ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN CUỐI KỲ

Môn học: An toàn và Bảo mật trong Hệ thống Thông tin

Sinh viên thực hiện:

20120132 - Nguyễn Tuấn Minh

21120394 - Nguyễn Hoàng Ngọc Bảo

21120405 - Trần Minh Triết

21120434 - Bùi Tiến Dũng

Giảng viên hướng dẫn:

Phạm Thị Bạch Huệ

Tiết Gia Hồng

Lương Vĩ Minh

MỤC LỤC

I. Thông tin nhóm và phân công

MSSV	Họ và tên	Nhiệm vụ	Đánh giá
20120132	Nguyễn Tuấn Minh	YC1: Database	70%
21120394	Nguyễn Hoàng Ngọc Bảo	YC1: CS1, CS2, CS4 YC2: OLS Giao diện ứng dụng	100%
21120405	Trần Minh Triết	YC1: CS6, YC3: Audit Các chức năng liên quan.	100%
21120434	Bùi Tiến Dũng	YC1: CS3, CS5 YC4: Backup & Recovery Các chức năng liên quan.	100%

II. Các chức năng đã cài đặt thành công

A. PHÂN HỆ 2: Ứng dụng quản lý dữ liệu nội bộ

Mục	Nội dung	Hoàn thành
YC1	**Cấp quyền truy cập**	
	CS#1	100%
	CS#2	100%
	CS#3	100%
	CS#4	100%
	CS#5	100%
	CS#6	100%
YC2	**Điều khiển truy cập với OLS**	
	Số lượng chức năng hoàn thành: 8/8	100%
YC3	**Ghi nhật ký hệ thống**	
	Kích hoạt việc ghi nhật ký hệ thống	100%
	Thực hiện standard audit trên một vài loại đối tượng	100%
	Thực hiện Fine-grained Audit	100%
	Kiểm tra dữ liệu nhật ký hệ thống	100%
YC4	**Sao lưu và phục hồi dữ liệu**	
	Tìm hiểu phương pháp sao lưu và phục hồi dữ liệu	100%
	Thực hành trên Oracle	100%
	Đánh giá và kết luận	100%

III. Các lý thuyết và thực hành áp dụng

A. PHÂN HỆ 2: Ứng dụng quản lý dữ liệu nội bộ

1. Yêu cầu 1 - Cấp quyền truy cập:

a) Các cơ chế điều khiển truy cập Access Control được sử dụng:

(1) RBAC (Role-based Access Control)

- RBAC gán quyền truy cập dựa trên vai trò của người dùng trong hệ thống. Thay vì gán quyền truy cập trực tiếp cho từng người dùng, quyền truy cập được gán cho vai trò, và người dùng sẽ được gán vào các vai trò đó.
- Áp dụng trong đồ án: các lệnh `GRANT` được sử dụng để gán quyền cho các vai trò như "NhanVienCoBan", "GiangVien", "GiaoVu", "TruongDonVi", "TruongKhoa".

(2) DAC (Discretionary Access Control)

- Cho phép xác định người dùng nào có quyền truy cập vào dữ liệu gì. Cơ chế kiểm soát truy cập linh động, cho phép chỉ định quyền truy cập cho từng người dùng cụ thể.
- Áp dụng trong đồ án: được sử dụng để cấp quyền `SELECT`, `UPDATE`, `INSERT`, `DELETE` cho các bảng và các views cụ thể

(3) CBAC (Content-based Access Control using Views)

- View có thể được dùng để dấu đi các thông tin nhạy cảm bằng cách giới hạn những thông tin user có thể được xem.
- Quyền trên views là độc lập đối với quyền trên bảng bên dưới.
- Để kiểm soát truy cập, chỉ cho phép user được thao tác trên view thay vì trên bảng.

(4) Virtual Private Database (VPD)

- Đặt một chính sách lên bảng, views hoặc synonyms được định nghĩa bằng một function, function này sẽ trả về một predicate (vị từ). Vị từ này sẽ tự động được thêm vào mệnh đề `WHERE` của câu truy vấn khi xảy ra truy xuất đến đối tượng. Khi có nhiều hàm policy cùng trả về vị từ, các vị từ sẽ được ghép lại bằng phép logic "`AND`", bao gồm vị từ ban đầu của câu truy vấn.
- Hàm chính sách được gắn vào đối tượng bằng procedure `DBMS_RLS.ADD_POLICY`.
- Đối với các câu truy vấn không có mệnh đề `WHERE` như `INSERT` hoặc có sự thay đổi về dữ liệu như `UPDATE`, có thể sử dụng tham số update_check của procedure `APP_POLICY` để kiểm tra dòng dữ liệu mới được thêm/sửa có thỏa điều kiện của vị từ hay không.

(5) Application Context:

- Sử dụng hàm SYS_CONTEXT và các namespace có sẵn như USERENV để lấy các thông tin liên quan đến session của người dùng, từ đó truy vấn được các thông tin khác.

b) Chính sách 1 - Người dùng "Nhân viên cơ bản":

- Tạo view sử dụng câu truy vấn kết hợp SYS_CONTEXT để chọn ra dòng liên quan đến user hiện tại. Chỉ cấp quyền SELECT trên view này thay vì trên bảng phía dưới.
- Từ view này, user có thể update số điện thoại với quyền update trên cột DT, Oracle Database sẽ tự ánh xạ các thay đổi trên view để truyền xuống bảng.
- Đối với các bảng mà nhân viên cơ bản được xem, cấp quyền SELECT cho họ trên các bảng này.

c) Chính sách 2 - Người dùng "Giảng viên":

- Do Giảng viên có các quyền như Nhân viên cơ bản, cấp role NhanVienCoBan cho giảng viên để họ có thể kế thừa các quyền của role này.
- Tương tự với CS1, để giới hạn phạm vi xem của bảng PHANCONG và DANGKY, sử dụng SYS_CONTEXT và VIEW để chọn ra những dòng liên quan, chỉ cấp quyền SELECT trên view này thay vì trên bảng.
- Cấp quyền update trên các cột điểm trên view tương ứng của bảng DANGKY.

d) Chính sách 3 - Người dùng "Giáo vụ":

- Do Giáo vụ có các quyền như một Nhân viên cơ bản, cấp role NhanVienCoBan cho giáo viên để họ kế thừa các quyền của role này.
- Cấp các quyền XEM, THÊM, CẬP NHẬT dữ liệu trên các quan hệ SINHVIEN, ĐONVI, HOCPHAN, KHMO
- Cấp quyền XEM trên quan hệ PHANCONG
- Tạo một view kết hợp PHANCONG với MADV để lấy ra những đơn vị có tên là 'Văn phòng khoa'. Và cấp quyền CẬP NHẬP trên view này cho Giáo vụ
- Tạo 1 view để lấy danh sách các đăng ký trong DANGKY. Những đăng ký thoả là những đăng ký có HK bằng 1,2 hoặc 3 thì sẽ so sánh với các tháng tương ứng hay không lần lượt là 1, 5 hoặc 9. so sánh NAM với TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')). Và ngày hiện tại có đang < 15 hay không. Cấp quyền THÊM và XOÁ trên view này cho Giáo Vụ

e) Chính sách 4 - Người dùng "Trưởng đơn vị":

- Tạo một view kết hợp với SYS_CONTEXT để chọn ra những dòng trên bảng PHANCONG đối với các môn học thuộc đơn vị mình làm trưởng. Chỉ cấp quyền đọc và cấp role GIANGVIEN cho người dùng để họ được kế thừa các quyền của role này.

- Tương tự, tạo một view từ bảng PHANCONG và chọn ra các dòng liên quan đến các giảng viên thuộc về đơn vị mà người dùng đang làm trưởng đơn vị. Cấp quyền SELECT trên view này cho người dùng.

f) Chính sách 5 - Người dùng "Trưởng khoa":

- Cấp role GIANGVIEN cho người dùng để họ được kế thừa các quyền của role này.
- Tạo một view từ bảng PHANCONG và chọn ra các dòng liên quan đến các học học phần thuộc về đơn vị "Văn phòng Khoa"mà người dùng đang làm trưởng đơn vị.
- Cấp quyền SELECT trên view này cho người dùng.

g) Chính sách 6 - Người dùng "Sinh viên":

- Trên bảng SINHVIEN, người dùng chỉ được xem các thông tin của bản thân. Để quản lí truy cập, tạo một function với chức năng như sau: sử dụng namespace USERNV để lấy tên đăng nhập của người dùng, nếu người dùng là SV thì trả về vị từ `predicate := 'MASV = ''' || v_current_user || '''';` chỉ cho phép user xem dòng thông tin của mình. Gắn function này vào bảng SINHVIEN và chỉ định statement được áp dụng là SELECT.
- Để giới hạn quyền update trên cột DCHI và ĐT của chính sinh viên, grant quyền update trên 2 cột tương ứng. Để giới hạn dòng dữ liệu người dùng có thể update, tương tự với quyền SELECT, nhưng chỉ định statement áp dụng là UPDATE và đi kèm tham số `update_check = TRUE` để kiểm tra dòng dữ liệu sau cập nhật
- Để giới hạn khả năng xem HOCPHAN và KHMO của chương trình đào tạo của SINHVIEN, ta muốn chương trình thực thi một câu truy vấn như sau:

```
SELECT *
FROM KHMO
WHERE MACT = (SELECT MACT FROM SINHVIEN WHERE MASV = <MASV> )
AND MAHP IN (SELECT MAHP FROM HOCPHAN WHERE MADV = (SELECT
MADV
FROM DONVI
JOIN SINHVIEN ON SINHVIEN.MANGANH = DONVI.TENDV
WHERE MASV = <MASV>));
```

- Nhưng do trên bảng SINHVIEN sẽ tự động trả về vị từ `MASV = SYS_CONTEXT('USERENV','CURRENT_USER')` nên không cần trả về thêm vị từ này, do đó policy function chỉ cần trả về phần vị từ còn lại như sau: `MACT = (SELECT MACT FROM SINHVIEN WHERE MASV =`

<MASV>) AND MAHP IN (SELECT MAHP FROM HOCPHAN WHERE MADV = (SELECT MADV FROM DONVI JOIN SINHVIEN ON SINHVIEN.MANGANH = DONVI.TENDV)).

- Để kiểm soát việc xóa sửa trên bảng DANGKY, tạo một function để xác định kỳ học dựa trên SYSDATE. Sau đó ta tạo một function trả về một vị từ thỏa điều kiện của đề. Khi gắn policy lên bảng, sử dụng tham số update_check = TRUE để kiểm tra các dòng mới được thêm vào.
- Không cho người dùng update điểm bằng cách không cấp quyền update trên cột điểm. Để ngăn chặn người dùng vượt quyền mà update các cột điểm, tạo trigger ngăn chặn hành vi update này.

2. Yêu cầu 2 - Điều khiển truy cập với OLS:

Cài đặt thành công OLS và ứng dụng OLS để phát tán các thông báo tươngứng đến các người dùng nhất định. Dựa vào cơ chế gán nhãn, gán nhãn cho user và gán nhãn cho dữ liệu và những thông báo sẽ được đảm bảo gửi đến những đối tượng nhất định, những đối tượng không thỏa mãn sẽ không thể thấy được các thông báo.

Ta tạo các level cho OLS: SV - Sinh viên, NV - Nhân viên, GV - Giảng viên, GVu - Giáo vụ, TRGDV - Trưởng đơn vị, TRGKH - Trưởng khoa với mức độ bảo mật tăng dần từ NV đến TRGKH.

Ta tạo các compartment cho OLS: HTTT - Hệ thống thông tin, CNPM - Công nghệ phần mềm, KHMT - Khoa học máy tính, CNTT - Công nghệ thông tin, TGMT - Thị giác máy tính, MMT - Mạng máy tính.

Ta tạo các group cho OLS: UNI - Đại học, CS1 - Cơ sở 1, CS2 - Cơ sở 2, với CS1, CS2 là con của UNI.

Tiếp theo, tạo bảng THONGBAO và add dữ liệu vào để test. Sau đó, áp dụng chính sách OLS bên trên cho table này.

Tạo nhãn thủ công cho các dòng dữ liệu trong bảng THONGBAO.

Tạo các user để test và dùng procedure SA_USER_ADMIN.SET_USER_LABELS để gắn nhãn cho các user được tạo ra để test.

Cuối cùng , ta tạo các procedure để gắn nhãn lên các user có trong hệ thống. Do bảng dữ liệu được yêu cầu không có thông tin liên quan đến cơ sở, nên các người dùng trong hệ thống chỉ được gán nhãn tối đa:

LEVEL:COMPARTMENT riêng Trưởng Khoa thì được gán nhãn mạnh nhất để xem toàn bộ thông báo.

Các nhãn được yêu cầu trong đồ án:

1) Hãy gán nhãn cho người dùng là Trưởng khoa có thể đọc được toàn bộ thông báo: TRGKHOA:HTTT,CNPM,KHMT,CNTT,TGMT,MMT:UNI
2) 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: TRGDV::UNI
3) Hãy gán nhãn cho 01 Giáo vụ có thể đọc toàn bộ thông báo dành cho giáo vụ: GVu::UNI
4) Hãy cho biết nhãn của dòng thông báo t1 để t1 được phát tán (đọc) bởi tất cả Trưởng đơn vị: TRGDV
5) Hãy cho biết nhãn của dòng thông báo t2 để phát tán t2 đến Sinh viên thuộc ngành HTTT học ở Cơ sở 1: SV:HTTT:CS1
6) Hãy cho biết nhãn của dòng thông báo t3 để phát tán t3 đến Trưởng bộ môn KHMT ở Cơ sở 1: TRGDV:KHMT:CS1
7) Hãy 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: TRGDV:KHMT
8) Hãy cho thêm 3 chính sách phát tán dòng dữ liệu nữa trên mô hình OLS đã cài đặt:
    a) Nhãn của dòng thông báo cho tất cả SinhVien: SV
    b) Nhãn của dòng thông báo cho tất cả NhanVien: NV
    c) Nhãn của dòng thông báo cho tất cả Giảng viên ở cơ sở 2: GV::CS2
    d) Nhãn của dòng thông báo cho tất cả Giáo vụ ở cơ sở 2: GVu::CS2
    e) Nhãn của dòng thông báo cho Trưởng đơn vị ở cơ sở 1: TRGDV::CS1
    f) Nhãn của dòng thông báo cho Giảng viên CNPM: GV:CNPM

3. Yêu cầu 3 - Ghi nhật ký hoạt động với Audit:

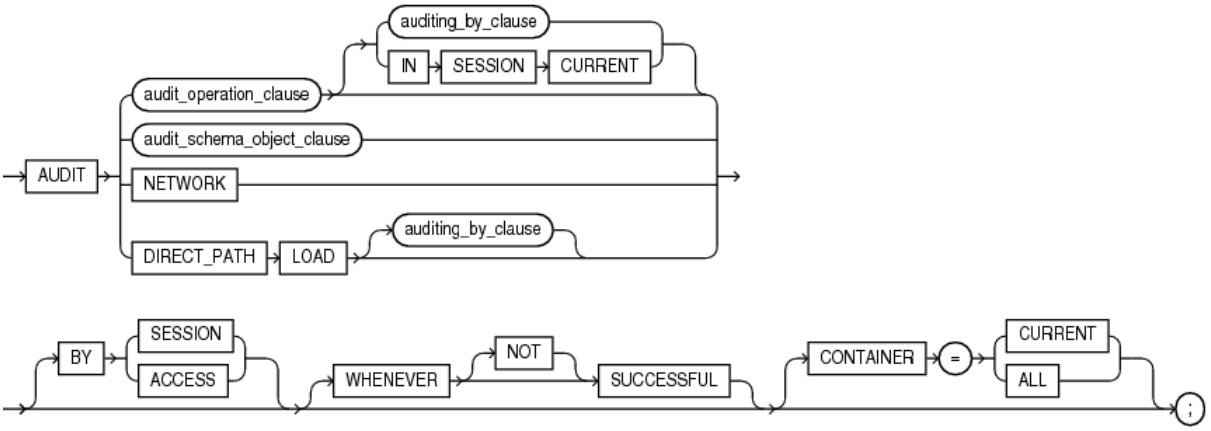
a) Kích hoạt Audit

- Để thực hiện việc kích hoạt audit, trong SQL*Plus dưới quyền SYSDBA, thực hiện các câu lệnh sau: `ALTER SYSTEM SET audit_trail=DB, extended SCOPE=SPFILE;`
- Lệnh trên sẽ bật chế độ audit của DBMS và lưu các bản ghi audit vào database audit trail, có thể được xem bằng cách truy vấn các view trong data dictionary, việc set giá trị thành extended còn cho phép ghi thêm các

thông tin liên quan đến câu SQL và các biến SQL_BIND bị audit. Hai giá trị này chỉ được ghi lại khi khai báo tham số này. Ngoài ra các vị từ VPD và hàm policy cũng sẽ được ghi lại.

b) Standard Audit

- Để thực hiện audit, cần có quyền AUDIT SYSTEM.
- Standard Audit hay còn được gọi là Traditional Audit, đã bị thay thế bởi Unified Audit trong các phiên bản mới của Oracle, trong tương lai có thể không còn được hỗ trợ.
- Dùng để ghi vết các hành vi trên các đối tượng nhất định như table, view, stored procedure, function… hoặc audit một user cụ thể.
- Dưới đây là bảng cú pháp của câu lệnh AUDIT, tùy theo đối tượng muốn audit mà sẽ có chút biến thể. Ví dụ, khi muốn audit một bảng: AUDIT ALL SCHEMA.TABLE_NAME BY ACCESS hay AUDIT ALL BY USER_NAME sẽ audit tất cả hành động của user trên cơ sở dữ liệu.
- Mệnh đề WHENEVER (NOT) SUCCESSFUL dùng để audit khi hành động thành công (hoặc không thành công đối với NOT)



- Để xem các bảng ghi audit được ghi bởi standard audit, truy vấn trong view DBA_AUDIT_TRAIT. Một vài cột quan trọng như: username, owner, object_name, sql_text, sql_bind, current_user, extended_timestamp.

c) Fine-grained Audit

- Fine-grained Audit cho phép audit trên mức độ cột và có thể khai báo một khối PL/SQL làm event handler, ví dụ như gửi email khi audit ghi được một bản ghi mới.
- Tương tự VPD, fine-grained audit cũng sử dụng một vị từ đại diện cho điều kiện kích hoạt audit. Câu SQL sẽ bị ghi lại khi có bất kỳ dòng nào trả về thỏa điều kiện audit.
- Để sử dụng fine-grained audit, cần có quyền AUDIT_ADMIN hoặc quyền thực thi trên package DBMS_FGA.

- Để gán một hàm chính sách lên bảng, sử dụng thủ tục ADD_POLICY của package DBMS_FGA. Syntax như sau:

```
DBMS_FGA.ADD_POLICY(
   object_schema      IN  VARCHAR2 DEFAULT NULL
   object_name        IN  VARCHAR2,
   policy_name        IN  VARCHAR2,
   audit_condition    IN  VARCHAR2 DEFAULT NULL,
   audit_column       IN  VARCHAR2 DEFAULT NULL
   handler_schema     IN  VARCHAR2 DEFAULT NULL,
   handler_module     IN  VARCHAR2 DEFAULT NULL,
   enable             IN  BOOLEAN DEFAULT TRUE,
   statement_types    IN  VARCHAR2 DEFAULT SELECT,
   audit_trail        IN  BINARY_INTEGER DEFAULT NULL,
   audit_column_opts  IN  BINARY_INTEGER DEFAULT ANY_COLUMNS,
   policy_owner       IN  VARCHAR2 DEFAULT NULL);
```

- Trong đó audit_condition là câu điều kiện kích hoạt audit. Audit column chỉ định cột nào sẽ được audit. Enable chỉ định việc sau khi gắn policy, policy có được áp dụng ngay lập tức hay không. handler_schema và handler_module **dùng** để khai báo khối PL/SQL thực hiện việc xử lý hậu audit. Statement_types chỉ mệnh đề nào sẽ được audit: ví dụ SELECT, INSERT, UPDATE, DELETE…
- Để xem các bản ghi audit được ghi bằng fine-grained audit, xem view DBA_FGA_AUDIT_TRAIL trên từ điển dữ liệu.
- Chính sách 1: Đăng ký một chính sách audit trên bảng DANGKY và các cột liên quan tới DIEM với điều kiện SYS_CONTEXT(''SYS_SESSION_ROLES'', ''GIANGVIEN'') = ''FALSE''và mệnh đề được chỉ định là UPDATE. Namespace SYS_SESSION_ROLES sẽ trả về true hoặc false dựa trên kết quả so sánh giữa tham số khai báo và role của người dùng.
- Chính sách 2: Đăng ký một chính sách audit trên bảng NHANSU, cột PHUCAP với điều kiện SYS_CONTEXT(''USERENV'', ''SESSION_USER'') != MANV và mệnh đề được chỉ định là SELECT. Khi có bất kỳ dòng nào trả về với giá trị trên cột PHUCAP mà MANV khác với session_user thì sẽ kích hoạt audit.

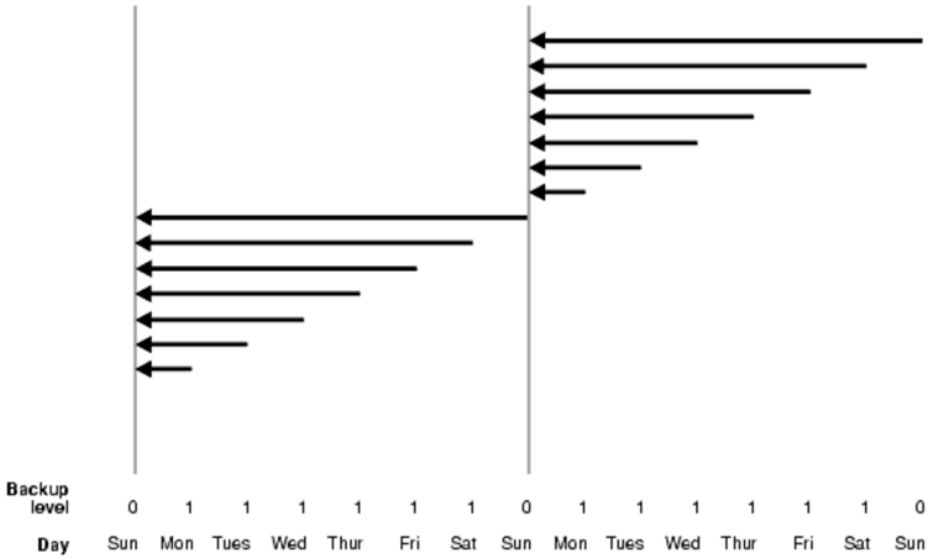
4. Yêu cầu 4 - Sao lưu và phục hồi dữ liệu:

a) Tìm hiểu BACKUP:

- **Full Backup, sao lưu đầy đủ:** Là quá trình sao lưu toàn bộ các khối dữ liệu của một tập tin dữ liệu mà không quan tâm tới chúng có bị thay đổi hay không.

- **Incremental Backup, sao lưu gia tăng:** Là quá trình chỉ sao lưu những khối dữ liệu của một tập tin dữ liệu mà có sự thay đổi so với lần sao lưu trước đó.
    - **Cumulative incremental backup**: Một bản sao lưu gia tăng tích lũy, sao lưu tất cả các khối đã thay đổi sau lần sao lưu gia tăng gần đây nhất ở cấp 0.

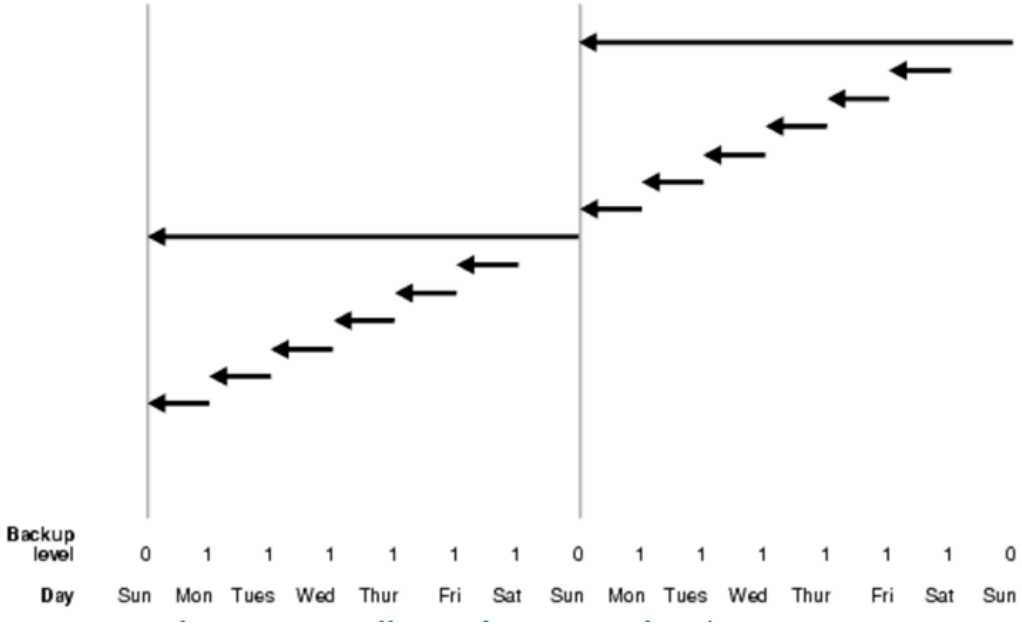
Figure 8-3 Cumulative Incremental Backups



- Ví dụ. Vào mỗi cuối ngày chủ nhật hàng tuần, người quản trị sẽ thực hiện **backup level 0** lưu toàn bộ các khối dữ liệu của tập tin. Các ngày còn lại trong tuần sẽ thực hiện **backup level 1** tức là chỉ lưu các khối dữ liệu thay đổi so với lần backup hồi chủ nhật. Vào chủ nhật tiếp theo sẽ thực hiện **backup level 0** và các ngày còn lại sẽ tiếp tục thực hiện như các tuần trước.

    - **Differential incremental backup**: một bản sao lưu gia tăng khác biệt, sao lưu tất cả các khối đã thay đổi sau lần sao lưu gia tăng gần đây nhất ở cấp 1 hoặc 0.

Figure 8-2 Differential Incremental Backups



- Ví dụ: Cũng vào mỗi cuối chủ nhật hàng tuần, người quản trị sẽ thực hiện **backup level 0** lưu toàn bộ các khối dữ liệu của tập tin. Ngày thứ 2 sẽ thực hiện **backup level 1**, chỉ lưu các khối lệnh có thay đổi so với ngày chủ nhật gần nhất, ngày thứ 3 sẽ thực hiện **backup level 1**, chỉ lưu các khối lệnh có thay đổi so với ngày thứ 2 gần nhất, tiếp tục với các ngày khác trong tuần. Cuối tuần người quản trị sẽ thực hiện **backup level 0** lại lần nữa và tiếp tục với các ngày khác trong tuần.
- **Cơ chế thực hiện:**

b) Đánh giá ưu khuyết điểm:

Full Backup:

- **Ưu điểm:** Dễ thực hiện vì mỗi lần sao lưu chỉ cần sao lưu toàn bộ khối dữ liệu mà không cần phải theo dõi những thay đổi kể từ lần sao lưu trước. Dễ dàng quản lý vì mỗi lần sao lưu đều là một bản sao đầy đủ và độc lập của dữ liệu, giúp dễ dàng khôi phục dữ liệu từ một bản sao lưu cụ thể mà không cần áp dụng thêm các bản sao lưu gia tăng hay khác biệt.
- **Nhược điểm:** Tốn thời gian, tốn bộ nhớ vì việc sao lưu toàn bộ dữ liệu mất nhiều thời gian hơn, đặc biệt là đối với các cơ sở dữ liệu lớn và đòi hỏi mỗi lần sao lưu là một bản sao đầy đủ của toàn bộ dữ liệu, dung lượng lưu trữ yêu cầu sẽ tăng lên nhanh chóng, dẫn đến việc lãng phí không gian lưu trữ do nhiều bản sao lưu chứa cùng một dữ liệu không thay đổi.

Incremental Backup:

- **Ưu điểm:** Tiết kiệm không gian bộ nhớ vì chỉ các khối dữ liệu thay đổi kể từ lần sao lưu trước được sao lưu, không gian lưu trữ cần thiết giảm đáng kể so với việc sao lưu đầy đủ. Tiết kiệm thời gian sao lưu việc sao lưu chỉ các thay đổi giúp giảm thời gian thực hiện sao lưu, đặc biệt có lợi khi sao lưu thường xuyên và dữ liệu thay đổi không nhiều.
- **Nhược điểm:** Cần hiểu rõ về kỹ thuật và chiến lược sao lưu để thiết lập và quản lý hiệu quả sao lưu gia tăng, người quản trị cần có kiến thức sâu về các khái niệm và công cụ sao lưu. Quá trình khôi phục phức tạp hơn vì việc khôi phục dữ liệu đòi hỏi áp dụng tuần tự các bản sao lưu gia tăng cùng với bản sao lưu đầy đủ ban đầu, làm tăng độ phức tạp và thời gian khôi phục. Quản lý nhiều bản sao lưu liên quan: Do các bản sao lưu gia tăng phụ thuộc lẫn nhau, cần phải quản lý và bảo quản cẩn thận tất cả các bản sao lưu để đảm bảo khả năng khôi phục chính xác.

c) Kết luận:

- Full backup là cách tiếp cận đơn giản và nhanh chóng, nhưng sẽ tốn nhiều thời gian và không gian lưu trữ.
- Incremental backup là cách tiếp cận hiệu quả hơn, cải thiện được những vấn đề mà Full backup gặp, đòi hỏi phải hiểu rõ quy trình để thực hiện chuẩn xác.