TRƯỜNG ĐẠI HỌC CÔNG NGHỆ KỸ THUẬT TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CUỐI KÌ

ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG WEB THEO MÔ HÌNH 3 LỚP QUẢN TRỊ NGƯỜI DÙNG TẬP TRUNG VÀ BẢO MẬT TRÊN DỮ LIỆU DỰ ÁN CỦA CÔNG TY

Môn học: Bảo mật cơ sở dữ liệu

GVHD: ThS. Lê Thị Minh Châu

Mã môn học: DBSE431284

Nhóm: 12

Sinh viên thực hiện:

Đinh Trọng Đức Anh	22110096
Phạm Phúc Hưng	21110275

TP Hồ Chí Minh, ngày 31 tháng 12 năm 2025

ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG WEB THEO MÔ HÌNH 3 LỚP QUẢN TRỊ NGƯỜI DÙNG TẬP TRUNG VÀ BẢO MẬT TRÊN DỮ LIỆU DỰ ÁN CỦA CÔNG TY

DANH SÁCH THÀNH VIÊN THỰC HIỆN

Họ và tên	**MSSV**
Đinh Trọng Đức Anh	22110096
Phạm Phúc Hưng	21110275

Nhận xét của giáo viên:

Ngày 31 tháng 12 năm 2025

Giảng viên chấm điểm

TP.HCM, ngày 31 tháng 12 năm 2025

MỤC LỤC

LỜI CẢM ƠN

Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Công nghệ Kỹ thuật TP.HCM và Khoa Công nghệ Thông tin đã tạo điều kiện môi trường học tập tốt nhất cho chúng em.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến ThS. Lê Thị Minh Châu, giảng viên bộ môn Bảo mật Cơ sở dữ liệu. Trong suốt quá trình học tập và thực hiện đề tài, cô đã tận tình giảng dạy, truyền đạt những kiến thức nền tảng quý báu về quản trị và bảo mật trên hệ quản trị cơ sở dữ liệu Oracle. Những hướng dẫn chi tiết và sự góp ý của cô chính là kim chỉ nam giúp nhóm định hướng đúng đắn và hoàn thiện sản phẩm cuối cùng.

Mặc dù đã nỗ lực hết mình để tìm hiểu và áp dụng các công nghệ mới và các cơ chế bảo mật nâng cao, nhưng do hạn chế về mặt thời gian và kinh nghiệm thực tế, bài báo cáo khó tránh khỏi những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp từ cô để có thể hoàn thiện kiến thức và phát triển tốt hơn trong tương lai.

Nhóm chúng em xin chân thành cảm ơn cô!

BẢNG PHÂN CÔNG NHIỆM VỤ

Nhóm: 12

Đề tài: XÂY DỰNG ỨNG DỤNG WEB THEO MÔ HÌNH 3 LỚP QUẢN TRỊ NGƯỜI DÙNG TẬP TRUNG VÀ BẢO MẬT TRÊN DỮ LIỆU DỰ ÁN CỦA CÔNG TY.

STT	MSSV	Họ và tên	Nội dung thực hiện	Mức độ hoàn thành
1	22110096	Đinh Trọng Đức Anh	- Quản lý vòng đời User (Tạo/Sửa/Xóa/Khóa). - Quản lý Profile và Password Policy. - Cấu hình Unified Auditing giám sát DDL/DML. - Data Redaction che giấu dữ liệu nhạy cảm.	100%
2	21110275	Phạm Phúc Hưng	- Quản lý Role và cấp phát quyền (Grant/Revoke). - Virtual Private Database (VPD) phân quyền theo phòng ban. - Fine-Grained Auditing (FGA) giám sát cột ngân sách	100%

Ghi chú: % mức độ hoàn thành công việc

CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

Trong kỷ nguyên số, dữ liệu được xem là tài sản quý giá nhất của mọi doanh nghiệp. Tuy nhiên, đi kèm với giá trị đó là những nguy cơ tiềm ẩn về an toàn thông tin, không chỉ từ các cuộc tấn công bên ngoài mà còn từ chính sự lạm dụng quyền hạn của người dùng nội bộ (insider threats).

Hiện nay, việc quản trị người dùng và bảo mật trong các hệ quản trị cơ sở dữ liệu (DBMS) lớn như Oracle thường được thực hiện thông qua các công cụ dòng lệnh (SQL*Plus) hoặc các phần mềm quản trị chuyên dụng (Oracle SQL Developer, TOAD). Các công cụ này tuy mạnh mẽ nhưng đòi hỏi người quản trị phải có kiến thức chuyên sâu về câu lệnh SQL, gây khó khăn cho công tác quản lý tập trung và thiếu tính trực quan. Hơn nữa, việc bảo mật dữ liệu thường bị dồn gánh nặng lên tầng ứng dụng (Application Layer), dẫn đến rủi ro "bỏ lọt" nếu kẻ tấn công truy cập trực tiếp vào cơ sở dữ liệu.

Xuất phát từ thực tế đó, nhóm quyết định lựa chọn đề tài " XÂY DỰNG ỨNG DỤNG WEB THEO MÔ HÌNH 3 LỚP QUẢN TRỊ NGƯỜI DÙNG TẬP TRUNG VÀ BẢO MẬT TRÊN DỮ LIỆU DỰ ÁN CỦA CÔNG TY". Đề tài hướng tới việc xây dựng một giao diện web thân thiện giúp đơn giản hóa các tác vụ quản trị, đồng thời áp dụng triệt để các cơ chế bảo mật tiên tiến tích hợp sẵn trong Oracle Database 23ai như VPD, FGA và Data Redaction để bảo vệ dữ liệu từ gốc.

1.2. Mục tiêu và phạm vi nghiên cứu

1.2.1. Mục tiêu

Đề tài tập trung vào hai mục tiêu chính:

1. Xây dựng hệ thống quản trị. Phát triển ứng dụng web theo mô hình 3 lớp (3-layer model) cho phép quản trị viên thực hiện các thao tác quản lý vòng đời người dùng (User Lifecycle), Profile, Role và phân quyền một cách trực quan, chính xác mà không cần nhớ nhiều câu lệnh phức tạp.
2. Triển khai bảo mật chuyên sâu. Nghiên cứu và hiện thực các giải pháp bảo mật nâng cao của Oracle để giải quyết các bài toán thực tế:
    - Phân quyền dữ liệu theo phòng ban (Ai ở đâu chỉ thấy dữ liệu ở đó).

- Giám sát chặt chẽ các hành động nhạy cảm liên quan đến tài chính.
- Che giấu thông tin cá nhân (PII) đối với những người dùng không có thẩm quyền.

1.2.2. Phạm vi nghiên cứu

- Về mặt dữ liệu: Hệ thống sử dụng lược đồ dữ liệu giả lập quản lý dự án công ty, bao gồm các bảng chính: USERS (Thông tin nhân viên), PROJECTS (Dự án), BUDGET (Ngân sách).
- Về mặt chức năng:
  - Quản lý User: Tạo, sửa, xóa, khóa/mở khóa tài khoản, reset mật khẩu.
  - Quản lý Profile: Thiết lập chính sách mật khẩu (số lần đăng nhập sai, thời hạn mật khẩu) và giới hạn tài nguyên (session).
  - Bảo mật:
    - Unified Auditing: Ghi vết các hành động DDL (Create/Drop User) và DML.
    - Fine-Grained Auditing (FGA): Giám sát sự thay đổi trên cột Ngân sách (Budget).
    - Virtual Private Database (VPD): Phân quyền truy cập dòng dữ liệu theo phòng ban (IT, Sales).
    - Data Redaction: Che giấu số điện thoại và email trên giao diện truy vấn.

1.3. Công nghệ sử dụng

Để hiện thực hóa hệ thống, nhóm sử dụng các công nghệ hiện đại và phù hợp với yêu cầu của môn học:

1. Hệ quản trị cơ sở dữ liệu (Database):

- Oracle Database 23ai Free: Phiên bản mới nhất của Oracle với nhiều cải tiến về hiệu năng và bảo mật.
- Docker: Sử dụng Container để đóng gói môi trường database, giúp việc triển khai nhanh chóng và đồng nhất trên các máy cá nhân.

2. Backend (Business Logic Layer):

- Python 3.12: Ngôn ngữ lập trình chính.

- FastAPI: Web framework hiện đại, hiệu năng cao, hỗ trợ tốt cho việc xây dựng RESTful API.
- Thư viện python-oracledb: Driver kết nối chính thức từ Oracle cho Python (chạy ở chế độ Thin mode, không cần cài đặt Oracle Instant Client phức tạp).

3. Frontend (Presentation Layer):
   - Jinja2 Templates: Render giao diện phía server (Server-side rendering), tích hợp chặt chẽ với FastAPI.
   - Bootstrap 5: Framework CSS giúp xây dựng giao diện responsive và thẩm mỹ nhanh chóng.
   - HTML5/CSS3/JavaScript: Các công nghệ web cơ bản.
4. Công cụ phát triển:
   - Visual Studio Code: IDE chính.
   - Git/GitHub: Quản lý mã nguồn.
   - SQLPlus: Công cụ dòng lệnh để kiểm thử trực tiếp các policy bảo mật trong database.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT VỀ BẢO MẬT TRONG ORACLE

Để xây dựng một hệ thống quản trị cơ sở dữ liệu an toàn và hiệu quả, việc nắm vững các cơ chế bảo mật cốt lõi của Oracle Database là điều kiện tiên quyết. Chương này sẽ trình bày các nền tảng lý thuyết về quản lý tài nguyên, kiểm soát truy cập và các giải pháp bảo mật nâng cao mà nhóm đã áp dụng trong đề tài.

2.1. Quản lý người dùng và tài nguyên

Trong Oracle Database, việc bảo mật bắt đầu từ việc kiểm soát danh tính và tài nguyên của người dùng.

2.1.1. User và Authentication

Người dùng (User) là một tài khoản được định danh trong cơ sở dữ liệu, cho phép đăng nhập và truy cập vào các đối tượng dữ liệu. Oracle sử dụng cơ chế Authentication (Xác thực) để xác minh danh tính người dùng, thông thường qua mật khẩu được lưu trữ dưới dạng mã hóa (hash) trong Data Dictionary.

Một User khi được khởi tạo cần được quy định các thuộc tính lưu trữ:

- Default Tablespace: Không gian lưu trữ mặc định cho các đối tượng (bảng, chỉ mục) mà user tạo ra.
- Temporary Tablespace: Không gian dùng cho các tác vụ xử lý tạm thời như sắp xếp (sorting) hay gom nhóm (grouping) dữ liệu
- Quota: Giới hạn dung lượng tối đa mà user được phép sử dụng trên một Tablespace cụ thể, ngăn chặn việc một cá nhân chiếm dụng toàn bộ tài nguyên ổ cứng.

2.1.2. Profile và Password Policy

Profile là một tập hợp các giới hạn về tài nguyên hệ thống và chính sách mật khẩu được gán cho user. Việc sử dụng Profile giúp quản trị viên kiểm soát:

Giới hạn tài nguyên (Resource Limits):

- SESSIONS_PER_USER: Giới hạn số lượng kết nối đồng thời từ một user.
- CONNECT_TIME và IDLE_TIME: Giới hạn thời gian kết nối và thời gian nhàn rỗi, giúp giải phóng tài nguyên server.

Chính sách mật khẩu (Password Limits):

- FAILED_LOGIN_ATTEMPTS: Số lần đăng nhập sai cho phép trước khi tài khoản bị khóa (LOCKED).
- PASSWORD_LIFE_TIME: Thời hạn sử dụng của mật khẩu, yêu cầu người dùng thay đổi định kỳ.

2.2. Kiểm soát truy cập (Access Control - Role & Privilege)

Hệ thống bảo mật của Oracle dựa trên nguyên tắc đặc quyền tối thiểu (Least Privilege), đảm bảo người dùng chỉ có những quyền hạn cần thiết cho công việc của họ.

2.2.1. Privilege (Quyền hạn)

Quyền hạn trong Oracle được chia làm hai loại chính:

- System Privileges (Quyền hệ thống): Cho phép thực hiện các hành động quản trị tác động đến toàn bộ hệ thống hoặc các schema khác (ví dụ: CREATE SESSION, CREATE TABLE, DROP USER, CREATE PROCEDURE).
- Object Privileges (Quyền đối tượng): Cho phép thao tác trên các đối tượng dữ liệu cụ thể (ví dụ: SELECT, INSERT, UPDATE trên bảng EMPLOYEES).

2.2.2. Role (Vai trò)

Để đơn giản hóa việc quản lý quyền hạn trong các hệ thống lớn, Oracle cung cấp khái niệm Role. Role là một nhóm các quyền (System và Object privileges) được đặt tên.

Thay vì cấp từng quyền lẻ tẻ cho hàng trăm người dùng, quản trị viên tạo một Role (ví dụ: MANAGER_ROLE), gán các quyền cần thiết vào Role đó, và sau đó gán Role cho người dùng.

Cơ chế này được gọi là RBAC (Role-Based Access Control), giúp việc thu hồi hoặc bổ sung quyền trở nên nhanh chóng và đồng bộ.

2.3. Bảo mật mức dòng (Virtual Private Database - VPD)

2.3.1. Khái niệm

Virtual Private Database (VPD), hay còn gọi là Fine-Grained Access Control (FGAC) hoặc Row-Level Security (RLS), là tính năng cho phép kiểm soát truy cập dữ

liệu ở mức độ dòng (record). VPD cho phép nhiều người dùng truy cập vào cùng một bảng dữ liệu nhưng chỉ nhìn thấy những dòng dữ liệu mà họ được phép thấy, dựa trên các chính sách bảo mật động.

2.3.2. Cơ chế hoạt động

VPD hoạt động bằng cách tự động can thiệp vào câu lệnh SQL của người dùng:

- Người dùng gửi câu truy vấn (ví dụ: SELECT * FROM PROJECTS).
- Oracle Database kích hoạt Policy Function (Hàm chính sách) gắn với bảng đó.
- Hàm này trả về một chuỗi điều kiện (predicate), ví dụ: DEPARTMENT_ID = 'IT'.
- Oracle tự động nối chuỗi điều kiện này vào mệnh đề WHERE của câu truy vấn gốc (trở thành SELECT * FROM PROJECTS WHERE DEPARTMENT_ID = 'IT').
- Câu lệnh đã sửa đổi được thực thi và trả về kết quả đã lọc.

Toàn bộ quá trình này diễn ra trong suốt (transparent) đối với người dùng và ứng dụng, giúp đảm bảo an toàn dữ liệu ngay cả khi ứng dụng bị tấn công SQL Injection. Các gói (package) chính được sử dụng là DBMS_RLS để quản lý chính sách và SYS_CONTEXT để lấy thông tin phiên làm việc của người dùng.

2.4. Giám sát và Kiểm toán (Unified Auditing & FGA)

Auditing (Kiểm toán) là quá trình giám sát và ghi lại các hoạt động diễn ra trong cơ sở dữ liệu nhằm mục đích an ninh, tuân thủ quy định và điều tra sự cố.

2.4.1. Unified Auditing

Trong các phiên bản Oracle cũ, dữ liệu audit nằm rải rác ở nhiều nơi (AUD$, FGA_LOG$, OS files). Từ phiên bản 12c trở đi, Oracle giới thiệu Unified Auditing, hợp nhất tất cả các bản ghi vào một khung nhìn duy nhất là UNIFIED_AUDIT_TRAIL.

Unified Auditing có hiệu năng cao hơn và cho phép tạo các chính sách (Audit Policy) linh hoạt, có thể giám sát cả hành động DDL (thay đổi cấu trúc) và DML (thay đổi dữ liệu) dựa trên role, user hoặc ngữ cảnh cụ thể.

2.4.2. Fine-Grained Auditing (FGA)

Khác với Standard Auditing (giám sát thô, ví dụ: ghi lại mọi lệnh SELECT trên bảng), Fine-Grained Auditing (FGA) cho phép giám sát "tinh" hơn để giảm thiểu lượng log rác.

FGA sử dụng gói DBMS_FGA để định nghĩa các chính sách dựa trên nội dung:

- Audit Condition: Chỉ ghi log khi thỏa mãn điều kiện (ví dụ: SALARY > 10000000).
- Audit Column: Chỉ ghi log khi người dùng truy cập vào cột nhạy cảm cụ thể (ví dụ: cột SALARY hoặc COMMISSION).

FGA cung cấp thông tin chi tiết bao gồm câu lệnh SQL thực tế và giá trị của các biến bind, giúp ích rất lớn cho việc phát hiện gian lận.

2.5. Che giấu dữ liệu (Data Redaction)

Oracle Data Redaction cung cấp khả năng che giấu (masking) dữ liệu nhạy cảm (như số thẻ tín dụng, số căn cước, số điện thoại) khi dữ liệu được trả về cho ứng dụng, trong khi dữ liệu gốc lưu trữ trên đĩa vẫn giữ nguyên.

Tính năng này hoạt động ngay tại lớp Database (on-the-fly redaction) khi truy vấn được thực thi, giúp bảo vệ dữ liệu mà không cần thay đổi mã nguồn ứng dụng. Oracle hỗ trợ các phương thức che giấu:

- Full Redaction: Che toàn bộ giá trị (trả về NULL hoặc khoảng trắng).
- Partial Redaction: Che một phần (ví dụ: XXXX-XXXX-XXXX-1234).
- Regular Expression Redaction: Sử dụng biểu thức chính quy để tìm và che các mẫu dữ liệu (như định dạng email).
- Random Redaction: Thay thế bằng giá trị ngẫu nhiên.

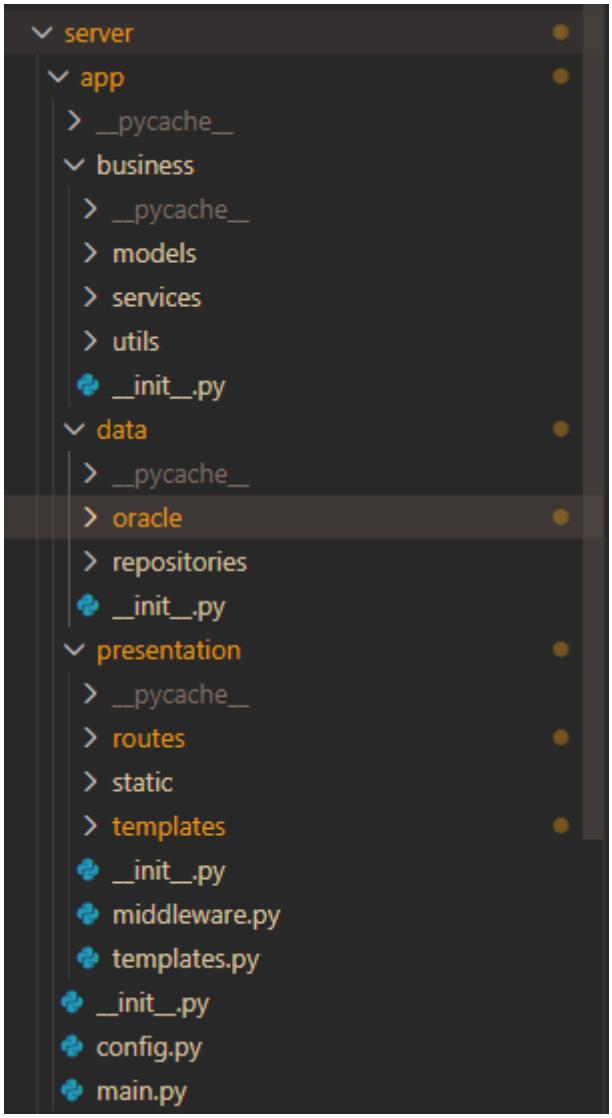
Sự kết hợp giữa VPD (kiểm soát dòng nào được thấy) và Data Redaction (kiểm soát cột nào bị che) tạo nên một lớp bảo vệ toàn diện cho dữ liệu nhạy cảm.

CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

Chương này trình bày chi tiết về kiến trúc tổng thể, thiết kế cơ sở dữ liệu và quá trình hiện thực các chức năng quản trị cũng như các chính sách bảo mật trên Oracle Database 23ai.

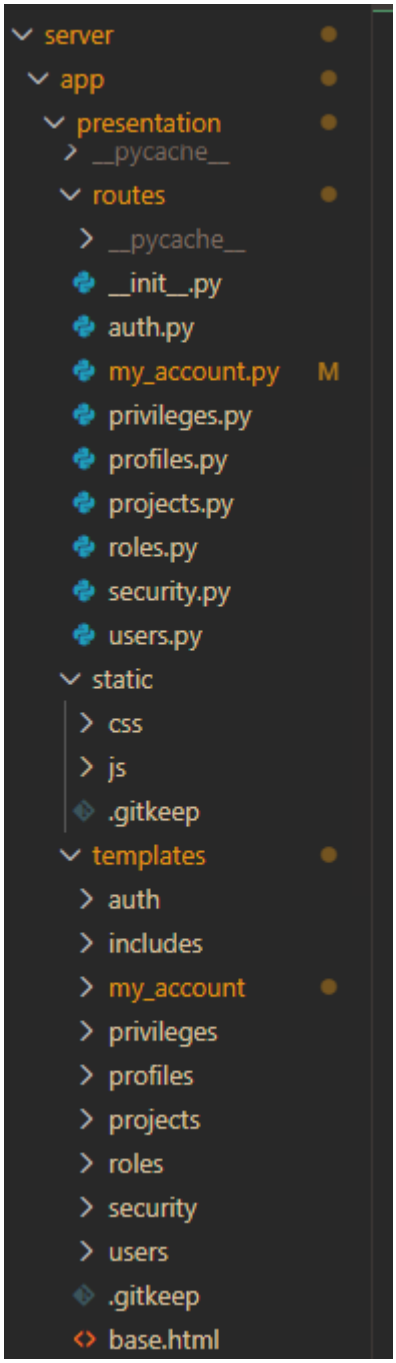
3.1. Kiến trúc hệ thống (Mô hình 3 lớp)

Hệ thống được xây dựng theo mô hình kiến trúc 3 lớp (3-Layer Architecture) chuẩn mực, giúp tách biệt rõ ràng giữa giao diện, xử lý nghiệp vụ và truy xuất dữ liệu. Điều này đảm bảo tính bảo mật, dễ dàng bảo trì và mở rộng.



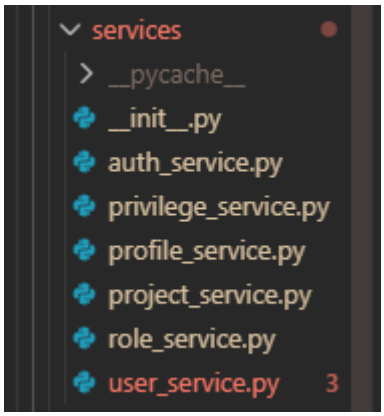
- **Lớp Presentation (Giao diện & API):**
    - Thành phần được xây dựng bằng FastAPI Router và Jinja2 Templates.
    - Có chức năng tiếp nhận yêu cầu HTTP từ người dùng (Browser), thực hiện xác thực cơ bản (Authentication) và hiển thị dữ liệu dưới dạng trang web HTML.

- Đường dẫn thư mục: server/app/presentation/routes/ và server/app/presentation/templates/.

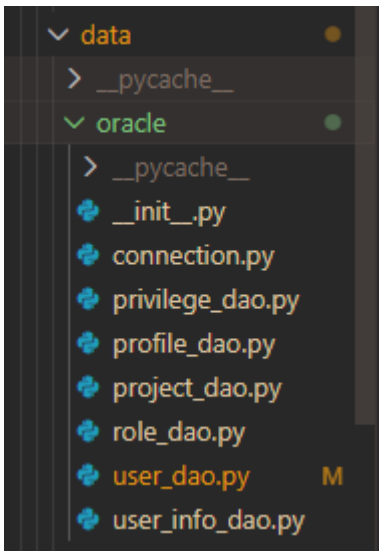


- **Lớp Business (Nghiệp vụ):**
  - Thành phần bao gồm các Service Classes (ví dụ: UserService, SecurityService).
  - Chức năng chứa logic nghiệp vụ chính của ứng dụng. Lớp này không trực tiếp truy vấn SQL mà gọi xuống lớp Data để lấy dữ liệu, sau đó xử lý (ví dụ: kiểm tra quy tắc mật khẩu, định dạng dữ liệu audit log) trước khi trả về cho lớp Presentation.

- Đường dẫn thư mục: server/app/business/services/.



- **Lớp Data Access (Truy xuất dữ liệu):**
  - Thành phần bao gồm các DAO (Data Access Object) sử dụng thư viện python-oracledb.
  - Chức năng nhằm trực tiếp thực thi các câu lệnh SQL/PLSQL xuống Oracle Database. Đây là nơi duy nhất trong mã nguồn ứng dụng tương tác với cơ sở dữ liệu.
  - Đường dẫn thư mục: server/app/data/oracle/.



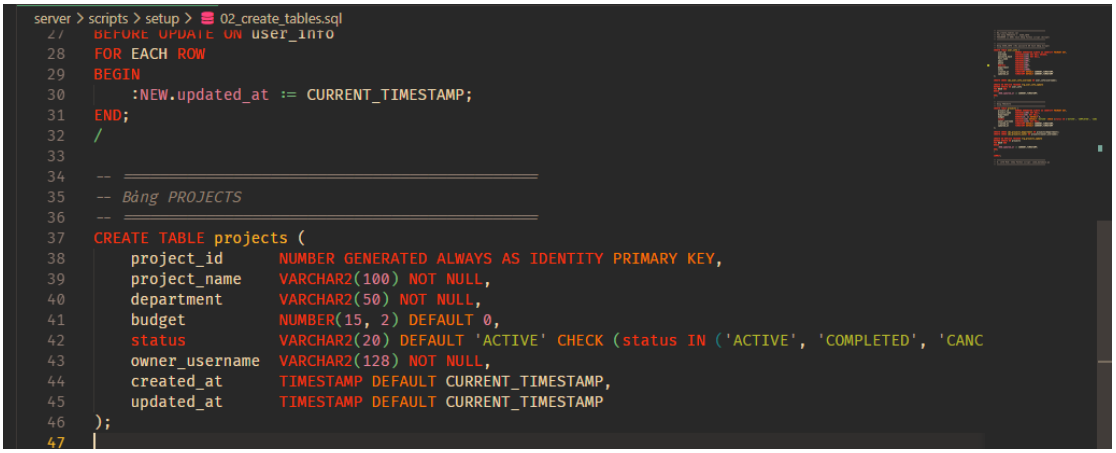
3.2. Thiết kế cơ sở dữ liệu

Dữ liệu của hệ thống được tổ chức dựa trên kiến trúc Multitenant của Oracle 23ai, chạy trên một Pluggable Database (PDB) cụ thể (FREEPDB1).

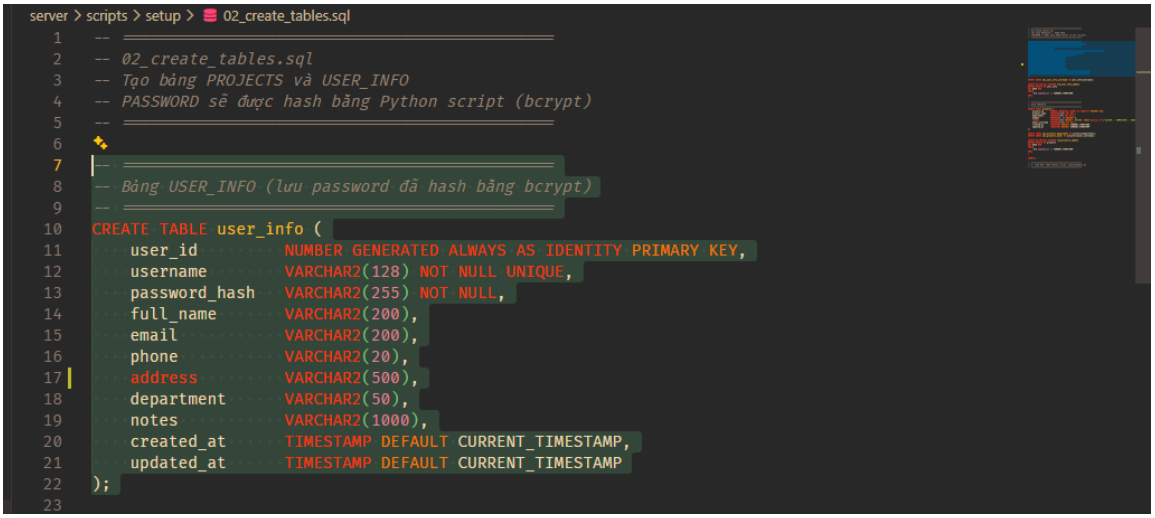
3.2.1. Mô hình dữ liệu nghiệp vụ

Hệ thống sử dụng schema SYSTEM (hoặc một schema quản trị riêng như APP_ADMIN) để chứa các bảng nghiệp vụ chính:

1. **Bảng PROJECTS (Dự án):**
    - Lưu trữ thông tin các dự án nội bộ của công ty. Đây là đối tượng chính để áp dụng các chính sách bảo mật VPD và FGA.
    - Các cột chính: PROJECT_ID (PK), PROJECT_NAME, DEPARTMENT (Phòng ban: 'IT', 'SALES'...), BUDGET (Ngân sách - Dữ liệu nhạy cảm).



2. **Bảng USER_INFO (Thông tin nhân viên):**
    - Lưu trữ thông tin bổ sung của nhân viên phục vụ cho việc demo Data Redaction.
    - Các cột chính: USER_ID, FULL_NAME, EMAIL (Nhạy cảm), PHONE_NUMBER (Nhạy cảm).



3.2.2. Các View quản trị hệ thống (Data Dictionary Views)

Thay vì tạo bảng riêng để quản lý user database, hệ thống tận dụng sức mạnh của các View hệ thống có sẵn trong Oracle để quản trị trực tiếp:

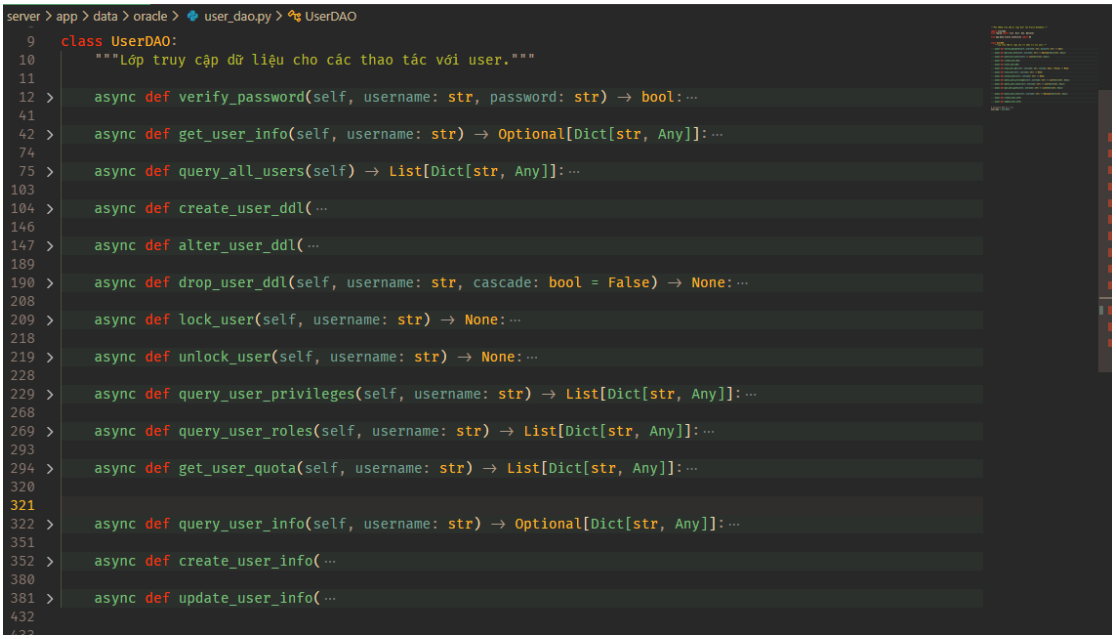
- DBA_USERS: Tra cứu thông tin người dùng, trạng thái (OPEN/LOCKED), ngày hết hạn mật khẩu.
- DBA_PROFILES: Quản lý các chính sách tài nguyên và mật khẩu.
- DBA_ROLES / DBA_SYS_PRIVS: Quản lý quyền hạn hệ thống.
- UNIFIED_AUDIT_TRAIL: View tổng hợp logs từ hệ thống Unified Auditing.

3.3. Hiện thực module quản trị (Admin Module)

3.3.1. Quản lý vòng đời User và Profile

Chức năng này cho phép quản trị viên thao tác trực tiếp với các tài khoản Oracle Database thật sự.

- **Xử lý nghiệp vụ (Backend):**
  - File user_dao.py thực thi các lệnh DDL như CREATE USER, ALTER USER.
  - Hàm create_user thực hiện cấp phát DEFAULT TABLESPACE là USERS và TEMPORARY TABLESPACE là TEMP, đồng thời gán Quota (ví dụ: 500M) để giới hạn dung lượng lưu trữ.
  - Hàm lock_user / unlock_user sử dụng câu lệnh ALTER USER ... ACCOUNT LOCK/UNLOCK để kiểm soát quyền đăng nhập ngay lập tức.



- **Quản lý Profile:**
  - Hệ thống cho phép tạo các Profile tùy chỉnh (ví dụ: PROF_LIMIT) để áp đặt chính sách bảo mật.

3.3.2. Quản lý Role và Phân quyền

Module này hiện thực mô hình RBAC (Role-Based Access Control) để đơn giản hóa việc quản trị quyền hạn.

- **Quản lý Role:**
    - Tạo Role mới (ví dụ: ROLE_NHANVIEN, ROLE_MANAGER) thông qua role_dao.py.
    - Gán các quyền hệ thống (CREATE SESSION, CREATE TABLE) hoặc quyền đối tượng (SELECT ON PROJECTS) cho Role thay vì gán trực tiếp cho từng user.

- **Cấp quyền (Grant/Revoke):**
    - Giao diện web cho phép chọn User và Role/Privilege cần cấp.
    - Hệ thống thực thi lệnh GRANT role_name TO user_name ở tầng database.

3.4. Hiện thực các giải pháp bảo mật nâng cao

Đây là phần trọng tâm của đề tài, áp dụng các công nghệ "Security Inside the Database".

3.4.1. Cấu hình Unified Auditing và FGA

Hệ thống chuyển đổi hoàn toàn sang cơ chế Unified Auditing hiện đại của Oracle 23ai để ghi vết tập trung.

- **Audit DDL (Thay đổi cấu trúc):**

- Tạo chính sách audit_user_admin để giám sát mọi hành động tạo, sửa, xóa người dùng.
- SQL: CREATE AUDIT POLICY audit_user_admin ACTIONS CREATE USER, ALTER USER, DROP USER;

server > scripts > setup > 04_create_audit.sql

```
12  -- ═══════════════════════════════════════════════
13
14  -- Xóa policy cũ nếu có
15  BEGIN
16      EXECUTE IMMEDIATE 'DROP AUDIT POLICY audit_projects_dml';
17  EXCEPTION
18      WHEN OTHERS THEN NULL;
19  END;
20  /
21
22  -- Tạo unified audit policy cho DML trên PROJECTS
23  CREATE AUDIT POLICY audit_projects_dml
24      ACTIONS SELECT, INSERT, UPDATE, DELETE ON SYSTEM.PROJECTS;
25
26  -- Enable audit policy
27  AUDIT POLICY audit_projects_dml;
28
29  -- ═══════════════════════════════════════════════
30  -- 1b. Unified Auditing Policy cho User Management using ACTIONS
31  -- Audit CREATE/ALTER/DROP USER/ROLE/PROFILE
32  -- ═══════════════════════════════════════════════
33  BEGIN
34      EXECUTE IMMEDIATE 'DROP AUDIT POLICY audit_user_admin';
35  EXCEPTION
36      WHEN OTHERS THEN NULL;
37  END;
38  /
39
40  CREATE AUDIT POLICY audit_user_admin
41      ACTIONS CREATE USER, ALTER USER, DROP USER,
42              CREATE ROLE, ALTER ROLE, DROP ROLE,
43              CREATE PROFILE, ALTER PROFILE, DROP PROFILE;
44
45  AUDIT POLICY audit_user_admin;
```

- **Fine-Grained Auditing (FGA) - Giám sát dữ liệu nhạy cảm:**
  - Chỉ ghi log khi có người cố tình sửa đổi cột Ngân sách (BUDGET) với giá trị lớn.
  - Sử dụng gói DBMS_FGA.

```
BEGIN
  DBMS_FGA.ADD_POLICY(
    object_schema   => 'SYSTEM',
    object_name     => 'PROJECTS',
    policy_name     => 'AUDIT_HIGH_BUDGET',
    audit_condition => 'BUDGET > 1000000000', -- Chỉ audit dự án > 1 tỷ
    audit_column    => 'BUDGET',  -- Chỉ khi đụng vào cột tiền
```

```
    statement_types => 'INSERT,UPDATE'
  );
END;
```

server > scripts > setup > 04_create_audit.sql

```
46
47  -- ═══════════════════════════════════════════════
48  -- 2. Fine-Grained Auditing (FGA) cho cột BUDGET
49  -- Audit khi có ai đọc/sửa budget
50  -- ═══════════════════════════════════════════════
51  BEGIN
52      -- Xóa policy FGA cũ nếu có
53      BEGIN
54          DBMS_FGA.DROP_POLICY(
55              object_schema => 'SYSTEM',
56              object_name => 'PROJECTS',
57              policy_name => 'AUDIT_BUDGET_ACCESS'
58          );
59      EXCEPTION
60          WHEN OTHERS THEN NULL;
61      END;
62
63      -- FGA cho SELECT trên cột BUDGET
64      DBMS_FGA.ADD_POLICY(
65          object_schema => 'SYSTEM',
66          object_name => 'PROJECTS',
67          policy_name => 'AUDIT_BUDGET_ACCESS',
68          audit_column => 'BUDGET',
69          audit_condition => NULL,
70          statement_types => 'SELECT',
71          audit_trail => DBMS_FGA.DB + DBMS_FGA.EXTENDED,
72          audit_column_opts => DBMS_FGA.ANY_COLUMNS
73      );
74  END;
75  /
76
77  BEGIN
78      -- Xóa policy FGA cũ nếu có
79      BEGIN
```

3.4.2. Triển khai chính sách VPD phân quyền theo phòng ban

Virtual Private Database (VPD) được sử dụng để đảm bảo nhân viên phòng ban nào chỉ thấy dữ liệu của phòng ban đó.

- **Bước 1: Hàm chính sách (Policy Function):**

- Viết hàm PL/SQL vpd_function để kiểm tra ngữ cảnh người dùng và trả về chuỗi điều kiện WHERE DEPARTMENT = '...'.

- **Bước 2: Gắn chính sách vào bảng:**
  - Sử dụng DBMS_RLS.ADD_POLICY để kích hoạt hàm chính sách trên bảng PROJECTS.
  - Kết quả: Khi User IT truy vấn SELECT * FROM PROJECTS, Oracle tự động thêm WHERE DEPARTMENT = 'IT', làm cho dữ liệu của phòng Sales "tàng hình" đối với họ.

server > scripts > setup > 03_create_vpd_policy.sql

```
46  -- ═══════════════════════════════════════════
47  -- 4. VPD Policy Function
48  -- Kiểm tra app_user_ctx trước, sau đó mới SESSION_USER
49  -- ═══════════════════════════════════════════
50  CREATE OR REPLACE FUNCTION vpd_projects_policy (
51      schema_name IN VARCHAR2,
52      table_name IN VARCHAR2
53  ) RETURN VARCHAR2 AS
54      v_app_user VARCHAR2(128);
55      v_session_user VARCHAR2(128);
56      v_department VARCHAR2(50);
57  BEGIN
58      -- Lấy app user từ application context
59      v_app_user := SYS_CONTEXT('app_user_ctx', 'current_user');
60      v_session_user := SYS_CONTEXT('USERENV', 'SESSION_USER');
61
62      -- Nếu có app user trong context, dùng nó để filter
63      IF v_app_user IS NOT NULL THEN
64          -- ADMIN và SYSTEM có thể thấy tất cả
65          IF v_app_user IN ('ADMIN', 'SYSTEM') THEN
66              RETURN NULL; -- No restriction
67          END IF;
68
69          -- Lấy department từ context (nếu có)
70          v_department := SYS_CONTEXT('user_dept_ctx', 'department');
71
72          IF v_department IS NOT NULL THEN
73              -- Filter theo department HOẶC owner
74              RETURN 'department = ''' || v_department || ''' OR owner_username = ''' || v_app_user || '''';
75          ELSE
76              -- Chỉ thấy projects của mình
77              RETURN 'owner_username = ''' || v_app_user || '''';
78          END IF;
```

3.4.3. Áp dụng Data Redaction cho thông tin nhạy cảm

Tính năng này giúp che giấu dữ liệu cá nhân (PII) trên giao diện ứng dụng mà không làm thay đổi dữ liệu gốc.

- Che giấu số điện thoại và Email đối với các user thường, chỉ hiển thị cho nhân sự (HR).
- Sử dụng DBMS_REDACT với chế độ PARTIAL và REGEXP.

```
BEGIN
DBMS_REDACT.ADD_POLICY(
    object_schema => 'SYSTEM',
    object_name   => 'USER_INFO',
    column_name   => 'PHONE',
```

```
    function_type => DBMS_REDACT.PARTIAL,
    function_parameters => '9,1,3' -- Chỉ hiển thị 3 số cuối (******789)
);
END
```

server > scripts > setup > 06_configure_redaction.sql

```
BEGIN
    END IF;

    -- 1. Tạo Policy và che cột PHONE (FULL Redaction)
    DBMS_REDACT.ADD_POLICY(
        object_schema ⇒ 'SYSTEM',
        object_name   ⇒ 'USER_INFO',
        column_name   ⇒ 'PHONE',
        policy_name   ⇒ 'REDACT_USER_INFO_POLICY',
        function_type ⇒ DBMS_REDACT.FULL,
        expression    ⇒ '1=1',
        policy_description ⇒ 'Che số điện thoại và email của người dùng'
    );

    -- 2. Thêm cột EMAIL vào policy (FULL Redaction)
    DBMS_REDACT.ALTER_POLICY(
        object_schema ⇒ 'SYSTEM',
        object_name   ⇒ 'USER_INFO',
        policy_name   ⇒ 'REDACT_USER_INFO_POLICY',
        action        ⇒ DBMS_REDACT.ADD_COLUMN,
        column_name   ⇒ 'EMAIL',
        function_type ⇒ DBMS_REDACT.FULL
    );

    COMMIT;
END;
/
```

CHƯƠNG IV: KẾT QUẢ THỰC NGHIỆM VÀ DEMO

Chương này trình bày chi tiết về môi trường triển khai thực tế và kết quả của các kịch bản kiểm thử. Các kịch bản này được thiết kế để minh chứng cho khả năng hoạt động của hệ thống trong việc quản trị người dùng và hiệu quả của các cơ chế bảo mật "Security Inside the Database" đã được hiện thực.

4.1. Môi trường triển khai

Hệ thống được triển khai và vận hành trên môi trường container hóa để đảm bảo tính đồng nhất và dễ dàng tái lập.

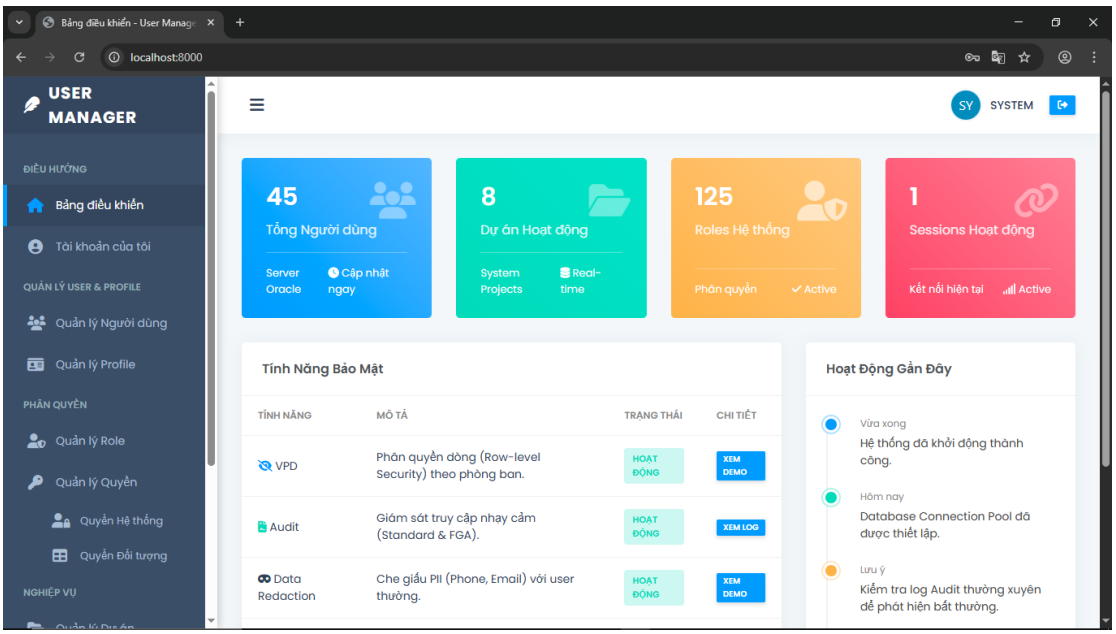
- Hệ điều hành: Windows 10.
- Container Platform: Docker Desktop.
- Database Container:
    - Image: container-registry.oracle.com/database/free:23ai (Oracle Database 23ai Free).
    - Port Mapping: 1521:1521 (Cho phép kết nối từ Host).
    - Resource: Giới hạn 2 CPU, 4GB RAM.
- Application Server:
    - Runtime: Python 3.12.
    - Server: Uvicorn (ASGI Server) chạy tại http://localhost:8000.

4.2. Quản lý vòng đời User và Profile

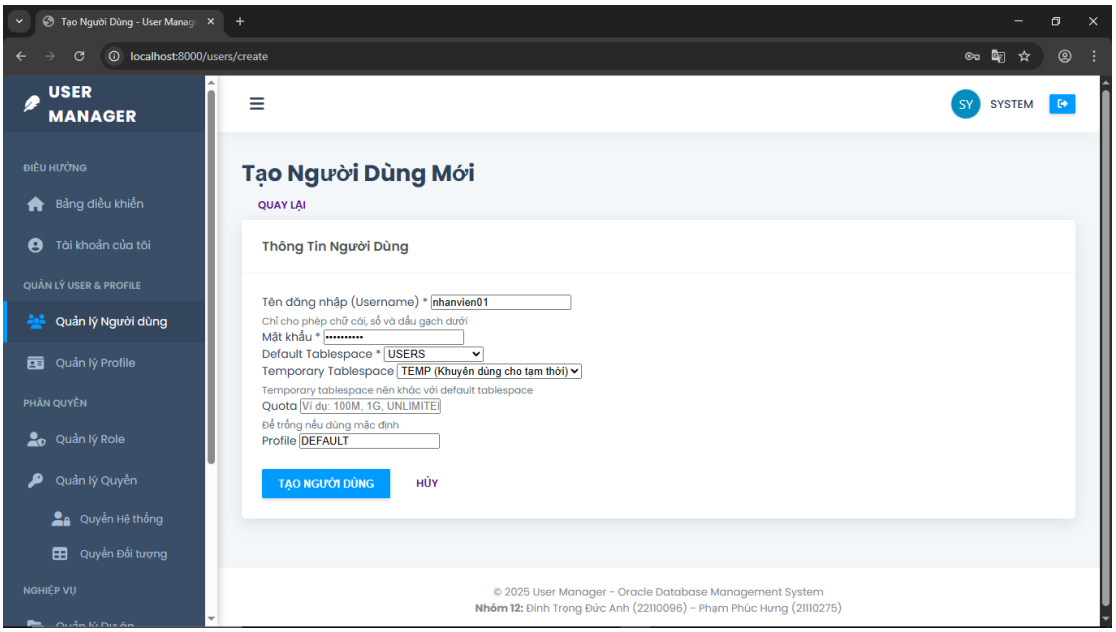
Kiểm tra khả năng tạo, sửa, khóa tài khoản và áp dụng chính sách tài nguyên thực tế trên Oracle Database thông qua giao diện Web.

Các bước thực hiện:

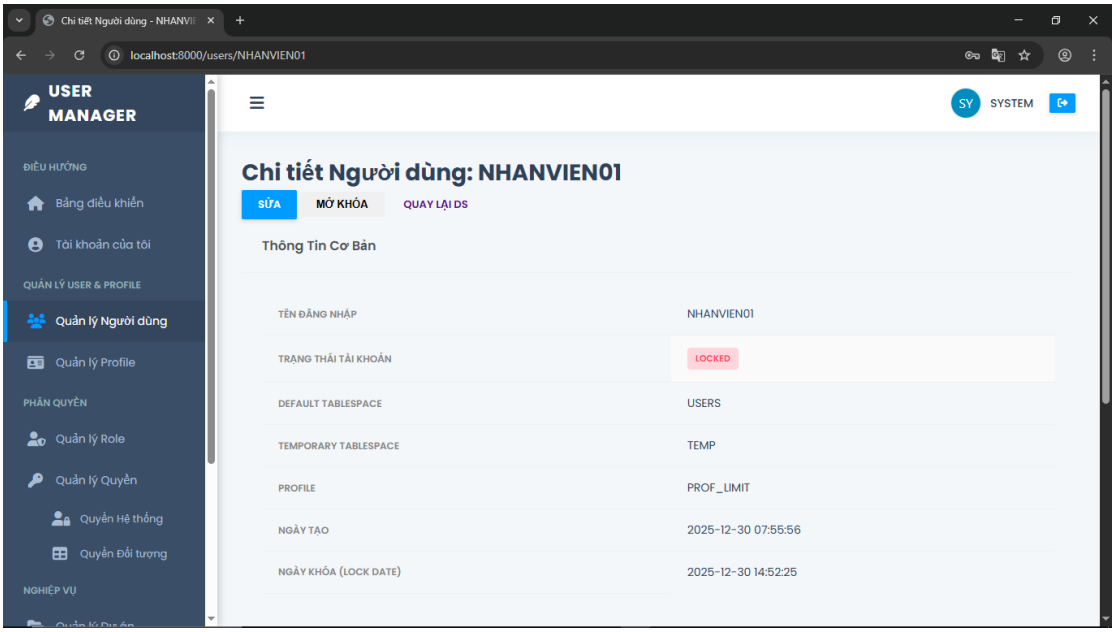
1. Đăng nhập vào hệ thống với quyền Admin



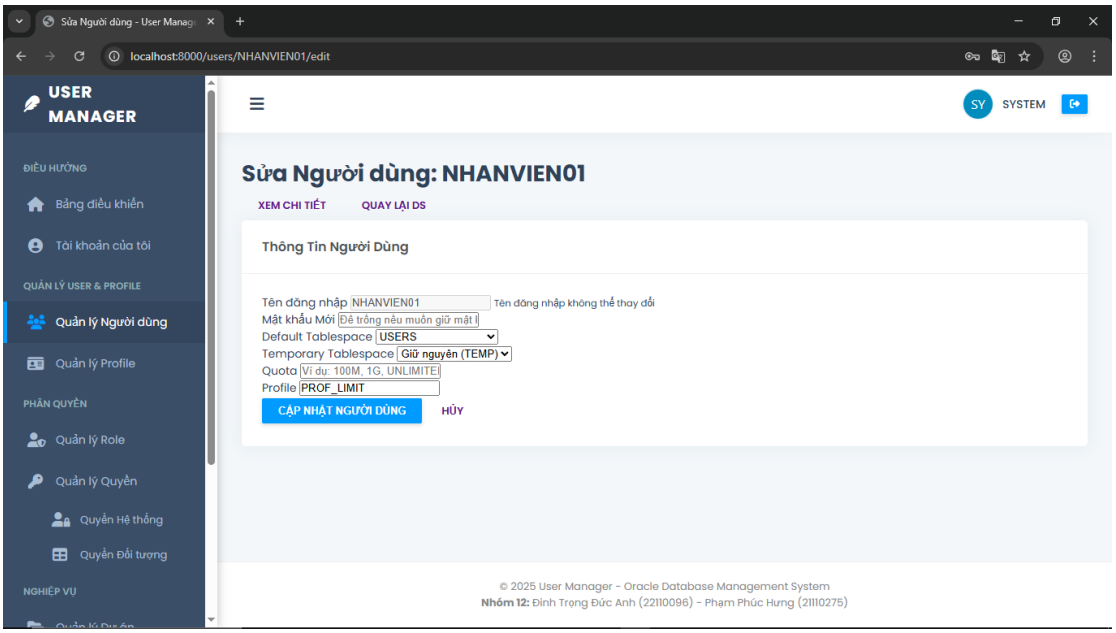
2. Tạo người dùng mới với thông tin:
    - Username: nhanvien01
    - Profile: DEFAULT
    - Tablespace: USERS



3. Thực hiện khóa tài khoản (Lock User) trên giao diện Web.



4. Gán Profile giới hạn PROF_LIMIT (Giới hạn 1 session) cho user này.



Kết quả thực nghiệm:

- Trên giao diện Web: Trạng thái user chuyển sang "LOCKED".
- Kiểm tra trực tiếp trong Database (SQL*Plus):

```
SELECT username, account_status, profile FROM dba_users WHERE username = 'NHANVIEN01';
```

- Kết quả trả về: ACCOUNT_STATUS = 'LOCKED', PROFILE = 'PROF_LIMIT'.

```
SELECT username, account_status, profile FROM dba_users WHERE username = 'NHANVIEN01';

USERNAME
--------------------------------------------------------------------------------
ACCOUNT_STATUS
--------------------------------
PROFILE
--------------------------------------------------------------------------------
NHANVIEN01
LOCKED
PROF_LIMIT
```

- Kết luận: Các thao tác trên Web đã tác động chính xác xuống tầng Database vật lý.

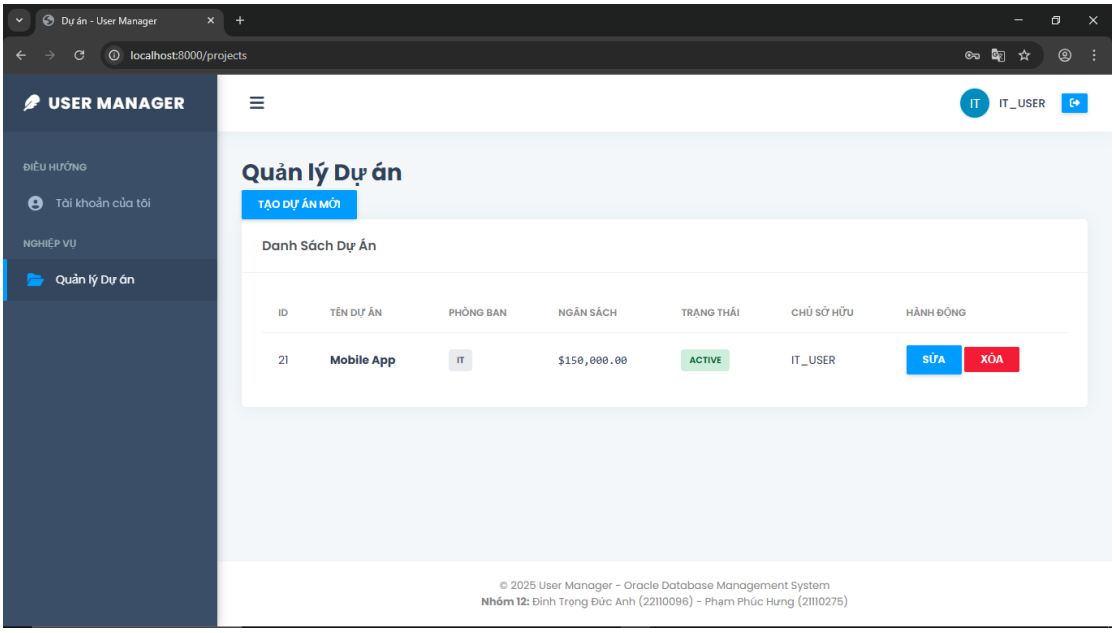
4.3. Phân quyền dữ liệu với VPD

Chứng minh tính năng bảo mật mức dòng (Row-Level Security), đảm bảo người dùng chỉ nhìn thấy dữ liệu thuộc phạm vi quyền hạn của mình.

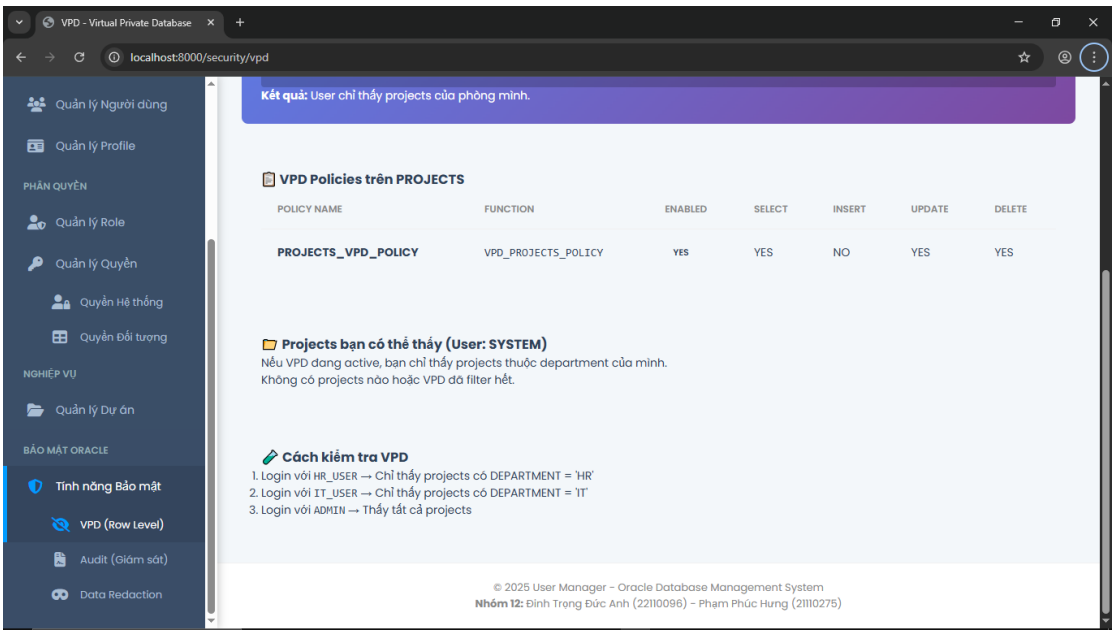
Dữ liệu chuẩn bị: Bảng PROJECTS chứa các dự án của phòng 'IT' và 'SALES'.

Các bước thực hiện:

1. Đăng nhập (giả lập) với tài khoản it_user (Trưởng phòng IT).
    - Kết quả: Hệ thống chỉ hiển thị danh sách các dự án có DEPARTMENT = 'IT'.



2. Đăng nhập (giả lập) với tài khoản marketing_user (Trưởng phòng Marketing).
    - Kết quả: Hệ thống chỉ hiển thị danh sách các dự án có DEPARTMENT = 'MARKETING'.



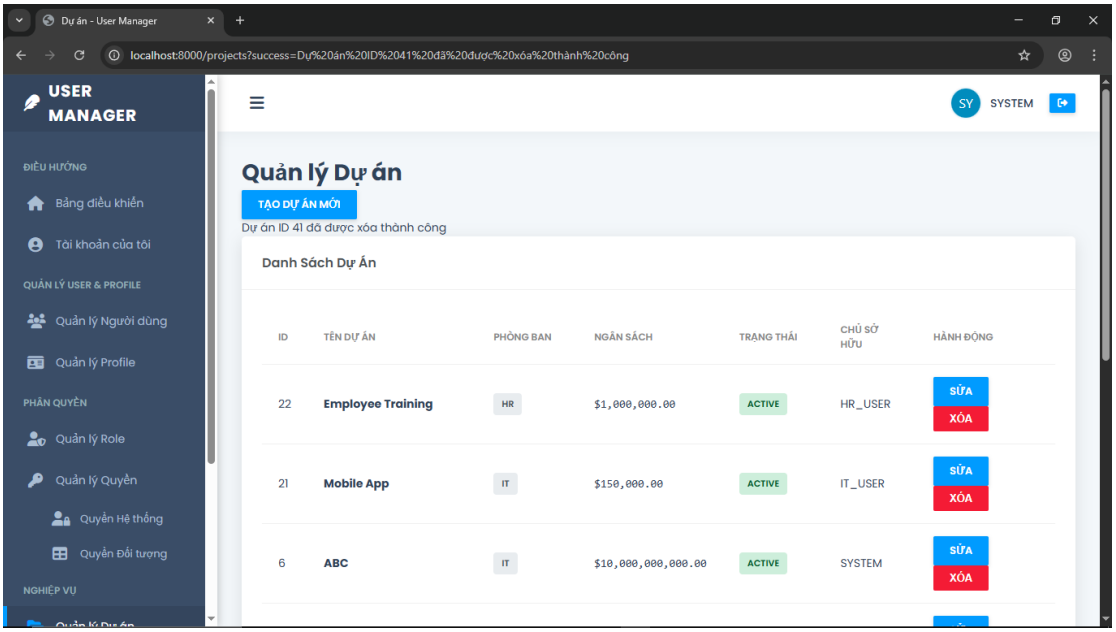
Hệ thống không cần viết code lọc WHERE trong ứng dụng. Oracle Database đã tự động "tiêm" vị từ (predicate) vào câu lệnh SQL dựa trên Context của session đăng nhập, ngăn chặn triệt để việc truy cập chéo dữ liệu giữa các phòng ban.

4.4. Giám sát an ninh (Unified Auditing & FGA)

Kiểm tra khả năng ghi vết các hành động quản trị (DDL) và các truy cập nhạy cảm (DML) vào vùng dữ liệu quan trọng.

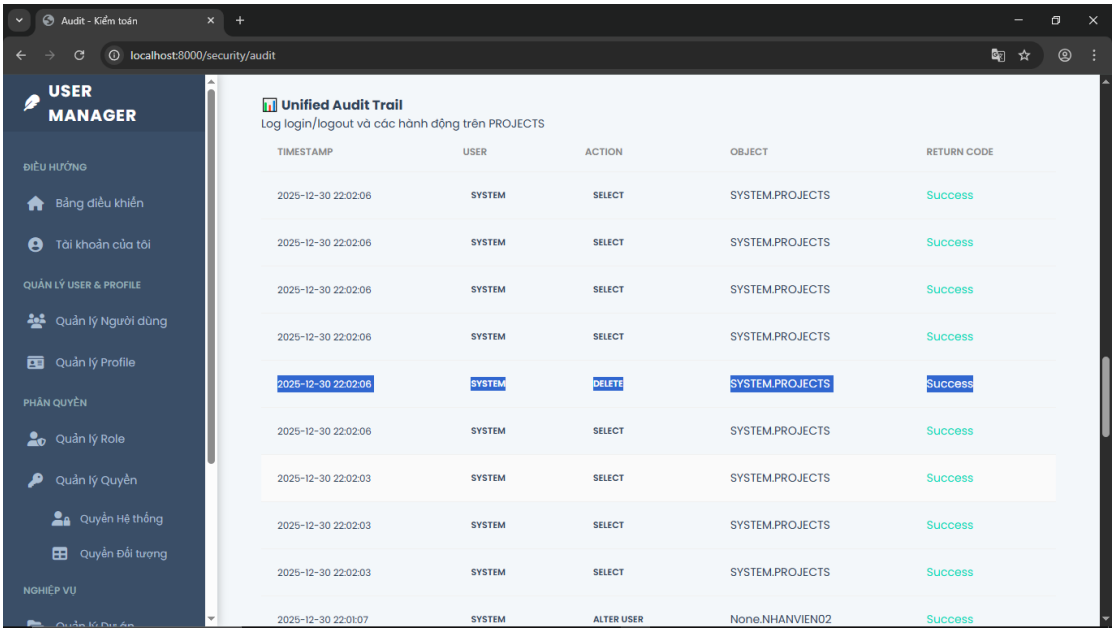
Phần A: Unified Auditing (Giám sát DDL)

1. Admin thực hiện xóa (Drop) một dự án mẫu trên giao diện Web.



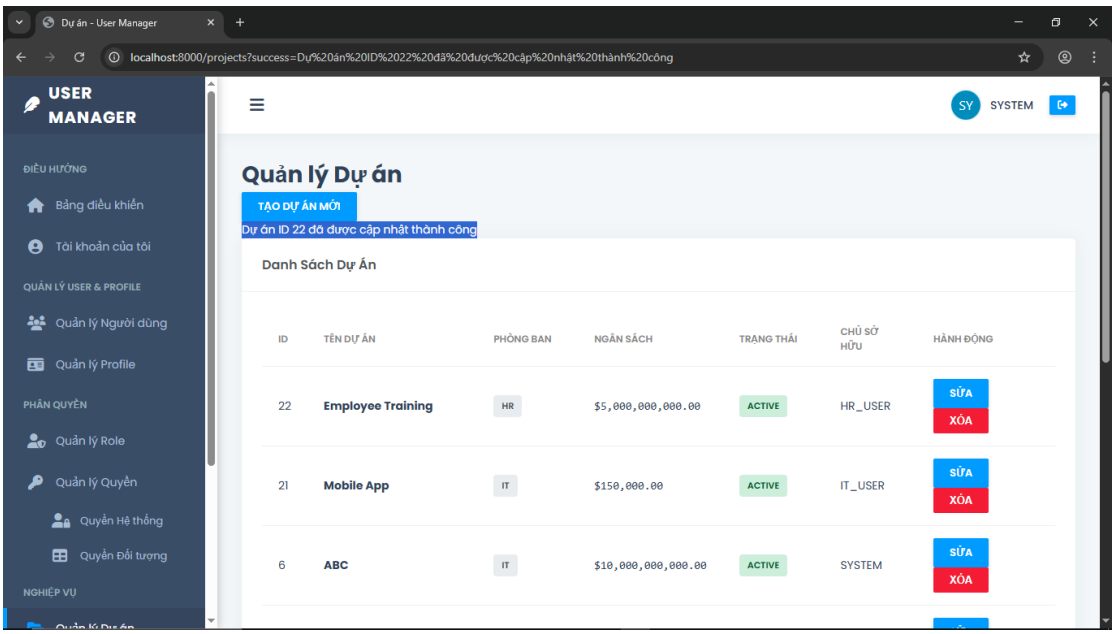
2. Truy cập menu "Audit Logs".

3. Kết quả: Hệ thống hiển thị dòng log mới nhất với ACTION_NAME = DELETE', ghi rõ thời gian và người thực hiện.

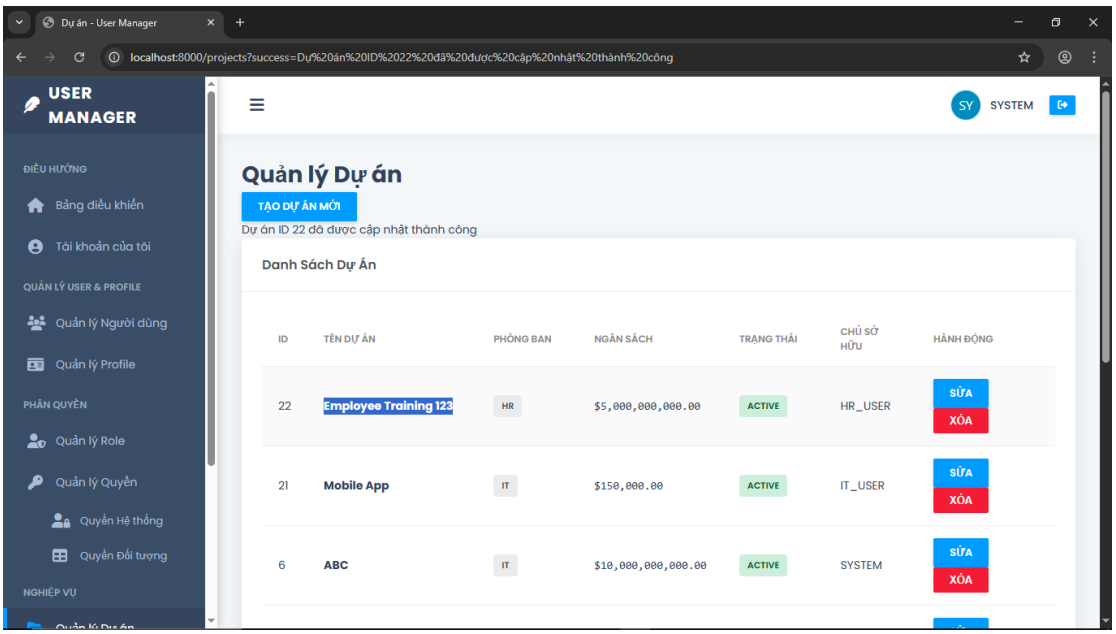


Phần B: Fine-Grained Auditing (Giám sát thay đổi Ngân sách)

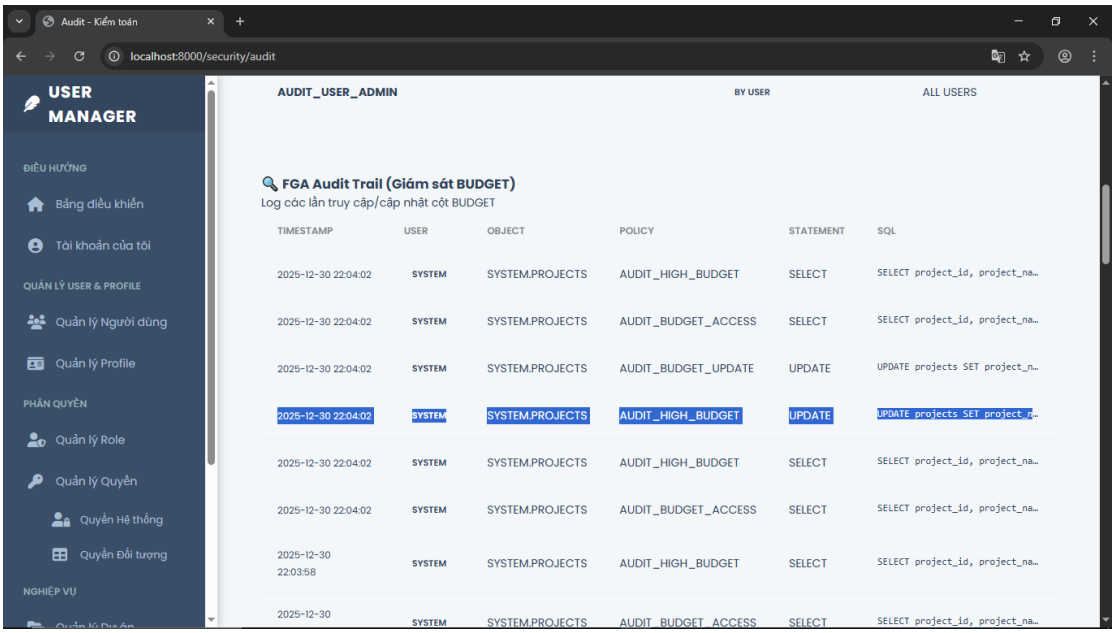
1. User thực hiện cập nhật cột BUDGET của một dự án (Ví dụ: Tăng vốn từ 1 tỷ lên 5 tỷ).



2. User thực hiện cập nhật cột PROJECT_NAME (Không nhạy cảm).



3. Kiểm tra Audit Log.
4. Kết quả:
    - Chỉ hành động cập nhật BUDGET bị ghi lại.
    - Hành động cập nhật tên dự án không bị ghi log (giúp giảm thiểu rác log, tối ưu hiệu năng).
    - Log FGA hiển thị chi tiết câu lệnh SQL đã thực thi.

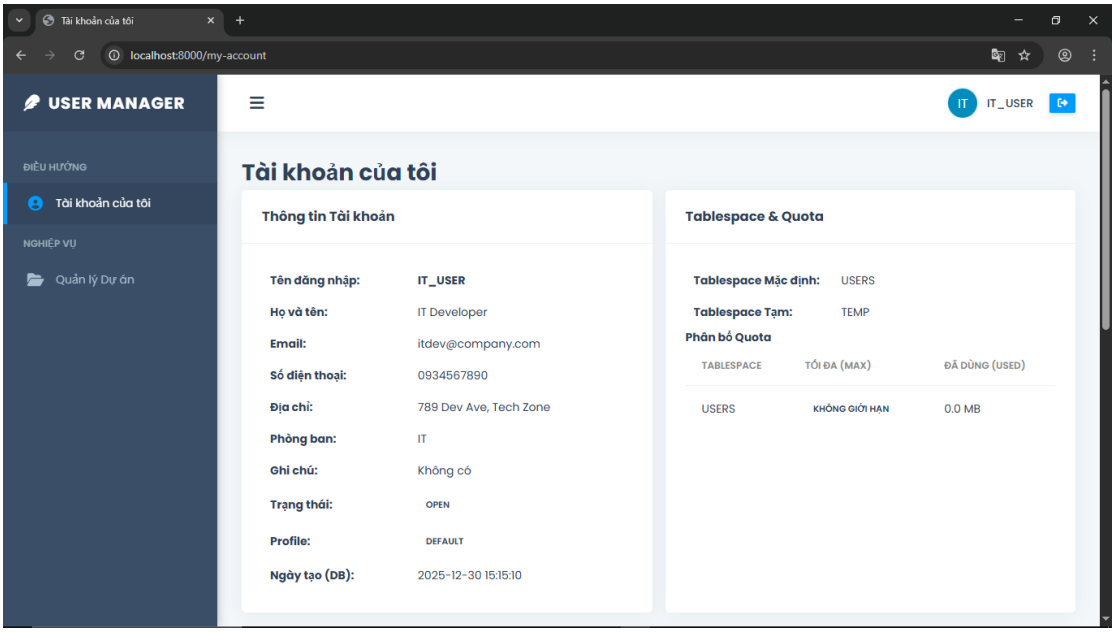


4.5. Che giấu dữ liệu (Data Redaction)

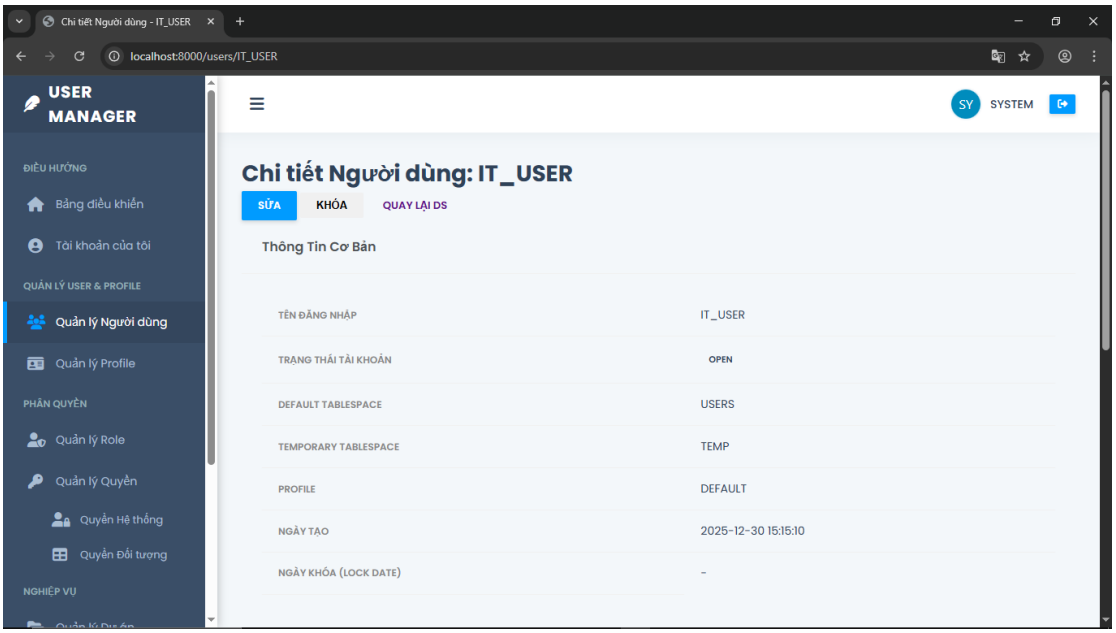
Bảo vệ thông tin cá nhân (PII) như số điện thoại, email khỏi sự tò mò của những người dùng không có phận sự (ví dụ: nhân viên IT vận hành hệ thống).

Các bước thực hiện:

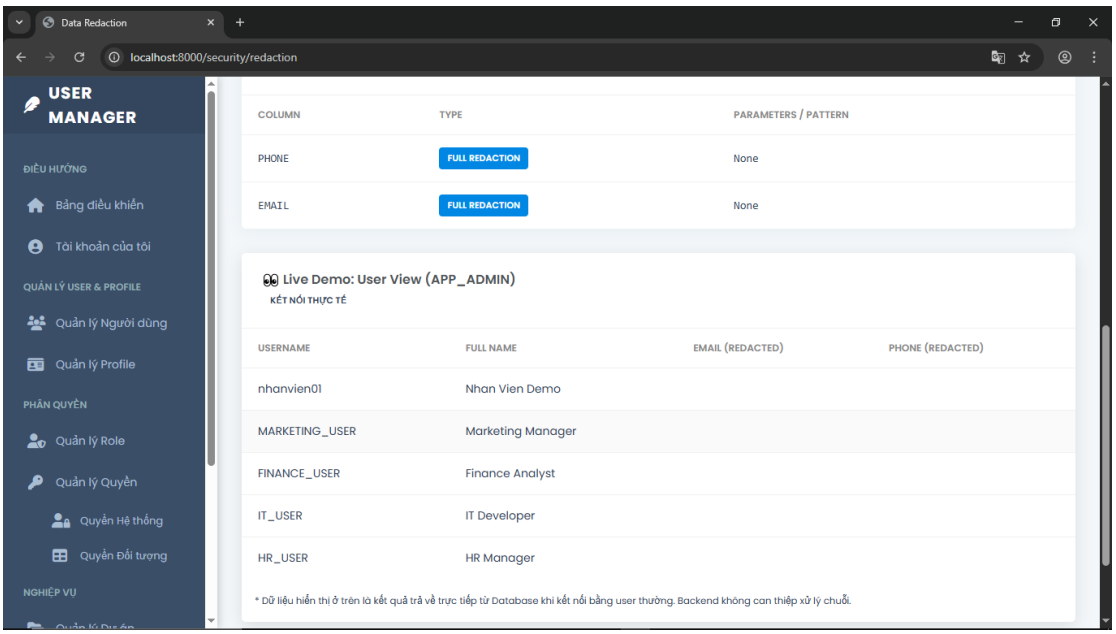
1. Đăng nhập với tài khoản có quyền IT_USER.
    - Kết quả: Nhìn thấy đầy đủ số điện thoại và email của nhân viên (Ví dụ: 0901234567).



2. Đăng nhập với tài khoản thường nhanvien01 hoặc sys_admin (Admin hệ thống nhưng không phải HR).
    - Xem danh sách nhân viên.



    - Kết quả:
        - Cột Email bị che giấu
        - Cột Số điện thoại bị che giấu



Chính sách Data Redaction hoạt động hiệu quả ngay tại tầng Database ("On-the-fly"), đảm bảo dữ liệu trả về cho ứng dụng đã được làm sạch mà không cần thay đổi dữ liệu gốc trên đĩa cứng.

CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

Sau quá trình nghiên cứu và thực hiện đề tài **" XÂY DỰNG ỨNG DỤNG WEB THEO MÔ HÌNH 3 LỚP QUẢN TRỊ NGƯỜI DÙNG TẬP TRUNG VÀ BẢO MẬT TRÊN DỮ LIỆU DỰ ÁN CỦA CÔNG TY"**, nhóm đã hoàn thành các mục tiêu đề ra và đạt được những kết quả cụ thể sau:

1. **Về mặt ứng dụng quản trị:**
    - Xây dựng thành công ứng dụng web dựa trên nền tảng Python FastAPI và Oracle Database 23ai, cung cấp giao diện trực quan cho việc quản trị cơ sở dữ liệu.
    - Hiện thực hóa quy trình quản lý vòng đời người dùng (User Lifecycle) trọn vẹn: từ lúc tạo mới, cấp phát tài nguyên (Tablespace, Quota), gán Profile giới hạn phiên làm việc, đến việc khóa/mở khóa tài khoản.
    - Đơn giản hóa việc phân quyền thông qua cơ chế Role (RBAC), giúp quản trị viên thao tác nhanh chóng và chính xác hơn so với việc sử dụng dòng lệnh.
2. **Về mặt bảo mật:**
    - Áp dụng thành công Virtual Private Database (VPD) để cô lập dữ liệu giữa các phòng ban. Kết quả thực nghiệm cho thấy người dùng chỉ có thể truy cập đúng phạm vi dữ liệu được cấp phép, đảm bảo tính riêng tư và an toàn thông tin.
    - Triển khai Unified Auditing và Fine-Grained Auditing (FGA) giúp ghi vết toàn bộ các hành động thay đổi cấu trúc hệ thống (DDL) và các truy cập nhạy cảm vào cột ngân sách. Hệ thống log tập trung giúp dễ dàng tra cứu và phát hiện các hành vi bất thường.
    - Tính năng Data Redaction hoạt động ổn định, tự động che giấu thông tin nhạy cảm (Email, SĐT) đối với người dùng không có thẩm quyền ngay khi truy vấn, giảm thiểu rủi ro rò rỉ dữ liệu.

5.2. Hạn chế

Bên cạnh những kết quả đạt được, đề tài vẫn còn tồn tại một số hạn chế nhất định do giới hạn về thời gian và nguồn lực:

1. Phần Frontend sử dụng Jinja2 Templates và Bootstrap ở mức cơ bản, tập trung vào tính năng demo hơn là trải nghiệm người dùng (UX). Giao diện chưa thực sự linh hoạt như các ứng dụng Single Page Application (SPA) hiện đại.
2. Nhóm mới chỉ tập trung vào các giải pháp bảo vệ dữ liệu khi truy cập (VPD, Redaction) và giám sát (Audit). Các giải pháp bảo vệ dữ liệu lưu trữ (Data at Rest) như mã hóa dữ liệu trong suốt (Transparent Data Encryption - TDE) chưa được triển khai.
3. Tính năng Oracle Database Vault (ODV) để ngăn chặn hoàn toàn quyền truy cập của tài khoản quản trị cấp cao (SYSDBA) vào dữ liệu nghiệp vụ mới chỉ được nghiên cứu ở mức lý thuyết và mô phỏng cơ bản, chưa triển khai đầy đủ các Realm phức tạp.
4. Cơ chế xác thực hiện tại dựa hoàn toàn vào Database Authentication. Trong môi trường doanh nghiệp thực tế, cần tích hợp thêm các cơ chế hiện đại như LDAP, Active Directory hoặc Multi-Factor Authentication (MFA).

5.3. Hướng phát triển

Dựa trên nền tảng hiện có, nhóm đề xuất các hướng phát triển để hoàn thiện hệ thống trong tương lai:

1. **Nâng cấp kiến trúc và giao diện:**
    - Chuyển đổi Frontend sang các Framework hiện đại như ReactJS hoặc VueJS để cải thiện trải nghiệm người dùng.
    - Phát triển hệ thống theo hướng Microservices để tăng khả năng mở rộng.
2. **Tăng cường bảo mật chuyên sâu:**
    - Triển khai Oracle Database Vault (ODV) đầy đủ để thực hiện phân tách nhiệm vụ (Separation of Duties), đảm bảo ngay cả quản trị viên hệ thống cũng không thể xem dữ liệu nhạy cảm của khách hàng.
    - Áp dụng Transparent Data Encryption (TDE) để mã hóa toàn bộ Tablespace, bảo vệ dữ liệu ngay cả khi ổ cứng máy chủ bị đánh cắp.
    - Tích hợp Oracle Label Security (OLS) để nâng cao khả năng phân loại và kiểm soát truy cập dữ liệu đa cấp độ (Ví dụ: Mật, Tối mật).

3. **Tối ưu hóa vận hành:**
    - Xây dựng các Dashboard (Biểu đồ) trực quan để thống kê log Audit và cảnh báo các hành vi xâm nhập theo thời gian thực.
    - Đóng gói quy trình triển khai (CI/CD) để tự động hóa việc cập nhật các chính sách bảo mật.

TÀI LIỆU THAM KHẢO

1. Oracle Corporation. (2024). *Oracle Database 23ai Security Guide*. Oracle Help Center.
2. Oracle Corporation. (2024). *Oracle Database PL/SQL Packages and Types Reference*.
3. Sebastian Ramirez. (2024). *FastAPI Documentation*. https://fastapi.tiangolo.com/
4. Tài liệu bài giảng môn Bảo mật Cơ sở dữ liệu - ThS. Lê Thị Minh Châu, Trường Đại Học Công nghệ Kỹ thuật TP.HCM.