KHOA CÔNG NGHỆ THÔNG TIN

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

CSC12001

AN TOÀN VÀ BẢO MẬT DỮ LIỆU TRONG HỆ THỐNG THÔNG TIN

Giáo viên hướng dẫn

Phạm Thị Bạch Huệ

18CLC HTTT – Học kì II 2020 – 2021

DANH SÁCH THÀNH VIÊN

STT	MSSV	Họ Tên
1	18127241	Nguyễn Đăng Triều
2	18127274	Nguyễn Lê Đức Hoàng
3	18127086	Lê Thị Thùy Dương

BẢNG PHÂN CÔNG CÔNG VIỆC

Thành viên	Công việc	Đóng góp
Nguyễn Đăng Triều	PH1: Viết proc: - Cấp quyền cho role/user - Thu hồi quyền từ role/user - Cho phép kiểm tra quyền của các chủ thể vừa được cấp quyền. - Cho phép chỉnh sửa quyền của user/ role PH2: - Áp dụng cơ chế bảo mật VPD, Mã hóa.	33%
Nguyễn Lê Đức Hoàng	PH1: - Tổng hợp code và Thiết kế giao diện. PH2: - Áp dụng cơ chế bảo mật RBAC, FGA Audit Thiết kế giao diện	33%
Lê Thị Thùy Dương	PH1: Viết proc: - Xem danh sách người dùng - Thông tin về quyền (privileges) của mỗi user/ role	33%

	- Cho phép tạo mới, xóa, sửa (hiệu chỉnh) user hoặc role PH2: - Áp dụng cơ chế bảo mật DAC, Standard audit	

YÊU CẦU ĐỒ ÁN

PHÂN HỆ 1: DÀNH CHO NGƯỜI QUẢN TRỊ CƠ SỞ DỮ LIỆU

Sinh viên hãy xây dựng ứng dụng cho phép các người dùng có quyền quản trị thực hiện công việc sau:

- Xem danh sách người dùng trong hệ thống.

- Thông tin về quyền (privileges) của mỗi user/ role trên các đối tượng dữ liệu.

- Cho phép tạo mới, xóa, sửa (hiệu chỉnh) user hoặc role.

- Cho phép thực hiện việc cấp quyền: cấp quyền cho user, cấp quyền cho role, cấp role cho user. Quá trình cấp quyền có tùy chọn là có cho phép người được cấp quyền có thể cấp quyền đó cho user/ role khác hay không (có chỉ định WITH GRANT OPTION hay không). Quyền, select, update thì cho phép phân quyền tinh đến mức cột; quyền insert, delete thì không.

- Cho phép thu hồi quyền từ người dùng/ role.

- Cho phép kiểm tra quyền của các chủ thể vừa được cấp quyền.

- Cho phép chỉnh sửa quyền của user/ role.

Sinh viên hãy thực hiện chức năng ghi nhật ký hệ thống (chỉ yêu cầu thực hiện mức HQT CSDL Oracle):

- Admin có quyền enable/ disable việc ghi nhật ký toàn hệ thống.

- Admin được chỉ định ghi nhật ký của những hành động thực hiện bởi những user nào trên những đối tượng cụ thể. Các hành động đó là: đăng nhập, thay đổi thông tin user account, select, insert, update, delete, execute.; các đối tượng là table, view, stored procedure, function. Admin cũng được quyền chọn ghi nhật ký hành động được thực hiện thành công hay không thành công.

- Kiểm tra dữ liệu nhật ký hệ thống. Sinh viên nên đề ra một số kịch bản theo dõi hệ thống để phân tích dữ liệu nhật ký.

PHÂN HỆ 2: QUẢN LÝ THÔNG TIN CỦA MỘT BỆNH VIỆN

Một bệnh viện quy mô vừa có những vai trò sau: bộ phận quản lý, bộ phận tiếp tân và điều phối bệnh, bác sĩ điều trị, phòng tài vụ, phòng bán thuốc và bộ phận kế toán. Bệnh nhân đến bệnh viện sẽ gặp bộ phận tiếp tân và điều phối bệnh để khai bệnh ban đầu gồm tên, năm sinh, địa chỉ liên lạc, số điện thoại, triệu chứng bệnh. Nếu bệnh nhân trước đó đã khám bệnh thì đọc mã khám bệnh thì thông tin bệnh nhân đã có và không cần phải nhập lại. Sau khi hoàn tất giai đoạn tiếp bệnh, nhân viên tiếp tân chỉ định phòng khám và bác sĩ khám. Tại phòng tài vụ, nhân viên phòng tài vụ nhìn thấy thông tin khám bệnh của bệnh nhân mới sẽ thu tiền khám của bệnh nhân và hướng dẫn bệnh nhân đến gặp bác sĩ. Sau khi xem bệnh, bác sĩ chỉ định và ghi nhận vào CSDL liên quan đến bệnh nhân đó là phải dùng thuốc gì, hoặc phải tiếp tục làm những thủ tục xét nghiệm hoặc chẩn đoán hình ảnh nào. Nhân viên phòng tài vụ căn cứ vào đó thu tiền trước khi bệnh nhân được xét nghiệm hoặc chụp hình theo yêu cầu của bác sĩ. Bộ phận tiếp tân và điều phối bệnh dựa vào dữ liệu của hệ thống ghi lại yêu cầu của bác sĩ sẽ ghi lại trên CSDL thông tin điều phối bệnh vào các phòng liên quan và hướng dẫn bệnh nhân vào phòng nào gặp bác sĩ nào. Sau khi hoàn tất các yêu cầu, bệnh nhân mang kết quả về cho bác sĩ khám bệnh ban đầu đọc kết quả và đề nghị dùng thuốc theo toa bác sĩ kê. Nhân viên phòng thuốc căn cứ vào đó bán thuốc cho bệnh nhân.

Chính sách bảo mật trong ứng dụng trên được mô tả như sau:

• Thành viên của bộ phận quản lý được chia ra làm 3 nhóm: nhóm quản lý tài nguyên và nhân sự (phòng ban, bác sĩ, nhân viên, chấm công), nhóm quản lý tài vụ (đơn giá các loại dịch vụ khám bệnh, đơn giá thuốc), và nhóm quản lý chuyên môn. Nhóm quản lý tài nguyên nhân sự chỉ được thêm, xóa, sửa các thông tin trong cách danh mục như: phòng ban, bác sĩ, nhân viên trong từng phòng ban, bác sĩ nào trực phòng nào vào thời gian nào, ...và được xem tất cả các thông tin khác kể cả thông tin nhân viên kế toán tạo ra nhưng không được quyền sửa. Nhóm quản lý tài vụ chỉ được nhập mới chỉnh sửa các thông tin liên quan, những thông tin khác được quyền xem tất cả nhưng không được phép sửa. Nhóm quản lý chuyên môn được xem tất cả thông tin trong đó có thông tin điều trị bệnh của các bác sĩ để theo dõi về chuyên môn của

bệnh viện và có chiến lược trong tương lai mà không được chỉnh sửa bất cứ thông tin nào.

• Bộ phận tiếp tân và điều phối bệnh được quyền thêm, xóa, sửa, tìm kiếm thông tin bệnh nhân, được điều phối bệnh nhưng không thể xem các thông tin liên quan đến số tiền cho từng thủ tục khám, xét nghiệm hoặc chụp hình hoặc thông tin thuốc điều trị bệnh cho bệnh nhân.

• Nhân viên phòng tài vụ chỉ nhìn thấy các thủ tục mà bác sĩ yêu cầu bệnh nhân phải làm khi điều trị bệnh, thông tin mà bộ phận điều phối bệnh đã điều phối và tính tiền. Nhân viên phòng tài vụ chỉ được cập nhật số tiền phải trả cho từng chi tiết khám trị bệnh của bệnh nhân mà không được chỉnh sửa bất cứ thông tin gì.

• Bác sĩ: chỉ có thể thêm hoặc sửa thông tin liên quan đến việc điều trị bệnh và các loại thuốc phải dùng, liều dùng cho bệnh nhân mà bác sĩ chịu trách nhiệm điều trị. Bác sĩ không được xem hoặc chỉnh sửa thông tin khác của những bệnh nhân do bác sĩ khác điều trị hoặc những thông tin khác trong hệ thống.

• Nhân viên bộ phận bán thuốc: chỉ có thể nhìn thấy toa thuốc mà bác sĩ kê cho từng bệnh nhân để tính tiền thuốc cho bệnh nhân mà không thể xem được bệnh nhân bệnh gì hay bất cứ thông tin gì khác.

• Nhân viên kế toán: tính lương cho các bác sĩ và các nhân viên khác dựa vào lương cơ bản, phụ cấp, số ngày công. Nhân viên kế toán không nhìn thấy bất cứ thông tin gì trong hệ thống liên quan đến quá trình điều trị bệnh cho bệnh nhân của những bộ phận liên quan.

Yêu cầu:

1. Sinh viên tự thiết kế mô hình dữ liệu và tạo dữ liệu thử cho ứng dụng trên. Hãy dùng các cơ chế bảo mật đã học của Oracle để hiện thực các cơ chế bảo mật đề ra.

2. Sinh viên hãy đề ra bối cảnh sử dụng cơ chế mã hóa trong ứng dụng trên, và dùng thư viện hỗ trợ mã dữ liệu của Oracle. Cho biết mục đích, đối tượng cần bảo vệ dữ liệu bằng phương pháp mã hóa, phương pháp quản lý khóa.

3. Sinh viên hãy đề ra bối cảnh sử dụng cơ chế OLS của Oracle. Nhãn gồm đầy đủ 3 thành phần: level, compartment và group. Hãy gán nhãn cho dữ liệu, người dùng và minh họa chính sách bảo mật đã cài đặt.

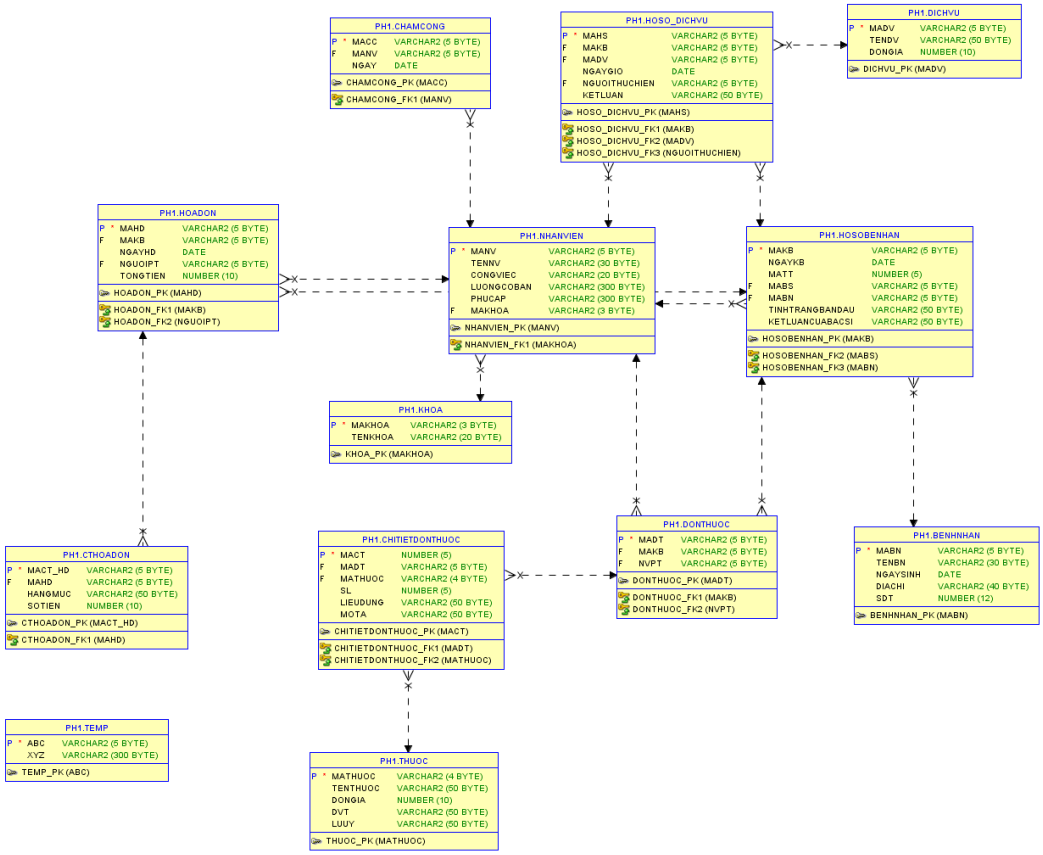
4. Nếu sinh viên cài đặt thêm các chính sách bảo mật có ứng dụng thực tế trong ứng dụng đã cho sẽ được xem xét điểm.

MỤC LỤC

NỘI DUNG ĐỒ ÁN

I. THIẾT KẾ MÔ HÌNH DỮ LIỆU

1. Sơ đồ lớp (Class Diagram)



2. Thiết kế bảng dữ liệu

2.1. Bảng Khoa

```
CREATE TABLE KHOA(
        MAKHOA      VARCHAR2(3) PRIMARY KEY,
        TENKHOA      VARCHAR2(20)
);
```

2.2. Bảng Nhân viên

```
CREATE TABLE NHANVIEN(
        MANV      VARCHAR2(5) PRIMARY KEY,
        TENNV      VARCHAR2(30),
        CONGVIEC    VARCHAR2(20),
        LUONGCOBAN        NUMBER(10),
        PHUCAP     NUMBER(10),
        MAKHOA     VARCHAR2(3)
);

ALTER TABLE NHANVIEN ADD CONSTRAINT NHANVIEN_FK1 FOREIGN KEY(MAKHOA) REFERENCES
KHOA(MAKHOA)ON DELETE CASCADE ENABLE;
```

2.3. Bảng Bệnh nhân

```
CREATE TABLE BENHNHAN(
        MABN      VARCHAR2(5) PRIMARY KEY,
        TENBN      VARCHAR2(30),
        NGAYSINH        DATE,
        DIACHI        VARCHAR2(40),
        SDT     NUMBER(12)
);
```

2.4. Bảng Hồ sơ bện án

```
CREATE TABLE HOSOBENHAN(
        MAKB      VARCHAR2(5) PRIMARY KEY,
        NGAYKB      DATE,
        MATT        NUMBER(5),
        MABS        VARCHAR2(5),
        MABN       VARCHAR2(5),
        TINHTRANGBANDAU        VARCHAR2(50),
        KETLUANCUABACSI        VARCHAR2(50)
);

ALTER TABLE HOSOBENHAN ADD CONSTRAINT HOSOBENHAN_FK2 FOREIGN KEY(MABS)REFERENCES
NHANVIEN(MANV)ON DELETE CASCADE ENABLE;

ALTER TABLE HOSOBENHAN ADD CONSTRAINT HOSOBENHAN_FK3 FOREIGN KEY(MABN)REFERENCES
BENHNHAN(MABN)ON DELETE CASCADE ENABLE;
```

2.5. Bảng Hồ sơ dịch vụ

```
CREATE TABLE HOSO_DICHVU(
        MAHS      VARCHAR2(5) PRIMARY KEY,
        MAKB      VARCHAR2(5),
        MADV      VARCHAR2(5),
        NGAYGIO         DATE,
        NGUOITHUCHIEN  VARCHAR2(5),
        KETLUAN         VARCHAR2(50)
);

ALTER TABLE HOSO_DICHVU ADD CONSTRAINT HOSO_DICHVU_FK1 FOREIGN KEY(MAKB)REFERENCES
HOSOBENHAN(MAKB)ON DELETE CASCADE ENABLE;
ALTER TABLE HOSO_DICHVU ADD CONSTRAINT HOSO_DICHVU_FK2 FOREIGN KEY(MADV)REFERENCES
DICHVU(MADV)ON DELETE CASCADE ENABLE;
ALTER TABLE HOSO_DICHVU ADD CONSTRAINT HOSO_DICHVU_FK3 FOREIGN
KEY(NGUOITHUCHIEN)REFERENCES NHANVIEN(MANV)ON DELETE CASCADE ENABLE;
```

2.6. Bảng Dịch vụ

```
CREATE TABLE DICHVU(
        MADV       VARCHAR2(5) PRIMARY KEY,
        TENDV      VARCHAR2(50),
        DONGIA     NUMBER(10)
);
```

2.7. Bảng Đơn thuốc

```
CREATE TABLE DONTHUOC(
        MADT      VARCHAR2(5) PRIMARY KEY,
        MAKB      VARCHAR2(5),
        NVPT     VARCHAR2(5)
);

ALTER TABLE DONTHUOC ADD CONSTRAINT DONTHUOC_FK1 FOREIGN KEY(MAKB)REFERENCES
HOSOBENHAN(MAKB)ON DELETE CASCADE ENABLE;
ALTER TABLE DONTHUOC ADD CONSTRAINT DONTHUOC_FK2 FOREIGN KEY(NVPT)REFERENCES
NHANVIEN(MANV)ON DELETE CASCADE ENABLE;
```

2.8. Bảng Chi tiết đơn thuốc

```
CREATE TABLE CHITIETDONTHUOC(
        MACT      NUMBER(5) PRIMARY KEY,
        MADT      VARCHAR2(5),
        MATHUOC      VARCHAR2(4),
        SL        NUMBER(5),
        LIEUDUNG  VARCHAR2(50),
        MOTA        VARCHAR2(50)
);
```

```
ALTER TABLE CHITIETDONTHUOC ADD CONSTRAINT CHITIETDONTHUOC_FK1 FOREIGN
KEY(MADT)REFERENCES DONTHUOC(MADT)ON DELETE CASCADE ENABLE;
ALTER TABLE CHITIETDONTHUOC ADD CONSTRAINT CHITIETDONTHUOC_FK2 FOREIGN
KEY(MATHUOC)REFERENCES THUOC(MATHUOC)ON DELETE CASCADE ENABLE;
```

2.9. Bảng Thuốc

```
CREATE TABLE THUOC(
        MATHUOC      VARCHAR2(4) PRIMARY KEY,
        TENTHUOC      VARCHAR2(50),
        DONGIA      NUMBER(10),
        DVT       VARCHAR2(50),
        LUUY  VARCHAR2(50)
);
```

2.10. Bảng Hóa đơn

```
CREATE TABLE HOADON(
        MAHD      VARCHAR2(5) PRIMARY KEY,
        MAKB      VARCHAR2(5),
        NGAYHD      DATE,
        NGUOIPT   VARCHAR2(5),
        TONGTIEN    NUMBER(10)
);
```

```
ALTER TABLE HOADON ADD CONSTRAINT HOADON_FK1 FOREIGN KEY(MAKB)REFERENCES
HOSOBENHAN(MAKB)ON DELETE CASCADE ENABLE;
ALTER TABLE HOADON ADD CONSTRAINT HOADON_FK2 FOREIGN KEY(NGUOIPT)REFERENCES
NHANVIEN(MANV)ON DELETE CASCADE ENABLE;
```

2.11. Bảng chi tiết hóa đơn

```
CREATE TABLE CTHOADON(
        MACT_HD      VARCHAR2(5) PRIMARY KEY,
        MAHD       VARCHAR2(5),
        HANGMUC     VARCHAR2(50),
        SOTIEN     NUMBER(10)
);

ALTER TABLE CTHOADON ADD CONSTRAINT CTHOADON_FK1 FOREIGN KEY(MAHD)REFERENCES
HOADON(MAHD)ON DELETE CASCADE ENABLE;
```

2.12. Bảng chấm công

```
CREATE TABLE CHAMCONG(
        MACC       VARCHAR2(5) PRIMARY KEY,
        MANV       VARCHAR2(5),
        NGAY     DATE
);

ALTER TABLE CHAMCONG ADD CONSTRAINT CHAMCONG_FK1 FOREIGN KEY(MANV)REFERENCES
NHANVIEN(MANV)ON DELETE CASCADE ENABLE;
```

II. HỆ THỐNG DÀNH CHO NGƯỜI QUẢN TRỊ BẢO MẬT

1. Xem danh sách người dùng hệ thống

```
create or replace procedure sp_Xemdanhsachnguoidung
as
    temp sys_refcursor;
begin
open temp for
    SELECT * FROM all_users;
    DBMS_SQL.RETURN_RESULT(temp);
end;
```

2. Thông tin về quyền của mỗi user/role trên các đối tượng dữ liệu

2.1. Hiển thị quyền của user trên các cột của bảng

```
create or replace procedure sp_UserXemQuyenTrenCot
as
    temp sys_refcursor;
begin
    open temp for
        --SELECT * FROM USER_TAB_PRIVS;
        SELECT GRANTEE, TABLE_NAME, COLUMN_NAME, PRIVILEGE, GRANTABLE FROM USER_COL_PRIVS;
    DBMS_SQL.RETURN_RESULT(temp);
end;
```

2.2. Hiển thị quyền của user/role trên bảng

```
create or replace procedure sp_UserXemQuyenTrenBang
as
    temp sys_refcursor;
begin
    open temp for
        SELECT GRANTEE, TABLE_NAME, PRIVILEGE, GRANTABLE FROM USER_TAB_PRIVS;
    DBMS_SQL.RETURN_RESULT(temp);
end;
```

2.3. Hiển thị role của user

```
create or replace procedure sp_XemRoleCuaUser
as
    temp sys_refcursor;
begin
    open temp for
        SELECT USERNAME, GRANTED_ROLE FROM USER_ROLE_PRIVS;
    DBMS_SQL.RETURN_RESULT(temp);
end;
```

2.4. Hiển thị quyền của role trên system

```
create or replace procedure sp_RoleXemQuyenHeThong
as
    temp sys_refcursor;
begin
    open temp for
        SELECT ROLE, PRIVILEGE FROM ROLE_SYS_PRIVS;
    DBMS_SQL.RETURN_RESULT(temp);
end;
```

2.5. Hiển thị quyền của role trên bảng

```
create or replace procedure sp_RoleXemQuyenTrenBang
as
    temp sys_refcursor;
begin
    open temp for
        SELECT ROLE, TABLE_NAME, COLUMN_NAME, PRIVILEGE, GRANTABLE FROM ROLE_TAB_PRIVS;
    DBMS_SQL.RETURN_RESULT(temp);
end;
```

3. Tạo mới, xóa, sữa user hoặc role

3.1. Tạo user

```
create or replace procedure sp_TaoUser(user_name in varchar2, password varchar2, tablespace_name varchar2, quota varchar2)
as
temp varchar2(100);
begin
    temp:= 'CREATE USER ' || user_name || ' IDENTIFIED BY ' || password || ' default tablespace '
    || tablespace_name || ' quota ' || quota || ' on ' || tablespace_name;
    execute immediate temp;
end;
```

3.2. Tạo role

```
create or replace procedure sp_Taorole(role_name in varchar2)
IS
begin
    EXECUTE IMMEDIATE 'ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE';
    EXECUTE IMMEDIATE 'CREATE ROLE ' || role_name;
end;
```

3.3. Xóa user

```
create or replace procedure sp_Xoauser(user_name in varchar2)
as
begin
    EXECUTE IMMEDIATE 'ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE';
    EXECUTE IMMEDIATE 'DROP USER ' || user_name;
end;
```

3.4. Xóa role

```
create or replace procedure sp_Xoarole(role_name in varchar2)
IS
begin
    EXECUTE IMMEDIATE 'ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE';
    EXECUTE IMMEDIATE 'DROP ROLE ' || role_name;
end;
```

3.5. Sữa password user

```
create or replace procedure sp_Suauser(user_name in varchar2, new_user_password in varchar2)
as
begin
    EXECUTE IMMEDIATE 'ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE';
    EXECUTE IMMEDIATE 'ALTER USER ' || user_name || ' IDENTIFIED BY ' || new_user_password;
end;
```

3.6. Chỉnh sữa tablespace của user

```
create or replace procedure sp_UpdateTablespaceUser(user_name in varchar2, new_tablespace varchar2)
as
temp VARCHAR2(500);
temp1 VARCHAR2(100);
BEGIN
    temp1 := 'alter session set "_ORACLE_SCRIPT"=TRUE';
    EXECUTE IMMEDIATE temp1;
    temp := 'alter user ' || user_name || ' default tablespace ' || new_tablespace;
    EXECUTE IMMEDIATE temp;
END;
```

3.7. Chỉnh sữa quota của user

```
create or replace procedure sp_UpdateQuotaUser(user_name in varchar2, new_Quota varchar2, tablespace_name varchar2)
as
temp VARCHAR2(500);
temp1 VARCHAR2(100);
BEGIN
    temp1 := 'alter session set "_ORACLE_SCRIPT"=TRUE';
    EXECUTE IMMEDIATE temp1;
    temp := 'alter user ' || user_name || ' quota ' || new_Quota || ' on ' || tablespace_name;
    EXECUTE IMMEDIATE temp;
END;
```

4. Thực hiện cấp quyền

4.1. Cấp role cho user

```
create or replace procedure sp_GrantRoleToUser(user_name in varchar2, role_name in varchar2)
as
temp VARCHAR2(500);
BEGIN
    temp := 'grant ' || role_name || ' TO ' || user_name;
    EXECUTE IMMEDIATE temp;
END;
```

4.2. Cấp quyền select trên cột

```
create or replace procedure sp_GrantSelectOnTable(user_name in varchar2, table_name in varchar2, column_name in varchar2, grant_option in varchar2)
as
    temp VARCHAR2(500);
    temp1 VARCHAR2(500);
    temp2 VARCHAR2(500);
BEGIN
   if grant_option = 1 then
        temp2 := 'grant insert ' || 'ON ' || table_name || ' TO ' || user_name || ' with grant option';
    else
        temp2 := 'grant insert ' || 'ON ' || table_name || ' TO ' || user_name;
    end if;
    temp := 'CREATE or REPLACE VIEW ' || user_name || table_name || ' AS SELECT ' || column_name || ' FROM ' || table_name;
    EXECUTE IMMEDIATE temp;
    temp1 := 'GRANT SELECT ON ' || user_name || table_name || ' TO ' || user_name;
    EXECUTE IMMEDIATE temp1;
END;
```

4.3. Cấp quyền insert cho user/role

```
create or replace procedure sp_Insert(user_name in varchar2, table_name in varchar2, grant_option in int)
as
temp VARCHAR2(500);
BEGIN
    if grant_option = 1 then
        temp := 'grant insert ' || 'ON ' || table_name || ' TO ' || user_name || ' with grant option';
    else
        temp := 'grant insert ' || 'ON ' || table_name || ' TO ' || user_name;
    end if;
    EXECUTE IMMEDIATE temp;
END;
```

4.4. Cấp quyền select trên cột và grant trên view

```
create or replace procedure sp_GrantSelectOnTable(user_name in varchar2, table_name in varchar2, column_name in varchar2, grant_option in int)
as
    temp VARCHAR2(500);
    temp1 VARCHAR2(500);
BEGIN
    temp := 'CREATE or REPLACE VIEW ' || user_name || table_name || ' AS SELECT ' || column_name || ' FROM ' || table_name;
    EXECUTE IMMEDIATE temp;
    if grant_option=0 then
        temp1 := 'GRANT SELECT ON ' || user_name || table_name || ' TO ' || user_name;
    else
        temp1 := 'GRANT SELECT ON ' || user_name || table_name || ' TO ' || user_name || ' with grant option';
    end if;
    EXECUTE IMMEDIATE temp1;
END;
```

4.5. Cấp select trên table

```
create or replace procedure sp_prSelect(table_name in varchar2, user_name in varchar2)
as
BEGIN
        EXECUTE IMMEDIATE 'GRANT SELECT ON ' || table_name || ' TO ' || user_name;
END;
```

4.6. Cấp quyền delete

```
create or replace procedure sp_Delete(user_name in varchar2, table_name in varchar2, grant_option in int)
as
temp VARCHAR2(500);
BEGIN
    if grant_option=1 then
        temp := 'grant delete ' || 'ON ' || table_name || ' TO ' || user_name || ' with grant option';
    else
        temp := 'grant delete ' || 'ON ' || table_name || ' TO ' || user_name;
    end if;
    EXECUTE IMMEDIATE temp;
END;
```

4.7. Cấp quyền update

```
--Cap quyen update
create or replace NONEDITIONABLE procedure sp_Update(user_name in varchar2, column_name in varchar2, table_name in varchar2, grant_option in int)
as
temp VARCHAR2(500);
BEGIN
    if grant_option=1 then
        temp := 'grant update(' || column_name  || ') ON ' || table_name || ' TO ' || user_name || ' with grant option';
    else
        temp := 'grant update(' || column_name  || ') ON ' || table_name || ' TO ' || user_name;
    end if;
    EXECUTE IMMEDIATE temp;
END;
```

5. Thu hồi quyền của user/role

```
create or replace procedure sp_Revoke(user_name in varchar2, table_name in varchar2, type in int)
as
temp VARCHAR2(500);
BEGIN
    if type = 1 then
        temp := 'revoke select ' || 'ON ' || table_name || ' from ' || user_name;
    elsif type = 2 then
        temp := 'revoke insert ' || 'ON ' || table_name || ' from ' || user_name;
    elsif type = 3 then
        temp := 'revoke update ' || 'ON ' || table_name || ' from ' || user_name;
    elsif type = 4 then
        temp := 'revoke delete ' || 'ON ' || table_name || ' from ' || user_name;
    end if;
    EXECUTE IMMEDIATE temp;
END;
```

6. Kiểm tra quyền

6.1. Kiểm tra quyền update trên cột vừa cấp cho user

```
create or replace procedure sp_UserKTraQuyenTrenCot(user_name varchar2)
as
    temp sys_refcursor;
begin
    open temp for
        --SELECT * FROM USER_TAB_PRIVS;
        SELECT GRANTEE, TABLE_NAME, COLUMN_NAME, PRIVILEGE, GRANTABLE FROM USER_COL_PRIVS where grantee = user_name;
    DBMS_SQL.RETURN_RESULT(temp);
end;
```

6.2. Kiểm tra quyền thêm, xóa, sữa vừa cấp cho user

```
create or replace procedure sp_UserKtraQuyenTrenBang(user_name varchar2)
as
    temp sys_refcursor;
begin
    open temp for
        SELECT GRANTEE, TABLE_NAME, PRIVILEGE, GRANTABLE FROM USER_TAB_PRIVS where grantee = user_name;
    DBMS_SQL.RETURN_RESULT(temp);
end;
```

6.3. Kiểm tra quyền của role vừa được cấp

```
create or replace procedure sp_RoleKtraQuyenTrenBang(role_name varchar2)
as
    temp sys_refcursor;
begin
    open temp for
        SELECT ROLE, TABLE_NAME, COLUMN_NAME, PRIVILEGE, GRANTABLE FROM ROLE_TAB_PRIVS where ROLE = role_name;
    DBMS_SQL.RETURN_RESULT(temp);
end;
```

7. Chỉnh sữa quyền

7.1. Chỉnh sữa quyền update trên cột cho user/role

```
create or replace procedure sp_ChinhSuaUpdateTrenCot(user_name in varchar2, column_name in varchar2, table_name in varchar2, grant_option in int)
as
temp VARCHAR2(500);
BEGIN
    sp_Revoke(user_name,table_name, 3);
    if grant_option=1 then
        temp := 'grant update(' || column_name  || ') ON ' || table_name || ' TO ' || user_name || ' with grant option';
    else
        temp := 'grant update(' || column_name  || ') ON ' || table_name || ' TO ' || user_name;
    end if;
    EXECUTE IMMEDIATE temp;
END;
```

7.2. Chỉnh sữa quyền insert cho user/role

```
create or replace procedure sp_ChinhSuaInsert(user_name in varchar2, table_name in varchar2, grant_option in int)
as
temp VARCHAR2(500);
BEGIN
    sp_Revoke(user_name,table_name, 2);
    if grant_option=1 then
        temp := 'grant insert ' || 'ON ' || table_name || ' TO ' || user_name || ' with grant option';
    else
        temp := 'grant insert ' || 'ON ' || table_name || ' TO ' || user_name;
    end if;
    EXECUTE IMMEDIATE temp;
END;
```

7.3. Chỉnh sữa quyền delele cho user/role

```
create or replace procedure sp_ChinhSuaDelete(user_name in varchar2, table_name in varchar2, grant_option in int)
as
temp VARCHAR2(500);
BEGIN
    sp_Revoke(user_name,table_name, 4);
    if grant_option=1 then
        temp := 'grant delete ' || 'ON ' || table_name || ' TO ' || user_name || ' with grant option';
    else
        temp := 'grant delete ' || 'ON ' || table_name || ' TO ' || user_name;
    end if;
    EXECUTE IMMEDIATE temp;
END;
```

7.4. Chỉnh sữa quyền select trên view

```
create or replace procedure sp_ChinhSuaSelectTrenView(user_name in varchar2, table_name in varchar2, column_name in varchar2, grant_option in int)
as
    temp VARCHAR2(500);
    temp1 VARCHAR2(500);
BEGIN
    sp_Revoke(user_name, user_name || table_name, 1);
    temp := 'CREATE or REPLACE VIEW ' || user_name || table_name || ' AS SELECT ' || column_name || ' FROM ' || table_name;
    EXECUTE IMMEDIATE temp;
    if grant_option=0 then
        temp1 := 'GRANT SELECT ON ' || user_name || table_name || ' TO ' || user_name;
    else
        temp1 := 'GRANT SELECT ON ' || user_name || table_name || ' TO ' || user_name || ' with grant option';
    end if;
    EXECUTE IMMEDIATE temp1;
END;
```

III. HIỆN THỰC CÁC CHÍNH SÁCH BẢO MẬT

1. Các loại chính sách bảo mật, phân tích, phân loại

Chính sách bảo mật	Phân tích	Phân loại
Quản lý nhân sự	Xem thêm xóa sửa bảng nhân viên và chấm công. các bảng còn lại chỉ được xem	DAC, RBAC
Quản lý tài vụ	Xem thêm sửa bảng dịch vụ và bảng thuốc. các bảng còn lại chỉ được xem	DAC, RBAC
Quản lý chuyên môn	Xem tất cả các bảng	DAC, RBAC
Tiếp tân và điều phối	Xem thêm xóa sửa tìm kiếm bảng bệnh nhân, hồ sơ bệnh án(trừ cột kết luận của bác sĩ), hồ sơ dịch vụ(trừ cột kết luận)	DAC, RBAC, Mã hóa
NV phòng tài vụ	Xem bảng hồ sơ bệnh án, hồ sơ dịch vụ. Xem sửa bảng hóa đơn, chi tiết hóa đơn (cột tổng tiền và số tiền ở mỗi bảng)	DAC, RBAC. Mã hóa, Audit
Bác sĩ	Xem sửa bảng Hồ sơ bệnh án (có mã BS của mình) cột Kết luận của BS. Xem thêm sửa bảng hồ sơ dịch vụ, đơn thuốc, chi tiết đơn thuốc có mã KB mình xử lý.	DAC, RBAC, Mã hóa, VPD
NV bán thuốc	Xem sửa bảng đơn thuốc cột nvphát thuốc. Xem bảng chi tiết đơn thuốc để tính tiền cho bệnh nhân (tổng tiền lưu ở bảng đơn thuốc: giá trị suy diễn từ tổng tiền những dòng của chi tiết đơn thuốc)	DAC, RBAC, Mã hóa, VPD
NV kế toán	Xem bảng chấm công. Xem sửa bảng nhân viên (thêm cột lương: thuộc tính suy diễn suy ra từ	DAC, RBAC,

	lương cơ bản phụ cấp và số ngày công)	MAC, Audit, VPD

2. Cài đặt chính sách bảo mật

2.1. DAC

- Thực hiện tạo và cấp quyền cho những nhân viên thuộc bộ phận Bán thuốc và bộ phận Tiếp tân và điều phối bệnh.

NV013 và NV014 là nhân viên thuộc bộ phận bán thuốc.

NV004 và NV005 là nhân viên thuộc bộ phận tiếp tân và điều phối bệnh.

```
--Tao va cap quyen
create user NV013 identified by 123456;
grant connect to NV013;
grant create session to NV013;


create user NV014 identified by 123456;
grant connect to NV014;
grant create session to NV014;


create user NV004 identified by 123456;
grant connect to NV004;
grant create session to NV004;


create user NV005 identified by 123456;
grant connect to NV005;
grant create session to NV005;
```

- Tạo view **Xem_thong_tin_toa_thuoc** chỉ chứa thông tin toa thuốc mà bác sĩ kê cho từng bệnh nhân thông qua view nhân viên bán thuốc có thể biết các loại thuốc, liều dùng cũng như đơn giá từng loại để tiến hành tính tiền thuốc cho bệnh nhân.

```
create or replace view Xem_thong_tin_toa_thuoc
as
    select BENHNHAN.TENBN, THUOC.TENTHUOC, THUOC.DONGIA, CHITIETDONTHUOC.SL, CHITIETDONTHUOC.LIEUDUNG
    from THUOC
    join CHITIETDONTHUOC
    on CHITIETDONTHUOC.MATHUOC = THUOC.MATHUOC
    join DONTHUOC
    on CHITIETDONTHUOC.MADT = DONTHUOC.MADT
    join HOSOBENHAN
    on HOSOBENHAN.MAKB = DONTHUOC.MAKB
    join BENHNHAN
    on BENHNHAN.MABN = HOSOBENHAN.MABN
    join NHANVIEN
    on NHANVIEN.MANV = DONTHUOC.NVPT
    where CONGVIEC = 'NVBanthuoc';
```

- Cấp quyền select cho nhân viên thuộc bộ phận bán thuốc để có thể xem thông tin toa thuốc thông qua view Xem_thong_tin_toa_thuoc

```
grant select on Xem_thong_tin_toa_thuoc  to NV013;
grant select on Xem_thong_tin_toa_thuoc  to NV014;
```

- Thêm một cột TONGTIEN trong bảng DONTHUOC, nhân viên bán thuốc sử dụng cột này để tính và lưu tiền thuốc cho bệnh nhân. View Xem_chinh_sua_don_thuoc chứa thông tin về tổng tiền mà nhân viên bán thuốc tính dựa trên đơn của bệnh nhân.

```
alter table DONTHUOC add TONGTIEN NUMBER(10);

create or replace view Xem_chinh_sua_don_thuoc
as
    select *
    from DONTHUOC;
```

- Cấp quyền select , update cho nhân viên thuộc bộ phận bán thuốc để có thể xem cthông tin thông qua view Xem_chinh_sua_don_thuoc và update cột TONGTIEN, NVPT.

```
grant select on Xem_chinh_sua_don_thuoc to NV013;
grant update (NVPT,TONGTIEN) on Xem_chinh_sua_don_thuoc to NV013;
grant select on Xem_chinh_sua_don_thuoc to NV014;
grant update (NVPT,TONGTIEN) on Xem_chinh_sua_don_thuoc to NV014;
```

- Tạo view Xem_chinh_sua_thong_tin_BN chỉ chứa thông tin bệnh nhân, thông qua view nhân viên tiếp tân và điều phối bệnh có thể xem, chỉnh sửa thông tin bệnh nhận

```
create or replace view Xem_chinh_sua_thong_tin_BN
as
    select *from BENHNHAN;
```

- Cấp quyền xem, chỉnh sữa cho nhân viên tiếp tân và điều phối bệnh.

```
grant select, insert, update on Xem_chinh_sua_thong_tin_BN to NV004;
grant select, insert, update on Xem_chinh_sua_thong_tin_BN to NV005;
```

- Tạo view Xem_ho_so_benh_an thông qua view nhân viên tiếp tân và điều chứa thông tin như tình trạng bệnh ban đầu của bệnh nhân, ngày lập hồ sơ,...

```
create or replace view Xem_ho_so_benh_an
as
    select MAKB, NGAYKB, MATT,MABS, MABN, TINHTRANGBANDAU
    from HOSOBENHAN;
```

- Cấp quyền xem view Xem_ho_so_benh_an cho nhân viên tiếp tân và điều phối bệnh.

```
grant select on Xem_ho_so_benh_an to NV004;
grant select on Xem_ho_so_benh_an to NV005;
```

- Tạo view Xem_ho_so_dich_vu, cho nhân viên tiếp tân và điều phối bệnh để xem các thông tin như dịch vụ đã thực hiện, ngày lập hồ sơ,..

```
create or replace view Xem_ho_so_dich_vu
as
    select MAHS, MAKB, MADV, NGAYGIO, NGUOITHUCHIEN
    from HOSO_DICHVU;
```

- Cấp quyền xem trên view Xem_ho_so_dich_vu cho nhân viên tiếp tân và điều phối bệnh.

```
grant select on Xem_ho_so_dich_vu to NV004;
grant select on Xem_ho_so_dich_vu to NV005;
```

2.2. RABC

- Tạo role QLCM(Quản lý chuyên môn) và role NVKETOAN(Nhân viên kế toán), sau đó tạo user qlcm1 và nvkt1.

```
--tao role quan ly chuyen mon
create role QLCM;
--tao role nhan vien ke toan
create role NVKETOAN
--tao user qlcm1
create user qlcm1 identified by 1;
--tao user nvkt1
create user nvkt1 identified by 1;
```

- Sau đó gán user qlcm1 vào role QLCM và gán user nvkt1 vào role NVKETOAN.

```
grant NVKETOAN to nvkt1;
grant QLCM to qlcm1;
```

- Sau đó ta tạo các proc và cấp quyền cho role NVKETOAN.

```
--proc cấp quyền xem bảng nhân viên.
create or replace procedure sp_Grant_select_nhanvien
as
    temp sys_refcursor;
begin
open temp for
    select MANV, TENNV, CONGVIEC, LUONGCOBAN, PHUCAP, MAKHOA, LUONGCOBAN + PHUCAP AS LUONG from NHANVIEN;
    DBMS_SQL.RETURN_RESULT(temp);
end;
```

```
--proc cấp quyền xem bảng chấm công và sửa bảng nhân viên
create or replace procedure sp_Grant_select_chamcong_update_nhanvien(rolename in varchar2)
as
BEGIN
    EXECUTE IMMEDIATE 'GRANT SELECT ON CHAMCONG TO ' || rolename;
    EXECUTE IMMEDIATE 'GRANT UPDATE ON NHANVIEN TO ' || rolename;
END;
```

```
create or replace NONEDITIONABLE procedure Update_nhanvien(manv_t in varchar2, lcb in varchar2, pc in varchar2)
as
BEGIN
    update nhanvien set luongcoban = Encryption(lcb) where manv = manv_t;
    update nhanvien set phucap = Encryption(pc) where manv = manv_t;
end;
```

- Để có thể xem được bảng nhân viên và tính lương cho các nhân viên ta phải cấp quyền select trên bảng temp là bảng giữ khóa giải mã các thuộc tính lương cơ bản và phụ cấp được mã hóa và quyền chạy proc decryption.

```
grant select on tinhluong to NVKETOAN;
grant execute on decryption to NVKETOAN;
```

- Tạo view xem số ngày công.

```
CREATE or REPLACE VIEW SONGAYCONG AS
SELECT MANV, EXTRACT(MONTH FROM NGAY) THANG, EXTRACT(YEAR FROM NGAY) NAM, COUNT(EXTRACT(DAY FROM NGAY)) SONGAY
FROM CHAMCONG
GROUP BY MANV, EXTRACT(MONTH FROM NGAY), EXTRACT(YEAR FROM NGAY);
```

- Tạo view tính lương.

```
CREATE or REPLACE VIEW TINHLUONG AS
SELECT nv.manv, nv.tennv, nv.luongcoban, nv.phucap, snc.thang, snc.nam, snc.songay, (LUONGCOBAN + PHUCAP + SONGAY*200000) LUONG
FROM GIAIMA_NHANVIEN NV, SONGAYCONG SNC
WHERE NV.MANV=SNC.MANV;
```

- Sau đó ta thực hiện việc gán quyền các view và proc này cho role NVKETOAN.

```
grant select on sp_Grant_select_nhanvien to NVKETOAN;
grant execute on Update_nhanvien to NVKETOAN;
grant execute on sp_Grant_select_chamcong_update_nhanvien to NVKETOAN;
grant select on view SONGAYCONG to NVKETOAN;
grant select on view TINHLUONG to NVKETOAN;
```

- Ở role QLCM ta tạo proc gán quyền xem tất cả các bảng.

```
create or replace procedure sp_Viewalltable(role_name in varchar2)
as
BEGIN
    EXECUTE IMMEDIATE 'GRANT SELECT ON NHANVIEN TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON BENHNHAN TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON CHAMCONG TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON CHITIETDONTHUOC TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON CTHOADON TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON DICHVU TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON DONTHUOC TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON HOADON TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON HOSO_DICHVU TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON HOSOBENHAN TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON KHOA TO ' || role_name;
    EXECUTE IMMEDIATE 'GRANT SELECT ON THUOC TO ' || role_name;
END;
```

- Sau đó gán quyền execute proc này cho role QLCM.

```
grant execute on sp_Viewalltable to QLCM
```

2.3. VPD

Chính sách 1: Bác sĩ chỉ xem, sửa được hồ sơ bệnh án nào do mình phụ trách

- Tạo function

```
--BAC SI CHI DUOC XEM HOSOBENHAN CO MABS CUA MINH
create or replace FUNCTION BacSi1(
p_schema IN VARCHAR2 DEFAULT NULL,
p_object IN VARCHAR2 DEFAULT NULL)
return varchar2 as
begin
    --LOAI USER ADMIN RA KHOI POLICY
    if user = 'PH1' then
        return '1=1';
    else
        return 'MABS=USER';
    END IF;
end;
```

- Tạo policy với function mình vừa cài đặt

```
--TAO POLICY VA GAN FUNCTION BACSI1 VAO
BEGIN
    DBMS_RLS.add_policy
        (object_name => 'HOSOBENHAN',
        policy_name => 'BACSIHOSOBENHAN',
        policy_function => 'BACSI1',
        statement_types => 'SELECT, UPDATE',
        update_check => TRUE);
END;
/
```

Chính sách 2: Bác sĩ khám bệnh chỉ xem được những hồ sơ dịch vụ của những bệnh án mình phụ trách. Bác sĩ thực hiện dịch vụ có thể thêm và sửa những kết luận với mỗi dịch vụ mình thực hiện.

- Do Hồ sơ dịch vụ không có mã BS để áp dụng VPD cho việc xem nên phải tạo view và áp dụng cơ chế VPD select trên view đó .

```
--TAO VIEW HOSODICHVU DE GAN THEM COT MABS VAO DE GIUP CHO VIEC SELECT VDP CUA BACSI
CREATE or REPLACE VIEW BACSI_HOSO_DICHVU AS
    SELECT DV.*, BA.MABS
    FROM HOSOBENHAN BA, HOSO_DICHVU DV
    WHERE BA.MAKB=DV.MAKB;
```

- Áp dụng VPD cho view vừa tạo

```
--TAO FUNCTION BAC SI CHI DUOC XEM, THEM, SUA HO SO DICH VU MINH THUC HIEN HOAC CUA BENH NHAN MINH KHAM BENH
create or replace FUNCTION BacSi2(
p_schema IN VARCHAR2 DEFAULT NULL,
p_object IN VARCHAR2 DEFAULT NULL)
return varchar2 as
begin
    --LOAI USER ADMIN RA KHOI POLICY
    if user = 'PH1' then
        return '1=1';
    else
        return 'MABS=USER OR NGUOITHUCHIEN=USER';
    END IF;
end;

--TAO POLICY SELECT DOI VOI HOSODICHVU CUA BACSI
BEGIN
    DBMS_RLS.add_policy
        (object_name => 'BACSI_HOSO_DICHVU',
        policy_name => 'BACSIHOSODICHVU',
        policy_function => 'BACSI2',
        statement_types => 'SELECT',
        update_check => TRUE);
END;
/
```

- Bác sĩ thực hiện dịch vụ được quyền thêm và sửa bảng hồ sơ dịch vụ những dòng mà mình thực hiện (dùng lại function phía trên).

```
--TAO POLICY INSERT, UPDATE DOI VOI HOSODICHVU CUA BACSI
BEGIN
    DBMS_RLS.add_policy
        (object_name => 'HOSO_DICHVU',
        policy_name => 'BACSIHOSODICHVU2',
        policy_function => 'BACSI2',
        statement_types => 'INSERT, UPDATE',
        update_check => TRUE);
END;
/
```

2.4. MAC

Nhóm em không thực hiện phần này

2.5. Mã hóa

Chính sách: Áp dụng cơ chế mã hóa lên chính sách bảo mật lương cơ bản và phụ cấp của nhân viên. Và chỉ cấp quyền xem dữ liệu đã được mã hóa cho nhân viên kế toán.

Phương pháp quản lý khóa: Dùng phương pháp mã hóa đối xứng để mã/giải chỉ bằng 1 khóa và lưu khóa đó vào cơ sở dữ liệu. Sau đó áp dụng cơ chế VPD vào bảng lưu khóa để bảo mật.

- Tạo bảng lưu khóa

```
--TAO TABLE DE LUU KHOA (LUU Y DAT TEN KHONG NEN RO RANG)
CREATE TABLE TEMP(
        ABC     VARCHAR2(5) PRIMARY KEY,
        XYZ     VARCHAR2(300)
);
INSERT INTO TEMP VALUES('1', 'KEYMAHOADLANTOANHETHONGTHONGTIN1');
```

- Áp dụng VPD để hạn chế truy cập vào bảng

```
--TAO FUNCTION DE AP DUNG VPD HAN CHE VIEC TRUY CAP LEN TABLE TEMP
create or replace FUNCTION VPD_TEMP(
p_schema IN VARCHAR2 DEFAULT NULL,
p_object IN VARCHAR2 DEFAULT NULL)
return varchar2 as
begin
    --CHI ADMIN VA MOT VAI USER NHAN VIEN KE TOAN DUOC QUYEN TRUY CAP
    if user = 'PH1' OR USER='NV008' then
        return '1=1';
    else
        return '1=0';
    END IF;
end;
--TAO POLICY DE AP DUNG HAN CHE VIEC TRUY CAP LEN TABLE TEMP
BEGIN
    DBMS_RLS.add_policy
        (object_name => 'TEMP',
        policy_name => 'PREVENT_SELECT_ON_TEMP',
        policy_function => 'VPD_TEMP',
        statement_types => 'SELECT, INSERT, UPDATE, DELETE',
        update_check => TRUE);
END;
```

- Tạo hàm mã hóa

```
--Tao ham ma hoa chuoi
CREATE OR REPLACE FUNCTION Encryption(input_string IN VARCHAR2 DEFAULT NULL)
return RAW
as
    KEY VARCHAR2(300);
    encrypted_raw RAW (2000); -- stores encrypted binary text
    key_bytes_raw RAW (32); -- stores 256-bit encryption key
    encryption_type PLS_INTEGER := -- total encryption type
    DBMS_CRYPTO.ENCRYPT_AES256
    + DBMS_CRYPTO.CHAIN_CBC
    + DBMS_CRYPTO.PAD_PKCS5;
begin
    --LAY KHOA TU TABLE LUU VAO BIEN KEY
    SELECT XYZ INTO KEY FROM TEMP;
    key_bytes_raw := UTL_RAW.cast_to_raw(KEY);
    encrypted_raw := DBMS_CRYPTO.ENCRYPT
    ( src => UTL_I18N.STRING_TO_RAW (input_string, 'AL32UTF8'),
    typ => encryption_type,
    key => key_bytes_raw
    );
    return encrypted_raw;
end;
/
```

- Tạo hàm giải mã

```
--Tao ham giai ma chuoi
CREATE OR REPLACE FUNCTION Decryption(encrypted_raw IN RAW DEFAULT NULL)
return VARCHAR2
as
    KEY VARCHAR2(300);
    output_string VARCHAR2 (200);
    decrypted_raw RAW (2000); -- stores decrypted binary text
    key_bytes_raw RAW (32); -- stores 256-bit encryption key
    encryption_type PLS_INTEGER := -- total encryption type
    DBMS_CRYPTO.ENCRYPT_AES256
    + DBMS_CRYPTO.CHAIN_CBC
    + DBMS_CRYPTO.PAD_PKCS5;
begin
    --LAY KHOA TU TABLE LUU VAO BIEN KEY
    SELECT XYZ INTO KEY FROM TEMP;
    key_bytes_raw := UTL_RAW.cast_to_raw(KEY);
    decrypted_raw := DBMS_CRYPTO.DECRYPT
    (
    src => encrypted_raw,
    typ => encryption_type,
    key => key_bytes_raw
    );
    output_string := UTL_I18N.RAW_TO_CHAR (decrypted_raw, 'AL32UTF8');
    return output_string;
end;
```

- Thêm dữ liệu mã hóa vào bảng

```
insert into NHANVIEN values ('NV001','Tran Truc','NVQLTainguyennhanSu',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH1');
insert into NHANVIEN values ('NV002','Le Minh','NVQLTaivu',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH1');
insert into NHANVIEN values ('NV003','Nguyen Hoai Nam','NVQLChuyenmon',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH1');
insert into NHANVIEN values ('NV004','Nguyen Tran Minh Truc','NVBPTieptandieuphoi',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH2');
insert into NHANVIEN values ('NV005','Le Thi Be','NVBPTieptandieuphoi',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH2');
insert into NHANVIEN values ('NV006','Nguyen Nam','NVTaivu',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH4');
insert into NHANVIEN values ('NV007','Nguyen Thi My','NVTaivu',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH4');
insert into NHANVIEN values ('NV008','Le Hoang','Bac si',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH3');
insert into NHANVIEN values ('NV009','Le Hoang Nam','Bac si',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH3');
insert into NHANVIEN values ('NV010','Tran Trung Kien','Bac si',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH3');
insert into NHANVIEN values ('NV011','Le Hoang Thai','Bac si',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH3');
insert into NHANVIEN values ('NV012','Le Thi Duyen','Bac si',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH3');
insert into NHANVIEN values ('NV013','Nguyen Tran Minh','NVBanthuoc',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH5');
insert into NHANVIEN values ('NV014','Vo Van Dai','NVBanthuoc',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH5');
insert into NHANVIEN values ('NV015','Vo Van Hoai','NVKetoan',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH6');
insert into NHANVIEN values ('NV016','Nguyen Thi Ha','NVKetoan',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH6');
insert into NHANVIEN values ('NV017','Nguyen Thi Hong','NVChamcong',ENCRYPTION('5000000'), ENCRYPTION('50000'),'KH1');
```

- Tạo view giúp nhân viên xem được dữ liệu đã được giải mã

```
CREATE or REPLACE VIEW GIAIMA_NHANVIEN AS
    SELECT MANV, TENNV, CONGVIEC, Decryption(LUONGCOBAN) LUONGCOBAN, Decryption(PHUCAP) PHUCAP, MAKHOA
    from NHANVIEN;
SELECT * FROM GIAIMA_NHANVIEN;
UPDATE NHANVIEN SET LUONGCOBAN=ENCRYPTION(6000000) WHERE MANV='NV001';
```

2.6. Audit cơ bản và FGA

Audit cơ bản

- Tạo view BACSI_HOSO_DICHVU

```
--TAO VIEW HOSODICHVU DE GAN THEM COT MABS VAO DE GIUP CHO VIEC SELECT VDP CUA BACSI
CREATE or REPLACE VIEW BACSI_HOSO_DICHVU AS
    SELECT DV.*, BA.MABS
    FROM HOSOBENHAN BA, HOSO_DICHVU DV
    WHERE BA.MAKB=DV.MAKB;
```

- Thực hiện giám sát mọi hành động select và update trên view BACSI_HOSO_DICHVU

```
alter system set audit_trail = db scope = spfile;

audit select on BACSI_HOSO_DICHVU;
audit update on BACSI_HOSO_DICHVU;
```

FGA

- Áp dụng cơ chế FGA trên chính sách nhân viên kế toán vào việc quản lý các hành động sửa và xóa trên bảng nhân viên, cụ thể ở 2 cột lương cơ bản và phụ cấp

```
BEGIN
    DBMS_FGA.ADD_POLICY(
    object_name => 'NHANVIEN',
    policy_name => 'CHECK_NHANVIEN',
    audit_column => 'LUONGCOBAN, PHUCAP',
    statement_types=> 'update,delete');
END;
```