CSC12001 – AN TOÀN VÀ BẢO MẬT DỮ LIỆU TRONG HTTT

Phân hệ 1 – Ứng dụng quản trị CSDL Oracle
Phân hệ 2 – Ứng dụng quản lý dữ liệu nội bộ

Đồ án môn học 2023 – 2024

21HTTT1 – Trường ĐH Khoa học Tự nhiên - VNUHCM

GV hướng dẫn: ThS. Tiết Gia Hồng, TS. Phạm Thị Bạch Huệ, ThS. Lương Vĩ Minh

Nhóm ATBM-A-01

✦

Thông tin nhóm ATBM-A-01

MSSV	Họ tên	Đóng góp
21127004	Trần Nguyễn An Phong	100%
21127135	Diệp Hữu Phúc	100%
21127149	Huỳnh Minh Quang	100%
21127296	Đặng Hà Huy	100%

Báo cáo có hỗ trợ Bookmark, hãy tận dụng để thuận tiện cho việc di chuyển

Mục lục

Contents

Chức năng theo tiêu chí chấm

PHÂN HỆ 1 – HỆ THỐNG WINDOWS FORM CHO NGƯỜI QUẢN TRỊ

STT	Nội dung	Giao diện	Script SQL	% hoàn thành	Điểm (%)
1.1 1.2 1.6	Xem danh sách tài khoản user. Xem thông tin quyền của (user/role) trên các đối tượng DL.	X	X	100	25
1.3	Tạo mới, xóa, sửa (user/role).	X	X	100	25
1.4	Cấp quyền (user, role, role2user, with grant, mức cột).	X	X	100	35
1.5	Thu hồi quyền user/role.	X	X	100	15

PHÂN HỆ 2 – THỰC HIỆN CHÍNH SÁCH BẢO MẬT TRÊN HỆ THỐNG

STT	Nội dung	Giao diện	Script SQL	% hoàn thành	Điểm (%)
YC1	**Giải pháp cấp quyền truy cập cho 6 chính sách CS#i:**				30
	CS#1 – Sử dụng RBAC.	X	X	100	
	CS#2 – Sử dụng RBAC, VPD.	X	X	100	
	CS#3 – Sử dụng RBAC, VPD.	X	X	100	
	CS#4 – Sử dụng RBAC, VPD.	X	X	100	
	CS#5 – Sử dụng DAC, RBAC, VPD.	X	X	100	
	CS#6 – Sử dụng VPD (bắt buộc), RBAC.	X	X	100	
YC2	**Cơ chế phát tán thông báo.**				25
	Số lượng chức năng đã hoàn tất (trên 8 yêu cầu của đồ án): 8.	X	X	100	
YC3	**Ghi nhật ký hệ thống.**				25
3.1	Kích hoạt/tắt việc ghi nhật ký.		X	100	
3.2	Standard Audit (Table/View/SP/Func).	X	X	100	
3.3	Fine-grained Audit (Dangky.diem, Nhansu. phucap).	X	X	100	
3.4	Xem dữ liệu nhật ký.	X	X	100	
YC4	**Sao lưu & Phục hồi dữ liệu.**				20
	Báo cáo tìm hiểu giải pháp sao lưu & phục hồi, đánh giá, kết luận.			100	
	Hiện thực trên ứng dụng.		X	100	

Chức năng và phân công

Cột **STT** có định dạng như sau,

- **Vị trí 1 (v1):** Mô tả **Loại** của chức năng.
  - **0 – Bổ trợ:** Chức năng không được yêu cầu trực tiếp bởi đồ án , nhưng cần phải được thực hiện mới có thể thỏa mãn các đề bài.
  - **Khác 0 – Mã phân hệ:** Chức năng được yêu cầu trực tiếp bởi đồ án.
- **Vị trí 2, 3 (v23):** Đánh số cho chức năng.
  - Nếu **v1 != 0** thì **v23** là số thứ tự của yêu cầu theo đúng sắp xếp trong đồ án.
- **Vị trí còn lại:** Đánh số cho yêu cầu phụ của chức năng (nếu có).

Dùng chung và bổ trợ

STT	Mô tả	MSSV	Hoàn thành

001	Xây dựng schema/database – createDB.sql.	21127135	100%
002	Phát sinh data – data.sql.	21127004	100%
003	Tạo admin và users – policies.sql.	21127149	100%
004	Thiết kế giao diện ứng dụng (WinForms). Mọi tab/form.	21127296	100%
005	Xóa (uninstall) database và các thành phần liên quan – cleanupDB.sql.	21127135	100%
006	Cho phép Tạo, Xóa, Sửa view, hỗ trợ cho phân quyền select trên cột ở mục **104c**. Admin/ManageView.	21127149	100%
007	Cho phép Admin có thể tạo thông báo trên giao diện – Admin/Announcement.	21127004	100%
008	Cho phép Admin xem Audit trail trên giao diện. Admin/InspectAudit.	21127296	100%

Phân hệ 1

STT	Mô tả	MSSV	Hoàn thành
101	Xem danh sách tài khoản người dùng. Tab ManageUser.	21127004	100%
102	Xem thông tin về quyền của user/role. Tab InspectPriv.	21127149	100%
1031	Cho phép Tạo mới, Xóa, Sửa user. Tab ManageUser.	21127004	100%
1032	Cho phép Tạo mới, Xóa, Sửa role. Tab ManageRole.	21127296	100%
104a1	Cấp quyền cho user/role, cấp role cho user. Tab GrantRoleTable.	21127135	100%
104a2	Cấp quyền hệ thống cho user/role. Tab GrantSysPriv.	21127004	100%
104b	Quá trình cấp quyền có tùy chọn WITH GRANT/ADMIN OPTION. Tab GrantSysPriv, GrantRoleTable.	(*)	100%
104c	Quyền select, update cho phép phân quyền tính đến mức cột, insert, delete thì không. Tab GrantRoleTable.	21127135	100%
105	Cho phép thu hồi quyền hạn từ user/role. Tab GrantSysPriv, Grant RoleTable.	(*)	100%
106	Cho phép kiểm tra quyền các chủ thể vừa cấp. Tab InspectPriv, ManageUser, GrantSysPriv, GrantRoleTable.	(*)	100%

(*): Đây là các chức năng được tích hợp vào tab, do người cài tab đó phụ trách. Ví dụ, với **106**, ở GrantRoleTable, khi grant hay revoke thì giao diện sẽ hiện dữ liệu liên quan đến chủ thể vừa được cập nhật. Tuy nhiên, vẫn có thể sang tab InspectPriv để kiểm tra độc lập.

Phân hệ 2

Đối với các yêu cầu có đòi hỏi chức năng giao diện, người thực hiện công việc phải cài đặt cả trên cơ sở dữ liệu lẫn ứng dụng. Khi đó, sẽ hợp tác với người phụ trách thiết kế ở mục **004**. Hơn nữa, với OLS, người được phân công phải khai báo, gán các nhãn cần thiết, và tạo dữ liệu (thông báo) mẫu cho nhiệm vụ.

STT	Mô tả	MSSV	Hoàn thành
211	CS#1: Nhân viên cơ bản (NV). policies.sql, Employee Dashboard.	21127149	100%
212	CS#2: Giảng viên (V). policies.sql, Lecturer Dashboard.	21127149	100%
213	CS#3: Giáo vụ (GV). policies.sql, Ministry Dashboard.	21127135	100%
214	CS#4: Trưởng đơn vị (TD). policies.sql, Unit Head Dashboard.	21127004	100%
215	CS#5: Trưởng khoa (TK). policies.sql, Department Head Dashboard.	21127004	100%
216	CS#6: Sinh viên (S), bắt buộc VPD. policies.sql, Student Dashboard.	21127296	100%
220	Tạo bảng thông báo, thiết lập OLS, LBAC_DBA, và khởi tạo hệ thống nhãn. ols.sql, policies.sql.	21127135	100%
22a	TK có thể đọc toàn bộ thông báo. ols.sql, DeptHead/Announcement.	21127135	100%
22b	TD cơ sở 2 có thể đọc toàn bộ thông báo dành cho TD không phân biệt vị trí địa lý. ols.sql, UnitHead/Announcement.	21127135	100%
22c	GV0001 có thể đọc toàn bộ thông báo cho GV. ols.sql, Ministry/Announcement	21127004	100%
22d	T1 được phát tán đến tất cả TD. ols.sql, UnitHead/Announcement.	21127004	100%
22e	T2 được phát tán đến S ngành HTTT cơ sở 1. ols.sql, Student/Announcement.	21127004	100%
22f	T3 được phát tán đến TD KHMT cơ sở 1. ols.sql, UnitHead/Announcement.	21127149	100%
22g	T4 được phát tán đến TD KHMT cơ sở 1 và 2.	21127149	100%

	ols.sql, UnitHead/Announcement.		
22h1	Thông báo được phát tán đến toàn S. ols.sql, Student/Announcement.	21127296	100%
22h2	NV cơ sở 1 chỉ xem thông báo không có chủ đề (không thuộc bộ môn cụ thể) của cơ sở 1. ols.sql, Employee/Announcement.	21127296	100%
22h3	Một NV cơ sở 2 (NV0006) được xem mọi thông báo cho NV (không giới hạn chủ đề và cơ sở). ols.sql, Employee/Announcement.	21127296	100%
231	Kích hoạt việc ghi nhật ký hệ thống. Bật Pure Unified Auditing, xóa Unified Audit Trail. audit.sql.	21127135	100%
232a	Standard audit với privilege, action, role, when và evaluate. Trừ SYS, bất kể tính thành công. audit.sql.	21127135	100%
232b	Standard audit với procedure. Chỉ với TK0001 khi không thành công. – audit.sql.	21127149	100%
233a	FGA cập nhật DANGKY tại trường liên quan đến điểm số nhưng không phải V. – audit.sql	21127296	100%
233b	FGA đọc trường PHUCAP của người khác trên NHANSU. – audit.sql.	21127004	100%
234	Kiểm tra (đọc xuất) dữ liệu nhật ký hệ thống. audit.sql.	21127149	100%
241a	Tìm hiểu Cold, Hot backup và recovery.	21127149	100%
241b	Tìm hiểu Backup và recovery với RMAN.	21127296	100%
242a	Cài đặt Full backup và recovery.	21127004	100%
242b	Cài đặt Incremental backup và recovery.	21127135	100%
243a	Đánh giá Cold backup và recovery.	21127149	100%
243b	Đánh giá Hot, Full backup và recovery.	21127296	100%
243c	Đánh giá Incremental backup và recovery.	21127135	100%
244	Kết luận.	21127004	100%

Thông tin chung

Thư mục **ATBM-2024-ATBM-A-01** sau khi giải nén sẽ có các thành phần sau:

- **ISS_QLNoiBo:** Chứa source code ứng dụng WinForms.
- **QLNB_Release:** Chứa bản Release của ứng dụng, chạy file **ISS_QLNoiBo.exe**.
- **SQL:** Chứa script/source code của Oracle/sql.
- ATBM-A-01-PH1-Report.pdf.

Cấu trúc thư mục SQL

Nhằm phục vụ cho sự dễ dàng trong việc quản lý, hợp tác, và cải tiến trong quá trình cài đặt cơ sở dữ liệu, source code SQL được chia thành nhiều thành phần riêng lẻ nhau.

- **audit.sql** – Cài đặt ghi nhật ký hệ thống theo yêu cầu 3 của Phân hệ 2.
- **bkp&rec.sql** – Cài đặt sao lưu và phục hồi dữ liệu theo yêu cầu 4 của Phân hệ 2.
- **cleanupDB.sql** – Xóa/gỡ cài đặt database và các thành phần liên quan.
- **createDB.sql** – Khởi tạo/xây dựng schema/database.
- **data.sql** – Phát sinh data cho database.
- **ols.sql** – Cài đặt OLS theo Yêu cầu 2 của Phân hệ 2.
- **policies.sql** – Tạo users, và cài đặt các chính sách theo Yêu cầu 1 của Phân hệ 2.

Lưu ý

Nhóm sử dụng Oracle 21c Express, CDB và PDB được tạo sẵn trong lúc cài đặt, với các trường chi tiết như sau,

- **Database host name:** localhost,
- **Port number:** 1521,
- **Service name:** XEPDB1.

Như vậy, một lệnh kết nối đầy đủ vào PDB, ví dụ tài khoản sys, sẽ mang định dạng,

```
CONN SYS/password@LOCALHOST:1521/XEPDB1 AS SYSDBA;
```

Đồng thời, database được tạo trên schema/user **A01_QLNOIBO**, và tài khoản admin duy nhất là **AD0001** với mật khẩu **123**. Các user khác đều dùng mật khẩu **123** với tên tài khoản là mã (khóa chính) có thể được tìm thấy trong **data.sql**.

Cài đặt

- ***SYS.XE*** Làm abc – Thực hiện abc với tài khoản **sys** và trên **root** container.
- ***SYS.XEPB1*** Làm abc – Thực hiện abc với tài khoản **sys** và service/PDB **XEPBD1**.

Xây dựng database

Trước hết, cần thực hiện Kích hoạt OLS và Pure Unified Auditing. Sau đó, vào thư mục **SQL**, với ***SYS.XEPB1***, chạy các script theo thứ tự sau,

createDB.sql → data.sql → policies.sql → ols.sql → audit.sql.

Xóa/gỡ cài đặt database

SYS.XEPB1 Chạy **cleanupDB.sql**.

Nên đóng mọi connection trước khi chạy để tránh các lỗi như, **ORA-01940: cannot DROP a user that is currently logged in**. Trong quá trình thực thi, do phải **DROP** user/admin **AD0001**, nên sẽ cần chuyển qua tài khoản **sys** và yêu cầu nhập mật khẩu.

Kích hoạt OLS

1. ***SYS.XEPB1*** Kiểm tra trạng thái của OLS, nếu đều là **TRUE** thì không thực hiện tiếp.

```
SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Oracle Label Security';
SELECT STATUS FROM DBA_OLS_STATUS WHERE NAME = 'OLS_CONFIGURE_STATUS';
```

2. ***SYS.XEPB1*** Cấu hình và khởi động OLS.

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

3. Khởi động lại IDE/server/service (Tắt và bật lại chương trình, ví dụ sqldeveloper).

4. ***SYS.XE*** Mở khóa tài khoản admin OLS **LBACSYS**.

```
ALTER USER LBACSYS ACCOUNT UNLOCK IDENTIFIED BY your_password;
```

Kích hoạt Pure Unified Auditing (PUA)

1. Kiểm tra PUA đã được bật, nếu là **TRUE** thì không thực hiện tiếp.

```
SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Unified Auditing';
```

2. Dùng Windows Services để dừng **service** và **listener** của Oracle.

   - Windows + R → services.msc.
   - Tên sẽ mang định dạng tương tự **OracleServiceXE** và **OracleOraDB21Home1TNS-Listener**.

3. Dùng Windows File Explorer để tìm đường dẫn đến tệp **orauniaud**. Ví dụ,

...\Oracle21c\dbhomeXE\bin

4. Đổi tên tệp thành **orauniaud.dll**.

   - Ban đầu, tệp có thể mang các tên sau, **orauniaud.dll.option**, **orauniaud.dll.dbl**, hoặc một trong hai nhưng thêm số phiên bản vào cuối, như, **orauniaud12.dll.option**.
   - Có thể sẽ phải bật **Show File name extensions**, để thấy được phần định dạng đuôi.

5. Khởi động lại **service** và **listener** của Oracle.

Dùng ứng dụng ISS_QLNoiBo

1. Hoàn thành Xây dựng database.

2. Vào thư mục **QLNoiBo_Release**, và chạy **ISS_QLNoiBo.exe**.

Liên kết Oracle và WinForms

1. Vào thư mục **ISS_QLNoiBo** và mở **ISS_QLNoiBo.sln** với Visual Studio 2022.

2. Thay đổi giá trị các biến trong **Others/OracleConfig.cs** để phù hợp với môi trường.

3. Trên thanh công cụ, chọn **Project** → **Manage NuGet Packages...**.

4. Dùng tab **Browse**, tìm **oracle**, và cài đặt **Oracle.ManagedDataAccess.Core**.

5. Thử Build và chạy chương trình, đăng nhập bằng AD0001 và mật khẩu 123. Nếu thất bại do exception, tiếp tục bước 6.

6. Dùng Windows File Explorer để tìm đường dẫn đến **Oracle.DataAccess.dll**. Ví dụ,

...\Oracle21c\dbhomeXE\ODP.NET\bin\4

7. Quay lại VS2022, bên thanh **Solution Explorer**, chuột phải vào **ISS_QLNoiBo**, ngay bên dưới **Solution 'ISS_QLNoiBo' (1 of 1 project)**.

8. Chọn **Add** → **Project Reference...**.

9. Dùng tab **Browse**, và chọn **Browse...** ở góc dưới phải của hộp thoại.

10. Di chuyển tới đường dẫn đã tìm được ở bước 6.

11. Chọn **Oracle.DataAccess.dll** và **Add**.

Phân hệ 1: Ứng dụng quản trị CSDL Oracle

Mọi yêu cầu được thỏa mãn và cài đặt hoàn thiện trên ứng dụng **ISS_QLNoiBo**, thông qua giao diện Admin. Mở ứng dụng ISS_QLNoiBo, và truy cập với tài khoản,

AD0001 và mật khẩu là **123**.

Hoặc, được giới thiệu ở **Demo ATBM CSDL trong HTTT 2024 - Phân hệ 1 - ATBM-A-01**,

- https://youtu.be/Q9bQsbvNSeU

Phân hệ 2: Ứng dụng quản lý dữ liệu nội bộ

Mọi yêu cầu đòi hỏi chức năng minh họa đều được cài đặt đầy đủ trên ứng dụng. Tìm tài khoản cần thiết với mật khẩu **123** trong **data.sql** để truy cập giao diện tương ứng và thực hiện các chức năng mong muốn.

Đặc biệt ở Yêu cầu 2, không phải tài khoản nào cũng có nhãn, mà chỉ những người dùng liên quan, và phục vụ các câu hỏi mới được gán. Do đó, nên dùng tài khoản thuộc các diện được đề cập đến, hoặc được gợi ý rõ trong **ols.sql**.

Yêu cầu 1: Cấp quyền truy cập

Mục này nhóm chỉ giải thích cho các điểm đáng đề cập, sẽ không bàn về các yêu cầu mang tính "tầm thường" trong việc cài đặt, ví dụ,

"Xem thông tin của tất cả SINHVIEN, DONVI, HOCPHAN, KHMO."

Và, yếu tố **người dùng vai trò X có quyền truy cập như một người dùng vai trò Y**, được giải quyết đơn giản là gán thêm role RL_A01_Y (bên cạnh RL_A01_X) cho người dùng vai trò X.

CS#1: Nhân viên cơ bản

RL_A01_NV

Xem dòng dữ liệu của chính mình trong quan hệ NHANSU, có thể chỉnh sửa số điện thoại (ĐT) của chính mình (nếu số điện thoại có thay đổi).

Tạo view V_NHANSU_NV với điều kiện MANV phải bằng người dùng hiện tại, và sử dụng WITH CHECK OPTION để phục vụ UPDATE. Gán SELECT, UPDATE(DT) trên view này cho role.

CS#2: Giảng viên

RL_A01_V

Xem dữ liệu phân công giảng dạy liên quan đến bản thân mình (PHANCONG).

Nhận thấy ở CS#4 cũng có xem trên PHANCONG, nên thực hiện tạo chính sách **POL_PHANCONG_SEL** quản lý SELECT trên bảng A01_QLNOIBO.PHANCONG. Tạo policy function **PFN_PHANCONG_SEL** hiện tại chỉ kiểm tra nếu là GIANGVIEN (mã user có V) thì trả về vị ngữ MAGV = user tương ứng, ngược lại trả về NULL (để không ảnh hưởng các người dùng vai trò khác). Gán SELECT trên PHANCONG cho role.

Xem dữ liệu trên quan hệ DANGKY liên quan đến các lớp học phần mà giảng viên được phân công giảng dạy.

Tạo view V_DANGKY_V với điều kiện MAGV phải bằng người dùng hiện tại, và sử dụng WITH CHECK OPTION để phục vụ cho yêu cầu tiếp theo. Gán SELECT trên view này cho role.

Cập nhật dữ liệu tại các trường liên quan điểm số (trong quan hệ DANGKY) của các sinh viên có tham gia lớp học phần mà giảng viên đó được phân công giảng dạy. Các trường liên quan điểm số bao gồm: DIEMTH, DIEMQT, DIEMCK, DIEMTK.

Gán thêm UPDATE(DIEMTH,DIEMQT,DIEMCK,DIEMTK) trên V_DANGKY_V đã tạo cho role.

CS#3: Giáo vụ

RL_A01_GV

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.

Nhận thấy ở CS#4 và CS#5 cũng có các thao tác thay đổi dữ liệu trên PHANCONG, nên thực hiện tạo chính sách **POL_PHANCONG_EDIT**, hiện tại chỉ quản lý UPDATE, trên bảng A01_QLNOIBO.PHANCONG, và bật UPDATE_CHECK là TRUE.

Tạo policy function **PFN_PHANCONG_EDIT** hiện tại chỉ kiểm tra nếu là GIAOVU (mã user có GV) thì trả về vị ngữ xác định (đúng/sai) xem MAHP có trong danh sách các học phần do Văn phòng khoa quản lý (MADV = 1). Ngược lại trả về NULL (để không ảnh hưởng các người vai trò khác). Gán SELECT, UPDATE trên PHANCONG cho role.

- Mặc dù bảng PHANCONG đã có một policy giới hạn SELECT, thiết lập trước đó ở CS#2, nhưng do trả về NULL nếu không thỏa điều kiện, nên GIAOVU vẫn có thể xem toàn bộ PHANCONG mà không bị ảnh hưởng.

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 (thỏa điều kiện có thể hiệu chỉnh).

Nhận thấy ở CS#6 cũng có INSERT, DELETE trên DANGKY, nên thực hiện tạo chính sách **POL_DANGKY_INS_DEL** quản lý 2 thao tác đó, trên bảng A01_QLNOIBO.DANGKY, và bật UPDATE_CHECK là TRUE để phục vụ INSERT.

Tạo policy function **PFN_DANGKY_INS_DEL** hiện tại chỉ trả về vị ngữ, sau khi tính toán các giá trị cần thiết, bao gồm,

- HK = (1 nếu tháng hiện tại nhỏ hơn 5, 2 nếu nhỏ hơn 9, và còn lại là 3) AND
- NAM = (Năm hiện tại) AND
- 0 < DIFF <= 14. DIFF là số ngày đã qua kể từ ngày 01 của tháng (1 nếu tháng hiện tại nhỏ hơn 5, 5 nếu nhỏ hơn 9, và còn lại là 9).

Gán SELECT, INSERT, DELETE trên DANGKY cho role.

- Tuy SELECT không được yêu cầu, nhưng tuân theo giá trị **SQL92_SECURITY** mặc định là TRUE, nếu muốn sử dụng được mệnh đề WHERE khi UPDATE hoặc DELETE thì phải có cả quyền SELECT.

CS#4: Trưởng đơn vị

RL_A01_TD

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.

Cập nhật policy function **PFN_PHANCONG_EDIT** đã tạo ở CS#3, kiểm tra thêm nếu là TRUONGDONVI (mã user có TD) thì trả về vị ngữ xác định (đúng/sai) xem MAHP có trong danh

sách các học phần được phụ trách bởi đơn vị mà người dùng này làm trưởng (MADV = MADV của user).

DROP **POL_PHANCONG_EDIT** và tạo lại để cập nhật tầm quản lý bao gồm INSERT, DELETE, UPDATE. Gán INSERT, DELETE, UPDATE trên PHANCONG cho role.

Xem dữ liệu phân công giảng dạy của các giảng viên thuộc các đơn vị mình làm trưởng.

Cập nhật policy function **PFN_PHANCONG_SEL** đã tạo ở CS#2, kiểm tra thêm nếu là TRUONGDONVI (mã user có TD) thì trả về vị ngữ xác định (đúng/sai) xem MAGV có trong danh sách các giảng viên thuộc đơn vị mà người dùng làm trưởng (MADV = MADV của user). Gán SELECT trên PHANCONG cho role.

CS#5: Trưởng khoa

TK0001

Thêm, Xóa, Cập nhật dữ liệu trên quan hệ PHANCONG đối với các học phần quản lý bởi đơn vị "Văn phòng khoa".

Cập nhật policy function **PFN_PHANCONG_EDIT** mới ở CS#4, thêm vào phần kiểm tra trước đó ở CS#3 trở thành nếu là GIAOVU (mã user có GV) hoặc TRUONGKHOA (mã user có TK). Gán SELECT, INSERT, DELETE, UPDATE trên PHANCONG cho TK0001.

CS#6: Sinh viên – Bắt buộc dùng VPD

RL_A01_S

Trên quan hệ SINHVIEN, sinh viên chỉ được xem thông tin của chính mình, được Chỉnh sửa thông tin địa chỉ (DCHI) và số điện thoại liên lạc (DT) của chính sinh viên.

Tạo chính sách **POL_SINHVIEN** quản lý SELECT, UPDATE trên bảng A01_QLNOIBO.SINH-VIEN, và bật UPDATE_CHECK là TRUE. Tạo policy function **PFN_SINHVIEN** kiểm tra nếu là SINHVIEN (mã user có S) thì trả về vị ngữ MASV = user tương ứng, ngược lại trả về NULL (để không ảnh hưởng các người dùng vai trò khác). Gán SELECT, UPDATE(DCHI,DT) trên SINHVIEN cho role.

Xem danh sách tất cả học phần (HOCPHAN), kế hoạch mở môn (KHMO) của chương trình đào tạo mà sinh viên đang theo học.

Tạo chính sách **POL_KHMO** quản lý SELECT trên bảng A01_QLNOIBO.KHMO. Tạo policy function **PFN_KHMO** kiểm tra nếu là SINHVIEN (mã user có S) thì trả về vị ngữ xác định (đúng/sai) xem MACT có bằng chương trình của người dùng (MACT = MACT của user). Ngược lại trả về NULL (để không ảnh hưởng các người dùng vai trò khác). Gán SELECT trên HOCPHAN, KHMO cho role.

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).

Cập nhật policy function **PFN_DANGKY_INS_DEL** đã tạo ở CS#3, kiểm tra thêm nếu là SINHVIEN (mã user có S) thì kết thêm vào đầu vị ngữ trước đó giá trị,

MASV = (User hiện tại) AND

Gán INSERT, DELETE trên DANGKY cho role.

Sinh viên không được chỉnh sửa trên các trường liên quan đến điểm.

Cập nhật policy function **PFN_DANGKY_INS_DEL** mới đây, kết thêm vào đầu vị ngữ giá trị,

DIEMTH IS NULL AND DIEMQT IS NULL AND DIEMCK IS NULL AND DIEMTK IS NULL AND

Sinh viên được Xem tất cả thông tin trên quan hệ DANGKY tại các dòng dữ liệu liên quan đến chính sinh viên.

Tạo chính sách **POL_DANGKY_SEL** quản lý SELECT trên bảng A01_QLNOIBO.DANGKY. Sử dụng chung policy function **PFN_SINHVIEN** đã tạo cho bảng SINHVIEN. Gán SELECT trên DANGKY cho role.

Yêu cầu 2: Vận dụng mô hình điều khiển truy cập OLS

Với **LBAC_DBA** và các thành phần cần thiết của **LBACSYS** gán trên **AD0001**, hệ thống nhãn được cài đặt với chính sách **POL_A01_THONGBAO** và cột nhãn **LAB_TB**. Bao gồm **6 levels**, **6 compartments**, và **3 groups**. Chi tiết như sau,

STT	Level : Num	Compartment : Num	Group : Num : Parent
1	TK : 9000	HTTT : 200	DHX : 1000
2	TD : 8000	CNPM : 300	CS1 : 1100 : DHX
3	V : 7000	KHMT : 400	CS2 : 1200 : DHX
4	GV : 6000	CNTT : 500	
5	NV : 5000	TGMT : 600	
6	S : 4000	MMT : 700	

Chú thích: TK (TRUONGKHOA), TD (TRUONGDONVI), V (GIANGVIEN), GV (GIAOVU), NV (NHANVIEN), S (SINHVIEN), DHX (DAIHOCX).

Bên cạnh đó, khi tạo/khai báo nhãn, nhóm cũng đề ra công thức để giữ tính đồng bộ,

```
LABEL_TAG = LEVEL_NUM + SUM(COMP_NUM)
          + (GROUP_NUM(DHX) * 10 IF DHX ELSE SUM(GROUP_NUM)).
```

Do vậy, nhãn cao nhất (không giới hạn truy cập) sẽ là,

```
TK:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:DHX và LABEL_TAG = 21700.
```

Hơn nữa, khi xảy ra trùng lắp tag, thực hiện cộng 10 vào nhãn thấp hơn về ngữ nghĩa. Ví dụ,

```
GV:HTTT,CNPM,KHMT,CNTT,TGMT,MTT:DHX và LABEL_TAG = 18700,
TD:MMT:DHX                          và LABEL_TAG = 18710.
```

Thực hiện yêu cầu

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

`TK:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:DHX và LABEL_TAG = 21700.`

Nhãn được gán cho TK0001 và AD0001.

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.

Gọi **BM** là bộ môn của một Trưởng đơn vị cơ sở 2 bất kỳ, khi đó nhãn cho người này là,

`TD:BM:DHX và LABEL_TAG = 8000 + COMP_NUM(BM) + 1000 * 10.`

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

`GV:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:DHX và LABEL_TAG = 18700.`

Nhãn được gán cho GV0001.

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

`TD và LABEL_TAG = 8000.`

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

`S:HTTT:CS1 và LABEL_TAG = 5300.`

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

`TD:KHMT:CS1 và LABEL_TAG = 9500.`

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.

`TD:KHMT:CS1,CS2 và LABEL_TAG = 10700.`

h) Em hãy cho thêm 3 chính sách phát tán dòng dữ liệu nữa trên mô hình OLS đã cài đặt.

h1) Cho biết nhãn của thông báo phát tán đến toàn thể Sinh viên.

`S và LABEL_TAG = 4000.`

h2) Gán nhãn cho Nhân viên Cơ sở 1 chỉ được xem thông báo không có chủ đề (không thuộc về bộ môn cụ thể) của Cơ sở 1.

`NV::CS1 và LABEL_TAG = 6100.`

h3) Gán nhãn cho 01 Nhân viên Cơ sở 2 có thể xem mọi thông báo cho Nhân viên (không giới hạn chủ đề và cơ sở).

`NV:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:DHX và LABEL_TAG = 17700.`

Nhãn được gán cho NV0006.

Yêu cầu 3: Ghi nhật ký hệ thống

AD0001 được gán role **AUDIT_ADMIN**.

1. Kích hoạt việc ghi nhật ký hệ thống

Nhóm chọn sử dụng **Pure Unified Auditing**, với cài đặt đã được đề cập ở mục Kích hoạt PUA. Đồng thời, nên làm sạch (purge/clean) **Unified Audit Trail** nếu cần thiết, để dễ dàng hơn khi quan sát kết quả các yêu cầu.

```
DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL(
    AUDIT_TRAIL_TYPE  => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
    USE_LAST_ARCH_TIMESTAMP  => FALSE);
```

2. Thực hiện ghi nhật ký hệ thống dùng Standard Audit

"Theo dõi hành vi của những user nào trên những đối tượng cụ thể, trên các đối tượng khác nhau (table, view, stored procedure, function), hay chỉ định theo dõi các hành vi hiện thành công hay không thành công."

SA với privilege, action, role, mệnh đề when và evaluate. Trên mọi user trừ SYS, bất kể tính thành công.

Tạo và AUDIT chính sách ghi vết **AUDPOL_GENERAL** với chi tiết,

- **Privileges:** UPDATE ANY TABLE, DROP ANY TABLE, DROP USER, DROP ANY ROLE.
- **Actions:**
  - Tất cả thao tác trên A01_QLNOIBO.NHANSU,
  - INSERT, DELETE, UPDATE trên A01_QLNOIBO.SINHVIEN,
  - INSERT, DELETE, UPDATE trên A01_QLNOIBO.PHANCONG,
  - UPDATE trên A01_QLNOIBO.V_DANGKY_V,
  - ALTER USER, DROP USER, DROP ROLE.
- **Roles:** RL_A01_GV.

Áp dụng khi thỏa điều kiện người dùng hiện tại **khác SYS**, được xác định vào **đầu mỗi phiên**.

Mọi privilege và action nhóm đều tạo được trường hợp thử nghiệm, với ngoại lệ là **role**. Bởi vì trên role, Oracle chỉ ghi vết các system privilege được gán cho role (tức chỉ những query cần tới quyền hệ thống mới bị audit). Tuy nhiên, các role nhóm tạo không được cấp bất cứ quyền hệ thống nào. Do đó, ở đây chỉ để thể hiện là có thể tạo ghi vết trên role.

- Nhóm có thực hiện được ghi vết trên role **DBA**, nhưng vì nó bao gồm quyền **CREATE SESSION** và **LOGON**, sẽ làm đầy audit trail và gây khó khăn trong việc theo dõi các hành vi khác.

SA với procedure. Trên TK0001, chỉ khi không thành công.

Tạo **AUDPOL_GENERAL_USP** trên 2 **action** là,

- EXECUTE ON A01_QLNOIBO.USP_CREATE_USERS,
- EXECUTE ON A01_QLNOIBO.USP_ASSIGN_ROLE.

Thực hiện AUDIT chính sách trên **chỉ với TK0001**, mỗi khi không thành công.

Lưu ý, khi ghi vết trên EXECUTE thủ tục (hoặc hàm), thì Oracle chỉ quan tâm việc có tìm được thủ tục và quyền tương ứng, để đánh giá sự thành công của thao tác. Tức, với WHENEVER NOT SUCCESSFUL, thì chỉ lỗi không tồn tại và thiếu quyền được ghi vết, các vấn đề xảy ra bên trong thân thủ tục sẽ không được công nhận.

Vì TK0001 không có quyền thực hiện 2 thủ tục trên, nên ngữ cảnh sẽ sinh ra 2 lỗi không tồn tại, cụ thể,

```
PLS-00201: identifier 'A01_QLNOIBO.USP_CREATE_USERS' must be declared,
PLS-00201: identifier 'A01_QLNOIBO.USP_ASSIGN_ROLE' must be declared.
```

Sau đó, script vẫn chạy bình thường, và như đã đề cập, 2 hành động này sẽ được ghi vết do không thành công.

3. Thực hiện Fine-grained Audit các tình huống và tạo ngữ cảnh ghi vết

Các chính sách FGA được thiết lập với thủ tục **ADD_POLICY** thuộc gói **DBMS_FGA**.

a) Hành vi Cập nhật trên quan hệ DANGKY tại các trường liên quan đến điểm số nhưng người đó không thuộc vai trò Giảng viên.

Tạo **AUDPOL_DANGKY_UPD_DIEM_NOT_V** trên bảng **A01_QLNOIBO.DANGKY**. Kiểm tra **UPDATE** trên các cột **DIEMTH,DIEMQT,DIEMCK,DIEMTK**, với điều kiện mã người dùng hiện tại **không bắt đầu bằng V**.

Hệ thống chỉ có **AD0001** và GIANGVIEN (V) được UPDATE các cột trên, tuy nhiên V chỉ được thực hiện thông qua view V_DANGKY_V. Do vậy, sẽ tạm thời cấp cho **V20062** quyền **SELECT** và **UPDATE(DIEMTH)**. Khi đó, nếu cùng thực hiện UPDATE(DIEMTH) của các dòng DANGKY có MASV = S10081 và MAGV = V20062, thì chỉ **AD0001** sẽ bị ghi vết.

b) Hành vi của người dùng này có thể đọc trên trường PHUCAP của người khác ở quan hệ NHANSU.

Tạo **AUDPOL_NHANSU_SEL_PHUCAP** trên bảng **A01_QLNOIBO.NHANSU**. Kiểm tra **SELECT** trên cột **PHUCAP**, với điều kiện MANV của các dòng truy cập **khác mã người dùng** hiện tại.

Hệ thống chỉ có **AD0001** và **TK0001** được SELECT trên toàn bảng NHANSU. Do đó, nếu một trong hai thực hiện SELECT có bao gồm cột PHUCAP, vào các dòng không thuộc về mình, thì sẽ bị ghi vết.

4. Kiểm tra (đọc xuất) dữ liệu nhật ký hệ thống

Nhật ký hệ thống có thể được tra cứu thông qua bảng **UNIFIED_AUDIT_TRAIL**, nhóm chọn những cột cần thiết, và sắp xếp theo thời gian,

```
SELECT AUDIT_TYPE, DBUSERNAME, EVENT_TIMESTAMP, ACTION_NAME, OBJECT_SCHEMA,
    OBJECT_NAME, SQL_TEXT, SYSTEM_PRIVILEGE_USED, OBJECT_PRIVILEGES,
    TARGET_USER, CURRENT_USER, UNIFIED_AUDIT_POLICIES,
    FGA_POLICY_NAME FROM UNIFIED_AUDIT_TRAIL
ORDER BY EVENT_TIMESTAMP;
```

Đối với 2 yêu cầu trước, để thu chỉ những kết quả đáng quan tâm, thực hiện lọc thêm trên các cột với giá trị như sau,

- **AUDIT_TYPE:** Standard hoặc FineGrainedAudit.
- **UNIFIED_AUDIT_POLICIES:** Chứa AUDPOL_GENERAL hoặc AUD_GENERAL_USP.
- **OBJECT_NAME:** DANGKY hoặc NHANSU.
- **DBUSERNAME:** Khác SYS và A01_QLNOIBO.

Yêu cầu 4: Sao lưu và phục hồi dữ liệu

Một số khái niệm chung cần điểm qua,

- **Logical backup:** Sao lưu các bảng trong cơ sở dữ liệu, với các tiện ích mà Oracle có hỗ trợ như, Export và Import, Datapump,....
- **Physical backup:** Sao lưu các tệp tin ở mức hệ điều hành, ví dụ, thư mục ORACLE HOME, parameter files, datafiles, control files, hay log files,....
- **Consistent backup:** Được thực hiện khi cơ sở dữ liệu đã được SHUTDOWN toàn vẹn, và không cần thực hiện recovery sau khi phục hồi.
- **Inconsistent backup:** Được thực hiện khi cơ sở dữ liệu vẫn đang hoạt động, hoặc SHUTDOWN một cách bất thường, nên sẽ cần recovery.
- **Restore:** Khôi phục cơ sở dữ liệu qua sao chép dữ liệu từ backup.
- **Recover:** Phục hồi cơ sở dữ liệu, sau restore, bằng cách thực hiện lại các hành động ghi nhận trong log files.
- **Incomplete recovery, hay Point-in-time recovery (PITR):** Cơ sở dữ liệu chỉ có thể quay về thời điểm mà backup được thực hiện, không thể về đúng khi xảy ra sự cố.
- **Complete recovery:** Áp dụng log files (recover) sau khi thực hiện PITR để đưa cơ sở dữ liệu về thời điểm gần nhất, hoặc chính xác lúc thất bại.

Bên cạnh đó, khi xảy ra sự cố, trong quá trình khôi phục sẽ cần thỏa mãn được các yêu cầu dữ liệu ở từng trạng thái nhất định, để cơ sở dữ liệu có thể hoạt động bình thường, như sau,

- **No mount:** Parameter files.
- **Mount:** Control files.
- **Open:** Datafiles và log files.

Cold backup và recovery

Cold backup được thực hiện khi cơ sở dữ liệu đã dừng hoạt động hoàn toàn (**SHUTDOWN** hoặc OFFLINE), có thể làm thủ công với các thao tác sao chép ở mức hệ điều hành. Nó đảm bảo sẽ không có dữ liệu đang được xử lý khi thực hiện backup, qua đó mang tính **consistent** và không cần sao lưu log files (do không phải recover sau khôi phục). Dù vậy, nó đòi hỏi một khoảng thời gian ngừng hoạt động (downtime) mà người dùng không thể truy cập.

Mặc dù cold backup có thể được lưu ở cùng ổ đĩa trên server chứa database, nhưng để tránh mất cả hệ thống và backup khi gặp sự cố, thì nó nên được đặt vào một vị trí riêng biệt (ổ cứng hoặc server khác).

Khôi phục bằng cách thỏa mãn yêu cầu dữ liệu để đưa hệ thống qua từng trạng thái **No mount → Mount → Open**. Do bản chất consistent, mọi transaction thực hiện sau backup sẽ không tồn tại, đồng thời, cơ sở dữ liệu chỉ được đưa về thời điểm backup, không thể ngay chính xác lúc phát sinh sự cố. Đây là tính chất của **Incomplete recovery**.

Ưu điểm: Nhờ vào việc thực hiện offline, cold backup tránh được những bất cẩn về ghi đè hay xóa dữ liệu, dù cố ý hay vô ý, giúp **tăng cường tính bảo mật**. Offline cũng tối thiểu các vấn đề về điện áp, và truyền dữ liệu, qua đó **giảm được khả năng hư hỏng dữ liệu**. Và, do không người dùng nào truy cập được trong quá trình backup, nên **tính nhất quán dữ liệu là chắc chắn**, bảo toàn nguyên vẹn những tệp được sao lưu.

Nhược điểm: Trên thực tế, cold backup không được áp dụng phổ biến, do **đòi hỏi phải dừng tất cả hoạt động** trong cơ sở dữ liệu xuyên suốt quá trình backup, đặc biệt không phù hợp với các hệ thống yêu cầu làm việc liên tục 24/7. Hơn nữa, nếu cần cold backup sang một thiết bị lưu trữ ngoài (để thuận tiện quản lý dung lượng) thì sẽ phải **kéo dài downtime**.

Nếu cold backup chỉ được dùng độc lập và thiếu thường xuyên, thì **không thể cung cấp được các mốc thời gian linh hoạt** khi cần khôi phục (thời điểm backup quá xa so với sự cố).

Hot backup và recovery

Hot backup được thực hiện khi cơ sở dữ liệu vẫn đang hoạt động bình thường, có thể được làm thủ công nhưng yêu cầu đưa hệ thống vào chế độ **ARCHIVELOG**. Bên cạnh để sao lưu cả log files, trạng thái này tối ưu việc phục hồi hoàn toàn các transaction trong khung thời gian từ khi backup đến recover.

Vì không đòi hỏi downtime và cho phép người dùng truy cập bình thường, nên nếu dữ liệu bị tác động trong quá trình backup, sẽ dẫn đến tính **inconsistent**, và cần recover sau khôi phục. Đồng thời, backup xảy ra khi vẫn vận hành database nên sẽ cần thêm tài nguyên máy, có thể gây ảnh hưởng hiệu suất hệ thống, và việc backup nhiều dữ liệu hơn cũng dẫn đến sự tăng lên trong dung lượng lưu trữ.

- Một biến thể của hot backup là **Warm backup**, khi đó chỉ bật hệ thống mà không được thực hiện bất kỳ công việc gì, hoặc chỉ bật khi cần lấy cập nhật từ cơ sở dữ liệu đang được backup.

Tương tự cold backup, cũng đưa hệ thống qua từng trạng thái yêu cầu dữ liệu nhất định để **khôi phục**. Tuy nhiên, trước khi Open cần thực hiện **phục hồi** dùng log files, vậy quy trình trở thành **No mount → Mount → Recover → Open**. Nếu trước đó đã có khôi phục control files, thì cũng cần phục hồi database với backup controlfile, và open resetlogs, để đặt lại trình tự lưu trữ log đếm từ 1. Hoàn thiện toàn bộ quy trình sẽ đạt được **Complete recovery**.

Ưu điểm: Hệ thống vẫn **hoạt động bình thường xuyên suốt quá trình backup**, đặc biệt quan trọng với các môi trường đa người dùng và làm việc liên tục. **Linh hoạt, nhanh chóng, và toàn vẹn trong khôi phục dữ liệu**, nhờ cắt giảm việc ngừng/khởi động database, và sao lưu log files mang lại nhiều tùy chọn về mốc thời gian cần phục hồi.

Nhược điểm: Hot backup **không được thiết kế để chịu lỗi tốt**, nếu gặp vấn đề trong quá trình backup, thì với một phần dữ liệu đã sao lưu cũng không thực hiện được PITR. Do đó, **nặng về chi phí vận hành, bảo trì, và đòi hỏi sự tỉ mỉ**, khi mà cơ sở dữ liệu vẫn có thể bị tác động trong lúc backup.

Backup và recovery với RMAN

Recovery Manager (RMAN) là một công cụ độc lập về nền tảng, cung cấp bởi Oracle để tự động và đơn giản hóa quá trình sao lưu và phục hồi cơ sở dữ liệu. Hiện tại, ngoài việc giải quyết được nhiều khó khăn và rủi ro khi thực hiện phương pháp thủ công với các thao tác copy/paste mức OS, RMAN còn đem đến một số ưu điểm vượt trội sau,

- **Block-level backup:** Bằng cách biểu diễn dữ liệu thành khối, sao lưu trở nên hiệu quả và nhanh hơn, do chỉ lấy backup của những khối được sử dụng thay vì toàn database.
- **Parallelism:** Có thể cấu hình RMAN để phân chia nhiều tiến trình song song, giúp tăng tốc backup.
- **Corruption detection:** RMAN hỗ trợ phát hiện và sửa chữa các khối dữ liệu hỏng.
- **Recovery catalog:** Bên cạnh lưu trữ metadata của backup vào control files, RMAN cũng cho phép thiết lập lưu thông tin ở nhiều cơ sở dữ liệu trên các server khác nhau, được quản lý bởi Recovery catalog.

Full backup và recovery

Full backup, về bản chất và hoạt động hoàn toàn tương đồng với **Hot backup**, kể cả các **ưu và nhược điểm**. Hơn nữa, cũng có thể thực hiện **Cold backup** với chức năng này ở chế độ NOARCHIVELOG, bằng cách đưa database về MOUNT trước khi backup.

Về **khôi phục**, có thể thực hiện cả **Complete recovery** (CR) và **PITR**, yêu cầu phải đưa cơ sở dữ liệu về MOUNT trước. Với **CR**, chỉ việc RESTORE database từ Full backup mong muốn, RECOVER để áp log files và OPEN. Còn **PITR**, cần chỉ định thêm thời điểm muốn khôi

phục và phục hồi tới với SET UNTIL, có thể dùng SCN (mốc đánh dấu một thời điểm trong Oracle) hoặc thời gian cụ thể, cuối cùng OPEN RESETLOGS để đặt lại thứ tự ghi log.

Incremental backup và recovery

Một chiến lược **Incremental backup** (IncB) bao gồm,

- **Một Level 0 IncB:** Full backup sao lưu tất cả những block dữ liệu có sử dụng.
- **Không hoặc nhiều Level 1 IncB**: Mỗi level 1 IncB, với mặc định là **Differential**, thì chỉ sao lưu những khối bị thay đổi kể từ lần IncB gần nhất. Hoặc tùy chọn **Cumulative**, sẽ backup tất thay đổi kể từ level 0 IncB.

Cần phân biệt rõ giữa Level 0 IncB và chiến lược Full backup (FB), mặc dù cả hai đều sao lưu trên toàn cơ sở dữ liệu, nhưng sẽ **không thể dùng Level 1 IncB lên FB**.

Đồng thời, có thể nhận thấy IncB được chia thành 2 loại nhỏ là **Differential** và **Cumulative**. Điểm khác biệt là sự đánh đổi giữa tốc độ và dung lượng. **Differential** backup sẽ **nhanh hơn, và ít tốn kém** vì cần sao lưu ít thay đổi, và không trùng lắp, nhưng **cần nhiều thời gian phục hồi** do phải recover tuần tự mỗi backup. Ngược lại, **Cumulative** đáp ứng **tốc độ trong recovery**, nhưng **tăng chi phí dung lượng và thời gian backup**, do sẽ bị lặp lại thông tin lưu.

Về **khôi phục**, yêu cầu trước hết đưa cơ sở dữ liệu về MOUNT, và thực hiện RESTORE với level 0 IncB. **Complete recovery**, nếu dùng **Differential**, RECOVER tuần tự từng level 1 IncB, hoặc **Cumulative**, chỉ cần RECOVER level 1 IncB gần nhất. Sau đó, RECOVER để áp log files và OPEN.

PITR, SET UNTIL trước RESTORE để chỉ định thời điểm (A) muốn khôi phục và phục hồi tới. **Differential**, RECOVER tuần tự từng level 1 IncB đến IncB gần A nhất, còn **Cumulative**, RECOVER level 1 IncB gần A nhất. Sau đó, RECOVER để áp log files và OPEN RESETLOGS để đặt lại thứ tự ghi log.

Ưu và nhược điểm sẽ được đánh giá theo tùy chọn mặc định của Oracle là Differential.

Ưu điểm: IncB là phương pháp được dùng phổ biến nhất, nhờ vào **sự linh hoạt và tiết kiệm** trong sao lưu và phục hồi, với các backup **không chiếm nhiều dung lượng**, và **thời gian backup được tối giản**. Không chỉ thế, nếu cần phải thay đổi vị trí lưu, đặc biệt là sang các dịch vụ trực tuyến, thì nhờ kích cỡ nhỏ của lượng dữ liệu cần truyền sẽ **không gây tốn kém băng thông**.

Nhược điểm: Có thể **đòi hỏi nhiều thời gian nếu cần Complete recovery**, do phải restore full backup của level 0 rồi recover tuần tự các level 1 và log files. Nếu có nhu cầu phục hồi một đối tượng dữ liệu độc lập, thì **cần tìm đủ mọi IncB liên quan và chắp vá** chúng với nhau. **Tồn tại khả năng xảy ra recover không toàn vẹn**, do chỉ cần một trong các bộ IncB bị mất thì các đối tượng ảnh hưởng sẽ rơi vào trạng thái không thể phục hồi.

Cài đặt

Nhóm chỉ thực hiện với RMAN các chiến lược sau,

- Full backup và recovery,
- Incremental differential backup và recovery.

Đầu tiên, Cold và Hot backup có thể thực hiện trong RMAN, dưới dạng Full backup đã trình bày khi bàn về Full backup và recovery. Hơn nữa, có thể nói quy trình Hot (Full) backup với RMAN bao hàm cả Cold, do cả hai hầu như tương đồng, mà Cold còn không thao tác với log files, nên việc cài Cold chỉ để trình bày là không đáng kể khi Full backup đã tổng quát hơn. Đồng thời, không làm thủ công Hot và Cold, bên cạnh lý do RMAN đã hỗ trợ, còn vì đã có thành viên phải cài lại Oracle do sai lệch metadata khi thử làm Cold backup thủ công.

Tương tự, với sự giống nhau giữa Differential và Cumulative, mà về sau lại không thể hiện được khả năng dùng nhiều Incremental backup để recover, hơn nữa, Differential là lựa chọn mặc định, nên cuối cùng chỉ thực hiện Full (Hot) backup và Incremental differential backup.

Mọi sao lưu và phục hồi được thực hiện trực tiếp trên root container database (CDB) **XE**, nếu XE hiện sở hữu nhiều PDB và dữ liệu thì cân nhắc trước khi làm theo quy trình của nhóm. Tuy chỉ dùng **XEPDB1**, nhưng lại backup và recover trên cả XE là vì rất nhiều khó khăn gặp phải khi chỉ giới hạn thao tác trên PDB, mà không động đến CDB, ví dụ,

- https://www.pythian.com/blog/restoring-a-dropped-oracle-pdb-without-a-cdb-backup

Database cần được đưa vào chế độ ARCHIVELOG, và giá trị tham số của RMAN về CONTROLFILE AUTOBACKUP phải là ON.

Full backup và recovery

1. Thực hiện Full backup PLUS ARCHIVELOG.

PITR (Incomplete recovery)

2. Một GIANGVIEN cập nhật điểm trên view V_DANGKY_V, lần 1 là chủ ý và lần 2 vô ý.

3. Cần phục hồi database về trước lần 2, nên tìm SCN của mục ghi vết cho lần 2, bằng cách tra bảng UNIFIED_AUDIT_POLICIES.

4. Đưa database về MOUNT, và thực hiện PITR với SET UNTIL SCN đã tìm được.

5. RESTORE với TAG của Full backup, có thể được tra với LIST BACKUP OF DATABASE SUMMARY, sau đó RECOVER, và OPEN RESETLOGS.

Complete recovery

6. Mô phỏng thất bại bằng cách xóa hết control files và datafiles của XE và XEPDB1.

7. RESTORE CONTROLFILE FROM control files được backup kèm với Full backup, có thể được tra với LIST BACKUP OF CONTROLFILE.

8. Đưa database về MOUNT, sau đó RESTORE với TAG, RECOVER, và OPEN.

Incremental differential backup và recovery

1. Thực hiện LEVEL 0 Full backup PLUS ARCHIVELOG.

Complete recovery

2. Một GIANGVIEN cập nhật điểm trên view V_DANGKY_V lần U#1.

3. Thực hiện LEVEL 1 Incremental backup PLUS ARCHIVELOG lần B#1.

4. Một GIANGVIEN cập nhật điểm trên view V_DANGKY_V lần U#2.

5. Thực hiện LEVEL 1 Incremental backup PLUS ARCHIVELOG lần B#2.

6. Một GIANGVIEN cập nhật điểm trên view V_DANGKY_V lần U#3.

7. Mô phỏng thất bại bằng cách xóa hết control files và datafiles của XE và XEPDB1.

8. RESTORE CONTROLFILE FROM control files được backup kèm với LEVEL 0 Full backup, có thể được tra với LIST BACKUP OF CONTROLFILE.

9. Đưa database về MOUNT, sau đó RESTORE với TAG của LEVEL 0 Full backup.

10. RECOVER với TAG của B#1 đạt được U#1, RECOVER với TAG của B#2 đạt U#2.

11. RECOVER đạt U#3, và OPEN.

PITR (Incomplete recovery)

12. Chỉ muốn đạt U#2, tức cần phục hồi về trước U#3, nên tìm SCN của mục ghi vết cho U#3, bằng cách tra bảng UNIFIED_ AUDIT_POLICIES.

13. Đưa database về MOUNT, và thực hiện PITR với SET UNTIL SCN đã tìm được.

14. RESTORE với TAG của LEVEL 0 Full backup.

15. RECOVER với TAG của B#1 đạt được U#1, RECOVER với TAG của B#2 đạt U#2.

16. RECOVER, và OPEN RESETLOGS.

Kết luận

Sử dụng phiên bản Oracle ở hiện tại thì **RMAN là giải pháp tốt nhất** cho nhu cầu sao lưu và phục hồi dữ liệu. Không chỉ làm mịn và cải thiện cả quy trình, mà công cụ còn cung cấp nhiều tính năng mạnh mẽ bảo đảm tối thiểu hóa rủi ro hư hỏng, hay mất dữ liệu. Đồng thời, với các phương pháp backup và recovery mà RMAN cung cấp, cần **xây dựng những chiến lược rõ ràng** để tận dụng hiệu quả và triệt để.

Full Hot backup cần tránh thực hiện thường xuyên, mà chỉ theo các chu kỳ cụ thể, ví dụ, sau mỗi cột mốc đáng kể theo tháng, quý, hay các sự kiện,... và nên rơi vào thời điểm có ít hoạt động để hạn chế nhất tính inconsistent. Hoặc, tận dụng những cơ hội có thể SHUTDOWN hệ thống hoàn toàn, ví dụ bảo trì hay chuẩn bị migrate, để dùng **Cold backup**.

Trên thực tế, nhờ vào sự đa dạng và linh hoạt, thì **Incremental backup nên là ưu tiên hàng đầu**. Nếu môi trường ổn định và không đặt nặng recovery thì lựa chọn **Differential**, khi đó một chiến lược cơ bản có thể là, Level 0 backup vào cuối tuần khi ít dữ liệu lưu thông, và Level 1 backup vào mỗi ngày trong tuần. Ngược lại, nếu rộng rãi về dung lượng, và tốc độ recovery là vấn đề quan tâm thì cân nhắc **Cumulative**, trong nhiều ngữ cảnh Cumulative hầu như tương đồng và có thể thay thế hoàn toàn Full backup.

Cuối cùng, sao lưu và phục hồi với RMAN **có thể được tự động hóa**, bằng cách viết batch file chỉ định quy trình thực hiện, và dùng Windows Task Scheduler để đặt lịch chạy file này định kỳ. Đọc thêm tại,

- https://support.dbagenesis.com/post/schedule-rman-backup-on-windows

Tham khảo

Giao diện

Với **Phân hệ 1**, nhóm tham khảo và cải tiến từ layout được giới thiệu trong video **Demo Hệ Thống cho người quản trị csdl - ATBMCQ-02 - An toàn bảo mật dữ liệu HTTT – HCMUS**.

- https://youtu.be/WrGmTfks_HA?si=htvejTw-2kbsrTjD

Tài liệu chung lý thuyết và code

- **Các slide trong thư mục Lectures** cung cấp bởi TS. Phạm Thị Bạch Huệ.
- **Các tài liệu trong thư mục Lab-GuideLine** cung cấp bởi ThS. Lương Vĩ Minh.
- **Hướng dẫn DAC RBAC VPD OLS** cung cấp bởi ThS. Tiết Gia Hồng.

Ngoài ra, các vấn đề phát sinh khi code được hỗ trợ thông qua,

- https://docs.oracle.com/en/
- https://asktom.oracle.com/
- https://oracle-base.com/

Phân hệ 2

Yêu cầu 2: Vận dụng mô hình điều khiển truy cập OLS

- https://youtu.be/o4-XpUQWfaM?si=jhyB0VxD761cxkg1

Yêu cầu 3: Ghi nhật ký hệ thống

- https://docs.oracle.com/database/121/TDPSG/GUID-BF747771-01D1-4BFB-8489-08988E1181F6.htm#TDPSG55281
- https://www.ibm.com/docs/en/dsm?topic=record-enabling-unified-auditing-in-oracle-12c
- https://youtu.be/vo3g_kAQM1E?si=HB89-hvF-7Js6Zvt

Yêu cầu 4: Sao lưu và phục hồi dữ liệu

Hot và Cold backup

- https://youtu.be/J5bZqz83Q34?si=GYHWnJANoA6-EdPN
- https://www.techtarget.com/searchdatabackup/definition/cold-backup
- https://storware.eu/blog/pros-and-cons-of-cold-backup-and-hot-backup-comparison/

RMAN

- https://youtu.be/FNPwCWeZO-A?si=tk586Zacr3paDWIg
- https://oracle-base.com/articles/9i/recovery-manager-9i
- https://oracle-base.com/articles/12c/multitenant-rman-backup-recovery-cdb-and-pdb-12cr1#cdb-backup
- https://www.pythian.com/blog/restoring-a-dropped-oracle-pdb-without-a-cdb-backup
- https://support.dbagenesis.com/post/oracle-rman-cold-backup-in-noarchivelog-mode
- https://support.dbagenesis.com/post/schedule-rman-backup-on-windows