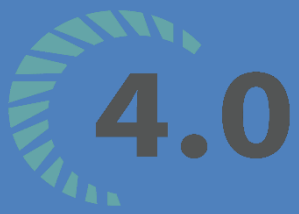
KHOA CÔNG NGHỆ THÔNG TIN

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM

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



Nhóm thực hiện: Nhóm 03

Giảng viên hướng dẫn: Phạm Thị Bạch Huệ
Tiết Gia Hồng
Lương Vĩ Minh

BÁO CÁO ĐỒ ÁN MÔN HỌC

HỌC KỲ II – NĂM HỌC 2022-2023

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
227 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM
Điện Thoại: (08) 38.354.266 - Fax:(08) 38.350.096

THÔNG TIN THÀNH VIÊN NHÓM

Mã nhóm:	20H3T-03		
STT	**MSSV**	**Họ tên**	**Email**
1	20120045	Hồ Thị Kim Chi	20120045@student.hcmus.edu.vn
2	20120065	Lâm Quang Duy	20120065@student.hcmus.edu.vn
3	20120099	Trần Huỳnh Hương	20120099@student.hcmus.edu.vn
4	20120180	Nguyễn Hữu Tài	20120180@student.hcmus.edu.vn

BẢNG PHÂN CÔNG & ĐÁNH GIÁ HOÀN THÀNH CÔNG VIỆC

Công việc thực hiện	Người thực hiện	Hoàn thành
Xác định yêu cầu đồ án	Cả nhóm	100%
Script tạo bảng, vẽ sơ đồ	Chi	100%
Generate dữ liệu	Duy + Hương	100%
Phân hệ 1: Phân tích chức năng ý 1, 2	Chi	100%
Phân hệ 1: Phân tích chức năng ý 3	Tài	100%
Phân hệ 1: Phân tích chức năng 5, 6	Hương	100%
Phân hệ 1: Phân tích chức năng ý 7	Duy	100%
Quay video demo phân hệ 1	Tài	100%
Triển khai phân hệ 1 lên ứng dụng	Chi + Tài	100%
Cài đặt các chính sách cho vai trò NHANVIEN và QLTRUCTIEP	Tài	100%
Cài đặt các chính sách cho vai trò TAICHINH và TRUONGDEAN	Duy	100%
Cài đặt các chính sách cho vai trò TRUONGPHONG	Hương	100%
Cài đặt các chính sách cho vai trò NHANSU	Chi	100%
Thực hiện chính sách mã hoá	Tài	100%
Audit	Duy + Hương	100%
Thực hiện chính sách sử dụng OLS	Chi + Hương	100%
Thiết kế giao diện cho ứng dụng	Cả nhóm	100%
Triển khai phân hệ 2 lên ứng dụng	Cả nhóm	100%
Kiểm tra và sửa lỗi	Cả nhóm	100%
Viết báo cáo	Cả nhóm	100%

DANH SÁCH CHỨC NĂNG

Chức năng	Hoàn thành
PHÂN HỆ 1	
Giao diện xem danh sách tên các đối tượng bạn đã tạo trong trong CSDL (user, role, table, view,...)	100%
Giao diện cho phép Admin thêm mới đối tượng (table, role, user, ...)	100%
Giao diện cho phép thêm quyền/ Lấy lại quyền của user/ role.	100%
Giao diện cho phép xem quyền của một chủ thể.	100%
PHÂN HỆ 2	
Giao diện cho toàn bộ nhân viên trong công ty theo từng vai trò.	100%
Nhân viên có quyền xem tất cả các thuộc tính trên quan hệ NHANVIEN và PHANCONG liên quan đến chính nhân viên đó.	100%
Nhân viên có thể cập nhật thông tin ngày sinh, địa chỉ, số điện thoại của chính họ.	100%
Nhân viên có thể xem dữ liệu của toàn bộ quan hệ PHONGBAN và DEAN.	100%
Quản lý trực tiếp được xem tất cả các thuộc tính, trừ thuộc tính LUONG và PHUCAP với các dòng dữ liệu trong quan hệ NHANVIEN liên quan đến các nhân viên mà người đó quản lý.	100%
Quản lý trực tiếp có quyền xem các dòng trong quan hệ PHANCONG liên quan đến chính họ và các nhân viên mà người đó quản lý	100%
Trưởng phòng có quyền xem tất cả các thuộc tính, trừ thuộc tính LUONG và PHUCAP với các dòng trong quan hệ NHANVIEN liên quan đến các nhân viên thuộc phòng ban mà người đó làm trưởng phòng.	100%
Trưởng phòng có thể thêm, xóa, cập nhật trên quan hệ PHANCONG liên quan đến các nhân viên thuộc phòng ban mà người đó làm trưởng phòng.	100%
Tài chính có thể xem trên toàn bộ quan hệ NHANVIEN và PHANCONG, có thể sửa trên thuộc tính LUONG và PHUCAP.	100%
Nhân sự được quyền thêm, cập nhật trên quan hệ PHONGBAN.	100%
Nhân sự không được xem LUONG, PHUCAP của người khác và chỉ được thêm, cập nhật dữ liệu trong quan hệ NHANVIEN với giá trị các trường LUONG, PHUCAP là mang giá trị mặc định là NULL.	100%
Trưởng đề án được quyền thêm, xóa, cập nhật trên quan hệ ĐEAN.	100%

MỤC LỤC

A. Yêu cầu đồ án:

Xây dựng một hệ thống S để quản lý thông tin nhân viên và việc tham gia đề án của nhân viên, gồm 2 phân hệ sau:

- Phân hệ 1: Dành cho người quản trị cơ sở dữ liệu, người dùng này có quyền xem danh sách người dùng trong hệ thống; xem thông tin về quyền (privileges) của mỗi user/role trên các đối tượng dữ liệu; tạo mới, xóa, sửa (hiệu chỉnh) user hoặc role; cấp quyền, thu hồi quyền, kiểm tra quyền của chủ thể được cấp quyền và chỉnh sửa quyền của user/role.
- Phân hệ 2: Cấp quyền truy cập cho từng đối tượng, thiết lập cơ chế, chính sách bảo mật và xem nhật ký hệ thống.

B. Báo cáo kết quả:

I. PHÂN HỆ 1:

1. Tạo user Admin cho hệ thống S:

```
CREATE USER ADMIN01 IDENTIFIED BY 1;
```

2. Các role/privileges được cấp cho phân hệ 1:

Role/Privileges	Mô tả
DBA	Là nhóm người dùng chịu trách nhiệm quản trị và vận hành các hoạt động liên quan đến cơ sở dữ liệu.
CREATE SESSION	Cho phép user kết nối vào database.
ALL PRIVILEGES	Cho phép user các quyền thực hiện tên đối tượng.
DBMS_RLS	Cho phép bạn áp dụng quyền truy cập dữ liệu trên mức hàng (row) của cơ sở dữ liệu.

3. Các bảng hệ thống mà admin xem trong phân hệ 1:

Table name	Mô tả
DBA_USERS	Cung cấp thông tin chi tiết về tất cả các người dùng.
DBA_ROLES	Cung cấp thông tin về các vai trò (roles) có sẵn trong hệ thống cơ sở dữ liệu.

DBA_TABLES	Cung cấp thông tin về các bảng có sẵn trong cơ sở dữ liệu.
DBA_SYS_PRIVS	Cung cấp thông tin về các quyền hệ thống được cấp cho người dùng.
DBA_TAB_PRIVS	Cung cấp thông tin về các quyền trên bảng được cấp cho người dùng.
DBA_ROLE_PRIVS	Cung cấp thông tin về các quyền của role.

4. Các câu lệnh thực thi chức năng trong phân hệ 1:

- Xem danh sách người dùng trong hệ thống:

```
SELECT * FROM DBA_USERS;
```

- Xem danh sách các role trong hệ thống:

```
SELECT * FROM DBA_ROLES;
```

- Xem thông tin quyền của mỗi user trên các đối tượng dữ liệu:

```
SELECT * FROM DBA_SYS_PRIVS WHERE GRANTEE = username;
```

- Xem thông tin quyền của mỗi role trên các đối tượng dữ liệu:

```
SELECT * FROM DBA_ROLE_PRIVS WHERE GRANTEE = rolename;
```

- Tạo role mới

```
CREATE ROLE rolename;
```

- Xoá role:

```
DROP ROLE role_name;
```

- Cấp quyền cho role:

```
GRANT privilege_name TO rolename;
```

- Thu hồi quyền từ role

```
REVOKE privilege_name FROM rolename;
```

- Tạo user mới:

```
CREATE USER username IDENTIFIED BY password;
```

- Xoá user:

```
DROP USER username;
```

- Cấp quyền cho user:

```
GRANT privilege_name TO username;
```

- Cấp role cho user:

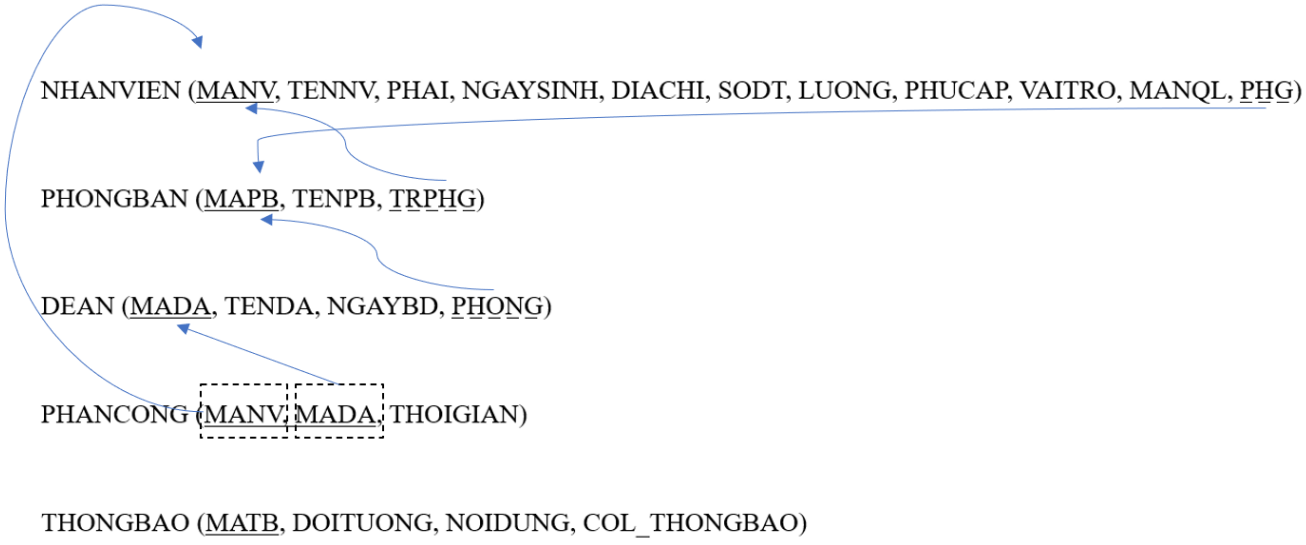
```
GRANT role_name TO user;
```

- Thu hồi quyền từ user

```
REVOKE privilege_name FROM username;
```

II. PHÂN HỆ 2:

1. Sơ đồ dữ liệu quan hệ:



2. Cài đặt các chính sách bảo mật:

2.1. Lý thuyết:

a. RBAC:

Khái niệm: RBAC là viết tắt của Role Based Access Control, đây là mô hình kiểm soát truy cập được sử dụng rộng rãi nhờ có những ưu điểm vượt trội hơn DAC, cũng khắc phục được một số điểm yếu của DAC.

Nguyên lý: Các chủ thể sẽ được phân chia và quản lý theo các nhóm/vai trò. Một chủ thể chỉ có thể truy cập những tài nguyên mà các nhóm/vai trò của chủ thể đó được quyền truy cập.

Đặc điểm: Với RBAC thì quyền truy cập tài nguyên sẽ không được gán trực tiếp cho chủ thể nữa. Bản thân mỗi chủ thể cũng không có quyền thay đổi nhóm/vai trò của mình, mà chỉ có quản

trị viên được quyền làm điều này. Thường thì các công ty, tổ chức sẽ quản lý tài nguyên theo mô hình này, Active Directory trên Windows cũng sử dụng mô hình RBAC.

Ưu điểm:

- Quản lý dễ dàng.
- Các thực thể không thể thay đổi nhóm/vai trò của mình, cũng không thể thay đổi quyền truy cập của các nhóm/vai trò.
- Thuận tiện cho việc tuân thủ nguyên tắc đặc quyền tối thiểu.

Nhược điểm:

- Việc cài đặt và quản lý phức tạp khi có quá nhiều nhóm/vai trò trong hệ thống.
- Các nhóm/vai trò chưa có sự phân cấp rõ ràng.

b. DAC:

Giới thiệu:

- Người dùng có thể bảo vệ những gì thuộc về mình
- Chủ của dữ liệu sẽ có toàn quyền trên dữ liệu đó
- Chủ của dữ liệu có quyền định nghĩa các loại truy cập đọc/ghi/thực thi (read/write/execute/…) và gán những quyền đó cho những người dùng khác.

Cách thức điều khiển truy cập: Cách thức cơ bản điều khiển truy cập của DAC trong một hệ cơ sở dữ liệu (HCSDL) là dựa vào 2 thao tác cơ bản:

- Gán quyền (granting privileges): cho phép người dùng khác được quyền truy cập lên đối tượng do mình làm chủ.
- Thu hồi quyền (revoking privileges): thu hồi lại quyền đã gán cho người dùng khác.

Các loại quyền:

- Quyền ở cấp tài khoản/hệ thống (account/system level): là những quyền này độc lập với các đối tượng trong HCSDL. Những quyền này do người quản trị hệ thống định nghĩa và gán cho mỗi người dùng.
  - `CREATE SCHEMA`: tạo lược đồ CSDL.
  - `CREATE TABLE`: tạo bảng dữ liệu/ quan hệ (relation).
  - `CREATE VIEW`: tạo view.

- `ALTER`: chỉnh sửa các schema/relation.
- `DROP`: xóa relation/view.
- `MODIFY`: quyền thêm/ xóa/ sửa các hàng dữ liệu (record/ tuple).
- `SELECT`: quyền thực hiện câu truy vấn thông tin trong CSDL.

- Quyền ở cấp đối tượng (object level): là những quyền trên mỗi đối tượng trong hệ CSDL. Người dùng tạo ra đối tượng nào thì sẽ có tất cả các quyền trên đối tượng đó. Các đối tượng dữ liệu này gồm: các relation hoặc view.
  - `INSERT`: thêm dữ liệu vào relation.
  - `UPDATE`: cập nhật /chỉnh sửa dữ liệu trong relation.
  - `DELETE`: xóa dữ liệu trong relation.
  - `REFERENCE`: tham khảo đến dữ liệu trong relation.

c. VPD:

Khái niệm: VPD là viết tắt của Virtual Private Database, đây là một phương pháp hiện thực việc bảo mật dữ liệu ở mức dòng dữ liệu trong hệ quản trị cơ sở dữ liệu Oracle. VPD kết hợp hai kỹ thuật chính là Fine-grained access control (FGAC) và Application Context.

- FGAC (Fine-grained access control) cho phép người quản trị thiết lập chính sách bảo mật chi tiết và liên kết chúng với các bảng, view hoặc synonym. Các chính sách bảo mật này được áp dụng cho người dùng với các quyền hạn khác nhau, tạo ra những "khung nhìn" khác nhau đối với dữ liệu được bảo vệ. FGAC giúp quản lý chính sách bảo mật dễ dàng hơn so với việc sử dụng view.
- Application Context cung cấp một nơi lưu trữ thông tin bảo mật cho các giá trị ngữ cảnh của ứng dụng. Sử dụng Application Context cùng với FGAC tăng cường hiệu quả thực hiện của việc bảo mật dữ liệu.

Row-level Security:

- Row-level Security (RLS) là một tính năng trong Oracle cho phép giới hạn truy cập vào các hàng (record) dữ liệu dựa trên chính sách bảo mật được thiết lập. Mỗi chính sách bảo mật mô tả các quy định để quản lý việc truy cập dòng dữ liệu.
- Cơ chế thực hiện RLS:

- Đầu tiên, ta tạo một hàm PL/SQL trả về một chuỗi chứa các điều kiện của chính sách bảo mật.
- Hàm PL/SQL được đăng ký cho các bảng, view cần bảo vệ bằng package PL/SQL `DBMS_RLS`.
- Khi có một câu truy vấn từ bất kỳ người dùng nào trên đối tượng được bảo vệ, Oracle sẽ nối chuỗi điều kiện từ hàm PL/SQL vào mệnh đề WHERE của câu lệnh SQL ban đầu. Điều này giúp lọc các hàng dữ liệu theo các điều kiện của chính sách bảo mật.
- Các lưu ý khi làm việc với RLS:
- Hàm PL/SQL được đăng ký cho bảng, view, hoặc synonym bằng cách sử dụng thủ tục `DBMS_RLS.ADD_POLICY`.
- Thủ tục `ADD_POLICY` yêu cầu ít nhất 3 tham số: object_name, policy_name, policy_function.
- Sự kết hợp của object_schema, object_name, và policy_name phải là duy nhất.
- Mặc định, chính sách bảo mật áp dụng cho tất cả các lệnh DML. Có thể chỉ định loại câu lệnh áp dụng chính sách bằng tham số `STATEMENT_TYPES`.
- Khi người dùng truy cập vào đối tượng được bảo vệ, hàm PL/SQL đã đăng ký sẽ được thực thi và câu lệnh SQL của người dùng sẽ được điều chỉnh và thực thi.
- Tài khoản `SYS` không bị ảnh hưởng bởi chính sách bảo mật.
- Nhiều chính sách có thể áp dụng cho cùng một đối tượng, và CSDL sẽ kết hợp các chính sách đó theo phép `AND`.
- Quyền sử dụng package `DBMS_RLS` không được gán cho tất cả người dùng. Người quản trị cần được gán quyền `EXECUTE ON DBMS_RLS` để sử dụng package này.
- Các policy function cần có đúng 2 tham số truyền vào: tên schema và tên đối tượng được bảo vệ. Các tham số này xác định đối tượng mà chính sách được áp dụng.
- Policy function cần được tạo trong schema của người quản trị bảo mật và không nên cho phép người dùng khác thực thi, sửa đổi hoặc xóa chúng.

2.2. Phân tích cách thực thi các chính sách bảo mật:

Chính sách 1: Những người dùng có thuộc tính VAITRO là "Nhân viên" cho biết đó là một nhân viên thông thường, không kiêm nhiệm công việc nào khác. Những người dùng có VAITRO là "Nhân viên" có quyền được mô tả như sau:

- Có quyền xem tất cả các thuộc tính trên quan hệ NHANVIEN và PHANCONG liên quan đến chính nhân viên đó.
- Có thể sửa trên các thuộc tính NGAYSINH, DIACHI, SODT liên quan đến chính nhân viên đó.
- Có thể xem dữ liệu của toàn bộ quan hệ PHONGBAN và DEAN.

Chủ thể: Người dùng có role "rNHANVIEN".

Quyền: SELECT – Đối tượng: NHANVIEN, PHANCONG.

Quyền: UPDATE – Đối tượng: NHANVIEN (chỉ thuộc tính NGAYSINH, DIACHI, SDT).

→ View: Tạo view "vw_NHANVIEN_XemThongTinCaNhan" khi muốn xem thông tin cá nhân của bản thân; Tạo procedure sửa thông tin NGAYSINH, DIACHI, SDT trên vw_NHANVIEN_XemThongTinCaNhan và trigger cập nhật trên bảng chính thay vì cập nhật trên view. Tạo view "vw_NHANVIEN_XemThongTinPhanCong" để xem thông tin phân công của cá nhân.

→ VPD: Nếu VAITRO = "Nhan vien" thì vị từ là MANV = ''‘||User||’'' (PHANCONG).

→ Sử dụng cơ chế: RBAC, VPD.

Chính sách 2: Những người dùng có VAITRO là "QL trực tiếp" nếu họ phụ trách quản lý trực tiếp nhân viên khác. Nhân viên Q là quản lý trực tiếp nhân viên N, có quyền được mô tả như sau:

- Q có quyền như là một nhân viên thông thường (vai trò "Nhân viên"). Ngoài ra, với các dòng dữ liệu trong quan hệ NHANVIEN liên quan đến các nhân viên N mà Q quản lý trực tiếp thì Q được xem tất cả các thuộc tính, trừ thuộc tính LUONG và PHUCAP.
- Có thể xem các dòng trong quan hệ PHANCONG liên quan đến chính Q và các nhân viên N được quản lý trực tiếp bởi Q.

Chủ thể: Người dùng có role "rQLTRUCTIEP".

Quyền: SELECT – Đối tượng: NHANVIEN (trừ thuộc tính LUONG, PHUCAP), PHANCONG.

→ View: Tạo view "vw_QLTRUCTIEP_XemThongTinNhanVien" để xem thông tin của nhân viên có MaNQL là mã login của user hiện tại. Tạo "vw_QLTRUCTIEP_XemPhanCongNhanVien" để xem phân công cho nhân viên thuộc quản lý của mình.

→ VPD: Nếu VAITRO = "QL Truc tiep" thì vị từ là "MANV='"||USER1||"' or MANV in (select MANV from ADMIN01.vw_QLTRUCTIEP_XemThongTinNhanVien where MANQL = '"||USER1||"')" (PHANCONG).

→ Sử dụng cơ chế: RBAC, VPD.

Chính sách 3: Những người dùng có VAITRO là "Trưởng phòng" cho biết đó là một nhân viên kiêm nhiệm thêm vai trò trưởng phòng. Một người dùng T có VAITRO là "Trưởng phòng" có quyền được mô tả như sau:

- T có quyền như là một nhân viên thông thường (vai trò "Nhân viên"). Ngoài ra, với các dòng trong quan hệ NHANVIEN liên quan đến các nhân viên thuộc phòng ban mà T làm trưởng phòng thì T có quyền xem tất cả các thuộc tính, trừ thuộc tính LUONG và PHUCAP.
- Có thể thêm, xóa, cập nhật trên quan hệ PHANCONG liên quan đến các nhân viên thuộc phòng ban mà T làm trưởng phòng.

Chủ thể: Người dùng có role "rTRUONGPHONG".

Quyền: SELECT – Đối tượng: NHANVIEN (trừ thuộc tính LUONG, PHUCAP), PHANCONG.

Quyền: INSERT, DELETE, UPDATE – Đối tượng: PHANCONG.

→ View: Tạo view "vw_NHANVIEN_XemThongTinCaNhan" khi muốn xem thông tin cá nhân của bản thân; Tạo view "vw_TRUONGPHONG_Xem_NHANVIEN" xem toàn bộ danh sách nhân viên thuộc phòng ban mà mình làm trưởng phòng thông qua view.

→ VPD: Nếu VAITRO = "Truong phong" thì vị từ là "PHANCONG.MANV in (select MANV from ADMIN01.vw_TRUONGPHONG_Xem_NHANVIEN where PHG =

'"||MAPHONGBAN||"')" với MAPHONGBAN là mã phòng ban của chính người trưởng phòng này. (VPD cho hành vi SELECT trên bảng PHANCONG).

→ Sử dụng cơ chế: RBAC, VPD.

Chính sách 4: Những người dùng có VAITRO là "Tài chính" cho biết đó là một nhân viên phụ trách công tác tài chính tiền lương của công ty. Một người dùng có vai trò là "Tài chính" có quyền được mô tả như sau:

- Có quyền như là một nhân viên thông thường (vai trò "Nhân viên").
- Xem trên toàn bộ quan hệ NHANVIEN và PHANCONG, có thể sửa trên thuộc tính LUONG và PHUCAP (thừa hành ban giám đốc).

Chủ thể: Người dùng có role "rTAICHINH".

Quyền: SELECT – Đối tượng: NHANVIEN, PHANCONG.

Quyền: UPDATE – Đối tượng: NHANVIEN (chỉ thuộc tính LUONG, PHUCAP).

→ View: Tạo view "vw_NHANVIEN_XemThongTinCaNhan" khi muốn xem thông tin cá nhân của bản thân; Tạo view "vw_TAICHINH_Xem_NHANVIEN" xem toàn bộ danh sách nhân viên thông qua view.

→ Update trên view, sử dụng Instead Of Trigger "trg_NHANVIEN_XemThongTinCaNhan" để cập nhật 2 trường LUONG và PHUCAP lên bảng NHANVIEN.

→ VPD: Nếu VAITRO = "Tai chinh" thì vị từ là MANV = '‘1=1'’' (PHANCONG).

→ Sử dụng cơ chế: RBAC, VPD.

Chính sách 5: Những người dùng có VAITRO là "Nhân sự" cho biết đó là nhân viên phụ trách công tác nhân sự trong công ty. Một người dùng có VAITRO là "Nhân sự" có quyền được mô tả như sau:

- Có quyền như là một nhân viên thông thường (vai trò "Nhân viên").
- Được quyền thêm, cập nhật trên quan hệ PHONGBAN.
- Thêm, cập nhật dữ liệu trong quan hệ NHANVIEN với giá trị các trường LUONG, PHUCAP là mang giá trị mặc định là NULL, không được xem LUONG, PHUCAP của người khác và không được cập nhật trên các trường LUONG, PHUCAP.

Chủ thể: Người dùng có role "rNHANSU".

Quyền: SELECT – Đối tượng: NHANVIEN, PHANCONG, PHONGBAN.

Quyền: INSERT, UPDATE – Đối tượng: NHANVIEN, PHONGBAN.

→ View: Tạo view "VIEW_NHANVIEN_NS" xem toàn bộ danh sách nhân viên với trường LUONG, PHUCAP mang giá trị mặc định là NULL thông qua view.

→ Insert, Update trên view, sử dụng Instead Of Trigger "trg_sync_nhanvien" để thêm và cập nhật thông tin lên bảng NHANVIEN.

→ VPD: Nếu VAITRO = "Nhan su" thì vị từ là MANV = '''||User||''' (PHANCONG).

→ Sử dụng cơ chế: RBAC, VPD.

Chính sách 6: Những người dùng có VAITRO là "Trưởng đề án" cho biết đó là nhân viên là trưởng các đề án. Một người dùng là "Trưởng đề án" có quyền được mô tả như sau:

- Có quyền như là một nhân viên thông thường (vai trò "Nhân viên").
- Được quyền thêm, xóa, cập nhật trên quan hệ ĐEAN.

Chủ thể: Người dùng có role "rTRUONGDEAN".

Quyền: SELECT, UPDATE, INSERT, DELETE – Đối tượng: DEAN.

Quyền: SELECT – Đối tượng: PHANCONG.

→ View: Tạo view "vw_NHANVIEN_XemThongTinCaNhan" khi muốn xem thông tin cá nhân của bản thân.

→ VPD: Nếu VAITRO = "Nhan su" thì vị từ là MANV = '''||User||''' (PHANCONG).

→ Sử dụng cơ chế: RBAC, VPD.

3. Mã hóa:

3.1. Lý thuyết:

Cơ chế khoá quản lý khoá: khoá được thiết lập bằng package và được lưu trữ trên Oracle. Ngoài ra khoá sẽ được lưu backup dưới dạng file txt ở một vị trí khác. Vì được thiết lập khoá bằng package nên có thể dễ dàng thay đổi khoá và cập nhật lại database.

Cơ chế khoá:

- Sử dụng thử viện `dbms_crypto` được cung cấp bởi Oracle Database để mã hoá và giải mã.

- Thuật toán mã hoá được sử dụng là DES_CBC_PKCS5. Đây là một thuật toán mã hoá đối xứng có độ dài 56 bit. Nó sử dụng một khối dữ liệu đầu vào có độ dài bằng với độ dài khóa (56 bit) và một khóa để tạo ra khối mã hóa đầu ra có độ dài bằng với khối dữ liệu đầu vào. CBC (Cipher-Block Chaining) là một chế độ hoạt động của thuật toán DES, trong đó mỗi khối dữ liệu đầu vào được XOR với khối mã hóa trước đó trước khi được mã hóa.
- PKCS5 là một chuẩn mã hóa dữ liệu được sử dụng để bảo vệ khối dữ liệu đầu vào. Nó bao gồm việc thêm các byte trống vào cuối khối dữ liệu để đảm bảo độ dài của khối dữ liệu đầu vào là bội số của 8 byte. Nếu khối dữ liệu đầu vào đã là bội số của 8 byte, thì một khối dữ liệu đặc biệt được thêm vào để đảm bảo tính đối xứng của mã hóa.
- Để mã hóa thông tin, đầu tiên, dữ liệu cần được chuyển đổi thành một chuỗi byte bằng cách sử dụng hàm `UTL_RAW.CAST_TO_RAW`. Sau đó, thuật toán DES_CBC_PKCS5 được áp dụng để mã hóa chuỗi byte này với khóa được lấy từ p_key.get_key(). Kết quả là một chuỗi byte được mã hóa, được sử dụng để bảo vệ thông tin trước khi lưu trữ hoặc truyền đi.
- Vì vậy, cơ chế này sử dụng một thuật toán mã hóa đối xứng và một chuẩn mã hóa dữ liệu để bảo vệ thông tin trước khi lưu trữ hoặc truyền đi. Nó cũng sử dụng một khóa để tạo ra khối mã hóa đầu ra, đảm bảo tính bảo mật của thông tin được mã hóa.
- Khoá được lưu trữ trong package của oracle và được gán động vô các hàm mã hoá, giải mã để dễ dàng thay đổi cập nhật về sau.

3.2. Phân tích thực thi:

User thực hiện vai trò mã hoá: ADMIN01.

Mã hoá dữ liệu ở mức database.

Thay đổi về cấu trúc dữ liệu: tăng kích thước lưu trữ cho trường LUONG và PHUCAP:

- Tạo một package lưu trữ khoá gồm 16 ký tự.

```
CREATE OR REPLACE PACKAGE p_key IS
    pk nchar(15):='123456789123456';
    Procedure set_key(new_key nchar);
  FUNCTION get_key RETURN nchar;
END;
/

CREATE OR REPLACE PACKAGE BODY p_key IS
    Procedure set_key( new_key nchar) is
    begin
    pk:=new_key;
    end;
  FUNCTION get_key RETURN nchar IS
  BEGIN
    RETURN p_key.pk;
  END;
END;
/
```

- Tiến hành mã hoá trường LUONG và PHUCAP bằng các bước:
  - Tạo 2 cột LUONG_C và PHUCAP_C tạm sau đó sẽ mã hoá dữ liệu cột LUONG và PHUCAP lưu vào cột LUONG_C và PHUCAP_C.
  - Xoá cột LUONG và PHUCAP ban đầu.
  - Đổi tên cột LUONG_C và PHUCAP_C thành LUONG và PHUCAP
  - Sau khi đã mã hoá các câu lệnh truy vấn trên cột NHANVIEN sẽ được thực hiện thông qua view và được giải mã ngay tại view. Những câu lệnh update mà có ảnh hưởng đến trường LUONG và PHUCAP sẽ được trigger xử lý mã hoá từ giá trị thật thêm vào sau đó mã hoá rồi mới cập nhật vào bảng thật.

4. OLS:

4.1. Lý thuyết:

Khái niệm: Oracle Label Security (OLS) là một sản phẩm được thực hiện dựa trên nền tảng công nghệ Virtual Private Database (VPD), cho phép các nhà quản trị điều khiển truy xuất dữ liệu ở mức hàng (row-level) một cách tiện lợi và dễ dàng hơn. Nó điều khiển việc truy xuất nội dung của các dòng dữ liệu bằng cách so sánh nhãn của hàng dữ liệu với nhãn và quyền của user. Các nhà quản trị có thể dễ dàng tạo thêm các chính sách kiểm soát việc truy xuất các hàng dữ

liệu cho các CSDL bằng giao diện đồ họa thân thiện người dùng có tên gọi là Oracle Policy Manager hoặc bằng 6 packages được xây dựng sẵn cho OLS như sau:

- `SA_SYSDBA`: tạo, thay đổi, xóa các chính sách.
- `SA_COMPONENTS`: định nghĩa và quản lý các thành phần của nhãn.
- `SA_LABEL_ADMIN`: thực hiện các thao tác quản trị chính sách, nhãn.
- `SA_POLICY_ADMIN`: áp dụng chính sách cho bảng và schema.
- `SA_USER_ADMIN`: quản lý việc cấp phát quyền truy xuất và quy định mức độ tin cậy cho các user liên quan.
- `SA_AUDIT_ADMIN`: thiết lập các tùy chọn cho các tác vụ quản trị việc audit.

Trong OLS, ta dùng các chính sách (policy) để quản lý truy xuất. Đối với mỗi chính sách, ta cần định ra một tập nhãn để phân lớp dữ liệu từ cao xuống thấp dựa theo mức độ nhạy cảm của dữ liệu (ngoài ra các nhãn còn có những yếu tố khác mà ta sẽ bàn đến khi đi vào chi tiết). Các nhãn đó được gọi là các nhãn dữ liệu - "data label". Sau đó ta áp dụng các chính sách lên các bảng hoặc schema mà mình mong muốn bảo vệ. Mỗi khi một người dùng muốn truy xuất một hàng dữ liệu nào đó, hệ thống sẽ so sánh nhãn của người dùng (user label) tại thời điểm đó với nhãn dữ liệu để quyết định có cho phép việc truy xuất hay không.

Năm bước thực hiện OLS:

- Bước 1: Tạo chính sách OLS (OLS policy).
- Bước 2: Định nghĩa các thành phần mà một label thuộc chính sách trên có thể có.
- Bước 3: Tạo các nhãn OLS thật sự mà bạn muốn dùng.
- Bước 4: Gán các chính sách OLS cho các table hoặc schema mà bạn muốn bảo vệ.
- Bước 5: Gán các nhãn cho các user.

4.2. Phân tích cách thực thi:

Phát biểu: Người ta muốn thiết lập cho hệ thống S chức năng phát tán thông báo có mục tiêu đến những nhóm người dùng trong hệ thống tùy vào cấp bậc, lĩnh vực hoạt động và vị trí địa lý nơi nhân viên công tác. Cho biết người dùng (nhân viên) và dữ liệu được chia ra làm các cấp bậc sau: giám đốc, trưởng phòng và nhân viên và độ ưu tiên là: giám đốc > trưởng phòng > nhân viên. Hệ thống hoạt động ở 3 lĩnh vực: mua bán, sản xuất, gia công. Công ty có chi nhánh

đặt tại ba nơi: miền Bắc, miền Trung và miền Nam. Cho biết cụ thể cách thiết lập hệ thống nhãn gồm 03 thành phần và những điều chỉnh mô hình dữ liệu (nếu có).

Cách thực hiện:

- ***Kích hoạt OLS và tạo bảng THONGBAO cùng dữ liệu mẫu***

```
------------------------------------------- OLS ---------------------
------Tao user voi vai tro sysdba
DROP USER DoAnATBM_Sys CASCADE;
create user DoAnATBM_Sys IDENTIFIED BY DoAnATBM_Sys;
GRANT ALL PRIVILEGES TO DoAnATBM_Sys;
GRANT sysdba TO DoAnATBM_Sys;
------Kich hoat ols va lbacsys
SELECT STATUS FROM DBA_OLS_STATUS WHERE NAME='OLS_CONFIGURE_STATUS';
ALTER USER LBACSYS ACCOUNT UNLOCK;
EXEC LBACSYS.CONFIGURE_OLS;
EXEC LBACSYS.OLS_ENFORCEMENT.ENABLE_OLS;
SELECT VALUE FROM V$OPTION WHERE PARAMETER='Oracle Label Security';
```

```
------------------------TAO TABLE THONG BAO-----------------------------------
conn ADMIN01/1;
drop table THONGBAO;
CREATE TABLE THONGBAO (
    MaTB NUMBER PRIMARY KEY,
    DOITUONG VARCHAR2(255),
    NOIDUNG VARCHAR2(255)
);

INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (1,'TP:ALL:ALL', 'Cuoc hop khan ve ke hoach san pham moi');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (2,'TP:SX:MN', 'Cuoc hop thong qua ngan sach moi');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (3,'TP:SX:MB', 'Thong bao cuoc hop bao cao tien du an');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (4,'TP:SX:MT', 'Cuoc hop khan ve co so ha tang');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (5,'TP:ALL:MB', 'Cuoc hop ke hoach hang tuan');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (6,'TP:ALL:ALL', 'Thong bao cuoc hop quan trong voi doi tac');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (7,'TP:SX:MT', 'Cuoc hop khan ve van de an ninh thong tin');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (8, 'TP:MB:ALL','Thong bao cuoc hop danh gia hieu suat nhan vien');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (9,'TP:SX:MT', 'Cuoc hop khan ve viec trien khai du an moi');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (10,'TP:SX:MT', 'Thong bao cuoc hop voi khach hang');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (11,'TP:MB:MN', 'Thong bao cuoc hop quan trong voi doi tac');
INSERT INTO THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (12,'GD:SX:ALL', 'Thong bao cuoc hop giam doc san xuat hang thang');
-----------------------------------------------------------------------------
```

- ***Tạo chính sách OLS (OLS policy)***

Ngữ cảnh: Khi có thông báo từ cấp trên, tùy vào mức độ quan trọng của thông báo sẽ quyết định xem nhân viên nào nhận và đọc được thông báo

→ Tạo chính sách ACCESS_THONGBAO và cộ COL_THONGBAO nhằm gán nhãn dữ liệu cho từng dòng thông báo.

```
connect lbacsys/lbacsys;
alter session set NLS_NUMERIC_CHARACTERS = '.,';
EXEC sa_sysdba.drop_policy(policy_name => 'ACCESS_THONGBAO');

-----TAO OLS POLICY-----
EXECUTE sa_sysdba.create_policy(policy_name => 'ACCESS_THONGBAO', column_name => 'COL_THONGBAO');
```

- ***Định nghĩa thành phần của nhãn***

3 cấp bậc ứng với LEVEL: giám đốc > trưởng phòng > nhân viên.

```
-----TAO LEVEL
EXECUTE sa_components.create_level(policy_name => 'ACCESS_THONGBAO', level_num=>100, short_name=>'GD', long_name=>'GIAM DOC');
EXECUTE sa_components.create_level(policy_name => 'ACCESS_THONGBAO', level_num=>60, short_name=>'TP', long_name=>'TRUONG PHONG');
EXECUTE sa_components.create_level(policy_name => 'ACCESS_THONGBAO', level_num=>20, short_name=>'NV', long_name=>'NHANVIEN');
```

3 lĩnh vực ứng với COMPARTMENT: mua bán, sản xuất, gia công.

```
-----TAO COMPARTMENT
EXECUTE sa_components.create_compartment(policy_name => 'ACCESS_THONGBAO', comp_num=>100, short_name=>'MB', long_name=>'MUA BAN');
EXECUTE sa_components.create_compartment(policy_name => 'ACCESS_THONGBAO', comp_num=>80, short_name=>'SX', long_name=>'SAN XUAT');
EXECUTE sa_components.create_compartment(policy_name => 'ACCESS_THONGBAO', comp_num=>69, short_name=>'GC', long_name=>'GIA CONG');
```

3 vùng ứng với GROUP: miền Bắc, miền Trung, miền Nam.

```
-----TAO GROUP
EXECUTE sa_components.create_group(policy_name => 'ACCESS_THONGBAO', group_num=>210, short_name=>'MB', long_name=>'MIEN BAC');
EXECUTE sa_components.create_group(policy_name => 'ACCESS_THONGBAO', group_num=>220, short_name=>'MT', long_name=>'MIEN TRUNG');
EXECUTE sa_components.create_group(policy_name => 'ACCESS_THONGBAO', group_num=>230, short_name=>'MN', long_name=>'MIEN NAM');
```

- ***Gán nhãn cho dữ liệu***

```
-----TAO LABEL
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>110, label_value=>'GD:MB,SX,GC:MB,MT,MN', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>120, label_value=>'GD:MB,SX,GC:MB,MT', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>130, label_value=>'GD:MB,SX,GC:MB,MN', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>140, label_value=>'GD:MB,SX,GC:MB', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>150, label_value=>'GD:MB,SX,GC:MN', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>160, label_value=>'GD:MB,SX,GC:MT', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>165, label_value=>'GD:SX:MB,MT,MN', data_label=>true);


EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>170, label_value=>'TP:MB,SX,GC:MB,MT,MN', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>180, label_value=>'TP:SX:MT,MB,MN', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>190, label_value=>'TP:SX:MN', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>210, label_value=>'TP:SX:MB', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>220, label_value=>'TP:SX:MT', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>230, label_value=>'TP:MB,SX,GC:MB', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>240, label_value=>'TP', data_label=>true);

EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>175, label_value=>'TP:MB:MB,MT,MN', data_label=>true);
EXECUTE sa_label_admin.create_label(policy_name => 'ACCESS_THONGBAO', label_tag=>200, label_value=>'TP:MB:MN', data_label=>true);
```

- ***Áp dụng chính sách OLS lên bảng THONGBAO***

```
----APLLY POLICY LEN BANG THONGBAO
EXECUTE sa_policy_admin.apply_table_policy(policy_name=>'ACCESS_THONGBAO', schema_name=>'ADMIN01',
table_name=>'THONGBAO',table_options=>'LABEL_DEFAULT,READ_CONTROL', label_function=>null, predicate=>null);
```

- ***Cập nhật nhãn dữ liệu cho các dòng THONGBAO***

```
----GAN DU LIEU CHO BANG THONGBAO
connect DoAnATBM_Sys/DoAnATBM_Sys as sysdba;
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP') where DOITUONG = 'TP:ALL:ALL';
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:SX:MB') where DOITUONG = 'TP:SX:MB';
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:SX:MN') where DOITUONG = 'TP:SX:MN';
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:MB,SX,GC:MB') where DOITUONG = 'TP:ALL:MB';
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','GD:SX:MB,MT,MN') where DOITUONG = 'GD:SX:ALL';
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:MB:MB,MT,MN') where DOITUONG = 'TP:MB:ALL';
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:SX:MT') where DOITUONG = 'TP:SX:MT';
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:MB:MN') where DOITUONG = 'TP:MB:MN';
```

Ngoài ra, ta tạo thêm trigger "*capnhat_ols*" nhằm cập nhật nhãn dữ liệu tự động khi bảng THONGBAO có thêm dòng dữ liệu mới được insert vào.

```
CREATE OR REPLACE TRIGGER capnhat_ols
AFTER INSERT ON ADMIN01.THONGBAO
FOR EACH ROW
DECLARE
    new_luong varchar2(100);
    new_phucap varchar2(100);
BEGIN
    UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP') where DOITUONG = 'TP:ALL:ALL';
    UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:SX:MB') where DOITUONG = 'TP:SX:MB';
    UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:SX:MN') where DOITUONG = 'TP:SX:MN';
    UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:MB,SX,GC:MB') where DOITUONG = 'TP:ALL:MB';
    UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','GD:SX:MB,MT,MN') where DOITUONG = 'GD:SX:ALL';
    UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:MB:MB,MT,MN') where DOITUONG = 'TP:MB:ALL';
    UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:SX:MT') where DOITUONG = 'TP:SX:MT';
    UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:MB:MN') where DOITUONG = 'TP:MB:MN';
END;
/
```

```
INSERT INTO ADMIN01.THONGBAO (MaTB,DOITUONG, NOIDUNG) VALUES (30,'TP:SX:MB', 'Thong bao cuoc hop bao cao tien du an');
```

- ***Gán nhãn cho người dùng***

<u>Câu a</u>: Gán nhãn cho 3 người dùng trong hệ thống:

- *01 giám đốc có thể đọc được toàn bộ dữ liệu*
- *01 trưởng phòng phụ trách lĩnh vực sản xuất miền Nam*
- *01 giám đốc phụ trách bất kỳ lĩnh vực nào ở chi nhánh miền Bắc (có thể đọc được toàn bộ dữ liệu theo đúng cấp bậc và không phân biệt lĩnh vực)*

Tạo 3 người dùng tương ứng như sau:

```
----CREATE 3 USER TUONG UNG
CONN ADMIN01/1;
DROP USER GIAMDOCALL;
DROP USER GIAMDOCMIENBAC;
DROP USER TRUONGPHONGSXMN;
```

```
CREATE USER GIAMDOCALL IDENTIFIED BY GIAMDOCALL;
CREATE USER GIAMDOCMIENBAC IDENTIFIED BY GIAMDOCMIENBAC;
CREATE USER TRUONGPHONGSXMN IDENTIFIED BY TRUONGPHONGSXMN;
```

Gán nhãn cho 3 người dùng:

```
----SET LEVEL CHO USER
connect lbacsys/lbacsys;

EXECUTE sa_user_admin.set_user_labels(policy_name=>'ACCESS_THONGBAO', user_name=>'GIAMDOCALL',
max_read_label=>'GD:MB,SX,GC:MB,MT,MN', def_label=>'GD:MB,SX,GC:MB,MT,MN', row_label=>'GD:MB,SX,GC:MB,MT,MN');
EXECUTE sa_user_admin.set_user_labels(policy_name=>'ACCESS_THONGBAO', user_name=>'GIAMDOCMIENBAC',
max_read_label=>'GD:MB,SX,GC:MB', def_label=>'GD:MB,SX,GC:MB', row_label=>'GD:MB,SX,GC:MB');
EXECUTE sa_user_admin.set_user_labels(policy_name=>'ACCESS_THONGBAO', user_name=>'TRUONGPHONGSXMN',
max_read_label=>'TP:SX:MN', def_label=>'TP:SX:MN', row_label=>'TP:SX:MN');
```

Để phục vụ cho câu b, c, d, ta thực hiện tạo các user minh họa như sau:

- TRUONGPHONGALL: trưởng phòng phụ trách tất cả lĩnh vực không phân biệt chi nhánh
- TRUONGPHONGSXMT: trưởng phòng phụ trách lĩnh vực sản xuất miền Trung
- TRUONGPHONGMBMN: trưởng phòng phụ trách lĩnh vực mua bán miền Nam

```
DROP USER TRUONGPHONGALL;
DROP USER TRUONGPHONGSXMT;
DROP USER TRUONGPHONGMBMN;
```

```
-- truong phong tat ca linh vuc, chi nhanh
CREATE USER TRUONGPHONGALL IDENTIFIED BY TRUONGPHONGALL;
-- truong phong san xuat mien trung
CREATE USER TRUONGPHONGSXMT IDENTIFIED BY TRUONGPHONGSXMT;
-- truong phong mua ban mien nam
CREATE USER TRUONGPHONGMBMN IDENTIFIED BY TRUONGPHONGMBMN;
```

```
GRANT CREATE SESSION TO GIAMDOCALL, GIAMDOCMIENBAC, TRUONGPHONGSXMN, TRUONGPHONGALL, TRUONGPHONGSXMT,TRUONGPHONGMBMN;
GRANT SELECT ON ADMIN01.THONGBAO TO GIAMDOCALL, GIAMDOCMIENBAC, TRUONGPHONGSXMN,TRUONGPHONGALL, TRUONGPHONGSXMT,TRUONGPHONGMBMN;

grant insert on ADMIN01.THONGBAO to NV347, NV348, GIAMDOCMIENBAC ;
GRANT SELECT  ON ADMIN01.THONGBAO TO rNHANVIEN, rQLTRUCTIEP,rTRUONGPHONG,rTRUONGDEAN,rTAICHINH,rNHANSU,rBANGIAMDOC;
```

<u>Câu b:</u> Hãy cho biết cách thức phát tán dòng thông báo t1 đến tất cả trưởng phòng phụ trách tất cả các lĩnh vực không phân biệt chi nhánh.

- Nhãn t1: Trưởng phòng:Mua bán, Sản xuất, Gia công:Miền Bắc, Miền Trung, Miền Nam;

```
connect DoAnATBM_Sys/DoAnATBM_Sys as sysdba;
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP') where DOITUONG = 'TP:ALL:ALL';
```

- Ta thực hiện gán nhãn cho người dùng TRUONGPHONGALL:

```
EXECUTE sa_user_admin.set_user_labels(policy_name=>'ACCESS_THONGBAO', user_name=>'TRUONGPHONGALL',
max_read_label=>'TP:MB,SX,GC:MB,MT,MN', def_label=>'TP:MB,SX,GC:MB,MT,MN', row_label=>'TP:MB,SX,GC:MB,MT,MN');
```

<u>Câu c:</u> Hãy cho biết cách thức phát tán dòng thông báo t2 đến trưởng phòng phụ trách lĩnh vực sản xuất ở miền Trung.

- Nhãn t2: Trưởng phòng: Sản xuất: Miền Trung.

```
connect DoAnATBM_Sys/DoAnATBM_Sys as sysdba;
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:SX:MT') where DOITUONG = 'TP:SX:MT';
```

- Ta thực hiện gán nhãn cho người dùng TRUONGPHONGTPSXMT:

```
EXECUTE sa_user_admin.set_user_labels(policy_name=>'ACCESS_THONGBAO', user_name=>'TRUONGPHONGSXMT',
max_read_label=>'TP:SX:MT', def_label=>'TP:SX:MT', row_label=>'TP:SX:MT');
```

<u>Câu d:</u> Thêm một số kịch bản phát tán dữ liệu nữa trên mô hình OLS đã cài đặt.

d.1. Kịch bản: Phát tán dòng thông báo t3 đến trưởng phòng ở miền Bắc mà không quan tâm đến phụ trách lĩnh vực nào.

- Nhãn t3: Trưởng phòng: Sản xuất, Mua bán, Gia công: Miền Bắc.

```
connect DoAnATBM_Sys/DoAnATBM_Sys as sysdba;
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:MB,SX,GC:MB') where DOITUONG = 'TP:ALL:MB';
```

d.2. Kịch bản: Phát tán dòng thông báo t4 đến trưởng phòng mua bán miền Nam.

- Nhãn t4: Trưởng phòng: Mua bán: Miền Nam

```
connect DoAnATBM_Sys/DoAnATBM_Sys as sysdba;
UPDATE ADMIN01.THONGBAO SET COL_THONGBAO=char_to_label('ACCESS_THONGBAO','TP:MB:MN') where DOITUONG = 'TP:MB:MN';
```

- Ta thực hiện gán nhãn cho người dùng TRUONGPHONGMBMN:

```
EXECUTE sa_user_admin.set_user_labels(policy_name=>'ACCESS_THONGBAO', user_name=>'TRUONGPHONGMBMN',
max_read_label=>'TP:MB:MN', def_label=>'TP:MB:MN', row_label=>'TP:MB:MN');
```

SCRIPT KIỂM TRA OLS:

```
----TEST HE THONG
CONN GIAMDOCALL/GIAMDOCALL;
SELECT * FROM ADMIN01.THONGBAO;
```

Script Output | Task completed in 0.414 seconds

```
Connected.

      MATB DOITUONG
---------- ----------------------------
         1 TP:ALL:ALL
         2 TP:SX:MN
         3 TP:SX:MB
         4 TP:SX:MT
         5 TP:ALL:MB
         6 TP:ALL:ALL
         7 TP:SX:MT
         8 TP:MB:ALL
         9 TP:SX:MT
        10 TP:SX:MT
        11 TP:MB:MN

      MATB DOITUONG
---------- ----------------------------
        12 GD:SX:ALL
        30 TP:SX:MB

13 rows selected.
```

```
CONN GIAMDOCMIENBAC/GIAMDOCMIENBAC;
SELECT * FROM ADMIN01.THONGBAO;
```

Script Output | Task completed in 0.074 seconds

```
Connected.

      MATB DOITUONG
---------- ----------------------------
         1 TP:ALL:ALL
         3 TP:SX:MB
         5 TP:ALL:MB
         6 TP:ALL:ALL
         8 TP:MB:ALL
        12 GD:SX:ALL
        30 TP:SX:MB

7 rows selected.
```

```
conn TRUONGPHONGSXMT/TRUONGPHONGSXMT;
select * from ADMIN01.THONGBAO;
```

Script Output | Task completed in 0.066 seconds

```
Connected.

      MATB DOITUONG
---------- ----------------------------
         1 TP:ALL:ALL
         4 TP:SX:MT
         6 TP:ALL:ALL
         7 TP:SX:MT
         9 TP:SX:MT
        10 TP:SX:MT

6 rows selected.
```

```
conn TRUONGPHONGMBMN/TRUONGPHONGMBMN;
select * from ADMIN01.THONGBAO;
```

Script Output | Task completed in 0.064 seconds

```
Connected.

      MATB DOITUONG
---------- ----------------------------
         1 TP:ALL:ALL
         6 TP:ALL:ALL
         8 TP:MB:ALL
        11 TP:MB:MN
```

```
conn TRUONGPHONGALL/TRUONGPHONGALL;
select * from ADMIN01.THONGBAO;
```

Script Output | Task completed in 0.066 seconds

```
Connected.

      MATB DOITUONG
---------- ----------------------------
         1 TP:ALL:ALL
         2 TP:SX:MN
         3 TP:SX:MB
         4 TP:SX:MT
         5 TP:ALL:MB
         6 TP:ALL:ALL
         7 TP:SX:MT
         8 TP:MB:ALL
         9 TP:SX:MT
        10 TP:SX:MT
        11 TP:MB:MN

      MATB DOITUONG
---------- ----------------------------
        30 TP:SX:MB
```

```
CONN TRUONGPHONGSXMN/TRUONGPHONGSXMN;
SELECT * FROM ADMIN01.THONGBAO;
```

Script Output | Task completed in 0.108 seconds

```
Connected.

      MATB DOITUONG
---------- ----------------------------
         1 TP:ALL:ALL
         2 TP:SX:MN
         6 TP:ALL:ALL
```

5. Audit:

4.1. Lý thuyết:

Khái niệm: Audit là hoạt động theo dõi và ghi lại nhật ký các hoạt động, thao tác của người dùng trên cơ sở dữ liệu.

Ngữ cảnh: Kiểm tra nhật ký các hoạt động liên quan đến thêm/xóa/sửa/chọn trên tất cả các bảng thuộc lược đồ cơ sở dữ liệu

Mục đích: Auditing cho phép ta bắt các user phải có trách nhiệm về hành động mà họ thực hiện, bằng cách theo dõi hành vi của họ.

- Dữ liệu audit giúp phát hiện lỗ hổng trong chính sách bảo mật.
- Liên quan đến trách nhiệm giải trình của user. Cần phải đảm bảo rằng user chỉ được thực hiện những gì họ được phép. Ghi nhận sự lạm quyền hoặc dùng sai quyền.
- Auditing để ghi nhận lại những gì đã xảy ra và có hồi đáp thích hợp.
- Không thực hiện auditing ta sẽ không thể biết khía cạnh bảo mật của hệ thống có đảm bảo hay không hay có ai đã đọc hoặc cập nhật dữ liệu một cách bất hợp pháp hay không.
- Việc auditing hiệu quả khi: Có kế hoạch thực hiện auding. Đọc lại và phân tích dữ liệu của quá trình auditing.

Lưu ý:

- Auditing tất cả các hành động của tất cả các user trên tất cả dữ liệu sẽ không có ích mà còn làm chậm hệ thống, và dữ liệu có được từ quá trình audit khó sử dụng.
- Audit một cách có chọn lọc và đúng đắn, dựa trên dữ liệu, xử lý và người dùng có thật.

4.2. Cài đặt Audit

a. Những người đã cập nhật trường THOIGIAN trong quan hệ PHANCONG

Ý tưởng: Vì trường THOIGIAN trong quan hệ PHANCONG được truy xuất trực tiếp trên table, nên ta sẽ thực hiện audit trực tiếp trên trường THOIGIAN của quan hệ PHANCONG.

Scrip cài đặt audit:

```
-------------------------- Audit nhung nguoi da cap nhat truong THOIGIAN trong quan he PHANCONG--
conn admin01/1;
EXEC DBMS_FGA.DISABLE_POLICY(POLICY_NAME=>'audit_update_time_phancong',OBJECT_NAME=>'PHANCONG');
EXEC DBMS_FGA.DROP_POLICY(POLICY_NAME=>'audit_update_time_phancong',OBJECT_NAME=>'PHANCONG');
BEGIN
    DBMS_FGA.add_policy
    (
        object_schema => 'ADMIN01',
        object_name => 'PHANCONG',
        policy_name => 'audit_update_time_phancong',
        audit_column => 'THOIGIAN',
        statement_types =>'UPDATE'
    );
END;
/
EXEC DBMS_FGA.ENABLE_POLICY(POLICY_NAME=>'audit_update_time_phancong',OBJECT_NAME=>'PHANCONG');
```

Scrip thực hiện hành vi kích hoạt audit:

```
--Cho nhan vien NV323 update truong THOIGIAN
conn NV323/123;
select * from ADMIN01.PHANCONG;
update ADMIN01.PHANCONG set THOIGIAN = TO_DATE('2023-01-11', 'YYYY-MM-DD') where MANV='NV343';
--Kiem tra user da update
conn admin01/1
select db_user, timestamp, sql_text from  dba_fga_audit_trail;
```

Script Output

Task completed in 0.432 seconds

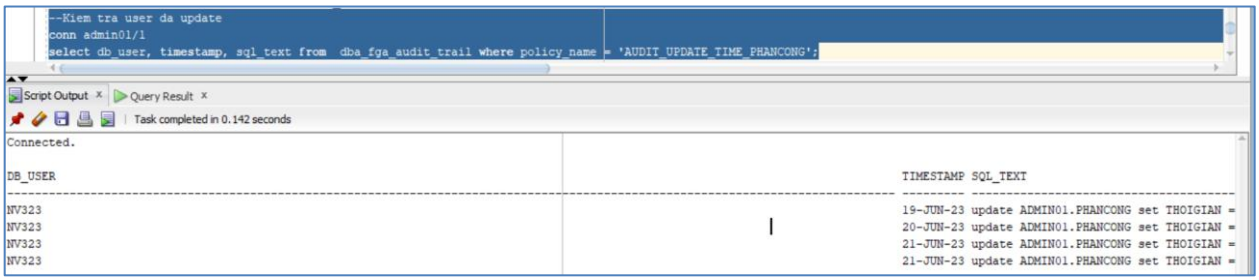
```
MANV  MADA  THOIGIAN
----- ----- ---------
NV111 DA05  22-APR-23

34 rows selected.


1 row updated.

Connection created by CONNECT script command disconnected
```

Scrip kiểm tra audit:



b. Những người đã đọc trên trường LUONG và PHUCAP của người khác

Ý tưởng: Vì trường LUONG và PHUCAP trong quan hệ NHANVIEN không thể truy xuất trực tiếp bởi các vai trò thông thường. Nên ta sẽ kiểm tra các vai trò có thể truy xuất vào 2 trường này, thì chỉ có Tài chính được truy xuất thông qua view "vw_TAICHINH_Xem_NHANVIEN". Do đó, ta gán audit cho view này để kiểm soát việc truy xuất LUONG và PHUCAP của Tài chính. Ngoài ra, để đảm bảo ta cũng sẽ cài Audit trên table NHANVIEN.

Scrip cài đặt audit:

- Audit LUONG trên table NHANVIEN

```
--audit LUONG tren table
conn admin01/1;
EXEC DBMS_FGA.DISABLE_POLICY(POLICY_NAME=>'audit_select_luong_nhanvien',OBJECT_NAME=>'NHANVIEN');
EXEC DBMS_FGA.DROP_POLICY(POLICY_NAME=>'audit_select_luong_nhanvien',OBJECT_NAME=>'NHANVIEN');
BEGIN
    DBMS_FGA.add_policy
    (
        object_schema => 'ADMIN01',
        object_name => 'NHANVIEN',
        policy_name => 'audit_select_luong_nhanvien',
        audit_condition => 'MANV != user',
        audit_column => 'LUONG',
        statement_types =>'SELECT'
    );
END;
/
conn admin01/1;
EXEC DBMS_FGA.ENABLE_POLICY(POLICY_NAME=>'audit_select_luong_nhanvien',OBJECT_NAME=>'NHANVIEN');
```

- Audit LUONG trên view vw_TAICHINH_Xem_NHANVIEN

```
--audit LUONG tren view vw_TAICHINH_Xem_NHANVIEN
conn admin01/1;
EXEC DBMS_FGA.DISABLE_POLICY(POLICY_NAME=>'audit_select_luong_nhanvien',OBJECT_NAME=>'vw_TAICHINH_Xem_NHANVIEN');
EXEC DBMS_FGA.DROP_POLICY(POLICY_NAME=>'audit_select_luong_nhanvien',OBJECT_NAME=>'vw_TAICHINH_Xem_NHANVIEN');
BEGIN
    DBMS_FGA.add_policy
    (
        object_schema => 'ADMIN01',
        object_name => 'vw_TAICHINH_Xem_NHANVIEN',
        policy_name => 'audit_select_luong_nhanvien',
        audit_condition => 'MANV != user',
        audit_column => 'LUONG',
        statement_types =>'SELECT'
    );
END;
/
conn admin01/1;
EXEC DBMS_FGA.ENABLE_POLICY(POLICY_NAME=>'audit_select_luong_nhanvien',OBJECT_NAME=>'vw_TAICHINH_Xem_NHANVIEN');
```

- Audit PHUCAP trên table NHANVIEN

```
--audit PHUCAP tren table
conn admin01/1;
EXEC DBMS_FGA.DISABLE_POLICY(POLICY_NAME=>'audit_select_phucap_nhanvien',OBJECT_NAME=>'NHANVIEN');
EXEC DBMS_FGA.DROP_POLICY(POLICY_NAME=>'audit_select_phucap_nhanvien',OBJECT_NAME=>'NHANVIEN');
BEGIN
    DBMS_FGA.add_policy
    (
        object_schema => 'ADMIN01',
        object_name => 'NHANVIEN',
        policy_name => 'audit_select_phucap_nhanvien',
        audit_condition => 'MANV != user',
        audit_column => 'PHUCAP',
        statement_types =>'SELECT'
    );
END;
/
conn admin01/1;
EXEC DBMS_FGA.ENABLE_POLICY(POLICY_NAME=>'audit_select_phucap_nhanvien',OBJECT_NAME=>'NHANVIEN');
```

- Audit PHUCAP trên view vw_TAICHINH_Xem_NHANVIEN

```
--audit PHUCAP tren view vw_TAICHINH_Xem_NHANVIEN
conn admin01/1;
EXEC DBMS_FGA.DISABLE_POLICY(POLICY_NAME=>'audit_select_phucap_nhanvien',OBJECT_NAME=>'vw_TAICHINH_Xem_NHANVIEN');
EXEC DBMS_FGA.DROP_POLICY(POLICY_NAME=>'audit_select_phucap_nhanvien',OBJECT_NAME=>'vw_TAICHINH_Xem_NHANVIEN');
BEGIN
    DBMS_FGA.add_policy
    (
        object_schema => 'ADMIN01',
        object_name => 'vw_TAICHINH_Xem_NHANVIEN',
        policy_name => 'audit_select_phucap_nhanvien',
        audit_condition => 'MANV != user',
        audit_column => 'PHUCAP',
        statement_types =>'SELECT'
    );
END;
/
conn admin01/1;
EXEC DBMS_FGA.ENABLE_POLICY(POLICY_NAME=>'audit_select_phucap_nhanvien',OBJECT_NAME=>'vw_TAICHINH_Xem_NHANVIEN');
```

Scrip thực hiện hành vi kích hoạt audit:

```
--Cho user TaiChinh select LUONG, PHUCAP cua ban than va cua nguoi khac
conn NV331/123;
select * from ADMIN01.vw_TAICHINH_Xem_NHANVIEN where MANV='NV331';
select * from ADMIN01.vw_TAICHINH_Xem_NHANVIEN;
select LUONG from ADMIN01.vw_TAICHINH_Xem_NHANVIEN where MANV='NV331';
select LUONG from ADMIN01.vw_TAICHINH_Xem_NHANVIEN;
select PHUCAP from ADMIN01.vw_TAICHINH_Xem_NHANVIEN where MANV='NV331';
select PHUCAP from ADMIN01.vw_TAICHINH_Xem_NHANVIEN;
```

Script Output

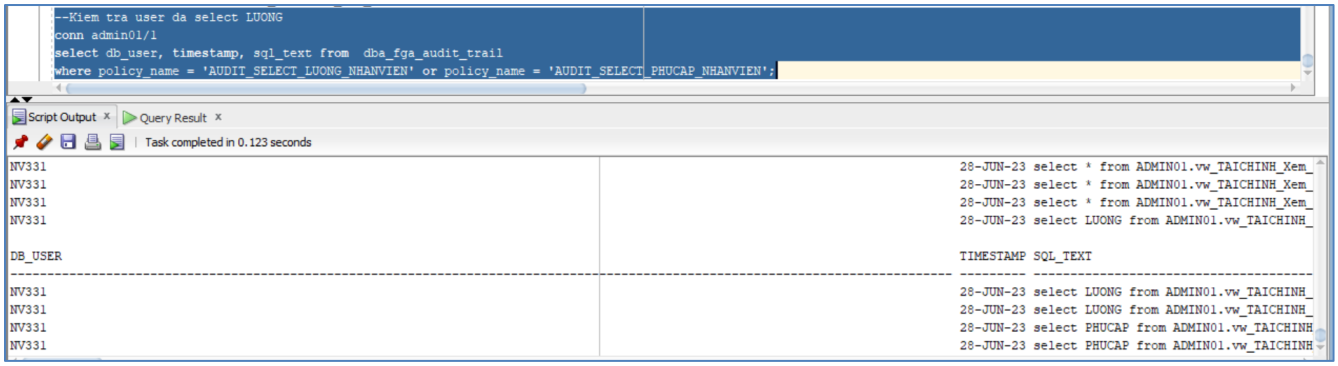
Task completed in 1.431 seconds

```
3868136
3591608
1289742
1744791
4619188

346 rows selected.

Connection created by CONNECT script command disconnected
```

Scrip kiểm tra audit:



c. Một người không thuộc vai trò "Tài chính" nhưng đã cập nhật thành công trên trường LUONG và PHUCAP.

Ý tưởng: Giả sử có một user HACKER01 (không thuộc vai trò "Tài chính") tấn công nhắm vào trường LUONG va PHUCAP, khi đó ta sẽ thực hiện audit trên trường LUONG và PHUCAP của quan hệ NHANVIEN.

```
conn ADMIN01/1;
ALTER SESSION SET "_ORACLE_SCRIPT"=TRUE;
DROP USER HACKER01 CASCADE;
create user HACKER01 IDENTIFIED BY 123 ;
grant create session to HACKER01;
grant connect to HACKER01;
grant select, update(luong,phucap) on ADMIN01.NHANVIEN to HACKER01;
grant select, update(luong,phucap) on ADMIN01.vw_TAICHINH_Xem_NHANVIEN to HACKER01;
```

Scrip cài đặt audit:

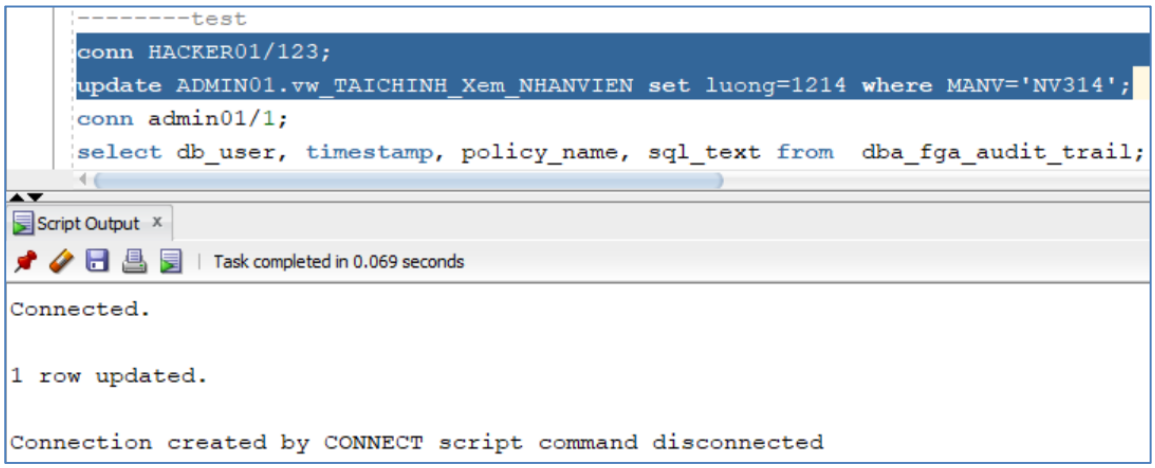
- Audit LUONG trên table NHANVIEN

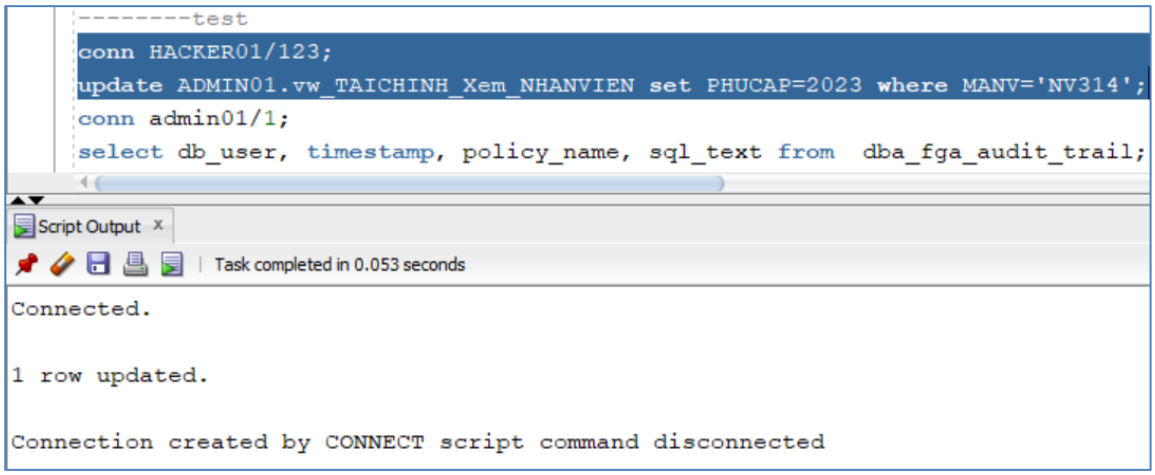
```
--audit update LUONG
conn admin01/1;
EXEC DBMS_FGA.DISABLE_POLICY(POLICY_NAME=>'audit_hacker_update_luong_nhanvien',OBJECT_NAME=>'NHANVIEN');
EXEC DBMS_FGA.DROP_POLICY(POLICY_NAME=>'audit_hacker_update_luong_nhanvien',OBJECT_NAME=>'NHANVIEN');
BEGIN
    DBMS_FGA.add_policy
    (
        object_schema => 'ADMIN01',
        object_name => 'NHANVIEN',
        policy_name => 'audit_hacker_update_luong_nhanvien',
        audit_condition => 'VAITRO != ''Tai chinh''',
        audit_column => 'LUONG',
        statement_types =>'UPDATE'
    );
END;
/
EXEC DBMS_FGA.ENABLE_POLICY(POLICY_NAME=>'audit_hacker_update_luong_nhanvien',OBJECT_NAME=>'NHANVIEN');
```

- Audit PHUCAP trên table NHANVIEN

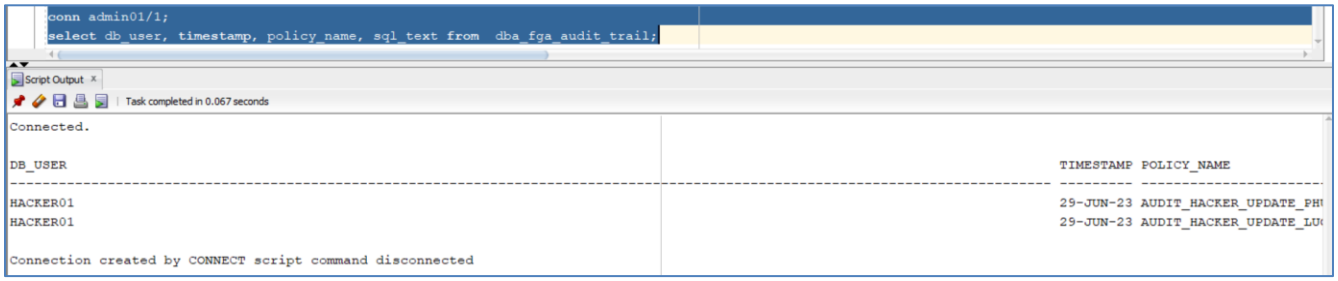
```
--audit update PHUCAP tren table
conn admin01/1;
EXEC DBMS_FGA.DISABLE_POLICY(POLICY_NAME=>'audit_hacker_update_phucap_nhanvien',OBJECT_NAME=>'NHANVIEN');
EXEC DBMS_FGA.DROP_POLICY(POLICY_NAME=>'audit_hacker_update_phucap_nhanvien',OBJECT_NAME=>'NHANVIEN');
BEGIN
    DBMS_FGA.add_policy
    (
        object_schema => 'ADMIN01',
        object_name => 'NHANVIEN',
        policy_name => 'audit_hacker_update_phucap_nhanvien',
        audit_condition => 'VAITRO != ''Tai chinh''',
        audit_column => 'PHUCAP',
        statement_types =>'UPDATE'
    );
END;
/
EXEC DBMS_FGA.ENABLE_POLICY(POLICY_NAME=>'audit_hacker_update_phucap_nhanvien',OBJECT_NAME=>'NHANVIEN');
```

Scrip thực hiện hành vi kích hoạt audit:





Scrip kiểm tra audit:



```
TIMESTAMP POLICY_NAME
--------- ------------------------------------
29-JUN-23 AUDIT_HACKER_UPDATE_PHUCAP_NHANVIEN
29-JUN-23 AUDIT_HACKER_UPDATE_LUONG_NHANVIEN
```

```
SQL_TEXT
--------------------------------------------------------
UPDATE ADMIN01.NHANVIEN SET PHUCAP = :B2 WHERE MANV = :B1
UPDATE ADMIN01.NHANVIEN SET LUONG = :B2 WHERE MANV = :B1
```

d. Kiểm tra nhật ký hệ thống.

Ta thực hiện kiểm tra nhật ký hệ thống bằng câu lệnh:

```
select db_user, timestamp, sql_text from dba_fga_audit_trail;
```

Kết quả cụ thể xem ở từng câu a, b, c phía trên.

C. Tài liệu tham khảo:

- Các slide môn học do giảng viên cung cấp.
- Audit
- FINE-GRAINED AUDITING
- OLS
- VPD
- RBAC
- DAC