**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

----o0o----



**TIỂU LUẬN HỌC PHẦN**

**PHẦN MỀM QUẢN LÝ HỆ THỐNG THU PHÍ TỰ ĐỘNG ETC**

**MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ORACLE**

**GVHD: NGUYỄN ĐÌNH THÀNH**

**Thành phố Hồ Chí Minh, năm 2023**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

----o0o----



**TIỂU LUẬN HỌC PHẦN**

**PHẦN MỀM QUẢN LÝ HỆ THỐNG THU PHÍ TỰ ĐỘNG ETC**

**MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ORACLE**

**GVHD: NGUYỄN ĐÌNH THÀNH**

**Thành viên nhóm**

1. Nguyễn Minh Hòa - 2001200123
2. Từ Tiền Giang - 2001206913
3. Nguyễn Văn Hiệu – 2001202080
4. Nguyễn Chí Nguyện – 2001203010
5. Võ Thị Huỳnh Như – 2001202184

**Thành phố Hồ Chí Minh, năm 2023**

**BẢNG PHÂN CÔNG NHÓM**

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Họ tên** | **Hoàn Thành** |
| 2001200123 | Nguyễn Minh Hòa | **40%** |
| 2001206913 | Từ Tiền Giang | **15%** |
| 2001202080 | Nguyễn Văn Hiệu | **15%** |
| 2001203010 | Nguyễn Chí Nguyện | **15%** |
| 2001202184 | Võ Thị Huỳnh Như | **15%** |

# **TỔNG QUAN ĐỀ TÀI**

## Sơ lược:

Công ty VETC được thành lập với mục tiêu trở thành đơn vị triển khai và vận hành hệ thống thu phí tự động ETC tại Việt Nam. Mong muốn của VETC là mang đến giải pháp thu phí ứng dụng công nghệ hiện đại góp phần tăng tốc cho sự phát triển chung của lĩnh vực Giao thông Vận tải nói riêng và nền kinh tế Việt Nam nói chung.

Ý tưởng đầu tư hệ thống thu phí tự động không dừng của VETC được bắt nguồn từ những đánh giá về lợi ích mà hệ thống này mang lại, đồng thời dựa trên thực tế thành công của nhiều quốc gia, vùng lãnh thổ trên thế giới. VETC mong muốn mang trên mình sứ mệnh cung cấp sản phẩm và dịch vụ Giao thông thông minh vì sự phát triển của văn minh xã hội.

Thu phí không dừng (ETC – Electronic Toll Collection) là hình thức thu phí áp dụng công nghệ tự động nhận diện phương tiện và trừ tiền vào tài khoản giao thông. Theo đó, phương tiện khi qua làn ETC không cần dừng lại trả tiền mà vẫn có thể đi qua trạm thu phí. Hệ thống này được chính thức đi vào hoạt động ngày 01/08/2022.

## Đối tượng hướng đến:

Đối tượng hướng đến của thu phí tự động là các phương tiện giao thông tham gia vào lưu thông trên các tuyến đường cao tốc hoặc cầu đường phía trên. Các phương tiện này bao gồm ô tô, xe tải, xe buýt, xe khách và các phương tiện vận tải khác.

## Mô tả chi tiết ứng dụng, giới thiệu chức năng:

Phần mềm quản lý hệ thống thu phí tự động được đặt ở các trạm thu phí bot, phần mềm tự động thu phí khi xe đi qua đồng thời xuất hóa đơn thông tin của xe dựa vào thẻ tag.

### Chức năng chính:

* Quản lý dữ liệu khách hàng
* Quản lý dữ liệu cung cấp thẻ tag
* Quản lý dữ liệu vé thu
* Quản lý dữ liệu xe

### Bảo mật của ứng dụng:

* Phân quyền sử dụng
* Kiểm soát phiên truy cập
* Thao tác, quản lý thông tin tài khoản

### Chức năng hỗ trợ:

* Thống kê khách hàng
* Kiểm soát người dùng
* Tìm kiếm, tra cứu thông tin

## Mô tả quy trình nghiệp vụ:

### Quy trình đăng ký và nạp tiền:

*Bước 1:* Khách hàng đăng ký và đăng nhập vào hệ thống để đăng ký tài khoản và thông tin về phương tiện (xe) và thẻ thu phí.

*Bước 2:* Khách hàng nạp tiền vào thẻ thu phí.

*Bước 3:* Khách hàng lắp thẻ thu phí trên xe.

### Quy trình lập hóa đơn và thanh toán:

*Bước 1:* Khi phương tiện đi qua trạm thu phí, thẻ thu phí trên xe sẽ được đọc để xác định số tiền phải trả.

*Bước 2:* Hệ thống thu phí tự động sẽ lấy thông tin về xe và thẻ thu phí của khách hàng từ các bảng dữ liệu tương ứng.

*Bước 3:* Hệ thống sẽ tính toán số tiền phải trả dựa trên thông tin về tuyến đường, loại phương tiện và khoảng cách đi qua trạm thu phí.

*Bước 4:* Hệ thống sẽ trích tiền từ thẻ thu phí của khách hàng và cập nhật số dư thẻ.

*Bước 5:* Hệ thống ghi lại thông tin về giao dịch thu phí tự động vào bảng VETHU.

*Bước 6:* Khách hàng có thể xem thông tin về các giao dịch thu phí tự động và số dư thẻ thu phí trên ứng dụng hoặc trên trang web của hệ thống.

# **PHÂN TÍCH THIẾT KẾ HỆ THỐNG, CHỨC NĂNG**



## Mô tả cơ sở dữ liệu:

**Bảng TINHTHANH:**

MaTinh (VARCHAR2(6), PRIMARY KEY): Mã tỉnh thành

TenTinh (VARCHAR2(25)): Tên tỉnh thành

**Bảng TRAMTHUPHIBOT:**

MaTram (VARCHAR2(6), PRIMARY KEY): Mã trạm thu phí

TenTram (VARCHAR2(25)): Tên trạm thu phí

DiaChi (VARCHAR2(100)): Địa chỉ trạm thu phí

MaTinh (VARCHAR2(6), FOREIGN KEY): Mã tỉnh thành

TuyenDuong (VARCHAR2(50)): Tuyến đường

**Bảng KHACHHANG:**

MaKH (VARCHAR2(6), PRIMARY KEY): Mã khách hàng

HoKH (VARCHAR2(20)): Họ của khách hàng

TenKH (VARCHAR2(25)): Tên của khách hàng

NgaySinhKH (DATE): Ngày sinh của khách hàng

GioiTinhKH (VARCHAR2(25)): Giới tính của khách hàng

DiaChiKH (VARCHAR2(100)): Địa chỉ của khách hàng

EmailKH (VARCHAR2(25)): Email của khách hàng

SDTKH (VARCHAR2(20)): Số điện thoại của khách hàng

CCCDKH (VARCHAR2(20)): Số CCCD của khách hàng

**Bảng XE:**

MaXe (VARCHAR2(6), PRIMARY KEY): Mã xe

MaKH (VARCHAR2(6), FOREIGN KEY): Mã khách hàng

LoaiXe (VARCHAR2(20)): Loại xe

Mau (VARCHAR2(20)): Màu xe

BienSoXe (VARCHAR2(20)): Biển số xe

**Bảng THETAG:**

MaVachThe (VARCHAR2(6), PRIMARY KEY): Mã vạch thẻ

MaXe (VARCHAR2(6), FOREIGN KEY): Mã xe

MaKH (VARCHAR2(6), FOREIGN KEY): Mã khách hàng

SoDuTK (NUMBER(12)): Số dư trong tài khoản

**Bảng VETHU:**

MaVeThu (VARCHAR2(6), PRIMARY KEY): Mã vé thu

MaTram (VARCHAR2(6), FOREIGN KEY): Mã trạm thu phí

MaVachThe (VARCHAR2(6), FOREIGN KEY): Mã vạch thẻ

NgayGioThu (DATE): Ngày giờ thu phí

## Lược đồ quan hệ:

TINHTHANH (MaTinh, TenTinh)

TRAMTHUPHIBOT (MaTram, MaTinh TenTram, DiaChi,, TuyenDuong)

KHACHHANG (MaKH, HoKH, TenKH, NgaySinhKH, GioiTinhKH, DiaChiKH, EmailKH, SDTKH, CCCDKH)

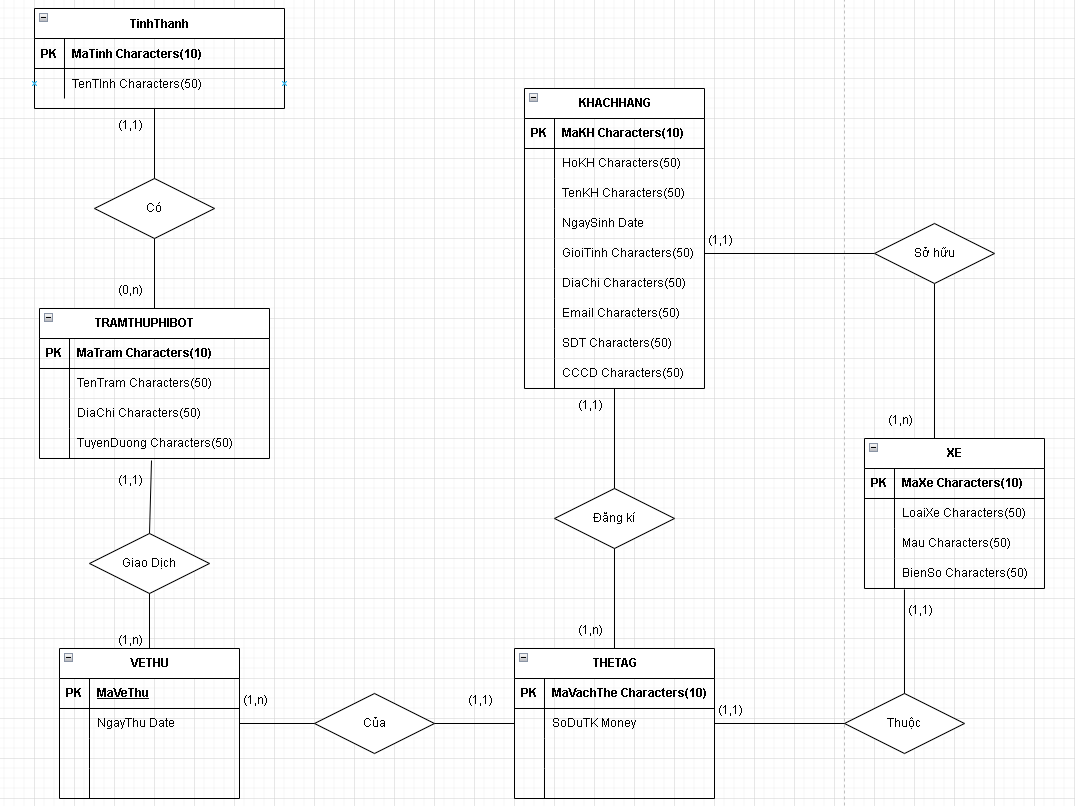
XE (MaXe, MaKH, LoaiXe, Mau, BienSoXe)

THETAG (MaVachThe, MaXe, MaKH, SoDuTK, DiemTichLuy, HSD)

VETHU (MaVeThu, MaTram, MaVachThe, NgayThu)

Khóa ngoại màu đỏ

## Mô Hình ERD:



Hình 1 Biểu đồ ERD

## Đặc tả Use Case:

*Đặc tả use case chức năng đăng nhập*

|  |  |
| --- | --- |
| Tên use case | Đăng nhập |
| Mô tả | Cho phép người dùng đăng nhập vào hệ thống, tùy vào đối tượng đăng nhập sẽ được thực hiện các thao tác khác nhau trong hệ thống |
| Người sử dụng | Nhân viên quản lý, nhân viên |
| Hoạt động | *Use case bắt đầu khi có người dùng đăng nhập vào hệ thống:*   1. Người dùng nhập tên đăng nhập và mật khẩu 2. Hệ thống kiểm tra thông tin đăng nhập nếu đúng thì cho phép người dùng đăng nhập vào hệ thống 3. Hệ thống kiểm tra người dùng và điều hướng người dùng sang các form mà người dùng đó có quyền thao tác 4. Hệ thống thông báo đăng nhập thành công 5. Người dùng có thể chọn đăng xuất trên màn hình   *Trường hợp đăng nhập không thành công do sai tên đăng nhập hoặc mật khẩu:*   1. Hệ thống yêu cầu người dùng nhập lại tên đăng nhập và mật khẩu chính xác 2. Người dùng có thể hủy thao tác đăng nhập |

*Đặc tả use case chức năng tạo khách hàng*

|  |  |
| --- | --- |
| Tên use case | Tạo khách hàng |
| Mô tả | Cho phép người dùng tạo khách hàng khi nhận được yêu cầu |
| Người sử dụng | Nhân viên |
| Hoạt động | Use case bắt đầu khi người dùng nhận được yêu cầu:   1. Nhân viên nhập thông tin khách hàng lên hệ thống |

*Đặc tả use case chức năng quản lý thông tin khách hàng*

|  |  |
| --- | --- |
| Tên use case | Tạo thẻ tag |
| Mô tả | Cho phép người tạo thẻ tagg khi đã nhập thông tin khách hàng |
| Người sử dụng | Nhân viên |
| Hoạt động | *Use case bắt đầu khi người dùng nhận được yêu cầu:*   1. Nhân viên thêm thông tin thẻ tag dựa theo mã khách hàng nhập và sẽ lưu trữ lại thông tin của khách hàng 2. Nhân viên chỉnh sửa hoặc xóa thông tin khách hàng ra khỏi hệ thống 3. Nhân viên xác nhận bằng cách nhấn “OK” để các thay đổi được lưu trữ trên hệ thống |

*Đặc tả use case chức năng quản lý vé thu*

|  |  |
| --- | --- |
| Tên use case | Quản lý vé thu |
| Mô tả | Cho phép quản lý xem lại những xe đã đi qua |
| Người sử dụng | Nhân viên quản lý |
| Hoạt động | *Use case bắt đầu khi người dùng cần xem, thao tác trên bảng thông tin thực đơn:*   1. Quản lý có thể xem thông tin các xe đã đi qua vào thời gian nào qua trạm thu phí nào |

*Đặc tả use case chức năng thống kê doanh thu*

|  |  |
| --- | --- |
| Tên use case | Quản lý trạm thu phí |
| Mô tả | Cho phép xe nào được phép đi qua khi đủ điều kiện |
| Người sử dụng | Hệ thống |
| Hoạt động | *Use case bắt đầu khi:*   1. Xe đi qua hệ thống quét mã sẽ hiện ra mã vạch thẻ ( thẻ tag ) 2. Hệ thống sẽ kiểm tra thông tin đăng ký của xe 3. Hệ thống sẽ kiểm tra số dư tài khoản của xe 4. Hệ thống đưa ra quyết định cho phép xe đi hay không |

## Quản trị hệ thống:

### Không gian lưu trữ Tablespace:

*Các câu lệnh thực thi trong tablespace:*

* Câu lệnh CREATE TABLESPACE: được sử dụng để tạo mới tablespace

CREATE TABLESPACE my\_tablespace DATAFILE '/u01/oracle/data/my\_tablespace.dbf' SIZE 50M;

* Câu lệnh DROP TABLESPACE: được sử dụng để xóa tablespace

DROP TABLESPACE my\_tablespace INCLUDING CONTENTS;

* Câu lệnh RENAME TABLESPACE: được sử dụng để đổi tên tablespace

ALTER TABLESPACE my\_tablespace RENAME TO new\_tablespace\_name;

* Câu lệnh ADD DATAFILE: được sử dụng để thêm datafile vào tablespace

ALTER TABLESPACE my\_tablespace ADD DATAFILE   '/u03/oracle/data/my\_tablespace\_03.dbf' SIZE 200M;

* Câu Lệnh Drop TABLESPACE :được sử dụng để xóa tablespace

Drop TABLESPACE tbs4 INCLUDING CONTENTS and DATAFILES;

* Câu lệnh DROP DATAFILE: được sử dụng để xóa datafile khỏi tablespace

ALTER DATABASE DATAFILE '/u03/oracle/data/my\_tablespace\_03.dbf' OFFLINE DROP;

* Câu lệnh RESIZE DATAFILE: được sử dụng để thay đổi kích thước datafile

ALTER DATABASE DATAFILE '/u02/oracle/data/my\_tablespace\_02.dbf' RESIZE 150M;

*Câu lệnh thực thi để hiển thị thông tin về tablespace*

* Hiển thị danh sách các tablespace

SELECT tablespace\_name FROM dba\_tablespaces;

* Hiển thị dung lượng còn trống của các tablespace

SELECT tablespace\_name, SUM(bytes)/1024/1024 "Free Size (MB)"

FROM dba\_free\_space

GROUP BY tablespace\_name;

* Hiển thị tên và dung lượng của các tablespace

SELECT tablespace\_name, sum(bytes)/1024/1024 AS "Size (MB)" FROM dba\_data\_files GROUP BY tablespace\_name;

* Hiển Thị dung lượng sử dụng của mỗi tablespace

SELECT tablespace\_name, used\_space\_mb

FROM (SELECT tablespace\_name, SUM(bytes) / 1024 / 1024 AS used\_space\_mb

FROM dba\_segments

GROUP BY tablespace\_name)

ORDER BY used\_space\_mb DESC;

* Hiển Thị tổng dung lượng sử dụng của toàn bộ tablespace

SELECT SUM(bytes) / 1024 / 1024 AS used\_space\_mb

FROM dba\_segments;

* Hiển Thị tổng dung lượng của toàn bộ tablespace

SELECT SUM(bytes) / 1024 / 1024 AS total\_space\_mb

FROM dba\_data\_files;

### Thông tin sga, pga, process, instance, database, datafile, control files, spfile

*Để kiểm tra thông tin các process:*

select \* from v$process.

*Để xem thông tin sga:*

select \* from v$sga

*Để xem thông tin pga:*

select \* from dba\_hist\_pgastat

*Để xem thông tin database:*

select \* from v$database

*Để xem thông tin data file:*

select \* from v$datafile

*Để xem thông tin control file:*

select \* from v$controlfile

*Để xem thông tin spfile*

select \* from sys.v\_$parameter where name = 'spfile'

*Để xem thông tin instance:*

select \* from v$instance

### Phiên đăng nhập

*Để kiểm tra thông tin session, ta có thể sử dụng câu lệnh sau:*

* Xem thông tin các session hiện tại của chương trình.

Select \* from v$session;

* Xem Session chạy những process nào

select p.pname, s.username, p.username, s.serial#, s.machine, s.program from v$process p, v$session s where s.creator\_addr = p.addr;

* Xem các session của user của chương trình.

Select inst\_id i# ,SID, serial#, username, machine, program, status from gv$session where type != 'BACKGROUND';

*Để hủy một session, ta dùng câu lệnh:*

Alter system kill session '241,11068';

Trong đó: 241 : là SID của session.

11068 : là serial#.

### Quản lý các user

*Để xem thông tin liên quan đến một user nào đó được chỉ định:*

select \* from dba\_users where username = 'ten\_username';

*Để xem thông tin tất cả user:*

select \* from dba\_users

*Xem thông tin về thời hạn và trạng thái lock của mật khẩu:*

SELECT username, password, account\_status, lock\_date, expiry\_date from dba\_users;

*Xem thông tin profile của mật khẩu:*

SELECT \* FROM dba\_profiles WHERE resource\_type='PASSWORD';

*Xem thông tin profile của tài nguyên:*

SELECT \* FROM dba\_profiles WHERE resource\_type='KERNEL';

### Policy và Audit

### Policy

*Kiểm tra các policy hiện có:*

select \* from dba\_policies BEGIN

DBMS\_FGA.ADD\_POLICY(

object\_schema => 'hr',

object\_name => 'KHACHHANG',

policy\_name => 'KHACHHANG\_audit\_policy',

statement\_types => 'DELETE');

END;

*Kiểm tra policy người dùng tạo:*

select \* from user\_policies

* + - 1. **Fine-Grained Auditing – FGA**

Fine-Grained Auditing do package DBMS\_FGA quản lý:

* Kiểm toán những truy cập dữ liệu dựa theo nội dung.
* Kiểm toán được đến cấp hàng và cột.
* Các câu lệnh INSERT, UPDATE, DELETE thường được kiểm toán. Câu lệnh SELECT ít được kiểm toán hơn do chi phí cao (được sử dụng thường xuyên)

Vd:

--Audit policy được quyền thao tác trên bảng nhân viên

BEGIN

DBMS\_FGA.ADD\_POLICY(

object\_schema => 'hr',

object\_name => 'NHANVIEN',

policy\_name => 'NHANVIEN\_audit\_policy',

--enable => TRUE,

statement\_types => 'INSERT, UPDATE, SELECT');

END;

--drop policy

BEGIN

DBMS\_FGA.DROP\_POLICY(

object\_schema => 'hr',

object\_name => 'NHANVIEN',

policy\_name => 'NHANVIEN\_audit\_policy');

END;

### Thiết lập

Các vấn đề cần kiểm toán trong đồ án:

* 1. Kiểm toán khi thêm , hoặc xóa , hoặc sửa thẻ tag: vì nếu thêm thẻ tag sẽ phải nạp tiền vào tài khoản nếu không kiểm soát được không biết nhân viên có trung thực hay không nên cần phải kiểm toán

Thiết lập

--audit policy duoc quyen thêm xóa sửa tren bang THETAG

BEGIN

DBMS\_FGA.ADD\_POLICY(

object\_schema => 'hr',

object\_name => THETAG,

policy\_name => THETAG \_audit\_policy',

statement\_types => 'DELETE';( có thể là INSERT HOẶC UPDATE)

END;

--drop policy

BEGIN

DBMS\_FGA.DROP\_POLICY(

object\_schema => 'hr',

object\_name => THETAG,

policy\_name => THETAG \_audit\_policy');

END;

Kiểm toán khi thêm vé thu :

Thiết lập

--audit policy duoc quyen thêm tren bang VETHU

BEGIN

DBMS\_FGA.ADD\_POLICY(

object\_schema => 'hr',

object\_name => VETHU,

policy\_name => VETHU \_audit\_policy',

statement\_types => ‘INSERT’;

END;

--drop policy

BEGIN

DBMS\_FGA.DROP\_POLICY(

object\_schema => 'hr',

object\_name => VETHU,

policy\_name => VETHU \_audit\_policy');

END;

* 1. Kiểm toán khi thêm thông tin khách hàng: vì thêm thông tin khách hàng làm thay đổi cách thức áp dụng khuyến mãi cho khách hàng đó, nên cần dùng kiểm toán để ghi lại những thao tác này

.

*Thiết lập*

*--audit policy duoc quyen insert tren bang khach hang*

BEGIN

DBMS\_FGA.ADD\_POLICY(

object\_schema => 'hr',

object\_name => 'KHACHHANG',

policy\_name => 'KHACHHANG\_audit\_policy',

-- enable => TRUE,

statement\_types => 'INSERT');

END;

*--drop policy*

BEGIN

DBMS\_FGA.DROP\_POLICY(

object\_schema => 'hr',

object\_name => 'KHACHHANG',

policy\_name => 'KHACHHANG\_audit\_policy');

END;

* 1. Kiểm toán khi xóa, sửa bảng Trạm Thu Phi Bot: khi xóa, sửa bảng Trạm Thu Phi làm thay đổi đến các thông tin của bot có khi là xe đi qua sẽ không nhận nên cần

*Thiết lập:*

*--audit policy duoc quyen update, delete tren bang nhan vien*

BEGIN

DBMS\_FGA.ADD\_POLICY(

object\_schema => 'hr',

object\_name => TRAMTHUPHIBOT,

policy\_name => TRAMTHUPHIBOT \_audit\_policy',

statement\_types => 'UPDATE, DELETE');

END;

*--drop policy*

BEGIN

DBMS\_FGA.DROP\_POLICY(

object\_schema => 'hr',

object\_name => TRAMTHUPHIBOT,

policy\_name => TRAMTHUPHIBOT \_audit\_policy');

END;

* + - 1. **Procedure**

**Thêm VÉ THU**

create or replace PROCEDURE INSERT\_VETHU (

MaVeThu VARCHAR2,

MaTram VARCHAR2,

MaVachThe VARCHAR,

NgayGioThu DATE

)

AS

BEGIN

INSERT INTO VETHU (MaVeThu, MaTram, MaVachThe, NgayGioThu)

VALUES (maVeThu, maTram, maVachThe, ngayGioThu);

END INSERT\_VETHU;

**THÊM KHÁCH HÀNG**

create or replace PROCEDURE THEM\_KHACH\_HANG (

p\_MaKH IN VARCHAR2,

p\_HoKH IN VARCHAR2,

p\_TenKH IN VARCHAR2,

p\_NgaySinhKH IN DATE,

p\_GioiTinhKH IN VARCHAR2,

p\_DiaChiKH IN VARCHAR2,

p\_EmailKH IN VARCHAR2,

p\_SDTKH IN VARCHAR2,

p\_CCCDKH IN VARCHAR2

)

AS

BEGIN

INSERT INTO KHACHHANG (

MaKH,

HoKH,

TenKH,

NgaySinhKH,

GioiTinhKH,

DiaChiKH,

EmailKH,

SDTKH,

CCCDKH

) VALUES (

p\_MaKH,

p\_HoKH,

p\_TenKH,

p\_NgaySinhKH,

p\_GioiTinhKH,

p\_DiaChiKH,

p\_EmailKH,

p\_SDTKH,

p\_CCCDKH

);

END THEM\_KHACH\_HANG;

**-- THÊM TRẠM THU PHÍ**

CREATE OR REPLACE PROCEDURE insert\_tramthuphibot (

Matram in VARCHAR2,

TenTram IN VARCHAR2(25),

DiaChi IN VARCHAR(100),

MaTinh IN VARCHAR(6),

TuyenDuong IN VARCHAR(50))

IS

BEGIN

INSERT INTO TRAMTHUPHIBOT VALUES (MaTram, TenTram, DiaChi, MaTinh, TuyenDuong) ;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE(' Thêm dữ liệu thành công. ');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Lỗi: ' || SQLCODE || ' - ' || SQLERRM);

END insert\_tramthuphibot;

\* Hiển thị dữ liệu:

CREATE OR REPLACE PROCEDURE HIENTHI\_DU\_LIEU\_TRAM\_THU\_PHI\_BOT

IS

BEGIN

FOR r IN (SELECT \* FROM TRAMTHUPHIBOT) LOOP

DBMS\_OUTPUT.PUT\_LINE('Ma: ' || r.MaTram ||

', Ten: ' || r.TenTram ||

', DiaChi: ' || r.DiaChi||

',MaTinh: ' || r.MaTinh ||

', TuyenDuong: ' || r.TuyenDuong||);

END LOOP;

END;

* + - 1. **Trigger**

**Chặn người dùng khi xóa lần trong 1 giờ**

create or replace TRIGGER hr.trg\_audit\_delete\_KH

BEFORE DELETE ON hr.KhachHang

FOR EACH ROW

DECLARE

cnt NUMBER;

BEGIN

SELECT COUNT(\*) INTO cnt

FROM dba\_fga\_audit\_trail

WHERE db\_user = USER AND statement\_type = 'DELETE'

AND (SYSDATE - timestamp) \* 24 \* 60 <= 1440;-- trong vòng một giờ qua

-- nếu user đã xóa quá hạn mức quy định thì cảnh báo và ngắt kết nối

IF cnt > 3 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Bạn đã xóa quá 3 lần trong vòng 1 giờ qua.');

EXECUTE IMMEDIATE 'ALTER SYSTEM KILL SESSION ''' || SYS\_CONTEXT('USERENV', 'SID') || ', ' || SYS\_CONTEXT('USERENV', 'SERIAL#') || ''' IMMEDIATE';

END IF;

-- lưu lại thông tin thao tác cuối cùng

INSERT INTO HR.delete\_history\_KhachHang(DB\_USER, TIMESTAMP, STATEMENT\_TYPE,SQL\_TEXT)

SELECT DB\_USER, TIMESTAMP, STATEMENT\_TYPE,SQL\_TEXT

FROM dba\_fga\_audit\_trail

ORDER BY TIMESTAMP DESC FETCH FIRST 1 ROW ONLY;

END;

* + - 1. **Cursor**

**Cursor tính tổng khách hàng**

DECLARE

total\_customers NUMBER;

CURSOR c\_customers IS

SELECT COUNT(\*) AS total

FROM KHACHHANG;

BEGIN

OPEN c\_customers;

FETCH c\_customers INTO total\_customers;

DBMS\_OUTPUT.PUT\_LINE('Total number of customers: ' || total\_customers);

CLOSE c\_customers;

END;

**Cursor tìm kiếm khách hàng**

create or replace PROCEDURE TimKiemKhachHang (p\_TenKH IN VARCHAR2)

IS

cursor cur\_KhachHang is

SELECT \* FROM KHACHHANG WHERE TenKH = p\_TenKH;

v\_MaKH KHACHHANG.MaKH%TYPE;

v\_HoKH KHACHHANG.HoKH%TYPE;

v\_TenKH KHACHHANG.TenKH%TYPE;

v\_NgaySinhKH KHACHHANG.NgaySinhKH%TYPE;

v\_GioiTinhKH KHACHHANG.GioiTinhKH%TYPE;

v\_DiaChiKH KHACHHANG.DiaChiKH%TYPE;

v\_EmailKH KHACHHANG.EmailKH%TYPE;

v\_SDTKH KHACHHANG.SDTKH%TYPE;

v\_CCCDKH KHACHHANG.CCCDKH%TYPE;

BEGIN

OPEN cur\_KhachHang;

LOOP

FETCH cur\_KhachHang INTO v\_MaKH, v\_HoKH, v\_TenKH, v\_NgaySinhKH, v\_GioiTinhKH, v\_DiaChiKH, v\_EmailKH, v\_SDTKH, v\_CCCDKH;

EXIT WHEN cur\_KhachHang%NOTFOUND;

-- Thực hiện các hành động khác với khách hàng được tìm thấy ở đây

DBMS\_OUTPUT.PUT\_LINE(v\_MaKH || ', ' || v\_HoKH || ' ' || v\_TenKH || ', ' || v\_NgaySinhKH || ', ' || v\_GioiTinhKH || ', ' || v\_DiaChiKH || ', ' || v\_EmailKH || ', ' || v\_SDTKH || ', ' || v\_CCCDKH);

END LOOP;

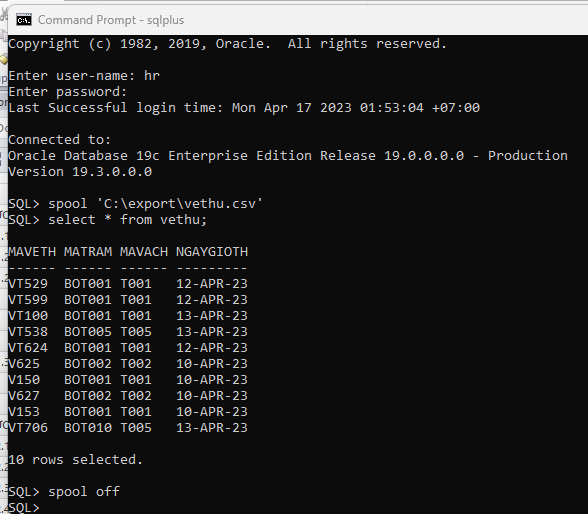
CLOSE cur\_KhachHang;

END;

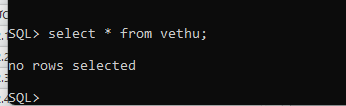
### Function

### Export và Import

**Export spool xuất dữ liệu từ bảng vé thu**

****

**Export**

****

Bây giờ thử xóa hết dữ liệu ở bảng vé thu sau đó import vào

### Package

# **BẢO MẬT**

# **CÀI ĐẶT ỨNG DỤNG**

# **TỔNG KÉT**