

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN - BỘ MÔN HỆ THỐNG THÔNG TIN



BÁO CÁO ĐỒ ÁN THỰC HÀNH

GIAI ĐOẠN 2 - TỔNG KẾT

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

Lớp: 22HTTT2

Nhóm: ATBM-07

GIÁO VIÊN HƯỚNG DẪN

Cô Phạm Thị Bạch Huệ

Cô Tiết Gia Hồng

Thầy Lương Vĩ Minh

TP.HCM, tháng 4 năm 2025

MỤC LỤC

1. THÔNG TIN CHUNG.....	3
1.1. Thông tin nhóm.....	3
1.2. Bảng phân công và đánh giá công việc.....	3
2. NỘI DUNG LÝ THUYẾT CẦN THIẾT CHO ĐỒ ÁN.....	5
2.1. Phân hệ 1.....	5
2.1.1. Người dùng - User.....	5
2.1.2. Vai trò - Role.....	5
2.1.3. Khung nhìn - View.....	5
2.1.4. Quyền người dùng - Privilege.....	6
2.2. Phân hệ 2.....	8
2.2.1. Cơ chế RBAC.....	8
2.2.2. Cơ chế VPD.....	8
2.2.3. Chính sách nhãn Oracle - OLS.....	9
2.2.4. Audit.....	10
2.2.5. Sao lưu và phục hồi dữ liệu trong Oracle.....	13
3. KẾT QUẢ THỰC HIỆN.....	15
3.1. Phân hệ 1.....	15
3.1.1. Giao diện đăng nhập.....	15
3.1.2. Xem danh sách user trong hệ thống.....	16
3.1.3. Xem danh sách role trong hệ thống.....	16
3.1.4. Xem quyền của user trong hệ thống.....	16
3.1.5. Xem quyền của role trong hệ thống.....	17
3.1.6. Xem danh sách role trong hệ thống.....	17
3.1.7. Tạo user mới.....	18
3.1.8. Xóa user.....	18
3.1.9. Thu hồi quyền của role.....	20
3.1.10. Tạo role mới.....	21
3.1.11. Xóa role.....	21
3.1.12. Cấp quyền cho user.....	22
3.1.13. Cấp quyền cho role.....	24
3.1.14. Cấp role cho user.....	25
3.1.15. Thu hồi quyền của user.....	26
3.1.16. Thu hồi role của user.....	27
3.2. Phân hệ 2.....	28
3.2.1. Lược đồ cơ sở dữ liệu.....	28
3.2.2. Đặc tả cơ sở dữ liệu.....	28
3.2.3. Các chính sách bảo mật.....	31
3.2.3.1. Yêu cầu 1 - Câu 1: Ứng dụng RBAC.....	31
3.2.3.2. Yêu cầu 1 - Câu 2: Ứng dụng RBAC.....	32
3.2.3.3. Yêu cầu 1 - Câu 3: Ứng dụng VPD.....	34
3.2.3.4. Yêu cầu 1 - Câu 4: Ứng dụng VPD.....	35
3.2.3.5. Yêu cầu 2 - Cơ chế phát tán thông báo dùng OLS.....	36
3.2.3.6. Yêu cầu 3 - Ghi nhật ký hệ thống bằng Audit.....	39
4. TÀI LIỆU THAM KHẢO.....	44

1. THÔNG TIN CHUNG

1.1. Thông tin nhóm

Mã nhóm	MSSV	Họ và tên	Email
ATBM-07	22127174	Ngô Văn Khải	22127174@student.hcmus.edu.vn
	22127205	Bùi Lê Khôi	22127205@student.hcmus.edu.vn
	22127312	Nguyễn Thị Yên Nhi	22127312@student.hcmus.edu.vn
	22127319	Lê Trần Kim Oanh	22127319@student.hcmus.edu.vn

1.2. Bảng phân công và đánh giá công việc

Phân hệ 1			
Công việc thực hiện	Người thực hiện	Mức độ hoàn thành	Đánh giá của nhóm
• Quay video demo.	22127174 - Ngô Văn Khải	100%	10/10
• Tạo tài khoản quản truser_admin và cấp quyền trong Oracle • Quay video demo.	22127205 - Bùi Lê Khôi	100%	10/10
• Quay video demo.	22127312 - Nguyễn Thị Yên Nhi	100%	10/10
• Kết nối database với winform • Thiết kế winform cho phép tạo mới, xóa, sửa (hiệu chỉnh) user hoặc role. • Thiết kế winform xem danh sách tài khoản người dùng và role trong hệ thống Oracle DB Server. • Thiết kế winform cho phép thực hiện việc cấp quyền. • Thiết kế winform thực hiện cấp quyền trên một số loại đối tượng của CSDL như: table, view, stored procedure, function. • Thiết kế winform cho phép thu hồi quyền từ user hoặc role. • Thiết kế winform xem thông tin về quyền của mỗi user hoặc role trên các đối tượng dữ liệu. • Quay video demo.	22127319 - Lê Trần Kim Oanh	100%	10/10
Toàn phân hệ		100%	10/10

Phân hệ 2			
Công việc thực hiện	Người thực hiện	Mức độ hoàn thành	Đánh giá của nhóm
<ul style="list-style-type: none"> Thực hiện Yêu cầu 1 - Câu 2 - RBAC trong Oracle Thực hiện Yêu cầu 1 - Câu 4 - VPD trong Oracle Tạo giao diện Winform cho Sinh viên (SV) Tạo giao diện Winform cho Trường Đơn vị (TRGDV) Demo giao diện Winform Demo trên Oracle Quay video demo Viết báo cáo chính sách RBAC Viết báo cáo chính sách VPD Viết Guideline 	22127174 - Ngô Văn Khải	100%	10/10
<ul style="list-style-type: none"> Viết script tạo dữ liệu và user. Thực hiện Yêu cầu 1 - Câu 1 - RBAC trong Oracle Thực hiện Yêu cầu 1 - Câu 3 - VPD trong Oracle Tạo giao diện Winform cho Nhân viên cơ bản (NVCB) Tạo giao diện Winform cho Nhân viên tổ chức hành chính (NVTCHC) Demo giao diện Winform Demo trên Oracle Quay video demo Viết báo cáo chính sách RBAC Viết báo cáo chính sách VPD Viết báo cáo chính sách OLS Tổng hợp và chỉnh sửa báo cáo 	22127205 - Bùi Lê Khôi	100%	10/10
<ul style="list-style-type: none"> Thực hiện Yêu cầu 2 - OLS trong Oracle Demo giao diện Winform cho Giảng viên (GV) Demo trên Oracle Quay video demo Viết báo cáo chính sách OLS 	22127312 - Nguyễn Thị Yên Nhi	100%	10/10
<ul style="list-style-type: none"> Demo giao diện Winform Demo trên Oracle Quay video demo Tạo Winform cho Nhân viên phòng khảo thí (NVPTK) và Nhân viên phòng công tác sinh viên (NVCTSV) 	22127319 - Lê Trần Kim Oanh	100%	10/10

Toàn phân hệ	90%	9/10
--------------	-----	------

Các nội dung chưa hoàn thành:

- Một số màn hình giao diện cho role NVPDT, ...
- Phần 4 - Sao lưu và phục hồi dữ liệu.

2. NỘI DUNG LÝ THUYẾT CẦN THIẾT CHO ĐỒ ÁN

2.1. Phân hệ 1

2.1.1. Người dùng - User

Người dùng (User) là một tài khoản, có thể có quyền quản lý một schema. Mỗi người dùng được định danh bằng username duy nhất và xác thực bằng mật khẩu (password). Người dùng được cấp quyền (privileges) bởi DBA hoặc hệ thống.

Ví dụ: Tạo user user1 với mật khẩu password1, chỉ định tablespace mặc định để lưu trữ các đối tượng của user.

CREATE USER user1 IDENTIFIED BY password1

DEFAULT TABLESPACE users

TEMPORARY TABLESPACE temp;

Chỉ có người dùng có quyền DROP USER hoặc quyền của DBA mới có thể xóa tài khoản người dùng. Ví dụ:

DROP USER user1 CASCADE;

2.1.2. Vai trò - Role

Vai trò (Role) được tạo bởi người dùng (thường là quản trị viên), được sử dụng để nhóm các quyền hoặc các vai trò khác lại với nhau, cho phép quản trị viên cấp hoặc thu hồi quyền từ người dùng(user) một cách hiệu quả. Quản trị viên có thể tạo một role chứa các quyền cần thiết và gán role đó cho người dùng (user) hoặc các role khác thay vì cấp từng quyền riêng lẻ.

Có hai loại Role:

- Common Role: Role này được dùng chung cho tất cả các container (Root và các PDB)
- Local Role: Role này được dùng cho các PDB cụ thể. Tên role có thể giống nhau ở các PDB khác nhau.
- Lệnh tạo Role:
CREATE ROLE role_name;
- Gán quyền cho Role:
GRANT privilege_name TO role_name;
- Gán Role cho người dùng:
GRANT role_name TO user_name;
- Thu hồi Role:
REVOKE SELECT, INSERT, UPDATE ON table_name FROM role_name;
- Có thể kiểm tra Role bằng các lệnh : **DBA_ROLE_PRIVS,**
USER_ROLE_PRIVS, DBA_TAB_PRIVS,...

2.1.3. Khung nhìn - View

View (khung nhìn) là một công cụ trong hệ quản trị CSDL quan hệ (RDBMS), giúp giới hạn dữ liệu mà người dùng được phép truy cập.

Nó cho phép xác định quyền truy cập dựa trên điều kiện cụ thể (predicates) – chỉ những dòng dữ liệu thỏa điều kiện mới được hiển thị cho người dùng.

View là một bảng ảo được định nghĩa bằng một truy vấn SQL trên một hoặc nhiều bảng cơ sở. View không lưu trữ dữ liệu thực tế; mỗi khi truy vấn một view, Oracle sẽ thực thi truy vấn gốc để lấy dữ liệu mới nhất từ các bảng liên quan.

Cách tiếp cận kiểm soát truy cập dựa trên nội dung:

1. Tạo một View chứa các điều kiện cần thiết để lọc dữ liệu cho người dùng cụ thể.
2. Cấp quyền SELECT cho người dùng đó trên View, không cấp trực tiếp trên bảng gốc.

Ví dụ: Chỉ cho người dùng TEST122 xem thông tin người dùng có số dư (BAL) > 20000:

```
CREATE VIEW BALMORETHAN2000 AS
SELECT * FROM ACCMASTER.ACCTS_22127319
WHERE BAL > 20000;
```

```
GRANT SELECT ON BALMORETHAN2000 TO TEST122;
```

Xem thông tin VIEW đã tạo:

```
SELECT owner, object_name, object_type
FROM ALL_OBJECTS
WHERE object_name = 'BALMORETHAN2000';
```

OWNER	OBJECT_NAME	OBJECT_TYPE
ACCMaster	BALMORETHAN2000	VIEW

Bảng đầy đủ:

ACCNO	ACCNAME	BAL
1	1 Alex	10000
2	2 Bill	15000
3	3 Charlie	20000
4	4 David	25000

TEST122 chạy code xem thông tin bảng:

```
SELECT * FROM ACCMASTER.BALMORETHAN2000;
```

ACCNO	ACCNAME	BAL
1	4 David	25000

2.1.4. Quyền người dùng - Privilege

Quyền người dùng là việc cho phép người dùng được phép thực thi một câu lệnh SQL cụ thể hoặc cho phép người dùng quyền truy cập vào đối tượng (object) của người dùng (user) khác hay schema khác.

Quyền người dùng được cấp hoặc thu hồi bởi người quản trị instance, user có quyền admin, hoặc chủ sở hữu đối tượng (object) với một đối tượng cụ thể.

Quyền người dùng được phân ra làm 2 loại:

- Quyền hệ thống (System privilege):

- Là quyền thực hiện hành động trên đối tượng cụ thể (như table, view, index, sequence, function, ...) của bất kì schema nào. Chỉ người quản trị instance hoặc user có quyền admin mới có thể cấp hoặc thu hồi quyền hệ thống, do đó chỉ nên cấp quyền này cho những user đáng tin cậy.
- Có thể truy vấn trong Oracle danh sách đầy đủ bằng câu lệnh:
SELECT * FROM DBA_SYS_PRIVS;
- Cú pháp của lệnh cấp quyền người dùng:
**GRANT {system_priv|role}
[, {system_priv|role}]...
TO {user|role|PUBLIC}
[, {user|role|PUBLIC}]...
[WITH ADMIN OPTION];**
- Cú pháp của lệnh thu hồi quyền người dùng:
**REVOKE {system_priv|role}
[, {system_priv|role}]...
FROM {user|role|PUBLIC}
[, {user|role|PUBLIC}]...;**

Ví dụ:

**GRANT CREATE SESSION, CREATE TABLE TO Janes;
REVOKE CREATE TABLE FROM Janes;**

- Quyền đối tượng (Object privilege):
 - Là quyền cho phép user cụ thể có thể thực hiện hành động hoặc truy cập vào các đối tượng của một user khác (schema khác). Bản thân chủ sở hữu đối tượng có tất cả các quyền trên đối tượng đó và các quyền này không thể bị thu hồi.
 - Một số quyền đối tượng được trình bày trong ảnh dưới:
 - Cú pháp của lệnh cấp quyền đối tượng:
**GRANT { object_priv [(column_list)]
[, object_priv [(column_list)]]...
|ALL [PRIVILEGES]}
ON [schema.]object
TO {user|role|PUBLIC}
[, {user|role|PUBLIC}]...
[WITH GRANT OPTION];**
 - Cú pháp của lệnh thu hồi quyền đối tượng:
**REVOKE { object_priv [(column_list)]
[, object_priv [(column_list)]]...
|ALL [PRIVILEGES] }
ON [schema.]object
FROM {user|role|PUBLIC}
[, {user|role|PUBLIC}]...
[CASCADE CONSTRAINTS];**

Ví dụ:

**GRANT SELECT,INSERT,UPDATE ON EMPLOYEES TO Janes;
REVOKE UPDATE (birthday) ON EMPLOYEES FROM Janes;**

2.2. Phân hệ 2

2.2.1. Cơ chế RBAC

RBAC (Role-Based Access Control) được sử dụng nhằm phân quyền theo vai trò thay vì theo từng user như cơ chế DAC.

Trong RBAC có một khái niệm quan trọng là Role - Vai trò. Role được hiểu là một tập hợp những quyền hạn được cấp cho người dùng. Khi gán một Role cho một người dùng thì họ sẽ được cấp tất cả các quyền mà role đó có.

Ví dụ, xét một cơ sở dữ liệu quản lý nhân sự trường đại học, ta có thể định nghĩa một Role nhân viên:

CREATE ROLE NHAN_VIEN_ROLE;

Sau đó ta gán quyền truy cập cho các bảng vào role này. Chẳng hạn cấp quyền Select trên bảng Nhân Viên:

GRANT SELECT ON NHAN_VIEN TO NHAN_VIEN_ROLE;

Khi đó, khi gán Role cho người dùng, chẳng hạn cho **USER_01**, người dùng **USER_01** sẽ có toàn bộ quyền mà **NHAN_VIEN_ROLE** có:

GRANT NHAN_VIEN_ROLE TO USER_01;

Mô hình RBAC có nhiều ưu điểm, như dễ dàng quản lý quyền truy cập thông qua Role, dễ mở rộng khi số lượng người dùng lớn, và tăng cường tính bảo mật hệ thống, tránh cấp thừa hoặc thiếu quyền.

2.2.2. Cơ chế VPD

Khi áp dụng DAC và RBAC để bảo mật dữ liệu, ta hay dùng View để kiểm soát truy cập dữ liệu theo dòng. Tuy nhiên, khi số lượng Role càng nhiều, số View cũng tăng lên theo và sẽ khó để có thể quản lý, mỗi lần cần thay đổi nghiệp vụ thì có thể phải cập nhật lại nhiều View.

VPD (Virtual Private Database) là một công cụ trong Oracle giúp kiểm soát dữ liệu linh hoạt hơn, thường dùng cho các chính sách bảo mật dữ liệu cấp dòng, bằng cách thêm vị từ (predicate) vào các truy vấn của người dùng. Khi người dùng thực hiện truy vấn, Oracle sẽ tự thêm điều kiện vào mệnh đề WHERE trong câu lệnh SQL. Nhờ đó mà mỗi người dùng chỉ có thể truy xuất dữ liệu theo quyền hạn của họ.

Để áp dụng VPD, đầu tiên ta sẽ tạo một hàm trả về điều kiện để lọc dữ liệu. Giả sử có bảng **NHANVIEN(MANV, HOTEN, VAITRO, LUONG)** và Nhân viên chỉ được xem thông tin của chính mình. Ta sẽ tạo hàm để kiểm soát quyền truy cập:

```
CREATE OR REPLACE FUNCTION VPD_POLICY (
    p_schema VARCHAR2,
    p_object VARCHAR2
)
RETURN VARCHAR2
IS
    v_role VARCHAR2(50);
    v_manv VARCHAR2(10);
AS
BEGIN
```

```
v_role := SYS_CONTEXT('USERENV', 'SESSION_USER_ROLE');
v_manv := SYS_CONTEXT('USERENV', 'SESSION_USER');

IF v_role = 'NHANVIEN' THEN
    RETURN 'MANV = "' || v_manv || '"';
ELSE
    RETURN '1 = 0';
END IF;
END;
/
```

Sau khi tạo hàm, ta gán **chính sách VPD** trên vào bảng SINHVIEN bằng cách sử dụng DBMS_RLS.ADD_POLICY:

```
BEGIN
DBMS_RLS.ADD_POLICY(
    object_schema => 'HR',
    object_name   => 'NHANVIEN',
    policy_name   => 'NHANVIEN_POLICY',
    function_schema => 'HR',
    policy_function => 'VPD_POLICY',
    statement_types => 'SELECT'
);
END;
/
```

Từ sau, mỗi khi truy vấn Oracle sẽ tự chèn thêm điều kiện vào, chẳng hạn truy vấn **SELECT * FROM NHANVIEN;**

Nếu có MANV là NV001 thì Oracle sẽ tự động đổi truy vấn thành
SELECT * FROM NHANVIEN WHERE MANV = 'NV001';

2.2.3. Chính sách nhãn Oracle - OLS

Oracle Label Security (OLS) là một tính năng bảo mật của Oracle Database cho phép bạn kiểm soát quyền truy cập dữ liệu dựa trên các nhãn (label).

OLS hoạt động bằng cách gán các nhãn cho dữ liệu và phiên làm việc của người dùng, sau đó chỉ cho phép người dùng truy cập vào dữ liệu có nhãn phù hợp với nhãn của họ.

OLS gồm 3 thành phần: level, compartment và group.

- Level: Xác định cấp bậc, mức độ bảo mật của dữ liệu muốn gán cho từng hàng, phạm vi đồ án bao gồm các level: Trưởng đơn vị > Nhân viên > Sinh viên với độ ưu tiên giảm dần tương ứng.
- Compartment: Dùng để giới hạn dữ liệu theo từng đơn vị cụ thể. Ví dụ Toán, Lý, Hóa, Hành chính.
- Group: Đề cập tiêu chí khu vực hoạt động của người dùng trong hệ thống. Ví dụ Cơ sở 1, Cơ sở 2.

Người dùng (user) nào muốn cài đặt OLS thì phải được SYSDBA cấp quyền. Ví dụ cấp quyền cho user ‘user_admin’

```
GRANT EXECUTE ON SA_LABEL_ADMIN TO user_admin;
GRANT EXECUTE ON SA_COMPONENTS TO user_admin;
GRANT EXECUTE ON SA_POLICY_ADMIN TO user_admin;
GRANT EXECUTE ON SA_USER_ADMIN TO user_admin;
```

GRANT EXECUTE ON CHAR_TO_LABEL TO user_admin;

Tạo chính sách

BEGIN

SA_SYSDBA.CREATE_POLICY(
 policy_name => 'r_policy',
 column_name => 'r_label');

END;

/

Tạo 3 nhãn cấp bậc (level). Mức độ bảo mật cách nhau 20 để nếu cần thiết, ta có thể chèn cấp bậc trung gian như Level 1.5 có mức bảo mật là 30.

EXECUTE SA_COMPONENTS.CREATE_LEVEL('r_policy',20,'L1','Level 1');
EXECUTE SA_COMPONENTS.CREATE_LEVEL('r_policy',40,'L2','Level 2');
EXECUTE SA_COMPONENTS.CREATE_LEVEL('r_policy',60,'L3','Level 3');

Tạo 4 nhãn đơn vị (compartment):

EXECUTE

SA_COMPONENTS.CREATE_COMPARTMENT('r_policy',100,'M','MANAGEMENT 1');

EXECUTE

SA_COMPONENTS.CREATE_COMPARTMENT('r_policy',100,'M','MANAGEMENT 2');

EXECUTE

SA_COMPONENTS.CREATE_COMPARTMENT('r_policy',120,'E','EMPLOYEE 1');

EXECUTE

SA_COMPONENTS.CREATE_COMPARTMENT('r_policy',120,'E','EMPLOYEE 2');

Tạo 2 nhãn khu vực (group):

EXECUTE SA_COMPONENTS.CREATE_GROUP('r_policy',20,'R20','REGION NORTH');

EXECUTE SA_COMPONENTS.CREATE_GROUP('r_policy',40,'R40','REGION SOUTH');

2.2.4. Audit

Vai trò của Audit trong vòng đòn bảo mật

- Audit đóng vai trò quan trọng trong vòng tuần hoàn bảo mật gồm 3 bước: **Phòng ngừa** → **Phát hiện** → **Phản ứng**. Trong đó, auditing đóng vai trò phát hiện, hỗ trợ kiểm soát truy cập và là cơ chế phản hồi trong hệ thống.

Mục tiêu của Audit

- Truy xuất trách nhiệm: Kiểm tra và ghi lại các hành động để phát hiện tấn công hoặc lạm quyền, sử dụng sai mục đích, có mục đích xấu.
- Chống vi phạm bảo mật: Việc biết rằng các hoạt động đang được giám sát có thể răn đe, ngăn chặn hành vi xấu.
- Hỗ trợ điều tra sự cố: Các bản ghi audit giúp truy xuất hành vi người dùng, xác minh nguyên nhân và khắc phục sự cố.
- Phản hồi hệ thống: Audit cung cấp thông tin ngược giúp điều chỉnh các cơ chế bảo vệ hiện tại.

Các phương pháp Audit phổ biến

- Application Server Logs:
 - Ghi lại truy cập ứng dụng, trạng thái (thành công/thất bại), địa chỉ IP.
 - Dùng để phát hiện các hành vi đáng ngờ như tấn công từ chối dịch vụ (DoS).
 - Application Auditing:
 - Tích hợp trực tiếp trong mã nguồn của ứng dụng.
 - Linh hoạt, dễ mở rộng nhưng phụ thuộc vào chất lượng mã và dễ bị bỏ qua nếu truy cập không qua ứng dụng.

- Trigger Auditing (quan trọng, dùng nhiều):

- Dùng trigger DML để ghi nhận thao tác dữ liệu.
- Ưu điểm: minh bạch, không cần sửa ứng dụng, có thể áp dụng chọn lọc.
- Hạn chế: không hỗ trợ mọi thao tác (ví dụ: TRUNCATE), giới hạn tham số.
- Ví dụ cách dùng:

Bước 1: Tạo bảng để lưu log audit

```
CREATE TABLE ACCOUNTS_AUDIT_22127319 (
    USERNAME    VARCHAR2(30),
    ACTION     VARCHAR2(10),
    ACCNO      NUMBER,
    OLD_BAL    NUMBER,
    NEW_BAL    NUMBER,
    ACTION_DATE DATE
) TABLESPACE USERS;
```

Bước 2: Tạo trigger để ghi lại khi có cập nhật số dư

```
CREATE OR REPLACE TRIGGER TRG_AUDIT_UPDATE_BAL
BEFORE UPDATE OF BAL ON ACCOUNTS_22127319
FOR EACH ROW
BEGIN
    INSERT INTO ACCOUNTS_AUDIT_22127319 (
        USERNAME, ACTION, ACCNO, OLD_BAL,
        NEW_BAL, ACTION_DATE
    ) VALUES (
        :USER, 'UPDATE', :OLD.ACCNO, :OLD.BAL,
        :NEW.BAL, SYSDATE
    );
END;
/
```

Bước 3: Test cập nhật thử dữ liệu

```
UPDATE ACCOUNTS_22127319
SET BAL = BAL - 5000
WHERE ACCNO = 2;
```

Bước 4: Xem kết quả trigger đã ghi lại

```
SELECT * FROM ACCOUNTS_AUDIT_22127319;
```

	USERNAME	ACTION	ACCNO	OLD_BAL	NEW_BAL	ACTION_DATE
1	ACCMMASTER	UPDATE	2	20000	15000	01-APR-25

- **Standard Database Auditing:**

- Ghi lại thao tác hệ thống, hoạt động của SYS, truy cập bảng/đối tượng, quyền hệ thống...
- Có thể lưu trữ bản ghi trong CSDL hoặc hệ điều hành.
- Ví dụ cách dùng:

Bước 1: Thực hiện ở quyền SYSDBA

```
ALTER SYSTEM SET audit_trail = DB, EXTENDED SCOPE =
SPFILE;
SHUTDOWN IMMEDIATE;
STARTUP;
```

Bước 2: Cấu hình ghi audit cho đối tượng

```
-- Kích hoạt ghi nhật ký cho thao tác SELECT trên bảng
NHANVIEN
```

```
AUDIT SELECT ON project_audit.NHANVIEN BY ACCESS;
```

Bước 3: Truy vấn thử dữ liệu

```
SELECT * FROM project_audit.NHANVIEN;
```

Bước 4: Xem kết quả audit đã được ghi lại

```
SELECT USERNAME, ACTION_NAME, OBJ_NAME,
TIMESTAMP, SQL_TEXT
FROM DBA_AUDIT_TRAIL
WHERE OBJ_NAME = 'NHANVIEN'
ORDER BY TIMESTAMP DESC;
```

- **Fine-Grained Auditing (FGA) (quan trọng, dùng nhiều):**

- Cho phép audit có điều kiện: chỉ khi truy vấn vào cột nhạy cảm, hoặc khi thỏa một điều kiện cụ thể.
- Hỗ trợ ghi lại SQL, sự kiện chi tiết, phù hợp khi cần kiểm soát chính xác.
- Ví dụ cách dùng:

Bước 1: Tạo chính sách FGA giám sát khi có một user nào đó truy xuất vào bảng ACCOUNTS_22127319 và xem số dư (BAL) ≥ 20000

```
BEGIN
```

```
DBMS_FGA.ADD_POLICY (
    object_schema => 'ACCMMASTER',
    object_name   => 'ACCOUNTS_22127319',
    policy_name   => 'ACC_BAL_AUDIT',
    audit_condition => 'BAL >= 20000',
    audit_column   => 'BAL',
    statement_types => 'SELECT',
    audit_trail    => DBMS_FGA.DB +
    DBMS_FGA.EXTENDED
);
```

```
END;
```

```
/
```

Bước 2: Tiến hành chạy code sql để test (cụ thể xem ở *SQL_TEXT* ở bước 3)

Bước 3: Xem dữ liệu Audit ghi lại

DB_USER	SQL_TEXT	OBJECT_NAME	POLICY_NAME	TIMESTAMP
16 ACCMASTER	SELECT ACCNAME FROM ACCOUNTS_22127319 WHERE BAL > 20000 AND ACCNO >= 3	ACCOUNTS_22127319	ACC_MAXBAL	31-MAR-25
17 ACCMASTER	SELECT ACCNAME FROM ACCOUNTS_22127319 WHERE BAL > 20000 AND ACCNO >= 3	ACCOUNTS_22127319	ACC_BAL_AUDIT	31-MAR-25
18 TEST122	SELECT * FROM ACCMASTER.ACCTS_22127319	ACCOUNTS_22127319	ACC_MAXBAL	01-APR-25
19 TEST122	SELECT * FROM ACCMASTER.ACCTS_22127319	ACCOUNTS_22127319	ACC_BAL_AUDIT	01-APR-25
20 ACCMASTER	SELECT * FROM ACCOUNTS_22127319	ACCOUNTS_22127319	ACC_MAXBAL	01-APR-25
21 ACCMASTER	SELECT * FROM ACCOUNTS_22127319	ACCOUNTS_22127319	ACC_BAL_AUDIT	01-APR-25
22 TEST122	SELECT * FROM ACCMASTER.ACCTS_22127319	ACCOUNTS_22127319	ACC_MAXBAL	01-APR-25
23 TEST122	SELECT * FROM ACCMASTER.ACCTS_22127319	ACCOUNTS_22127319	ACC_BAL_AUDIT	01-APR-25
24 TEST122	SELECT * FROM ACCMASTER.ACCTS_22127319	ACCOUNTS_22127319	ACC_MAXBAL	01-APR-25
25 TEST122	SELECT * FROM ACCMASTER.ACCTS_22127319	ACCOUNTS_22127319	ACC_BAL_AUDIT	01-APR-25
26 TEST122	SELECT * FROM ACCMASTER.BALMORETHAN20000	ACCOUNTS_22127319	ACC_BAL_AUDIT	01-APR-25
27 TEST122	SELECT * FROM ACCMASTER.BALMORETHAN20000	ACCOUNTS_22127319	ACC_MAXBAL	01-APR-25
28 ACCMASTER	SELECT * FROM ACCOUNTS_22127319	ACCOUNTS_22127319	ACC_MAXBAL	01-APR-25
29 ACCMASTER	SELECT * FROM ACCOUNTS_22127319	ACCOUNTS_22127319	ACC_BAL_AUDIT	01-APR-25
30 ACCMASTER	SELECT * FROM ACCOUNTS_22127319	ACCOUNTS_22127319	ACC_BAL_AUDIT	01-APR-25
31 ACCMASTER	SELECT * FROM ACCOUNTS_22127319	ACCOUNTS_22127319	ACC_MAXBAL	01-APR-25

Các tình huống nên Audit

- Đăng nhập, đăng xuất khỏi hệ thống.
- Truy cập dữ liệu trong giờ ngoài giờ làm việc.
- Truy cập dữ liệu nhạy cảm, ví dụ: lương, mật khẩu.
- Thay đổi các đối tượng CSDL, đặc biệt là thủ tục lưu trữ, trigger.
- Thay đổi quyền, vai trò người dùng, các liên kết và bản sao dữ liệu.

2.2.5. Sao lưu và phục hồi dữ liệu trong Oracle

Sao lưu và phục hồi dữ liệu là những hoạt động quan trọng để bảo vệ dữ liệu, đảm bảo toàn vẹn dữ liệu và khả năng khôi phục dữ liệu nếu xảy ra sự cố.

Trong Oracle có 2 phương pháp sao lưu chính:

- Sao lưu vật lý (Physical Backup):
 - Sao lưu vật lý là những bản sao của các tệp ghi cơ sở dữ liệu vật lý.
 - Sao lưu vật lý thường dùng RMAN (Recovery Manager).
 - Sao lưu vật lý có thể là Sao lưu nóng (Hot Backup) hoặc Sao lưu lạnh (Cold Backup).
 - Đối với Sao lưu nóng, người dùng có thể sửa cơ sở dữ liệu khi sao lưu, các thay đổi sẽ lưu lại trong file log và được đồng bộ giữa cơ sở dữ liệu gốc và bản sao. Thường dùng khi cần sao lưu đầy đủ dữ liệu và hệ thống không được phép ngừng hoạt động để thực hiện Sao lưu lạnh.
 - Đối với Sao lưu lạnh, người dùng không được phép sửa cơ sở dữ liệu khi tiến hành sao lưu, đảm bảo cơ sở dữ liệu gốc và bản sao được đồng bộ.
 - Sao lưu vật lý có thể được thực hiện theo kiểu Full Backup - sao lưu toàn bộ cơ sở dữ liệu, hoặc Incremental Backup - chỉ sao lưu các thay đổi từ lần backup gần nhất.
- Sao lưu Logic (Logical Backup)
 - Sao lưu Logic chỉ sao chép dữ liệu và tạo lệnh SQL để có thể nhập lại khi cần, được lưu thành file binary.
 - Sao lưu Logic không lưu thông tin về database instance, nên có thể phục hồi trên thiết bị khác

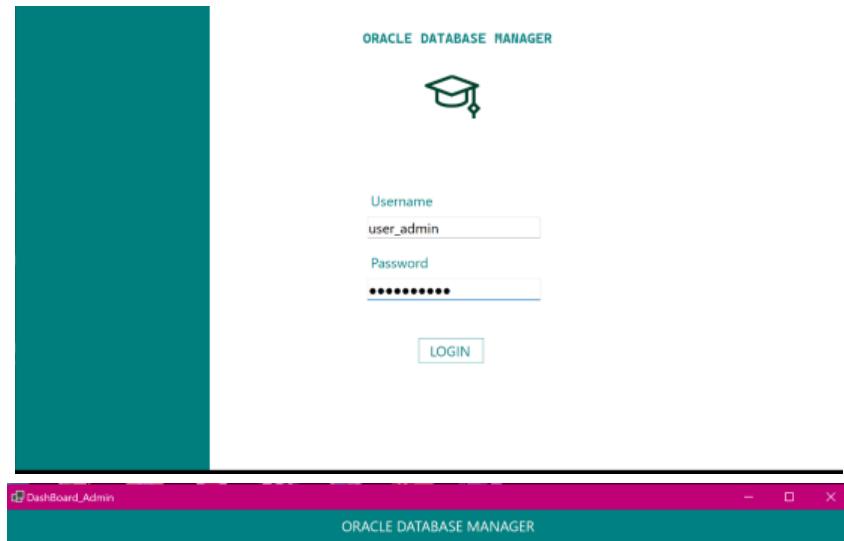
Đối với phục hồi, trong Oracle có các phương pháp sau:

- Crash Recovery: Oracle tự kích hoạt cơ chế phục hồi khi hệ thống gặp sự cố đột ngột, khôi phục dữ liệu dựa trên redo log.
- Media Recovery: Thực hiện khi có lỗi tệp hoặc khôi dữ liệu. Gồm:
 - Datafile Media Recovery: Dùng khi một hoặc nhiều tệp dữ liệu bị mất hoặc hỏng. Cần bản sao lưu và file log để khôi phục dữ liệu, có thể khôi phục toàn bộ (Complete Recovery) hoặc về một thời điểm nhất định (Point-in-Time Recovery).
 - Block Media Recovery: Chỉ khôi phục các khôi dữ liệu bị hỏng trong file. Dùng RMAN để sửa lỗi, nhanh hơn nhiều so với Datafile Recovery.

3. KẾT QUẢ THỰC HIỆN

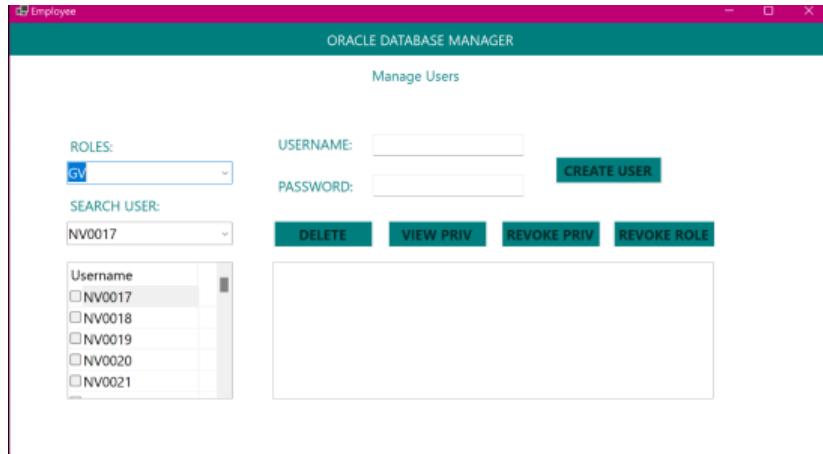
3.1. Phân hệ 1

3.1.1. Giao diện đăng nhập



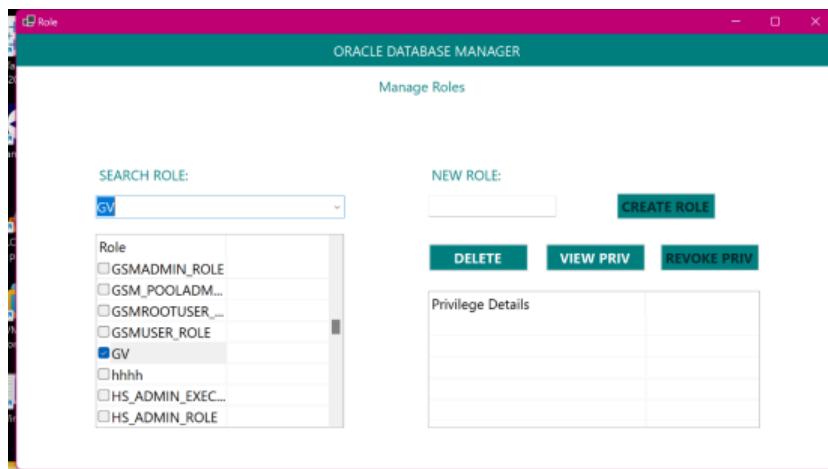
Đăng nhập với tài khoản user_admin (admin hệ thống) => dẫn đến dashboard cho chọn để thao tác.

3.1.2. Xem danh sách user trong hệ thống



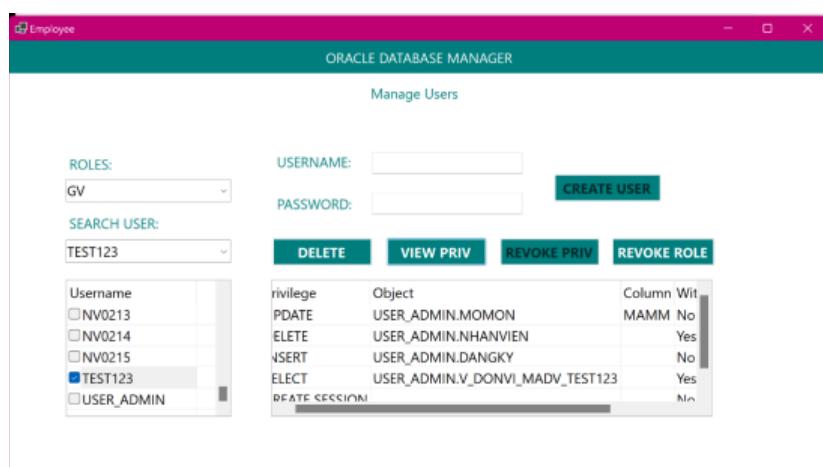
Chọn role cần tìm. Ké đến tìm kiếm mã user cần tìm (nếu user mới tạo hoặc không có role thì ở “Role” nhập “None”).

3.1.3. Xem danh sách role trong hệ thống



Có thanh tìm kiếm role.

3.1.4. Xem quyền của user trong hệ thống



Chọn vào checkbox user cần xem quyền rồi nhấn “View priv” (có xem được phân quyền tới cột, riêng view và Select tới cột thì thuộc tính cột hiển thị trên tên view). Select không cho select trên cột nên tạo view rồi cấp quyền.

3.1.5. Xem quyền của role trong hệ thống

Privilege	Object
SELECT	USER_ADMIN.DANGKY
SELECT	USER_ADMIN.NHANVIEN
SELECT	USER_ADMIN.SINHVIENT
SELECT	USER_ADMIN.V_MOMON_GV
CREATE SESSION	

Privilege	Object
UPDATE	USER_ADMIN.DONVILTRGDV
SELECT	USER_ADMIN.V_DONVI_MADV_HE
ALTER ANY TABLE	

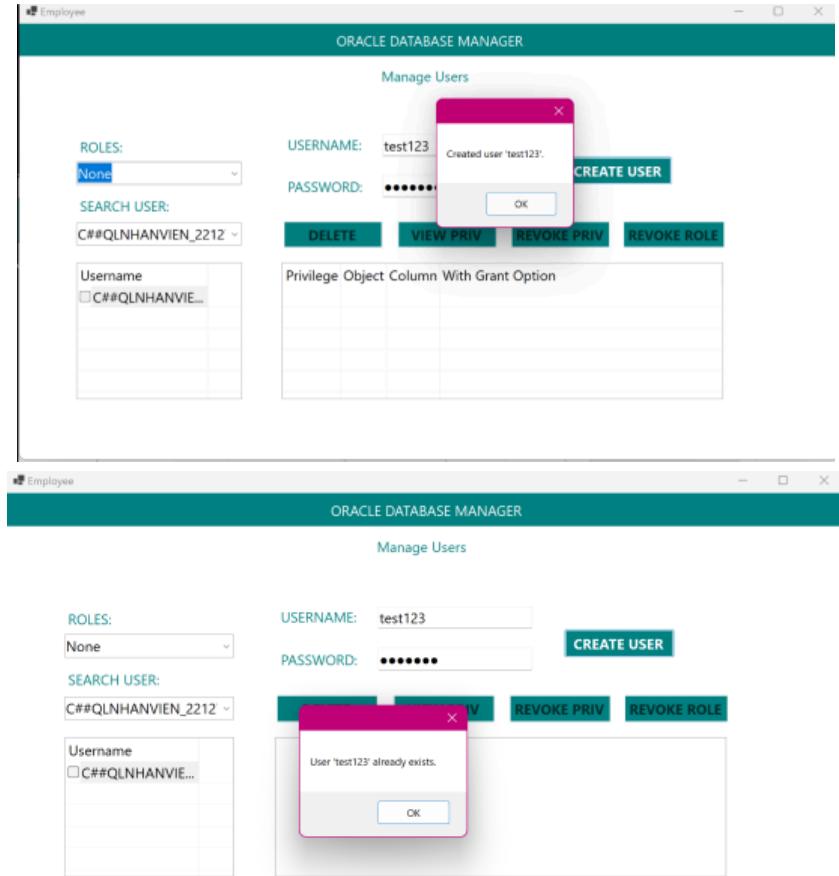
Chọn vào checkbox user cần xem quyền rồi nhấn “View priv”. Ở đây không chia thuộc tính cột nữa, cột nằm ngay trong quyền (Object).

3.1.6. Xem danh sách role trong hệ thống

Privilege Details

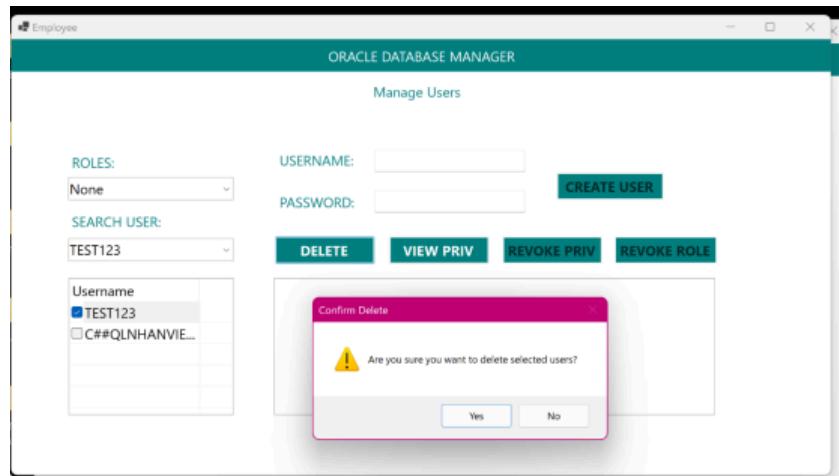
Nhấn nút tìm kiếm role.

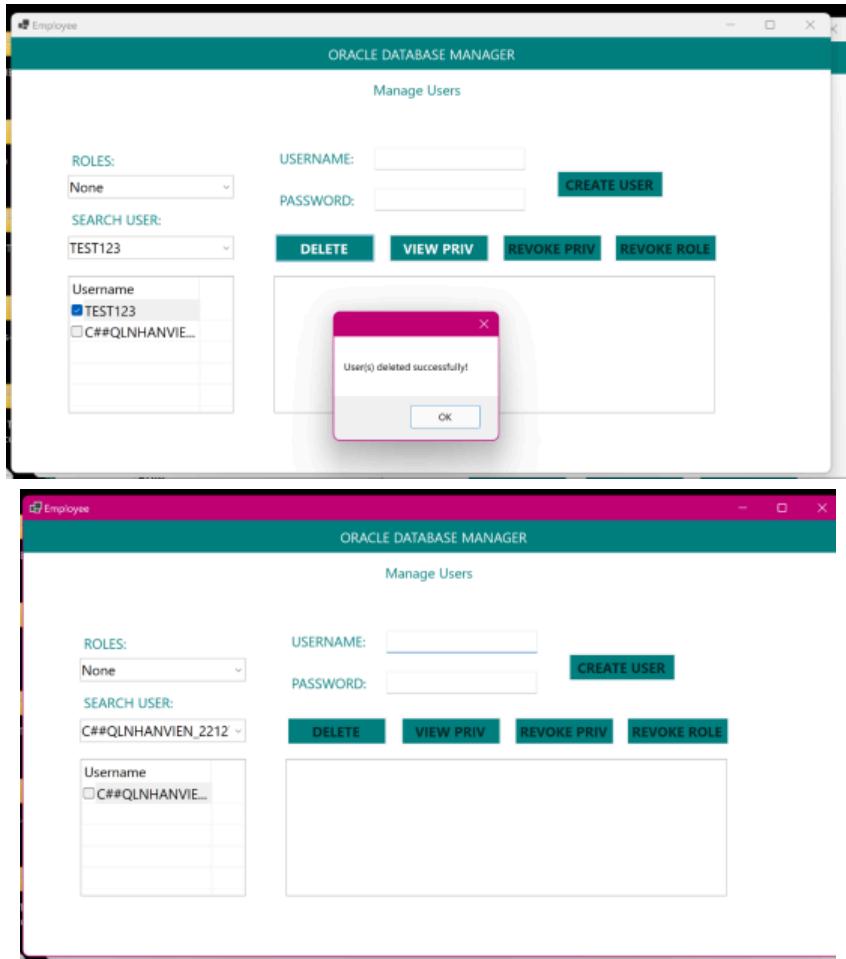
3.1.7. Tạo user mới



Nhập username password rồi nhấn “Create user”. Nếu tên user đã tồn tại thì báo lỗi.

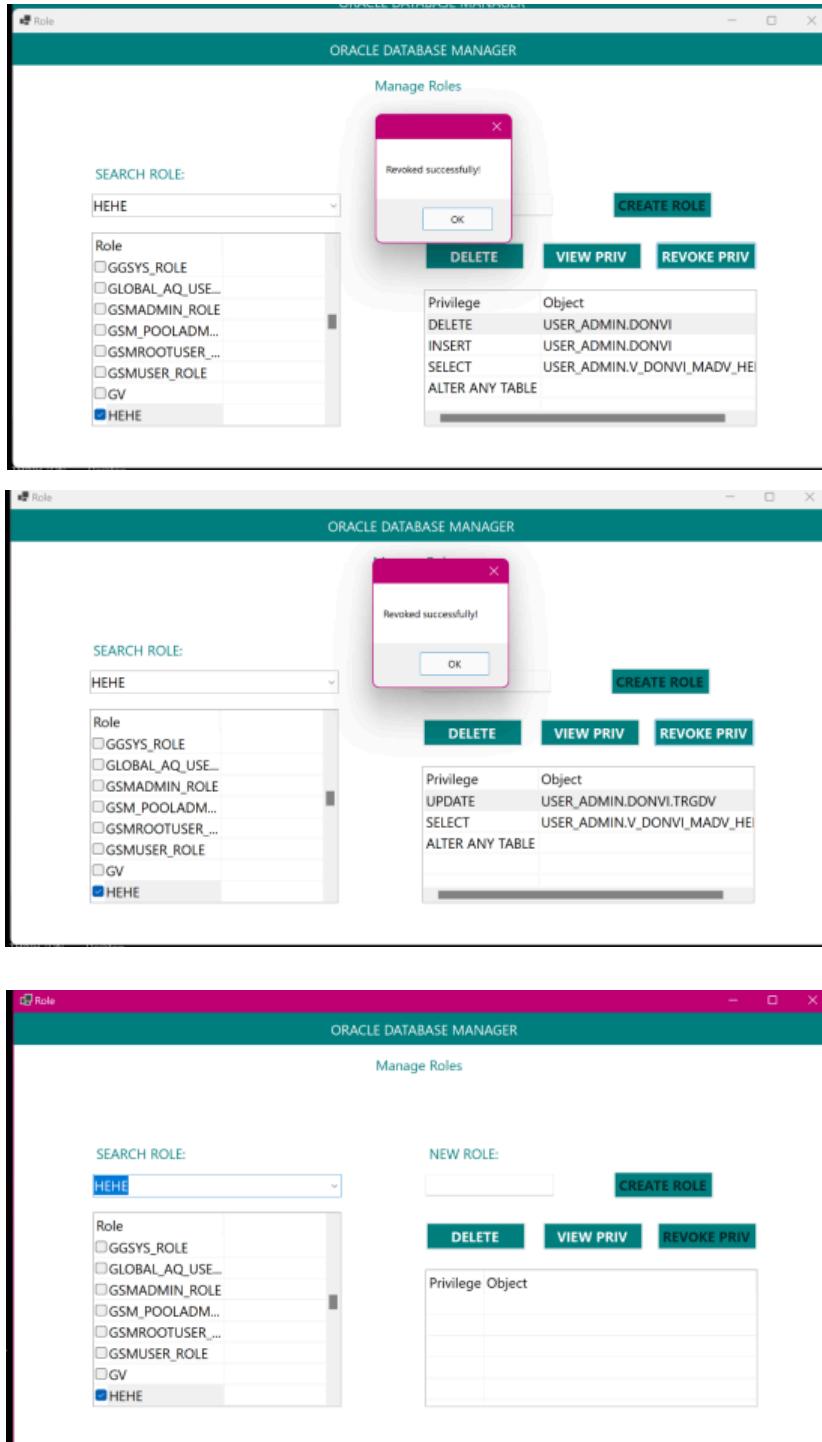
3.1.8. Xóa user





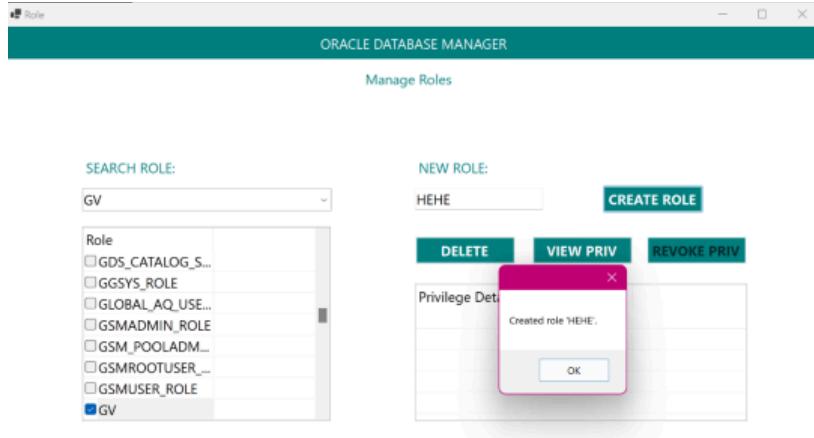
Check vào checkbox user cần xóa => Nhấn nút delete => Nhấn OK vào messagebox để xác nhận => Xóa thành công.

3.1.9. Thu hồi quyền của role



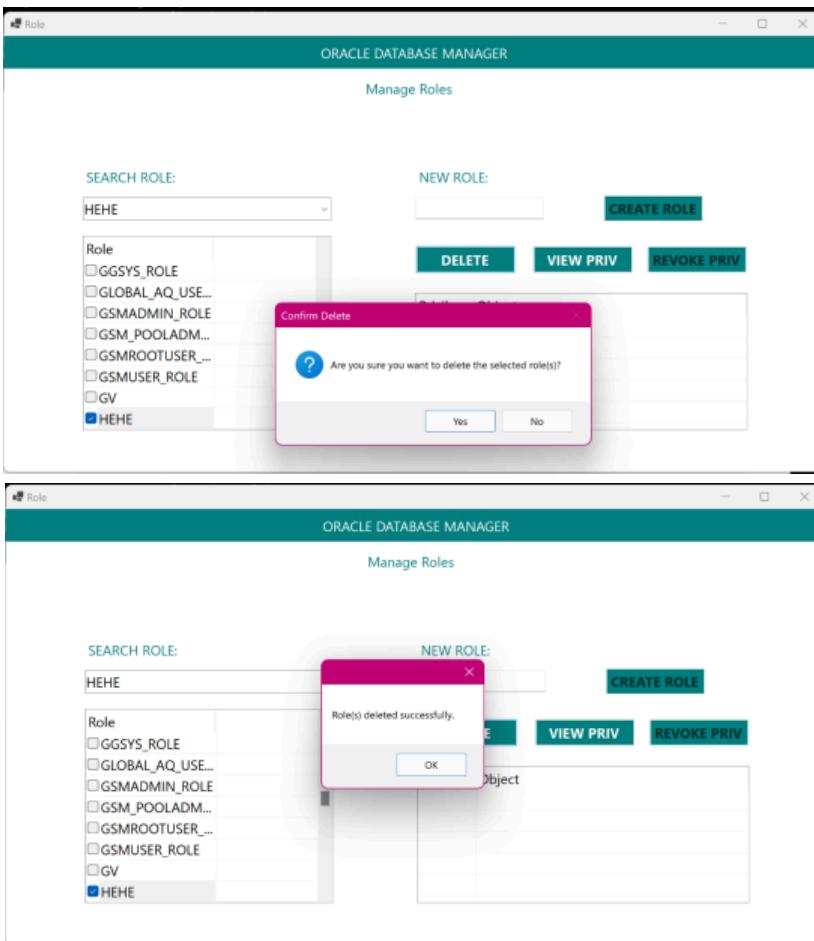
Check vào checkbox user => Nhấn “View priv” => Chọn vào quyền cần xóa => Revoke Priv.

3.1.10. Tạo role mới



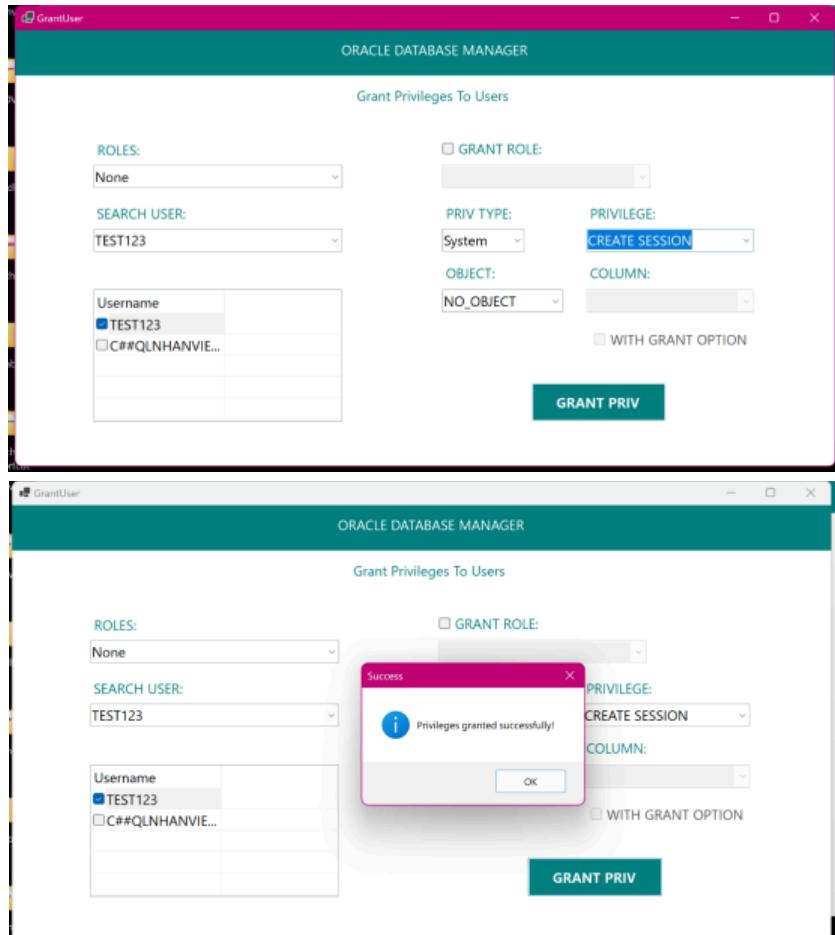
Nhập tên role mới rồi nhấn Create role. Nếu role đã tồn tại thì báo lỗi.

3.1.11. Xóa role

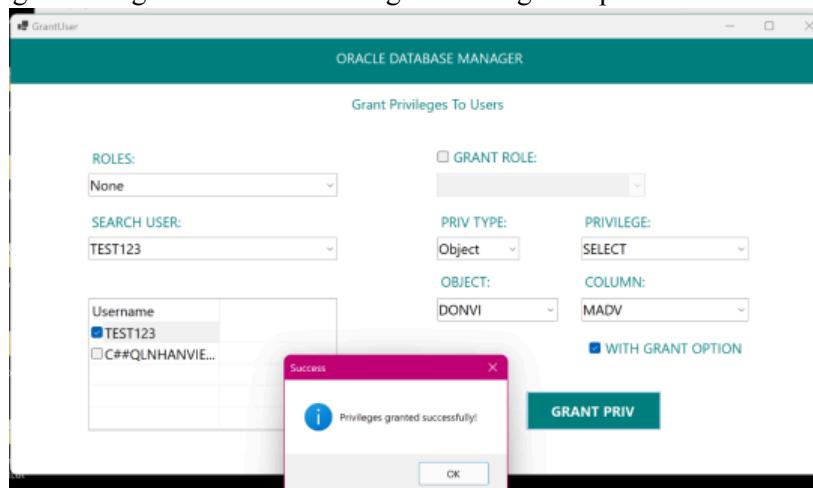


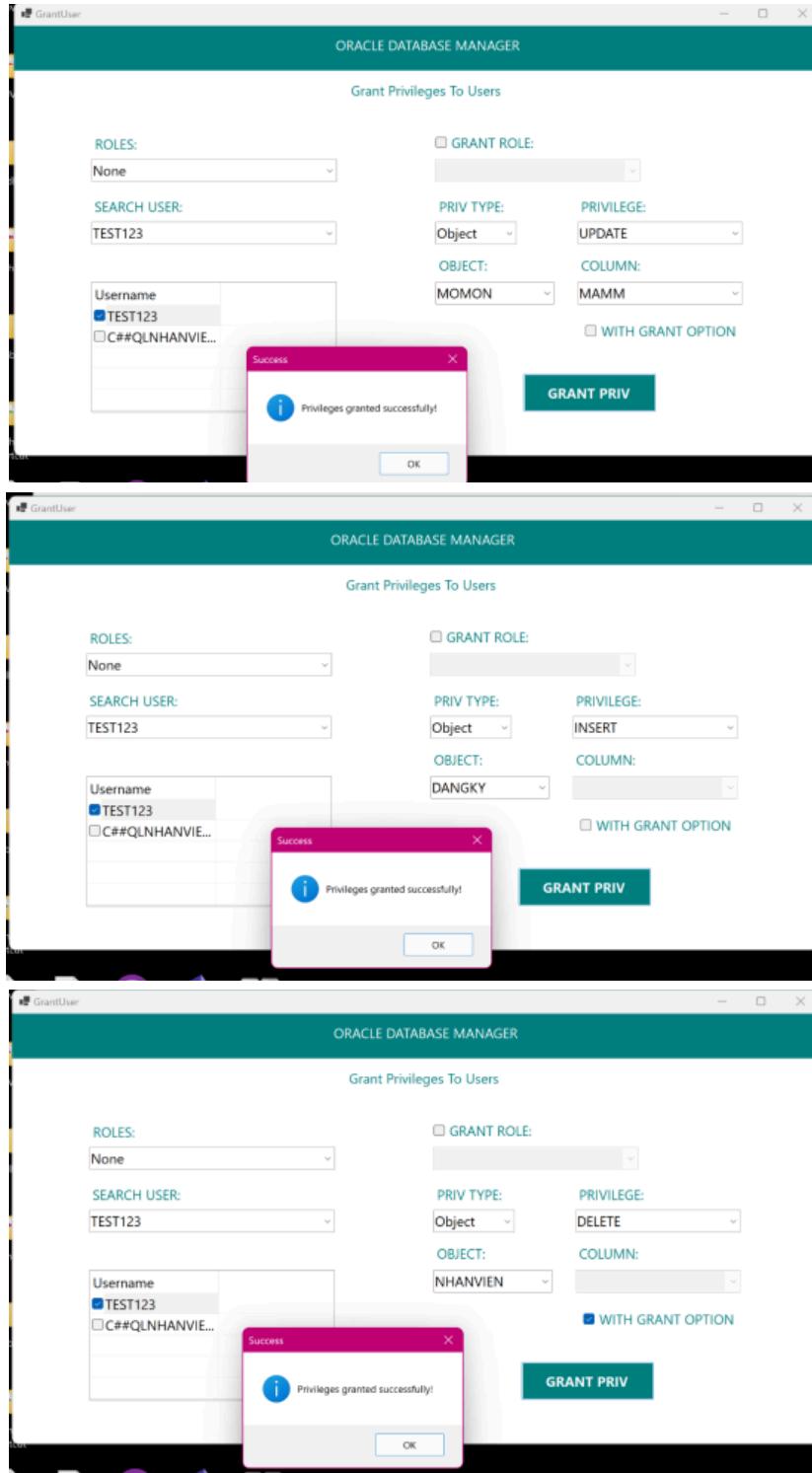
Chọn role cần xóa rồi nhấn Delete.

3.1.12. Cấp quyền cho user



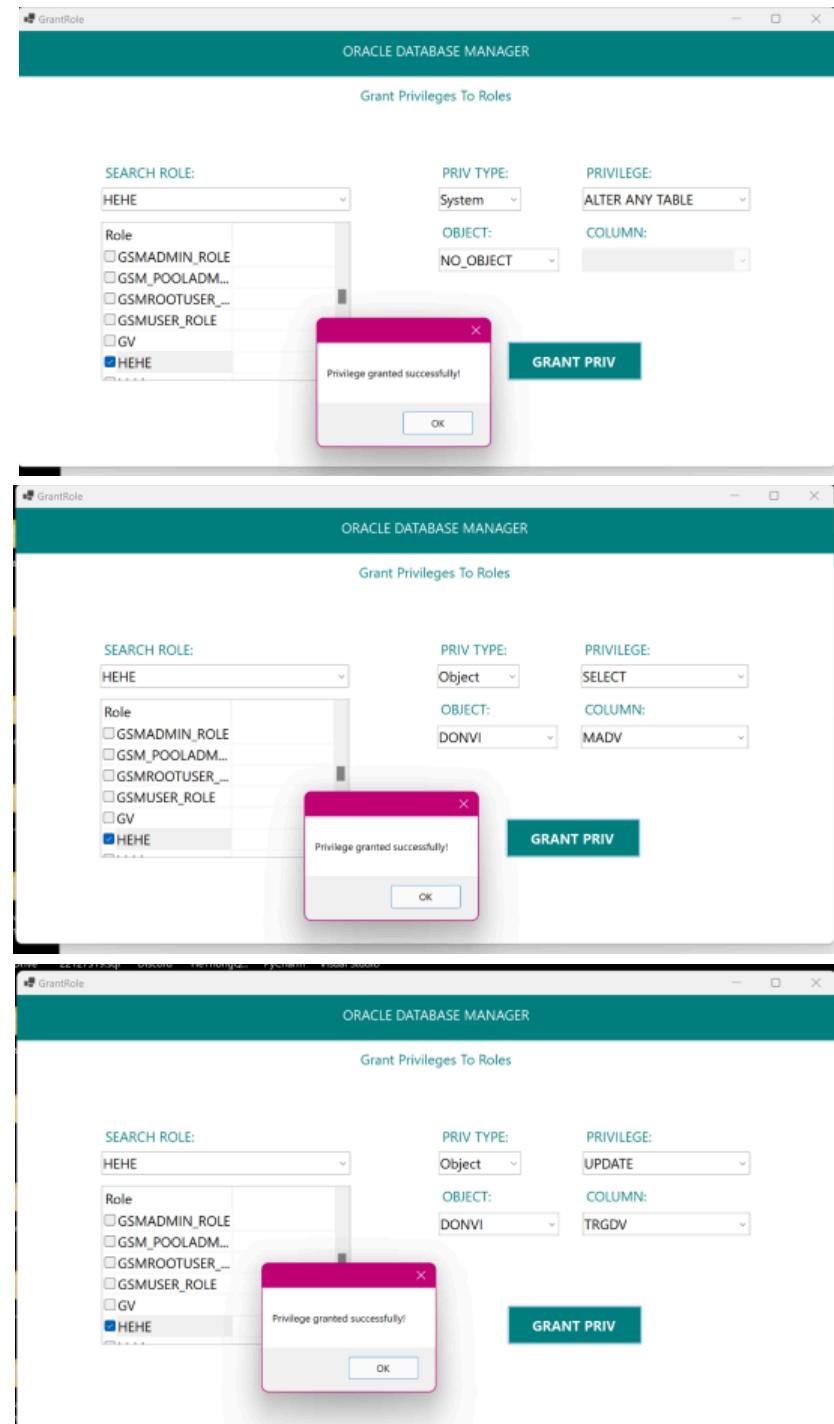
Cấp quyền hệ thống thì không xét column và không cho with grant option.

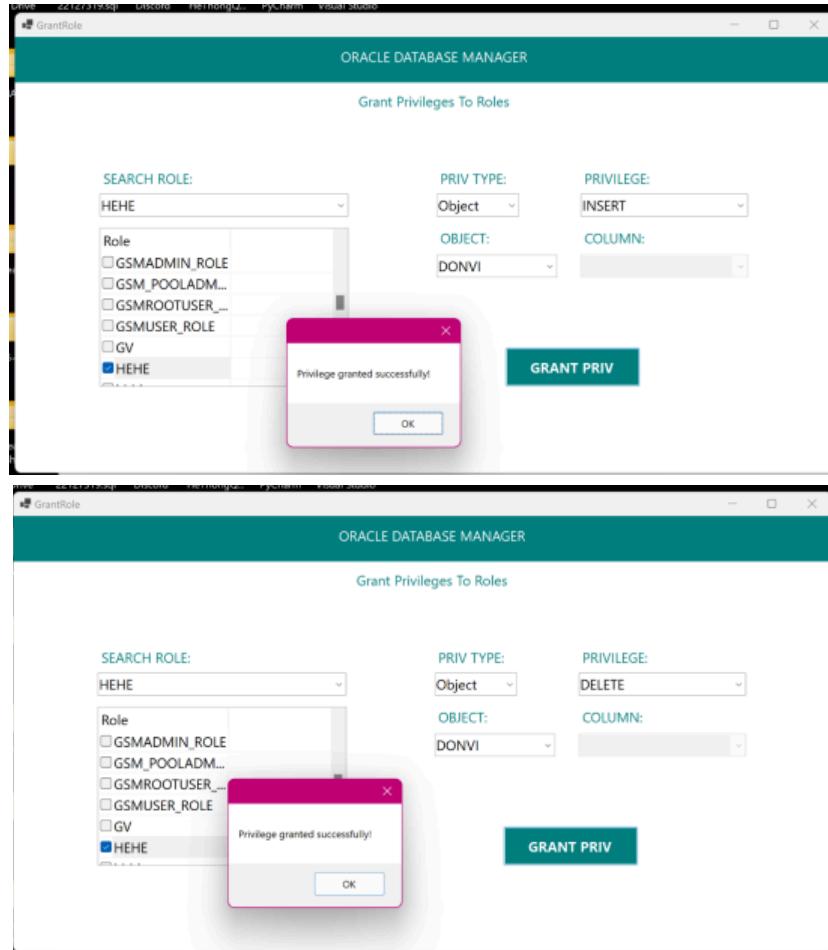




Cấp quyền object (table, view, function, stored procedure) thì có cho with grant option. Cấp quyền select và update thì cho xét tới column, các quyền khác thì không. Riêng quyền select trên cột cần tạo view rồi cấp quyền trên view vì oracle không cho cấp quyền select trên cột.

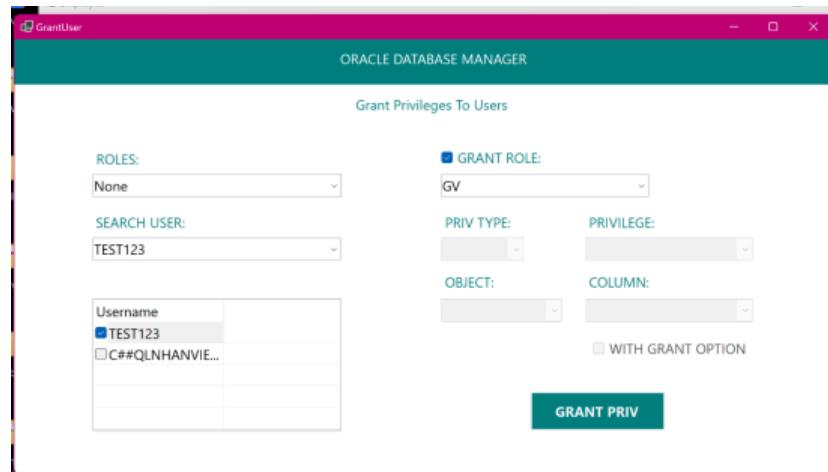
3.1.13. Cấp quyền cho role

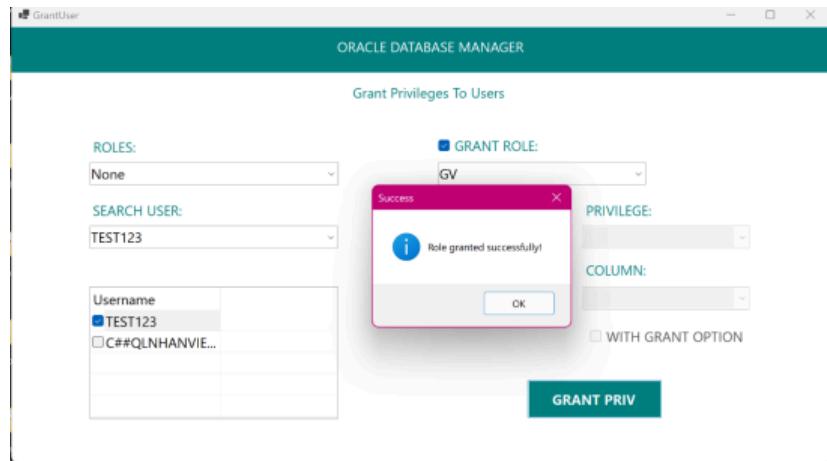




Tương tự như cấp quyền cho user nhưng không có with grant option.

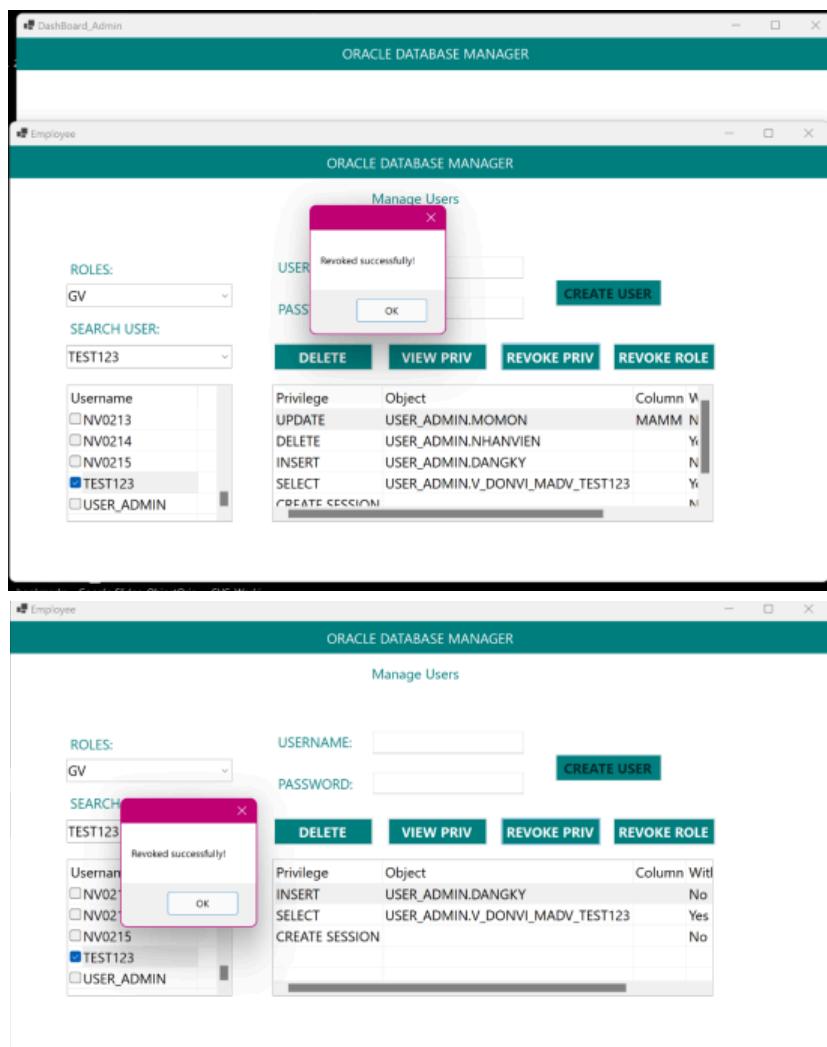
3.1.14. Cấp role cho user

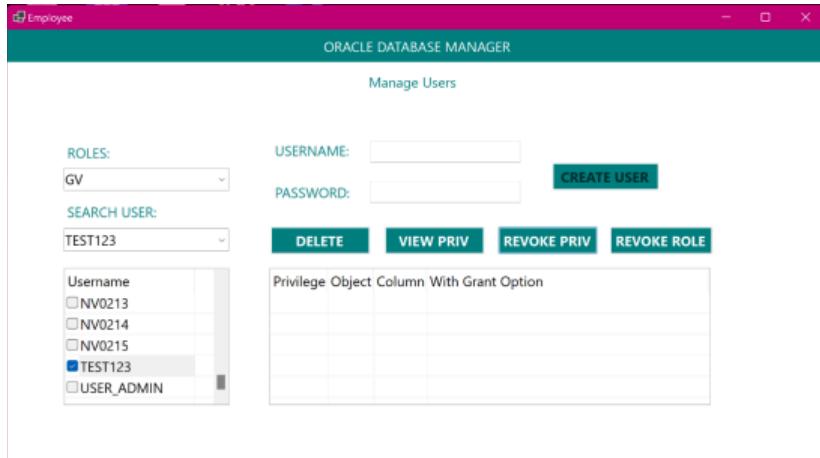




Khi tick vào ô grant role thì chuyển chế độ cấp role cho user. Chọn role muốn cấp rồi nhấn OK.

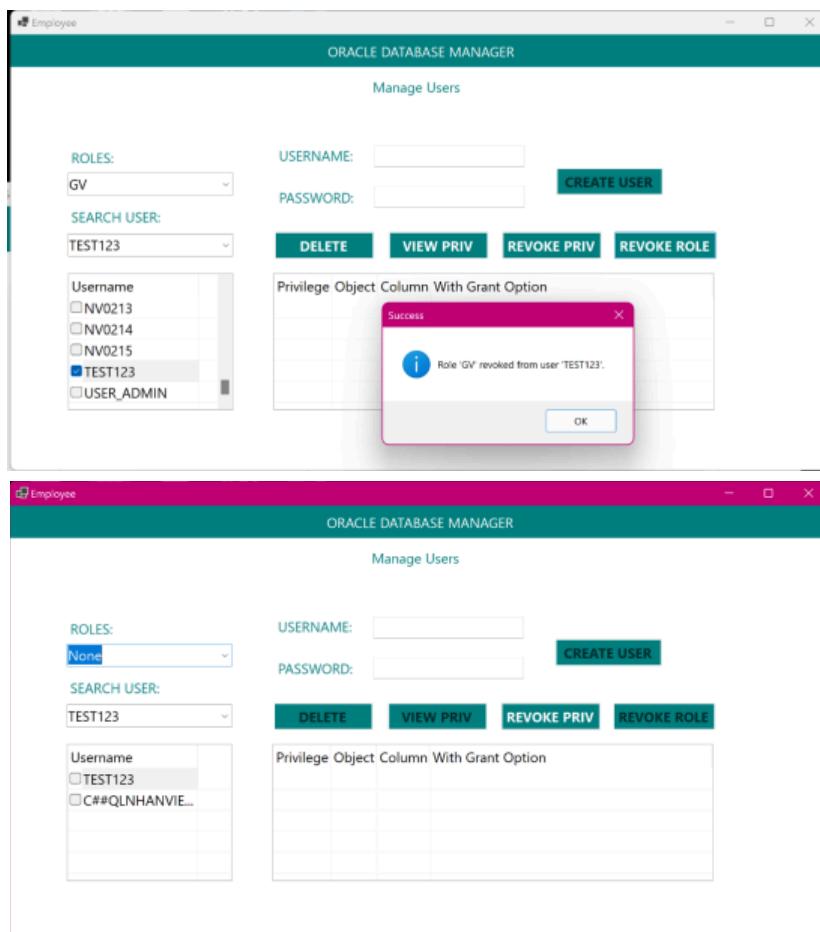
3.1.15. Thu hồi quyền của user





Chọn user => View priv => chọn quyền muốn thu hồi => Revoke priv.

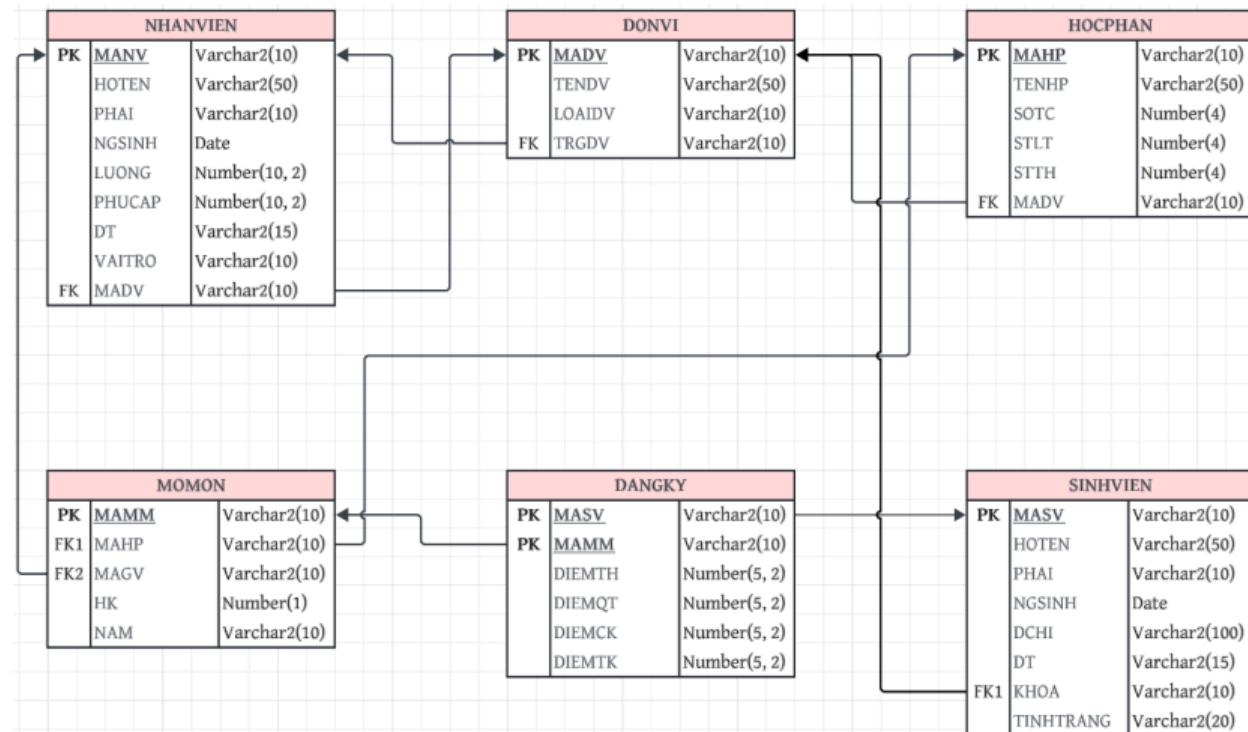
3.1.16. Thu hồi role của user



Chọn user muốn thu hồi quyền => nhấn Revoke role => Nhấn OK.

3.2. Phân hệ 2

3.2.1. Lược đồ cơ sở dữ liệu



3.2.2. Đặc tả cơ sở dữ liệu

Trong đặc tả mỗi bảng dưới đây, thuộc tính được gạch chân là khóa chính, được *in nghiêng* là khóa ngoại.

DONVI		Quan hệ lưu trữ dữ liệu về mỗi đơn vị là Khoa hoặc Phòng của trường.
Thuộc tính	Kiểu dữ liệu	Mô tả
<u>MADV</u>	Varchar2(10)	Mã đơn vị, là duy nhất, phân biệt giữa các Khoa, Phòng với nhau.
TENDV	Varchar2(50)	Tên của đơn vị tương ứng.
LOAIDV	Varchar2(10)	Nhận giá trị biểu thị Khoa hoặc Phòng
TRGDV	Varchar2(10)	Mã nhân viên là trưởng của đơn vị đó, tham chiếu đến cột MANV trong bảng NHANVIEN

NHANVIEN		Quan hệ lưu lại dữ liệu về tất cả nhân viên trong trường.
Thuộc tính	Kiểu dữ liệu	Mô tả
<u>MANV</u>	Varchar2(10)	Mã nhân viên, là duy nhất để phân biệt với các nhân

		viên khác.
HOTEN	Varchar2(50)	Họ tên của nhân viên
PHAI	Varchar2(10)	Giới tính nhân viên
NGSINH	Date	Ngày sinh của nhân viên
LUONG	Number(10, 2)	Lương của nhân viên
PHUCAP	Number(10, 2)	Tiền phụ cấp nhân viên được nhận thêm
DT	Varchar2(15)	Số điện thoại của nhân viên
VAITRO	Varchar2(10)	Vai trò của nhân viên, nhận một trong các giá trị: NVCB - Nhân viên cơ bản, GV - Giảng viên, NVPDT - Nhân viên phòng đào tạo, NVPKT - Nhân viên phòng khảo thí, NVTCHC - Nhân viên phòng tổ chức hành chính, NVCTSV - Nhân viên phòng cộng tác sinh viên, TRGDV - Trưởng đơn vị.
MADV	Varchar2(10)	Mã đơn vị của nhân viên, tham chiếu đến cột MADV trong bảng DONVI

SINHVIEN	Quan hệ lưu lại dữ liệu về tất cả sinh viên trong trường.	
Thuộc tính	Kiểu dữ liệu	Mô tả
<u>MASV</u>	Varchar2(10)	Mã sinh viên, là duy nhất để phân biệt với các sinh viên khác.
HOTEN	Varchar2(50)	Họ tên sinh viên
PHAI	Varchar2(10)	Giới tính sinh viên
NGSINH	Date	Ngày sinh của sinh viên
DCHI	Varchar2(100)	Địa chỉ của sinh viên
DT	Varchar2(15)	Số điện thoại của sinh viên
<i>KHOA</i>	Varchar2(10)	Khoa mà sinh viên đang theo học, tham chiếu đến cột MADV trong bảng DONVI
TINHTRANG	Varchar2(20)	Tình trạng của sinh viên trong trường, như Đang học, Bảo lưu, Thôi học, ...

HOCPHAN	Quan hệ lưu trữ thông tin về các học phần được dạy tại trường.	
Thuộc tính	Kiểu dữ liệu	Mô tả
<u>MAHP</u>	Varchar2(10)	Mã học phần, là duy nhất để phân biệt với các học phần khác.

TENHP	Varchar2(50)	Tên học phần
SOTC	Number(4)	Số tín chỉ của học phần
STLT	Number(4)	Số tiết học lý thuyết của học phần
STTH	Number(4)	Số tiết học thực hành của học phần
MADV	Varchar2(10)	Mã đơn vị phụ trách chuyên môn của học phần đó, tham chiếu đến cột MADV trong bảng DONVI

MOMON	Quan hệ lưu trữ thông tin về các học phần được mở trong mỗi kỳ tại trường.	
Thuộc tính	Kiểu dữ liệu	Mô tả
<u>MAMM</u>	Varchar2(10)	Mã mở môn, là duy nhất phân biệt với các lần mở môn khác.
<i>MAHP</i>	Varchar2(10)	Mã học phần được mở môn tương ứng, tham chiếu đến cột MAHP trong bảng HOPPHAN
<i>MAGV</i>	Varchar2(10)	Mã giáo viên phụ trách môn học, tham chiếu đến cột MANV trong bảng NHANVIEN
HK	Number(1)	Học kỳ mở môn, là 1, 2 hoặc 3.
NAM	Varchar2(10)	Năm học tương ứng có mở môn.

DANGKY	Quan hệ lưu các thông tin đăng ký học phần của sinh viên trong trường.	
Thuộc tính	Kiểu dữ liệu	Mô tả
<u>MASV</u>	Varchar2(10)	Mã sinh viên tạo ra đăng ký học phần, tham chiếu đến cột MASV trong bảng SINHVIEN
<u>MAMM</u>	Varchar2(50)	Mã mở môn tương ứng mà sinh viên đăng ký, tham chiếu đến cột MAMM trong bảng MOMON
DIEMTH	Number(5, 2)	Điểm thực hành của sinh viên với mở môn tương ứng.
DIEMQT	Number(5, 2)	Điểm quá trình của sinh viên với mở môn tương ứng.
DIEMCK	Number(5, 2)	Điểm thi cuối kỳ của sinh viên với mở môn tương ứng.
DIEMTK	Number(5, 2)	Điểm tổng kết mở môn của sinh viên tương ứng, được tính theo công thức DIEMTK = DIEMTH * 0.3 + DIEMQT * 0.2 + DIEMCK * 0.5

3.2.3. Các chính sách bảo mật

3.2.3.1. Yêu cầu 1 - Câu 1: Ứng dụng RBAC

Ta có yêu cầu nghiệp vụ như sau:

- Người dùng có vai trò là NVCB (nhân viên cơ bản) được xem dòng dữ liệu của chính mình trong bảng NHANVIEN, có thể chỉnh sửa DT (số điện thoại).
- Các nhân viên vai trò khác đều có quyền của NVCB.
- Người dùng có vai trò là TRGDV (trưởng đơn vị) có thể xem các dòng dữ liệu của các nhân viên thuộc đơn vị mình làm trưởng trừ cột LUONG và PHUCAP.
- Người dùng có vai trò là NVTCHC (nhân viên phòng Tổ chức hành chính) có thể xem, thêm, xóa, sửa trên quan hệ NHANVIEN.

Để giải quyết thỏa các chính sách bảo mật trên bảng quan hệ NHANVIEN, ta sẽ tạo các role tương ứng cho các vai trò NVCB, TRGDV và NVTCHC. Sau đó tạo các view và cấp các quyền cần thiết cho mỗi role.

Với role NVCB, họ được xem dữ liệu chính mình và được sửa số điện thoại, do đó ta cấp quyền SELECT và UPDATE tương ứng:

```
-- Tạo view NHANVIEN NVCB
-- NVCB xem dc dòng dữ liệu chính mình và có thể sửa SDT chính mình
CREATE OR REPLACE VIEW V_NHANVIEN_NVCB AS
SELECT MANV, HOTEN, PHAI, NGSINH, LUONG, PHUCAP, DT, VAITRO, MADV
FROM user_admin.NHANVIEN
WHERE MANV = SYS_CONTEXT('USERENV', 'SESSION_USER');

-- Role NVCB
GRANT SELECT, UPDATE (DT) ON V_NHANVIEN_NVCB TO NVCB;
```

Với role TRGDV, họ có thể xem các dòng dữ liệu của các nhân viên thuộc đơn vị mình làm trưởng trừ cột LUONG và PHUCAP, ta sẽ cấp quyền SELECT trên các cột còn lại:

```
-- View NHANVIEN TRGDV
-- TRGDV dc xem các dòng của các nv thuộc đơn vị mình làm trưởng trừ LUONG và PHUCAP
CREATE OR REPLACE VIEW V_NHANVIEN_TRGDV AS
SELECT MANV, HOTEN, PHAI, NGSINH, DT, VAITRO, MADV
FROM user_admin.NHANVIEN
WHERE MADV = (
    SELECT MADV FROM DONVI WHERE TRGDV = SYS_CONTEXT('USERENV', 'SESSION_USER')
);

-- Role TRGDV
GRANT SELECT ON V_NHANVIEN_TRGDV TO TRGDV;
```

Với role NVTCHC họ có thể thêm, xóa, sửa, xem trên bảng NHANVIEN, nên ta cấp toàn bộ các quyền.

```
-- View NHANVIEN NVTCHC
-- NVTCHC dc S/I/U/D full trên NHANVIEN
CREATE OR REPLACE VIEW V_NHANVIEN_NVTCHC AS
SELECT * FROM user_admin.NHANVIEN;
```

```
-- Role TCHC  
GRANT SELECT, INSERT, UPDATE, DELETE ON V_NHANVIEN_NVTCHC TO NVTCHC;
```

Ngoài ra, các role TRGDV và NVTCHC kế thừa các quyền của NVCB, do đó phải cấp quyền cho các role này:

```
GRANT NVCB TO NVTCHC;  
  
GRANT NVCB TO TRGDV;
```

3.2.3.2. Yêu cầu 1 - Câu 2: Ứng dụng RBAC

Ta có yêu cầu nghiệp vụ như sau:

- Người dùng có vai trò “GV” được quyền xem các dòng phân công giảng dạy liên quan đến chính giảng viên đó.
- Người dùng có vai trò “NV PDT” có quyền xem, thêm, cập nhật, xóa dòng trong bảng
- MOMON liên quan đến học kỳ hiện tại của năm học đang diễn ra.
- Người dùng có vai trò “TRGDV” có quyền xem các dòng phân công giảng dạy của các giảng viên thuộc đơn vị mình làm trưởng.
- Sinh viên có quyền xem các dòng dữ liệu trong quan hệ MOMON liên quan các dòng mới các học phần thuộc quyền phụ trách chuyên môn bởi Khoa mà sinh viên đang theo học.

Để giải quyết thỏa các chính sách bảo mật trên bảng quan hệ **MOMON**, ta sẽ tạo các role cho “GV”, “NV PDT” và “SV” ứng với yêu cầu này (role TRGDV đã được tạo ở yêu cầu trước). Ta sẽ cấp quyền cần thiết cho các role.

Role GV được quyền xem các dòng phân công giảng dạy liên quan đến chính giảng viên đó. Ta cần tạo view trước rồi cấp quyền cho role:

```
-- GV select được dòng phân công giảng dạy liên quan chính mình  
CREATE OR REPLACE VIEW V_MOMON_GV AS  
SELECT MAMM, MAHP, MAGV, HK, NAM  
FROM user_admin.MOMON  
WHERE MAGV = SYS_CONTEXT('USERENV', 'SESSION_USER');  
  
-- Role GV  
GRANT SELECT ON V_MOMON_GV TO GV;
```

Role NV PDT có quyền xem, thêm, cập nhật, xóa dòng trong bảng **MOMON**:

```
-- NVPDT được select, insert, update, delete trong MOMON liên quan học kỳ hiện tại
CREATE OR REPLACE VIEW V_MOMON_NVPDT AS
SELECT MAMM, MAHP, MAGV, HK, NAM
FROM user_admin.MOMON
WHERE
(
    (TO_CHAR(SYSDATE, 'MMDD') BETWEEN '0901' AND '1231'
        AND HK = 1
        AND SUBSTR(NAM, 1, 4) = TO_CHAR(EXTRACT(YEAR FROM SYSDATE))
    )
    OR
    (TO_CHAR(SYSDATE, 'MMDD') BETWEEN '0101' AND '0430'
        AND HK = 2
        AND SUBSTR(NAM, -4) = TO_CHAR(EXTRACT(YEAR FROM SYSDATE))
    )
    OR
    (TO_CHAR(SYSDATE, 'MMDD') BETWEEN '0501' AND '0831'
        AND HK = 3
        AND SUBSTR(NAM, -4) = TO_CHAR(EXTRACT(YEAR FROM SYSDATE))
    )
)
WITH CHECK OPTION CONSTRAINT V_MOMON_NVPDT_CHECK;

-- Role NVPDT
GRANT SELECT, INSERT, UPDATE, DELETE ON V_MOMON_NVPDT TO NVPDT;
/*
```

Role TRGDV có quyền xem các dòng phân công giảng dạy của các giảng viên thuộc đơn vị mình làm trưởng. Ta cũng tạo view trước rồi cấp quyền theo view:

```
-- TRGDV được select các dòng phân công giảng dạy của các giảng viên thuộc đơn vị mình làm trưởng
CREATE OR REPLACE VIEW V_MOMON_TRGDV AS
SELECT MAMM, MAHP, MAGV, HK, NAM
FROM user_admin.MOMON
WHERE MAGV IN (
    SELECT nv.MANV
    FROM NHANVIEN nv
    JOIN DONVI dnv ON nv.MADV = dnv.MADV
    WHERE dnv.TRGDV = SYS_CONTEXT('USERENV', 'SESSION_USER')
);
-- Role TRGDV
GRANT SELECT ON V_MOMON_TRGDV TO TRGDV;
```

Sinh viên có role có quyền xem các dòng dữ liệu trong quan hệ MOMON liên quan các dòng mở các học phần thuộc quyền phụ trách chuyên môn bởi Khoa mà sinh viên đang theo học:

```
-- SV được select các dòng phân MOMON liên quan các dòng
-- mở các học phần thuộc quyền phụ trách chuyên môn bởi Khoa của SV
CREATE OR REPLACE VIEW V_MOMON_SV AS
SELECT MAMM, MAHP, MAGV, HK, NAM
FROM user_admin.MOMON
WHERE MAHP IN (
    SELECT hp.MAHP
    FROM HOCPHAN hp
    JOIN SINHVIEN sv ON hp.MADV = sv.KHOA
    WHERE sv.MASV = SYS_CONTEXT('USERENV', 'SESSION_USER')
);
-- Role SV
GRANT SELECT ON V_MOMON_SV TO SV;
```

3.2.3.3. Yêu cầu 1 - Câu 3: Ứng dụng VPD

Ta có yêu cầu nghiệp vụ như sau:

- Sinh viên có thể xem dòng dữ liệu liên quan đến chính mình, được sửa DCHI và DT của chính mình.
- Các nhân viên vai trò khác đều có quyền của NVCB.
- Người dùng có vai trò là NVPCTSV (nhân viên phòng Công tác Sinh viên) có thể thêm, xóa, sửa thông tin trên quan hệ SINHVIEN. Tuy nhiên, trường TINHTRANG mang giá trị NULL cho đến khi người dùng với vai trò NVPDT (nhân viên phòng Đào tạo) cập nhật.
- Người dùng có vai trò là GV (giảng viên) được xem danh sách sinh viên thuộc đơn vị (khoa) mà giảng viên trực thuộc.

Để giải quyết thỏa các chính sách bảo mật trên bảng quan hệ, ta tạo các hàm trả ra vị từ cho bảng dữ liệu.

Hàm sẽ lấy tên người dùng hiện tại đang đăng nhập, để truy xuất user. Sau đó từ mã user này sẽ thực hiện lệnh Select để truy ra được vai trò và khoa từ bảng NHANVIEN, nếu không truy ra được gì thì sẽ mặc định là SINHVIEN. Sau đó sẽ xét điều kiện để trả về vị từ.

- Với VAITRO là SV thì trả về MASV là username, SV chỉ được xem dữ liệu chính mình.
- VAITRO là GV thì trả về KHOA, GV có thể xem được các sinh viên thuộc khoa tương ứng.
- NVCTSV thì có thể truy cập toàn bộ bảng SINHVIEN, nhưng không được sửa TINHTRANG, dùng trigger, sẽ được mô tả ở dưới.
- NVPDT cũng được truy cập toàn bộ bảng và có thể cập nhật TINHTRANG.
- Các vai trò còn lại trả về 1=0, không được xem dữ liệu.

Để hạn chế được NVCTSV, ta tạo trigger để cản role này chỉnh sửa cột TINHTRANG, đảm bảo chỉ có NVPDT mới được cập nhật, và để tìm được vai trò thì ta tạo một function phụ để lấy vai trò dựa theo mã nhân viên, chính là user đang đăng nhập.

Ngoài ra ta tạo một trigger để hạn chế SINHVIEN chỉ được sửa địa chỉ và số điện thoại:

3.2.3.4. Yêu cầu 1 - Câu 4: Ứng dụng VPD

Ta có yêu cầu nghiệp vụ như sau:

- Sinh viên được xem dữ liệu đăng ký học phần và bảng điểm liên quan đến chính sinh viên.
- Sinh viên có thể thêm, cập nhật đăng ký, xóa các dòng dữ liệu đăng ký học phần liên quan đến chính sinh viên trong 14 ngày đầu của học kỳ. Dữ liệu về điểm số liên quan các dòng đăng ký học phần đều mang giá trị NULL.
- Người dùng có vai trò “NV PDT” cũng có quyền xem, thêm, xóa, sửa trên quan hệ sinh viên trong thời gian 14 ngày đầu học kỳ theo nguyện vọng của sinh viên. Dữ liệu về điểm số liên quan các dòng đăng ký học phần đều mang giá trị NULL.
- Người dùng có vai trò “NV PKT” có quyền xem dữ liệu đăng ký học phần của SV và được quyền cập nhật các trường liên quan đến điểm số (theo bảng điểm do giảng viên quyết định).
- Người dùng có vai trò “GV” có quyền xem danh sách lớp, bảng điểm các lớp học phần mà giảng viên đó phụ trách giảng dạy.

Để giải quyết thỏa các chính sách bảo mật trên bảng quan hệ **SINHVIEN**, đầu tiên ta tạo các hàm trả ra vị từ cho bảng dữ liệu:

Hàm sẽ lấy user đang đăng nhập và gán cho v_user. Sau đó truy vấn bảng NHANVIEN để tìm thông tin VAITRO và MADV dựa trên MANV = v_user. Nếu không có kết quả, ta giả định v_user là sinh viên, suy ra vai trò v_vaitro sẽ được gán 'SV'. Tiếp đến hàm sẽ xác định học kì hiện tại và thời gian có thể chỉnh sửa thông tin dựa vào ngày trên hệ thống:

- Học kỳ 1: cho phép sửa thông tin từ 01/09 đến 14/09.
- Học kỳ 2: cho phép sửa thông tin từ 01/01 đến 14/01.
- Học kỳ 3: cho phép sửa thông tin từ 01/05 đến 14/05.

Dựa vào v_vaitro mà trả về vị từ tương ứng:

- Nếu là SV thì trả về mã SV, sinh viên chỉ được xem thông tin chính mình trong bảng DANGKY
- Nếu là Giảng viên, họ có thể xem thông tin các học phần mà họ giảng dạy nhờ JOIN các bảng NHANVIEN, MOMON và HOCPHAN.
- Nếu là NVPKT, hàm trả về 1=1, nhân viên có thể làm mọi thao tác như xem và cập nhật điểm số trên bảng DANGKY. Tuy nhiên các cột điểm thi sẽ bị hạn chế bởi Trigger, được mô tả ở dưới.
- Nếu là NVPDT, họ có thể select, update, insert, delete trên toàn bảng trong 14 ngày đầu của học kỳ.
- Với các vai trò còn lại, trả về 1=0, chặn truy cập.

Để đảm bảo các người dùng được truy cập trong thời gian tương ứng, ta tạo function IUD_DANGKY_14 để kiểm soát việc insert, delete và update trên DANGKY. Hàm lấy thông tin người dùng, lấy mã nhân viên, từ đó truy ngược lại ra vai trò của họ. Sau đó xác định học kì dựa ngày hệ thống (SYSDATE).

- Với SV họ chỉ được thao tác trên bảng với dòng có MASV trùng với user, và MAMM phải nằm trong học kì hiện tại, áp dụng cho 14 ngày đầu.
- Với NVPDT, họ được insert, delete và update trên toàn bộ các dòng có MAMM của học kì hiện tại trong 14 ngày đầu.
- Với NVPKT, họ được toàn quyền trên dữ liệu. Vị từ trả về là 1=1.
- Các người dùng còn lại không có quyền truy cập, trả về là 1=0.

3.2.3.5. Yêu cầu 2 - Cơ chế phát tán thông báo dùng OLS

Ta có yêu cầu nghiệp vụ như sau: Trường vận hành tại 2 cơ sở khác nhau gồm Cơ sở 1 và Cơ sở 2. Trường muốn thiết lập cho hệ thống thêm chức năng phát tán thông báo đến những người dùng trong hệ thống tùy vào cấp bậc, lĩnh vực phụ trách (theo đơn vị) và vị trí địa lý. Nội dung thông báo được lưu ở bảng THÔNGBÁO(NỘIDUNG).

Biết rằng nhân sự và các dòng thông báo được chia ra làm các cấp bậc sau: *Trưởng đơn vị, Nhân viên, Sinh viên*, độ ưu tiên giảm dần tương ứng là: *Trưởng đơn vị > Nhân viên > Sinh viên*.

Các lĩnh vực hoạt động của các đơn vị, gồm: *Toán, Lý, Hóa, Hành chính*.

Hãy thiết lập hệ thống nhằm gồm 03 thành phần và điều chỉnh mô hình dữ liệu (nếu cần thiết) để hệ thống có thể đáp ứng các yêu cầu sau, đồng thời, cài đặt giao diện minh họa trên ứng dụng

- Trong Oracle SQL Developer, có một người dùng là '**Ibacsy**' được dùng để hỗ trợ thực hiện các thao tác OLS, cần đảm bảo đăng nhập được vào tài khoản này để thực hiện các thao tác tiếp theo.
- Để giải quyết yêu cầu, đầu tiên cần kiểm tra các policy OLS hiện có trong hệ thống trước khi bắt đầu thiết lập (nên drop các policy trước đó để tránh bị lỗi trong quá trình thao tác tiếp theo)

```
--1. Kiểm tra các policy OLS hiện có trong hệ thống trước khi bắt đầu thiết lập.
SELECT policy_name FROM dba_sa_policies;

EXEC SA_SYSDBA.DROP_POLICY('SEC_POLICY');
EXEC SA_SYSDBA.DROP_POLICY('THONGBAO_OLS_DBA');
EXEC SA_SYSDBA.DROP_POLICY('THONGBAO_POLICY');
EXEC SA_SYSDBA.DROP_POLICY('NOTIFICATION1_POLICY');
EXEC SA_SYSDBA.DROP_POLICY('POL_THONGBAO');
EXEC SA_SYSDBA.DROP_POLICY('NOTIFICATION_POLICY');

SELECT policy_name FROM dba_sa_policies;
```

- Sau đó cấp quyền cần thiết cho user_admin để có thể sử dụng các thủ tục OLS

```
--2. Cấp quyền cần thiết cho user_admin để có thể sử dụng các thủ tục OLS, như tạo/cấu hình policy, component, label, v.v.

GRANT EXECUTE ON SA_LABEL_ADMIN TO user_admin;
GRANT EXECUTE ON SA_COMPONENTS TO user_admin;
GRANT EXECUTE ON SA_POLICY_ADMIN TO user_admin;
GRANT EXECUTE ON SA_USER_ADMIN TO user_admin;
GRANT EXECUTE ON CHAR_TO_LABEL TO user_admin;
```

- Sau đó tạo bảng THONGBAO để lưu những dữ liệu cần thiết.

```

--3. Tạo bảng THONGBAO - là bảng sẽ được áp dụng chính sách OLS. Cột LABEL_COL sẽ lưu nhãn bảo mật.
DROP TABLE THONGBAO;
CREATE TABLE THONGBAO (
    ID NUMBER PRIMARY KEY,
    NOIDUNG VARCHAR2(4000)
);
SELECT * FROM THONGBAO;

- Tiếp đến là các bước tạo POLICY với tên NOTIFICATION_POLICY và xác định tên cột
chứa nhãn LABEL_COL
- Sau đó tạo các levels, groups, compartments và các nhãn bảo mật (labels)

--4. Tạo chính sách OLS mới với tên NOTIFICATION_POLICY và xác định tên cột chứa nhãn (LABEL_COL).
EXEC SA_SYSDBA.CREATE_POLICY('NOTIFICATION_POLICY', 'LABEL_COL');

--5. Tạo các mức độ bảo mật (levels), đại diện cho vai trò như: Truong Don Vi, Nhan Vien, Sinh Vien
EXEC SA_COMPONENTS.CREATE_LEVEL('NOTIFICATION_POLICY', 3000, 'TRUONGDV', 'Truong Don Vi');
EXEC SA_COMPONENTS.CREATE_LEVEL('NOTIFICATION_POLICY', 2000, 'NHANVIEN', 'Nhan Vien');
EXEC SA_COMPONENTS.CREATE_LEVEL('NOTIFICATION_POLICY', 1000, 'SINHVIEN', 'Sinh Vien');

--6. Tạo các nhóm (groups), ví dụ như CS1, CS2 tương ứng với các cơ sở
EXEC SA_COMPONENTS.CREATE_GROUP('NOTIFICATION_POLICY', 10, 'CS1', 'Co So 1');
EXEC SA_COMPONENTS.CREATE_GROUP('NOTIFICATION_POLICY', 20, 'CS2', 'Co So 2');

--7. Tạo các ngăn (compartments) đại diện cho các phòng ban hoặc bộ môn như Toan, Ly, Hoa, Hanh Chinh
EXEC SA_COMPONENTS.CREATE_COMPARTMENT('NOTIFICATION_POLICY', 10, 'TOAN', 'Toan');
EXEC SA_COMPONENTS.CREATE_COMPARTMENT('NOTIFICATION_POLICY', 20, 'LY', 'Ly');
EXEC SA_COMPONENTS.CREATE_COMPARTMENT('NOTIFICATION_POLICY', 30, 'HOA', 'Hoa');
EXEC SA_COMPONENTS.CREATE_COMPARTMENT('NOTIFICATION_POLICY', 40, 'HANHCHINH', 'Hanh Chinh');

--8. Tạo các nhãn bảo mật (labels) bằng cách kết hợp level, compartment, và group theo nhu cầu bảo mật.
EXEC SA_LABEL_ADMIN.CREATE_LABEL('NOTIFICATION_POLICY', 2001, 'TRUONGDV:::');
EXEC SA_LABEL_ADMIN.CREATE_LABEL('NOTIFICATION_POLICY', 2002, 'NHANVIEN:::');
EXEC SA_LABEL_ADMIN.CREATE_LABEL('NOTIFICATION_POLICY', 2003, 'SINHVIEN:::');
EXEC SA_LABEL_ADMIN.CREATE_LABEL('NOTIFICATION_POLICY', 2004, 'SINHVIEN:HOA:CS1');
EXEC SA_LABEL_ADMIN.CREATE_LABEL('NOTIFICATION_POLICY', 2005, 'SINHVIEN:HOA:CS2');
EXEC SA_LABEL_ADMIN.CREATE_LABEL('NOTIFICATION_POLICY', 2006, 'SINHVIEN:HOA:CS1,CS2');
EXEC SA_LABEL_ADMIN.CREATE_LABEL('NOTIFICATION_POLICY', 2007, 'SINHVIEN::CS1,CS2');
EXEC SA_LABEL_ADMIN.CREATE_LABEL('NOTIFICATION_POLICY', 2008, 'TRUONGDV:HOA:CS1');
EXEC SA_LABEL_ADMIN.CREATE_LABEL('NOTIFICATION_POLICY', 2009, 'TRUONGDV:HOA:CS1,CS2');

- Thực hiện kiểm tra, sau đó áp dụng POLICY NOTIFICATION_POLICY cho bảng
THONGBAO với quyền đọc/ghi dựa trên label.

--9. Kiểm tra lại các thành phần vừa tạo để đảm bảo tạo đúng.
SELECT * FROM dba_sa_levels WHERE policy_name = 'NOTIFICATION_POLICY';
SELECT * FROM dba_sa_groups WHERE policy_name = 'NOTIFICATION_POLICY';
SELECT * FROM dba_sa_compartments WHERE policy_name = 'NOTIFICATION_POLICY';
SELECT label_tag, label FROM dba_sa_labels WHERE policy_name = 'NOTIFICATION_POLICY';

--10. Áp dụng chính sách OLS NOTIFICATION_POLICY cho bảng THONGBAO, với quyền kiểm soát đọc/ghi dựa trên label.
EXEC SA_POLICY_ADMIN.APPLY_TABLE_POLICY('NOTIFICATION_POLICY', 'USER_ADMIN', 'THONGBAO', 'READ_CONTROL,WRITE_CONTROL');

--11. Kiểm tra để xác nhận rằng bảng THONGBAO đã được áp dụng chính sách OLS.
SELECT policy_name, schema_name, table_name
FROM dba_sa_table_policies
WHERE schema_name = 'USER_ADMIN' AND table_name = 'THONGBAO';

- Sau đó tiến hành gán nhãn (labels) cho người dùng (user).

```

```
--13. Gán nhãn bảo mật cho các user (U1 đến U8), xác định ai có thể truy cập dữ liệu với nhãn tương ứng.

EXEC SA_USER_ADMIN.SET_USER_LABELS('NOTIFICATION_POLICY', 'U1', 'TRUONGDV::');
EXEC SA_USER_ADMIN.SET_USER_LABELS('NOTIFICATION_POLICY', 'U2', 'TRUONGDV:HOA:CS2');
EXEC SA_USER_ADMIN.SET_USER_LABELS('NOTIFICATION_POLICY', 'U3', 'TRUONGDV:LY:CS2');
EXEC SA_USER_ADMIN.SET_USER_LABELS('NOTIFICATION_POLICY', 'U4', 'Nhanvien:HOA:CS2');
EXEC SA_USER_ADMIN.SET_USER_LABELS('NOTIFICATION_POLICY', 'U5', 'SINHVIEN:HOA:CS2');
EXEC SA_USER_ADMIN.SET_USER_LABELS('NOTIFICATION_POLICY', 'U6', 'TRUONGDV:HANHCHINH:');
EXEC SA_USER_ADMIN.SET_USER_LABELS('NOTIFICATION_POLICY', 'U7', 'Nhanvien::');
EXEC SA_USER_ADMIN.SET_USER_LABELS('NOTIFICATION_POLICY', 'U8', 'Nhanvien:HANHCHINH:CS1');
```

- Thêm các nội dung thông báo tương ứng như đề bài

```
--14. Thêm các Thông báo vào bảng với mô tả tương ứng như trong yêu cầu đề bài

INSERT INTO THONGBAO (ID, NOIDUNG, LABEL_COL) VALUES (1, 'Thong bao cho tat ca truong don vi', CHAR_TO_LABEL('NOTIFICATION_POLICY', 'TRUONGDV::'));
INSERT INTO THONGBAO (ID, NOIDUNG, LABEL_COL) VALUES (2, 'Thong bao cho tat ca nhan vien', CHAR_TO_LABEL('NOTIFICATION_POLICY', 'Nhanvien::'));
INSERT INTO THONGBAO (ID, NOIDUNG, LABEL_COL) VALUES (3, 'Thong bao cho tat ca sinh vien', CHAR_TO_LABEL('NOTIFICATION_POLICY', 'SINHVIEN::'));
INSERT INTO THONGBAO (ID, NOIDUNG, LABEL_COL) VALUES (4, 'Thong bao cho sinh vien khoa Hoa o co so 1', CHAR_TO_LABEL('NOTIFICATION_POLICY', 'SINHVIEN:HOA:CS1'));
INSERT INTO THONGBAO (ID, NOIDUNG, LABEL_COL) VALUES (5, 'Thong bao cho sinh vien khoa Hoa o co so 2', CHAR_TO_LABEL('NOTIFICATION_POLICY', 'SINHVIEN:HOA:CS2'));
INSERT INTO THONGBAO (ID, NOIDUNG, LABEL_COL) VALUES (6, 'Thong bao cho sinh vien khoa Hoa o ca hai co so', CHAR_TO_LABEL('NOTIFICATION_POLICY', 'SINHVIEN:HOA:CS1,CS2'));
INSERT INTO THONGBAO (ID, NOIDUNG, LABEL_COL) VALUES (7, 'Thong bao cho tat ca sinh vien o ca hai co so', CHAR_TO_LABEL('NOTIFICATION_POLICY', 'SINHVIEN::CS1,CS2'));
INSERT INTO THONGBAO (ID, NOIDUNG, LABEL_COL) VALUES (8, 'Thong bao cho truong khoa Hoa o co so 1', CHAR_TO_LABEL('NOTIFICATION_POLICY', 'TRUONGDV:HOA:CS1'));
INSERT INTO THONGBAO (ID, NOIDUNG, LABEL_COL) VALUES (9, 'Thong bao cho truong khoa Hoa o ca hai co so', CHAR_TO_LABEL('NOTIFICATION_POLICY', 'TRUONGDV:HOA:CS1,CS2'));
```

- Cấp quyền cho người dùng từ u1 đến u8 để có thể kết nối

```
--15. Cấp quyền CONNECT cho các user U1-U8 để họ có thể đăng nhập vào database.

GRANT CONNECT TO u1;
GRANT CONNECT TO u2;
GRANT CONNECT TO u3;
GRANT CONNECT TO u4;
GRANT CONNECT TO u5;
GRANT CONNECT TO u6;
GRANT CONNECT TO u7;
GRANT CONNECT TO u8;
```

- Kết quả: ví dụ người dùng u2 là Trưởng đơn vị phụ trách khoa Hóa ở cơ sở 2, thì sẽ đọc được các thông báo sau:

The screenshot shows the MySQL Workbench interface with several tabs open at the top: ...sql, Welcome Page, 2304_useradmin, sapt2204.sql, 22_u1, and 22_u2. The active tab is '22_u2'. Below the tabs is a toolbar with various icons. The main area has two tabs: 'Worksheet' and 'Query Builder', with 'Worksheet' selected. In the worksheet, the following SQL query is written:

```
select * from user_admin.THONGBAO;
```

Below the worksheet is a 'Query Result' tab. It displays the results of the query as a table:

ID	NOIDUNG	LABEL_COL
1	1 Thong bao cho tat ca truong don vi	2001
2	2 Thong bao cho tat ca nhan vien	2002
3	3 Thong bao cho tat ca sinh vien	2003
4	5 Thong bao cho sinh vien khoa Hoa o co so 2	2005
5	6 Thong bao cho sinh vien khoa Hoa o ca hai co so	2006
6	7 Thong bao cho tat ca sinh vien o ca hai co so	2007
7	9 Thong bao cho truong khoa Hoa o ca hai co so	2009

3.2.3.6. Yêu cầu 3 - Ghi nhật ký hệ thống bằng Audit

Kích hoạt việc ghi nhật ký hệ thống.

Kết nối với quyền sys để kích hoạt:

ALTER SYSTEM SET audit_trail = DB, EXTENDED SCOPE = SPFILE;

-- Khởi động lại CSDL để áp dụng

SHUTDOWN IMMEDIATE;

STARTUP;

Thực hiện ghi nhật ký hệ thống dùng Standard audit:

Sinh viên và nhân viên là những người dùng trực tiếp của hệ thống. Việc thêm, xóa nhân viên hoặc sinh viên cần được ghi nhận để đảm bảo tính minh bạch và truy vết được hành vi sai phạm nếu xảy ra.

→ Audit hành động INSERT, DELETE trên bảng NHANVIEN và SINHVIEN

Dữ liệu đăng ký học phần là rất quan trọng vì liên quan đến quá trình học tập và bảng điểm của sinh viên. Cần ghi nhận lại các hành vi thêm, xóa dữ liệu để đảm bảo hệ thống hoạt động chính xác và có thể truy cứu nếu có thao tác bất thường.

→ Audit INSERT, DELETE trên bảng DANGKY

Thông tin cơ bản của sinh viên được hiển thị qua view, nhưng cũng cần kiểm soát việc truy xuất dữ liệu nhạy cảm, nhất là trong môi trường nhiều người dùng.

→ Audit hành động SELECT trên view v_sv_tt

Các thao tác đăng ký học phần có thể được thực hiện qua thủ tục lưu trữ (procedure).

Cần ghi nhận khi người dùng thực thi các procedure để theo dõi hoạt động hệ thống.

→ Audit hành động EXECUTE trên procedure dk_hocphan

Việc tính điểm tổng kết cũng được thực hiện qua một hàm tính toán. Ghi nhận việc gọi hàm giúp đảm bảo rằng dữ liệu được tính toán đúng thời điểm, đúng người.

→ Audit hành động EXECUTE trên function tinh_diemtk

Cài đặt Standard audit bằng tài khoản ADMIN/DBA của PDB:

-- TẠO USER TEST

CREATE USER user_nv IDENTIFIED BY nv123;

GRANT CONNECT, CREATE SESSION TO user_nv;

GRANT RESTRICTED SESSION TO user_nv;

CREATE USER user_sv IDENTIFIED BY sv123;

GRANT CONNECT, CREATE SESSION TO user_sv;

GRANT RESTRICTED SESSION TO user_sv;

-- TẠO VIEW

CREATE OR REPLACE VIEW v_sv_tt AS

SELECT MASV, HOTEN, KHOA, TINHTRANG FROM SINHVIEN;

-- TẠO PROCEDURE ĐĂNG KÝ HỌC PHẦN

CREATE OR REPLACE PROCEDURE dk_hocphan (

p_masv VARCHAR2,

p_mamm VARCHAR2

) AS

BEGIN

```
INSERT INTO DANGKY (MASV, MAMM)
VALUES (p_masv, p_mamm);
END;
/  

-- TẠO FUNCTION TÍNH ĐIỂM
CREATE OR REPLACE FUNCTION tinh_diemtk(
    th NUMBER, qt NUMBER, ck NUMBER
) RETURN NUMBER IS
BEGIN
    RETURN ROUND(th*0.2 + qt*0.3 + ck*0.5, 2);
END;
/  

-- CẤP QUYỀN CHO USER
GRANT INSERT, SELECT, UPDATE, DELETE ON DONVI TO user_nv;
GRANT INSERT, SELECT, UPDATE, DELETE ON NHANVIEN TO user_nv;
GRANT INSERT, SELECT, UPDATE, DELETE ON SINHVIEN TO user_nv;  

GRANT INSERT, SELECT, UPDATE ON SINHVIEN TO user_sv;
GRANT INSERT, SELECT, DELETE ON DANGKY TO user_sv;  

GRANT EXECUTE ON tinh_diemtk TO user_sv;
GRANT EXECUTE ON dk_hocphan TO user_sv;  

-- BẬT GHI NHẬT KÝ – STANDARD AUDIT  

-- 1. Theo dõi INSERT, DELETE trên bảng NHANVIEN
AUDIT INSERT, DELETE ON NHANVIEN BY ACCESS;  

-- 2. Theo dõi DELETE thất bại trên bảng SINHVIEN
AUDIT DELETE ON SINHVIEN WHENEVER NOT SUCCESSFUL;  

-- 3. Theo dõi INSERT, DELETE trên bảng DANGKY
AUDIT INSERT, DELETE ON DANGKY BY ACCESS;  

-- 4. Theo dõi SELECT trên view v_sv_tt
AUDIT SELECT ON v_sv_tt BY ACCESS;  

-- 5. Theo dõi gọi thủ tục dk_hocphan
AUDIT EXECUTE ON dk_hocphan BY ACCESS;  

-- 6. Theo dõi gọi hàm tinh_diemtk
AUDIT EXECUTE ON tinh_diemtk BY ACCESS;
```

Ngữ cảnh: Người dùng user_nv là nhân viên có quyền chỉnh sửa bảng NHANVIEN và DONVI.

Các thao tác kiểm thử gồm:

- Thêm mới một đơn vị có mã "PDT" và tên "Phòng Đào tạo" vào bảng DONVI.
- Thêm mới một nhân viên tên "Trần Minh Tuấn" thuộc đơn vị "PDT" vào bảng NHANVIEN. Nhân viên này có vai trò "NV TCHC".

- Xóa nhân viên vừa tạo ra khỏi bảng NHANVIEN

Mục đích:

- Ghi nhận các thao tác INSERT và DELETE trên bảng NHANVIEN bằng cơ chế Standard Audit đã khai báo.
- Kiểm tra rằng nhật ký ghi nhận các thao tác thành công thực hiện bởi user_nv.

Ngữ cảnh: user_sv là người dùng đại diện cho sinh viên trong hệ thống được cấp các quyền truy cập và thao tác giới hạn trên bảng SINHVIEN, DANGKY, view và một số thủ tục (procedure, function).

Các thao tác kiểm thử gồm:

- Thêm một sinh viên mới vào bảng SINHVIEN, có mã số sinh viên là SV6000.
- Truy vấn thông tin sinh viên từ view v_sv_tt. Do chưa được cấp quyền nên báo lỗi.
- Thủ xóa sinh viên vừa thêm. Do không được cấp quyền DELETE, thao tác này sẽ bị từ chối và hệ thống sẽ ghi nhận lỗi với mã ORA-01031: insufficient privileges.
- Thêm dòng đăng ký học phần vào bảng DANGKY (sinh viên đăng ký học phần MM001).
- Gọi thủ tục dk_hocphan để thực hiện đăng ký học phần qua procedure.
- Gọi hàm tinh_diemtk để tính điểm tổng kết dựa trên điểm thành phần.

Mục đích kiểm thử:

- Ghi nhận các thao tác SELECT, INSERT, EXECUTE thành công từ user_sv thông qua Standard Audit.
- Ghi nhận thao tác DELETE bị từ chối trên bảng SINHVIEN thông qua chính sách WHENEVER NOT SUCCESSFUL.
- Kiểm thử bổ sung cho Fine-Grained Audit (FGA) thông qua thao tác sai vai trò hoặc vượt quyền trên bảng DANGKY, NHANVIEN.

Dùng Fine-grained Audit hoặc Unified Audit để thực hiện ghi nhật ký các tình huống sau và tạo tình huống để ghi nhật ký được (có dữ liệu ghi nhật ký) các hành vi sau:

a. Hành vi cập nhật quan hệ DANGKY tại các trường liên quan đến điểm số nhưng người đó không thuộc vai trò “NV PKT”.

b. Hành vi của người dùng (không thuộc vai trò “NV TCHC”) có thể đọc trên trường LUONG, PHUCAP của người khác hoặc cập nhật ở quan hệ NHANVIEN.

c. Hành vi thêm, xóa, sửa trên quan hệ DANGKY của sinh viên nhưng trên dòng dữ liệu của sinh viên khác hoặc thực hiện hiệu chỉnh đăng ký học phần ngoài thời gian cho phép hiệu chỉnh đăng ký học phần.

Dùng Fine-grained Audit để thực hiện ghi nhật ký các tình huống

-- 3(a): Cập nhật bảng DANGKY ở các trường liên quan đến điểm số khi không thuộc vai trò “NV PKT”

BEGIN

```
DBMS_FGA.ADD_POLICY (
    object_name      => 'DANGKY',
    policy_name      => 'POL_AUDIT_DIEM_UPDATE',
```

```
audit_condition => 'SYS_CONTEXT("USERENV", "SESSION_USER") NOT IN
(SELECT USERNAME FROM USERS_EXTENDED WHERE VAITRO = "NV
P KT")',
audit_column    => 'DIEMTH, DIEMQT, DIEMCK, DIEMTK',
statement_types => 'UPDATE',
audit_trail     => DBMS_FGA.DB_EXTENDED
);
END;
/
```

-- 3(b): Người không thuộc vai trò “NV TCHC” đọc hoặc cập nhật LUONG, PHUCAP của người khác

```
BEGIN
DBMS_FGA.ADD_POLICY (
object_name    => 'NHANVIEN',
policy_name    => 'POL_LUONG_PHUCAP_ACCESS',
audit_condition => 'SYS_CONTEXT("USERENV", "SESSION_USER") NOT IN
(SELECT USERNAME FROM USERS_EXTENDED WHERE VAITRO = "NV
TCHC")',
audit_column    => 'LUONG, PHUCAP',
statement_types => 'SELECT, UPDATE',
audit_trail     => DBMS_FGA.DB_EXTENDED
);
END;
/
```

-- 3(c): Sinh viên thêm/xóa/sửa DANGKY trên dòng của sinh viên khác hoặc ngoài thời gian cho phép

```
BEGIN
DBMS_FGA.ADD_POLICY (
object_name    => 'DANGKY',
policy_name    => 'POL_DANGKY_SV_CHECK',
audit_condition => 'MASV <> SYS_CONTEXT("USERENV", "SESSION_USER")
OR NOT IS_VALID_TIME()', 
statement_types => 'INSERT, UPDATE, DELETE',
audit_trail     => DBMS_FGA.DB_EXTENDED
);
END;
/
```

Ngữ cảnh FGA (a): Cập nhật điểm khi không phải “NV P KT”

Người dùng không thuộc vai trò "NV P KT" có tình trạng cập nhật điểm số (DIEMTH, DIEMQT, DIEMCK, DIEMTK) trong bảng DANGKY.

=> Mục tiêu: Ghi nhật ký hành vi cập nhật sai vai trò.

Ngữ cảnh (b): Truy cập lương/phụ cấp khi không phải “NV TCHC”

Người dùng không thuộc vai trò "NV TCHC" truy vấn hoặc cập nhật thông tin LUONG, PHUCAP trong bảng NHANVIEN.

=> Mục tiêu: Ghi nhật ký truy cập trái phép dữ liệu nhạy cảm.

Ngữ cảnh (c): Sinh viên sửa/xóa đăng ký sai dòng hoặc sai thời điểm

Sinh viên thao tác trên dòng không thuộc mã số sinh viên của mình, hoặc thao tác ngoài thời gian cho phép trên bảng DANGKY.

=> Mục tiêu: Ghi nhật ký vi phạm quyền dữ liệu hoặc thời gian đăng ký.

Đọc dữ liệu nhật ký hệ thống

-- 1. Standard Audit

```
SELECT USERNAME, ACTION_NAME, OBJ_NAME, SQL_TEXT, TIMESTAMP
FROM DBA_AUDIT_TRAIL
WHERE OBJ_NAME IN ('NHANVIEN', 'DANGKY');
```

USERNAME	ACTION_NAME	OBJ_NAME	SQL_TEXT	TIMESTAMP
1 USER_NV	INSERT	NHANVIEN	(null)	01-APR-25
2 USER_SV	INSERT	DANGKY	(null)	01-APR-25
3 USER_NV	INSERT	NHANVIEN	(null)	01-APR-25
4 USER_NV	INSERT	NHANVIEN	(null)	01-APR-25
5 USER_NV	INSERT	NHANVIEN	(null)	01-APR-25
6 USER_SV	INSERT	DANGKY	(null)	01-APR-25
7 USER_SV	INSERT	DANGKY	(null)	01-APR-25
8 USER_NV	DELETE	NHANVIEN	(null)	01-APR-25

-- 2. Fine-Grained Audit

```
SELECT DB_USER, OBJECT_NAME, SQL_TEXT, STATEMENT_TYPE,
POLICY_NAME, TIMESTAMP
FROM DBA_FGA_AUDIT_TRAIL
WHERE OBJECT_NAME IN ('NHANVIEN', 'DANGKY')
ORDER BY TIMESTAMP DESC;
```

4. TÀI LIỆU THAM KHẢO

[Oracle-base Oracle Label Security](#)

[Recovery Principle](#)

[Physical Backup](#)

[Logical Backup](#)

Sách **Oracle® Label Security**

Sách **McGraw-Hill Osborne - Effective Oracle Database**

Slide môn học **An toàn và bảo mật dữ liệu trong hệ thống thông tin.**