

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM**





**Đồ án**

**MÔN:Hệ quản trị cơ sở dữ liệu Oracle**

**Đề tài: Quản lý bệnh viện**

**Giáo viên hướng dẫn: Nguyễn Phương Hạc**

*TP.Hồ Chí Minh, tháng 9 năm 2020*

🙢🕮🙠

**Lời mở đầu**

Con người hiện nay ngày càng nhiều bệnh tật sẽ làm khó cho các bác sĩ trong việc quản lý được bệnh nhân của mình mắc các loại bệnh gì. Một bệnh viện không quản lý được tất cả mọi việc trong bệnh viện sẽ gặp nhiều khó khăn trong việc điều trị bệnh cho bệnh nhân cũng như quản lý các nhân viên trong bệnh viện sẽ dẫn đến gặp nhiều rủi ro.

Nếu như các bệnh viện chỉ lưu thông tin của bệnh viện bằng các phương thức truyền thống thì có thể sẽ không còn hiệu quả vì có thể sẽ làm mất, bị làm giả thông tin, khiến nhiều kẻ gian đánh cắp tài liệu. Thế nên, giải pháp tối ưu để giải quyết các vấn đề trên là sử dụng phần mềm quản lý bệnh viện.

|  |  |
| --- | --- |
| **Điểm** | **Nhận xét** |
|  |  |

Danh sách thành viên trong nhóm:

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Họ Tên** | **Góp phần (%)** |
| 2001181383 | Vũ Minh Trung | 80% |
| 2001181280 | Nguyễn Thanh Quang | 20% |

1. **Mục tiêu đồ án:**

Một bệnh viện khi có quá nhiều bệnh nhân và bác sỹ, khi đó phần mềm hỗ trợ quản lý bệnh viện sẽ giúp các nhân viên bệnh viện kiểm soát được các ca bệnh, bệnh nhân, bác sĩ và người nhà bệnh nhân.

1. **Nội dung chi tiết đồ án:**

* Quản lý nhân viên bệnh viện giúp tra cứu và bổ sung các thông tin cần thiết của các nhân viên một cách nhanh chóng.
* Quản lý các khoa của bệnh viện giúp phân chia công việc và chuyên môn của từng khoa.
* Quản lý bệnh nhân giúp tra cứu và bổ sung các thông tin cá nhân của từng bệnh nhân, xếp loại mức độ nghiêm trọng của bệnh cho bệnh nhân.
* Quản lý xét nghiệm, chữa bệnh, vật tư y tế.
* Quản lý các trường hợp bệnh của bệnh nhân giúp các bác sĩ phân loại các mức độ nghiêm trọng cũng như đưa ra giải pháp tốt nhất cho bệnh nhân.
* Quản lý tình trạng cấp cứu giúp các nhân viên thu ngân phân loại các tình trạng bệnh của bệnh nhân để đưa ra các mức chi phí mà bệnh nhân phải trả.
* Quản lý trưởng khoa giúp quản lý thông tin của các bác sĩ trưởng khoa của từng khoa.
* Quản lý phân công giúp phân chia các ngày trực trong bệnh viện cho các bác sĩ, y tá, điều dưỡng một cách phù hợp.
* Đánh giá tình trạng của bệnh nhân cũng như cho biết số ngày bệnh nhân trong bệnh viện và ngày cuối cùng vào bệnh viện.

**Table EMPLOYEE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **EMPID** | **NUMBER(38,0)** | **NO** | **Primary\_key** |
| **FNAME** | **VARCHAR2(16 BYTE)** | **YES** |  |
| **LNAME** | **VARCHAR2(16 BYTE)** | **NO** |  |
| **GENDER** | **VARCHAR2(16 BYTE)** | **NO** |  |
| **EMPTYPE** | **VARCHAR2(30 BYTE)** | **NO** | **Foreign\_Key** |
| **STREET** | **VARCHAR2(20 BYTE)** | **YES** |  |
| **CITY** | **VARCHAR2(6 BYTE)** | **YES** |  |
| **DATE\_OF\_JOINING** | **Date** | **YES** |  |
| **EMAIL** | **VARCHAR2(20 BYTE)** | **YES** |  |
| **DEPTID** | **NUMBER(38,0)** | **NO** | **Foreign\_Key** |
| **SINCE** | **Date** | **YES** |  |
| **DATE\_OF\_BIRTH** | **Date** | **YES** |  |

**Table department:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **DEPTID** | **NUMBER(38,0)** | **No** | **Primary\_key** |
| **DNAME** | **VARCHAR2(20 BYTE)** | **No** |  |
| **DEPT\_HEADID** | **NUMBER(38,0)** | **Yes** |  |

**Table Patient:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **BLOOD\_GROUP** | **VARCHAR2(30 BYTE)** | **NO** |  |
| **CITY** | **VARCHAR2(16 BYTE)** | **YES** |  |
| **DOB** | **DATE** | **NO** |  |
| **DOC\_ID** | **NUMBER(38,0)** |  | **Foreign\_Key** |
| **EMAIL** | **VARCHAR2(30 BYTE)** | **YES** |  |
| **FNAME** | **VARCHAR2(20 BYTE)** | **NO** |  |
| **GENDER** | **VARCHAR2(6 BYTE)** | **YES** |  |
| **LNAME** | **VARCHAR2(20 BYTE)** | **YES** |  |
| **PID** | **NUMBER(38,0)** | **NO** | **Primary\_Key** |
| **STREET** | **VARCHAR2(20 BYTE)** | **YES** |  |

Table Medicine:

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **MID** | **NUMBER(38,0)** | **No** | **Primary\_Key** |
| **MNAME** | **VARCHAR2(40 BYTE)** | **No** |  |
| **MCOST** | **NUMBER(20,2)** | **Yes** |  |

**Table Salary:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **ETYPE** | **VARCHAR2(20 BYTE)** | **No** | **Primary\_key** |
| **SALARY** | **NUMBER(20,2)** | **Yes** |  |

**Table Bill:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **PID** | **NUMBER(38,0)** | **No** | **Primary\_key** |
| **MCOST** | **NUMBER(20,2)** | **Yes** |  |
| **TCOST** | **NUMBER(20,2)** | **Yes** |  |
| **ROOMCHARGES** | **NUMBER(20,2)** | **Yes** |  |
| **OTHERCHARGES** | **NUMBER(20,2)** | **Yes** |  |
| **BILLDATE** | **DATE** | **No** | **Primary\_key** |

**Table IN\_EMPLOYEE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **EMPID** | **NUMBER(38,0)** | **NO** | **Primary\_key** |
| **FNAME** | **VARCHAR2(16 BYTE)** | **YES** |  |
| **LNAME** | **VARCHAR2(16 BYTE)** | **NO** |  |
| **GENDER** | **VARCHAR2(16 BYTE)** | **NO** |  |
| **EMPTYPE** | **VARCHAR2(30 BYTE)** | **NO** | **Foreign\_Key** |
| **STREET** | **VARCHAR2(20 BYTE)** | **YES** |  |
| **CITY** | **VARCHAR2(6 BYTE)** | **YES** |  |
| **DATE\_OF\_JOINING** | **Date** | **YES** |  |
| **EMAIL** | **VARCHAR2(20 BYTE)** | **YES** |  |
| **DEPTID** | **NUMBER(38,0)** | **NO** | **Foreign\_Key** |
| **SINCE** | **Date** | **YES** |  |
| **DATE\_OF\_BIRTH** | **Date** | **YES** |  |
| **DATE\_OF\_LEAVING** | **Date** | **YES** |  |

**Table Had\_medicine:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **PID** | **NUMBER(38,0)** | **No** | **Primary\_key** |
| **MID** | **NUMBER(38,0)** | **No** | **Primary\_key** |
| **MED\_DATE** | **DATE** | **No** | **Primary\_key** |
| **QTY** | **NUMBER(38,0)** | **Yes** |  |

**Table Had\_Test:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **PID** | **NUMBER(38,0)** | **No** | **Primary\_key** |
| **TID** | **NUMBER(38,0)** | **No** | **Primary\_key,**  **Foreign\_key** |
| **TESTDATE** | **DATE** | **No** |  |

**Table Test:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **TID** | **NUMBER(38,0)** | **No** | **Primary\_key** |
| **TNAME** | **VARCHAR2(20 BYTE)** | **No** |  |
| **TCOST** | **NUMBER(10,2)** | **yes** |  |

**Table Relative:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **PID** | **NUMBER(38,0)** | **No** | **Primary\_key,**  **Foreign\_key** |
| **RNAME** | **VARCHAR2(30**  **BYTE)** | **Yes** |  |
| **RTYPE** | **VARCHAR2(30**  **BYTE)** | **Yes** |  |

**Table NURSE\_ASSIGN:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **NID** | **NUMBER(38,0)** | **No** | **Primary\_key,**  **Foreign\_key** |
| **COUNTPATIENT** | **NUMBER(38,0)** | **Yes** |  |

**Table ROOM:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column\_name** | **DATA\_TYPE** | **NULLABLE** | **Constraint** |
| **RID** | **NUMBER(38,0)** | **No** | **Primary\_key** |
| **ROOMTYPE** | **VARCHAR2(20 BYTE)** | **Yes** |  |
| **RCOST** | **NUMBER(38,0)** | **Yes** |  |

**QUI ĐỊNH:**

* Phòng ban với trưởng phòng ban: 1 trưởng phòng ban chỉ quản lý duy nhất 1 phòng ban và ngược lại.-->quan hệ 1-1

1. **Các chức năng truy vấn:**

* Thêm, Xoá, Sửa, thông tin bệnh nhân.
* Phân quyền cho mỗi tài khoản.
* Làm mới lại thông tin mỗi khi nhập liệu sai.
* Tìm kiếm thông tin bệnh nhân và nhân viên bệnh viện.

1. **Danh mục nhóm quyền profile:**

\_Trong đây có 4 nhóm quyền cơ bản:

Quyền 1: Quyền quản trị hệ thống chỉ cho user sysdba

Quyền 2: Dành cho cho user có địa vị là giám đốc , có hầu hết gần như toàn bộ chức năng trừ các chức năng sysdba

Quyền3: Dành cho các user là nhân viên bình thường có chức năng thêm, xóa, xem thông tin bệnh nhân

Quyền 4: Nhóm quyền dành cho user người thăm, chỉ cho xem thông tin người bệnh của mình .

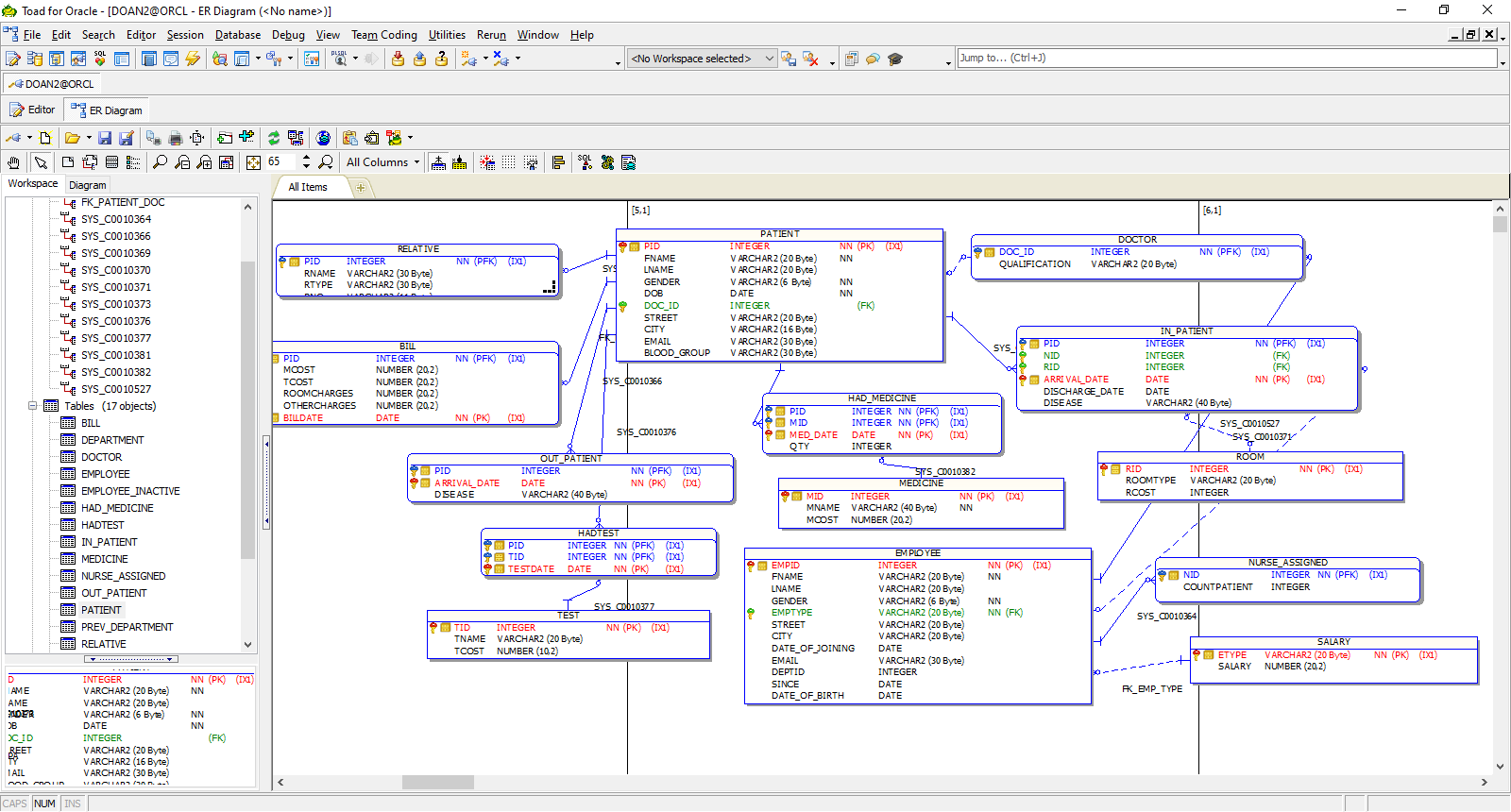
1. **Các giao tác được đặt trong đồ án:**

Các giao tác liên quan INSERT, UPDATE, DELETE.

Giao tác thêm tablespace(kiểm tra có trùng hay không) ,giao tác thêm nhân viên , giao tác xóa đi tablespace

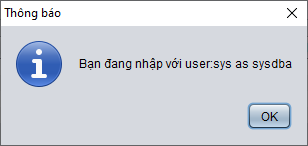
Giao tác đăng nhập user, Giao tác thêm việc cho nhân viên.

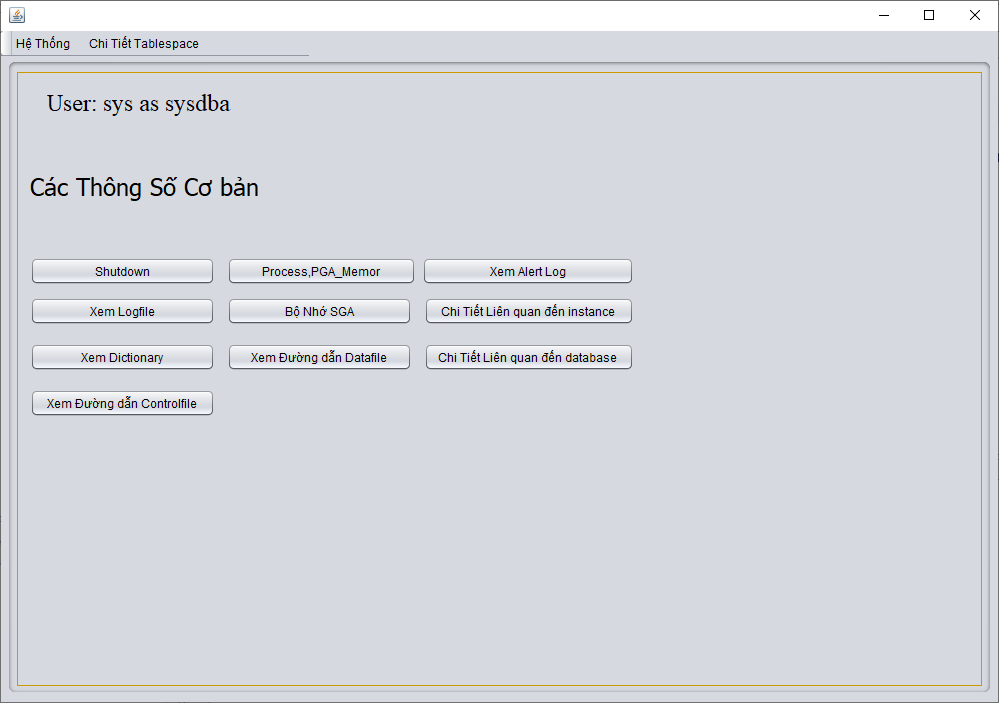
1. **Phân tích và thiết kế mô hình quan hệ:**



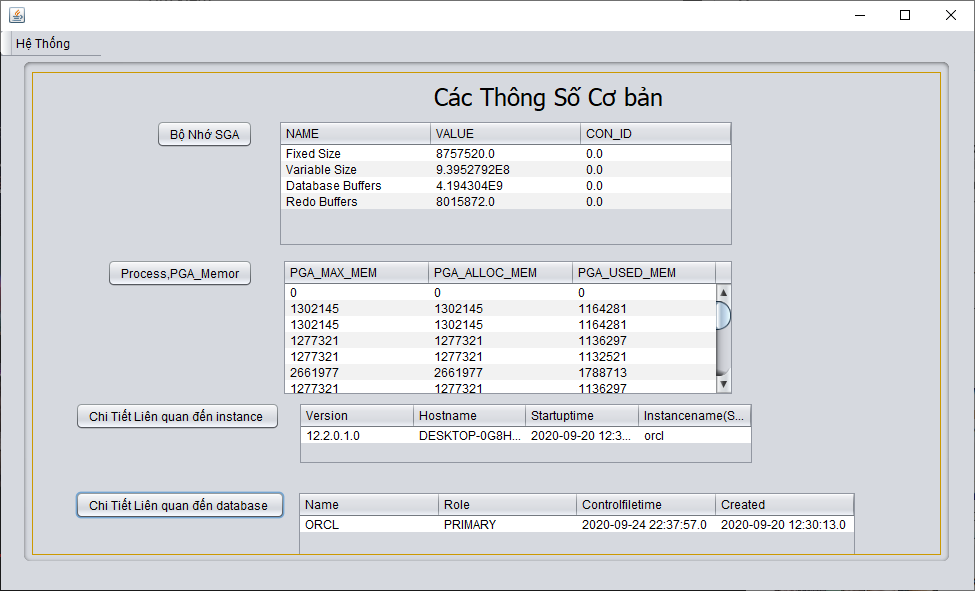
1. **Thiết kế giao diện hệ thống, giao diện các chức năng theo yêu cầu truy vấn đồ án:**





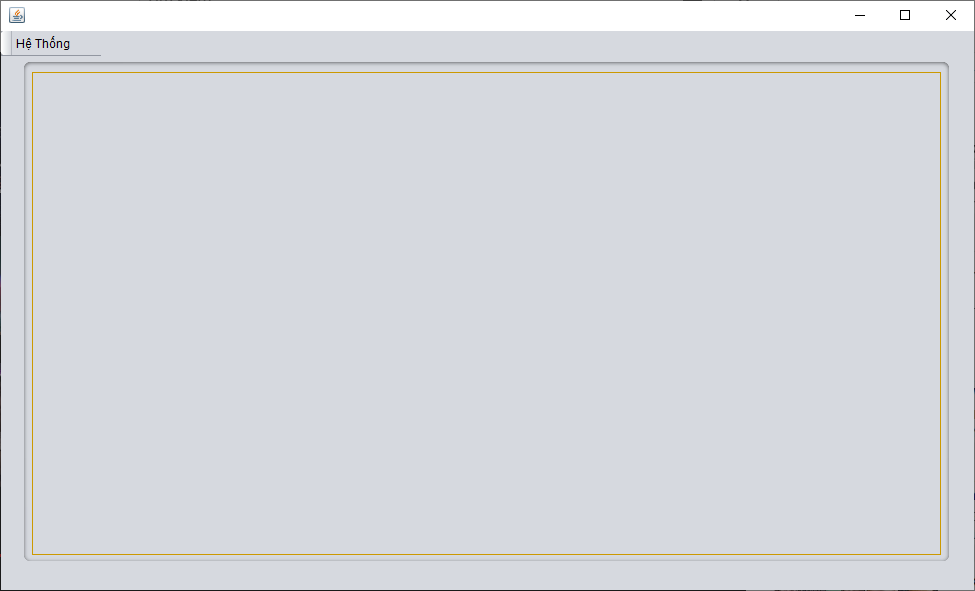


**Các button dc kích hoạt:**

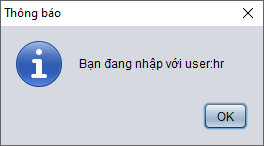


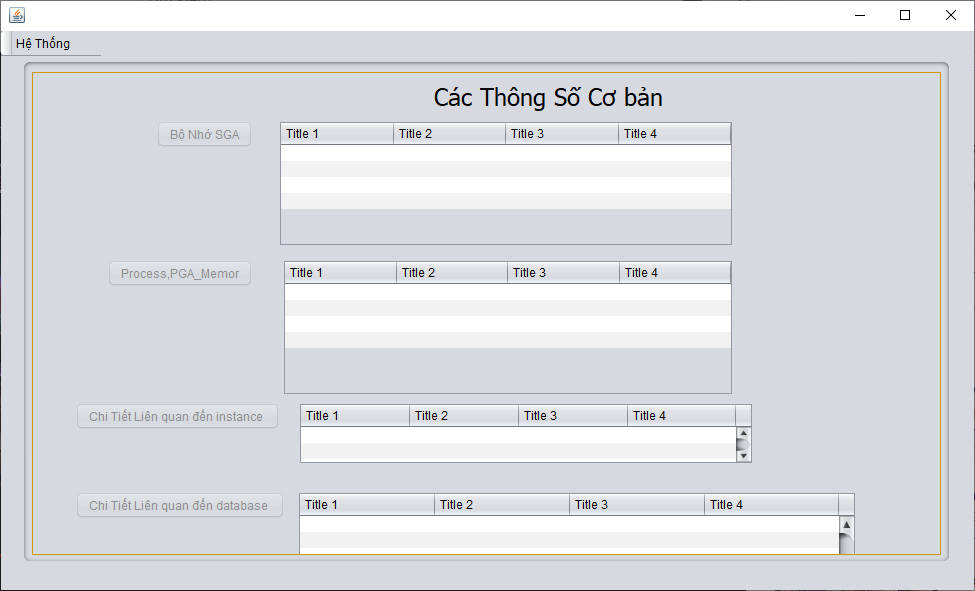
**Đăng nhập không phải là sysdba**





**Click vào hệ thống :**

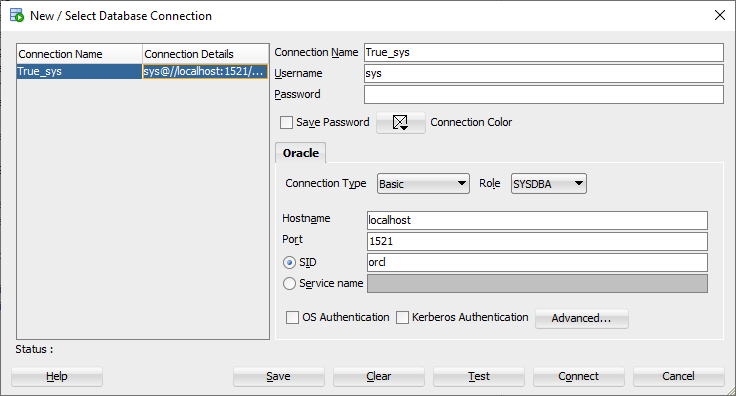




**Các chức năng bị vô hiệu hóa.**

**Các bước làm:**

Bước 1 :Tạo kết nối vào sql Developper và chọn các thông số sau đây.



Tuy nhiên khi username sys không hoạt động được bằng role sysdba

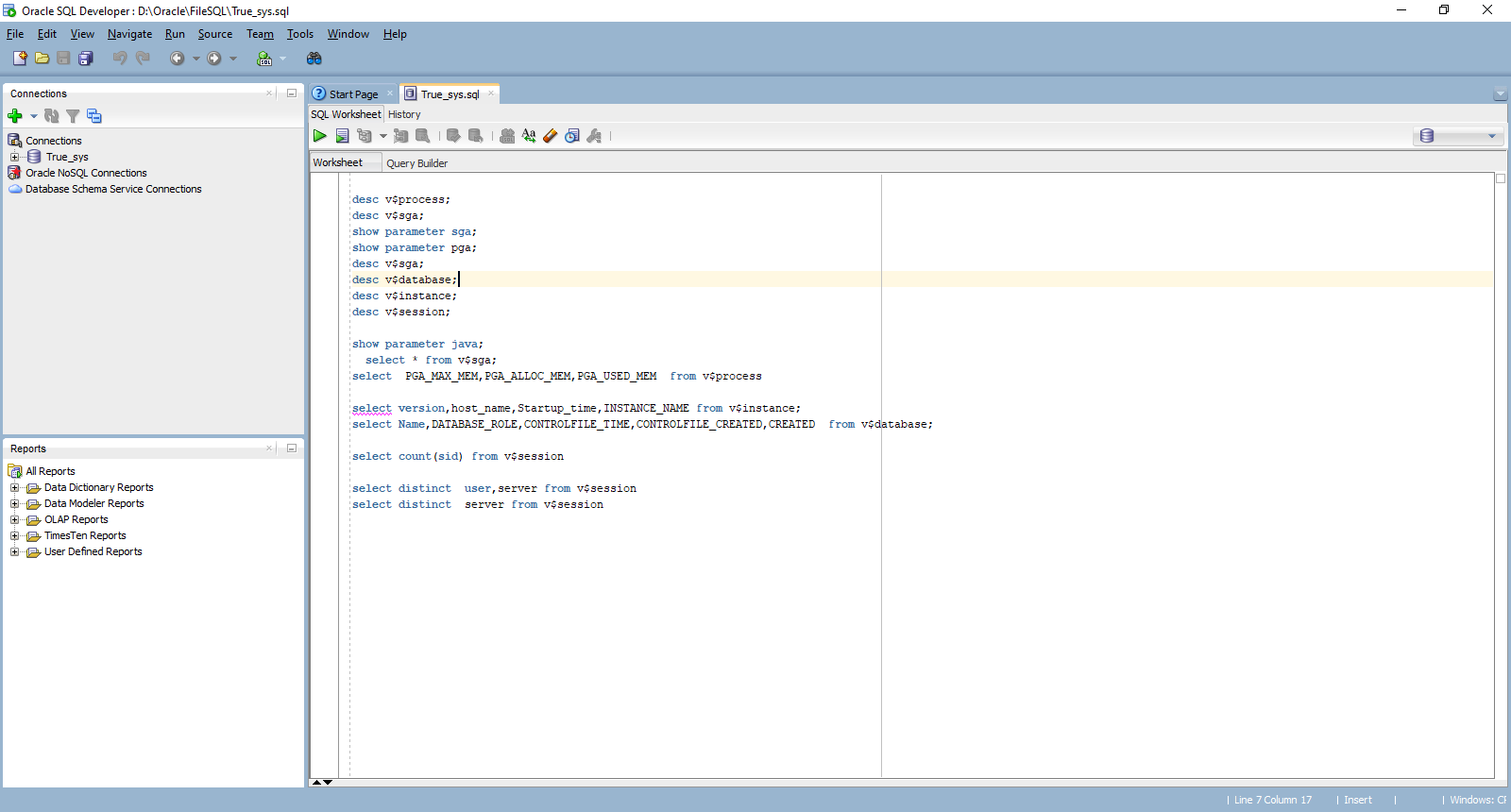
Thì vào cmd->sqlplus sys/sys as sysdba ,chạy lệnh như sau:

Alter user <username> identified by <new password>;

Alter user sys identified by sys ;

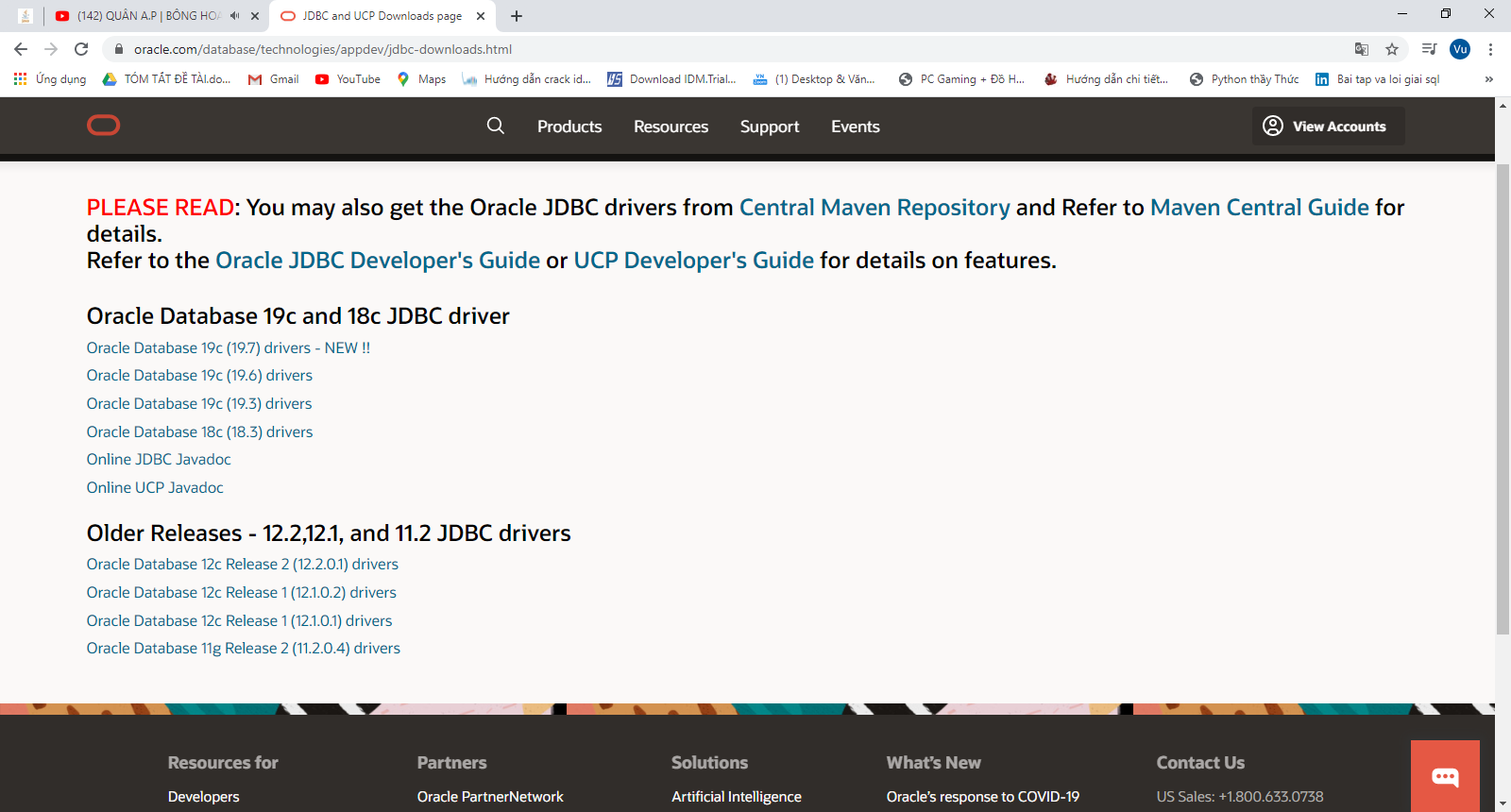
->kết nối dc developper bình thường.

Bước 2: Tạo các câu lệnh truy vấn cơ bản liên quan đến thông tin oracle



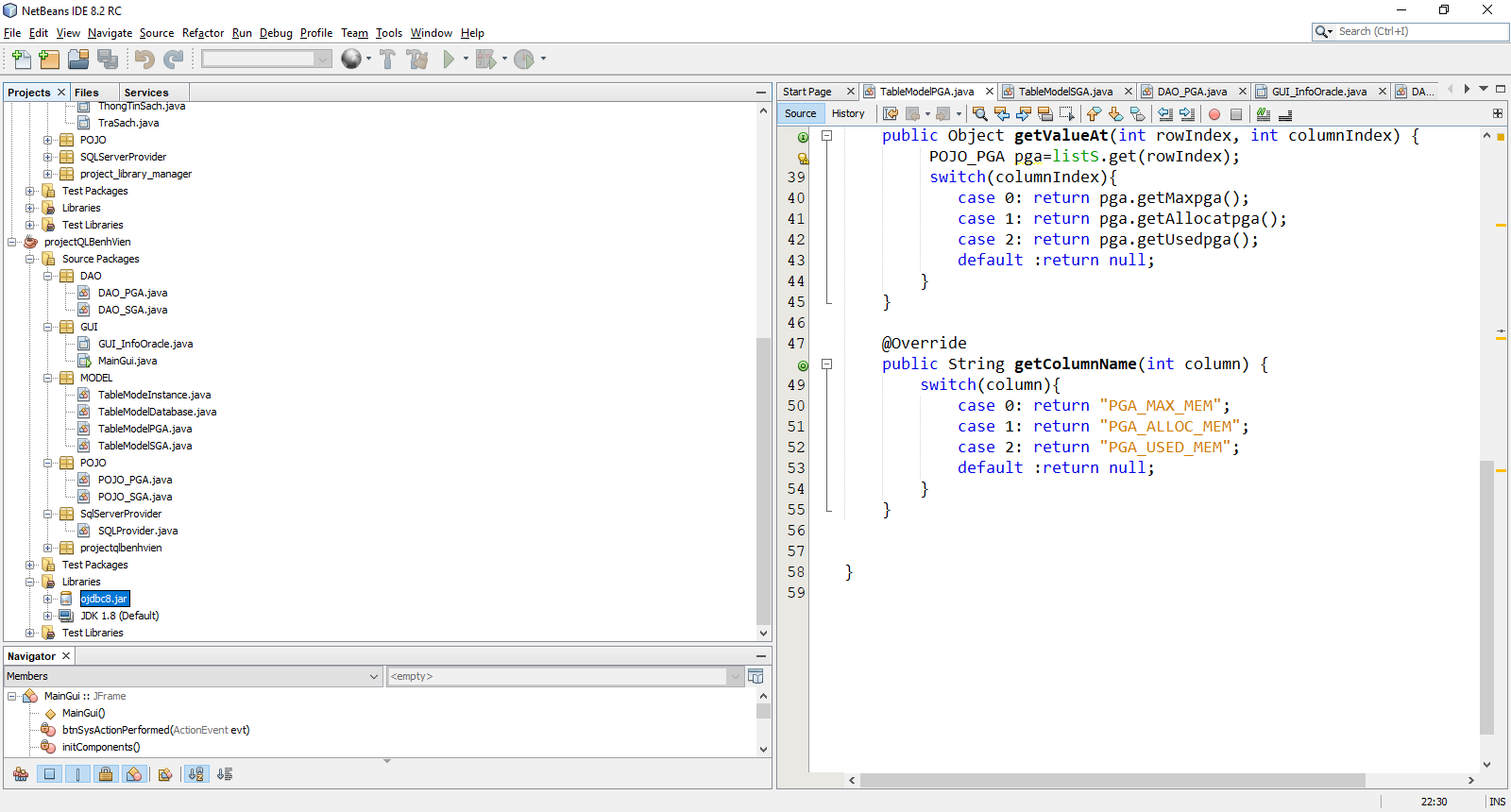
Bước 3: Vào IDE netbean :

Trước khi vào phải tải driver kết nối jdbc



Ở đây mình dùng O12 C Release 2 -> click-> tải bản orcle8.jar

Rồi đưa vào library



Xong, Tạo các package SQL provider, POJO, DAO, MODEL, GUI và các lớp tương ứng.

Bước 4: Viết code sql provider để kết nối với Oracle = sqlplus:

Cách 1:

String driver="oracle.jdbc.OracleDriver";

String urlConnection="jdbc:oracle:thin:@localhost:1521:orcl";

String username="sys as sysDBA";

String password="sys";

Connection connection=null;

public void Open() throws ClassNotFoundException, SQLException{

Class.forName(driver);

connection=DriverManager.getConnection(urlConnection,username,password);

}

public void Close() throws SQLException{

connection.close();

}

public ResultSet executeQuery(String sql) throws SQLException

{

Statement statement=connection.createStatement();

ResultSet rs =statement.executeQuery(sql);

return rs;

}

public int executeUpdate(String sql) throws SQLException

{

int n=-1;

Statement stm=connection.createStatement();

n=stm.executeUpdate(sql);

return n;

}

Cách 2:

String driver="oracle.jdbc.OracleDriver";

String urlConnection="jdbc:oracle:thin:hr/hr@localhost:1521:orcl";

public void Open() throws ClassNotFoundException, SQLException{

Class.forName(driver);

connection=DriverManager.getConnection(urlConnection);

}

Cách 3:dùng properties obj trong java:

String driver="oracle.jdbc.OracleDriver";

String urlConnection="jdbc:oracle:oce:@localhost";

Properties properties=new Properties();

Properties.put(“user”,”scott”);

Properties.put(“password”,”scott”);

Properties.put(“defaultRowPrefetch”,”20”);

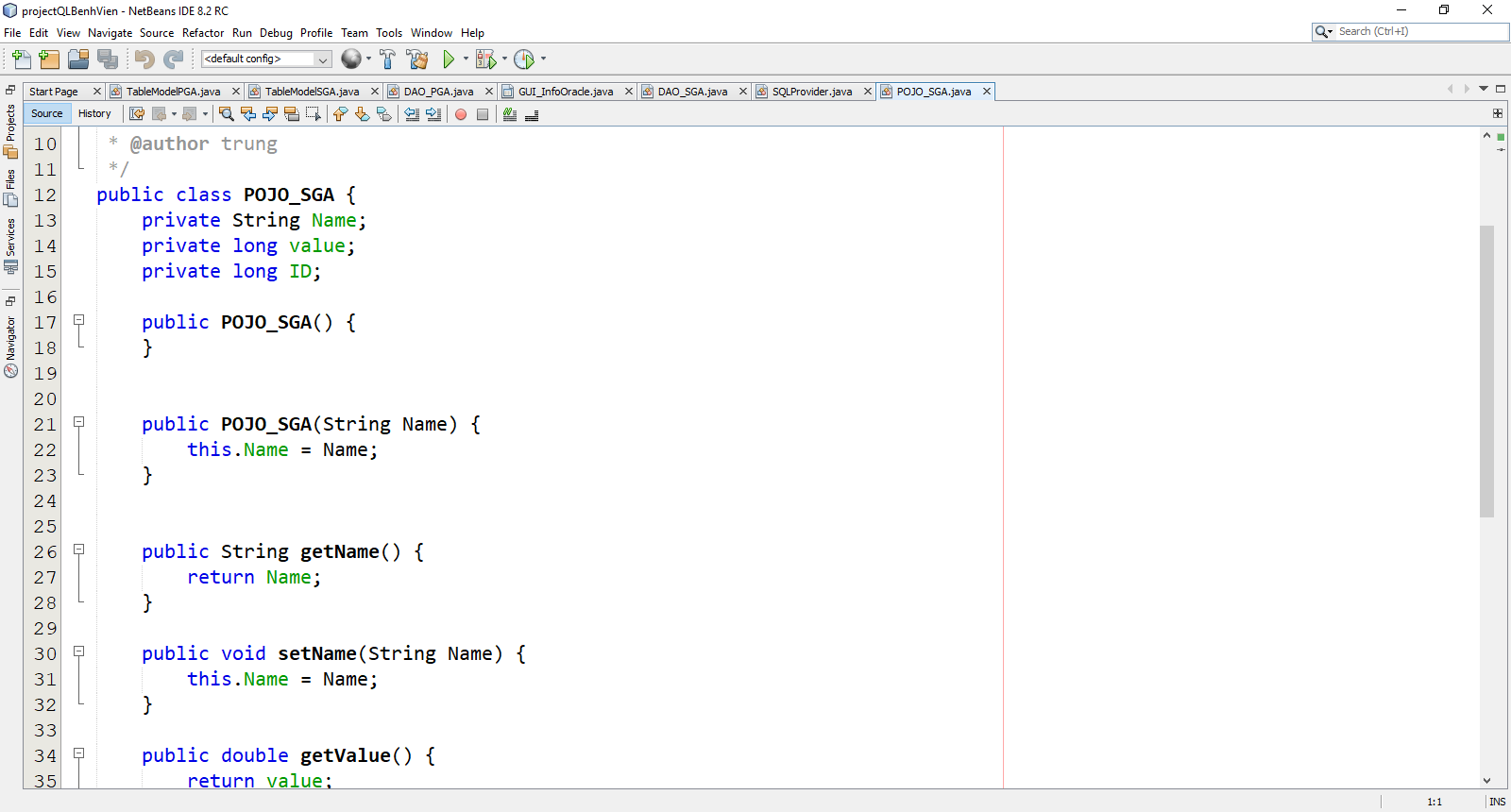
public void Open() throws ClassNotFoundException, SQLException{

Class.forName(driver);

connection=DriverManager.getConnection(urlConnection,properties);

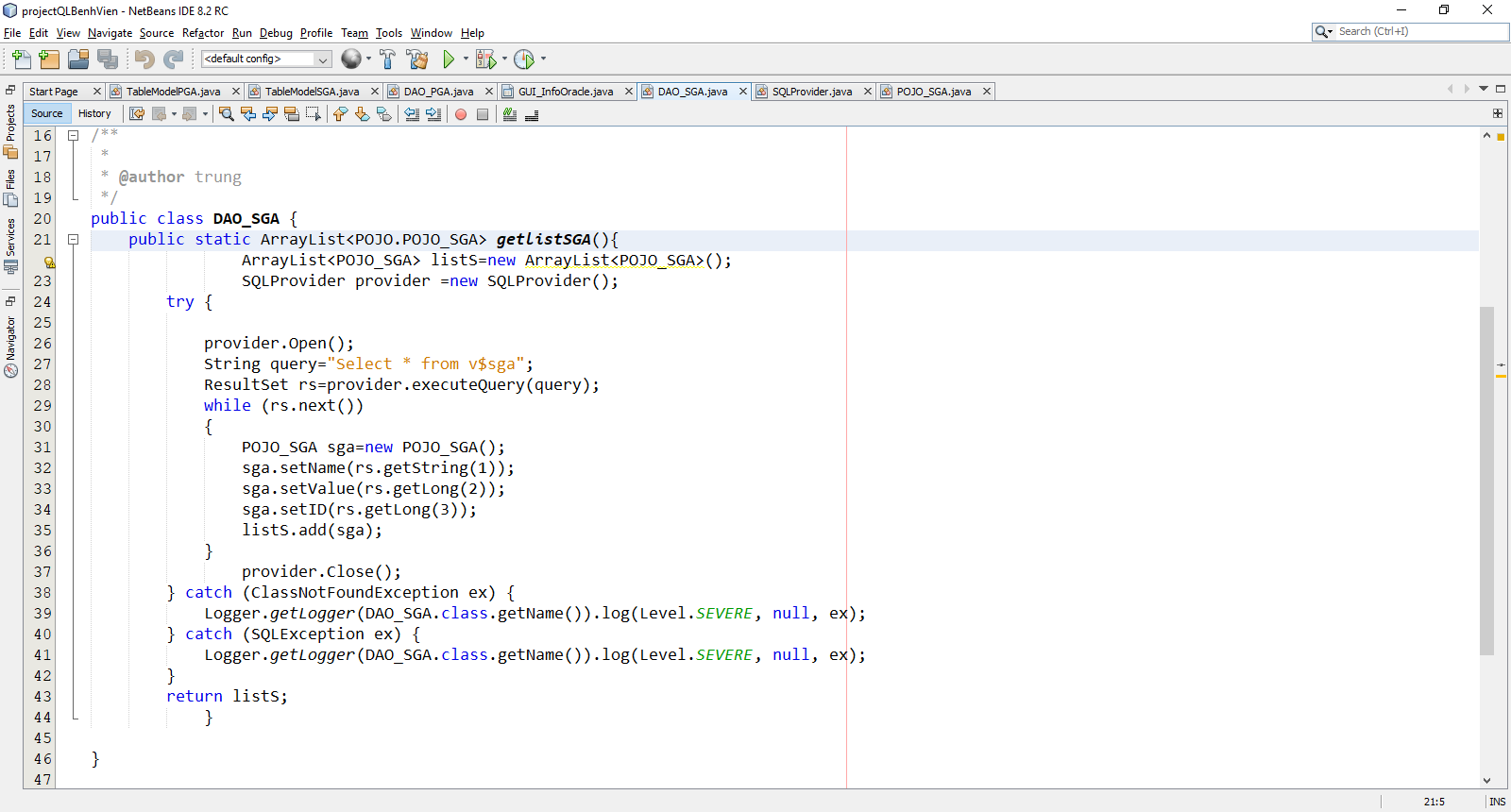
}

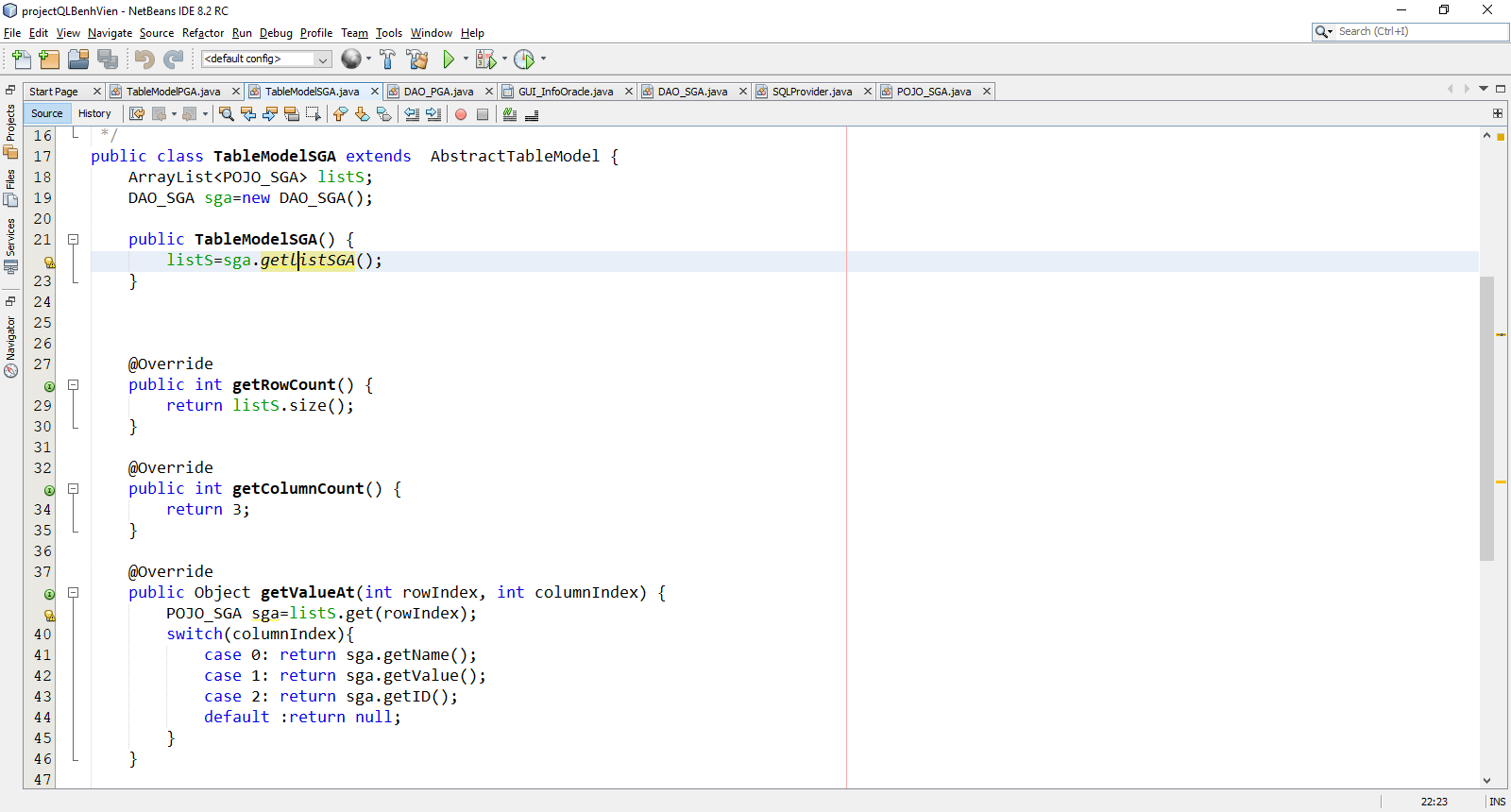
Bước 5: Tạo các lớp pojo ->dao->model->Gui:

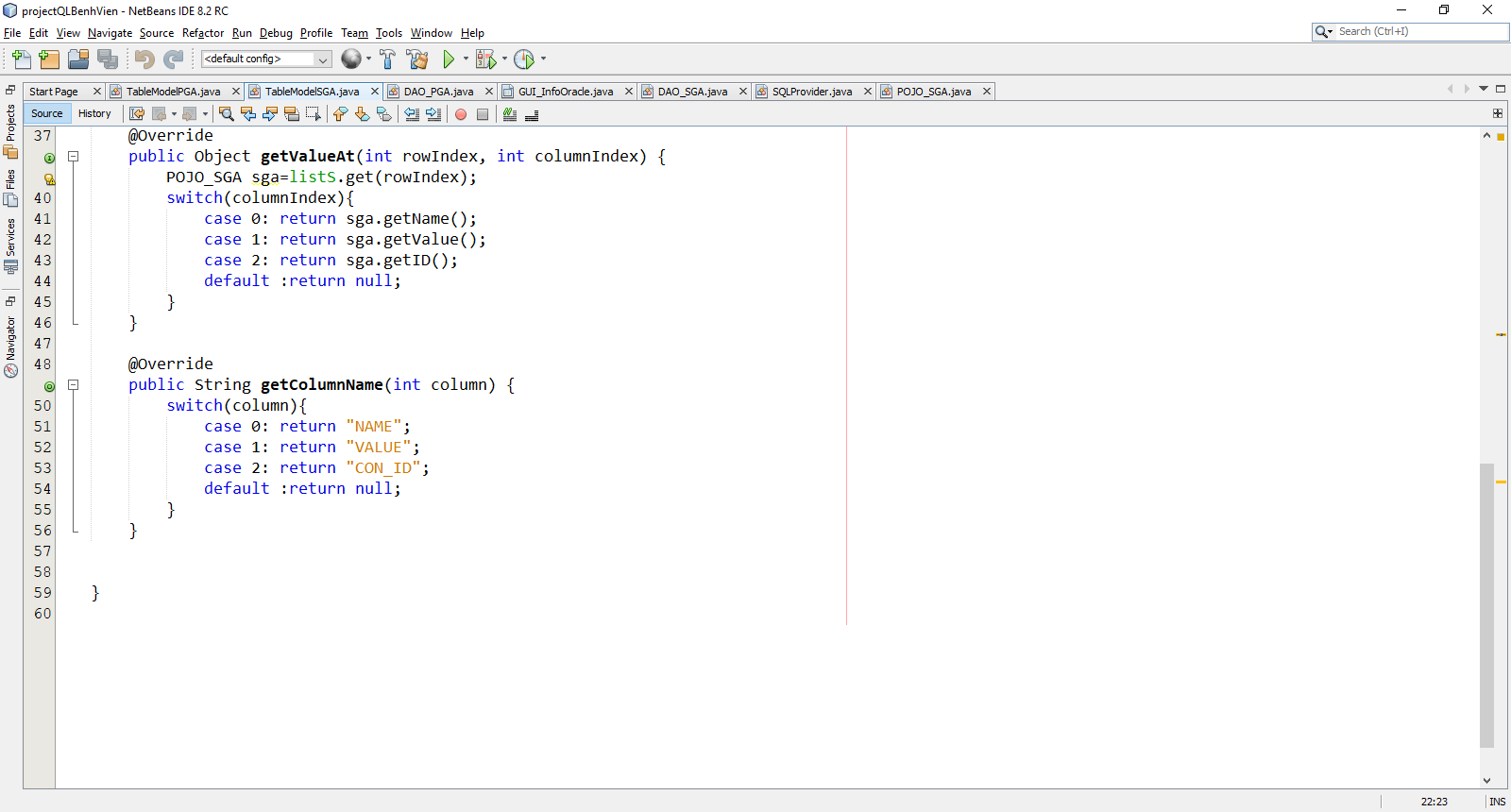


Get set cho các thuộc tính.

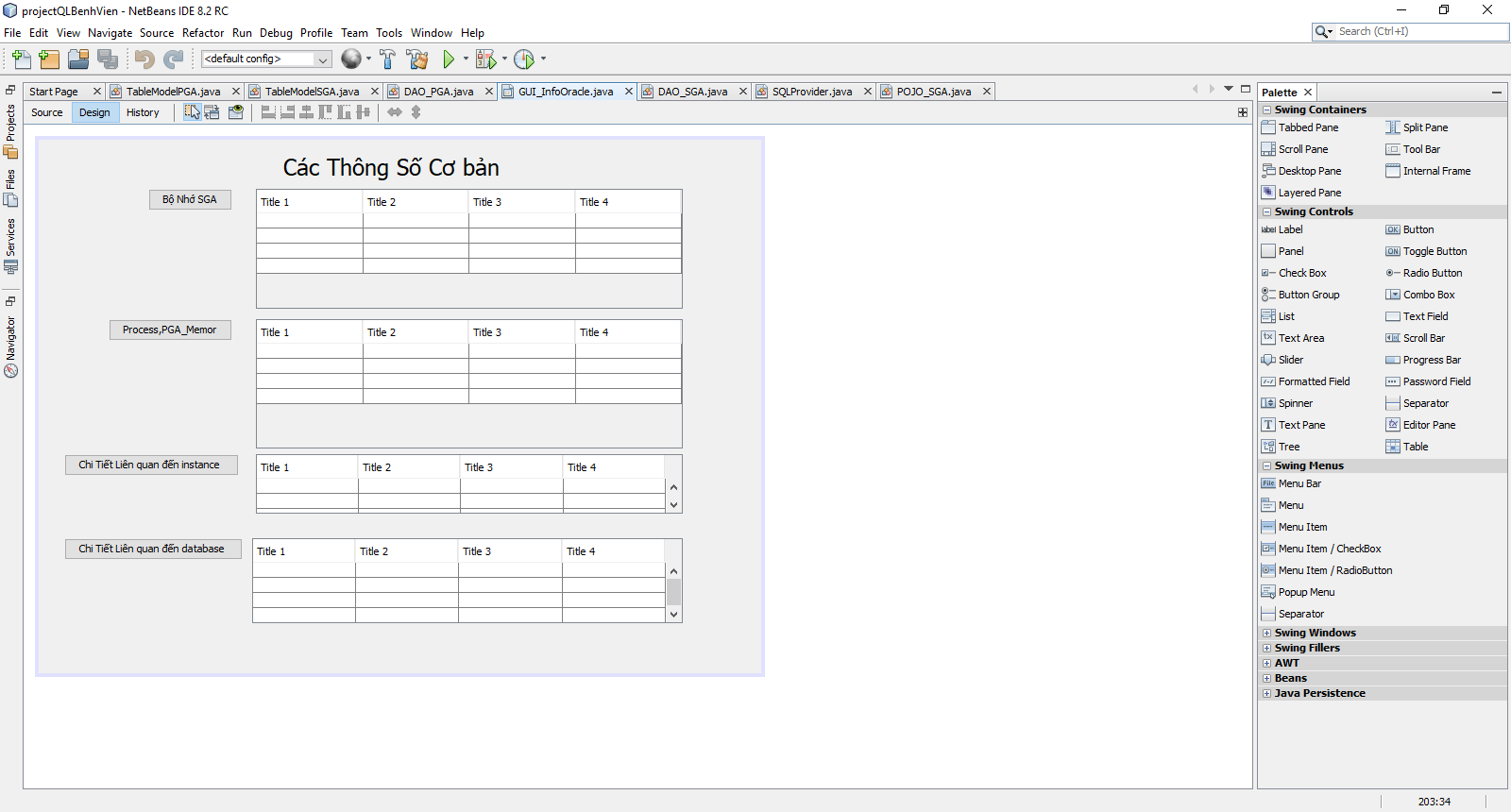
DAO:



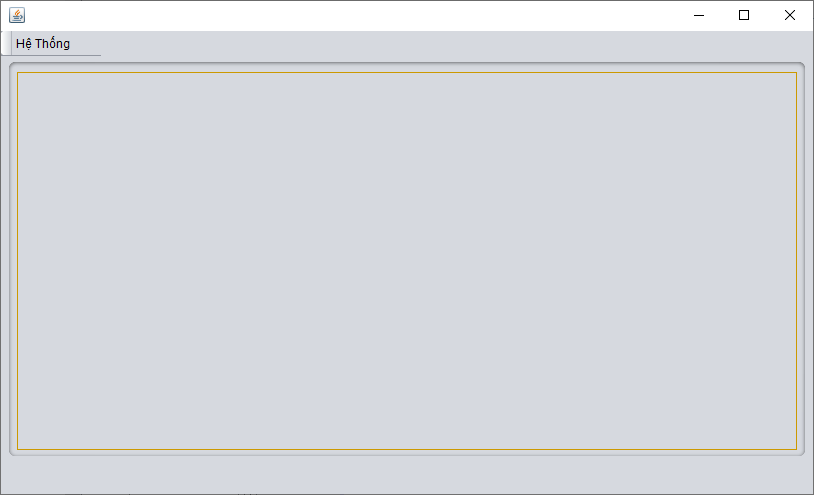
MODEL:  




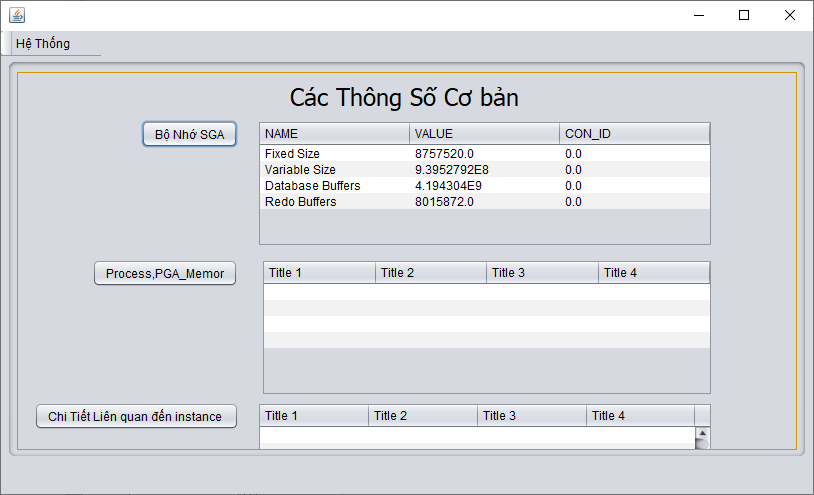
GUI: kéo thả :

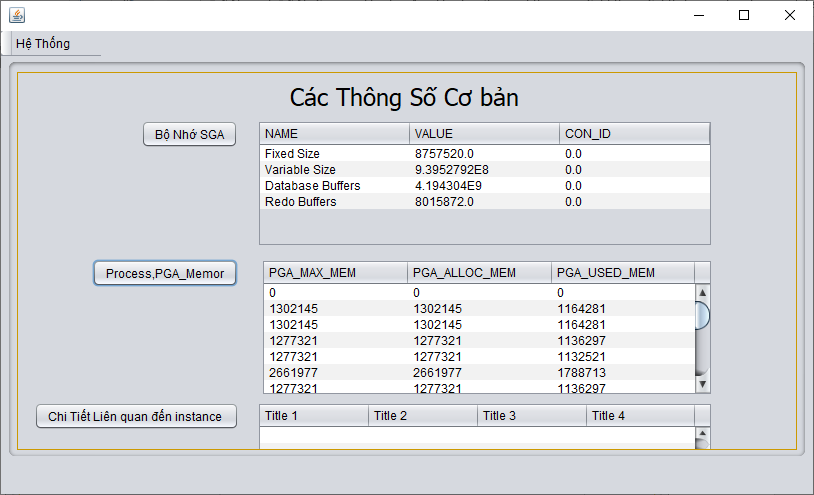


Kết quả:

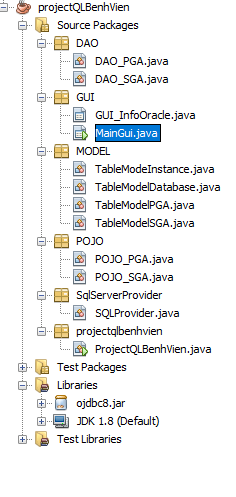


Click hệ thống->bộ nhớ sga :





Tương tự các lớp khác:



**Với trang login kéo thả như sau :**

Tạo các lớp pojo của user là biến static vì biến này sẽ giữ lun kết quả ,không bị mất khi qua các trang khác :

public class POJO\_User {

public static String username,password;

public static String getUsername() {

return username;

}

public static void setUsername(String username) {

POJO\_User.username = username;

}

public static String getPassword() {

return password;

}

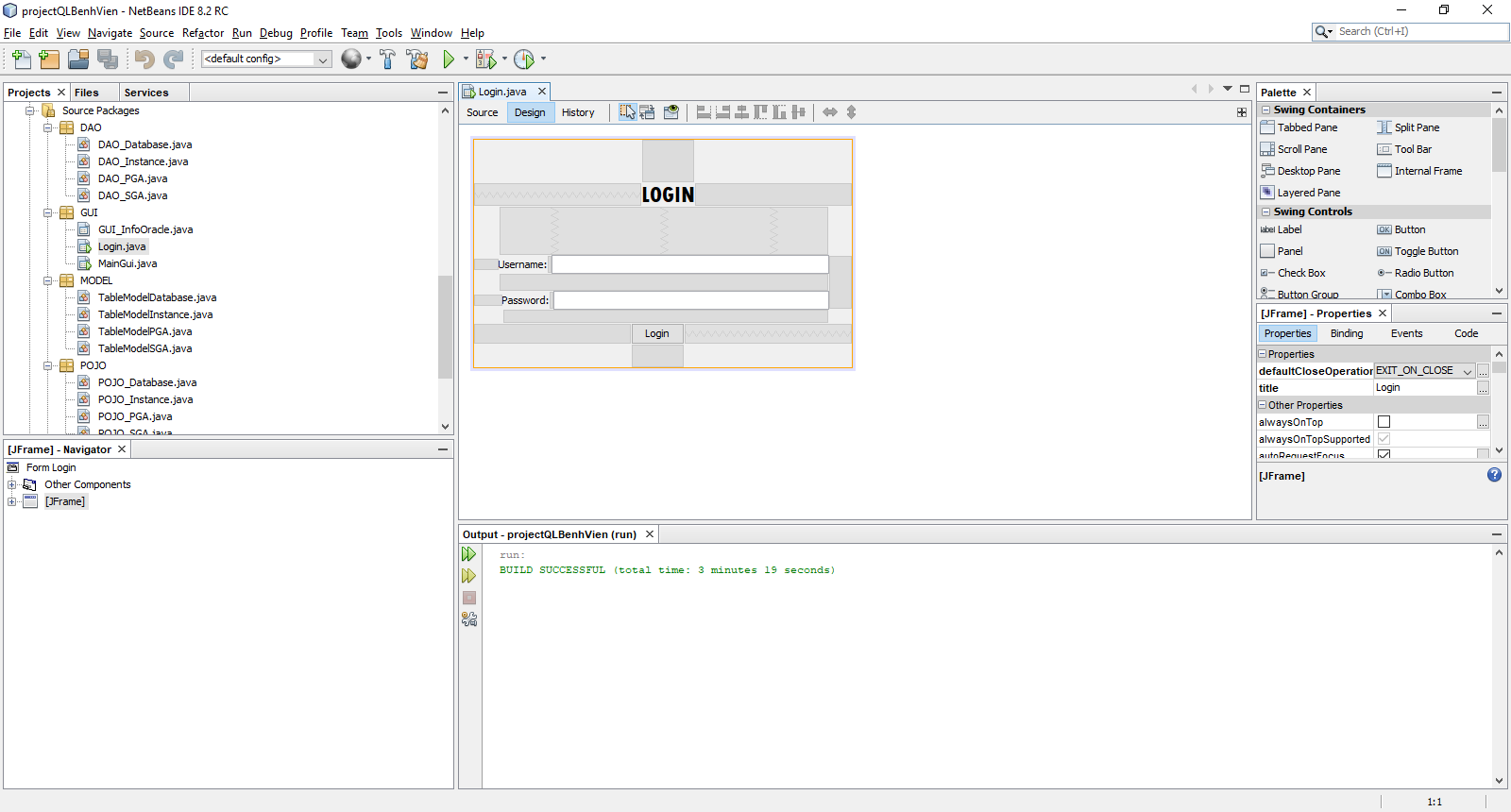
public static void setPassword(String password) {

POJO\_User.password = password;

}

public POJO\_User() {

}



**Kích hoạt chức năng nút bấm login:**

private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {

user.setUsername(txtUser.getText());

user.setPassword(txtPass.getText());

SQLProvider provider =new SQLProvider();

try {

// TODO add your handling code here:

provider.Open(user.getUsername(),user.getPassword());

provider.Close();

dispose();

JOptionPane.showMessageDialog(this,"Login Successfully","Notice",JOptionPane.INFORMATION\_MESSAGE);

MainGui gui=new MainGui();

gui.setLocationRelativeTo(null);

gui.show();

} catch (ClassNotFoundException ex) {

Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(this,"Login Failed","Error",JOptionPane.ERROR\_MESSAGE);

return;

} catch (SQLException ex) {

Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(this,"Login Failed","Error",JOptionPane.ERROR\_MESSAGE);

return;

}

}

Để Sử dụng chức năng tắt mở database trên JAVA:

public class SQLConnectThrowJDBC {

Properties prop = new Properties();

OracleDataSource ods;

OracleConnection ocon;

public void Open(String user,String pass) throws SQLException{

prop.setProperty("user","sys as sysdba");

prop.setProperty("password","sys");

prop.setProperty("internal\_logon","sysdba");

prop.setProperty("prelim\_auth","true");

ods = new OracleDataSource();

ods.setConnectionProperties(prop);

ods.setURL("jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=localhost)(PORT=1521))"

+ "(CONNECT\_DATA=(SERVICE\_NAME=orcl)))");

ocon = (OracleConnection)ods.getConnection();

System.out.println("Connected");

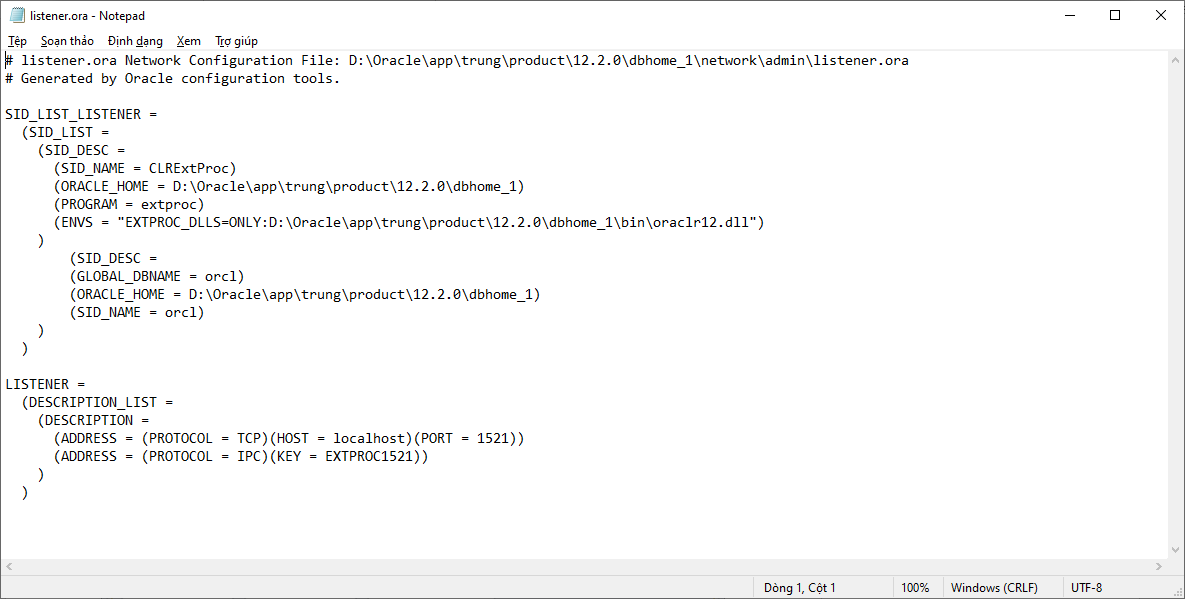
}

Nếu Gặp lỗi : ORA-12514: TNS:listener does not currently know of service requested in connect descriptor

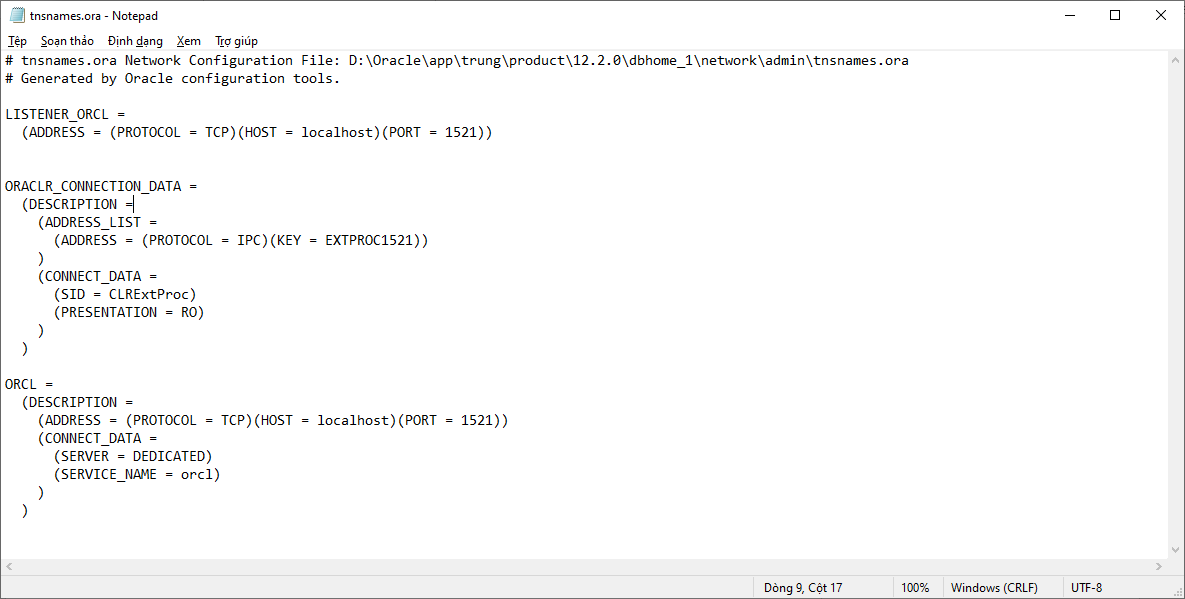
Tiến hành các bước :

Bước 1: vào tìm file có trong đường dẫn D:\Oracle\app\trung\product\12.2.0\dbhome\_1\network\admin

Tại file listener.ora chỉnh lại như sau:



Tại file tnsnames.ora

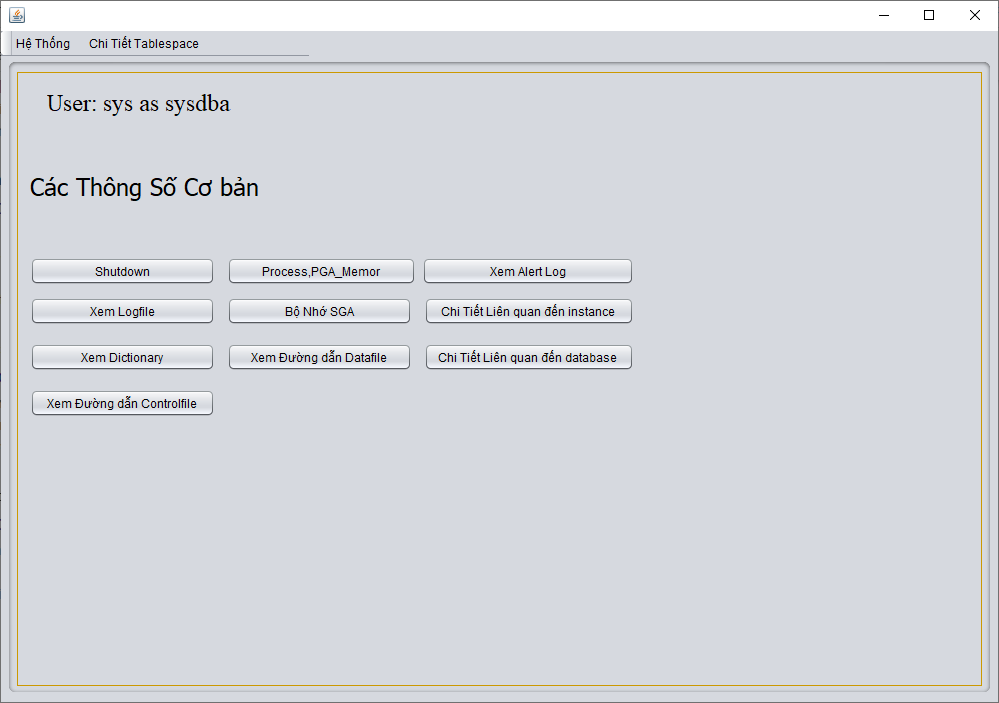


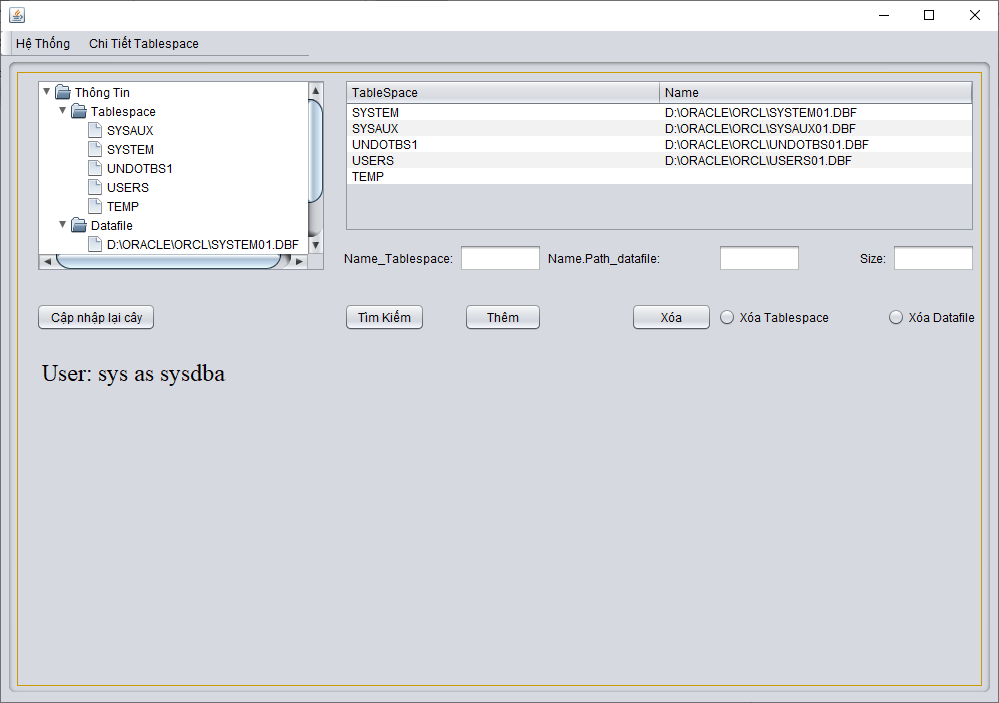
Bước 3 :

Vào cmd khởi động lại file listener

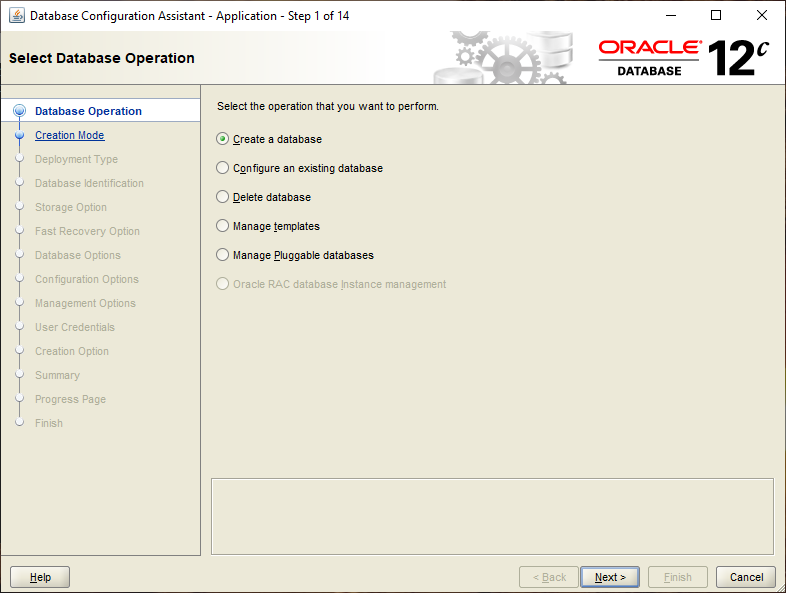
Chạy lệnh lsnrctl stop

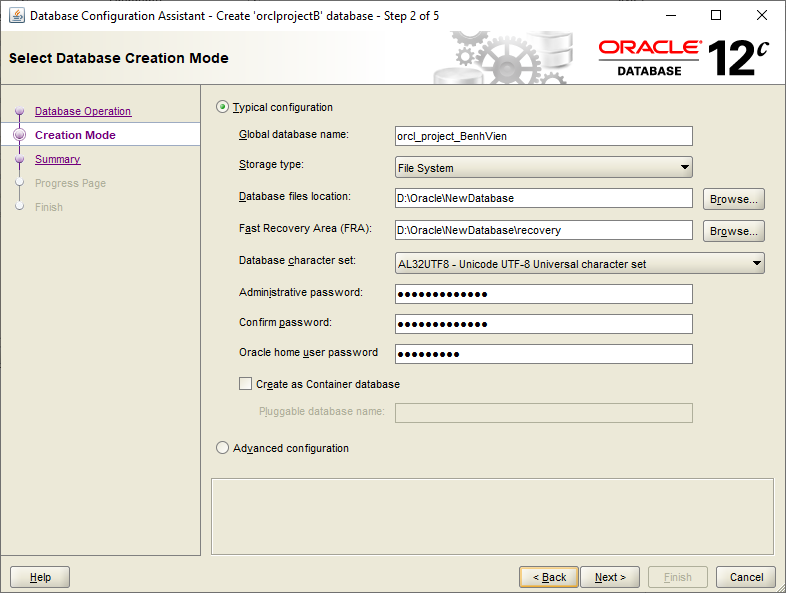
lsnrctl start

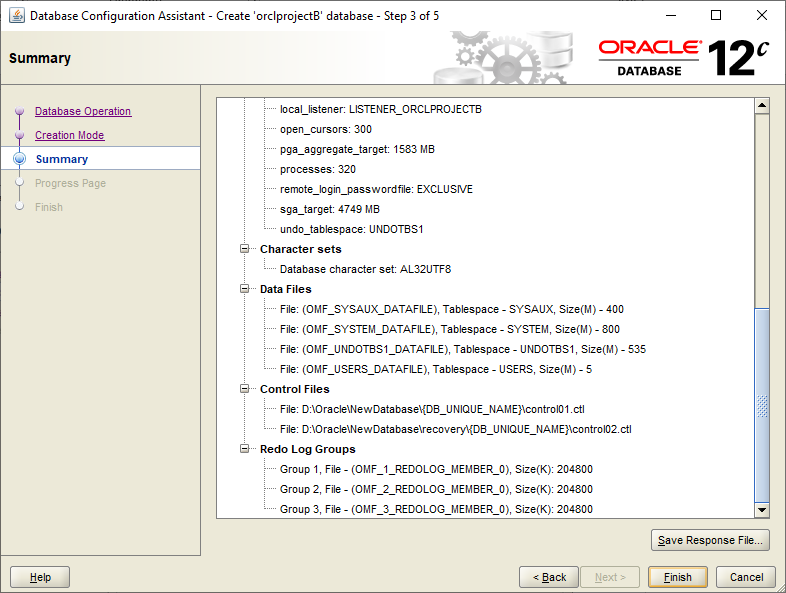


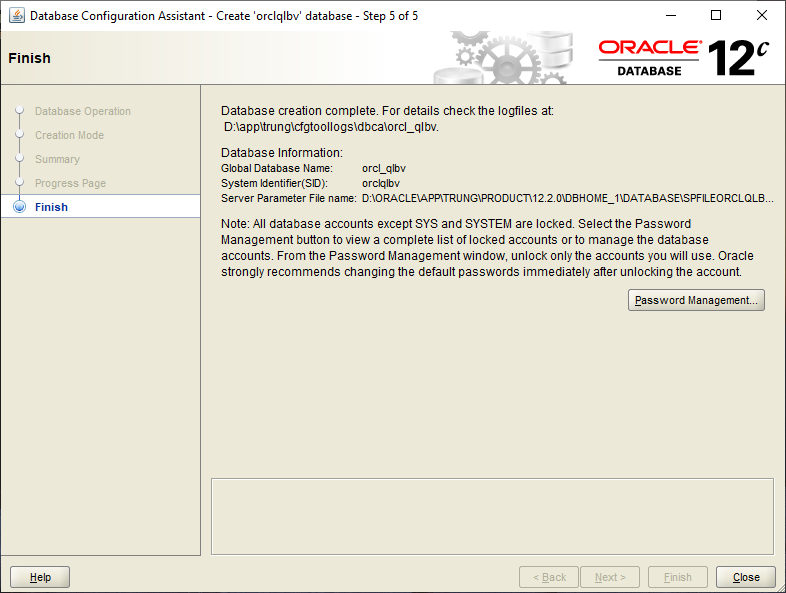


DBCA



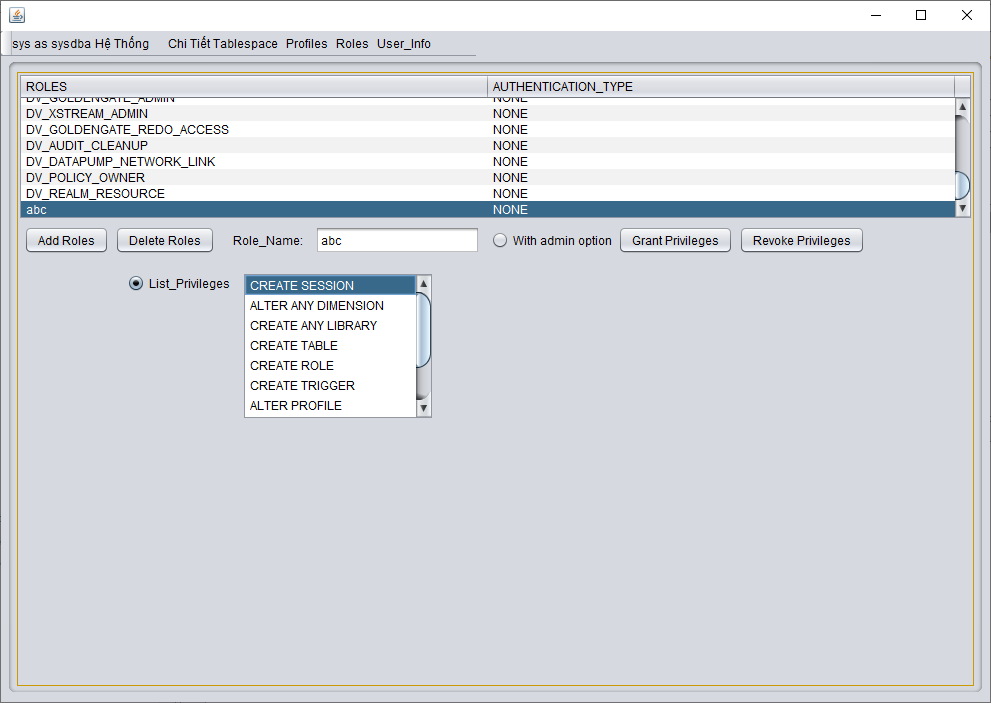






Close xong!

Nhóm Quyền / Thu hồi nhóm quyền.



Code behind

Trong bảng list chứa các câu lệnh quyền:

public void initList()

{

listModelPrivileges =new DefaultListModel();

listModelPrivileges.addElement("CREATE SESSION");

listModelPrivileges.addElement("ALTER ANY DIMENSION");

listModelPrivileges.addElement("CREATE ANY LIBRARY");

listModelPrivileges.addElement("CREATE TABLE");

listModelPrivileges.addElement("CREATE ROLE ");

listModelPrivileges.addElement("CREATE TRIGGER");

listModelPrivileges.addElement("ALTER PROFILE");

listModelPrivileges.addElement("CREATE ANY INDEX");

listModelPrivileges.addElement("DROP USER");

listModelPrivileges.addElement("READ ANY TABLE");

lstPrivileges.setModel(listModelPrivileges);

}

Các câu lệnh cấp /thu trong lớp DAO :

public static boolean grantPrivileges(String privileges,String name)

{

boolean kq=false ;

SQLConnectThrowJDBC provider =new SQLConnectThrowJDBC();

POJO\_User u=new POJO\_User();

try {

provider.Open(u.getUsername(),u.getPassword(),u.getEnable());

String query=String.format("grant %s to \"%s\" ",privileges,name);

int n=provider.executeUpdate(query);

if (n==1)

{

kq=true;

}

provider.Close();

} catch (SQLException ex) {

Logger.getLogger(DAO\_SGA.class.getName()).log(Level.SEVERE, null, ex);

}

return kq;

}

public static boolean revokePrivileges(String privileges,String name)

{

boolean kq=false ;

SQLConnectThrowJDBC provider =new SQLConnectThrowJDBC();

POJO\_User u=new POJO\_User();

try {

provider.Open(u.getUsername(),u.getPassword(),u.getEnable());

String query=String.format("revoke %s from \"%s\" ",privileges,name);

int n=provider.executeUpdate(query);

if (n==1)

{

kq=true;

}

provider.Close();

} catch (SQLException ex) {

Logger.getLogger(DAO\_SGA.class.getName()).log(Level.SEVERE, null, ex);

}

return kq;

}

public static boolean grantPrivileges\_WithAdmin(String privileges,String name)

{

boolean kq=false ;

SQLConnectThrowJDBC provider =new SQLConnectThrowJDBC();

POJO\_User u=new POJO\_User();

try {

provider.Open(u.getUsername(),u.getPassword(),u.getEnable());

String query=String.format("grant %s to \"%s\" with admin option ",privileges,name);

int n=provider.executeUpdate(query);

if (n==1)

{

kq=true;

}

provider.Close();

} catch (SQLException ex) {

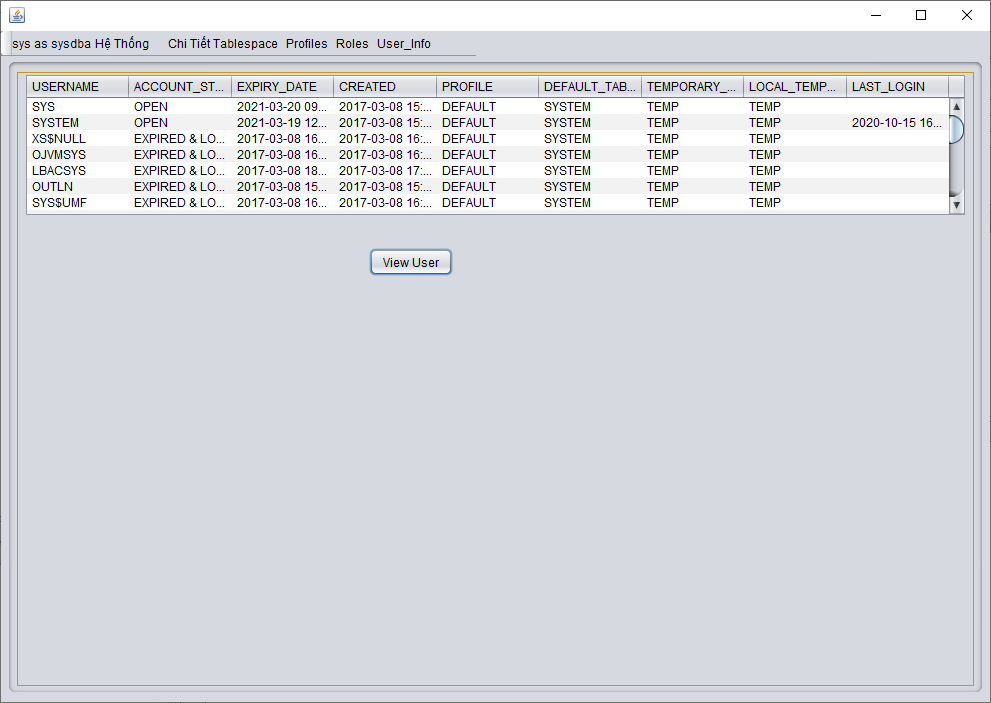
Logger.getLogger(DAO\_SGA.class.getName()).log(Level.SEVERE, null, ex);

}

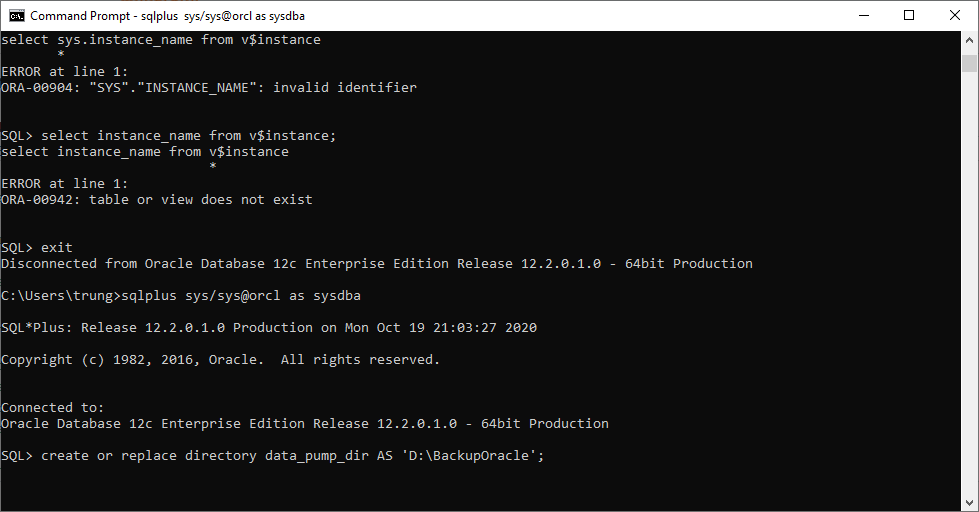
return kq;

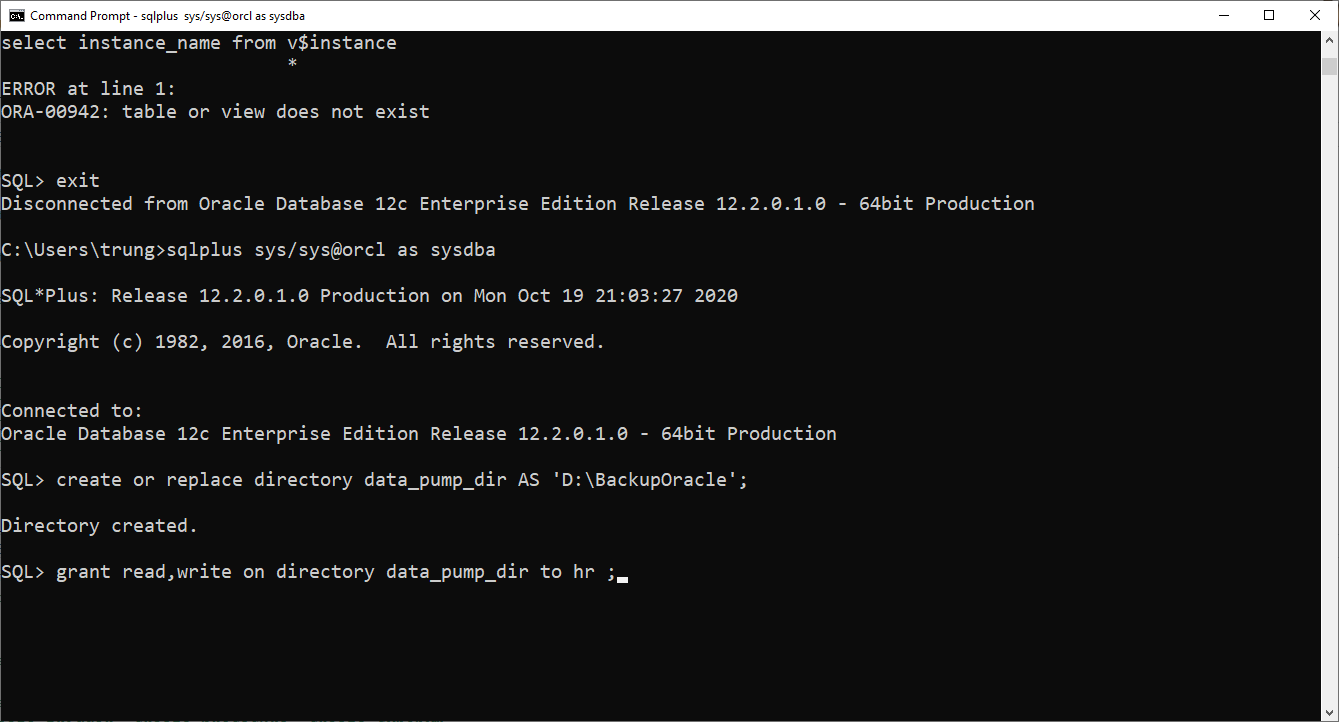
}

Giao diện User:

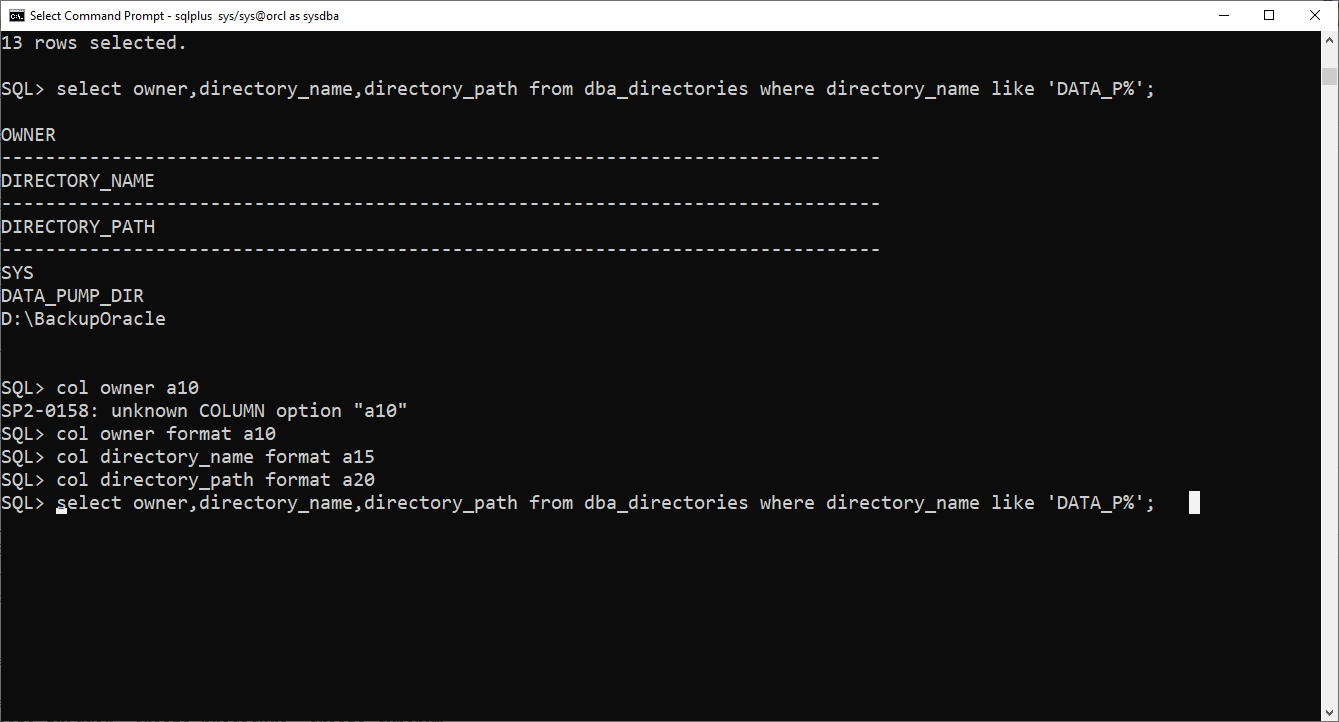


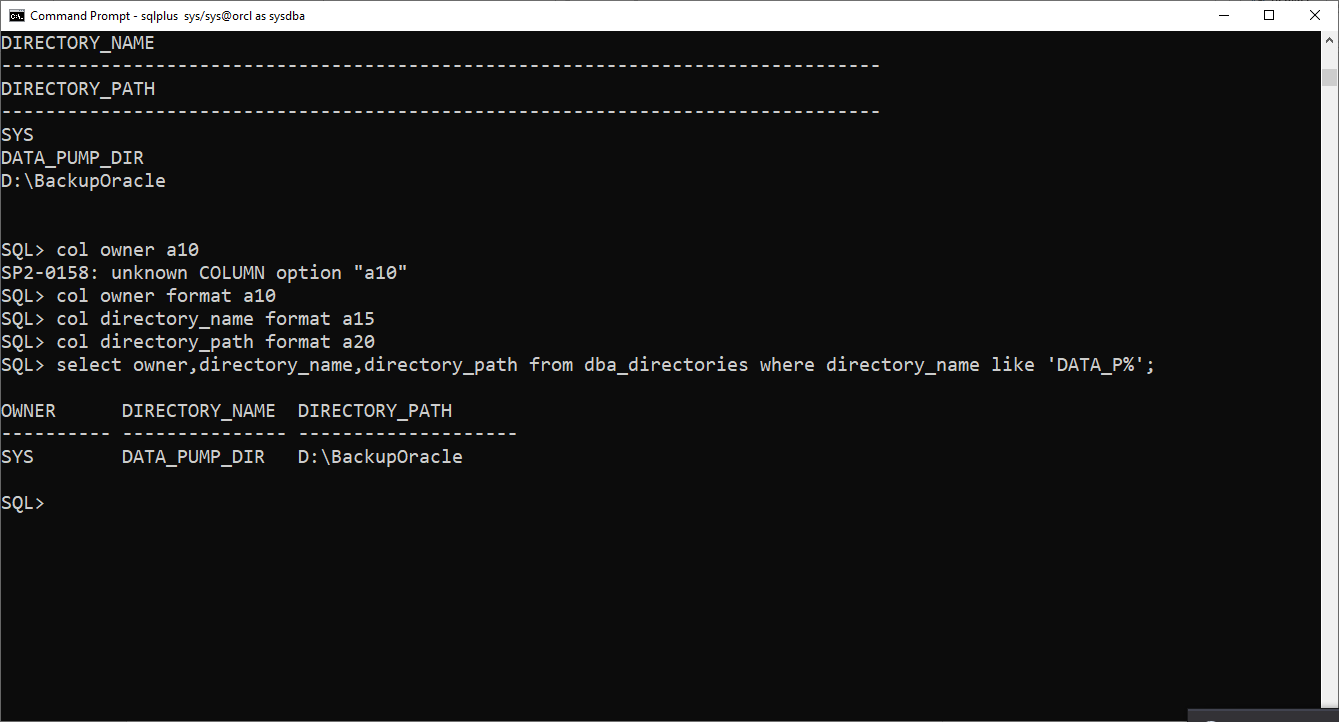
**Các Bước Sử Dụng Công Cụ Oracle Data Pump:**



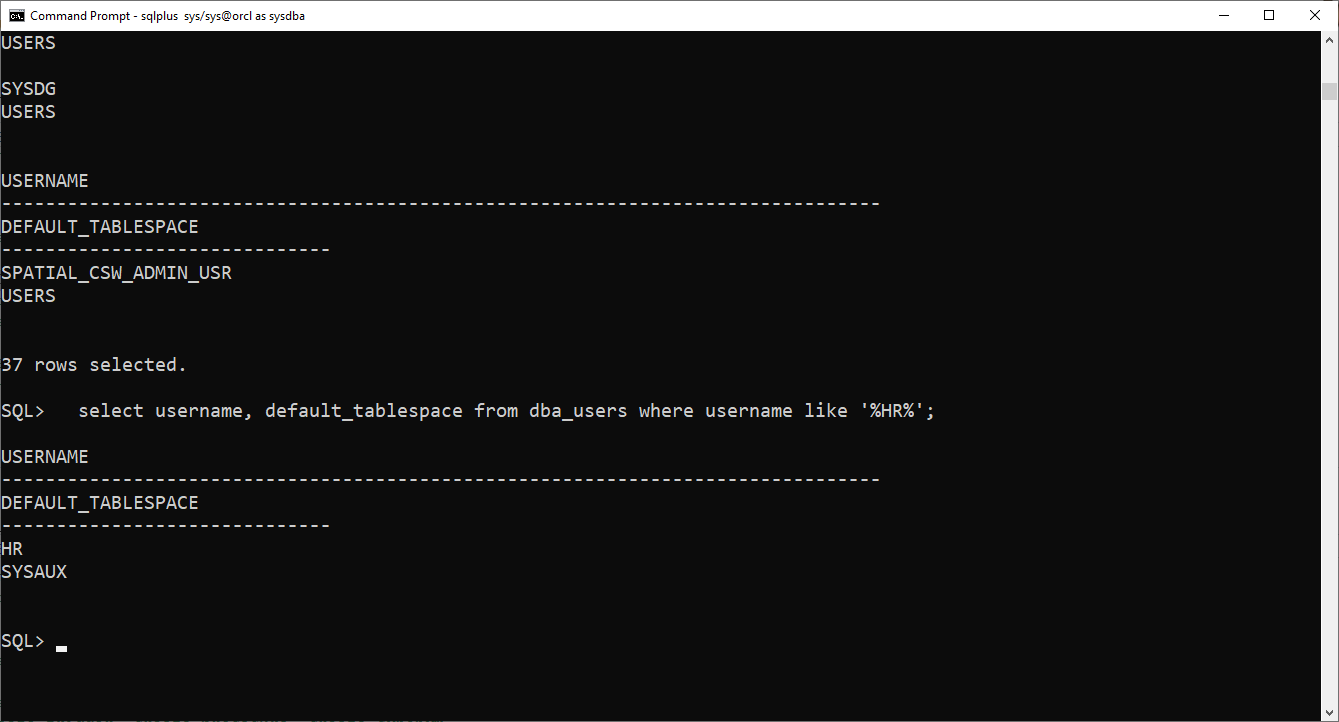


Xét độ rộng của từng cột trong DBA\_directories;

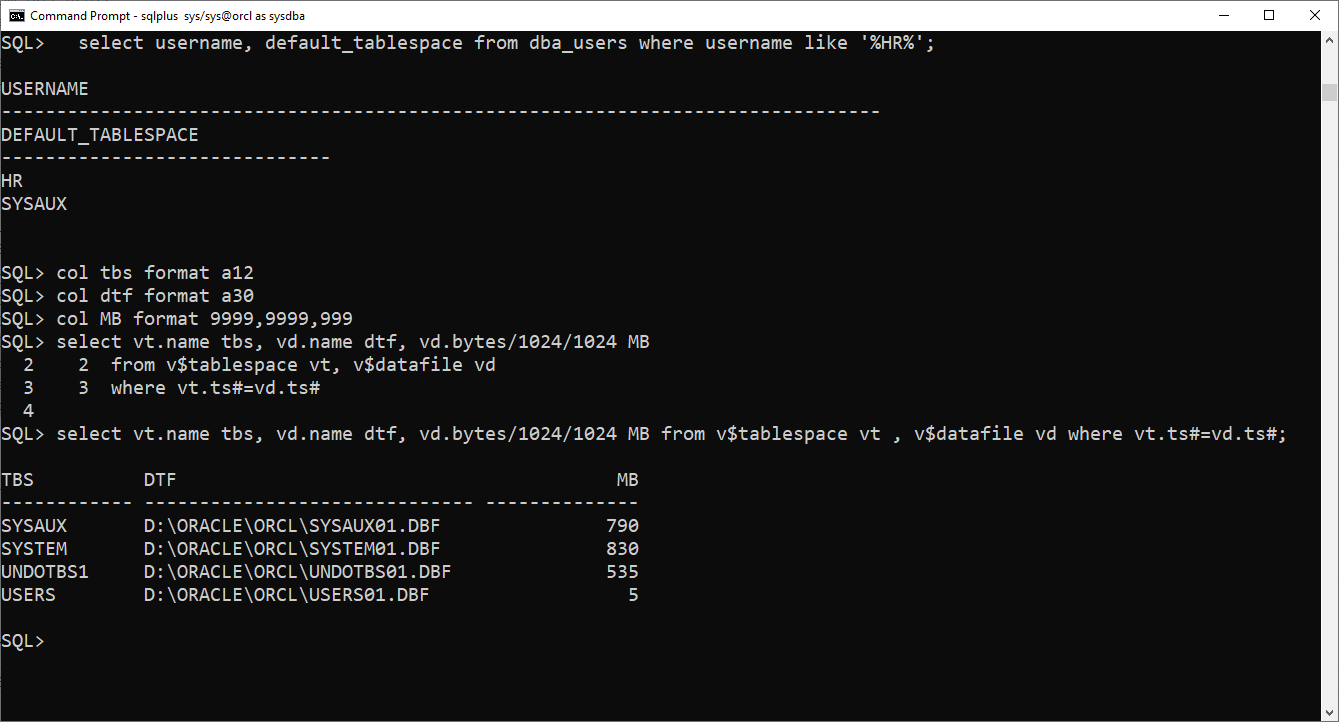




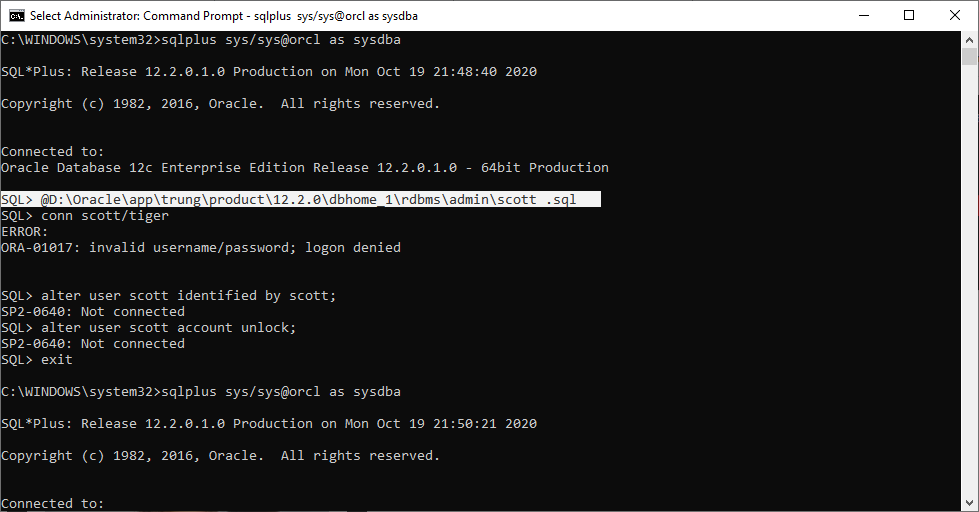
Kiểm tra tablespace của HR



Bước 2:



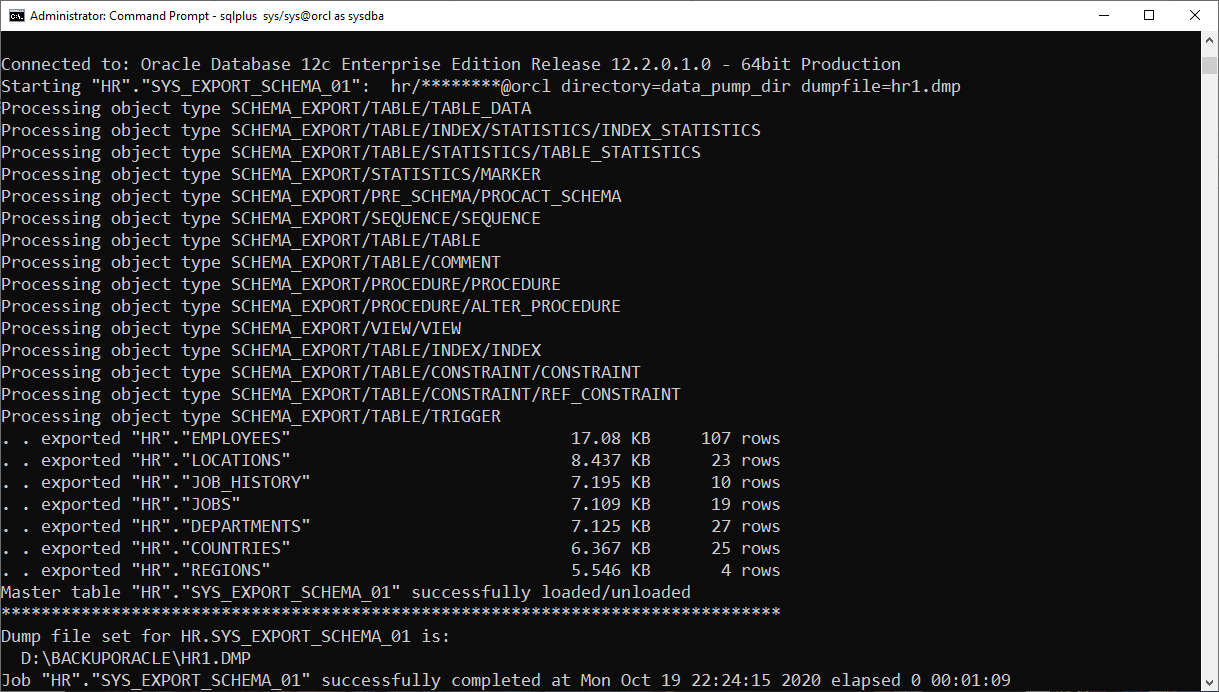
Nếu các user như scott, hr không vào được thì phải kiểm tra xem mình đang sử dụng SID nào và vào

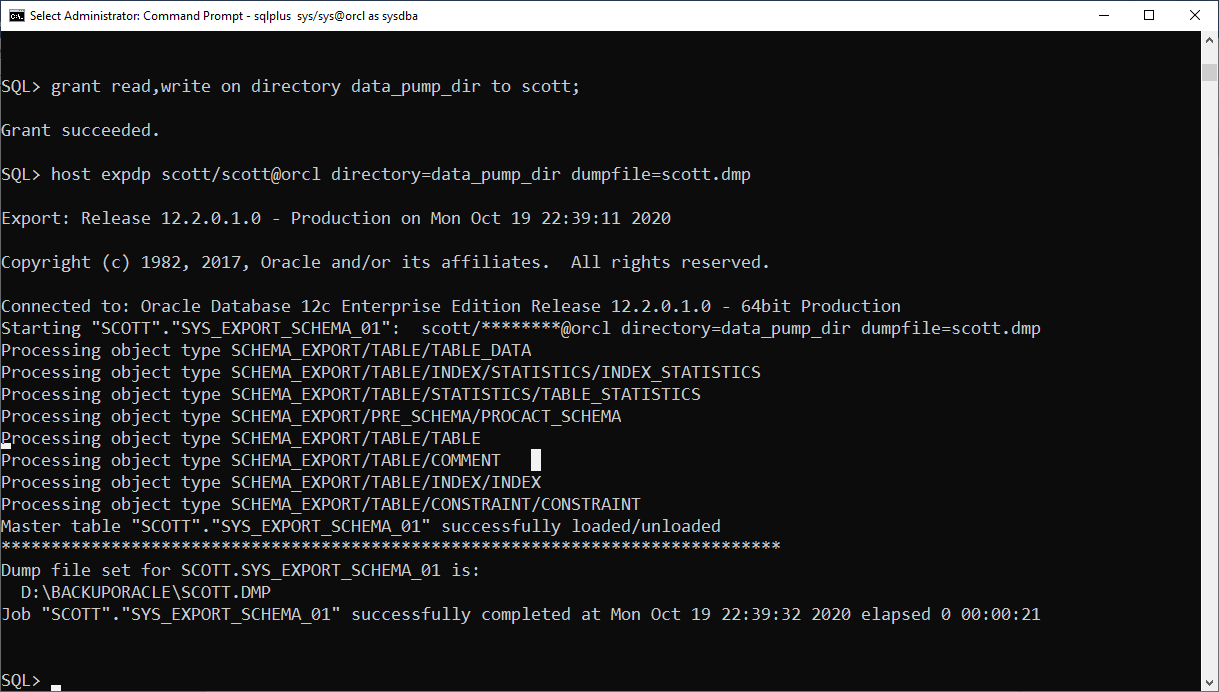


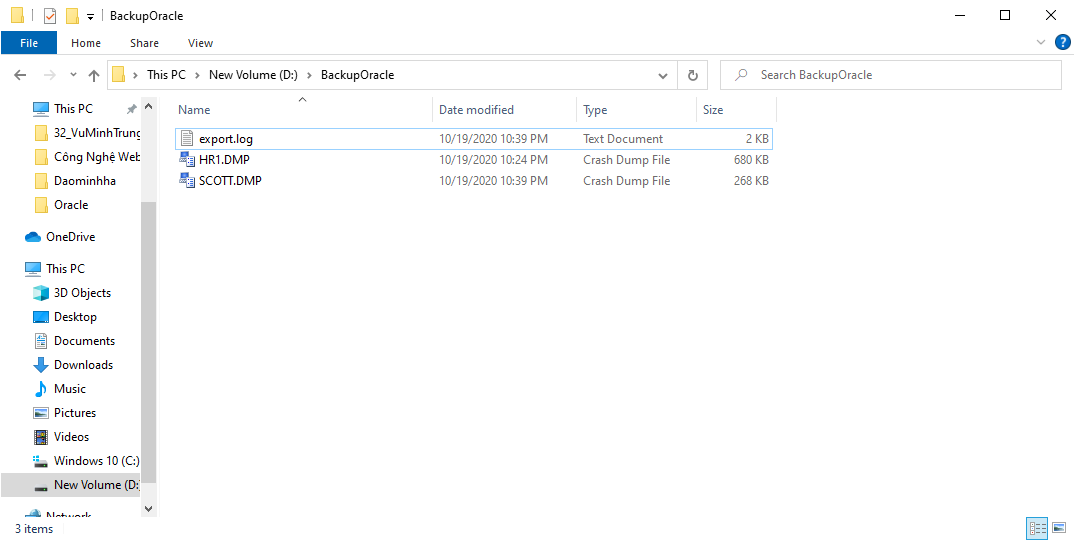
Để unlock

Dùng lệnh host expdp hr/hr@orcl directory=data\_pump\_dir dumpfile=hr1.dmp

host expdp scott/scott@orcl directory=data\_pump\_dir dumpfile=scott.dmp







Host impdp hr/hr@orcl directory=data\_pump\_dir dumpfile=scott.dmp tables={scott.emp,scott.dept} REMAP\_SCHEMA=scott:h

Gặp lỗi ORA-31631: privileges are required

ORA-39122: Unprivileged users may not perform REMAP\_SCHEMA remappings.

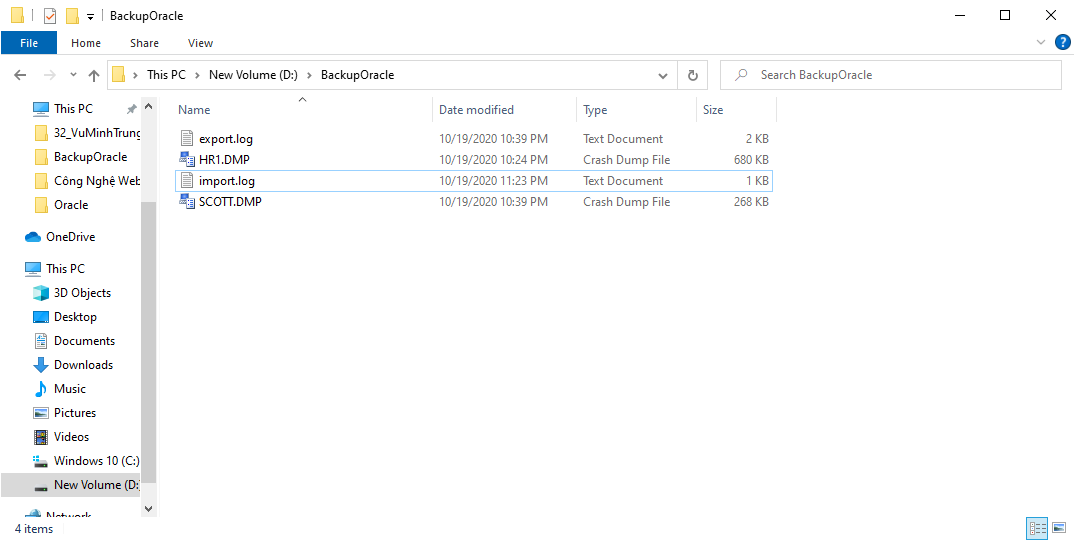
Dùng lệnh

grant import full database to hr;

host impdp hr/hr@orcl directory=data\_pump\_dir dumpfile=scott.dmp tables={scott.emp,scott.dept} REMAP\_SCHEMA=scott:h

sẽ gặp lỗi vì không tìm thấy bảng trong scott (chưa tạo )

tuy gặp lỗi nhưng file import vẫn được tạo.



Các Câu Trigger Trong CSDL

CREATE or replace TRIGGER transfer\_to\_passive after DELETE ON Employee

FOR EACH ROW

BEGIN

INSERT into employee\_inactive(empid,FNAME,LNAME,GENDER,EMPTYPE,STREET,CITY,DATE\_OF\_JOINING,DATE\_OF\_LEAVING,EMAIL)

values (:OLD.EMPID,:OLD.FNAME,:OLD.LNAME,:OLD.GENDER,:OLD.EMPTYPE,:OLD.STREET,:OLD.CITY,:OLD.DATE\_OF\_JOINING,sysdate,:OLD.EMAIL);

END;

CREATE or replace TRIGGER on\_insertemployee\_update\_dept

AFTER INSERT ON employee

for EACH ROW

declare

ab int;

BEGIN

SELECT dept\_headid into ab FROM department WHERE department.deptid=:NEW.DEPTID;

UPDATE department

SET dept\_headid=CASE

WHEN (ab is NULL AND department.deptid=:NEW.deptid)

THEN :NEW.empid

else dept\_headid

END;

END;

CREATE or replace TRIGGER transfer\_to\_prev\_department

For UPDATE of deptid

ON employee

compound trigger

new\_dept\_headid employee.deptid%type;

old\_dept\_headid employee.deptid%type;

gb employee.empid%type;

mn date;

mm int;

after each row is

BEGIN

IF :NEW.deptid<>:OLD.deptid THEN

INSERT into prev\_department values(:OLD.empid,:OLD.deptid,:OLD.since,CURRENT\_TIMESTAMP);

--- dưới đây bị lỗi khi cập nhập trưởng phòng là rỗng vì khi nhân viên bị xóa đi là trưởng phòng sẽ lấy nhân viên khác

/\*SELECT dept\_headid into new\_dept\_headid FROM department WHERE department.deptid=:NEW.deptid;

UPDATE department SET dept\_headid=CASE

WHEN (new\_dept\_headid is NULL)

THEN :NEW.empid

else dept\_headid

END;

SELECT dept\_headid into old\_dept\_headid FROM department WHERE department.deptid=:OLD.deptid;

IF :OLD.empid=old\_dept\_headid THEN

SELECT empid into gb FROM employee WHERE deptid=:OLD.deptid;

SELECT MIN(since)into mn FROM employee WHERE empid=gb;

SELECT MIN(empid) into mm FROM employee WHERE employee.since=mn;

UPDATE department SET dept\_headid=mm WHERE department.deptid=:OLD.deptid;

END IF;\*/

END IF;

END after each row;

end transfer\_to\_prev\_department;

CREATE or replace TRIGGER employee\_on\_delete

for DELETE ON employee

compound trigger

bb department.dept\_headid%type;

mn date;

mm int;

bien int;

after each row is

BEGIN

INSERT into prev\_department values(:OLD.empid,:OLD.deptid,:OLD.since,sysdate);

SELECT dept\_headid into bb FROM department WHERE department.deptid=:OLD.deptid;

-- select count(\*) into bien from employee group by deptid ;

--IF (:OLD.empid=bb) and bien > 1 THEN

-- SELECT MIN(SINCE) into mn FROM employee WHERE empid in (select empid from employee WHERE deptid=bb );

-- SELECT MIN(empid) into mm FROM employee WHERE employee.since=mn;

-- UPDATE department SET dept\_headid=mm WHERE department.deptid=:OLD.deptid;

-- Else

-- UPDATE department SET dept\_headid=null WHERE department.deptid=:OLD.deptid;

-- END IF;

END after each row;

end employee\_on\_delete;

save point ff1;

rollback ff1;

/

Câu Truy Vấn lồng :

select R\_2.NID

from

(select e.deptid as "dep"

from doan2.patient,doan2.employee e,doan2.department d

where doc\_id=e.empid and e.deptid=d.deptid and pid='8')R\_1,

(select NID,deptid

from doan2.NURSE\_ASSIGNED n,doan2.employee e

where nid=empid)R\_2

where R\_1."dep"=R\_2.deptid ;

select NID

from doan2.NURSE\_ASSIGNED ,

(select empid from employee where deptid in

(select deptid from department where Dname='SURGERY'))R\_1

where NURSE\_ASSIGNED.NID=R\_1.empid;

PROCEDURE GAN QUYEN CHO THANH VIEN

create or replace procedure grant\_role(

v\_username varchar2,

v\_role varchar,

v\_stmt varchar2)

IS

begin

v\_stmt := 'grant ' ||v\_role || ' to ' || v\_username ;

Execute immediate v\_stmt;

end grant\_role;

HÀM CẮT CHỮ CÁI

CREATE OR REPLACE FUNCTION FC\_CATCHUCAI

(p\_hoten IN VARCHAR ) return VARCHAR

IS

p\_i int :=1 ;

p\_chucai varchar(30) ;

p\_chuoi varchar(50):='';

p\_chucaidautien varchar(50):=SUBSTR(p\_hoten,1,1);

BEGIN

While (p\_i <=LENGTH(p\_hoten)) LOOP

--- lấy ra vị trí khoảng trống trong chuỗi đưa vào

IF (p\_i=INSTR(p\_hoten,' ',p\_i))

then

p\_chucai:=SUBSTRB(p\_hoten,p\_i+1,1); -- tính theo kiểu substring byte

---nếu substr thôi thì tính theo kiểu chuỗi SubStr('',start,end)--

p\_chuoi:=p\_chuoi||p\_chucai ;

END IF;

p\_i:=p\_i+1 ;

END LOOP;

return p\_chucaidautien||p\_chuoi;

END FC\_CATCHUCAI ;

select FC\_CATCHUCAI('Nguyen van An') from dual;

HÀM NHẬN ROLE

create or replace FUNCTION id\_role (emp\_type in varchar) return boolean

is

begin

if ( RTRIM(emp\_type) = 'DOCTOR')

then

return true;

end if ;

end id\_role ;

----------------------------------Tạo Spec PACKAGE ---------------------------------

------------Trong đây có 2 hàm public còn 1 thủ tục chứa trong thân sẽ ko hiển thị ra specs

CREATE OR REPLACE PACKAGE CREATE\_ACCOUNT\_pkg IS

FUNCTION fc\_username (p\_hoten IN VARCHAR) return VARCHAR;

FUNCTION id\_role (emp\_type IN VARCHAR) return BOOLEAN;

end create\_account\_pkg ;

----------------------------------TẠO PACKAGE BODY -------------------------------

/

CREATE OR REPLACE PACKAGE BODY CREATE\_ACCOUNT\_pkg IS

FUNCTION fc\_username (p\_hoten IN VARCHAR) return VARCHAR

AS

p\_i int :=1 ;

p\_chucai varchar(30);

p\_chuoi varchar(50):='';

p\_chucaidautien varchar(50):=SUBSTR(p\_hoten,1,1);

BEGIN

While (p\_i <= LENGTH(p\_hoten)) LOOP

IF (p\_i = INSTR(p\_hoten,' ',p\_i) )

then

p\_chucai:=SUBSTRB(p\_hoten,p\_i+1,1);

p\_chuoi:=p\_chuoi||p\_chucai ;

END IF;

p\_i:=p\_i+1 ;

END LOOP;

return p\_chucaidautien||p\_chuoi;

END fc\_username;

FUNCTION id\_role (emp\_type in varchar) return boolean

is

begin

if ( RTRIM(emp\_type) = 'DOCTOR')

then

return true;

end if ;

end id\_role ;

procedure create\_grant\_role(v\_fname IN varchar2 ,v\_lname IN varchar2,v\_emptype IN varchar2)

is

v\_username varchar2(50);

v\_stmt varchar2(30);

v\_role varchar2(30):='BOSS';

begin

if (id\_role(v\_emptype) <> TRUE )

then v\_role :='WORKER';

end if;

v\_username:=RTRIM(v\_fname)||fc\_username(v\_lname);

v\_stmt := 'grant ' ||v\_role || ' to ' || v\_username ;

Execute immediate v\_stmt;

end create\_grant\_role;

end create\_account\_pkg;