREATE PROC INSERT\_MAY\_BAY

CREATE PROC UPDATE\_MAY\_BAY

CREATE PROC DELETE\_MAY\_BAY

CREATE PROC proc\_goi\_y\_chuyen\_bay

CREATE PROC INSERT\_LOAIMAYBAY

CREATE PROC UPDATE\_LOAIMAYBAY

CREATE PROC DELETE\_LOAIMAYBAY

CREATE PROC proc\_Thong\_ke\_tinh\_trang\_chuyen\_bay\_catcanh\_tu\_ngay\_xy

CREATE PROCEDURE Insert\_HanhKhach

CREATE PROCEDURE Update\_HanhKhach

CREATE PROCEDURE Delete\_HanhKhach

CREATE PROCEDURE DS\_HANHKHACH

CREATE PROCEDURE AddLichBay

CREATE PROCEDURE UpdateLichBay

CREATE PROCEDURE DeleteLichBay

CREATE PROC proc\_quan\_li\_don\_hang

CREATE PROC proc\_insert\_Ve

CREATE PROC proc\_update\_Ve

CREATE PROC proc\_delete\_Ve

CREATE PROC proc\_insert\_KhaNang

CREATE PROC proc\_delete\_KhaNang

CREATE PROC proc\_insert\_PC

CREATE PROC proc\_update\_PC

CREATE PROC proc\_delete\_PC

CREATE PROCEDURE ChinhSuaQuyen

CREATE PROCEDURE ThemTaiKhoan

CREATE PROCEDURE XoaTaiKhoan

CREATE PROCEDURE KiemTraDangNhap

CREATE PROCEDURE KiemTraQuyen

CREATE PROCEDURE DoiMatKhau

TRANSACTION 8

CREATE TRIGGER Luong\_Giobay\_Bandau

CREATE TRIGGER ValidBangVe

CREATE TRIGGER [dbo].[trigger\_Phan\_cong\_PC] on [dbo].[LICHBAY]

CREATE TRIGGER [dbo].[ghe\_trong\_delete] on [dbo].[VE]

CREATE TRIGGER PQ\_ThemTaiKhoan

CREATE TRIGGER PQ\_ChinhSuaQuyen

CREATE TRIGGER PQ\_DoiMatKhau

CREATE TRIGGER tg\_CheckTaiKhoan

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_END\_\_\_\_\_\_\_\_\_\_\_\_\_\_**