

Bài 3: QUYỀN VÀ NHÓM QUYỀN

1. Quyền và Role

1.1. Quyền (privilege)

- Một quyền là một sự cho phép thực hiện một câu lệnh SQL nào đó hoặc được phép truy xuất đến một đối tượng nào đó (ví dụ: quyền tạo bảng CREATE TABLE, quyền connect đến cơ sở dữ liệu CREATE SESSION, quyền SELECT trên một bảng cụ thể nào đó,...).
- Chỉ cấp cho user chính xác những quyền mà user cần đến. Việc cấp dư thừa những quyền không cần thiết có thể gây nguy hại cho việc bảo mật hệ thống.
- Có 2 loại quyền:
 - o Quyền hệ thống (System Privilege):
 - Là quyền thực hiện một tác vụ CSDL cụ thể hoặc quyền thực hiện một loại hành động trên tất cả những đối tượng schema của hệ thống. Vd: quyền ALTER SYSTEM, quyền CREATE TABLE, quyền DELETE ANY TABLE (xóa các hàng của bất kỳ bảng nào trong CSDL),...
 - User có thể cấp 1 quyền hệ thống nếu có một trong các điều kiện sau:
 - User đã được cấp quyền hệ thống đó với tùy chọn WITH ADMIN OPTION.
 - User có quyền GRANT ANY PRIVILEGE
 - o Quyền đối tượng (Schema Object Privilege hoặc Object Privilege)
 - Là quyền thực hiện một hành động cụ thể trên một đối tượng schema cụ thể. Vd: quyền xóa các hàng dữ liệu khỏi bảng Department.
 - Có nhiều quyền đối tượng khác nhau dành cho các loại đối tượng schema khác nhau.
 - Dùng để quản lý việc truy xuất đến các đối tượng schema cụ thể nào đó.
 - User có thể cấp 1 quyền đối tượng nếu có một trong các điều kiện sau:
 - User có tất cả mọi quyền đối tượng trên tất cả các đối tượng thuộc schema của mình. Vì vậy user có quyền cấp bất kỳ quyền đối tượng trên bất kỳ đối tượng nào thuộc sở hữu của mình cho bất cứ user nào khác.
 - User có quyền GRANT ANY OBJECT PRIVILEGE.
 - User được cấp quyền đối tượng đó với tùy chọn WITH GRANT OPTION

1.2. Role

- Role là một tập hợp bao gồm các quyền và các nhóm quyền khác.
- Role được gán cho các user hoặc các role khác.
- Role giúp cho việc quản trị người dùng dễ dàng và tiết kiệm công sức hơn.

- Có một số role có sẵn do hệ thống định nghĩa (vd: DBA, RESOURCE, CONNECT,...) nhưng đa phần các role là do người quản trị CSDL tạo ra.
- Role không phải là một đối tượng schema (schema object) nên không được lưu trữ trong schema của user tạo ra nó. Do vậy, user tạo ra một role có thể bị xóa mà không ảnh hưởng đến role đó.
- User có thể cấp 1 role nếu có một trong các điều kiện sau:
 - o User đã tạo ra role đó.
 - o User đã được cấp role đó với tùy chọn WITH ADMIN OPTION.
 - o Có quyền GRANT ANY ROLE

1.3. Thực hành quyền và role

1.3.1. Tạo ROLE

- Tạo một role mới với câu lệnh:

CREATE ROLE myrole;

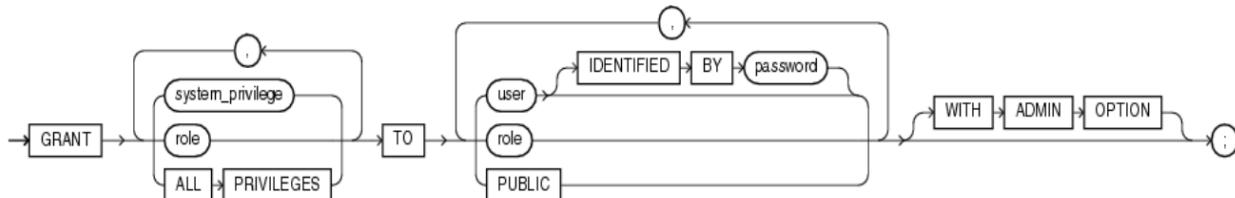
Role created.

- Lưu ý, để tạo được role, phải có quyền hệ thống **CREATE ROLE**

1.3.2. Lệnh GRANT

- Gán quyền hệ thống/role:

- Ta dùng cú pháp dưới đây để gán các quyền hệ thống/role cho các user/role khác:



Ví dụ:

GRANT DELETE ANY TABLE TO salapati;

Grant succeeded.

GRANT CREATE USER TO myrole;

Grant succeeded.

GRANT myrole TO salapati;

Grant succeeded.

GRANT myrole TO lavender;

Grant succeeded.

- Xem lệnh sau:

GRANT CREATE SESSION TO lavender IDENTIFIED BY purple;

Grant succeeded.

Với câu lệnh vừa rồi, nếu user lavender đã tồn tại, password của lavender sẽ được thay đổi thành purple. Ngược lại, hệ thống sẽ tạo ra 1 người dùng mới có username là lavender và password là purple. Sinh viên tự tìm hiểu xem để câu lệnh trên có thể thực hiện được, user cần phải có quyền gì?

- Dùng từ khóa PUBLIC nếu muốn cấp quyền/role cho mọi user:

GRANT CREATE SESSION TO PUBLIC;

- Dùng từ khóa ALL PRIVILEGES nếu muốn cấp tất cả các quyền hệ thống (trừ quyền SELECT ANY DICTIONARY):

GRANT ALL PRIVILEGES TO salapati;

Để thực hiện câu lệnh trên thành công thì user cần phải có quyền gì?

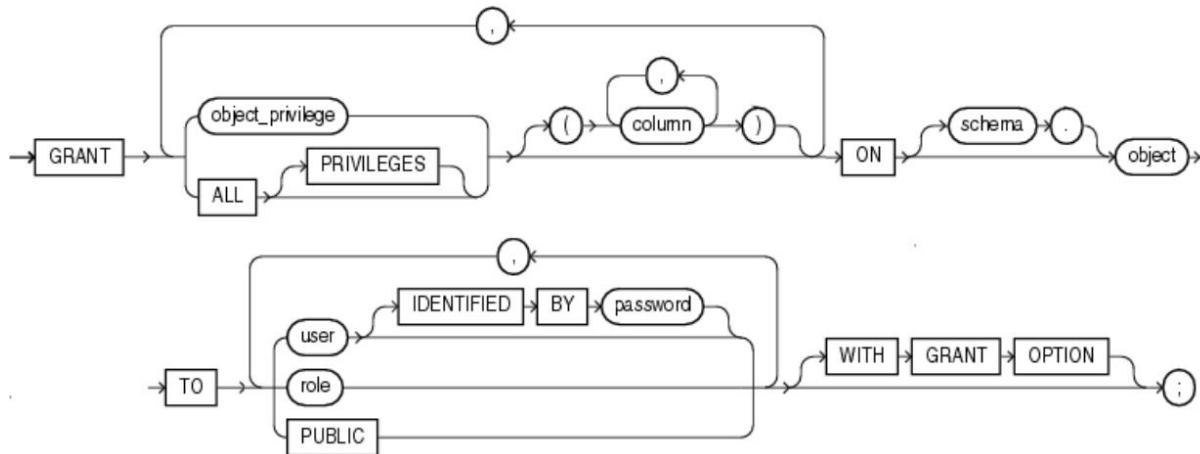
- Tùy chọn WITH ADMIN OPTION sẽ cho phép người được cấp role/quyền:
 - o Cấp lại role/quyền đó cho một user hoặc role khác (có hoặc không có tùy chọn WITH ADMIN OPTION).
 - o Thu hồi lại role/quyền đó từ một user hoặc role bất kỳ.
 - o Thay đổi role đó bằng lệnh ALTER ROLE.
 - o Xóa role đó.

Ví dụ:

GRANT CREATE SESSION TO salapati WITH ADMIN OPTION;

Grant succeeded.

b) Gán quyền đối tượng



Ví dụ:

GRANT DELETE ON mytable TO salapati;

GRANT SELECT ON mytable TO public;

GRANT SELECT,INSERT,UPDATE,DELETE ON mytable TO lavender;

GRANT SELECT ON salapati.xyz TO myrole;

- Dùng từ khóa ALL [PRIVILEGES] khi muốn cấp tất cả các quyền đối tượng mà user có trên 1 đối tượng nào đó (với điều kiện user phải có quyền cấp những quyền đó):

GRANT ALL ON salapati.xyz TO paris;

GRANT ALL PRIVILEGES ON salapati.xyz TO paris;

- Nếu chỉ muốn cấp quyền trên vài cột nào đó của table hoặc view, ta chỉ ra cụ thể tên các cột đó:

GRANT UPDATE (name) ON salapati.xyz TO myrole;

Grant succeeded.

Lưu ý là ta chỉ có thể chỉ ra các cột cụ thể khi cấp quyền INSERT và UPDATE.

- Dùng tùy chọn WITH GRANT OPTION khi muốn user được cấp quyền có thể cấp quyền đó cho user/role khác. Tuy nhiên chỉ được dùng tùy chọn này khi cấp quyền cho một user hay PUBLIC:

GRANT ALL ON salapati.xyz TO paris WITH GRANT OPTION;

c) Xem thông tin

- Các quyền hệ thống đã được gán cho user hiện tại:

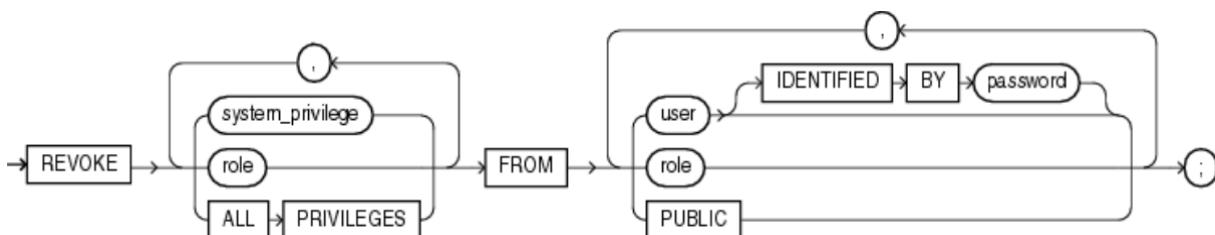
*SELECT * FROM user_sys_privs;*

- Các quyền đối tượng đã được gán cho user hiện tại:

*SELECT * FROM user_tab_privs_recd;*

1.3.3. Lệnh REVOKE

d) Thu hồi quyền hệ thống/role:



- Để thu hồi quyền hệ thống, user phải được cấp quyền đó với WITH ADMIN OPTION hoặc có quyền GRANT ANY PRIVILEGES.
- Để thu hồi role, user phải được cấp role đó với WITH ADMIN OPTION hoặc có quyền GRANT ANY ROLE.
- Không thể dùng lệnh REVOKE để thu hồi những role/quyền hệ thống được cấp thông qua những role khác.

REVOKE DELETE ANY TABLE FROM salapati;

Revoke succeeded.

REVOKE myrole FROM lavender;

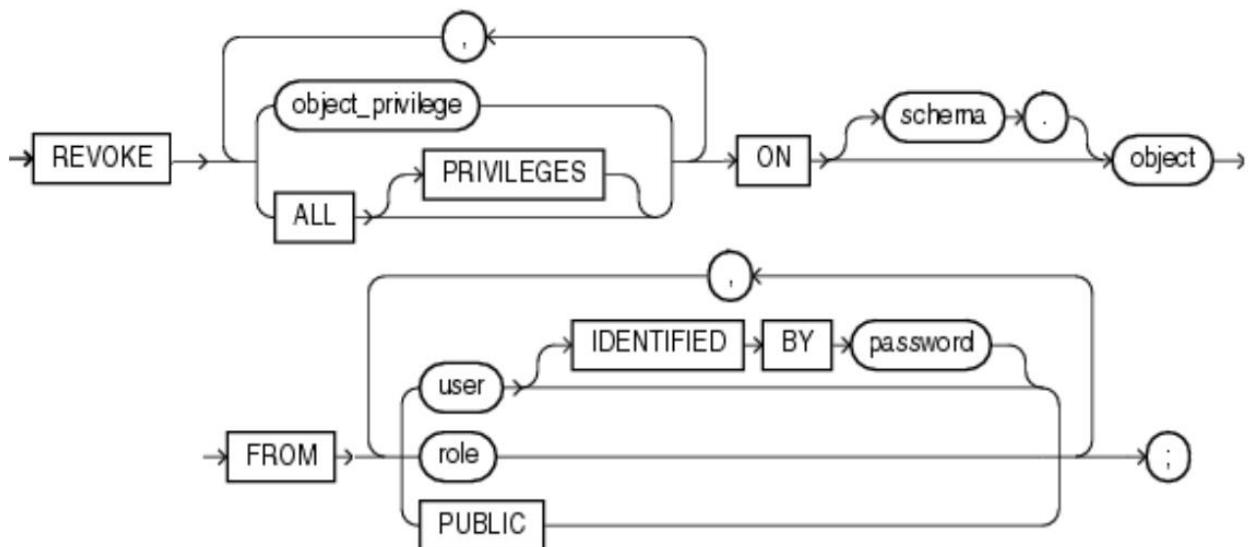
Revoke succeeded.

- Dùng từ khóa PUBLIC để thu hồi một quyền hệ thống/role đã được cấp cho tất cả các user thông qua PUBLIC. Tuy nhiên không thể dùng PUBLIC để thu hồi những quyền được gán trực tiếp hoặc thông qua 1 role khác.

REVOKE CREATE SESSION FROM PUBLIC;

- Các từ khóa khác có ý nghĩa như trong lệnh GRANT.

- e) Thu hồi quyền đối tượng



- Để thu hồi một quyền đối tượng từ một user/role, người dùng là người đã cấp quyền đó cho user/role đó hoặc người dùng có quyền GRANT ANY OBJECT PRIVILEGE. Tuy nhiên, đối với trường hợp người dùng có quyền GRANT ANY OBJECT PRIVILEGE, người dùng chỉ có thể thu hồi những quyền đối tượng được cấp cho user/role, nếu quyền đối tượng đó đã được cấp cho user/role bởi chủ nhân của chính đối tượng hoặc bởi những user có quyền GRANT ANY OBJECT PRIVILEGE.
- Không thể dùng lệnh REVOKE để thu hồi quyền/role đã được cấp thông qua các role khác.

REVOKE SELECT ON mytable FROM salapati;

- Dùng từ khóa ALL [PRIVILEGES] để thu hồi tất cả những quyền đối tượng mà người dùng đã cấp cho user/role đó.

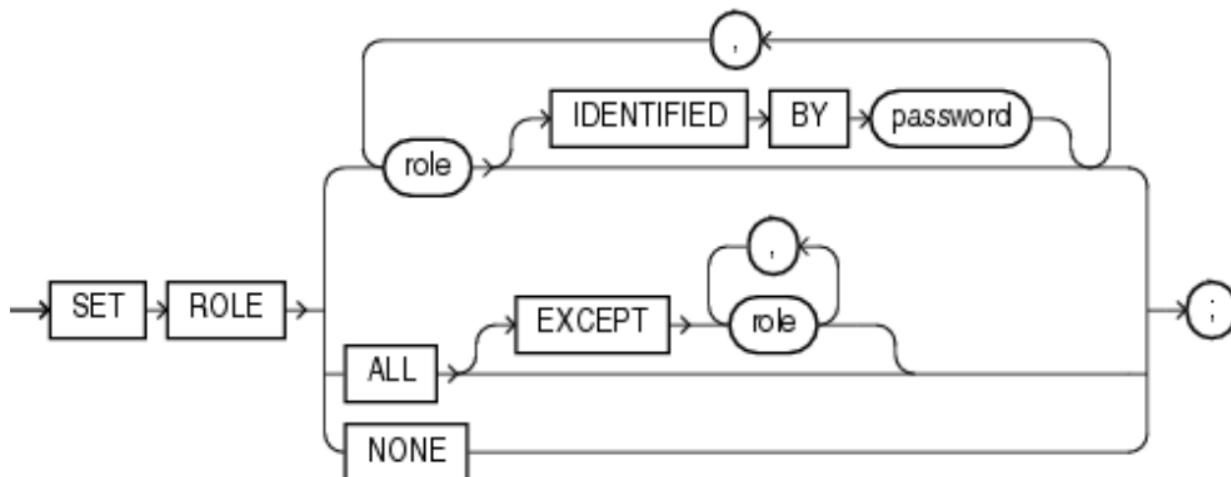
REVOKE ALL ON salapati.xyz FROM paris;

- Dùng từ khóa PUBLIC để thu hồi một quyền đối tượng khỏi những user đã được cấp quyền đó thông qua việc gán cho PUBLIC. Không thể dùng PUBLIC để thu hồi những quyền được gán trực tiếp hoặc thông qua một role khác.

REVOKE INSERT ON salapati.xyz FROM paris;

- Lưu ý:
 - o Nếu user bị thu hồi quyền đối tượng mà quyền này đã được cấp cho user/role khác, hệ thống cũng thu hồi quyền đối tượng đó của những user/role kia.
 - o Nếu schema của user bị thu hồi quyền đối tượng có chứa các procedure, function, package có sử dụng lệnh SQL liên quan đến quyền bị thu hồi thì những procedure, function, package đó không còn có thể thực thi được.

1.3.4. Enable và disable một ROLE



- Một user có thể có nhiều role.
 - o Không phải session nào cũng cần sử dụng tất cả các role đó. Oracle cho phép bản thân user enable/disable các role mà mình được cấp để quản lý sự cần thiết của các role trong session hiện tại.
 - o Mặc định khi bắt đầu session mới, user sẽ được hệ thống enable tất cả các role mặc định (default role). Sau đó user có thể dùng lệnh SET ROLE để enable/disable các role theo ý mình theo cú pháp sau:

Ví dụ:

SET ROLE myrole, sysrole;

Lệnh trên sẽ enable 2 role được chỉ định và disable tất cả các role còn lại của user.

- o Để enable tất cả các role dùng lệnh: SET ROLE ALL;
 - o Để disable tất cả các role dùng lệnh: SET ROLE NONE;
 - o Để enable tất cả các role ngoại trừ role lavender ta dùng lệnh: SET ROLE ALL EXCEPT sysrole;
- Bảo vệ một role

- Để bảo vệ một role với mục đích không cho phép các user tùy ý enable/disable một role, người tạo role có thể thiết lập password cho role đó ngay khi tạo role:

CREATE ROLE newrole IDENTIFIED BY protected;

- Thay đổi việc thiết lập password cho role:

ALTER ROLE newrole IDENTIFIED BY changed;

ALTER ROLE newrole NOT IDENTIFIED;

- Để biết hiện tại role nào đang được enable ta truy xuất view SESSION_ROLES:

*SELECT * FROM SESSION_ROLES;*

- Để quy định những role nào là role mặc định, dùng lệnh ALTER USER:

ALTER USER salapati DEFAULT ROLE myrole, sysrole;

ALTER USER salapati DEFAULT ROLE ALL;

ALTER USER salapati DEFAULT ROLE ALL EXCEPT myrole;

ALTER USER salapati DEFAULT ROLE NONE;

1.3.5. Xóa ROLE

DROP ROLE myrole;

2. Từ điển dữ liệu (Data Dictionary)

2.1. Tổng quan

- Mọi CSDL Oracle đều có một Từ điển dữ liệu. Từ điển dữ liệu được tạo ra khi CSDL được tạo
- Từ điển dữ liệu trong Oracle là một tập các bảng và view được sử dụng như một tham khảo dạng chỉ đọc (read-only) về bản thân CSDL đó.
- Từ điển dữ liệu nằm trên tablespace SYSTEM, thuộc schema của user SYS, bao gồm 2 loại:
 - Các bảng cơ bản (Base table): Là các bảng lưu trữ thông tin của từ điển dữ liệu. Dữ liệu được lưu trong các bảng này dưới dạng mã hóa.
 - Các view dành cho người dùng truy xuất (User-accessible View): Tổng hợp và hiển thị thông tin được lưu trong các bảng cơ bản ở dạng người bình thường có thể đọc hiểu. Tùy vào quyền của mỗi user mà user đó có thể truy xuất view nào và truy xuất những dữ liệu nào của view đó.
- Một Từ điển dữ liệu sẽ lưu trữ tất cả các thông tin về cấu trúc luận lý và cấu trúc vật lý của CSDL:
 - Định nghĩa của tất cả các đối tượng schema trong CSDL.
 - Các quy định, giới hạn về sử dụng tài nguyên của các user,v.v
 - Danh sách các user. Các quyền, role được cấp cho các user.
 - Các ràng buộc toàn vẹn của dữ liệu
 - Thông tin audit

- Các thông tin CSDL tổng quát khác.
- Oracle tự động cập nhật từ điển dữ liệu để phản ánh chính xác trạng thái thực tế của CSDL.

2.2. Các tiếp đầu ngữ trong tên view

- Trong nhiều trường hợp, một tập gồm 3 view chứa những thông tin tương tự nhau và tên của chúng chỉ khác nhau ở các tiếp đầu ngữ: user, all, dba.
 - USER: hiển thị những gì thuộc schema của user đó.
 - ALL: hiển thị những gì mà user đó có thể truy xuất.
 - DBA: hiển thị tất cả thông tin thuộc schema của mọi user (view dành cho những người quản trị).
- Các column trong các view thuộc 1 bộ ba view hầu như là giống nhau, ngoại trừ một số ngoại lệ.

2.3. Các view thường sử dụng

- DBA_USERS: cung cấp thông tin của các user trong CSDL.
- DBA_TS_QUOTAS: cung cấp thông tin tablespace quota của các user.
- DBA_PROFILES: cung cấp thông tin về các profile.
- DBA_ROLES: cung cấp thông tin về các role.
- DBA_SYS_PRIVS: hiển thị thông tin về việc cấp quyền hệ thống cho các user.
- DBA_TAB_PRIVS: hiển thị thông tin về việc cấp quyền đối tượng cho các user.
- DBA_COL_PRIVS: hiển thị thông tin về việc cấp quyền đối tượng mức cột cho các user.
- DBA_ROLE_PRIVS: hiển thị tất cả các user và role của họ.
- ROLE_ROLE_PRIVS: hiển thị thông tin về việc cấp role cho các role.
- ROLE_SYS_PRIVS: hiển thị thông tin về việc cấp quyền hệ thống cho các role.
- ROLE_TAB_PRIVS: hiển thị thông tin về việc cấp quyền đối tượng cho các role.
- SESSION_ROLES: hiển thị các role hiện tại đang được enable cho user.
- SESSION_PRIVS: hiển thị các quyền hiện tại mà user có thể sử dụng.

3. Bài tập

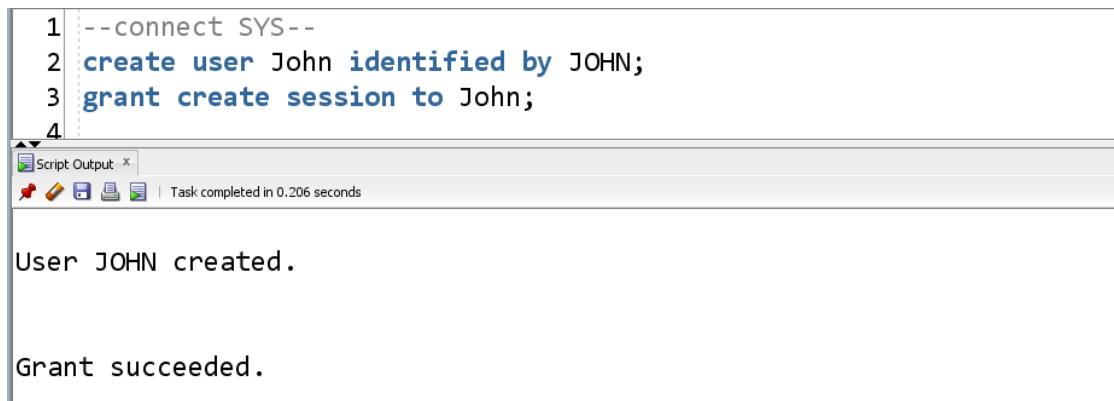
3.1. Bài tập 1

3.1.1. Yêu cầu

- Tạo các users John, Joe, Fred, Lynn, Amy, and Beth
- Password lần lượt là tên username viết hoa.
- Đảm bảo các user này có thể tạo bất kỳ bảng nào trong tablespace với quota 10M.

3.1.2. Hướng dẫn

- Password lần lượt là tên username viết hoa (hình 1).

```
1 --connect SYS--  
2 create user John identified by JOHN;  
3 grant create session to John;  
4  


The screenshot shows a SQL script window with the following code:  
1 --connect SYS--  
2 create user John identified by JOHN;  
3 grant create session to John;  
4  
Below the code, the output window displays:  
User JOHN created.  
  
Grant succeeded.


```

Hình 1

- Đảm bảo các user này có thể tạo bất kỳ bảng nào trong tablespace với quota 10M.

```
5 --connect SYS--  
6 alter user John quota 10M on USERS;  


The screenshot shows a SQL script window with the following code:  
5 --connect SYS--  
6 alter user John quota 10M on USERS;  
  
Below the code, the output window displays:  
User JOHN altered.


```

Hình 2

- Cài đặt package dùng để check, tạo và sửa đổi User.

```
8 --connect SYS--  
9 create or replace package pkg_CrUser  
10 as  
11     procedure pro_create_user(username in varchar2, pass in varchar2);  
12     procedure pro_alter_user(username in varchar2, pass in varchar2);  
13     function fun_check_account(user in varchar2) return int;  
14  
15     procedure Pro_CrUser(username in varchar2, pass in varchar2);  
16  
17 end pkg_CrUser;
```

Hình 3

- Thủ tục trong package create user dùng để tạo user.

```

create or replace package body pkg_CrUser
as
    --Procedure create user--
    procedure pro_create_user(username in varchar2, pass in varchar2)
    is
    begin
        execute immediate 'create user ' || username || ' identified by ' || pass
        || ' quota 10M on USERS';
        execute immediate 'grant create session to ' || username;
    end pro_create_user;
}

```

Hình 4

- Thủ tục trong package alter user dùng để đổi mật khẩu user

```

--Procedure alter user--
procedure pro_alter_user(username in varchar2, pass in varchar2)
is
begin
    execute immediate 'alter user ' || username || ' identified by ' || pass;
    commit work;
end pro_alter_user;
}

```

Hình 5

- Hàm check user dùng để kiểm tra user đã tồn tại hay chưa

```

--Function account status--
function fun_check_account(user in varchar2)
return int
is
    t varchar2(50);
    kq int;
begin
    select account_status into t from dba_users where username = user;
    if t is null then
        kq := 0;
    else
        kq := 1;
    end if;
    return kq;
exception when others then kq := 0;
    return kq;
end fun_check_account;
}

```

Hình 6

- Thủ tục dùng để gọi chạy tất cả các hàm và thủ tục trên dựa vào kết quả trả về của hàm check user

```
--Procedure CrUser--
procedure Pro_CrUser(username in varchar2, pass in varchar2)
is
    ckUser int := fun_check_account(username);
begin
    if ckUser = 0 then
        pro_create_user(username, pass);
    else
        pro_alter_user(username, pass);
    end if;
    commit work;
end Pro_CrUser;
end pkg_CrUser;
```

Hình 7

- Gán quyền gọi chạy package cho User: sysuser

```
68 --Gan quyen thuc thi package cho SYSUSER--
69 grant execute on pkg_cruser to sysuser;
```

Grant succeeded.

Hình 8

- Chạy thủ tục trên sql developer

```
70 --chay thu tuc tren sql developer--
71 --connect SYSUSER--
72 begin
73     sys.pkg_CrUser.Pro_CrUser('JOHN', '123');
74 end;
```

PL/SQL procedure successfully completed.

Hình 9

- Đăng nhập thử user vừa tạo bằng thủ tục trên sql plus

```
Microsoft Windows [Version 10.0.19045.3393]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nguye>sqlplus john/123

SQL*Plus: Release 12.1.0.2.0 Production on Sun Sep 10 20:27:26 2023

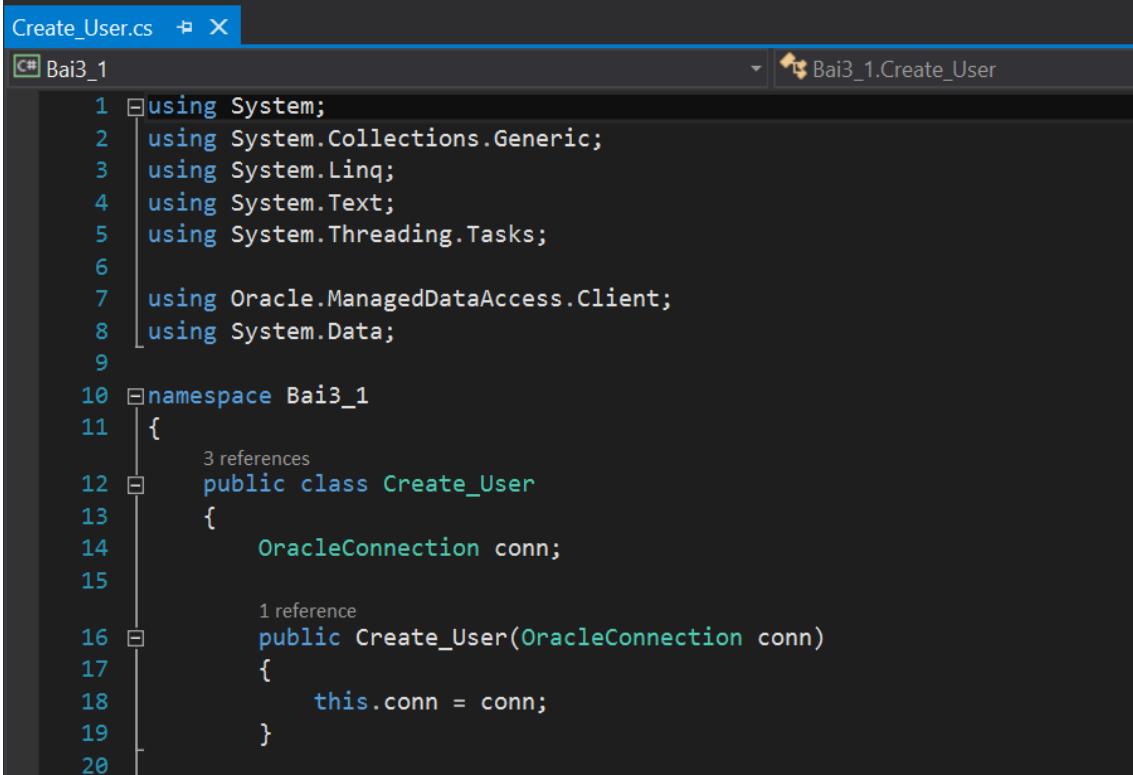
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> -
```

Hình 10

- Tạo Class Create_User



```
Create_User.cs  => X
C# Bai3_1
Bai3_1.Create_User

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  using Oracle.ManagedDataAccess.Client;
8  using System.Data;
9
10 using namespace Bai3_1
11 {
12     public class Create_User
13     {
14         OracleConnection conn;
15
16         public Create_User(OracleConnection conn)
17         {
18             this.conn = conn;
19         }
20     }
}
```

Hình 11

- Viết hàm gọi chạy hàm kiểm tra tài khoản oracle (hình 12).

```
1 reference
public int Pro_CheckUser(string UserName)
{
    try
    {
        string Function = "sys.pkg_cruser.fun_check_account";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Function;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@kq";
        resultParam.OracleDbType = OracleDbType.Int16;
        resultParam.Direction = ParameterDirection.ReturnValue;
        cmd.Parameters.Add(resultParam);

        OracleParameter User = new OracleParameter();
        User.ParameterName = "@username";
        User.OracleDbType = OracleDbType.Varchar2;
        User.Value = UserName.ToUpper();
        User.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(User);

        cmd.ExecuteNonQuery();

        if (resultParam.Value != DBNull.Value)
        {
            return int.Parse(resultParam.Value.ToString());
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Lỗi gọi chạy hàm ở Pakage!!!");
    }
    return -1;
}
```

Hình 12

- Viết hàm gọi chạy thủ tục tạo tài khoản (hình 13).

```
public bool Pro_CreateUser(string UserName, string Password)
{
    try
    {
        string Function = "sys.pkg_cruser.Pro_CrUser";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Function;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter User = new OracleParameter();
        User.ParameterName = "@username";
        User.OracleDbType = OracleDbType.Varchar2;
        User.Value = UserName.ToUpper();
        User.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(User);

        OracleParameter Pass = new OracleParameter();
        Pass.ParameterName = "@pass";
        Pass.OracleDbType = OracleDbType.Varchar2;
        Pass.Value = Password;
        Pass.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(Pass);

        cmd.ExecuteNonQuery();

        return true;
    }
    catch (Exception ex)
    {
        return false;
    }
}
```

Hình 13

- Tạo form CreateUser



Hình 14

- Using thư viện và khai báo class Create_User

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using Oracle.ManagedDataAccess.Client;

namespace Bai3_1
{
    3 references
    public partial class CreateUser : Form
    {
        OracleConnection conn;
        Create_User u;

        1 reference
        public CreateUser()
        {
            InitializeComponent();
            CenterToScreen();
            conn = Database.Get_ConnectSys();
            u = new Create_User(conn);
        }
    }
}
```

Hình 15

- Cài đặt nút create (hình 16)

```

private void btn_cr_Click(object sender, EventArgs e)
{
    if (txt_user.Text.Equals(""))
    {
        MessageBox.Show("Chưa điền User Name!");
        txt_user.Focus();
        return;
    }

    if (txt_pass.Text.Equals(""))
    {
        MessageBox.Show("Chưa điền Password!");
        txt_pass.Focus();
        return;
    }

    int kq = u.Pro_CheckUser(txt_user.Text);
    if (kq == 0)
    {
        if (u.Pro_CreateUser(txt_user.Text, txt_pass.Text))
        {
            MessageBox.Show("Tạo tài khoản: " + txt_user.Text + " thành công");
        }
        else
        {
            MessageBox.Show("Tạo tài khoản: " + txt_user.Text + " thất bại!");
        }
    }
    else if(kq == 1)
    {
        DialogResult res = MessageBox.Show("Bạn có muốn thay đổi mật khẩu User: "
            + txt_user.Text, "Thông Báo", MessageBoxButtons.YesNo);
        if (res == DialogResult.Yes)
        {
            if (u.Pro_CreateUser(txt_user.Text, txt_pass.Text))
            {
                MessageBox.Show("Đổi mật khẩu tài khoản: " + txt_user.Text + " thành công");
                Database.Set_Database(txt_user.Text, txt_pass.Text);
                Database.Connect();
            }
            else
            {
                MessageBox.Show("Đổi mật khẩu tài khoản: " + txt_user.Text + " thất bại!");
            }
        }
    }
}

```

Hình 16

- Sự kiện đóng form

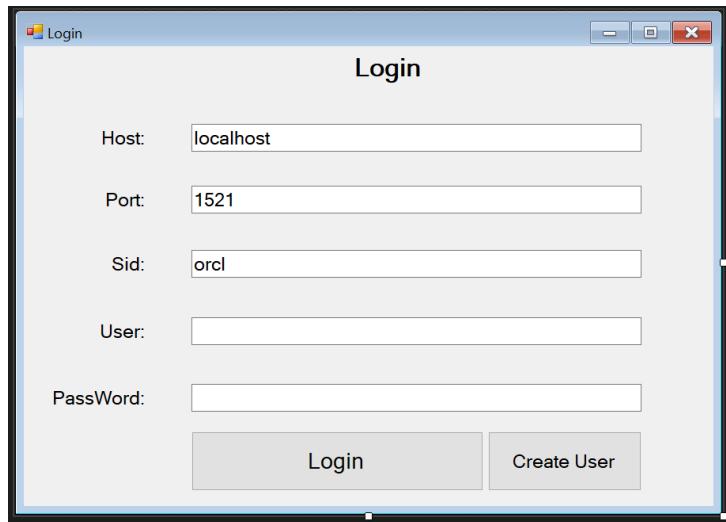
```

1 reference
private void CreateUser_FormClosing(object sender, FormClosingEventArgs e)
{
    Database.Close_ConnectSYS();
}

```

Hình 17

- Thêm nút tạo user ở form Login ở bài2_2

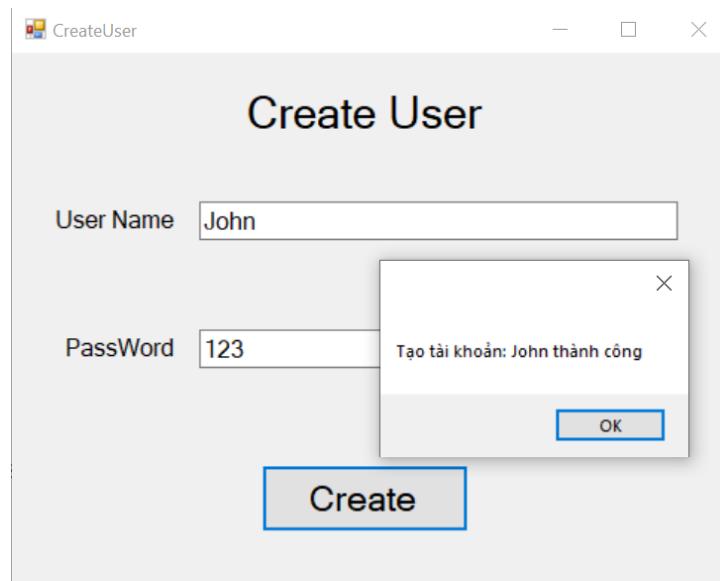


Hình 18

```
    I reference
private void btn_create_Click(object sender, EventArgs e)
{
    string host = txt_host.Text;
    string port = txt_port.Text;
    string sid = txt_sid.Text;
    Database.Set_Database(host, port, sid, "", "");
    if (Database.ConnectSys())
    {
        new CreateUser().Show();
    }
}
```

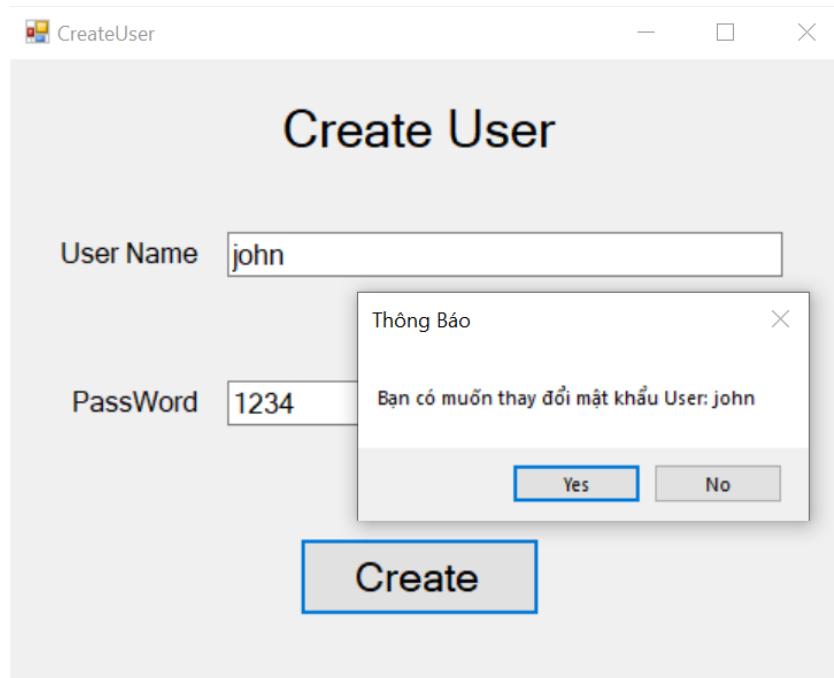
Hình 19

- Tạo các user còn lại bằng form: vd demo với user John



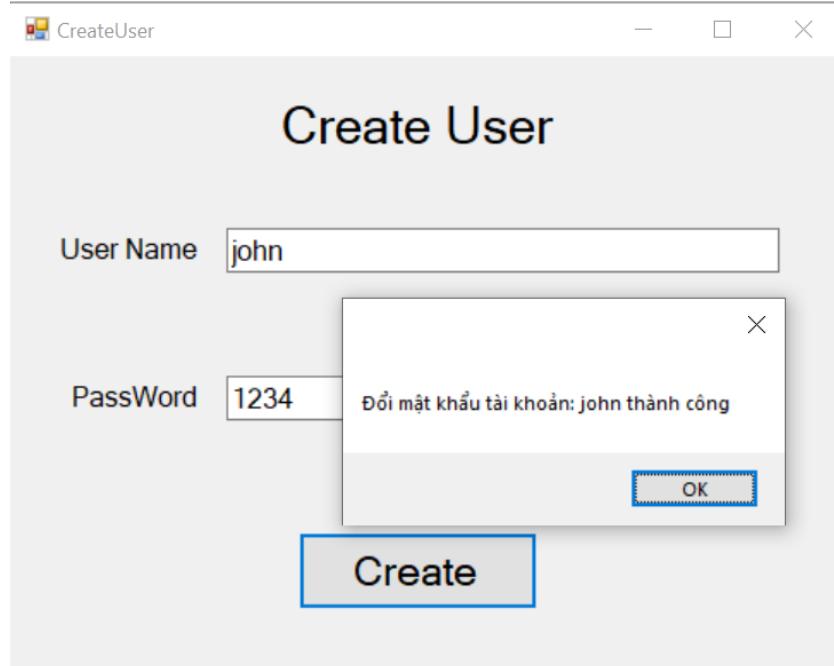
Hình 20

- Thực hiện thay đổi mật khẩu và nhấn nút Create lại



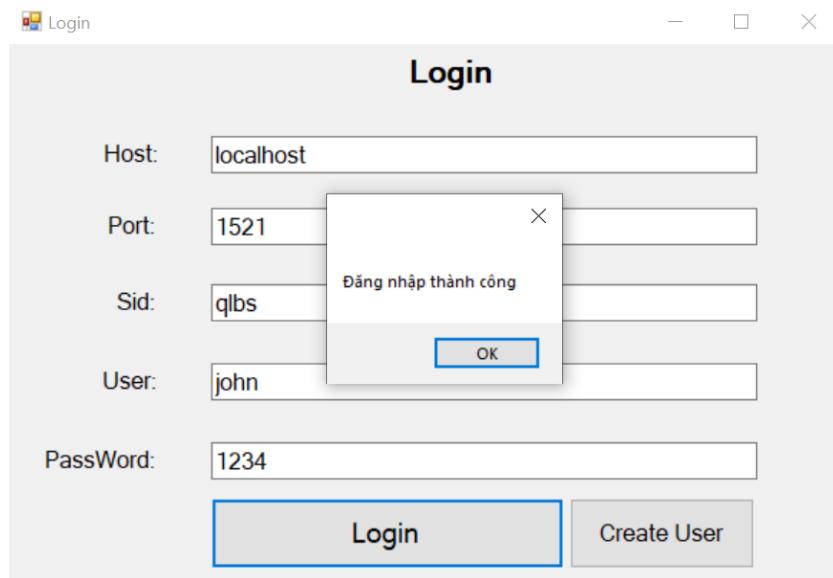
Hình 21

- Khi nhấn đồng ý lệnh alter user được chạy trong package



Hình 22

- Đăng nhập thử với tài khoản John vừa tạo



Hình 23

3.2. Bài tập 2

3.2.1. Yêu cầu

Cho bảng Attendance

(

ID INT PRIMARY KEY,

Name NVARCHAR2

)

- Tạo các role sau: DataEntry, Supervisor, và Management.
- Gán John, Joe, và Lynn vào role DataEntry, gán Fred vào role Supervisor, và gán Amy Beth vào role Management.
- Cho role DataEntry các quyền SELECT, INSERT, và UPDATE trên bảng Attendance.
- Cho role Supervisor các quyền SELECT và DELETE trên bảng Attendance.
- Cho role Management quyền SELECT trên bảng Attendance.
- Lần lượt kiểm tra kết quả phân quyền đã cấp cho các role

3.2.2. Hướng dẫn

- Tạo bảng Attendance bằng schema HR

The screenshot shows the Oracle SQL Developer interface. A code editor window contains the following SQL script:

```
1 --connect HR--
2 create table Attendance
3 (
4     ID INT PRIMARY KEY,
5     Name NVARCHAR2(100)
6 );
```

Below the code editor, a message in the 'Script Output' window reads: "Table ATTENDANCE created." The status bar at the bottom of the interface indicates: "Task completed in 0.143 seconds".

Hình 24

- Gán quyền cho HR để có thể tạo và chỉnh sửa thủ tục (Procedure). Đăng nhập vào HR tạo thủ tục dùng để truy vấn đến bảng Attendance

```
8 --connect SYS--
9 grant create any procedure, execute any procedure, drop any procedure, alter any procedure to hr;
10
11 --connect HR--
12 --Select--
13 create or replace procedure pro_select_Attendance(cur out sys_refcursor)
14 is
15 begin
16     open cur for
17         select * from hr.attendance;
18 end;
```

Hình 25

- Tạo tiếp thủ tục dùng để thêm dữ liệu cho bảng Attendance

```
19 --Insert-----
20 create or replace procedure pro_insert_Attendance(Id in int, Name in varchar2)
21 is
22 begin
23     insert into hr.attendance
24     values (Id, Name);
25     commit;
26 end;
```

Hình 26

- Tạo thủ tục dùng để sửa dữ liệu trên bảng Attendance

```
27 --Update-----
28 create or replace procedure pro_update_Attendance(Id_Up in int, Name_up in varchar2)
29 is
30 begin
31     update hr.attendance
32     set Name = Name_Up
33     where ID = Id_Up;
34     commit;
35 end;
```

Hình 27

- Tạo thủ tục dùng để xóa dữ liệu trên bảng Attendance

```

36 |-----Delete-----
37 |create or replace procedure pro_delete_Attendance(Id_Delete in int)
38 |is
39 |begin
40 |    delete hr.attendance
41 |    where ID = Id_Delete;
42 |    commit;
43 |end;

```

Hình 28

- Tạo Package dùng để phân quyền gồm có các thủ tục như hình

```

52 |--2.2--
53 |--connect SYS--
54 |create or replace package pkg_PhanQuyen
55 |as
56 |    --1
57 |    procedure pro_select_procedure_user(useowner in varchar2, pro_type in varchar2, cur out sys_refcursor);
58 |    --2
59 |    procedure pro_select_user(cur out sys_refcursor);
60 |    --3
61 |    procedure pro_select_roles(cur out sys_refcursor);
62 |    --4
63 |    procedure pro_user_roles(username in varchar2, cur out sys_refcursor);
64 |    --5
65 |    procedure pro_user_roles_check(username in varchar2, roles in varchar2, cout out number);
66 |    --6
67 |    procedure pro_select_table(username in varchar2, cur out sys_refcursor);
68 |    --7
69 |    procedure pro_select_grant(username in varchar2, userschema in varchar2, tablename in varchar2, cur out sys_refcursor);
70 |    --8
71 |    procedure pro_select_grant_user(username in varchar2, cur out sys_refcursor);
72 |    --9
73 |    procedure pro_grant_revoke(username in varchar2, schema_user in varchar2, pro_tab in varchar2, type_pro in varchar2, dk in
74 |    --10
75 |    procedure pro_grant_revoke_Roles(username in varchar2, roles in varchar2, dk in number);
76 |end;
--
```

Hình 29

- Phần body Package đầu tiên viết 2 thủ tục dùng để truy vấn tất cả các Procedure, Function, Package của Schema được truyền vào

```

79 |create or replace package body pkg_PhanQuyen
80 |as
81 |    --1--
82 |    procedure pro_select_procedure_user(useowner in varchar2, pro_type in varchar2, cur out sys_refcursor)
83 |    is
84 |    begin
85 |        open cur for
86 |            select object_name from dba_procedures where owner = useowner and object_type = pro_type;
87 |    end;
88 |

```

Hình 30

- Thủ tục dùng để truy vấn tất cả các User có trạng thái Open

```

89      --2--
90  procedure pro_select_user(cur out sys_refcursor)
91  is
92  begin
93      open cur for
94          select username from dba_users where account_status = 'OPEN';
95  end;
96

```

Hình 31

- Thủ tục truy vấn tất cả các Role có trong cơ sở dữ liệu

```

97      --3--
98  procedure pro_select_roles(cur out sys_refcursor)
99  is
100 begin
101     open cur for
102         select role from dba_roles;
103 end;

```

Hình 32

- Thủ tục truy vấn tất cả các Role được cấp cho User

```

104      --4--
105  procedure pro_user_roles(username in varchar2, cur out sys_refcursor)
106  is
107  begin
108      open cur for
109          select granted_role from dba_role_privs where grantee = username;
110 end;

```

Hình 33

- Thủ tục check xem Role có được gán cho User hay không

```

111      --5--
112  procedure pro_user_roles_check(username in varchar2, roles in varchar2, cout out number)
113  is
114  begin
115      select COUNT(*) into cout from dba_role_privs where grantee = username and granted_role = roles;
116 end;

```

Hình 34

- Thủ tục truy vấn tất cả các Table thuộc Schema được truyền vào

```

117 |     --6--
118 | procedure pro_select_table(username in varchar2, cur out sys_refcursor)
119 | is
120 | begin
121 |     open cur for
122 |         select table_name from dba_all_tables where owner = username;
123 |

```

Hình 35

- Thủ tục truy vấn các quyền của Table được cấp cho User

```

124 |     --7--
125 | procedure pro_select_grant(username in varchar2, userschema in varchar2,
126 |                                 tablename in varchar2, cur out sys_refcursor)
127 | is
128 | begin
129 |     open cur for
130 |         select privilege from dba_tab_privs where grantee = username and
131 |                                         table_name = tablename and owner = userschema;
132 |

```

Hình 36

- Thủ tục truy vấn tất cả các Procedure, Function, Package được gán quyền cho User

```

133 |     --8--
134 | procedure pro_select_grant_user(username in varchar2, cur out sys_refcursor)
135 | is
136 | begin
137 |     open cur for
138 |         select table_name, type, owner from dba_tab_privs where grantee = username and
139 |                                         type in ('PROCEDURE', 'FUNCTION', 'PACKAGE');
140 |

```

Hình 37

- Thủ tục dùng để Grant (gán quyền) và Revoke (thu hồi quyền) của Table, Procedure, Function, Package cho User tùy theo điều kiện truyền vào

```

141 |     --9--
142 | procedure pro_grant_revoke(username in varchar2, schema_user in varchar2,
143 |                               pro_tab in varchar2, type_pro in varchar2, dk in number)
144 | is
145 | begin
146 |     if dk = 1 then
147 |         execute immediate 'grant ' || type_pro || ' on ' || schema_user || '.' || pro_tab || ' to ' || username;
148 |     else
149 |         execute immediate 'revoke ' || type_pro || ' on ' || schema_user || '.' || pro_tab || ' from ' || username;
150 |     end if;
151 |

```

Hình 38

- Thủ tục dùng để gán/thu hồi Role cho User

```

152    --10--
153  procedure pro_grant_revoke_Roles(username in varchar2, roles in varchar2, dk in number)
154  is
155  begin
156    if dk = 1 then
157      execute immediate 'grant ' || roles || ' to ' || username;
158    else
159      execute immediate 'revoke ' || roles || ' from ' || username;
160    end if;
161  end;
162 end;

```

Hình 39

Đăng nhập vào SYS và chạy lệnh gán quyền chạy Package cho User: sysuser

```
grant Execute on sys.pkg_PhanQuyen to sysuser;
```

Hình 40

- Thực hiện tạo và cấp Role cho User trên SQL Developer: Tạo các role sau: DataEntry, Supervisor, và Management.

```

9  --connect SYS--
10 create role DataEntry;
11
12 create role Supervisor;
13
14 create role Management;

```

```

Script Output x
SQL Editor Task completed in 0.088 seconds

Role DATAENTRY created.

Role SUPERVISOR created.

Role MANAGEMENT created.

```

Hình 41

- Gán John, Joe, và Lynn vào role DataEntry, gán Fred vào role Supervisor, và gán Amy Beth vào role Management.

```

16 --2.2--
17 --connect SYS--
18 grant DataEntry to John, Joe, Lynn;
19
20 grant Supervisor to Fred;
21
22 grant Management to Amy, Beth;

```

```

Script Output X
Grant succeeded.

Grant succeeded.

Grant succeeded.

```

Hình 42

- Cho role DataEntry các quyền SELECT, INSERT, và UPDATE trên bảng Attendance.

```

24 --2.3--
25 --connect SYS--
26 grant select, insert, update on hr.Attendance to DataEntry;

```

```

Script Output X
Grant succeeded.

```

Hình 43

- Cho role Supervisor các quyền SELECT và DELETE trên bảng Attendance.

```

28 --2.4--
29 --connect SYS--
30 grant select, delete on hr.Attendance to Supervisor;

```

```

Script Output X
Grant succeeded.

```

Hình 44

- Cho role Management quyền SELECT trên bảng Attendance.

```

32 --2.5--
33 --connect SYS--
34 grant select on hr.Attendance to Management;

```

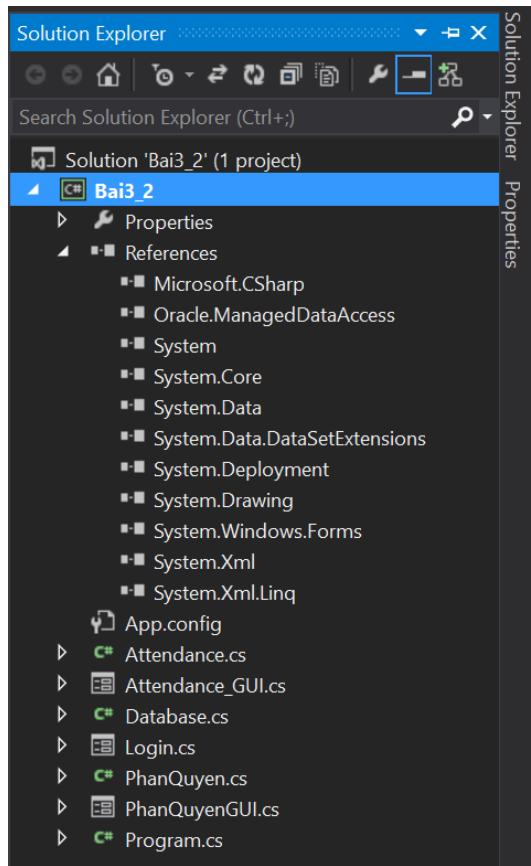
```

Script Output X
Grant succeeded.

```

Hình 45

- Phân quyền



Hình 46

- Tạo Class PhanQuyen

```
PhanQuyen.cs  ✎ X
Bai3_2
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  using Oracle.ManagedDataAccess.Client;
8  using Oracle.ManagedDataAccess.Types;
9  using System.Data;
10 using System.Windows.Forms;
11
12 namespace Bai3_2
13 {
14     public class PhanQuyen
15     {
16         OracleConnection conn;
17
18         public PhanQuyen(OracleConnection conn)
19         {
20             this.conn = conn;
21         }
22     }
}
```

Hình 47

- Hàm truy vấn User

```
1 reference
public OracleDataReader Get_User()
{
    try
    {
        string Procedure = "sys.pkg_PhanQuyen.pro_select_user";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@Result";
        resultParam.OracleDbType = OracleDbType.RefCursor;
        resultParam.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(resultParam);

        cmd.ExecuteNonQuery();

        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();

            return ret;
        }
    }
    catch
    {
        MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_select_user");
        return null;
    }
    return null;
}
```

Hình 48

- Hàm truy vấn Role (hình 49-50)

```
1 reference
public OracleDataReader Get_Roles()
{
    try
    {
        string Procedure = "sys.pkg_PhanQuyen.pro_select_roles";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@Result";
        resultParam.OracleDbType = OracleDbType.RefCursor;
        resultParam.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(resultParam);

        cmd.ExecuteNonQuery();
```

Hình 49

```
        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();

            return ret;
        }
    }
    catch
    {
        MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_select_roles");
        return null;
    }
    return null;
}
```

Hình 50

- Hàm truy vấn Procedure, Function, Package dựa trên User truyền vào hàm (hình 51-52)

```
3 references
public OracleDataReader Get_Procedure_User(string userowner, string type)
{
    try
    {
        string Procedure = "sys.pkg_PhanQuyen.pro_select_procedure_user";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter UserOwner = new OracleParameter();
        UserOwner.ParameterName = "@userowner";
        UserOwner.OracleDbType = OracleDbType.Varchar2;
        UserOwner.Value = userowner.ToUpper();
        UserOwner.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(UserOwner);

        OracleParameter pro_type = new OracleParameter();
        pro_type.ParameterName = "@pro_type";
        pro_type.OracleDbType = OracleDbType.Varchar2;
        pro_type.Value = type.ToUpper();
        pro_type.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(pro_type);
    }
}
```

Hình 51

```

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@Result";
        resultParam.OracleDbType = OracleDbType.RefCursor;
        resultParam.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(resultParam);

        cmd.ExecuteNonQuery();

        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();

            return ret;
        }
    }
    catch
    {
        MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_select_procedure_user");
        return null;
    }
    return null;
}

```

Hình 52

- Hàm truy vấn Table (hình 53-54)

```

1 reference
public OracleDataReader Get_Table_User(string userowner)
{
    try
    {
        string Procedure = "sys.pkg_PhanQuyen.pro_select_table";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter UserOwner = new OracleParameter();
        UserOwner.ParameterName = "@userowner";
        UserOwner.OracleDbType = OracleDbType.Varchar2;
        UserOwner.Value = userowner.ToUpper();
        UserOwner.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(UserOwner);

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@Result";
        resultParam.OracleDbType = OracleDbType.RefCursor;
        resultParam.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(resultParam);

        cmd.ExecuteNonQuery();
    }
}

```

Hình 53

```

        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();

            return ret;
        }
    catch
    {
        MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_select_table");
        return null;
    }
    return null;
}

```

Hình 54

- Hàm truy vấn tất cả Role được gán cho User

```

public DataTable Get_Roles_User(string username)
{
    try
    {
        string Procedure = "sys.pkg_PhanQuyen.pro_user_roles";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter UserName = new OracleParameter();
        UserName.ParameterName = "@username";
        UserName.OracleDbType = OracleDbType.Varchar2;
        UserName.Value = username.ToUpper();
        UserName.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(UserName);

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@Result";
        resultParam.OracleDbType = OracleDbType.RefCursor;
        resultParam.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(resultParam);

        cmd.ExecuteNonQuery();

        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();
            DataTable data = new DataTable();
            data.Load(ret);
            return data;
        }
    }
    catch
    {
        MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_user_roles");
        return null;
    }
    return null;
}

```

Hình 55

- Hàm check Role có được gán cho User không

```
    public int Get_Roles_User_Check(string username, string roles)
    {
        try
        {
            string Procedure = "sys.pkg_PhanQuyen.pro_user_roles_check";

            OracleCommand cmd = new OracleCommand();
            cmd.Connection = conn;
            cmd.CommandText = Procedure;
            cmd.CommandType = CommandType.StoredProcedure;

            OracleParameter UserName = new OracleParameter();
            UserName.ParameterName = "@username";
            UserName.OracleDbType = OracleDbType.Varchar2;
            UserName.Value = username.ToUpper();
            UserName.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(UserName);

            OracleParameter Roles = new OracleParameter();
            Roles.ParameterName = "@roles";
            Roles.OracleDbType = OracleDbType.Varchar2;
            Roles.Value = roles.ToUpper();
            Roles.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(Roles);

            OracleParameter resultParam = new OracleParameter();
            resultParam.ParameterName = "@Result";
            resultParam.OracleDbType = OracleDbType.Int16;
            resultParam.Direction = ParameterDirection.Output;
            cmd.Parameters.Add(resultParam);

            cmd.ExecuteNonQuery();

            if (resultParam.Value != DBNull.Value)
            {
                return int.Parse(resultParam.Value.ToString());
            }
        }
        catch
        {
            MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_user_roles");
            return -1;
        }
        return -1;
    }
}
```

Hình 56

- Truy vấn tất cả các Procedure, Function, Package được gán quyền cho User

```
public DataTable Get_Grant_User(string username)
{
    try
    {
        string Procedure = "sys.pkg_PhanQuyen.pro_select_grant_user";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter UserName = new OracleParameter();
        UserName.ParameterName = "@username";
        UserName.OracleDbType = OracleDbType.Varchar2;
        UserName.Value = username.ToUpper();
        UserName.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(UserName);

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@Result";
        resultParam.OracleDbType = OracleDbType.RefCursor;
        resultParam.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(resultParam);

        cmd.ExecuteNonQuery();

        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();
            DataTable data = new DataTable();
            data.Load(ret);
            return data;
        }
    }
    catch
    {
        MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_select_grant_user");
        return null;
    }
}
return null;
}
```

Hình 57

- Hàm truy vấn quyền của Table được gán cho User

```

3 references
public OracleDataReader Get_Grant(string username, string userschema, string tab)
{
    try
    {
        string Procedure = "sys.pkg_PhanQuyen.pro_select_grant";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter UserName = new OracleParameter();
        UserName.ParameterName = "@username";
        UserName.OracleDbType = OracleDbType.Varchar2;
        UserName.Value = username.ToUpper();
        UserName.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(UserName);

        OracleParameter UserSchema = new OracleParameter();
        UserSchema.ParameterName = "@userschema";
        UserSchema.OracleDbType = OracleDbType.Varchar2;
        UserSchema.Value = userschema.ToUpper();
        UserSchema.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(UserSchema);

        OracleParameter TableName = new OracleParameter();
        TableName.ParameterName = "@tablename";
        TableName.OracleDbType = OracleDbType.Varchar2;
        TableName.Value = tab.ToUpper();
        TableName.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(TableName);

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@Result";
        resultParam.OracleDbType = OracleDbType.RefCursor;
        resultParam.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(resultParam);

        cmd.ExecuteNonQuery();

        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();

            return ret;
        }
    }
    catch
    {
        MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_select_grant");
        return null;
    }
    return null;
}

```

Hình 58

- Hàm gán và hủy quyền của Procedure, Function, Package, Table dựa vào biến truyền vào

```

1 reference
public bool Grant_Revoke_Pro(string username, string user_schema, string pro_tab,
                               string type_pro, int dk)
{
    try
    {
        string Procedure = "sys.pkg_PhanQuyen.pro_grant_revoke";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter UserName = new OracleParameter();
        UserName.ParameterName = "@username";
        UserName.OracleDbType = OracleDbType.Varchar2;
        UserName.Value = username.ToUpper();
        UserName.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(UserName);

        OracleParameter UserSchema = new OracleParameter();
        UserSchema.ParameterName = "@userschema";
        UserSchema.OracleDbType = OracleDbType.Varchar2;
        UserSchema.Value = user_schema.ToUpper();
        UserSchema.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(UserSchema);

        OracleParameter ProTab = new OracleParameter();
        ProTab.ParameterName = "@tablename";
        ProTab.OracleDbType = OracleDbType.Varchar2;
        ProTab.Value = pro_tab.ToUpper();
        ProTab.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(ProTab);

        OracleParameter TypePro = new OracleParameter();
        TypePro.ParameterName = "@tyepro";
        TypePro.OracleDbType = OracleDbType.Varchar2;
        TypePro.Value = type_pro.ToUpper();
        TypePro.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(TypePro);

        OracleParameter DK = new OracleParameter();
        DK.ParameterName = "@dk";
        DK.OracleDbType = OracleDbType.Int16;
        DK.Value = dk;
        DK.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(DK);

        cmd.ExecuteNonQuery();

        return true;
    }
    catch
    {
        MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_grant_revoke");
        return false;
    }
}

```

Hình 59

- Hàm gán/hủy Role cho User

```
    public bool Grant_Revoke_Role(string username, string role, int dk)
    {
        try
        {
            string Procedure = "sys.pkg_PhanQuyen.pro_grant_revoke_Roles";

            OracleCommand cmd = new OracleCommand();
            cmd.Connection = conn;
            cmd.CommandText = Procedure;
            cmd.CommandType = CommandType.StoredProcedure;

            OracleParameter UserName = new OracleParameter();
            UserName.ParameterName = "@username";
            UserName.OracleDbType = OracleDbType.Varchar2;
            UserName.Value = username.ToUpper();
            UserName.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(UserName);

            OracleParameter Role = new OracleParameter();
            Role.ParameterName = "@userschema";
            Role.OracleDbType = OracleDbType.Varchar2;
            Role.Value = role.ToUpper();
            Role.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(Role);

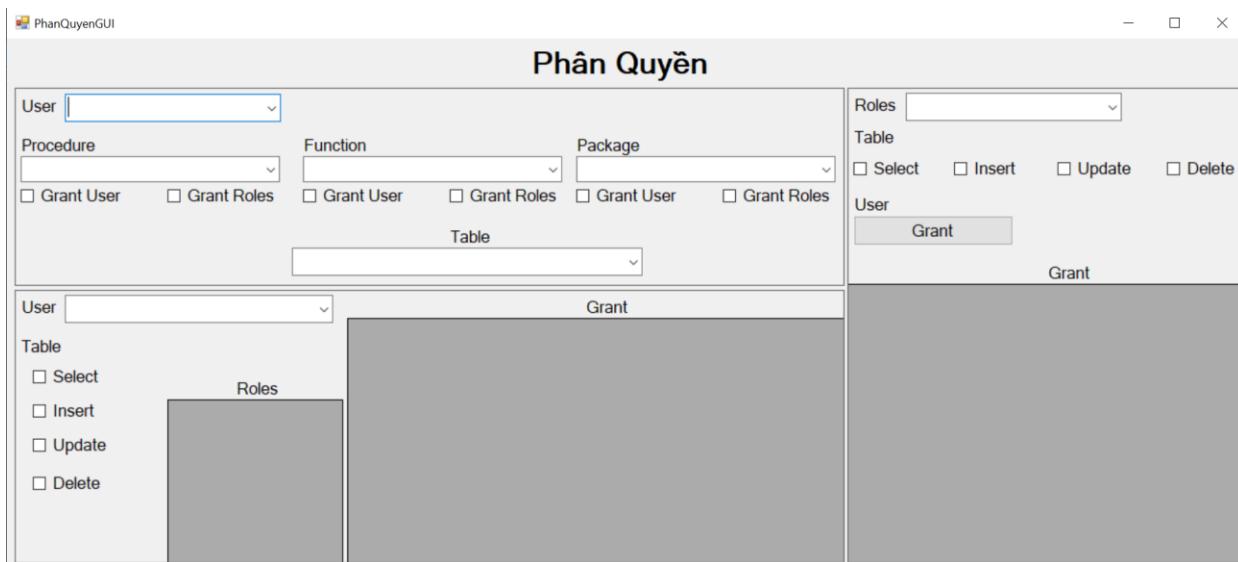
            OracleParameter DK = new OracleParameter();
            DK.ParameterName = "@dk";
            DK.OracleDbType = OracleDbType.Int16;
            DK.Value = dk;
            DK.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(DK);

            cmd.ExecuteNonQuery();

            return true;
        }
        catch
        {
            MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_grant_revoke_Roles");
            return false;
        }
    }
}
```

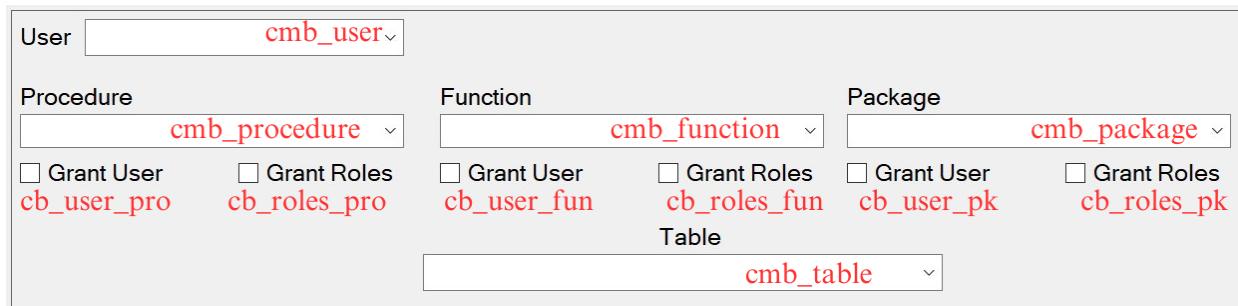
Hình 60

- Tạo Class PhanQuyenGUI và thiết kế giao diện như sau



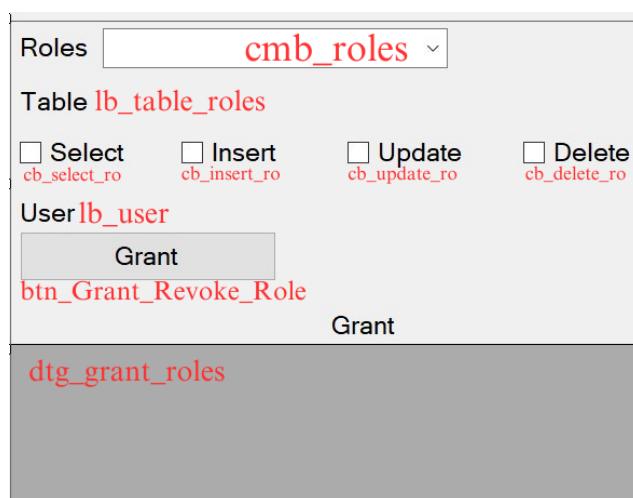
Hình 61

- Tên của các toolbox trong panel Schema



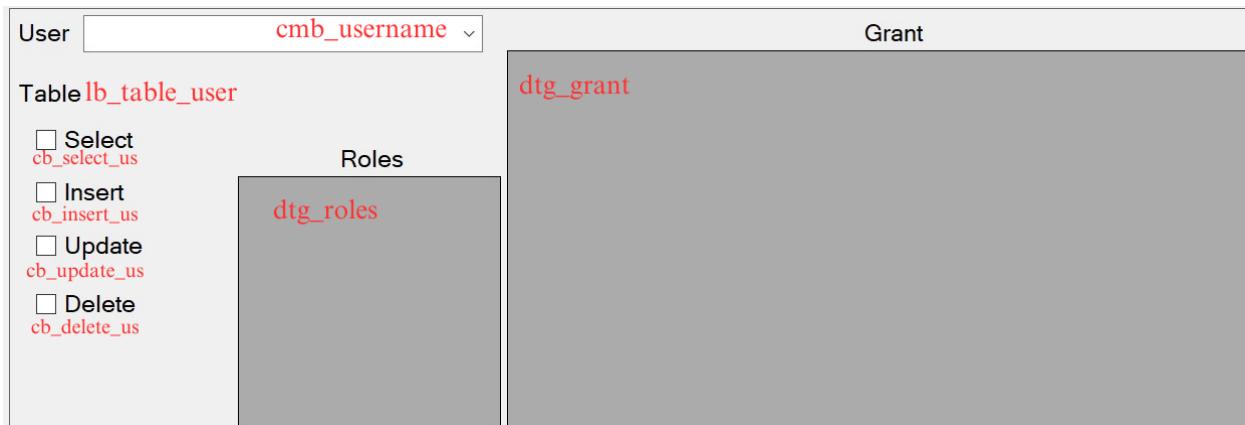
Hình 62

- Tên của các toolbox trong panel Role



Hình 63

- Tên của các toolbox trong panel User



Hình 64

- Khai báo các biến cần thiết

```

PhanQuyenGUI.cs  X  PhanQuyenGUI.cs [Design]
Bai3_2

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 using Oracle.ManagedDataAccess.Client;
12 namespace Bai3_2
13 {
14     public partial class PhanQuyenGUI : Form
15     {
16         OracleConnection conn;
17         PhanQuyen p;
18         public PhanQuyenGUI()
19         {
20             InitializeComponent();
21             CenterToScreen();
22             conn = Database.Get_ConnectSys();
23             p = new PhanQuyen(conn);
24
25             Load_User();
26             Load_Roles();
27         }
28     }

```

Hình 65

- Hàm thiết lập màu cho các checkbox quyền trên bảng của User

```

3 references
void set_Color_Checkbox_user()
{
    if (cb_select_us.Checked)
        cb_select_us.ForeColor = Color.Green;
    else
        cb_select_us.ForeColor = Color.Red;

    if (cb_insert_us.Checked)
        cb_insert_us.ForeColor = Color.Green;
    else
        cb_insert_us.ForeColor = Color.Red;

    if (cb_update_us.Checked)
        cb_update_us.ForeColor = Color.Green;
    else
        cb_update_us.ForeColor = Color.Red;

    if (cb_delete_us.Checked)
        cb_delete_us.ForeColor = Color.Green;
    else
        cb_delete_us.ForeColor = Color.Red;
}

```

Hình 66

- Hàm thiết lập màu cho các checkbox quyền trên bảng của Role

```

3 references
void set_Color_Checkbox_roles()
{
    if (cb_select_ro.Checked)
        cb_select_ro.ForeColor = Color.Green;
    else
        cb_select_ro.ForeColor = Color.Red;

    if (cb_insert_ro.Checked)
        cb_insert_ro.ForeColor = Color.Green;
    else
        cb_insert_ro.ForeColor = Color.Red;

    if (cb_update_ro.Checked)
        cb_update_ro.ForeColor = Color.Green;
    else
        cb_update_ro.ForeColor = Color.Red;

    if (cb_delete_ro.Checked)
        cb_delete_ro.ForeColor = Color.Green;
    else
        cb_delete_ro.ForeColor = Color.Red;
}

```

Hình 67

- Hàm thiết lập màu cho các checkbox quyền của User trên Procedure, Function, Package

```

5 references
void Set_Color_Grant_User()
{
    if (cb_user_pro.Checked)
        cb_user_pro.ForeColor = Color.Green;
    else
        cb_user_pro.ForeColor = Color.Red;

    if (cb_user_fun.Checked)
        cb_user_fun.ForeColor = Color.Green;
    else
        cb_user_fun.ForeColor = Color.Red;

    if (cb_user_pk.Checked)
        cb_user_pk.ForeColor = Color.Green;
    else
        cb_user_pk.ForeColor = Color.Red;
}

```

Hình 68

- Hàm thiết lập màu cho các checkbox quyền của Role trên Procedure, Function, Package

```

5 references
void Set_Color_Grant_Roles()
{
    if (cb_roles_pro.Checked)
        cb_roles_pro.ForeColor = Color.Green;
    else
        cb_roles_pro.ForeColor = Color.Red;

    if (cb_roles_fun.Checked)
        cb_roles_fun.ForeColor = Color.Green;
    else
        cb_roles_fun.ForeColor = Color.Red;

    if (cb_roles_pk.Checked)
        cb_roles_pk.ForeColor = Color.Green;
    else
        cb_roles_pk.ForeColor = Color.Red;
}

```

Hình 69

- Hàm gán tên bảng vào lable khi Table được chọn

```

1 reference
void Set_Table_Table()
{
    string t = "Table: ";
    if (cmb_table.SelectedItem != null)
    {
        t += cmb_table.SelectedItem.ToString();
    }

    lb_table_roles.Text = t;
    lb_table_user.Text = t;
}

```

Hình 70

- Hàm thiết lập text hiển thị trên nút gán quyền/hủy quyền dựa vào hàm truy vấn quyền giữa Role và User

```

3 references
void Set_Text_Buttun(string user, string role)
{
    int kq = p.Get_Roles_User_Check(user, role);
    if (kq == 1)
    {
        btn_Grant_Revoke_Role.Text = "Revoke";
    }
    else if (kq == 0)
    {
        btn_Grant_Revoke_Role.Text = "Grant";
    }
}

```

Hình 71

- Hàm Load User vào ComboBox

```

1 reference
void Load_User()
{
    OracleDataReader read = p.Get_User();
    while (read.Read())
    {
        cmb_user.Items.Add(read[0].ToString());
        cmb_username.Items.Add(read[0].ToString());
    }
    read.Close();
    cmb_user.SelectedIndex = 0;
    cmb_username.SelectedIndex = 0;
}

```

Hình 72

- Hàm Load Role vào ComboBox

```

1 reference
void Load_Roles()
{
    OracleDataReader read = p.Get_Roles();
    while (read.Read())
    {
        cmb_roles.Items.Add(read[0].ToString());
    }
    read.Close();

    cmb_roles.SelectedIndex = 0;
}

```

Hình 73

- Hàm xóa tất cả item tròn ComboBox và Select ComboBox khi có dữ liệu mới được truy vấn

```

1 reference
void Clear_Combobox()
{
    cmb_procedure.Items.Clear();
    cmb_function.Items.Clear();
    cmb_package.Items.Clear();
    cmb_table.Items.Clear();
}

1 reference
void Select_Combobox()
{
    if (cmb_procedure.Items.Count == 0)
        cmb_procedure.Items.Add("");
    if (cmb_function.Items.Count == 0)
        cmb_function.Items.Add("");
    if (cmb_package.Items.Count == 0)
        cmb_package.Items.Add("");
    if (cmb_table.Items.Count == 0)
        cmb_table.Items.Add("");

    cmb_procedure.SelectedIndex = 0;
    cmb_function.SelectedIndex = 0;
    cmb_package.SelectedIndex = 0;
    cmb_table.SelectedIndex = 0;
}

```

Hình 74

- Hàm Load Procedure, Function, Package vào ComboBox dựa vào ComboBox User đang được chọn

```

1 reference
void Load_pro_user(string userowner)
{
    Clear_Combobox();

    OracleDataReader read_pro = p.Get_Procedure_User(userowner, "PROCEDURE");
    while (read_pro.Read())
    {
        cmb_procedure.Items.Add(read_pro[0].ToString());
    }
    read_pro.Close();

    OracleDataReader read_fun = p.Get_Procedure_User(userowner, "FUNCTION");
    while (read_fun.Read())
    {
        cmb_function.Items.Add(read_fun[0].ToString());
    }
    read_fun.Close();

    OracleDataReader read_pack = p.Get_Procedure_User(userowner, "PACKAGE");
    while (read_pack.Read())
    {
        cmb_package.Items.Add(read_pack[0].ToString());
    }
    read_pack.Close();

    OracleDataReader read_tab = p.Get_Table_User(userowner);
    while (read_tab.Read())
    {
        cmb_table.Items.Add(read_tab[0].ToString());
    }
    read_tab.Close();

    Select_Combobox();
}

```

Hình 75

- Các hàm Load tất cả các quyền vào DataGridView của User và Role

```

2 references
void Load_Roles_User(string user)
{
    dtg_roles.DataSource = p.Get_Roles_User(user);
}

2 references
void Load_Grant_User(string user)
{
    dtg_grant.DataSource = p.Get_Grant_User(user);
}

2 references
void Load_Grant_Roles(string roles)
{
    dtg_grant_roles.DataSource = p.Get_Grant_User(roles);
}

```

Hình 76

- Hàm Load tất cả các quyền của bảng được gán cho User (quyền được gán CheckBox tương ứng sẽ tự động tick)

```

    void Load_Grant_Table_User(string user, string userschema, string table)
{
    OracleDataReader read = p.Get_Grant(user, userschema, table);
    bool select, insert, update, delete;
    select = insert = update = delete = false;
    while (read.Read())
    {
        string r = read[0].ToString();

        if (r.Equals("SELECT"))
            select = true;

        if (r.Equals("INSERT"))
            insert = true;

        if (r.Equals("UPDATE"))
            update = true;

        if (r.Equals("DELETE"))
            delete = true;
    }
    cb_select_us.Checked = select;
    cb_insert_us.Checked = insert;
    cb_update_us.Checked = update;
    cb_delete_us.Checked = delete;
}

```

Hình 77

- Hàm Load tất cả các quyền của bảng được gán cho Role (quyền được gán CheckBox tương ứng sẽ tự động tick)

```

    void Load_Grant_Table_Roles(string roles, string userschema, string table)
{
    OracleDataReader read = p.Get_Grant(roles, userschema, table);
    bool select, insert, update, delete;
    select = insert = update = delete = false;
    while (read.Read())
    {
        string r = read[0].ToString();

        if (r.Equals("SELECT"))
            select = true;

        if (r.Equals("INSERT"))
            insert = true;

        if (r.Equals("UPDATE"))
            update = true;

        if (r.Equals("DELETE"))
            delete = true;
    }
    cb_select_ro.Checked = select;
    cb_insert_ro.Checked = insert;
    cb_update_ro.Checked = update;
    cb_delete_ro.Checked = delete;
}

```

Hình 78

- Hàm Load quyền Execute được gán cho User/Role tùy vào điều kiện truyền vào (quyền được gán CheckBox tương ứng sẽ tự động tick)

```

12 references
bool Load_Execute(string user_roles, string userschema, string pro)
{
    OracleDataReader read = p.Get_Grant(user_roles, userschema, pro);
    bool execute = false;
    while (read.Read())
    {
        if (read[0].ToString().Equals("EXECUTE"))
            execute = true;
    }
    return execute;
}

```

Hình 79

- Sự kiện chọn ComboBox User Schema

```

1 reference
private void cmb_user_SelectedIndexChanged(object sender, EventArgs e)
{
    string userowner = cmb_user.SelectedItem.ToString();
    Load_pro_user(userowner);
}

```

Hình 80

- Sự kiện chọn ComboBox User Name

```

private void cmb_username_SelectedIndexChanged(object sender, EventArgs e)
{
    string user = cmb_username.SelectedItem.ToString();
    string userschema = cmb_user.SelectedItem.ToString();
    Load_Roles_User(user);
    Load_Grant_User(user);
    lb_user.Text = "User: " + user;

    if (cmb_table.SelectedItem != null)
    {
        string table = cmb_table.SelectedItem.ToString();

        Load_Grant_Table_User(user, userschema, table);
        set_Color_Checkbox_user();
    }

    if (cmb_procedure.SelectedItem != null)
    {
        string procedure = cmb_procedure.SelectedItem.ToString();
        cb_user_pro.Checked = Load_Execute(user, userschema, procedure);
    }
}

```

Hình 81

```

        if (cmb_function.SelectedItem != null)
        {
            string function = cmb_function.SelectedItem.ToString();
            cb_user_fun.Checked = Load_Execute(user, userschema, function);
        }

        if (cmb_package.SelectedItem != null)
        {
            string package = cmb_package.SelectedItem.ToString();
            cb_user_pk.Checked = Load_Execute(user, userschema, package);
        }

        if (cmb_roles.SelectedItem != null)
        {
            string role = cmb_roles.SelectedItem.ToString();
            Set_Text_Buttun(user, role);
        }
        Set_Color_Grant_User();
    }
}

```

Hình 82

- Sự kiện chọn ComboBox Table

```

1 reference
private void cmb_table_SelectedIndexChanged(object sender, EventArgs e)
{
    string userschema = cmb_user.SelectedItem.ToString();
    Set_Table_Table();
    string table = cmb_table.SelectedItem.ToString();
    if (cmb_username.SelectedItem != null)
    {
        string user = cmb_username.SelectedItem.ToString();

        Load_Grant_Table_User(user, userschema, table);
        set_Color_Chebox_user();
    }

    if (cmb_roles.SelectedItem != null)
    {
        string roles = cmb_roles.SelectedItem.ToString();

        Load_Grant_Table_Roles(roles, userschema, table);
        set_Color_Chebox_roles();
    }
}

```

Hình 83

- Sự kiện chọn ComboBox Role

```
1 reference
private void cmb_roles_SelectedIndexChanged(object sender, EventArgs e)
{
    string userschema = cmb_user.SelectedItem.ToString();
    string role = cmb_roles.SelectedItem.ToString();
    Load_Grant_Roles(role);

    if (cmb_table.SelectedItem != null)
    {
        string table = cmb_table.SelectedItem.ToString();

        Load_Grant_Table_Roles(role, userschema, table);
        set_Color_Checkbox_roles();
    }

    if (cmb_procedure.SelectedItem != null)
    {
        string procedure = cmb_procedure.SelectedItem.ToString();
        cb_roles_pro.Checked = Load_Execute(role, userschema, procedure);
    }

    if (cmb_function.SelectedItem != null)
    {
        string function = cmb_function.SelectedItem.ToString();
        cb_roles_fun.Checked = Load_Execute(role, userschema, function);
    }

    if (cmb_package.SelectedItem != null)
    {
        string package = cmb_package.SelectedItem.ToString();
        cb_roles_pk.Checked = Load_Execute(role, userschema, package);
    }

    if (cmb_username.SelectedItem != null)
    {
        string user = cmb_username.SelectedItem.ToString();
        Set_Text_Button(user, role);
    }
    Set_Color_Grant_Roles();
}
```

Hình 84

- Sự kiện chọn ComboBox Procedure

```
1 reference
private void cmb_procedure_SelectedIndexChanged(object sender, EventArgs e)
{
    string procedure = cmb_procedure.SelectedItem.ToString();
    string userschema = cmb_user.SelectedItem.ToString();

    if (cmb_username.SelectedItem != null)
    {
        string user = cmb_username.SelectedItem.ToString();
        cb_user_pro.Checked = Load_Execute(user, userschema, procedure);
    }

    if (cmb_roles.SelectedItem != null)
    {
        string role = cmb_roles.SelectedItem.ToString();
        cb_roles_pro.Checked = Load_Execute(role, userschema, procedure);
    }

    Set_Color_Grant_Roles();
    Set_Color_Grant_User();
}
```

Hình 85

- Sự kiện chọn ComboBox Function

```
1 reference
private void cmb_function_SelectedIndexChanged(object sender, EventArgs e)
{
    string function = cmb_function.SelectedItem.ToString();
    string userschema = cmb_user.SelectedItem.ToString();

    if (cmb_username.SelectedItem != null)
    {
        string user = cmb_username.SelectedItem.ToString();
        cb_user_fun.Checked = Load_Execute(user, userschema, function);
    }

    if (cmb_roles.SelectedItem != null)
    {
        string role = cmb_roles.SelectedItem.ToString();
        cb_roles_fun.Checked = Load_Execute(role, userschema, function);
    }

    Set_Color_Grant_Roles();
    Set_Color_Grant_User();
}
```

Hình 86

- Sự kiện chọn ComboBox Package

```

1 reference
private void cmb_package_SelectedIndexChanged(object sender, EventArgs e)
{
    string package = cmb_package.SelectedItem.ToString();
    string userschema = cmb_user.SelectedItem.ToString();

    if (cmb_username.SelectedItem != null)
    {
        string user = cmb_username.SelectedItem.ToString();
        cb_user_pk.Checked = Load_Execute(user, userschema, package);
    }

    if (cmb_roles.SelectedItem != null)
    {
        string role = cmb_roles.SelectedItem.ToString();
        cb_roles_pk.Checked = Load_Execute(role, userschema, package);
    }

    Set_Color_Grant_Roles();
    Set_Color_Grant_User();
}

```

Hình 87

- Hàm gán/hủy quyền Procedure, Function, Package, Table cho User, Role tùy vào dữ liệu truyền vào

```

4 references
bool Grant_Revoke_pro(string user_roles, string schema, string pro_tab, string type_user,
                      string type_pro_tab, string type, bool grant_revoke)
{
    if (pro_tab.Equals(""))
    {
        MessageBox.Show("Mục " + type_pro_tab + " Trống!!!");
        return false;
    }
    if (user_roles.Equals(""))
    {
        MessageBox.Show("Mục " + type_user + " Trống!!!");
        return false;
    }

    int dk;
    DialogResult res;
    if (grant_revoke)
    {
        dk = 1;
        res = MessageBox.Show("Bạn muốn gán quyền " + type + " " + type_pro_tab + ":" +
                             " " + pro_tab + " cho " + type_user + ":" + user_roles + " không??",
                             "Thông báo", MessageBoxButtons.YesNo);
    }
}

```

Hình 88

```

        }
        else
        {
            dk = 0;
            res = MessageBox.Show("Bạn muốn hủy quyền " + type + " " + type_pro_tab + ":" +
                " " + pro_tab + " cho " + type_user + ": " + user_roles + " không??",
                "Thông báo", MessageBoxButtons.YesNo);
        }

        if (res == DialogResult.Yes)
        {

            if (p.Grant_Revoke_Pro(user_roles, schema, pro_tab, type, dk))
            {
                if (grant_revocate)
                    MessageBox.Show("Gán quyền thành công");
                else
                    MessageBox.Show("Thu hồi quyền thành công");
            }
            else
                return false;

            return true;
        }
        else
            return false;
    }
}

```

Hình 89

- Hàm gán/hủy gán Role cho User

```

    bool Grant_Revoke_Role(string user, string role, int dk)
    {
        if (user.Equals(""))
        {
            MessageBox.Show("Mục User trống!!!");
            return false;
        }

        if (role.Equals(""))
        {
            MessageBox.Show("Mục Role trống!!!");
            return false;
        }

        DialogResult res;
        if (dk == 1)
        {
            res = MessageBox.Show("Bạn muốn gán User: " + user + " vào Role: " + role + " không??",
                "Thông báo", MessageBoxButtons.YesNo);
        }
        else
        {
            res = MessageBox.Show("Bạn muốn gỡ User: " + user + " vào Role: " + role + " không??",
                "Thông báo", MessageBoxButtons.YesNo);
        }
    }
}

```

Hình 90

```

        if (res == DialogResult.Yes)
    {
        if (p.Grant_Revoke_Role(user, role, dk))
        {
            if (dk == 1)
                MessageBox.Show("Gán nhóm quyền thành công");
            else
                MessageBox.Show("Thu hồi nhóm quyền thành công");
        }
        else
            return false;
    }
    return true;
}
else
    return false;
}

```

Hình 91

- Hàm lấy dữ liệu và gọi hàm gán/hủy quyền Procedure, Function, Package cho User

```

3 references
void Grant_Revoke_Pro_User(ComboBox cmb, CheckBox c, string pro)
{
    bool check = c.Checked;
    string procedure = cmb.SelectedItem.ToString();
    string user = cmb_username.SelectedItem.ToString();
    string schema = cmb_user.SelectedItem.ToString();
    if (Grant_Revoke_pro(user, schema, procedure, "User", pro, "Execute", check))
    {
        Set_Color_Grant_User();
        Load_Grant_User(user);
    }
    else
        c.Checked = !check;
}

```

Hình 92

- Hàm lấy dữ liệu và gọi hàm gán/hủy quyền Procedure, Function, Package cho Role

```

3 references
void Grant_Revoke_Pro_Role(ComboBox cmb ,CheckBox c, string pro)
{
    bool check = c.Checked;
    string procedure = cmb.SelectedItem.ToString();
    string role = cmb_roles.SelectedItem.ToString();
    string schema = cmb_user.SelectedItem.ToString();
    if (Grant_Revoke_pro(role, schema, procedure, "Role", pro, "Execute", check))
    {
        Set_Color_Grant_Roles();
        Load_Grant_Roles(role);
    }
    else
        c.Checked = !check;
}

```

Hình 93

- Hàm lấy dữ liệu và gọi hàm gán/hủy quyền Table cho User

```
4 references
void Grant_Revoke_Table_User(CheckBox c, string type)
{
    bool check = c.Checked;
    string table = cmb_table.SelectedItem.ToString();
    string user = cmb_username.SelectedItem.ToString();
    string schema = cmb_user.SelectedItem.ToString();
    if (Grant_Revoke_pro(user, schema, table, "User", "Table", type, check))
        set_Color_Checbox_user();
    else
        c.Checked = !check;
}
```

Hình 94

- Hàm lấy dữ liệu và gọi hàm gán/hủy quyền Table cho Role

```
4 references
void Grant_Revoke_Table_Role(CheckBox c, string type)
{
    bool check = c.Checked;
    string table = cmb_table.SelectedItem.ToString();
    string role = cmb_roles.SelectedItem.ToString();
    string schema = cmb_user.SelectedItem.ToString();
    if (Grant_Revoke_pro(role, schema, table, "Role", "Table", type, check))
        set_Color_Checbox_roles();
    else
        c.Checked = !check;
}
```

Hình 95

- Sự kiện tick CheckBox gán quyền Procedure cho User, Role

```
1 reference
private void cb_user_pro_Click(object sender, EventArgs e)
{
    Grant_Revoke_Pro_User(cmb_procedure, cb_user_pro, "Procedure");
}

1 reference
private void cb_roles_pro_Click(object sender, EventArgs e)
{
    Grant_Revoke_Pro_Role(cmb_procedure, cb_roles_pro, "Procedure");
}
```

Hình 96

- Sự kiện tick CheckBox gán quyền Function cho User, Role

```

    1 reference
    private void cb_user_fun_Click(object sender, EventArgs e)
    {
        Grant_Revoke_Pro_User(cmb_function, cb_user_fun, "Function");
    }

    1 reference
    private void cb_roles_fun_Click(object sender, EventArgs e)
    {
        Grant_Revoke_Pro_Role(cmb_function ,cb_roles_fun, "Function");
    }

```

Hình 97

- Sự kiện tick CheckBox gán quyền Package cho User, Role

```

    1 reference
    private void cb_user_pk_Click(object sender, EventArgs e)
    {
        Grant_Revoke_Pro_User(cmb_package, cb_user_pk, "Package");
    }

    1 reference
    private void cb_roles_pk_Click(object sender, EventArgs e)
    {
        Grant_Revoke_Pro_Role(cmb_package ,cb_roles_pk, "Package");
    }

```

Hình 98

- Sự kiện tick các CheckBox gán quyền (Select, Insert, Update, Delete) Table cho User

```

    1 reference
    private void cb_select_us_Click(object sender, EventArgs e)
    {
        Grant_Revoke_Table_User(cb_select_us, "Select");
    }

    1 reference
    private void cb_insert_us_Click(object sender, EventArgs e)
    {
        Grant_Revoke_Table_User(cb_insert_us, "Insert");
    }

    1 reference
    private void cb_update_us_Click(object sender, EventArgs e)
    {
        Grant_Revoke_Table_User(cb_update_us, "Update");
    }

    1 reference
    private void cb_delete_us_Click(object sender, EventArgs e)
    {
        Grant_Revoke_Table_User(cb_delete_us, "Delete");
    }

```

Hình 99

- Sự kiện click các CheckBox gán quyền (Select, Insert, Update, Delete) Table cho Role

```

1 reference
private void cb_select_ro_Click(object sender, EventArgs e)
{
    Grant_Revoke_Table_Role(cb_select_ro, "Select");
}

1 reference
private void cb_insert_ro_Click(object sender, EventArgs e)
{
    Grant_Revoke_Table_Role(cb_insert_ro, "Insert");
}

1 reference
private void cb_update_ro_Click(object sender, EventArgs e)
{
    Grant_Revoke_Table_Role(cb_update_ro, "Update");
}

1 reference
private void cb_delete_ro_Click(object sender, EventArgs e)
{
    Grant_Revoke_Table_Role(cb_delete_ro, "Delete");
}

```

Hình 100

- Sự kiện click nút gán/hủy gán Role cho User

```

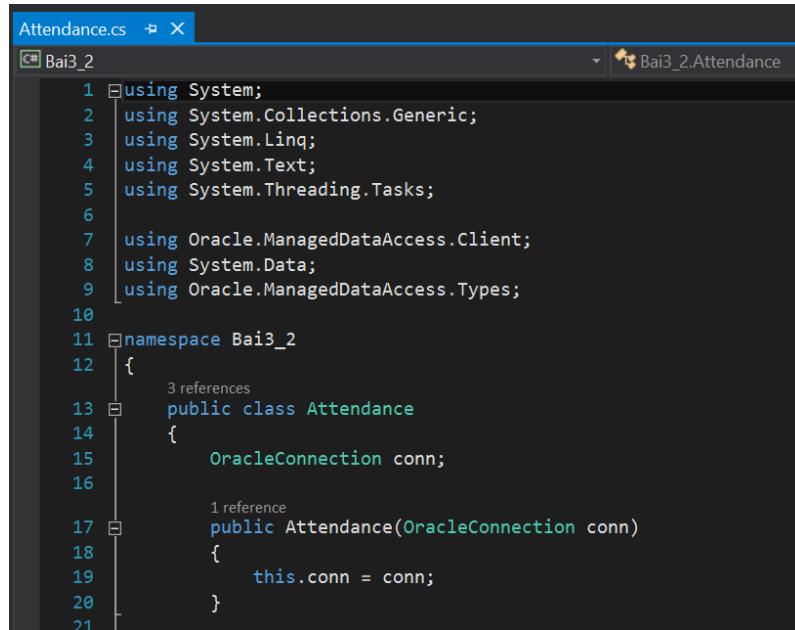
1 reference
private void btn_Grant_Revoke_Role_Click(object sender, EventArgs e)
{
    string user = cmb_username.SelectedItem.ToString();
    string role = cmb_roles.SelectedItem.ToString();
    int dk;
    if (btn_Grant_Revoke_Role.Text.Equals("Grant"))
    {
        dk = 1;
    }
    else
    {
        dk = 0;
    }

    if (Grant_Revoke_Role(user, role, dk))
    {
        Set_Text_Buttun(user, role);
        Load_Roles_User(user);
    }
}

```

Hình 101

- Tạo Class Attendance thực hiện gọi chạy các thủ tục liên quan đến bảng Attendance



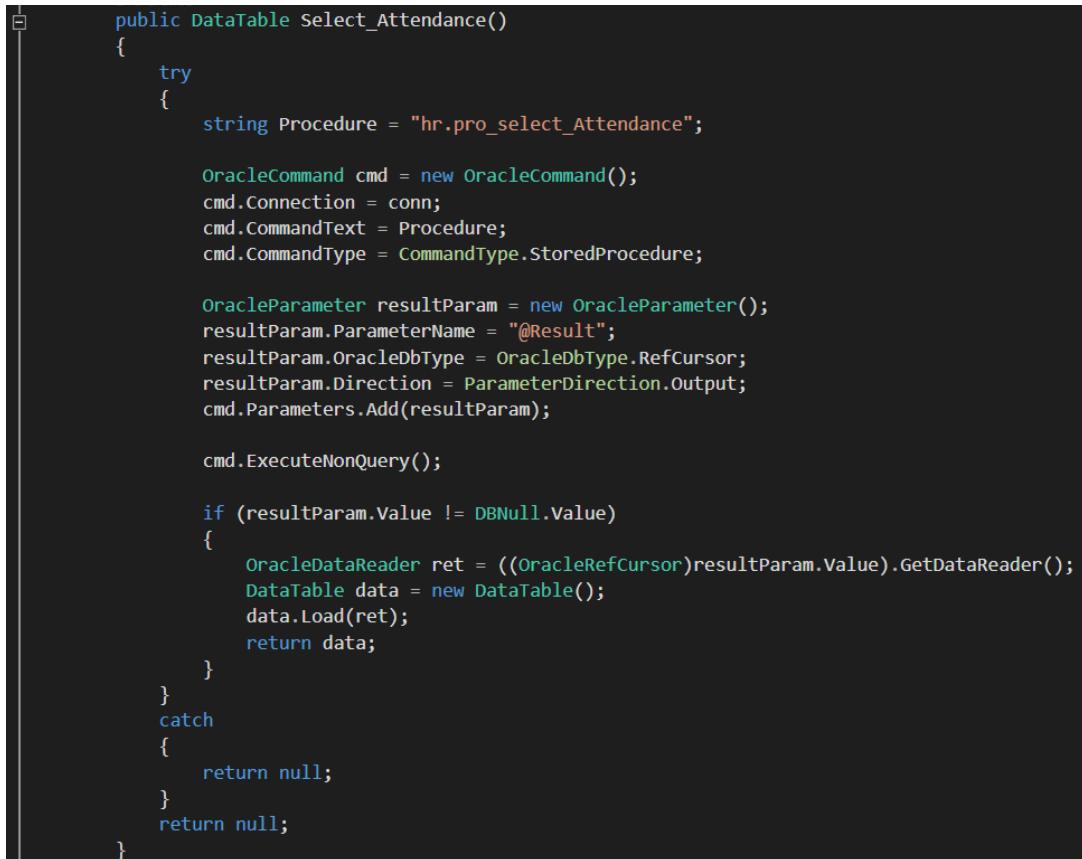
```

Attendance.cs  ✘ X
C# Bai3_2
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  using Oracle.ManagedDataAccess.Client;
8  using System.Data;
9  using Oracle.ManagedDataAccess.Types;
10
11 namespace Bai3_2
12 {
13     public class Attendance
14     {
15         OracleConnection conn;
16
17         public Attendance(OracleConnection conn)
18         {
19             this.conn = conn;
20         }
21     }

```

Hình 102

- Hàm Select



```

public DataTable Select_Attendance()
{
    try
    {
        string Procedure = "hr.pro_select_Attendance";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@Result";
        resultParam.OracleDbType = OracleDbType.RefCursor;
        resultParam.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(resultParam);

        cmd.ExecuteNonQuery();

        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();
            DataTable data = new DataTable();
            data.Load(ret);
            return data;
        }
    }
    catch
    {
        return null;
    }
    return null;
}

```

Hình 103

- Hàm Insert

```
public bool Insert_Attendance(int ID, string Name)
{
    try
    {
        string Procedure = "hr.pro_insert_Attendance";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter Id = new OracleParameter();
        Id.ParameterName = "@ID";
        Id.OracleDbType = OracleDbType.Int16;
        Id.Value = ID;
        Id.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(Id);

        OracleParameter name = new OracleParameter();
        name.ParameterName = "@Name";
        name.OracleDbType = OracleDbType.Varchar2;
        name.Value = Name;
        name.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(name);

        cmd.ExecuteNonQuery();

        return true;
    }
    catch
    {
        return false;
    }
}
```

Hình 104

- Hàm Update

```
    public bool Update_Attendance(int ID, string Name)
    {
        try
        {
            string Procedure = "hr.pro_update_Attendance";

            OracleCommand cmd = new OracleCommand();
            cmd.Connection = conn;
            cmd.CommandText = Procedure;
            cmd.CommandType = CommandType.StoredProcedure;

            OracleParameter Id = new OracleParameter();
            Id.ParameterName = "@ID";
            Id.OracleDbType = OracleDbType.Int16;
            Id.Value = ID;
            Id.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(Id);

            OracleParameter name = new OracleParameter();
            name.ParameterName = "@Name";
            name.OracleDbType = OracleDbType.Varchar2;
            name.Value = Name;
            name.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(name);

            cmd.ExecuteNonQuery();

            return true;
        }
        catch
        {
            return false;
        }
    }
```

Hình 105

- Hàm Delete

```
public bool Delete_Attendance(int ID)
{
    try
    {
        string Procedure = "hr.pro_delete_Attendance";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

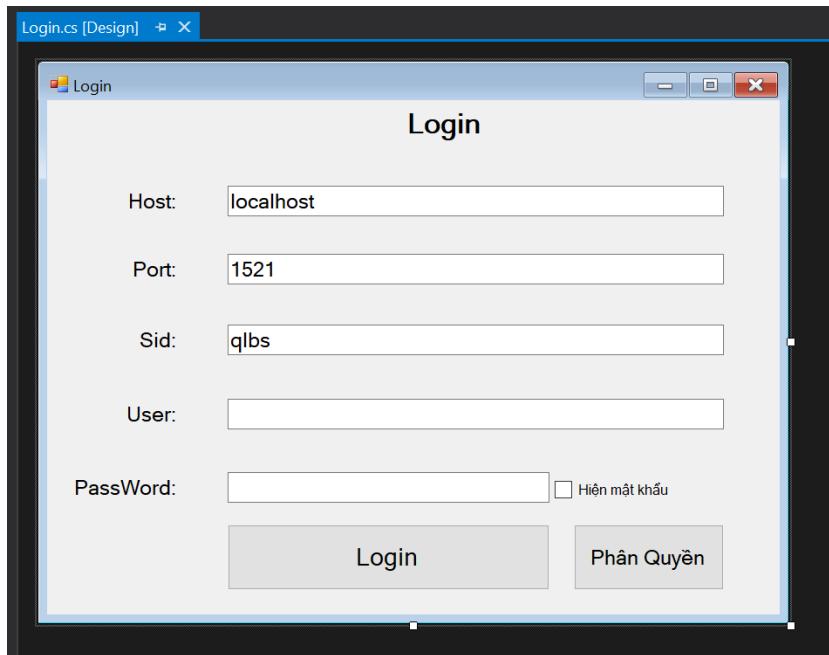
        OracleParameter Id = new OracleParameter();
        Id.ParameterName = "@ID";
        Id.OracleDbType = OracleDbType.Int16;
        Id.Value = ID;
        Id.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(Id);

        cmd.ExecuteNonQuery();

        return true;
    }
    catch
    {
        return false;
    }
}
```

Hình 106

- Lần lượt kiểm tra kết quả phân quyền đã cấp cho các role



Hình 107

- Sử dụng lại form Login ở Bài2_2, Sửa lại nút Login như (hình 108)

```

    private void btn_login_Click(object sender, EventArgs e)
    {
        string host = txt_host.Text;
        string port = txt_port.Text;
        string sid = txt_sid.Text;
        string user = txt_user.Text;
        string pass = txt_password.Text;

        if (Check_Textbox(host, port, sid, user, pass))
        {
            Database.Set_Database(host, port, sid, user, pass);
            if (Database.Connect())
            {
                MessageBox.Show("Đăng nhập thành công");
                new Attendance_GUI().Show();
                this.Hide();
            }
            else
            {
                Check_Status(user);
                return;
            }
        }
    }
}

```

Hình 108

- Cài đặt nút Phân quyền

```

    1 reference
    private void btn_pq_Click(object sender, EventArgs e)
    {
        Database.Set_Database(txt_host.Text, txt_port.Text, txt_sid.Text, "", "");
        if (Database.ConnectSys())
        {
            new PhanQuyenGUI().Show();
        }
    }
}

```

Hình 109

- Cài đặt CheckBox ẩn/hiện mật khẩu

```

    1 reference
    private void cb_pass_CheckedChanged(object sender, EventArgs e)
    {
        txt_password.UseSystemPasswordChar = !cb_pass.Checked;
    }
}

```

Hình 110

- Cài đặt ẩn mật khẩu khi chạy Form

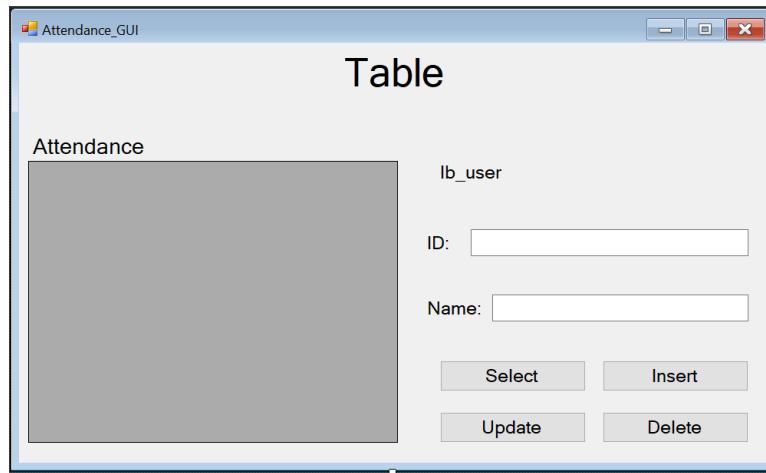
```

    1 reference
    public Login()
    {
        InitializeComponent();
        CenterToScreen();
        txt_password.UseSystemPasswordChar = !cb_pass.Checked;
    }
}

```

Hình 111

- Tạo form Table để kiểm tra các thao tác của user trên bảng



Hình 112

- Khai báo thư viện và các biến cần thiết

```

Attendance_GUI.cs  ✎ X Attendance_GUI.cs [Design]
[Bai3_2] Bai3_2.Attendance_GUI.cs
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 using Oracle.ManagedDataAccess.Client;
12
13 namespace Bai3_2
14 {
15     public partial class Attendance_GUI : Form
16     {
17         OracleConnection conn;
18         Attendance tb;
19         public Attendance_GUI()
20         {
21             InitializeComponent();
22             CenterToScreen();
23             conn = Database.Get_Connect();
24             tb = new Attendance(conn);
25             lb_user.Text = Database.User;
26         }
27     }

```

Hình 113

```
2 references
bool Check_TextBox()
{
    if (txt_id.Text.Equals(""))
    {
        MessageBox.Show("Bạn chưa điền ID");
        txt_id.Focus();
        return false;
    }

    if (txt_name.Text.Equals(""))
    {
        MessageBox.Show("Bạn chưa điền Name");
        txt_name.Focus();
        return false;
    }

    return true;
}
```

Hình 114

- Cài đặt nút Select

```
1 reference
private void btn_select_Click(object sender, EventArgs e)
{
    DataTable dt = tb.Select_Attendance();

    if (dt != null)
    {
        dtg_Attendance.DataSource = dt;
    }
    else
    {
        MessageBox.Show("Bạn không có quyền xem");
        return;
    }
}
```

Hình 115

- Cài đặt nút Insert

```
1 reference
private void btn_insert_Click(object sender, EventArgs e)
{
    if (Check_TextBox())
    {
        if (tb.Insert_Attendance(Convert.ToInt16(txt_id.Text), txt_name.Text))
        {
            MessageBox.Show("Thêm thành công");
        }
        else
        {
            MessageBox.Show("Thêm thất bại");
        }
    }
}
```

Hình 116

- Cài đặt nút Update

```
1 reference
private void btn_update_Click(object sender, EventArgs e)
{
    if (Check_TextBox())
    {
        if (tb.Update_Attendance(Convert.ToInt16(txt_id.Text), txt_name.Text))
        {
            MessageBox.Show("Sửa thành công");
        }
        else
        {
            MessageBox.Show("Sửa thất bại");
        }
    }
}
```

Hình 117

- Cài đặt nút Delete

```
1 reference
private void btn_delete_Click(object sender, EventArgs e)
{
    if (tb.Delete_Attendance(Convert.ToInt16(txt_id.Text)))
    {
        MessageBox.Show("Xóa thành công");
    }
    else
    {
        MessageBox.Show("Xóa thất bại");
    }
}
```

Hình 118

- Sự kiện đóng form và chọn bảng

```

    1 reference
    private void Attendance_GUI_FormClosing(object sender, FormClosingEventArgs e)
    {
        Application.Exit();
    }

    1 reference
    private void dtg_Attendance_SelectionChanged(object sender, EventArgs e)
    {
        int r = dtg_Attendance.CurrentCell.RowIndex;

        txt_id.Text = dtg_Attendance.Rows[r].Cells[0].Value.ToString();
        txt_name.Text = dtg_Attendance.Rows[r].Cells[1].Value.ToString();
    }

```

Hình 119

- Kiểm tra với các user thuộc role DataEntry: demo user John

Select

	ID	NAME
▶	1	John
*	2	Joe

john

ID: 1

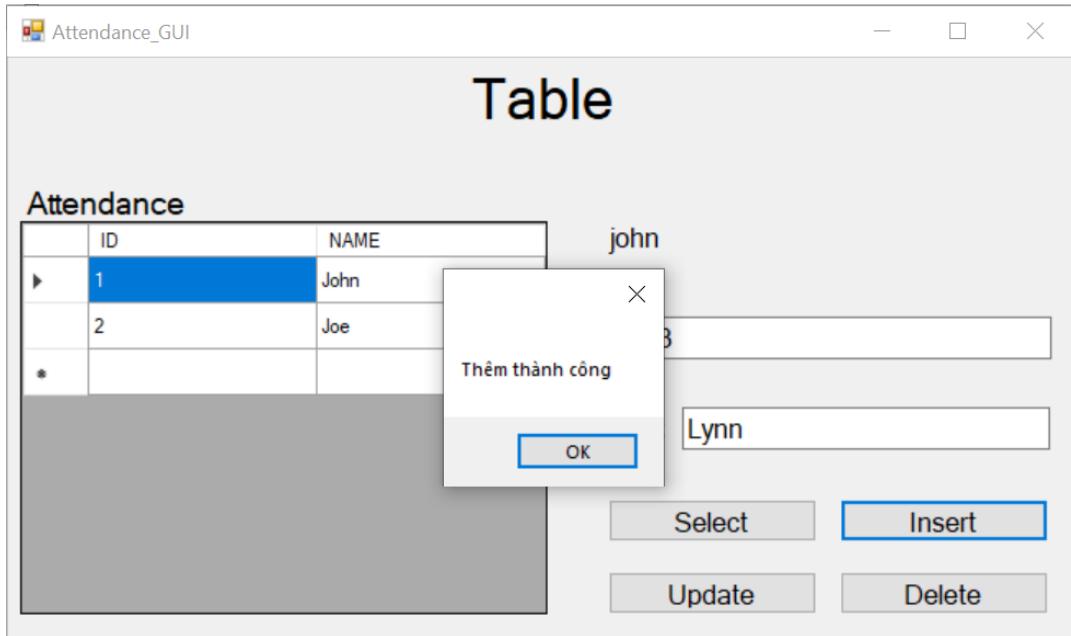
Name: John

Select Insert

Update Delete

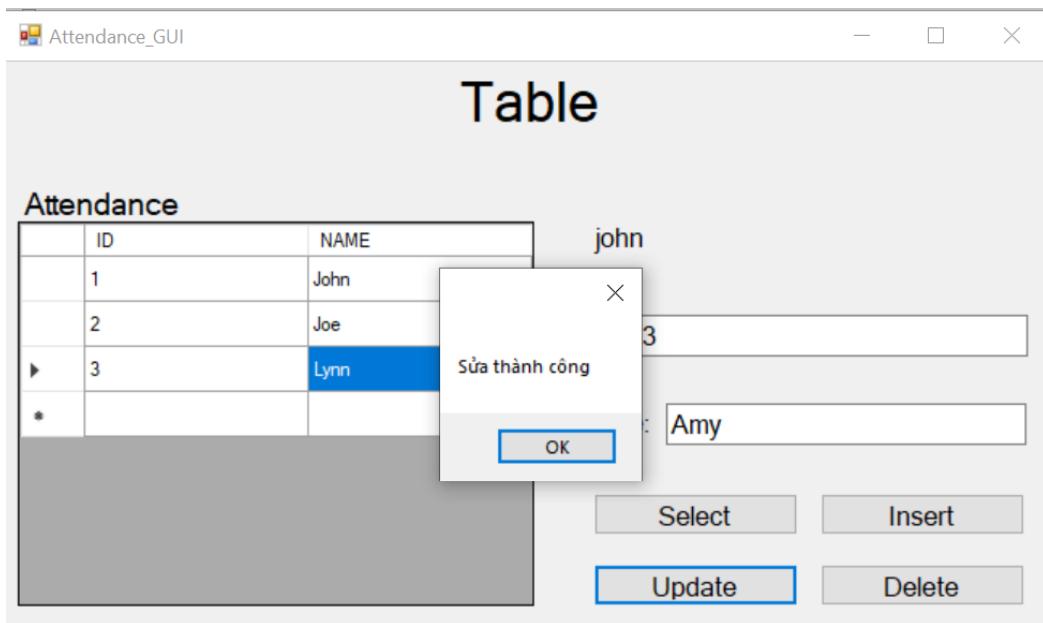
Hình 120

- Insert



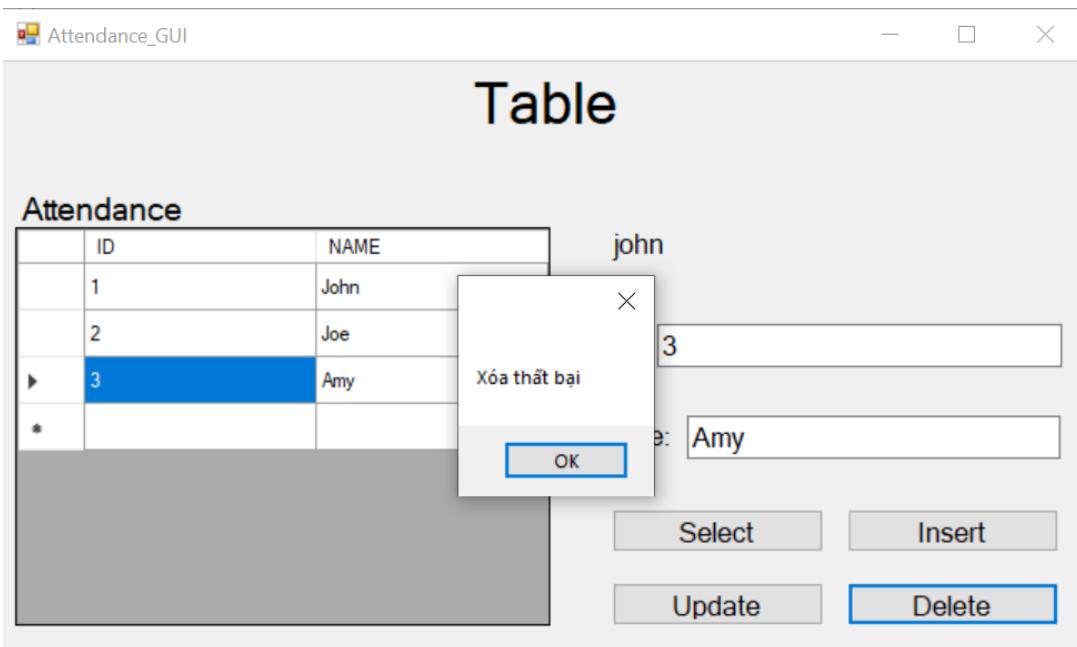
Hình 121

- Update



Hình 122

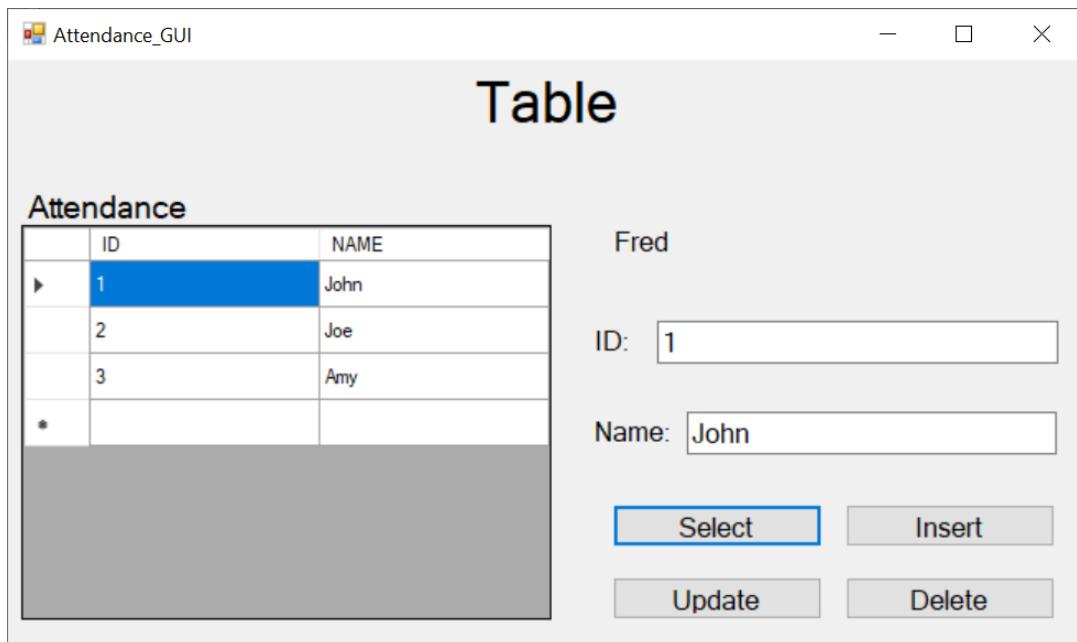
- Delete



Hình 123

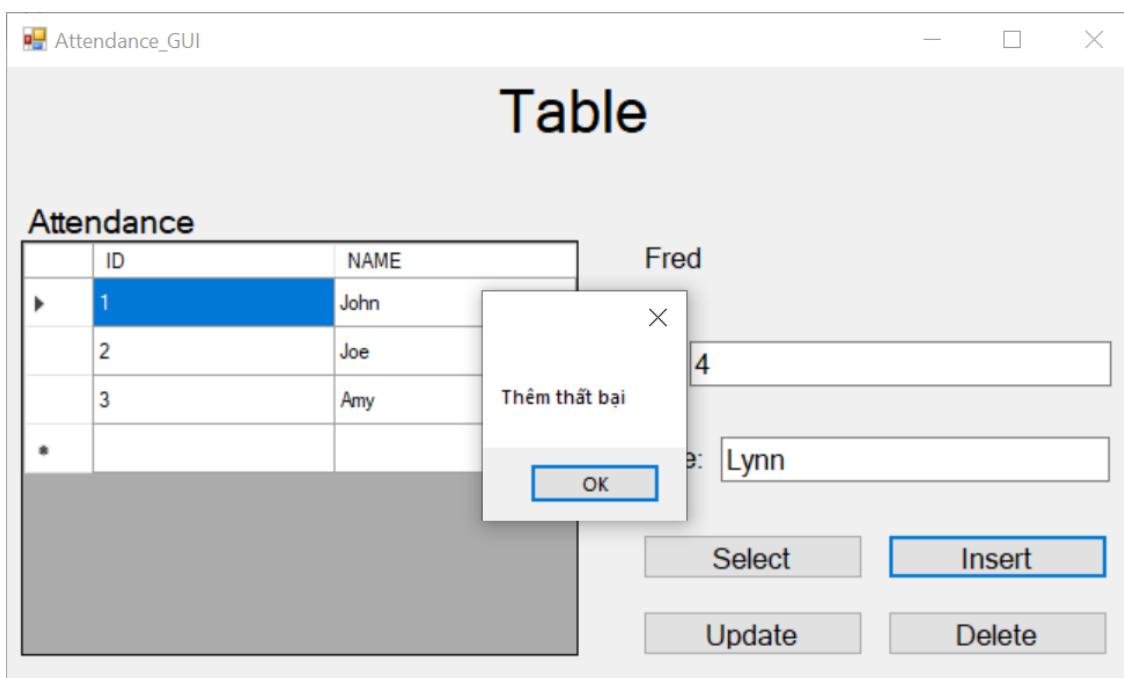
- Kiểm tra với các user thuộc role Supervisor: demo user Fred

Select



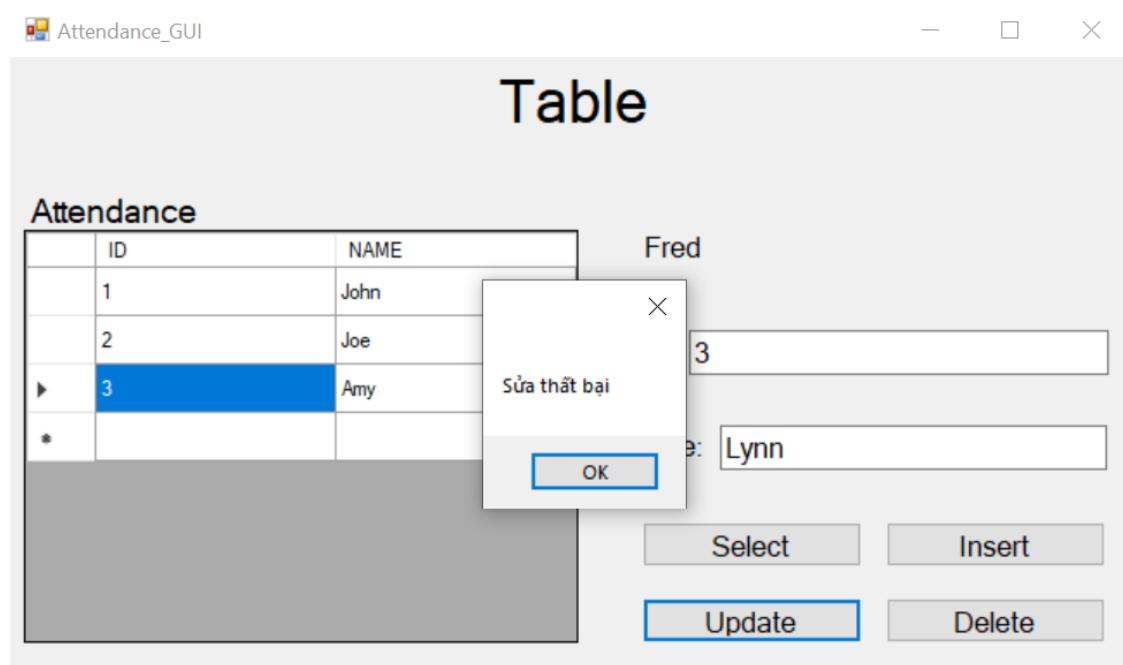
Hình 124

- Insert



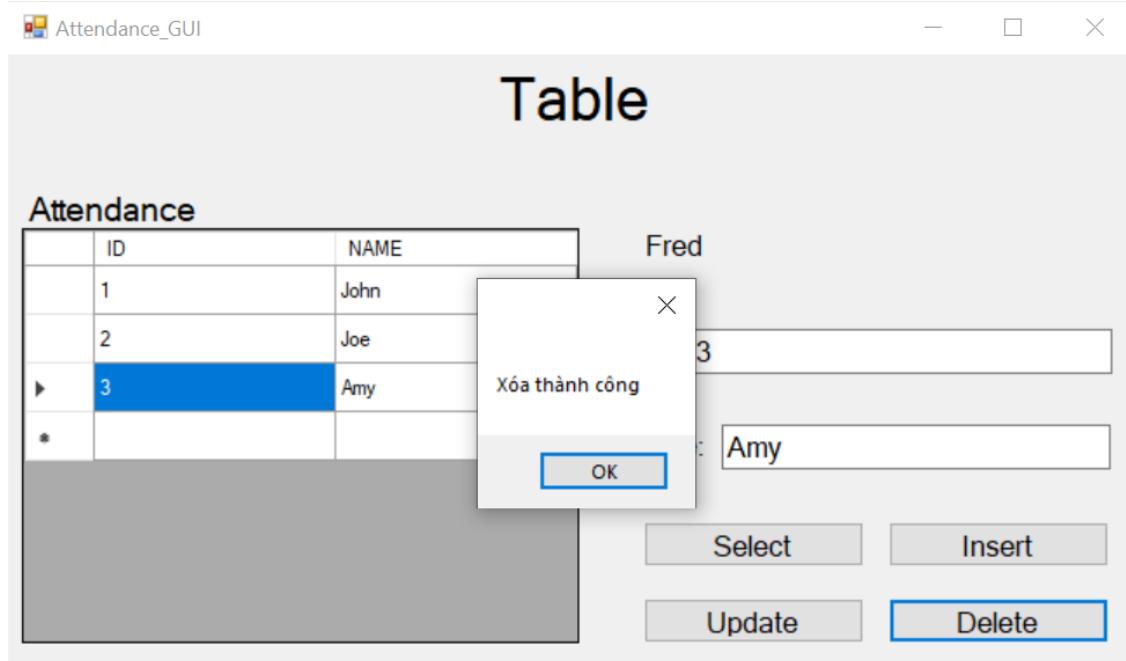
Hình 125

- Update



Hình 126

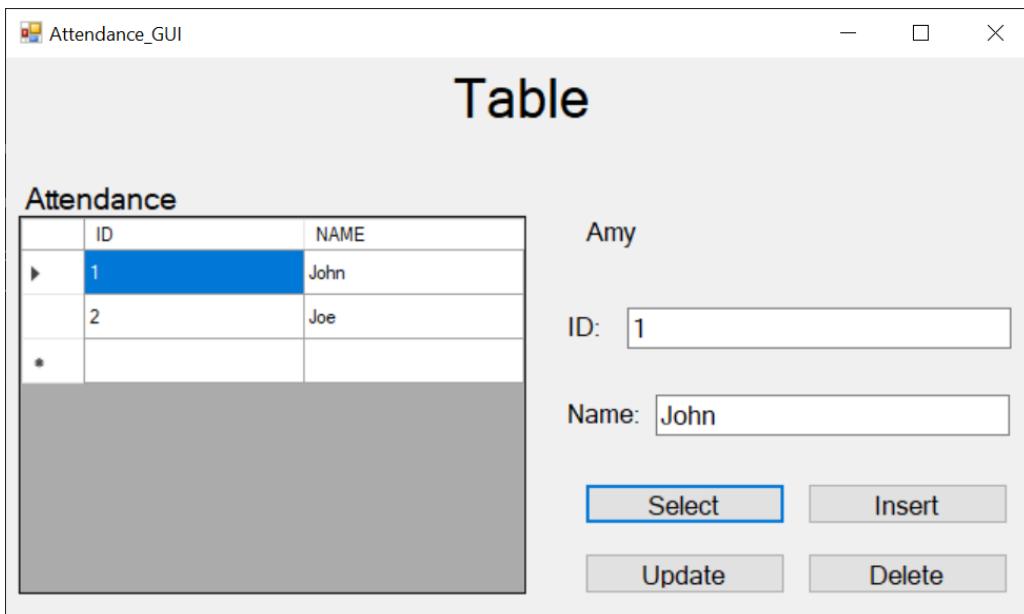
- Delete



Hình 127

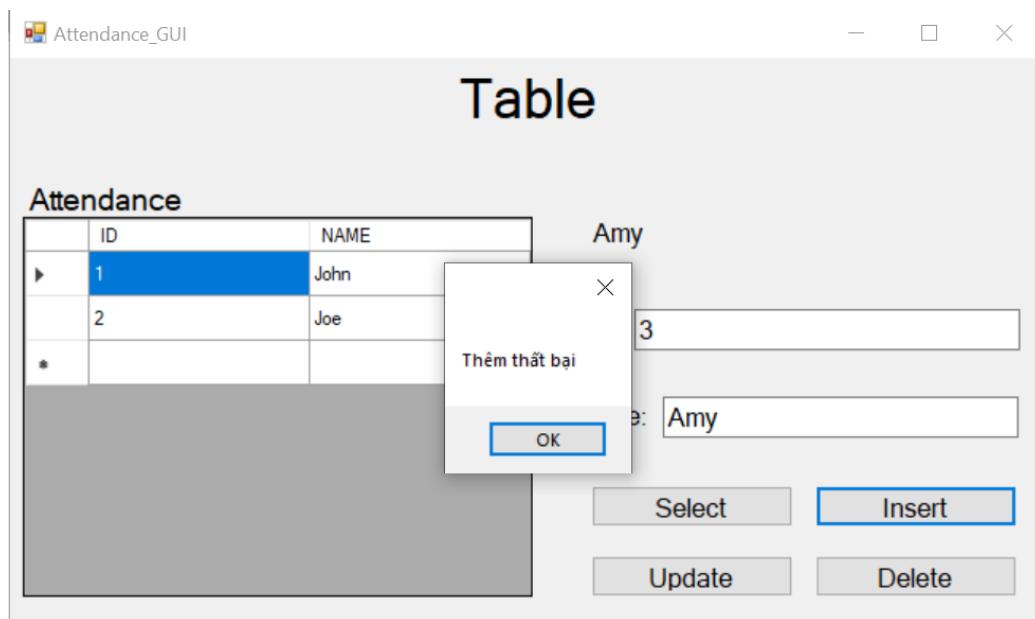
- Kiểm tra với các user thuộc role Management: demo user Amy

Select



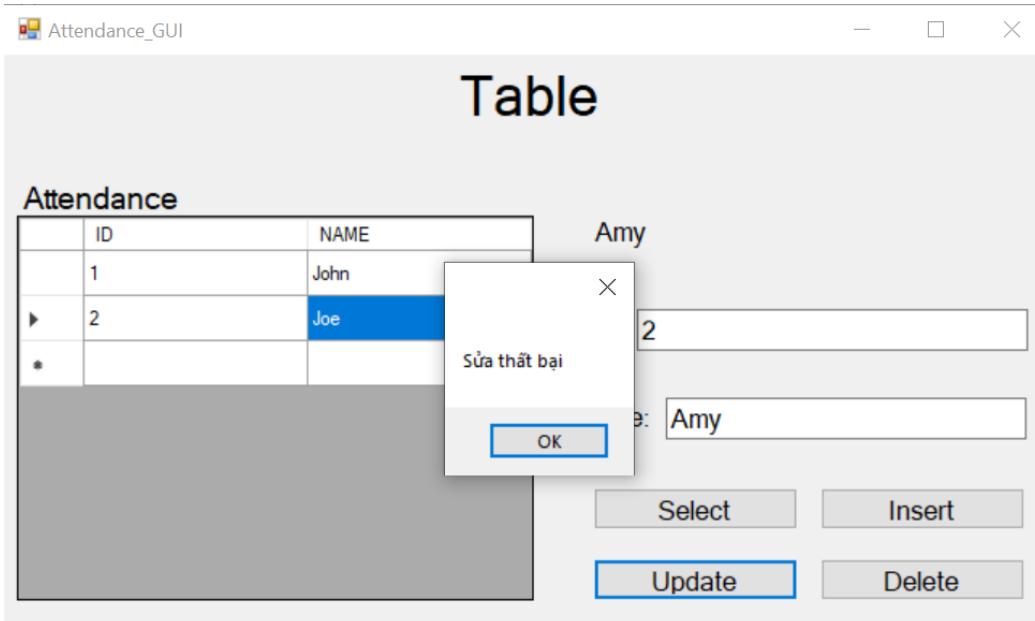
Hình 128

- Insert



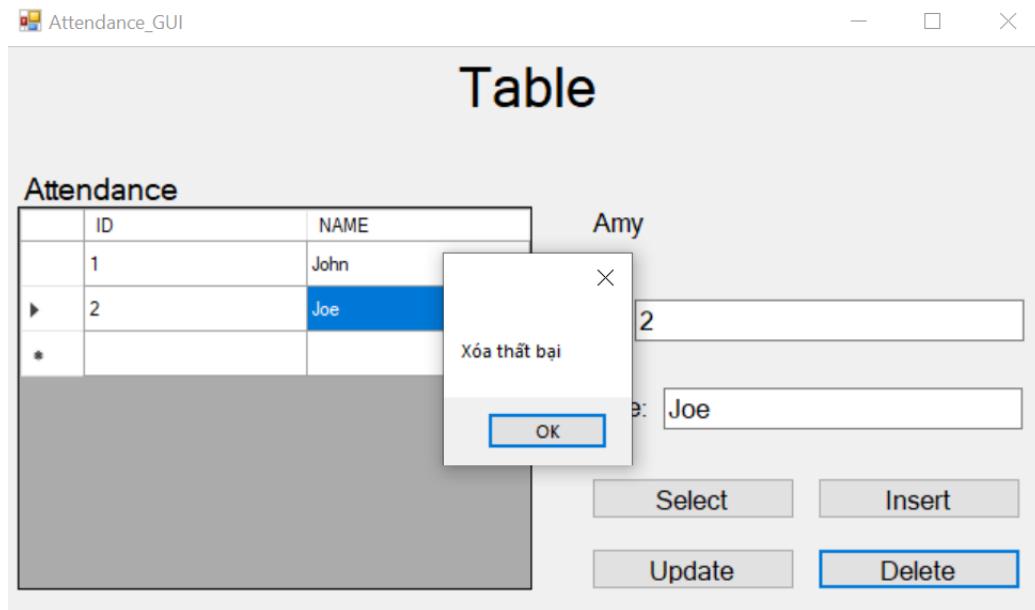
Hình 129

- Update



Hình 130

- Delete



Hình 131

3.3. Bài tập 3

3.3.1. Yêu cầu

Tạo một user mới tên NameManager với password là pc123. Gán quyền update cho user này trên cột Name của bảng Attendance.

3.3.2. Hướng dẫn

```

1 --connect SYS--
2 create user NameManager identified by pc123;
3 grant create session to NameManager;

User NAMEMANAGER created.

Grant succeeded.

5 --connect SYS--
6 grant update (Name) on HR.attendance to NameManager;

Grant succeeded.

```

Hình 132

- Kiểm tra trường hợp chỉ cập nhật cột Name

```

8 --connect NameManager/p123--
9 update hr.Attendance set name = 'Attendance' where id = 1;

```

Script Output | Query Result | Task completed in 0.029 seconds

1 row updated.

Hình 133

- Kiểm tra trường hợp cập nhật cả cột ID và Name

```

8 --connect NameManager/p123--
9 update hr.Attendance set id = 8, name = 'John' where id = 1;

```

Script Output | Query Result | Task completed in 0.031 seconds

Error starting at line : 9 in command -
update hr.Attendance set id = 8, name = 'John' where id = 1
Error at Command Line : 9 Column : 11
Error report -
SQL Error: ORA-01031: insufficient privileges
01031. 00000 - "insufficient privileges"
*Cause: An attempt was made to perform a database operation without
the necessary privileges.
*Action: Ask your database administrator or designated security
administrator to grant you the necessary privileges

Hình 134

3.4. Bài tập 4

3.4.1. Yêu cầu

- Thực hiện các yêu cầu sau đối với các view được liệt kê ở phần 2 (Tự điền dữ liệu):
- Tìm quyền mà trong tên của quyền có chữ CONTEXT
- Liệt kê tất cả user có quyền SELECT ANY TABLE

3.4.2. Hướng dẫn

- Tìm quyền mà trong tên của quyền có chữ CONTEXT

```

1 --4.1--
2 --connect SYS--
3 select distinct privilege from dba_sys_privs where privilege like '%CONTEXT%';

```

Query Result | SQL | All Rows Fetched: 7 in 0.019 seconds

PRIVILEGE
CREATE ANY EVALUATION CONTEXT
EXECUTE ANY EVALUATION CONTEXT
DROP ANY CONTEXT
CREATE EVALUATION CONTEXT
DROP ANY EVALUATION CONTEXT
CREATE ANY CONTEXT
ALTER ANY EVALUATION CONTEXT

Hình 135

- Liệt kê tất cả user có quyền SELECT ANY TABLE

```

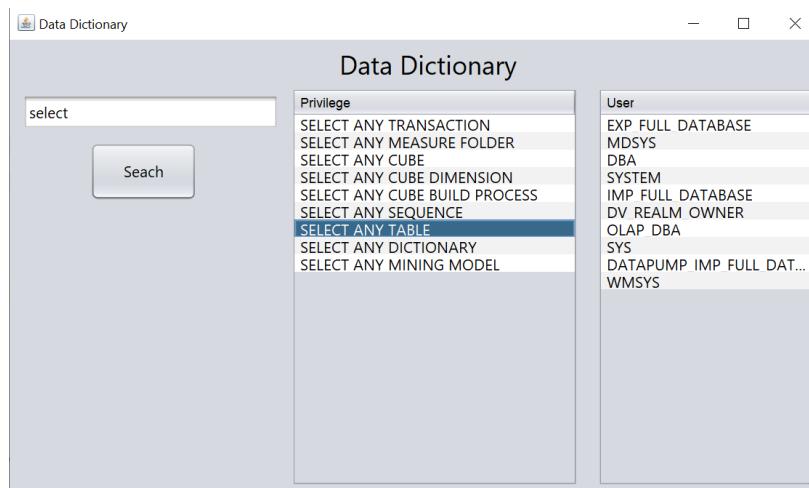
5 --4.2--
6 --connect SYS--
7 select grantee from dba_sys_privs where privilege = 'SELECT ANY TABLE';
8

```

GRANTEE
1 EXP_FULL_DATABASE
2 MDSYS
3 DBA
4 SYSTEM
5 IMP_FULL_DATABASE
6 DV_REALM_OWNER
7 OLAP_DBA
8 SYS
9 DATAPUMP_IMP_FULL_DATABASE
10 WMSYS

Hình 136

- Tạo form tìm kiếm quyền hệ thống, sử dụng lại form đăng nhập ở bài 2



Hình 137

- o Cần 2 thủ tục để tìm kiếm theo tên quyền và tìm kiếm user theo tên quyền

```

9 --thu thuc tim kiem theo tu dien du lieu dba_sys_privs tren--
10 --connect SYS--
11 create or replace procedure pro_seach_privilege(cur out sys_refcursor, seach in varchar2)
12 is
13 begin
14   open cur for
15     q'[select distinct privilege from dba_sys_privs where privilege like '%' || seach || q'%']");
16 end;

```

Hình 138

```

18 create or replace procedure pro_privilege_user(cur out sys_refcursor, privil in varchar2)
19 is
20 begin
21     open cur for
22         select grantee from dba_sys_privs where privilege = privil;
23 end;

```

The screenshot shows the SQL Developer interface with the 'Query Result' tab active. It displays the SQL code for creating the procedure 'pro_privilege_user'. Below the code, a message indicates 'Task completed in 0.052 seconds'.

Procedure PRO_PRIVILEGE_USER compiled

Hình 139

- Chạy thủ tục trên sql developer

```

26 --connect SYS--
27 set SERVEROUTPUT ON;
28 declare
29     cur sys_refcursor;
30     s varchar2(50) := 'SELECT';
31     c varchar2(100);
32 begin
33     pro_seach_privilege(cur, s);
34
35 loop
36     fetch cur into c;
37     exit when cur%notfound;
38     dbms_output.put_line('privilege: ' || c);
39 end loop;
40 end;

```

The screenshot shows the SQL Developer interface with the 'Script Output' tab active. It displays the output of the procedure execution, listing various privileges: SELECT ANY TRANSACTION, SELECT ANY MEASURE FOLDER, SELECT ANY CUBE, SELECT ANY CUBE DIMENSION, SELECT ANY CUBE BUILD PROCESS, SELECT ANY SEQUENCE, SELECT ANY TABLE, SELECT ANY MINING MODEL, and SELECT ANY DICTIONARY. Below the output, a message indicates 'PL/SQL procedure successfully completed.'

privilege: SELECT ANY TRANSACTION

privilege: SELECT ANY MEASURE FOLDER

privilege: SELECT ANY CUBE

privilege: SELECT ANY CUBE DIMENSION

privilege: SELECT ANY CUBE BUILD PROCESS

privilege: SELECT ANY SEQUENCE

privilege: SELECT ANY TABLE

privilege: SELECT ANY MINING MODEL

privilege: SELECT ANY DICTIONARY

PL/SQL procedure successfully completed.

Hình 140

```

42|declare
43|    cur sys_refcursor;
44|    s varchar2(50) := 'SELECT ANY TABLE';
45|    c varchar2(100);
46|begin
47|    pro_privilege_user(cur, s);
48|
49|    loop
50|        fetch cur into c;
51|        exit when cur%notfound;
52|        dbms_output.put_line('User: ' || c);
53|    end loop;
54|end;

```

Script Output X | Task completed in 0.074 seconds

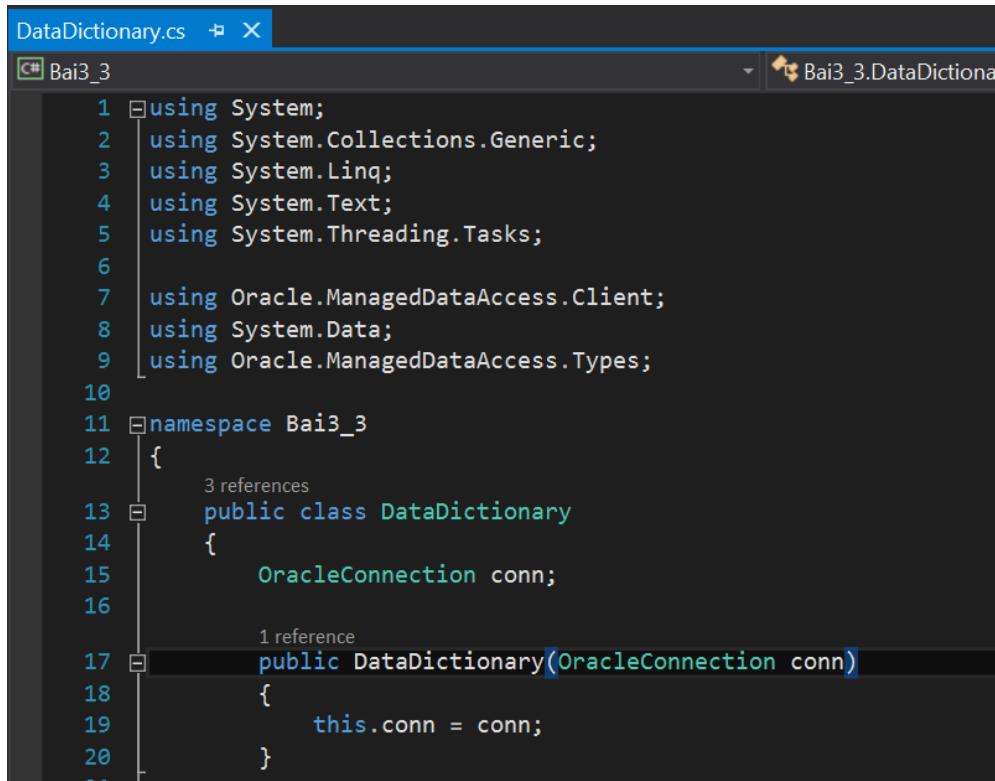
```

User: SEC_MGR
User: EXP_FULL_DATABASE
User: MDSYS
User: DBA
User: SYSTEM
User: IMP_FULL_DATABASE
User: DV_REALM_OWNER
User: OLAP_DBA
User: SYS
User: DATAPUMP_IMP_FULL_DATABASE
User: WMSYS

```

Hình 141

- Tạo Class DataDictionary truy vấn dữ liệu từ bảng



```

DataDictionary.cs ✘ X
Bai3_3
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 using Oracle.ManagedDataAccess.Client;
8 using System.Data;
9 using Oracle.ManagedDataAccess.Types;
10
11 namespace Bai3_3
12 {
13     public class DataDictionary
14     {
15         OracleConnection conn;
16
17         public DataDictionary(OracleConnection conn)
18         {
19             this.conn = conn;
20         }
21     }
}

```

Hình 142

- Hàm tìm kiếm tên quyền

```
1 reference
public DataTable Get_Privilege(string text)
{
    try
    {
        string Procedure = "pro_seach_privilege";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@Result";
        resultParam.OracleDbType = OracleDbType.RefCursor;
        resultParam.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(resultParam);

        OracleParameter Seach = new OracleParameter();
        Seach.ParameterName = "@seach";
        Seach.OracleDbType = OracleDbType.Varchar2;
        Seach.Value = text;
        Seach.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(Seach);

        cmd.ExecuteNonQuery();

        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();
            DataTable data = new DataTable();
            data.Load(ret);
            return data;
        }
    }
    catch
    {
        return null;
    }
    return null;
}
```

Hình 143

- Hàm tìm kiếm user có quyền ở bảng Privilege

```
1 reference
public DataTable Get_User(string text)
{
    try
    {
        string Procedure = "pro_privilege_user";

        OracleCommand cmd = new OracleCommand();
        cmd.Connection = conn;
        cmd.CommandText = Procedure;
        cmd.CommandType = CommandType.StoredProcedure;

        OracleParameter resultParam = new OracleParameter();
        resultParam.ParameterName = "@Result";
        resultParam.OracleDbType = OracleDbType.RefCursor;
        resultParam.Direction = ParameterDirection.Output;
        cmd.Parameters.Add(resultParam);

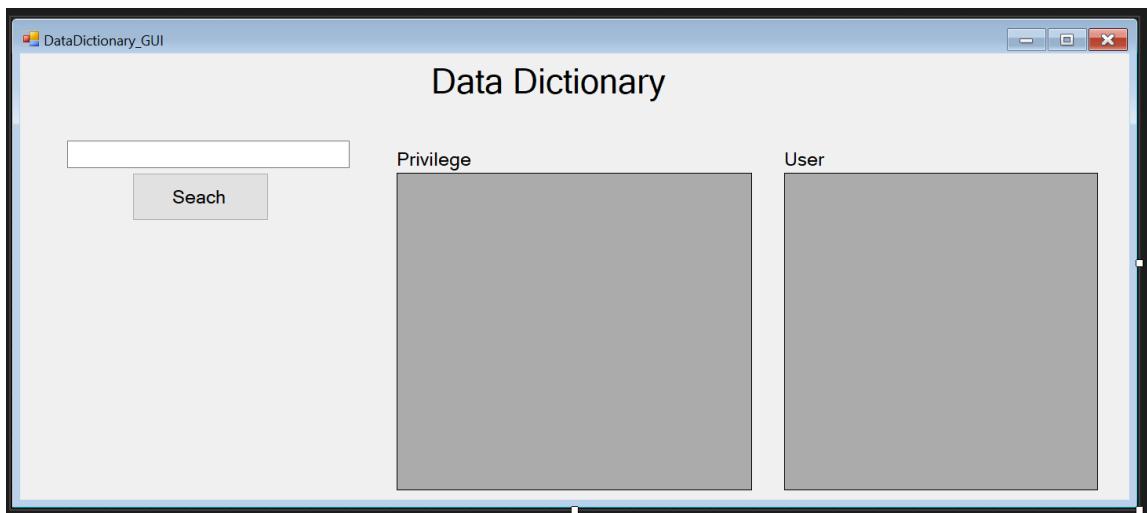
        OracleParameter User = new OracleParameter();
        User.ParameterName = "@user";
        User.OracleDbType = OracleDbType.Varchar2;
        User.Value = text;
        User.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(User);

        cmd.ExecuteNonQuery();

        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();
            DataTable data = new DataTable();
            data.Load(ret);
            return data;
        }
    }
    catch
    {
        return null;
    }
    return null;
}
```

Hình 144

- Gọi thủ tục trong C#



Hình 145

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using Oracle.ManagedDataAccess.Client;

namespace Bai3_3
{
    3 references
    public partial class DataDictionary_GUI : Form
    {
        OracleConnection conn;
        DataDictionary dt;

        1 reference
        public DataDictionary_GUI()
        {
            InitializeComponent();
            CenterToScreen();
            conn = Database.Get_Connect();
            dt = new DataDictionary(conn);
        }
    }
}

```

Hình 146

o Nút Seach

```
1 reference
private void btn_seach_Click(object sender, EventArgs e)
{
    DataTable tb = dt.Get_Privilege(txt_seach.Text.ToUpper());
    if (tb != null)
    {
        dtg_privilege.DataSource = tb;
    }
}
```

Hình 147

o Sự kiện chọn bảng privilege

```
1 reference
private void dtg_privilege_SelectionChanged(object sender, EventArgs e)
{
    int r = dtg_privilege.CurrentCell.RowIndex;
    string privilege = dtg_privilege.Rows[r].Cells[0].Value.ToString();
    DataTable tb = dt.Get_User(privilege);
    if (tb != null)
    {
        dtg_user.DataSource = tb;
    }
}
```

Hình 148

o Sự kiện đóng form

```
1 reference
private void DataDictionary_GUI_FormClosing(object sender, FormClosingEventArgs e)
{
    Application.Exit();
}
```

Hình 149

3.5. Bài tập 5

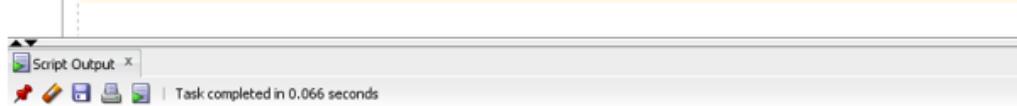
3.5.1. Yêu cầu

- Gán password cho role DataEntry ở bài 1 là “mgt”
- Cho phép user John quyền cấp quyền cho các user khác
- Gán tất cả các quyền mà John có cho Beth. Beth có quyền INSERT và UPDATE trên bảng Attendance không?

3.5.2. Hướng dẫn

- Gán password cho role DataEntry ở bài 1 là “mgt”

```
1 | --5.1--
2 | --connect SYS--
3 | alter role DataEntry identified by mgt;
```



Role DATAENTRY altered.

Hình 150

- Cho phép user John quyền cấp quyền cho các user khác

```
5 | --5.2--
6 | --connect SYS--
7 | grant DataEntry to John with admin option;
```

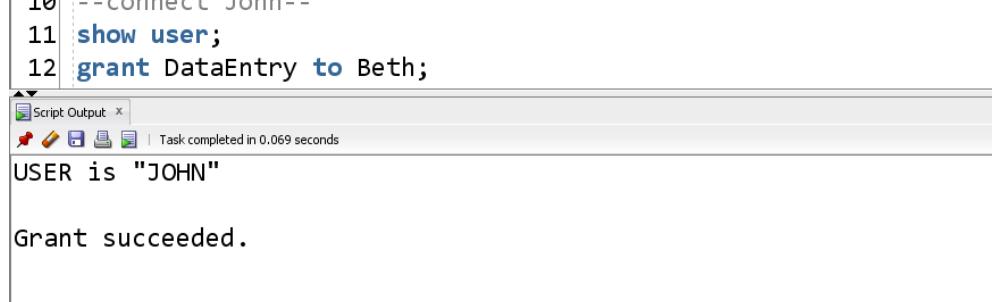


Hình 151

- Gán tất cả các quyền mà John có cho Beth. Beth có quyền INSERT và UPDATE trên bảng Attendance không?

- Tiến hành cấp quyền của John cho Beth

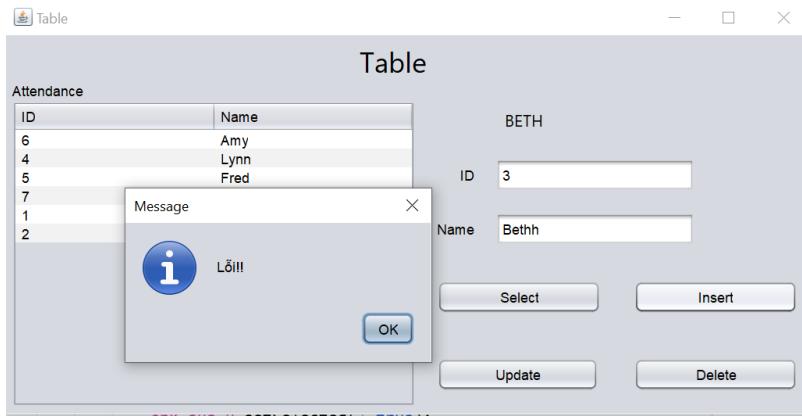
```
9 | --5.3--
10 | --connect John--
11 | show user;
12 | grant DataEntry to Beth;
```



Hình 152

- Dùng form ở bài tập 2 để kiểm tra:

Lỗi khi thêm dữ liệu



Hình 153

Lỗi sửa dữ liệu



Hình 154

Nguyên nhân do chế độ xác thực bằng mật khẩu của Role DataEntry đã được bật

3.6. Bài tập 6

3.6.1. Yêu cầu

Cho đoạn code sau:

CONN giaoVien/p123;

CREATE ROLE sinhVien;

GRANT select ON Attendance TO sinhVien WITH GRANT OPTION;

Đoạn code trên sẽ báo lỗi ở đâu và trong trường hợp nào?

3.6.2. Hướng dẫn

- Đoạn code trên sẽ báo lỗi ở dòng lệnh: GRANT select ON Attendance TO sinhvien WITH GRANT OPTION;
- Trong trường hợp gán quyền cho ROLE thì không được dùng WITH GRANT OPTION
- Lấy ví dụ minh họa ở Role DataEntry ở bài trước