**BÁO CÁO ĐỒ ÁN LÝ THUYẾT**

**AN TOÀN BẢO MẬT DỮ LIỆU TRONG HỆ THỐNG THÔNG TIN**

*LỚP 19HTTT1 - NHÓM 01*

*GIẢNG VIÊN HƯỚNG DẪN*

* *TS.*Phạm Thị Bạch Huệ
* *Ths.*Lương Vĩ Minh

Mục lục

[**A.** **Thông tin chung** 3](#_Toc100413561)

[**I.** **Thông tin nhóm** 3](#_Toc100413562)

[**II.** **Thông tin đồ án** 3](#_Toc100413563)

[**B.** **Kết quả đồ án** 3](#_Toc100413564)

[**I.** **Phân hệ 1** 3](#_Toc100413565)

[**1.** **Lý thuyết** 3](#_Toc100413566)

[**a.** **Người dùng (User)** 3](#_Toc100413567)

[**b.** **Vai trò (Role)** 4](#_Toc100413568)

[**c.** **Khung nhìn (View)** 4](#_Toc100413569)

[**d.** **Quyền người dùng (Privilege)** 6](#_Toc100413570)

[**2.** **Triển khai cài đặt** 7](#_Toc100413571)

[**a.** **Xem thông tin các đối tượng** 7](#_Toc100413572)

[**b.** **Tạo/xóa các đối tượng** 10](#_Toc100413573)

[**c.** **Quản lý quyền truy cập** 10](#_Toc100413574)

[**II.** **Phân hệ 2** 12](#_Toc100413575)

[**C.** **Tham khảo** 12](#_Toc100413576)

1. **Thông tin chung**
2. **Thông tin nhóm**

Nhóm 01

Số lượng thành viên: 3

**Thông tin thành viên**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ tên** | **Email** | **Ghi chú** |
| 1 | 19127652 | Hồ Nhật Linh | 19127652@student.hcmus.edu.vn | Nhóm trưởng |
| 2 | 19127512 | Lâm Hoàng Phúc | 19127512@student.hcmus.edu.vn |  |
| 3 | 19127507 | Nguyễn Quang Phú | 19127507@student.hcmus.edu.vn |  |

1. **Thông tin đồ án**

Hệ quản trị CSDL: Oracle database 18c.

Nền tảng ứng dụng: Windows, MacOS, Linux.

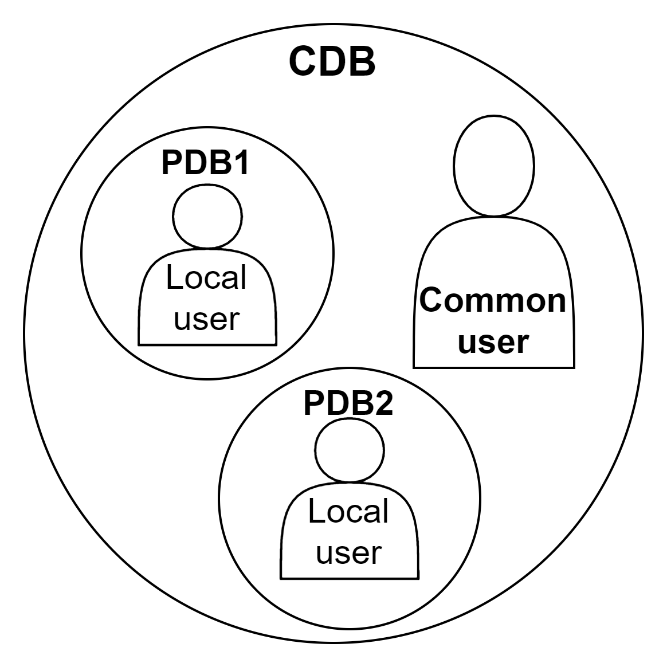
Ngôn ngữ lập trình: Java.

Framework: Java swing.  
API kết nối database: JDBC (Java Database Connectivity) sử dụng **công nghệ SSL** để bảo mật thông tin trên đường truyền.

1. **Kết quả đồ án**
2. **Phân hệ 1**
3. **Lý thuyết**
4. **Người dùng (User)**

User sở hữu 1 tài khoản trên HQTCSDL, user có thể dùng tài khoản này để đăng nhập và thực hiện các công việc (thêm, sửa, xóa dữ liệu, cấu trúc dữ liệu,…) được cho phép thực thi.

Oracle lưu trữ cơ sở dữ liệu trong 1 **CDB** (Container database hay Root) lớn để quản lý tất cả dữ liệu, trong CDB có chứa các **PDB** (Pluggable database), PDB chứa các lược đồ, dữ liệu của riêng nó.

Trong Oracle, có 2 loại user là common user và local user

(Mô tả common user và local user trong Oracle)

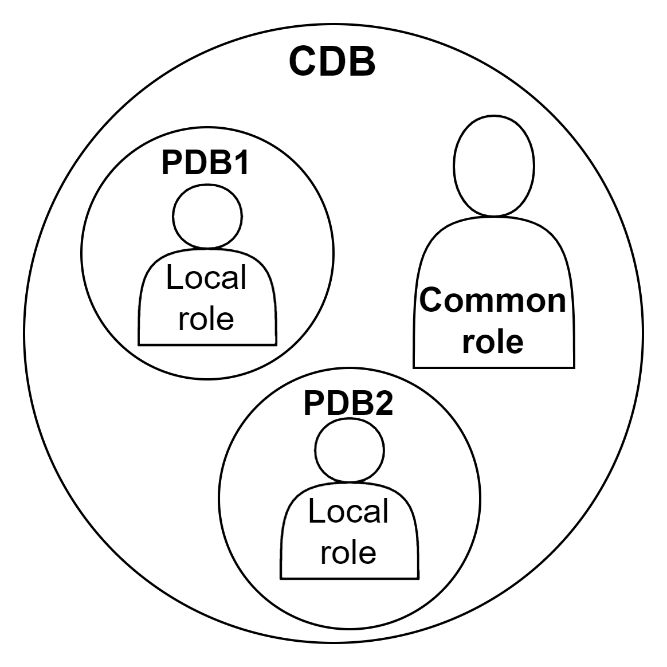
* **Common user**: Có thể sử dụng tài nguyên trên toàn bộ CDB (toàn bộ CSDL), tên người dùng phải bắt đầu bằng kí tự “C##” và không được trùng nhau.
* **Local user**: Chỉ có thể sử dụng tài nguyên ở 1 PDB nó thuộc về, tên user có thể khác nhau nếu không cùng PDB.

Trong Oracle, **1 user tương ứng 1 schema**, chứa tất cả những đối tượng (table, view, procedure) của riêng schema đó, những đối tượng khác schema có thể khác tên.

Ví dụ ta có 1 user C##NHOM1 thì tuong ứng ta sẽ có 1 schema C##NHOM1, ta có thể tạo các đối tượng trên shema (sử dụng toán tử ‘.’ để truy cập vào các đối tượng), chẳng hạn C##NHOM1.BANG\_1, C##NHOM1.KHUNG\_NHIN\_1.

1. **Vai trò (Role)**

Role là 1 tập hợp những quyền liên quan với nhau. Role có thể được sử dụng để gán cho 1 user hay 1 role khác để user hay role đó kế thừa những quyền của role gán.

Tương tự như user, role cũng có 2 loại là **Common role** (có thể truy cập trên CDB, tên phải bắt đầu là “C##”) và **Local role** (chỉ được truy cập PDB của mình). 

(Mô tả common role và local role trong Oracle)

1. **Khung nhìn (View)**

**Khung nhìn (View)** là một bảng ảo, nó không chứa dữ liệu mà chỉ là đối tượng trung gian để tương tác với các bảng thật. View được tạo từ câu truy vấn dữ liệu

Ví dụ:Tạo view để xem thông tin giáo viên thuộc bộ môn HTTT

CREATE VIEW GIAO\_VIEN\_HTTT

AS

SELECT \*

FROM GIAO\_VIEN

**WHERE MA\_BM = ‘HTTT’;**

Thực chất, View trong Oracle chỉ là những câu truy vấn đã được xác thực được lưu lại trong data dictionary.Ảnh có chứa văn bản

Mô tả được tạo tự động

(Cách Oracle lưu view)

Bên cạnh việc tăng hiệu năng truy vấn. Thì mục đích chính của việc dùng View là để thực hiện các chính sách bảo mật trong CSDL

* **Bảo vệ dữ liệu mức dòng:** ta dùng điều kiện WHERE để loại những dòng dữ liệu cần bảo vệ khỏi View. Chẳng hạn, ta tạo View để truy vấn thông tin nhân viên kế toán (KT), những nhân viên khác sẽ không được xem

CREATE VIEW NHAN\_VIEN\_KE\_TOAN

AS

SELECT \*

FROM NHAN\_VIEN

**WHERE LOAI\_NHAN\_VIEN = ‘KE TOAN’;**

* **Bảo vệ dữ liệu mức cột**: ta có thể chọn ra những cột dữ liệu cần thiết cho View và ẩn những cột còn lại đi. Chẳng hạn, ta tạo View để truy vấn thông mã nhân viên và họ tên nhân viên, các trường còn lại (ngày sinh, số CMND, lương,…) sẽ bị ẩn đi khỏi View.

CREATE VIEW NHAN\_VIEN\_THONG\_TIN

AS

**SELECT MA\_NHAN\_VIEN, HO\_TEN**

FROM NHAN\_VIEN;

1. **Quyền người dùng (Privilege)**

**Quyền người dùng (privilege)** là sự cho phép người dùng thực hiện một câu lệnh SQL cụ thể hoặc thực hiện truy cập đến các objects của các người dùng khác (hay schema khác).

Có 2 loại quyền trong Oracle:

* **Quyền hệ thống (System privilege):** Cho phép người dùng thực hiện hành động trên đối tượng (table, view, index,…) của **bất kì schema nào**. Loại quyền này chỉ nên cấp cho những user có độ tin cậy và quyền hành cao. Ảnh có chứa văn bản

  Mô tả được tạo tự động

*(Ví dụ 1 số quyền hệ thống. Nguồn:* [*Privileges (oracle.com)*](https://docs.oracle.com/database/121/TTSQL/privileges.htm#TTSQL339)*)*

* **Quyền đối tượng (Object privilege):** Cho phép 1 user có thể thực hiện hành động hoặc truy cập vào các **đối tượng của 1 user khác** (schema khác). **User sẽ có toàn bộ quyền đối tượng trên schema của mình**Ảnh có chứa văn bản

  Mô tả được tạo tự động *(Ví dụ 1 số quyền đối tượng. Nguồn:* [*Privileges (oracle.com)*](https://docs.oracle.com/database/121/TTSQL/privileges.htm#TTSQL339)*)*

**Cú pháp câu lệnh cấp quyền:**

* **GRANT <PRIVILLEGE> ON <OBJECT> TO <USER>** (với những câu lệnh có mức cột thì privilege cần có thêm thông tin cột)
* Ví dụ:
  + **GRANT SELECT ON ANOTHER.TABLE\_TEST TO USER\_TEST**: cấp quyền SELECT trên bảng TABLE\_TEST của schema ANOTHER cho user USER\_TEST.
  + **GRANT UPDATE(COLUMN\_1, COLUMN\_2) ON ANOTHER.TABLE\_TEST TO USER\_TEST**: cấp quyền UPDATE trên các cột COLUMN\_1, COLUMN\_2 của bảng TABLE\_TEST thuộc schema ANOTHER cho user USER\_TEST.

**Cú pháp câu lệnh thu hồi quyền đã cấp:**

* **REVOKE <PRIVILLEGE> ON <OBJECT> FROM <USER>** (khi thu hồi chỉ thu hồi toàn bộ quyền, không thu hồi trên chi tiết từng cột)
* Ví dụ:
  + **REVOKE UPDATE(COLUMN\_1) ON ANOTHER.TABLE\_TEST FROM USER\_TEST**: Đây là **câu lệnh sai** do tiến hành thu hồi trên cột.
  + **REVOKE UPDATE ON ANOTHER.TABLE\_TEST FROM USER\_TEST**: Đây là câu lệnh đúng, tiến hành thu hồi quyền UPDATE trên bảng TABLE\_TEST thuộc shema ANOTHER của user USER\_TEST.

1. **Triển khai cài đặt**
2. **Xem thông tin các đối tượng**

**Xem các users trong hệ thống:** ta sử dụng view **DBA\_USERS** để lấy thông tin các user Ảnh có chứa bàn

Mô tả được tạo tự động

*(Tất cả user trong hệ thống)*

**Xem các roles trong hệ thống:** ta sử dụng view **DBA\_ROLES** để lấy tất cả roleẢnh có chứa văn bản

Mô tả được tạo tự động

*(Tất cả role trong hệ thống)*

**Xem các roles đã được cấp cho 1 user hoặc các users trong 1 role:** ta sử dụng view **DBA\_ROLE\_PRIVS** với cột **GRANTEE** là user hoặc role được gán, và **GRANTED\_ROLE** là role gán.

Ảnh có chứa văn bản

Mô tả được tạo tự động

*(Liệt kê những role/user được cấp role)*

1. **Tạo/xóa các đối tượng**

Để tạo các đối tượng ta cần lấy thông tin về tin về schema và tên của đối tượng đó.

Đầu tiên cần kiểm tra tên đối tượng đã tồn tại chưa

* **Đối với user, role** thì sử dụng view DBA\_ROLES và DBA\_USERS.
* **Đối với table** thì truy vấn view DBA\_TABLES (lấy ra thuộc tính OWNER và TABLE\_NAME) để xác nhận điều này.

Ta tiến hành tạo các đối tượng

* **Đối với role**, chỉ cần tên và schema của role đó: CREATE ROLE <schema>.<role>.
* **Đối với user**, ta cần thêm thông tin mặt khẩu của user: CREATE USER <user> IDENTIFIED BY <password>.
* **Đối với table**, cần cung cấp thông tin về các thuộc tính (kiểu dữ liệu, chiều dài dữ liệu, khóa chính, khóa ngoại (thuộc tính tham chiếu), cho phép null): Ta tạo 1 câu lệnh SQL để từ những thuộc tính đó theo mẫu (thế chuỗi):

CREATE TABLE <SCHEMA\_NAME>.<TABLE\_NAME> (

<COLUMN\_NAME> <DATA\_TYPE>(<DATA\_LENGTH>, <DATA\_PRECISION>) <NOT NULL> <UNIQUE>,

--Nếu thuộc tính là khóa ngoại

FOREIGN KEY (<COLUMN\_NAME>) REFERENCES <REF\_TABLE>(<REF\_COLUMN>)

…

PRIMARY KEY (<COLUMN\_NAME>,…));

1. **Quản lý quyền truy cập**

Ta đã điểm qua cú pháp cấp quyền và thu hồi quyền ở phần 1, ta chỉ cần lập trình để tạo những câu lệnh SQL dựa vào yêu cầu của DBA.

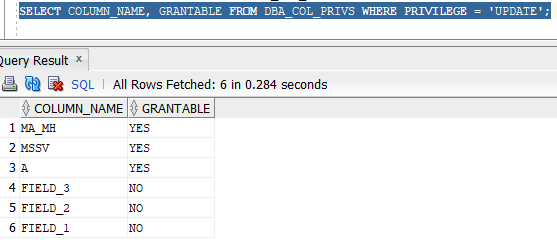
Để xem quyền của user, role trên các bảng:

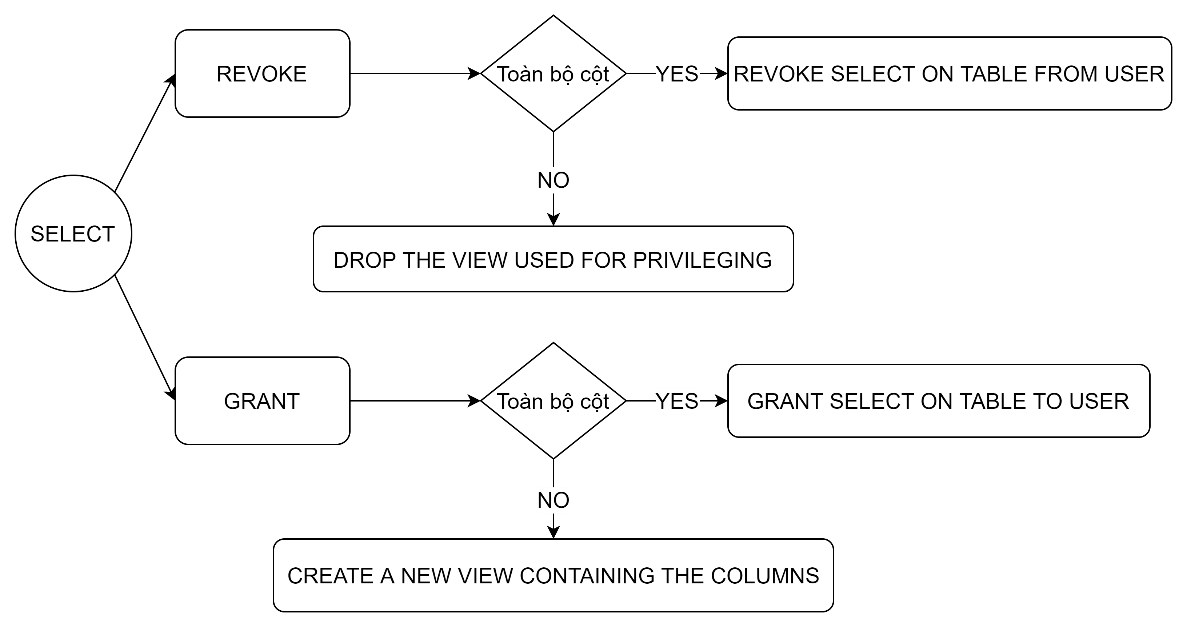
* Đối với quyền không chi tiết trên cột **(INSERT, DELETE)**, ta dùng view **DBA\_TAB\_PRIVS** để lấy thông tin về quyền, với PRIVILEGE là hành động, GRANTABLE là chỉ định WITH GRANT OPTION:

Ảnh có chứa bàn

Mô tả được tạo tự động

*(Liệt kê quyền trên bảng của user)*

* Đối với quyền chi tiết trên cột **(UPDATE),** ta dùng view **DBA\_COL\_PRIVS** để lấy thông tin về quyền trên các cột, với COLUMN\_NAME là tên cột, PRIVILEGE là hành động, GRANTABLE là chỉ định WITH GRANT OPTION: 

Đặc biệt chú ý, đối với quyền SELECT, do **Oracle không hỗ trợ cấp quyền SELECT chi tiết đến mức cột,** nên ta sẽ sử dụng View thay thế. Nếu DBA cấp quyền cho toàn bộ cột thì ta cấp bằng câu lệnh GRANT SELECT như bình thường, còn nếu cấp chi tiết đến cột thì ta sẽ tạo View mới có tên là <schema>.<user>\_SELECT\_ON\_<table> với những cột đã chọn. Khi thu hồi quyền ta sẽ tiến hành thu hồi quyền SELECT trên bảng đi (nếu user có quyền trên toàn bộ cột) hoặc xóa hẳn View dùng cho việc cấp quyền đi (nếu user có quyền trên 1 số cột). 

(Mô tả việc cấp/thu hồi quyền SELECT)

1. **Phân hệ 2**
2. **Tham khảo**