ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA: CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN
MÔN: AN TOÀN VÀ BẢO MẬT DỮ LIỆU TRONG HỆ THỐNG THÔNG TIN

NHÓM THỰC HIỆN –CQ2021/1-18:
MSSV:21120612- Nguyễn Minh Thuận
MSSV: Ngũ Duy Tính
MSSV: 2112578 – HỌ TÊN: Trần Minh Triết
MSSV: 20120424 – HỌ TÊN: Dương Khánh An
Giảng viên lý thuyết: TS. Phạm Thị Bạch Huệ
Giảng viên thực hành: ThS. Lương Vĩ Minh – Tiết Gia Hồng Lớp lý thuyết: 21_1
Học kỳ - Niên khoá: HK2 - 2023-2024

MỤC LỤC

THÔNG TIN CHUNG VỀ ĐỒ ÁN

THÔNG TIN THÀNH VIÊN

Bảng thông tin thành viên của nhóm S18 trong đồ án:

MÃ SỐ SINH VIÊN	HỌ VÀ TÊN	PHẦN TRĂM ĐÓNG GÓP
21120612	Nguyễn Minh Thuận(Nhóm trưởng)	30%
21210572	Ngũ Duy Tính	35%
20120188	Trần Minh Triết	35%
20120424	Dương Khánh An	0%

BẢNG PHÂN CÔNG CÔNG VIỆC TRONG ĐỒ ÁN VÀ MỨC ĐỘ HOÀN THÀNH

Phân hệ 1:

CÔNG VIỆC	NGƯỜI THỰC HIỆN	MỨC ĐỘ HOÀN THÀNH
Tạo database + Insert dữ liệu	Nguyễn Minh Thuận	100% - Đã hoàn thành
Connect Database làm form login, logout, viết báo cáo, Tổng hợp code, Demo	Nguyễn Minh Thuận	100% - Đã hoàn thành
Thông tin về quyền (privileges) của mỗi user/ role trên các đối tượng dữ liệu.	Nguyễn Minh Thuận	100% - Đã hoàn thành
Cho phép tạo mới, xóa, sửa (hiệu chỉnh) user hoặc role.	Trần Minh Triết	100% - Đã hoàn thành

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.	Ngũ Duy Tính	100% - Đã hoàn thành

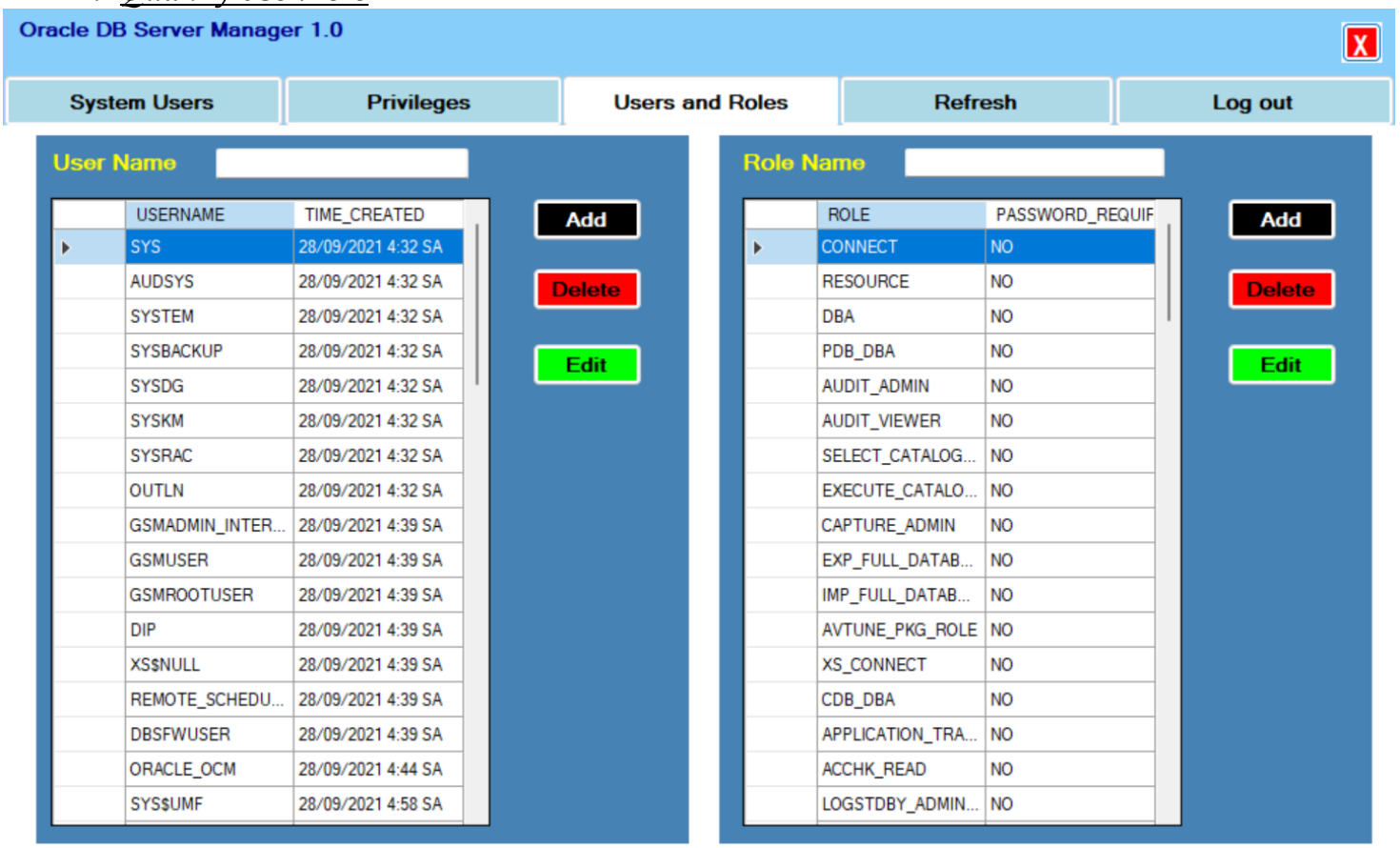
Phân hệ 2:

CÔNG VIỆC	NGƯỜI THỰC HIỆN	MỨC ĐỘ HOÀN THÀNH
Tạo database + Insert dữ liệu, thiết kế cơ sở dữ liệu, Tổng hợp viết báo cáo	Nguyễn Minh Thuận	100% - Đã hoàn thành
Connect Database làm form login, logout, Phần quyền	Trần Minh Triết	100% - Đã hoàn thành
Cài đặt chính sách 1, chính sách 2	Ngũ Duy Tính	100% - Đã hoàn thành
Cài đặt chính sách 3, chính sách 4	Trần Minh Triết	100% - Đã hoàn thành
Cài đặt chính sách 5 và chính sách 6	Nguyễn Minh Thuận	100% - Đã hoàn thành
Yêu cầu 2 : OLS	Ngũ Duy Tính	100% - Đã hoàn thành
Yêu cầu 3: Ghi Nhật ký hệ thống	Trần Minh Triết	100% - Đã hoàn thành
Yêu cầu 4: Sao lưu và phục hồi dữ liệu	Nguyễn Minh Thuận	40%

PHẦN BÁO CÁO

I. Phân hệ 1: Dành cho người quản trị cơ sở dữ liệu

1. *Quản lý user/role*



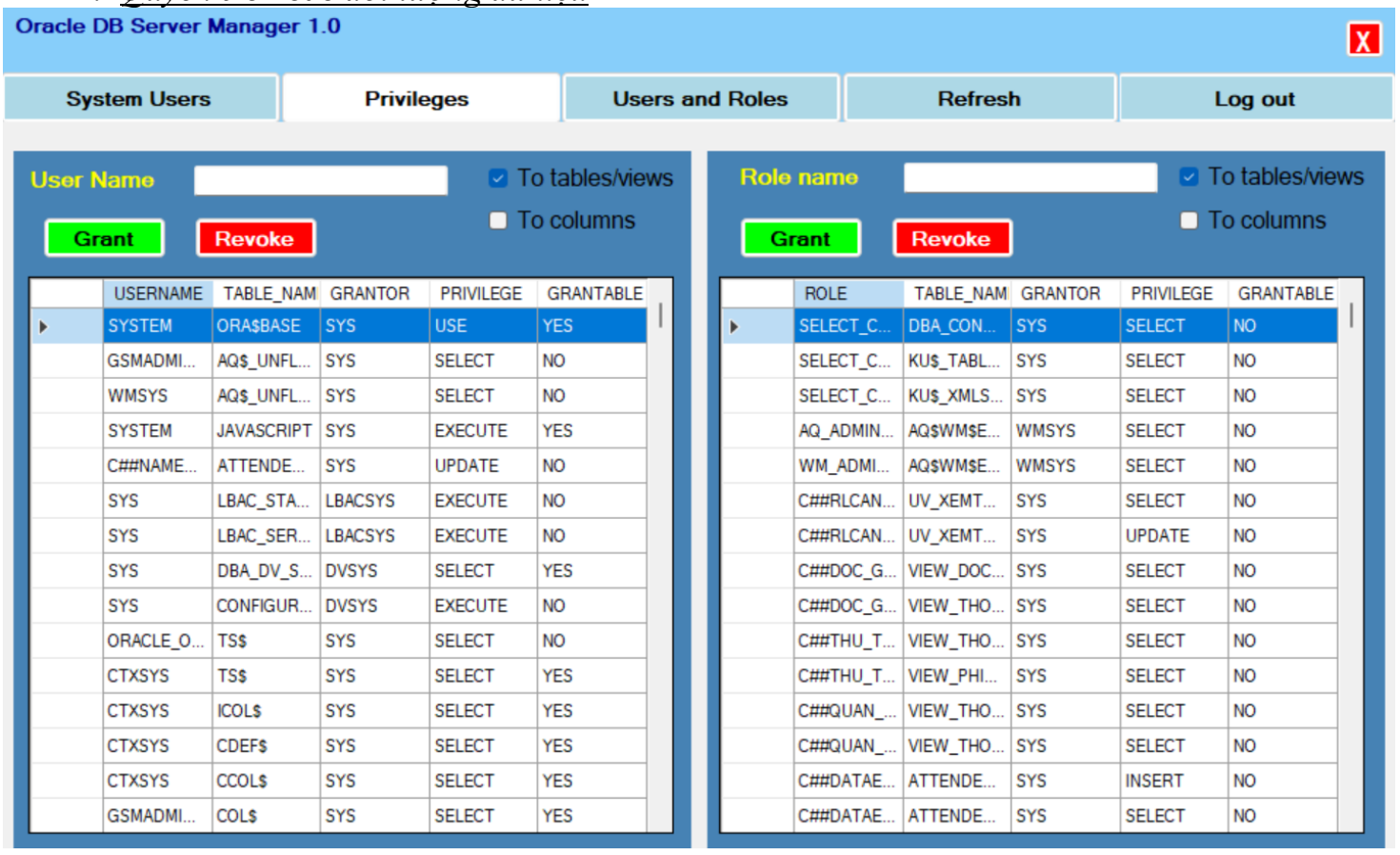
Trong ứng dụng này, chúng em tạo user admin là CAMPUSADMIN với các role và quyền như bên dưới:

```
ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE;
    CREATE USER CAMPUSADMIN IDENTIFIED BY 1;
    GRANT DBA TO CAMPUSADMIN;
    GRANT EXECUTE ANY PROCEDURE TO CAMPUSADMIN;
    GRANT CREATE SESSION TO CAMPUSADMIN;
    GRANT CREATE USER TO CAMPUSADMIN;
    GRANT CREATE ROLE TO CAMPUSADMIN;
    GRANT CONNECT TO CAMPUSADMIN WITH ADMIN OPTION;
```

Sử dụng các câu lệnh sau để thực hiện chức năng:

- Xem danh sách user hiện có:
    - SELECT USERNAME, USER_ID, CREATED FROM ALL_USERS
- Xem danh sách role hiện có:
    - SELECT ROLE, ROLE_ID, PASSWORD_REQUIRED FROM DBA_ROLES
- Tạo user:
    - CREATE USER username IDENTIFIED BY password;
- Cập nhật user (mật khẩu)
    - ALTER USER username IDENTIFIED BY new_password;
- Xóa user:
    - DROP USER username;
- Tạo role (có hoặc không có mật khẩu):
    - CREATE ROLE username {IDENTIFIED BY password};
- Cập nhật role (mật khẩu):
    - ALTER ROLE username {NOT IDENTIFIED/IDENTIFIED BY new_password};
- Xóa role:
    - DROP ROLE rolename;

2. *Quyền trên các đối tượng dữ liệu*



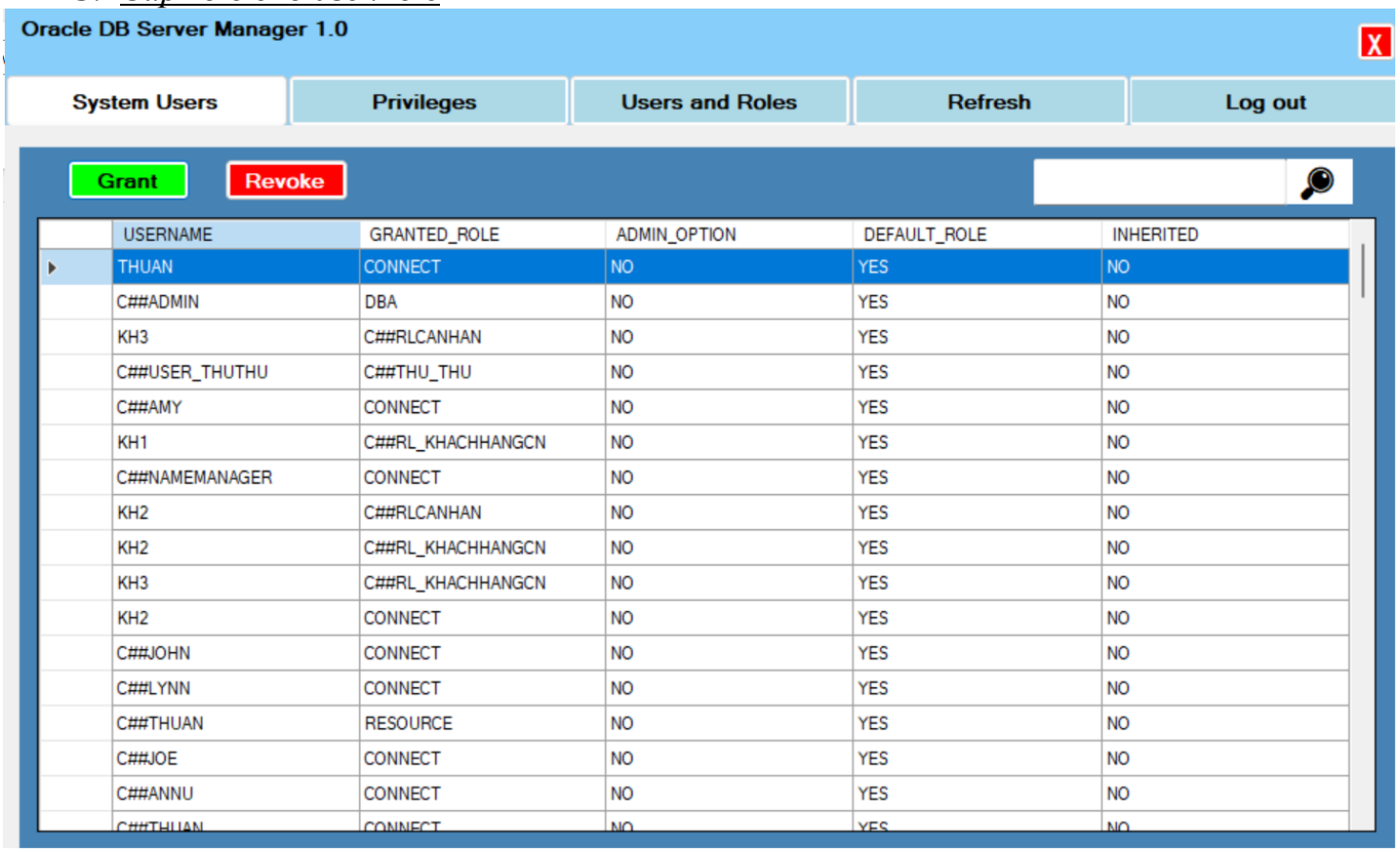
- Xem thông tin quyền trên TABLE và COLUMN của các user và role trên hệ thống:
    - select * from DBA_TAB_PRIVS where TABLE_NAME LIKE 'QLDA_%' OR

TABLE_NAME LIKE 'V_QLDA_%'

- select * from DBA_COL_PRIVS where TABLE_NAME LIKE 'QLDA_%'

- Cấp quyền cho role/user:
  - Đối với quyền SELECT: trên thực tế, Oracle không cho phép cấp quyền SELECT trên thuộc tính trực tiếp trên một table, nên khi cấp quyền SELECT trên thuộc tính, chúng em sẽ tạo một view với table và các thuộc tính được truy xuất và cấp quyền trên view đó.
  - Đối với quyền UPDATE: Oracle cho phép cấp quyền đến mức thuộc tính.
  - Quyền INSERT, UPDATE: chỉ cấp quyền cho toàn bộ table, không cấp riêng cho thuộc tính.
- Câu lệnh cấp quyền cho role/user
  - GRANT privilege TO username/rolename
- Lấy lại quyền của user/role:
  - REVOKE privilege TO username/rolename

3. *Cấp role cho user/role*



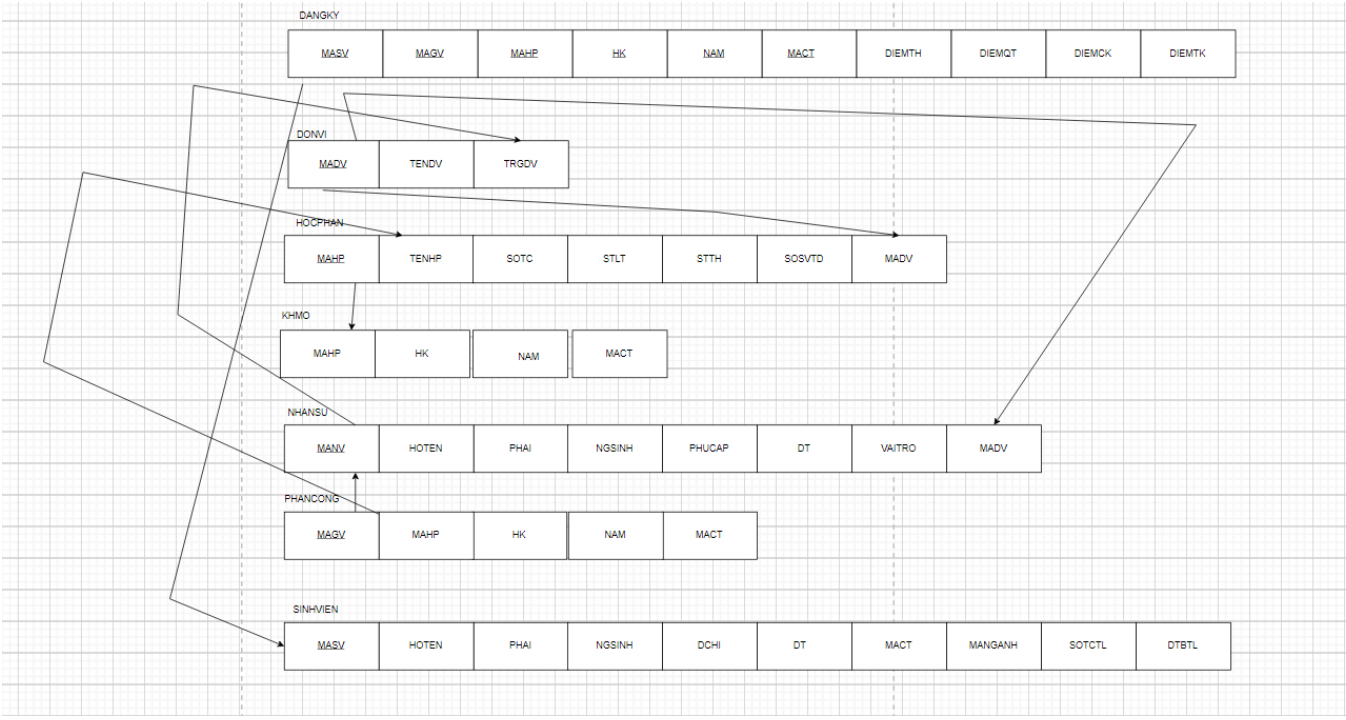
USERNAME	GRANTED_ROLE	ADMIN_OPTION	DEFAULT_ROLE	INHERITED
THUAN	CONNECT	NO	YES	NO
C##ADMIN	DBA	NO	YES	NO
KH3	C##RLCANHAN	NO	YES	NO
C##USER_THUTHU	C##THU_THU	NO	YES	NO
C##AMY	CONNECT	NO	YES	NO
KH1	C##RL_KHACHHANGCN	NO	YES	NO
C##NAMEMANAGER	CONNECT	NO	YES	NO
KH2	C##RLCANHAN	NO	YES	NO
KH2	C##RL_KHACHHANGCN	NO	YES	NO
KH3	C##RL_KHACHHANGCN	NO	YES	NO
KH2	CONNECT	NO	YES	NO
C##JOHN	CONNECT	NO	YES	NO
C##LYNN	CONNECT	NO	YES	NO
C##THUAN	RESOURCE	NO	YES	NO
C##JOE	CONNECT	NO	YES	NO
C##ANNU	CONNECT	NO	YES	NO
C##THUAN	CONNECT	NO	YES	NO

- Oracle cho phép cấp Role cho một user và cả cấp Role cho Role.
- Câu lệnh thực hiện cấp Role cho User/Role:
  - GRANT role TO username/rolename;
- Việc cấp role cho role là cần thiết vì có thể ta cần phải cấp role hệ thống cho role (như role DBA, CONNECT)

II. Phân hệ 2: Tạo và áp đặt chính sách bảo mật, mã hóa và ghi vết người dùng

1. *Chính sách điều khiển truy cập (Access Control)*

Lược đồ cơ sở dữ liệu:



Phát biểu lại các chính sách:

- Chính sách 1: Những người có VAITRO là ‘Nhân viên cơ bản’. Một người dùng NS có vai trò ‘Nhân viên cơ bản’ có quyền được mô tả như sau.
    - Xem thông tin tài khoản bản thân, Update sđt bản thân
    - Xem SINHVIEN, HOCPHAN, KHMO, DONVI.
- Chính sách 2: Những người có VAITRO là ‘Giảng viên’. Một người dùng NS có vai trò ‘Giảng viên’ có quyền được mô tả như sau.
    - Như một người dùng có vai trò là ‘Nhân viên cơ bản’.
    - Xem SINHVIEN, HOCPHAN, KHMO, DONVI.
    - Xem các PHANCONG của bản thân
    - Xem DANGKY có liên quan đến bản thân
    - Sửa các trường Diem trong quan hệ DANGKY có liên quan đến bản thân.
- Chính sách 3: Những người có VAITRO là ‘giáo vụ’. Một người dùng NS có vai trò ‘giáo vụ’ có quyền được mô tả như sau.
    - Như một người dùng có vai trò là ‘Nhân viên cơ bản’.
    - Xem, thêm mới, cập nhật SINHVIEN, HOCPHAN, PHANCONG, DONVI.
    - Xem toàn bộ quan hệ PHANCONG, nhưng chỉ chỉnh sửa trên học phần liên quan đến ‘văn phòng khoa’.

- Xóa, thêm mới quan hệ DANGKY trong thời gian quy định.

- Chính sách 4: Những người có VAITRO là 'trưởng đơn vị'. Một người dùng NS có vai trò 'trưởng đơn vị có quyền được mô tả như sau'
  - Như một người dùng có vai trò là giảng viên.
  - Xem, xóa, cập nhật PHANCONG học phần quản lý bởi văn phòng khoa.
  - Xem dữ liệu PHANCONG các giảng viên thuộc đơn vị mà mình làm trưởng đơn vị.
- Chính sách 5: Những người có VAITRO là "**trưởng khoa**". Một người dùng NS có vai trò "trưởng khoa" có quyền được mô tả như sau:
  - Như một người dùng có vai trò "Giảng viên".
  - Thêm, Xóa, Cập nhật dữ liệu trên quan hệ PHANCONG đối với các học phần quản lý bởi đơn vị "Văn phòng khoa".
  - Xem, Thêm, Xóa, Cập nhật trên quan hệ NHANSU.
  - Xem (không giới hạn) dữ liệu trên toàn bộ lược đồ CSDL.
- Chính sách 6: Những người dùng có VAITRO là "Sinh viên" cho biết đó là. Một người dùng là "Sinh viên " có quyền được mô tả như sau:
  - Trên quan hệ SINHVIEN, sinh viên chỉ được xem thông tin của chính mình, được chỉnh sửa thông tin địa chỉ (ĐCHI) và số điện thoại liên lạc (ĐT) của chính sinh viên.
  - Xem danh sách tất cả học phần (HOCPHAN), kế hoạch mở môn (KHMO) của chương trình đào tạo mà sinh viên đang theo học.
  - Thêm, Xóa các dòng dữ liệu đăng ký học phần (DANGKY) liên quan đến chính sinh viên đó trong học kỳ của năm học hiện tại (nếu thời điểm hiệu chỉnh đăng ký còn hợp lệ).
  - Sinh viên không được chỉnh sửa trên các trường liên quan đến điểm.

Kịch bản cài đặt:

- **Chính sách 1:**
  Kịch bản và phương thức.
  - Xem dữ liệu trên các quan hệ SINHVIEN, ĐONVI, HOCPHAN, KHMO.
    - Dùng RBAC: Tạo role NHANVIENCOBAN và gán các quyền xem dữ liệu cho role vừa tạo. Gán role NHANVIENCOBAN cho người dùng.
  - Xem dữ liệu dữ liệu của bản thân trên USER
    - Dùng View : Tạo view select dữ liệu trên bảng USER những học phần mà có USER_NAME = USER giữ session hiện tại.
- **Chính sách 2:**
  Kịch bản và phương thức.
  - Xem dữ liệu trên các quan hệ SINHVIEN, ĐONVI, HOCPHAN, KHMO, theo yêu cầu của trưởng khoa.

- Dùng RBAC: Tạo role GIANGVIEN và gán các quyền xem dữ liệu cho role vừa tạo. Gán role GIANGVIEN cho người dùng.

o Xem dữ liệu dữ liệu của bản thân trên USER

- Dùng View : Tạo view select dữ liệu trên bảng USER những học phần mà có USER_NAME = USER giữ session hiện tại.

o Xem dữ liệu dữ liệu của bản thân trên PHANCONG, DANGKY

- Dùng View + RBAC: Tạo view select dữ liệu trên bảng PHANCONG, DANGKY sao cho MAGV = USER giữ session hiện tại và gán quyền SELECT 2 view đó cho role GIANGVIEN.

o Cập dữ liệu dữ liệu điểm các sinh viên có liên quan đến bản thân trên DANGKY

- Dùng RBAC: Gán quyền Update các cột điểm trên bảng DANGKY cho người dùng GIANGVIEN

- **Chính sách 3:**
Kịch bản và phương thức.

o Xem, Thêm mới hoặc Cập nhật dữ liệu trên các quan hệ SINHVIEN, ĐONVI, HOCPHAN, KHMO, theo yêu cầu của trưởng khoa.

- Dùng RBAC: Tạo role GIAOVU và gán các quyền xem sửa dữ liệu cho role vừa tạo. Gán role GIAOVU cho người dùng.

o Xem dữ liệu trên toàn bộ quan hệ PHANCONG. Tuy nhiên, chỉ được sửa trên các dòng dữ liệu phân công liên quan các học phần do "Văn phòng khoa" phụ trách phân công giảng dạy, thừa hành người trưởng đơn vị tương ứng là trưởng khoa.

- Dùng View + RBAC: Tạo view select dữ liệu trên bảng PHANCONG những học phần mà 'Văn phòng khoa' phân công phụ trách giảng dạy. Gán quyền chỉnh sửa trên view cho role GIAOVU.

o Xóa hoặc Thêm mới dữ liệu trên quan hệ ĐANGKY theo yêu cầu của sinh viên trong khoảng thời gian còn cho hiệu chỉnh đăng ký, xem điều kiện có thể hiệu chỉnh đăng ký học phần được mô tả bên dưới..

- Dùng View + RBAC: Select dữ liệu được phép đăng ký trong khoản thời gian quy định. Gán quyền xem và sửa trên View cho role GIAOVU.

- **Chính sách 4:**
Kịch bản và phương thức.

o Thêm, Xóa, Cập nhật dữ liệu trên quan hệ PHANCONG, đối với các học phần được phụ trách chuyên môn bởi đơn vị mà mình làm trưởng,

- Sử dụng RBAC + view.
- Tạo view lấy dữ liệu bảng PHANCONG được phụ trách chuyên môn bởi nhân viên đang đăng nhập.
- Gán quyền SELECT, INSERT, UPDATE, DELETE cho role TRUONGDONVI. Tạo trigger thay đổi database khi view thay đổi.

o Được xem dữ liệu phân công giảng dạy của các giảng viên thuộc các đơn vị mà mình làm trưởng.

- Sử dụng RBAC + view.
- Tạo view lấy dữ liệu bảng PHANCONG đổi với các HOCPHAN do mình làm trưởng.
- Gán quyền SELECT trên view cho role TRUONGDONVI.

- Chính sách 5:
  - Chủ thể: Người dùng có role là trưởng khoa
  - Quyền:
    - PHANCONG: insert, delete, update đối với các học phần quản lý bởi đơn vị "*Văn phòng khoa*"
    - NHANSU: Xem, thêm, xóa, cập nhật
    - Xem (không giới hạn) dữ liệu trên toàn bộ lược đồ CSDL
  - Kịch bản:
    - Tạo vai trò Trưởng khoa và gán các quyền cần thiết
    - Tạo người dùng TRUONG_KHOA với mật khẩu là '1'
    - Gán vai trò Trưởng khoa cho người dùng TRUONG_KHOA
    - Cấp quyền CREATE SESSION cho người dùng TRUONG_KHOA để có thể đăng nhập
    - Tạo hàm chính sách RLS cho vai trò Trưởng khoa: viết function policy thực hiện kiểm tra vai trò của user trả về một chuỗi vị từ kiểm tra xem có phải là trường khoa hoặc là giảng viên giảng dạy môn học thuộc văn phòng khoa hay không

```
CREATE OR REPLACE FUNCTION policy_func_rls_truongkhoa (schema_name IN VARCHAR2, table_name IN VARCHAR2)
RETURN VARCHAR2 IS
  v_predicate VARCHAR2(4000);
BEGIN
  v_predicate := 'VAITRO = ''Trưởng khoa'' OR
                 (VAITRO = ''Giảng viên'' AND
                  EXISTS (SELECT 1 FROM DONVI D
                          WHERE D.MAĐV = ''Văn phòng khoa'' AND
                                D.TRGĐV = (SELECT MAGV FROM PHANCONG P
                                           WHERE P.MAHP = HOCPHAN.MAHP AND
                                                 P.MAGV = D.TRGĐV)))';
  RETURN v_predicate;
END;
```

- Thêm chính sách RLS cho bảng PHANCONG để áp dụng hàm chính sách:

```
-- Thêm chính sách RLS cho bảng PHANCONG cho vai trò TRUONGKHOA
BEGIN
  DBMS_RLS.ADD_POLICY (
    object_schema   => 'CAMPUSADMIN',  -- Schema của bảng PHANCONG
    object_name     => 'PHANCONG',     -- Tên của bảng
    policy_name     => 'phancong_rls_truongkhoa',
    function_schema => 'TRUONG_KHOA',  -- Schema của hàm chính sách
    policy_function => 'policy_func_rls_truongkhoa',
    statement_types => 'SELECT,INSERT, DELETE, UPDATE', -- Các loại câu lệnh áp dụng chính sách
    update_check => TRUE
  );
END;
/
```

  - Cơ chế sử dụng: VPD

- Chính sách 6:
  - Chủ thể: Người dùng có vai trò là Sinh viên.
  - Quyền:
    - SINHVIEN: Xem thông tin cá nhân, chỉnh sửa địa chỉ và số điện thoại.
    - HOCPHAN: Xem danh sách học phần.
    - KHMO: Xem kế hoạch mở môn.
    - DANGKY: Thêm và xóa dữ liệu đăng ký học phần liên quan đến chính mình, không được chỉnh sửa điểm.
  - Kịch bản:
    - Tạo vai trò Sinh viên và gán các quyền cần thiết.
    - Tạo người dùng SINH_VIEN với mật khẩu là '1'.

- Gán vai trò Sinh viên cho người dùng SINH_VIEN.
- Tạo hàm chính sách RLS cho bảng SINHVIEN và DANGKY để giới hạn truy cập và thao tác dữ liệu:
    - Tạo policy function trả về chuỗi vị từ lọc ra thông tin của chính sinh viên đăng nhập vào và chỉ cho phép chỉnh sửa thông tin của chính mình và khi không có trong khmo

```
CREATE OR REPLACE FUNCTION dangky_policy_func (schema_name IN VARCHAR2, table_name IN VARCHAR2)
RETURN VARCHAR2 IS
  v_predicate VARCHAR2(4000);
BEGIN
  -- Sinh viên chỉ được thêm, xóa dữ liệu đăng ký học phần liên quan đến chính mình và không được chỉnh sửa điểm
  v_predicate := 'MASV = SYS_CONTEXT(''USERENV'', ''SESSION_USER'') AND ' ||
                 'NOT EXISTS (SELECT 1 FROM KHMO WHERE DANGKY.MAHP = KHMO.MAHP
                 AND DANGKY.HK = KHMO.HK AND DANGKY.NAM = KHMO.NAM AND KHMO.MACT = DANGKY.MACT)';
  RETURN v_predicate;
END;
/
```

- Tạo trigger để ngăn chặn việc chỉnh sửa điểm trong bảng DANGKY và kiểm tra quyền hạn khi thêm, xóa dữ liệu:

```
CREATE OR REPLACE TRIGGER trg_dangky_update
BEFORE UPDATE ON DANGKY
FOR EACH ROW
BEGIN
  IF UPDATING('DIEMTH') OR UPDATING('DIEMQT') OR UPDATING('DIEMCK') OR UPDATING('DIEMTK') THEN
    RAISE_APPLICATION_ERROR(-20003, 'Bạn không được phép chỉnh sửa các trường liên quan đến điểm.');
  END IF;
END;
/
```

```
CREATE OR REPLACE TRIGGER trg_dangky_insert_delete
BEFORE INSERT OR DELETE ON DANGKY
FOR EACH ROW
DECLARE
  v_start_date DATE;
BEGIN
  IF :NEW.MASV != SYS_CONTEXT('USERENV', 'SESSION_USER') THEN
    RAISE_APPLICATION_ERROR(-20004, 'Bạn chỉ được phép thêm hoặc xóa dữ liệu đăng ký của chính mình.');
  END IF;

  -- Kiểm tra thời gian hiệu chỉnh đăng ký học phần
  SELECT MIN(KHMO.HK_START_DATE)
  INTO v_start_date
  FROM KHMO
  WHERE KHMO.MAHP = :NEW.MAHP
    AND KHMO.HK = :NEW.HK
    AND KHMO.NAM = :NEW.NAM;

  IF SYSDATE > v_start_date + 14 THEN
    RAISE_APPLICATION_ERROR(-20005, 'Thời gian hiệu chỉnh đăng ký đã hết.');
  END IF;
END;
/
```

- Cơ chế sử dụng: VPD và RBAC

2. *Nhãn an toàn - Oracle Label Security*

Oracle Label Security (OLS) là một tính năng trong hệ thống quản lý cơ sở dữ liệu Oracle Database. Nó cung cấp các công cụ và khả năng để triển khai và quản lý việc bảo mật dữ liệu trên cấp độ nhãn (label-level) trong hệ thống cơ sở dữ liệu.

OLS cho phép bạn xác định và gắn nhãn cho các đối tượng dữ liệu, chẳng hạn như bảng, cột, dòng, hoặc thậm chí từng giá trị riêng lẻ. Nhãn được sử dụng để đại diện cho các cấp độ bảo mật khác nhau, ví dụ như "cực kỳ bảo mật" (top secret), "bảo mật" (secret), "nội bộ" (internal), và "công khai" (public). Bằng cách gắn nhãn cho dữ liệu, bạn có thể áp dụng các chính sách bảo mật nhằm kiểm soát truy cập dựa trên các quyền và nhãn đã được xác định.

Oracle Label Security hỗ trợ tích hợp với các tính năng khác của Oracle Database như quản lý người dùng và vai trò, quyền hạn, và các công nghệ mã hóa dữ liệu khác. Nó cung cấp khả năng thực hiện kiểm tra kiểm soát truy cập để đảm bảo rằng chỉ những người có quyền được phép xem, sửa đổi, hoặc truy cập vào các đối tượng dữ liệu có nhãn tương ứng.

OLS thường được sử dụng trong các môi trường có yêu cầu bảo mật cao như trong ngành chính phủ, lĩnh vực quân sự, hoặc các tổ chức có nhu cầu bảo vệ dữ liệu nhạy cảm.

Các bước gán nhãn cho các dòng dữ liệu:

Các bước tạo OLS

Bước 1: OLS được tạo trên PDB nên cần tạo PDB trước.

```
ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/opt/oracle/product/21c/dbhome_1/oradata';
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST = '/opt/oracle/product/21c/dbhome_1/flash_recovery_area';
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE = 100;
ALTER SYSTEM SET DB_CREATE_ONLINE_LOG_DEST_1 = '/opt/oracle/product/21c/dbhome_1/oradata';


CREATE PLUGGABLE DATABASE CAMPUSPDB ADMIN USER admin IDENTIFIED BY admin ROLES=(DBA);
ALTER PLUGGABLE DATABASE CAMPUSPDB OPEN READ WRITE FORCE;

-- grant privileges
ALTER SESSION SET CONTAINER = CAMPUSPDB;
GRANT CREATE SESSION TO ADMIN CONTAINER=CURRENT;
GRANT SYSDBA TO ADMIN CONTAINER=CURRENT;
```

Bước 2: Kích hoạt OLS

```
EXEC LBACSYS.CONFIGURE_OLS;
EXEC LBACSYS.OLS_ENFORCEMENT.ENABLE_OLS;
```

Tắt khởi động lại và kiểm tra xem pdb có chưa

```
SHUTDOWN IMMEDIATE;
STARTUP;

select * from v$services;
```

SERVICE_ID	NAME	NAME_HASH	NETWORK_NAME	CREATION_DATE	CREATION_DATE_HASH	GOAL	DTP	AQ_HA_NOTIFICATION	CLB_GOAL	COMMIT_OUTCOME	RETENTION_TIME	REPLAY_INITIATION_TIMEOUT	SESSION_STATE_CONSISTENCY	GLOBAL	PDB
18	campuspdb	4014776398	campuspdb	23-JUN-24	3219452558	NONE	N	NO	LONG	NO	0	0 (null)		NO	CAMPUSPDB

Bước 3: Tạo ADMIN OLS và cấp quyền ADMIN, cấp quyền execute cho ADMIN OLS và add ADMIN OLS vào LBAC_DBA

Chuyển qua CAMPUSPDB

```
ALTER SESSION SET CONTAINER= CAMPUSPDB;
SHOW CON_NAME;

CREATE USER ADMIN_OLS IDENTIFIED BY 123 CONTAINER = CURRENT;
GRANT CONNECT,RESOURCE TO ADMIN_OLS; --C?P QUY?N CONNECT VÀ RESOURCE
GRANT UNLIMITED TABLESPACE TO ADMIN_OLS; --C?P QUOTA CHO ADMIN_OLS
GRANT SELECT ANY DICTIONARY TO ADMIN_OLS; --C?P QUY?N ??C DICTIONARY
---> C?P QUY?N EXECUTE CHO ADMIN_OLS
GRANT EXECUTE ON LBACSYS.SA_COMPONENTS TO ADMIN_OLS WITH GRANT OPTION;
GRANT EXECUTE ON LBACSYS.sa_user_admin TO ADMIN_OLS WITH GRANT OPTION;
GRANT EXECUTE ON LBACSYS.sa_label_admin TO ADMIN_OLS WITH GRANT OPTION;
GRANT EXECUTE ON sa_policy_admin TO ADMIN_OLS WITH GRANT OPTION;
GRANT EXECUTE ON char_to_label TO ADMIN_OLS WITH GRANT OPTION;
```

Bước 4: Tạo các user và gán quyền cần thiết trong PDB

```
-- Tao user va gan quyen can thiet trong PDB
CREATE USER TKH001 IDENTIFIED BY TKH001; --Truong khoa
CREATE USER TBM001 IDENTIFIED BY TBM001; --Truong bo mon co 2
CREATE USER TBM002 IDENTIFIED BY TBM002; --Truong bo mon HTTT co so 1
CREATE USER TBM003 IDENTIFIED BY TBM003; --Truong bo mon HTTT co so 2
CREATE USER TBM004 IDENTIFIED BY TBM004; --Truong bo mon KHMT co so 1
CREATE USER TBM005 IDENTIFIED BY TBM005; --Truong bo mon KHMT co so 2
CREATE USER GVU001 IDENTIFIED BY GVU001; --Giao vu co so 2
CREATE USER SVI001 IDENTIFIED BY SVI001; --Sinh vien HTTT Co so 1

CREATE USER NV106 IDENTIFIED BY NV106;    -- Truong khoa
CREATE USER NV100 IDENTIFIED BY NV100;    -- Truong bo mon co 2
CREATE USER NV101 IDENTIFIED BY NV101;    -- Truong bo mon HTTT co so 1
CREATE USER NV102 IDENTIFIED BY NV102;    -- Truong bo mon HTTT co so 2
CREATE USER NV103 IDENTIFIED BY NV103;    -- Truong bo mon KHMT co so 1
CREATE USER NV104 IDENTIFIED BY NV104;    -- Truong bo mon KHMT co so 2
CREATE USER NV091 IDENTIFIED BY NV091;    -- GIAO VU CS2
CREATE USER SV001 IDENTIFIED BY SV001;  -- Sinh vien HTTT co so 1

GRANT CONNECT TO TKH001, TBM001, TBM002, TBM003, TBM004, TBM005, GVU001, SVI001, NV106, NV100, NV101, NV102, NV103, NV104, NV091, SV001;


GRANT INHERIT PRIVILEGES ON USER ADMIN_OLS TO LBACSYS;
GRANT INHERIT PRIVILEGES ON USER SYS TO LBACSYS;
```

Bắt đầu kết nối với ADMIN_OLS

Bước 5: Tạo chính sách OLS (Khởi động lại SQLDEV để cập nhật OLS enable)

```
-- KET NOI VOI ADMIN OLS
BEGIN
 SA_SYSDBA.CREATE_POLICY(
 policy_name => 'region_policy',
 column_name => 'region_label'
);
END;
/

EXEC SA_SYSDBA.ENABLE_POLICY ('region_policy');
```

Bước 6: Tạo COMPONENTS của LABEL (LEVEL, COMPARTMENT, GROUP)

```
--->T?O LEVEL
EXECUTE SA_COMPONENTS.CREATE_LEVEL('region_policy',9000,'TRGK','TRUONG KHOA');
EXECUTE SA_COMPONENTS.CREATE_LEVEL('region_policy',8000,'TRGDV','TRUONG DON VI');
EXECUTE SA_COMPONENTS.CREATE_LEVEL('region_policy',7000,'GVIEN','GIANG VIEN');
EXECUTE SA_COMPONENTS.CREATE_LEVEL('region_policy',6000,'GVU','GIAO VU');
EXECUTE SA_COMPONENTS.CREATE_LEVEL('region_policy',5000,'NV','NHAN VIEN');
EXECUTE SA_COMPONENTS.CREATE_LEVEL('region_policy',4000,'SV','SINH VIEN');

-- T?O COMPARTMENT
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('region_policy',100,'HTTT','HE THONG THONG TIN');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('region_policy',110,'CNPM','CONG NGHE PHAN MEM');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('region_policy',120,'KHMT','KHOA HOC MAY TINH');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('region_policy',130,'CNTT','CONG NGHE THONG TIN');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('region_policy',140,'TGMT','THI GIAC MAY TINH');
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT('region_policy',150,'MMT','MANG MAY TINH');

--->T?O GROUP
EXECUTE SA_COMPONENTS.CREATE_GROUP('region_policy',20,'CS1','CO SO 1');
EXECUTE SA_COMPONENTS.CREATE_GROUP('region_policy',40,'CS2','CO SO 2');
```

Bước 6: Tạo table thông báo và insert dữ liệu.

```
CREATE TABLE THONGBAO(
    MATB NUMBER,
    NOIDUNG VARCHAR2(2000),
    DIADIEM VARCHAR2(100),
    CONSTRAINT PK_TB PRIMARY KEY (MATB)
);
/
INSERT INTO THONGBAO (MATB, NOIDUNG, DIADIEM) VALUES (1, N'Day la thong bao cho truong bo mon httt co so 1', N'Co so 1');
INSERT INTO THONGBAO (MATB, NOIDUNG, DIADIEM) VALUES (2, N'Day la thong bao den truong bo mon httt co so 2 ', N'Co so 2');
INSERT INTO THONGBAO (MATB, NOIDUNG, DIADIEM) VALUES (3, N'Day la thong bao cho giao vu co so 2', N'Co so 2');
INSERT INTO THONGBAO (MATB, NOIDUNG, DIADIEM) VALUES (4, N'Day la thong bao cho toan bo truong don vi', N'');
INSERT INTO THONGBAO (MATB, NOIDUNG, DIADIEM) VALUES (5, N'Day la thong bao cho sinh vien nganh HTTT co so 1', N'Co so 1');
INSERT INTO THONGBAO (MATB, NOIDUNG, DIADIEM) VALUES (6, N'Day la thong bao den cho truong bo mon KHMT co so 1', N'Co so 1');
INSERT INTO THONGBAO (MATB, NOIDUNG, DIADIEM) VALUES (7, N'Day la thong bao den cho truong bo mon KHMT co so 1 va co so 2', N'Co so 1 va Co so 2');
/
```

Bước 7: Tạo Label.

```
-- Truong khoa doc duoc toan bo thong bao
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 1000, 'TRGK');
-- Truong bo mon doc thong bao cho truong bo mon HTTT CS1
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 900, 'TRGDV:HTTT:CS1');
-- Truong bo mon doc thong bao cho truong bo mon HTTT CS2
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 890, 'TRGDV:HTTT:CS2');
-- Truong bo mon doc thong bao cho truong bo mon KHMT CS1
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 880, 'TRGDV:KHMT:CS1');
-- Truong bo mon doc thong bao cho truong bo mon KHMT CS1 va CS2
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 870, 'TRGDV:KHMT:');
-- Truong bo mon doc toan bo thong bao cho truong bo mon
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 860, 'TRGDV');
-- Giao vu doc toan bo thong bao cho giao vu
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 700, 'GVU');
-- Giao vu doc thong bao cho giao vu co so 2
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 710, 'GVU::CS2');
-- Sinh vien CS1
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 600, 'SV::CS1');
-- Sinh vien CS2
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 590, 'SV::CS2');
-- Sinh vien
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('region_policy', 580, 'SV');
/
```

Bước 8: Cập nhật nhãn trong bảng.

```
BEGIN
    SA_POLICY_ADMIN.APPLY_TABLE_POLICY (
    POLICY_NAME => 'REGION_POLICY',
    SCHEMA_NAME => 'ADMIN_OLS',
    TABLE_NAME => 'THONGBAO',
    TABLE_OPTIONS => 'NO_CONTROL'
    );
END;
/
```

Bước 9: Tạo nhãn bằng thủ công.

```
UPDATE THONGBAO
SET region_label = CHAR_TO_LABEL('REGION_POLICY','TRGDV:HTTT:CS2')
WHERE MATB = 2;

UPDATE THONGBAO
SET region_label = CHAR_TO_LABEL('REGION_POLICY','GVU::CS2')
WHERE MATB = 3;

UPDATE THONGBAO
SET region_label = CHAR_TO_LABEL('REGION_POLICY','TRGDV')
WHERE MATB = 4;

UPDATE THONGBAO
SET region_label = CHAR_TO_LABEL('REGION_POLICY','SV:HTTT:CS1')
WHERE MATB = 5;

UPDATE THONGBAO
SET region_label = CHAR_TO_LABEL('REGION_POLICY','TRGDV:KHMT:CS1')
WHERE MATB = 6;

UPDATE THONGBAO
SET region_label = CHAR_TO_LABEL('REGION_POLICY','TRGDV:KHMT')
WHERE MATB = 7;

COMMIT;
/
```

Bước 10: Xóa chính sách củ, áp dụng chính sách mới một cách đầy đủ hơn lên bảng THONGBAO

```
BEGIN
    SA_POLICY_ADMIN.REMOVE_TABLE_POLICY('REGION_POLICY','ADMIN_OLS','THONGBAO');
    SA_POLICY_ADMIN.APPLY_TABLE_POLICY (
    policy_name => 'REGION_POLICY',
    schema_name => 'ADMIN_OLS',
    table_name => 'THONGBAO',
    table_options => 'READ_CONTROL',
    predicate => NULL
);
END;
```

Bước 11: Gán Lable cho USER. Commit dữ liệu.

```
BEGIN
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','TKH001','TRGK:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:CS1,CS2');          -- Truong khoa
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','TBM001','TRGDV:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:CS1,CS2');       -- Truong don vi co so 2
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','TBM002','TRGDV:HTTT:CS1');                                    -- Truong don vi HTTT co so 1
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','TBM003','TRGDV:HTTT:CS2');                                    -- Truong don vi HTTT co so 2
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','TBM004','TRGDV:KHMT:CS1');                                    -- Truong don vi KHMT co so 1
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','TBM005','TRGDV:KHMT:CS2');                                    -- Truong don vi KHMT co so 2
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','GVU001','GVU:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:CS1,CS2');          -- Giao vu co so 2
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','SV1001','SV:HTTT:CS1');                                       -- Sinh vien HTTT Co so 1

    SA_USER_ADMIN.SET_USER_LABELS('region_policy','NV106','TRGK:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:CS1,CS2');          -- Truong khoa doc toan bo thong bao
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','NV100','TRGDV:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:CS1,CS2');         -- Truong don vi doc duoc toan bo thong bao
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','NV101','TRGDV:HTTT:CS1');                                     -- Truong don vi doc duoc thong bao cho HTTT CS1
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','NV102','TRGDV:HTTT:CS2');                                     -- Truong don vi HTTT co so 2
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','NV103','TRGDV:KHMT:CS1');                                     -- Truong don vi KHMT co so 1
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','NV104','TRGDV:KHMT:CS2');                                     -- Truong don vi KHMT co so 2
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','NV091','GVU:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:CS1,CS2');          -- Giao vu doc toan bo thong bao danh cho giao vu
    SA_USER_ADMIN.SET_USER_LABELS('region_policy','SV001','SV:HTTT:CS1');                                        -- Sinh vien HTTT Co so 1
```

a) Hãy gán nhãn cho người dùng là Trưởng khoa có thể đọc được toàn bộ thông báo.

Trưởng khoa sẽ có nhãn là: TRGK:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:CS1,CS2
Nghĩa là người dùng này là trưởng khoa có quyền đọc toàn bộ thông báo.

b) Hãy gán nhãn cho các Trưởng bộ môn phụ trách Cơ sở 2 có thể đọc được toàn bộ thông

báo. dành cho trưởng bộ môn không phân biệt vị trí địa lý.
Trưởng bộ môn cơ sở 2 có nhãn là: TRGDV: HTTT,CNPM,KHMT,CNTT,TGMT,MMT:CS1,CS2
Nghĩa là người dùng này là trưởng đơn vị, khi đọc dữ liệu bảng thông báo, trưởng phòng này có thể đọc được toàn bộ dữ liệu có level <= Trưởng đơn vị của tất cả các ngành không phân biệt địa lý.

c) Hãy gán nhãn cho 01 Giáo vụ có thể đọc toàn bộ thông báo dành cho giáo vụ.

Giáo vụ sẽ có nhãn là: GVU:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:CS1,CS2
Nghĩa là người dùng này có thể đọc toàn bộ thông báo dành cho giáo vụ bất kể đề cập đến lĩnh vực hay vị trí nào.

d) Hãy cho biết nhãn của dòng thông báo t1 để t1 được phát tán (đọc) bởi tất cả Trưởng đơn vị.

Nhãn thông báo: TRGDV
Cách phát tán: Kiểm tra level của người dùng có >= level của dữ liệu t1 hay không → Thỏa thì người dùng này sẽ đọc được.

e) Hãy cho biết nhãn của dòng thông báo t2 để phát tán t2 đến Sinh viên thuộc ngành HTTT học ở Cơ sở 1.

Nhãn thông báo: SV:HTTT:CS1
Cách phát tán:
- Phase 1: chính sách OLS sẽ kiểm tra xem level của người dùng có lớn hơn hoặc bằng level của dữ liệu không. Nếu điều kiện này đúng thì tới Phase 2, nếu sai thì không được truy cập dòng dữ liệu.
- Phase 2: kiểm tra group của nhãn người dùng có chứa bất kì group nào trong nhãn dòng dữ liệu hay không. Nếu điều kiện này đúng thì tới Phase 3, nếu sai thì không được truy cập dòng dữ liệu.
- Phase 3: kiểm tra compartment của nhãn người dùng có chứa tất cả các compartment của nhãn dòng dữ liệu hay không. Nếu điều kiện này đúng thì người dùng sẽ đọc được dữ liệu, nếu sai thì không được truy cập dòng dữ liệu.

f) Hãy cho biết nhãn của dòng thông báo t3 để phát tán t3 đến Trưởng bộ môn KHMT ở cơ sở 1.

Nhãn thông báo: TRGDV:KHMT:CS1
- Phase 1: chính sách OLS sẽ kiểm tra xem level của người dùng có lớn hơn hoặc bằng level của dữ liệu không. Nếu điều kiện này đúng thì tới Phase 2, nếu sai thì không được truy cập dòng dữ liệu.
- Phase 2: kiểm tra group của nhãn người dùng có chứa bất kì group nào trong nhãn dòng dữ liệu hay không. Nếu điều kiện này đúng thì tới Phase 3, nếu sai thì không được truy cập dòng dữ liệu.
- Phase 3: kiểm tra compartment của nhãn người dùng có chứa tất cả các compartment của nhãn dòng dữ liệu hay không. Nếu điều kiện này đúng thì người dùng sẽ đọc được dữ liệu, nếu sai thì không được truy cập dòng dữ liệu.

g) Cho biết nhãn của dòng thông báo t4 để phát tán t4 đến Trưởng bộ môn KHMT ở Cơ sở 1 và Cơ sở 2.

Nhãn thông báo: TRGDV:KHMT
- Phase 1: chính sách OLS sẽ kiểm tra xem level của người dùng có lớn hơn hoặc bằng level của dữ liệu không. Nếu điều kiện này đúng thì tới Phase 2, nếu sai thì không được truy cập dòng dữ liệu.
- Phase 2: kiểm tra group của nhãn người dùng có chứa bất kì group nào trong nhãn dòng dữ liệu hay không. Nếu điều kiện này đúng thì tới Phase 3, nếu sai thì không được truy cập dòng dữ liệu.

h) Hãy cho biết nhãn của trưởng đơn vị HTTT cơ sở 1.

Trưởng đơn vị có nhãn là: TRGDV:HTTT:CS1
Nghĩa là người dùng này là trưởng đơn vị, khi đọc dữ liệu bảng thông báo, trưởng phòng này có thể đọc được toàn bộ dữ liệu có level <= Trưởng đơn vị của ngành HTTT ở cơ sở 1.

i) Cho biết nhãn của dòng thông báo t5 để phát tán t5 đến Trưởng bộ môn HTTT ở Cơ sở 1.

Nhãn của thông báo: TRGDV:HTTT:CS1

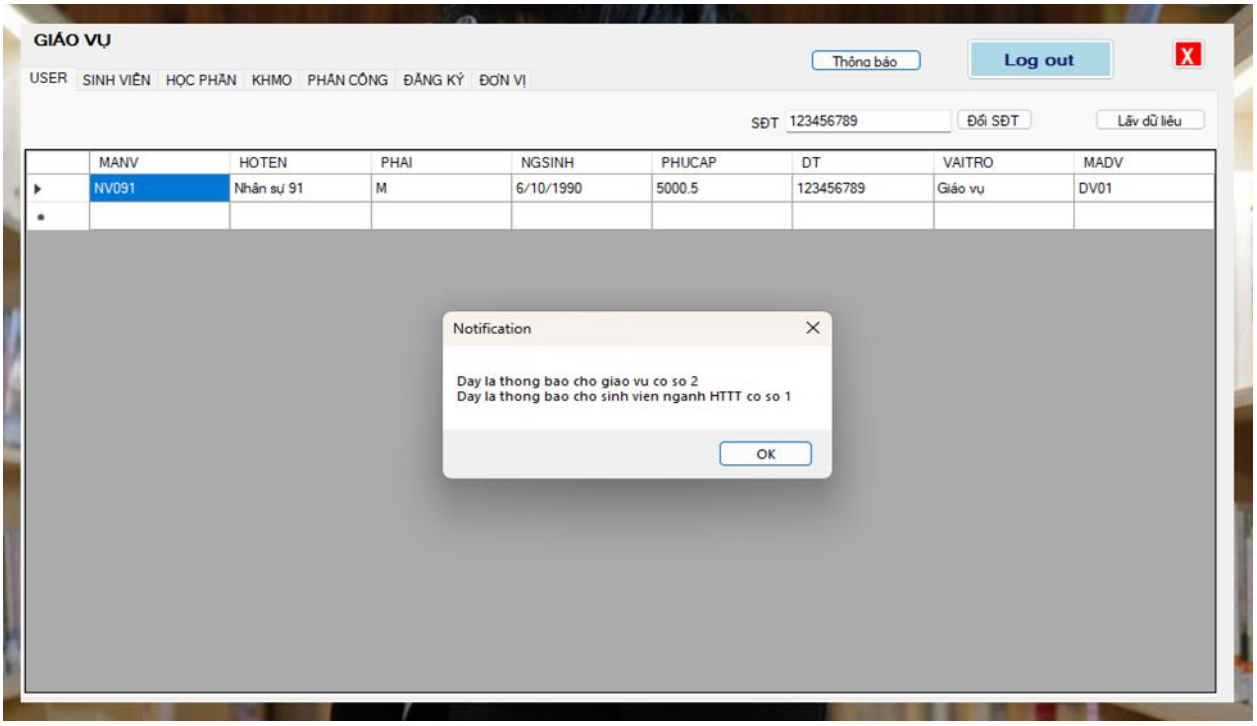
j) Cho biết nhãn của dòng thông báo t5 để phát tán t5 đến Trưởng bộ môn HTTT ở Cơ sở 2.

Nhãn của thông báo: TRGDV:HTTT:CS2

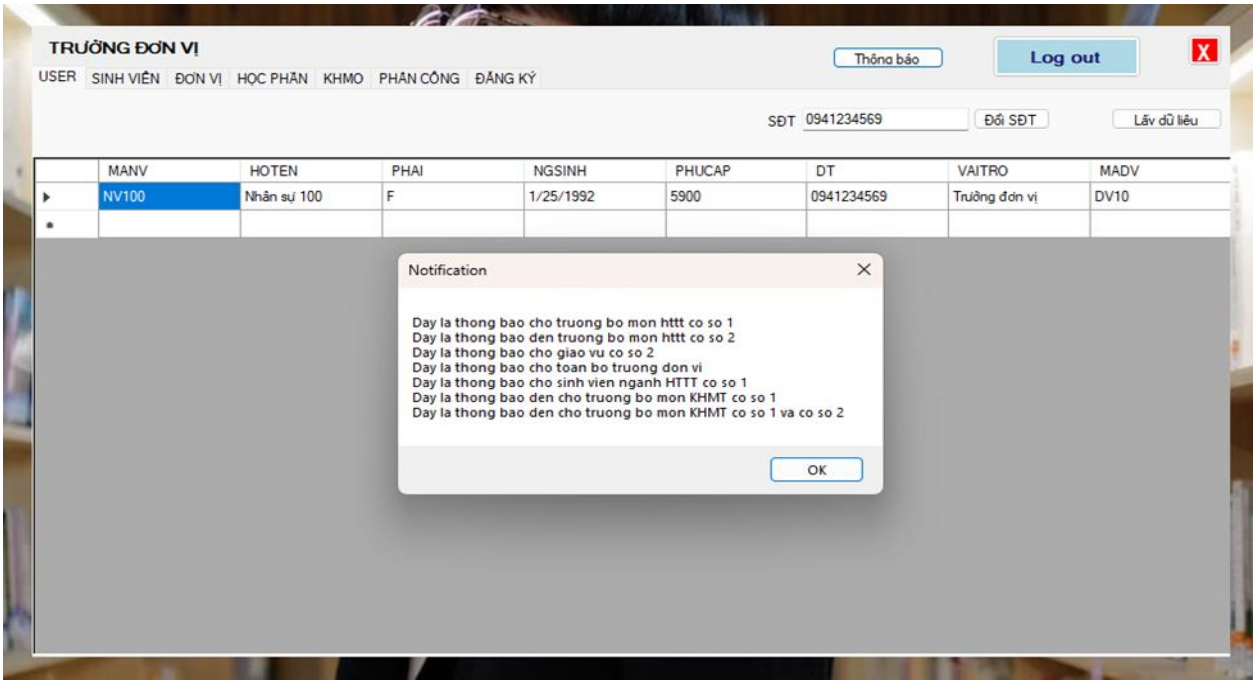
Test với từng người dùng:

Trước khi test đảm bảo pdb pluggable được mở.

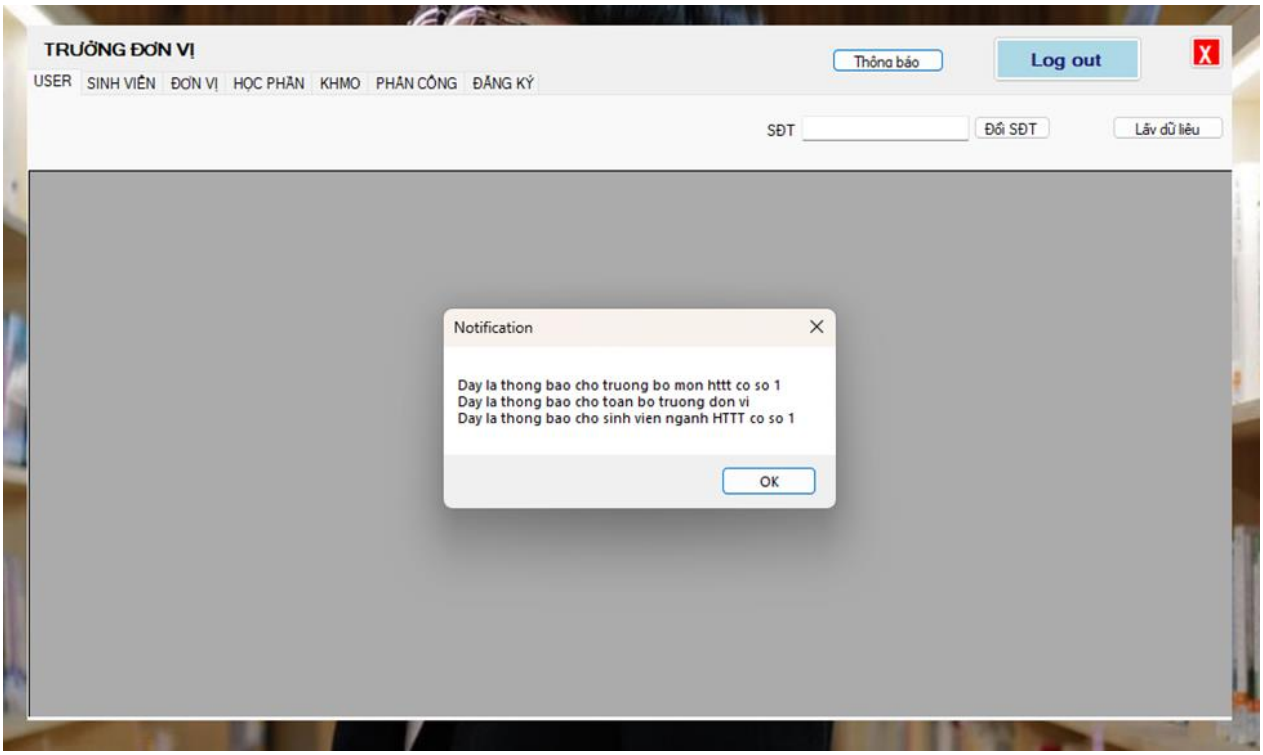
GIÁO VỤ:



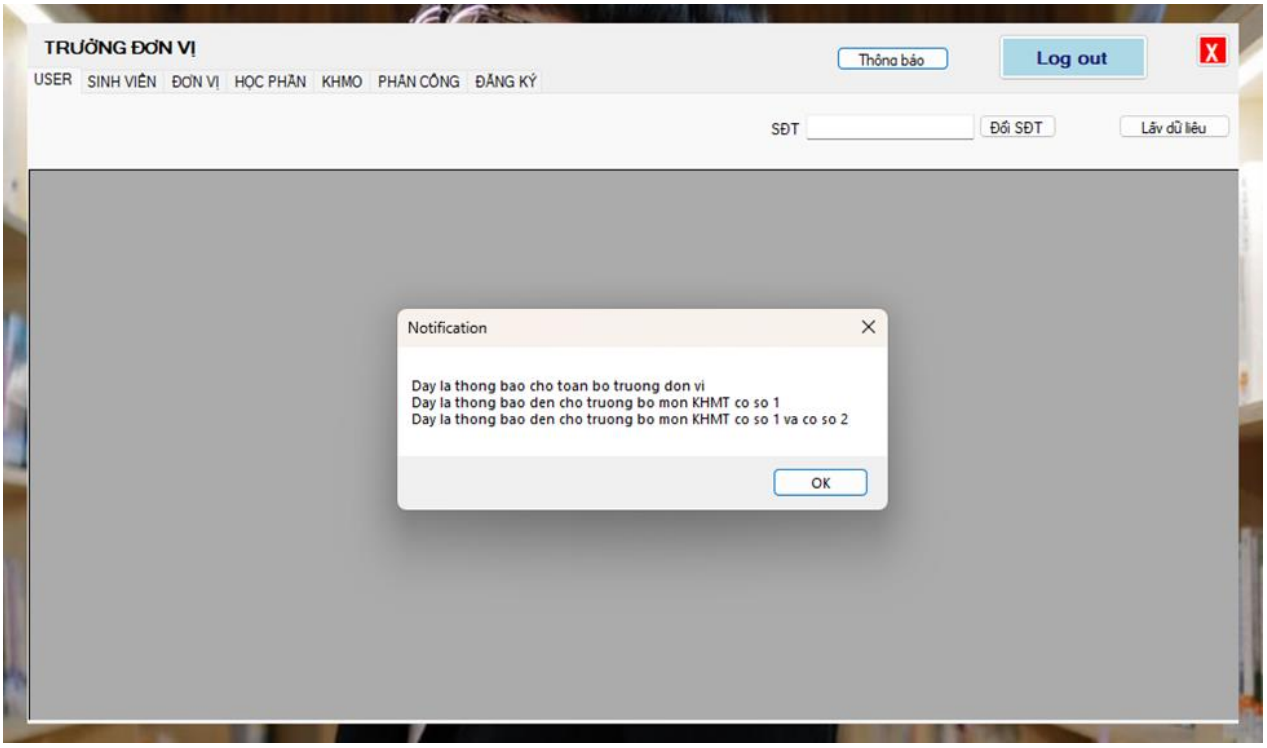
01 TRƯỞNG ĐƠN VỊ (TRƯỞNG KHOA) xem tất cả thông báo.



TRƯỞNG ĐƠN VỊ HTTT cơ sở 1:



TRƯỞNG ĐƠN VỊ KHMT cơ sở 1:



3. *Ghi vết hệ thống – Audit*

Auditing là hoạt động theo dõi và lưu vết lại các hoạt động thao tác của người dùng vào dữ liệu. Trong Oracle, người quản trị có thể cấu hình để thực hiện audit lại các hoạt động trong của cả người dùng trong cơ sở dữ liệu lẫn những người dùng không có trong cơ sở dữ liệu, giới hạn audit với một số lệnh cụ thể hay audit một số role cụ thể trong dữ liệu.

Các bước cài đặt Audit:

1. *Kích hoạt audit*

```
-- Kích ho?t vi?c ghi nh?t ký h? th?ng
ALTER SYSTEM SET audit_trail = db, EXTENDED SCOPE = SPFILE;

-- Kh?i ??ng l?i c? s? d? li?u ?? thay ??i có hi?u l?c
SHUTDOWN IMMEDIATE;
STARTUP;
```

Việc kích hoạt tính năng ghi nhật ký hệ thống với `audit_trail = db, EXTENDED` và `SCOPE = SPFILE` yêu cầu cơ sở dữ liệu phải được khởi động lại để các thay đổi có hiệu lực. Lệnh `SHUTDOWN IMMEDIATE` được sử dụng để tắt cơ sở dữ liệu một cách an toàn, và lệnh `STARTUP` để khởi động lại cơ sở dữ liệu với cấu hình mới đã được cập nhật.

2. Ghi vết tiêu chuẩn

```
-- Theo doi hanh vi cua tat ca user tren cac bang
AUDIT ALL ON CAMPUSADMIN.DANGKY BY ACCESS;
-- THEO DOI HANH VI CUA TAT CA USER TREN CAC VIEW
AUDIT ALL ON CAMPUSADMIN.UV_DANGKYBANTHAN BY ACCESS;
AUDIT ALL ON CAMPUSADMIN.UV_INFOR BY ACCESS;
AUDIT ALL ON CAMPUSADMIN.UV_PHANCONG_GV BY ACCESS;
AUDIT ALL ON CAMPUSADMIN.UV_PHANCONG_OF_DONVI BY ACCESS;
AUDIT ALL ON CAMPUSADMIN.UV_PHANCONGBANTHAN BY ACCESS;
AUDIT ALL ON CAMPUSADMIN.UV_SELDANGKY BY ACCESS;
AUDIT ALL ON CAMPUSADMIN.UV_SELDANGKY_4TEST BY ACCESS;
AUDIT ALL ON CAMPUSADMIN.UV_SELPHANCONG BY ACCESS;
AUDIT ALL ON CAMPUSADMIN.UV_SELPHANCONG_TRUONGDONVI BY ACCESS;
```

```
-- THEO DOI HANH VI THANH CONG
AUDIT ALL WHENEVER SUCCESSFUL;
-- THEO DOI HANH VI KHONG THANH CONG
AUDIT ALL WHENEVER NOT SUCCESSFUL;
```

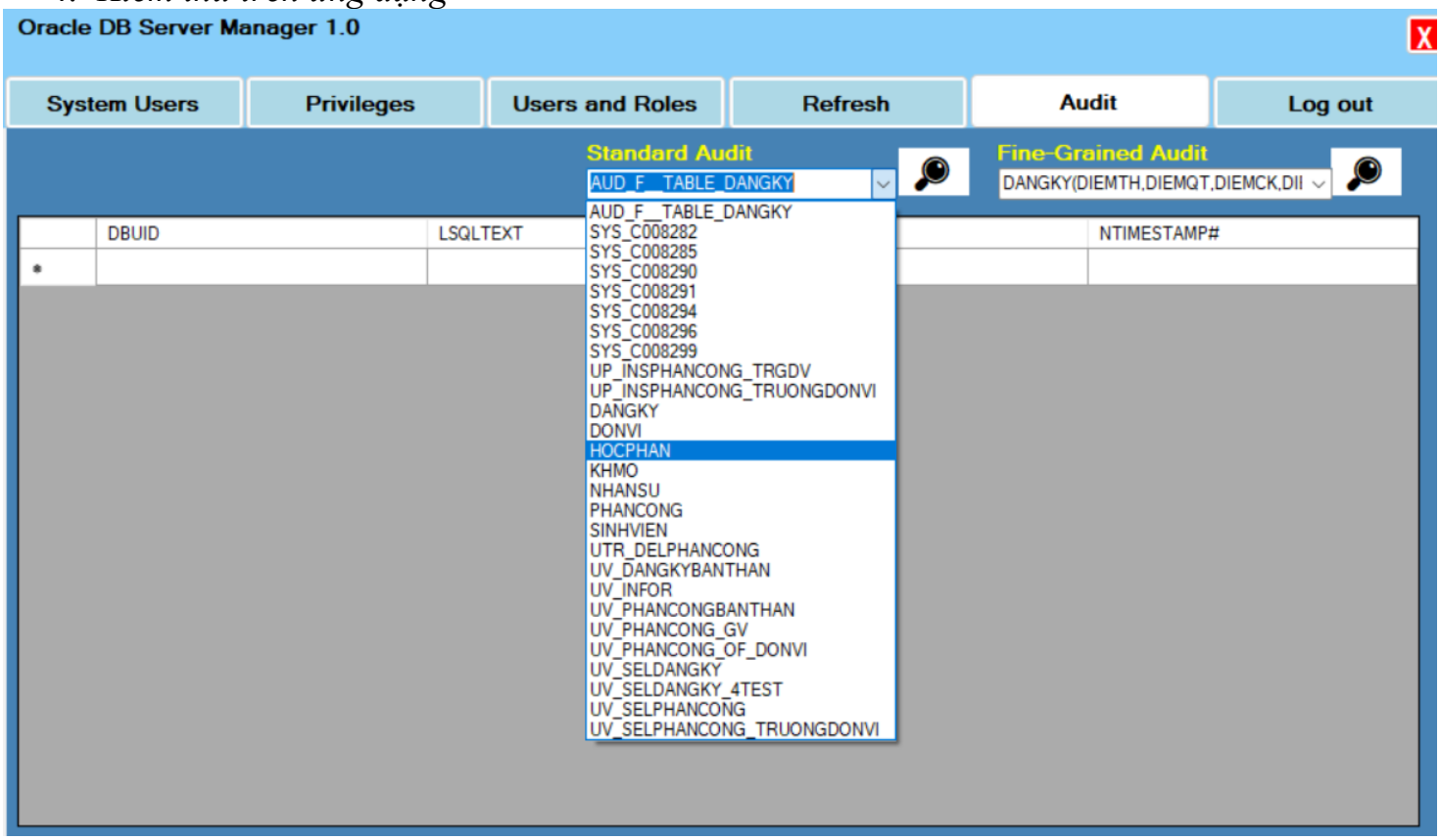
3. *Ghi vết chi tiết*

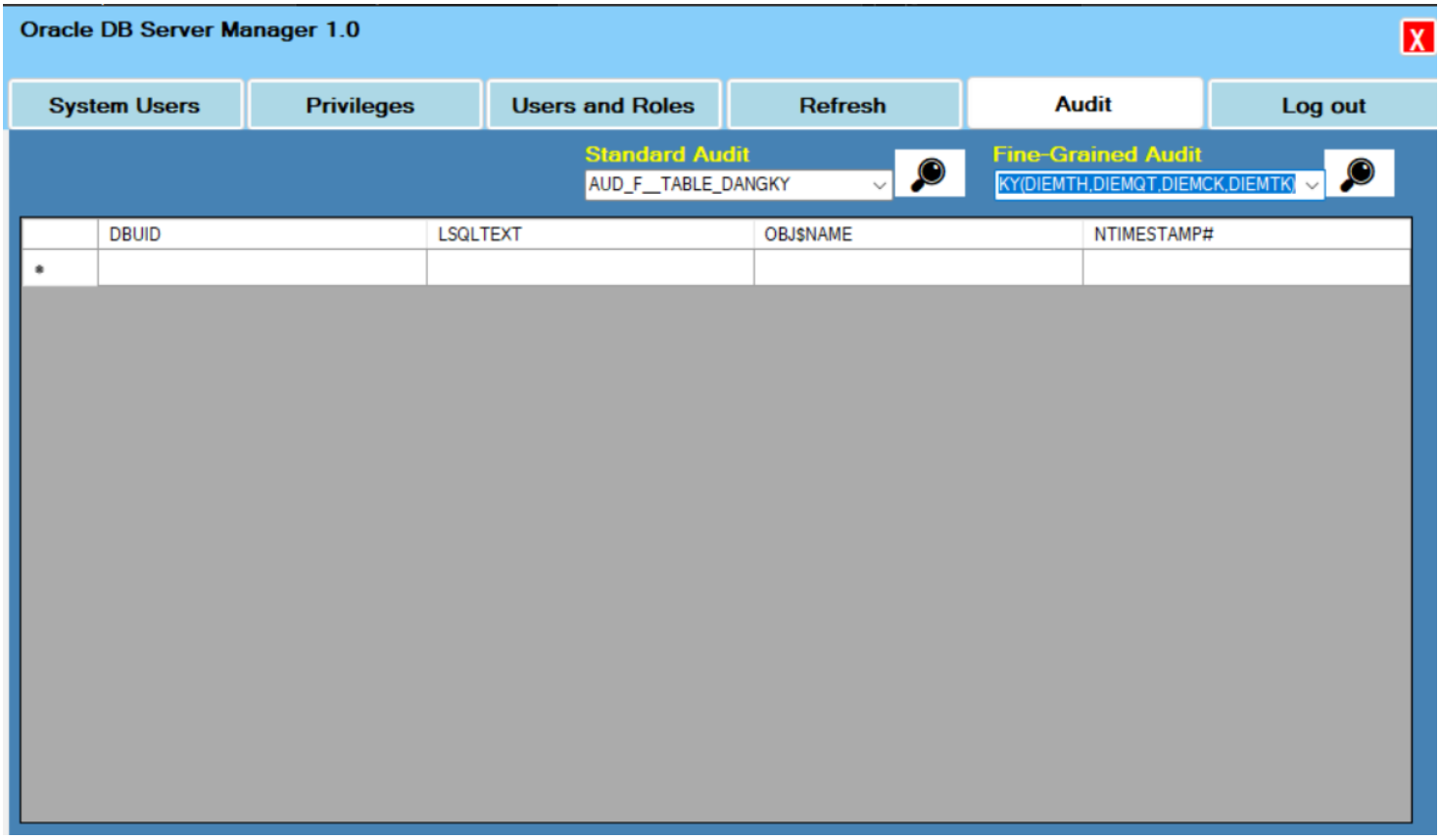
```
-- Cap nhat diem ma nguoi dung khong phai la gv
CREATE OR REPLACE FUNCTION AUD_F__TABLE_DANGKY(pTxtUser IN VARCHAR2)
RETURN NUMBER
AS
  USERROLE VARCHAR2(20);
BEGIN
  -- L?y vai trò c?a ng??i dùng t? b?ng NHANSU
  SELECT VAITRO INTO USERROLE FROM CAMPUSADMIN.NHANSU WHERE MANV = pTxtUser;

  -- Ki?m tra n?u vai trò là 'GIANGVIEN'
  IF USERROLE = 'GIANGVIEN' THEN
    RETURN 1;
  ELSE
    RETURN 0;
  END IF;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    -- Tr??ng h?p không tìm th?y ng??i dùng trong b?ng NHANSU
    RETURN 0;
END AUD_F__TABLE_DANGKY;
/
```

```
BEGIN
  DBMS_FGA.add_policy(
    object_schema   => 'CAMPUSADMIN',
    object_name     => 'DANGKY',
    policy_name     => 'audit_diemso_capnhat',
    audit_condition => 'CAMPUSADMIN.AUD_F__TABLE_DANGKY(SYS_CONTEXT(''USERENV'', ''SESSION_USER'')) = 0',
    audit_column    => 'DIEMTH, DIEMQT, DIEMCK, DIEMTK',
    statement_types => 'UPDATE',
    handler_schema  => NULL,
    handler_module  => NULL,
    enable          => TRUE,
    audit_trail     => dbms_fga.db + dbms_fga.extended);
END;
/
```

```
BEGIN
  DBMS_FGA.add_policy(
    object_schema   => 'CAMPUSADMIN',
    object_name     => 'NHANSU',
    policy_name     => 'audit_phucap_read',
    audit_condition => 'CAMPUSADMIN.AUD_F__TABLE_DANGKY(SYS_CONTEXT(''USERENV'', ''SESSION_USER'')) = 0',
    audit_column    => 'PHUCAP',
    statement_types => 'SELECT',
    handler_schema  => NULL,
    handler_module  => NULL,
    enable          => TRUE,
    audit_trail     => dbms_fga.db + dbms_fga.extended);
END;
/
```

4. *Kiểm thử trên ứng dụng*





4. *Sao lưu và phục hồi dữ liệu*

- Tìm hiểu các phương pháp sao lưu trong oralce:
    - Sao lưu toàn bộ cơ sở dữ liệu (Full Database Backup): Đây là phương pháp sao lưu toàn bộ dữ liệu, cấu trúc, và các thành phần khác của cơ sở dữ liệu.
    - Sao lưu theo gia tăng (Incremental Backup): Chỉ sao lưu các thay đổi kể từ lần sao lưu cuối cùng.
    - Sao lưu log (Archive Log Backup): Sao lưu các tập tin nhật ký để đảm bảo khả năng phục hồi các thay đổi từ lần sao lưu cuối cùng.
    - Sao lưu theo thời gian thực (Hot Backup): Sao lưu cơ sở dữ liệu khi nó vẫn đang hoạt động.
    - Sao lưu lạnh (Cold Backup): Sao lưu cơ sở dữ liệu khi nó đã được tắt.
- Tìm hiểu về các phương pháp phục hồi dữ liệu:
    - Phục hồi đầy đủ (Full Recovery): Sử dụng bản sao lưu đầy đủ để khôi phục toàn bộ cơ sở dữ liệu.
    - Phục hồi theo gia tăng (Incremental Recovery): Phục hồi dữ liệu từ bản sao lưu gia tăng.

- Phục hồi điểm thời gian (Point-in-Time Recovery): Phục hồi cơ sở dữ liệu đến một điểm thời gian cụ thể.

- Các bước thực hiện sao lưu trên oracle:
  - Đầu tiên bật chế độ nhật ký (Archive Log Mode):

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;
```

  - Khởi động RMAN:

```
rman target /
```

  - Cấu hình vị trí sao lưu:

```
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
CONFIGURE DEVICE TYPE DISK PARALLELISM 2 BACKUP TYPE TO BA
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/backup/%U';
```

  - Thực hiện sao lưu toàn bộ cơ sở dữ liệu:

```
BACKUP DATABASE PLUS ARCHIVELOG;
--Sao lưu gia tăng:
BACKUP INCREMENTAL LEVEL 1 DATABASE;
```

  - Sao lưu gia tăng:

```
BACKUP INCREMENTAL LEVEL 1 DATABASE;
```

  - Sao lưu bằng data pump:

```
--Sao lưu bằng Data Pump:
expdp sys/123456@orcl schemas=your_schema directory=dpump_dir dumpfile=full_%U.dmp log
```

  - Thực hiện phục hồi toàn bộ cơ sở dữ liệu:

```
--Phục hồi toàn bộ cơ sở dữ liệu:
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
RESTORE DATABASE;
RECOVER DATABASE;
ALTER DATABASE OPEN;
```

- Phục hồi dữ liệu theo thời gian:

```
RUN {
  SET UNTIL TIME 'YYYY-MM-DD HH24:MI:SS';
  RESTORE DATABASE;
  RECOVER DATABASE;
  ALTER DATABASE OPEN RESETLOGS;
}
```

- Phục hồi bằng data pump:

```
--Phục hồi bằng Data Pump:
impdp system/password@orcl schemas=your_schema directory=dpump_dir dumpfile=full_%U.dmp log
```

- Đánh giá ưu nhược điểm của từng phương pháp:
  - Sao lưu toàn bộ cơ sở dữ liệu (Full Backup):
    - Ưu điểm: Đơn giản, dễ thực hiện và phục hồi nhanh chóng toàn bộ dữ liệu.
    - Nhược điểm: Tốn nhiều thời gian và không gian lưu trữ.
  - Sao lưu gia tăng (Incremental Backup):
    - Ưu điểm: Tiết kiệm không gian lưu trữ và thời gian sao lưu.
    - Nhược điểm: Quá trình phục hồi phức tạp hơn và cần nhiều bản sao lưu để phục hồi đầy đủ.
  - RMAN (Recovery Manager):
    - Ưu điểm: Cung cấp đầy đủ các chức năng sao lưu và phục hồi, tự động hóa nhiều công việc.
    - Nhược điểm: Đòi hỏi người dùng phải có kiến thức sâu rộng và cấu hình phức tạp.
  - Data Pump: Oracle Data Pump là một công cụ mạnh mẽ được Oracle cung cấp để thực hiện các hoạt động xuất (export) và nhập (import) dữ liệu. Nó thay thế công cụ xuất/nhập dữ liệu cũ là EXP và IMP. Data Pump cung cấp hiệu suất

cao hơn và linh hoạt hơn so với công cụ cũ, cho phép bạn thực hiện sao lưu và phục hồi các đối tượng cơ sở dữ liệu một cách nhanh chóng và hiệu quả.

  - Ưu điểm: Sao lưu logic, tiện lợi cho việc di chuyển dữ liệu giữa các môi trường khác nhau.
  - Nhược điểm: Không phù hợp cho việc sao lưu và phục hồi toàn bộ cơ sở dữ liệu.

- Kết luận:
  Dựa trên đánh giá, RMAN là công cụ mạnh mẽ và linh hoạt nhất cho việc sao lưu và phục hồi cơ sở dữ liệu Oracle. Mặc dù có sự phức tạp trong việc sử dụng và cấu hình, RMAN đáp ứng đầy đủ các yêu cầu bảo vệ dữ liệu trong các hệ thống doanh nghiệp.

III. Tài liệu tham khảo

- Các slide lý thuyết + thực hành môn ATBM trong HTTT của trường ĐH KHTN.
- Understanding Oracle Label Security - https://youtu.be/o4-XpUQWfaM.
- DBMS_CRYPTO (oracle.com) - https://docs.oracle.com/en/database/oracle/oracle-database/21/arpls/DBMS_CRYPTO.html#GUID-4B200807-A740-4A2E-8828-AC0CFF6127D5
- TRẦN VĂN BÌNH MASTER: Các câu lệnh hay dùng với Oracle Auditing (tranvanbinh.vn) - https://www.tranvanbinh.vn/2022/03/cac-cau-lenh-hay-dung-voi-oracle.html