

VIRTUAL PRIVATE DATABASE (VPD)

https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_RLS.html#GUID-1E528A51-DE53-4961-8770-C53924E427CC

The **DBMS_RLS** package contains the fine-grained access control administrative interface, which is used to implement Virtual Private Database (VPD).

Parameter	Description
object_schema	Schema containing the table, view, or synonym. If no object_schema is specified or is NULL, then the current schema is used.
object_name	Name of table, view, or synonym to which the policy is added.
policy_name	Name of policy to be added. It must be unique for the same table or view. Do not enter special characters such as spaces or commas. If you want to use special characters for the policy name, then enclose the name in quotation marks.
function_schema	Schema owning the policy function. If no function_schema is specified or is NULL, then the current schema is used.
policy_function	Name of a function which generates a predicate for the policy. If the function is defined within a package, then the name of the package must be present.
statement_types	Statement types to which the policy applies. It can be any combination of INDEX, SELECT, UPDATE, or DELETE. The default is to apply to all of these types except INSERT and INDEX .
update_check	Optional argument for the INSERT or UPDATE statement type. The default is FALSE. If you plan to use the INSERT statement

Parameter	Description
	<p>type, then you must set update_check to TRUE. Otherwise, an ORA-28104 input value for string is not valid error is generated.</p> <p>The check applies only to the security relevant columns that are included in the policy definition. In other words, the INSERT or UPDATE operation will fail only if the security relevant column that is defined in the policy is added or updated in the INSERT or UPDATE statement.</p>
enable	<p>Indicates if the policy is enabled when it is added. The default is TRUE.</p>
static_policy	<p>The default is FALSE. If it is set to TRUE, the server assumes that the policy function for the static policy produces the same predicate string for anyone accessing the object, except for SYS or the privileged user who has the EXEMPT ACCESS POLICY privilege.</p>
policy_type	<p>Default is NULL, which means policy_type is decided by the value of static_policy. The available policy types are listed in Table 148-5. Specifying any of these policy types overrides the value of static_policy.</p>
long_predicate	<p>Default is FALSE, which means the policy function can return a predicate with a length of up to 4000 bytes. TRUE means the predicate text string length can be up to 32K bytes. Policies existing prior to the availability of this parameter retain a 32K limit.</p>
sec_relevant_cols	<p>Enables column-level Virtual Private Database (VPD), which enforces security policies when a column containing sensitive information is referenced in a query. Applies to tables and views, but not to synonyms. Specify a list of comma- or space-separated valid column names of the policy-protected object. The policy is enforced only if a specified column is referenced (or, for an abstract datatype</p>

Parameter	Description
	column, its attributes are referenced) in the user SQL statement or its underlying view definition. Default is all the user-defined columns for the object.
sec_relevant_cols_opt	Use with sec_relevant_cols to display all rows for column-level VPD filtered queries (SELECT only), but where sensitive columns appear as NULL. Default is set to NULL, which allows the filtering defined with sec_relevant_cols to take effect. Set to dbms_rls.ALL_ROWS to display all rows, but with sensitive column values, which are filtered by sec_relevant_cols, displayed as NULL. See Usage Notes for restrictions and additional information about this option.
namespace	Name which determines the application context namespace
attribute	Attribute which determines the application context attribute name

1. Bài tập 1

1.1. Tạo cấu trúc bảng sau:

EmpNo	Name	Holiday
1	Han	2/1/2010
2	An	12/5/2010
3	Thu	26/8/2009

EMPHOLIDAY

(

EmpNo NUMBER (5),

Name VARCHAR2 (60),

Holiday DATE

)

```

1 | --connect SCOTT-
2 | create table EMPHOLIDAY
3 | (
4 |   EmpNo NUMBER (5) primary key,
5 |   Name VARCHAR2 (60),
6 |   Holiday DATE
7 |

```

Script Output X | Task completed in 0.11 seconds

Table EMPHOLIDAY created.

Hình: 1

```

17 | insert into empholiday values(1, 'Han', '2/1/2010');
18 | insert into empholiday values(2, 'An', '15/5/2010');
19 | insert into empholiday values(3, 'Thu', '26/8/2009');


```

Script Output X | Task completed in 0.11 seconds

1 row inserted.

1 row inserted.

1 row inserted.

Hình: 2

Xây dựng một policy HolidayControl cho các trường hợp sau đây:

- Giả thiết trường Name có các dữ liệu phân biệt nhau và tương ứng với Username của các account trong hệ thống.
- Schema chứa đối tượng cần được bảo vệ là SCOTT.
- Schema chứa policy function được sử dụng là SEC_MGR
 - Tạo User SEC_MGR và gán quyền chạy package DBMS_RLS cho User để thao tác trên Policy Function

```

--connect SYS--
create user SEC_MGR identified by 123;
grant all privileges to SEC_MGR;
grant EXECUTE ON DBMS_RLS to SEC_MGR;

```

Hình: 3

- Chuẩn bị các thủ tục dùng để: Select, Insert, Update, Delete
- Thủ tục Select

```

--connect SCOTT--
--Select--
create or replace procedure Pro_Select_EMPHOLIDAY(cur out sys_refcursor)
is
begin
    open cur for
        select * from empholiday;
end;

```

Hình: 4

- Thủ tục Insert

```

--Insert--
create or replace procedure Pro_Insert_EMPHOLIDAY(EmpNo_in in number, Name_in in varchar2,
                                                    Holiday_in in varchar2, ex out number)
is
begin
    insert into empholiday values(EmpNo_in, Name_in, TO_DATE(Holiday_in,'DD-MM-YYYY'));
    ex := SQL%ROWCOUNT;
end;

```

Hình: 5

- Thủ tục Update

```

--Update--
create or replace procedure Pro_Update_EMPHOLIDAY(EmpNo_in in number, Name_in in varchar2,
                                                    Holiday_in in varchar2, ex out number)
is
begin
    update empholiday set name = name_in, holiday = TO_DATE(Holiday_in,'DD-MM-YYYY') where empno = empno_in;
    ex := SQL%ROWCOUNT;
end;

```

Hình: 6

- Thủ tục Delete

```

--Delete--
create or replace procedure Pro_Delete_EMPHOLIDAY(EmpNo_in in number, ex out number)
is
begin
    delete empholiday where empno = empno_in;
    ex := SQL%ROWCOUNT;
end;

```

Hình: 7

1.2. An chỉ được xem và chỉnh sửa (update) thông tin cá nhân của riêng mình.

1.2.1. Tạo function fun_HolidayControl1, fun_HolidayControl2

– fun_HolidayControl1 trả về điều kiện của An khi Select, Update

```

29 --connect SEC_MGR--
30 create or replace function fun_HolidayControl1(p_schema in varchar2, p_object in varchar2)
31 return varchar2
32 is
33 begin
34     if user = 'AN' then
35         return 'Name = ''An'''';
36     end if;
37 end;

```

Function FUN_HOLIDAYCONTROL1 compiled

Hình: 8

- fun_HolidayControl1 trả về điều kiện luôn sai khi An Insert, Delete

```

172 --connect SEC_MGR--
173 create or replace function fun_HolidayControl2(p_schema in varchar2, p_object in varchar2)
174 return varchar2
175 is
176 begin
177     if user = 'AN' then
178         return '1=0';
179     end if;
180 end;
181

```

Function FUN_HOLIDAYCONTROL2 compiled

Hình: 9

1.2.2. Tạo chính sách bảo mật cho An có tên HolidayControl1, HolidayControl2

```

221 --Connect SEC_MGR--
222 -- The DBMS_RLS package contains the fine-grained access control administrative interface,
223 -- which is used to implement Virtual Private Database (VPD).
224 begin
225     dbms_rls.add_policy
226     (
227         object_schema => 'SCOTT',
228         object_name => 'EMPHOLIDAY',
229         policy_name => 'HolidayControl1',
230         policy_function => 'fun_HolidayControl1',
231         statement_types => 'SELECT, UPDATE',
232         update_check => TRUE
233     );
234 end;

```

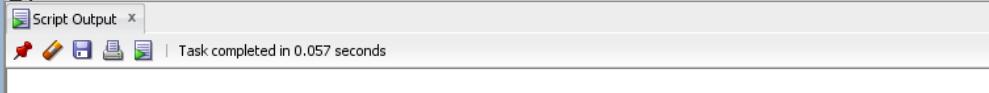
PL/SQL procedure successfully completed.

Hình: 10

```

236 begin
237     dbms_rls.add_policy
238     (
239         object_schema => 'SCOTT',
240         object_name => 'EMPHOLIDAY',
241         policy_name => 'HolidayControl2',
242         policy_function => 'fun_HolidayControl2',
243         statement_types => 'INSERT, DELETE',
244         update_check => TRUE
245     );
246 end;
247

```



Hình: 11

1.2.3. Đăng nhập vào User An

```
--connect SYS--
create user An identified by 123;
grant create session to An;
grant select, insert, update, delete on scott.empholiday to An;
grant execute on scott.Pro_Select_EMPHOLIDAY to An;
grant execute on scott.Pro_Insert_EMPHOLIDAY to An;
grant execute on scott.Pro_Update_EMPHOLIDAY to An;
grant execute on scott.Pro_Delete_EMPHOLIDAY to An;
```

Hình: 12

- An chỉ được truy vấn dữ liệu của mình dù An không đặt thêm điều kiện where trong câu truy vấn

```

51 --connect AN--
52 select * from SCOTT.empholiday;
53

```

EMPNO	NAME	HOLIDAY
2	An	15/05/2010

- An thêm một dòng dữ liệu

Hình: 13

```

156 | insert into scott.empholiday values(4, 'Hien', '15/5/2010');
      |
      | Script Output x | Query Result x
      | Task completed in 0.126 seconds
      |
      | Error starting at line : 156 in command -
      | insert into scott.empholiday values(4, 'Hien', '15/5/2010')
      | Error report -
      | ORA-28115: policy with check option violation

```

Hình: 14

- An xóa một dòng dữ liệu

```

159 | delete SCOTT.empholiday where EmpNo = 1;
160 |
      |
      | Script Output x | Query Result x
      | Task completed in 0.051 seconds
      |
      | 0 rows deleted.

```

Hình: 15

- An chỉ được sửa dữ liệu của mình cho dù An có dùng lệnh cập nhật không có điều kiện

```

54 | update SCOTT.empholiday set holiday = '1/1/2020';
55 |
      |
      | Script Output x | Query Result x
      | Task completed in 0.075 seconds
      |
      | 1 row updated.

```

Hình: 16

1.2.4. Hướng dẫn tạo form thêm vào xóa Policy Function

- Tạo Package chứa các thủ tục truy vấn liên quan đến User, Policy

```

Worksheet | Query Builder
1 | --connect SYS-
2 |@create or replace package pkg_Policy_Select
3 |as
4 |  --1--
5 |  procedure pro_select_user(cur out sys_refcursor);
6 |  --2--
7 |  procedure pro_select_table(username in varchar2, cur out sys_refcursor);
8 |  --3--
9 |  procedure pro_select_columns(useschema in varchar2, tablename in varchar2, cur out sys_refcursor);
10 |  --4--
11 |  procedure pro_select_procedure_user(useowner in varchar2, pro_type in varchar2, cur out sys_refcursor);
12 |  --5--
13 |  procedure pro_select_policy(pf_owner_us in varchar2, cur out sys_refcursor);
14 |end;

```

Hình: 17

– Tạo thân Package:

- Thủ tục truy vấn User

```
16 | create or replace package body pkg_Policy_Select
17 | as
18 |   --1--
19 |   procedure pro_select_user(cur out sys_refcursor)
20 |   is
21 |   begin
22 |     open cur for
23 |       select username from dba_users where account_status = 'OPEN';
24 |   end;
```

Hình: 18

- Thủ tục truy vấn Table theo Scema

```
25 |   --2--
26 |   procedure pro_select_table(username in varchar2, cur out sys_refcursor)
27 |   is
28 |   begin
29 |     open cur for
30 |       select table_name from dba_all_tables where owner = username;
31 |   end;
```

Hình: 19

- Thủ tục truy vấn Column của Table

```
32 |   --3--
33 |   procedure pro_select_columns(useschema in varchar2, tablename in varchar2, cur out sys_refcursor)
34 |   is
35 |   begin
36 |     open cur for
37 |       select column_name from dba_tab_columns where owner = useschema and table_name = tablename;
38 |   end;
```

Hình: 20

- Thủ tục truy vấn Function

```
39 |   --4--
40 |   procedure pro_select_procedure_user(useowner in varchar2, pro_type in varchar2, cur out sys_refcursor)
41 |   is
42 |   begin
43 |     open cur for
44 |       select object_name from dba_procedures where owner = useowner and object_type = pro_type;
45 |   end;
```

Hình: 21

- Thủ tục truy vấn Policy của Schema

```
46 |   --5--
47 |   procedure pro_select_policy(pf_owner_us in varchar2, cur out sys_refcursor)
48 |   is
49 |   begin
50 |     open cur for
51 |       select object_owner, object_name, policy_name, function, sel, ins, upd, del, chk_option from dba_policies where pf_owner = pf_owner_us;
52 |   end;
53 | end;
```

Hình: 22

- Gán quyền Execute Package vào User: SEC_MGR

```

54 |
55 | grant execute on pkg_Policy_Select to SEC_MGR;

```

Hình: 23

- Tạo thủ tục Add Policy

```

106 --connect SEC_MGR-
107 create or replace procedure Pro_addPolicy(object_schema in varchar2, object_name in varchar2,
108                                         policy_name in varchar2, policy_function in varchar2,
109                                         statement_types in varchar2, sec_relevant_cols in varchar2 default null)
110 is
111 begin
112     dbms_rls.add_policy
113     (
114         object_schema => object_schema,
115         object_name => object_name,
116         policy_name => policy_name,
117         policy_function => policy_function,
118         statement_types => statement_types,
119         sec_relevant_cols => sec_relevant_cols,
120         update_check => TRUE
121     );
122 end;

```

Hình: 24

- Tạo thủ tục Drop Policy

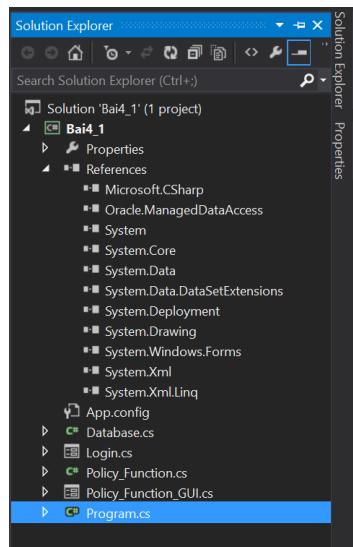
```

124 create or replace procedure Pro_dropPolicy(object_schema in varchar2, object_name in varchar2, policy_name in varchar2)
125 is
126 begin
127     DBMS_RLS.drop_policy
128     (
129         object_schema => object_schema,
130         object_name => object_name,
131         policy_name => policy_name
132     );
133 end;

```

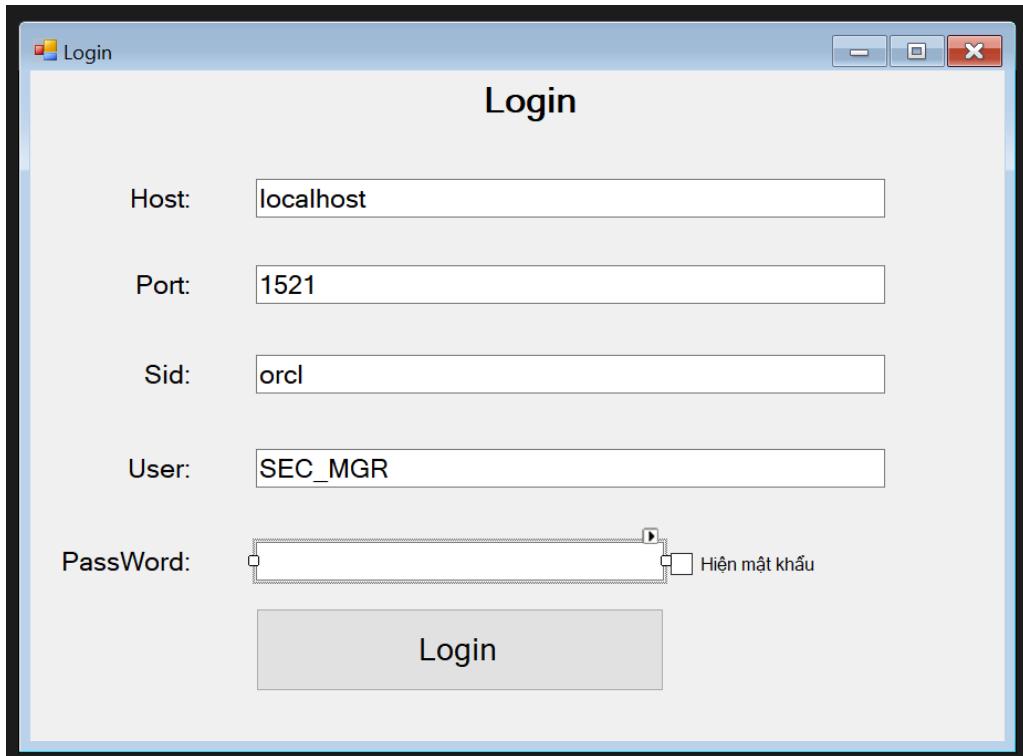
Hình: 25

- Tạo Project Bai4_1



Hình: 26

- Sử dụng lại Class Database và form Login của các bài trước



Hình: 27

- Chính sửa lại code của nút Login chuyển hướng đến form Policy_Function

```

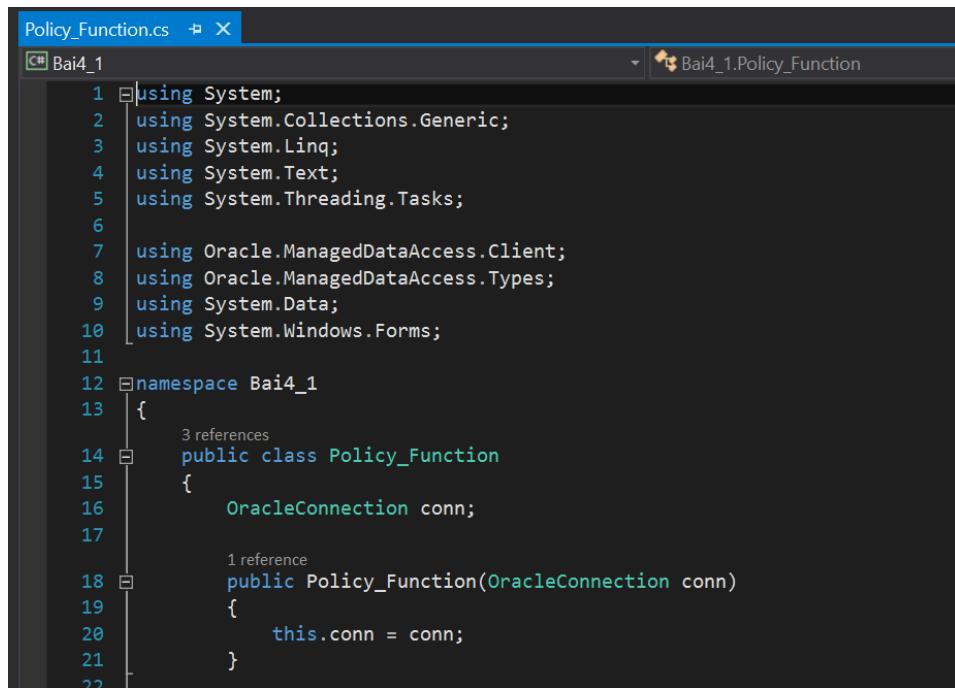
private void btn_login_Click(object sender, EventArgs e)
{
    string host = txt_host.Text;
    string port = txt_port.Text;
    string sid = txt_sid.Text;
    string user = txt_user.Text;
    string pass = txt_password.Text;

    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 Policy_Function_GUI().Show();
            this.Hide();
        }
        else
        {
            Check_Status(user);
            return;
        }
    }
}

```

Hình: 28

- Tạo Class Policy_Function để gọi chạy các thủ tục trên



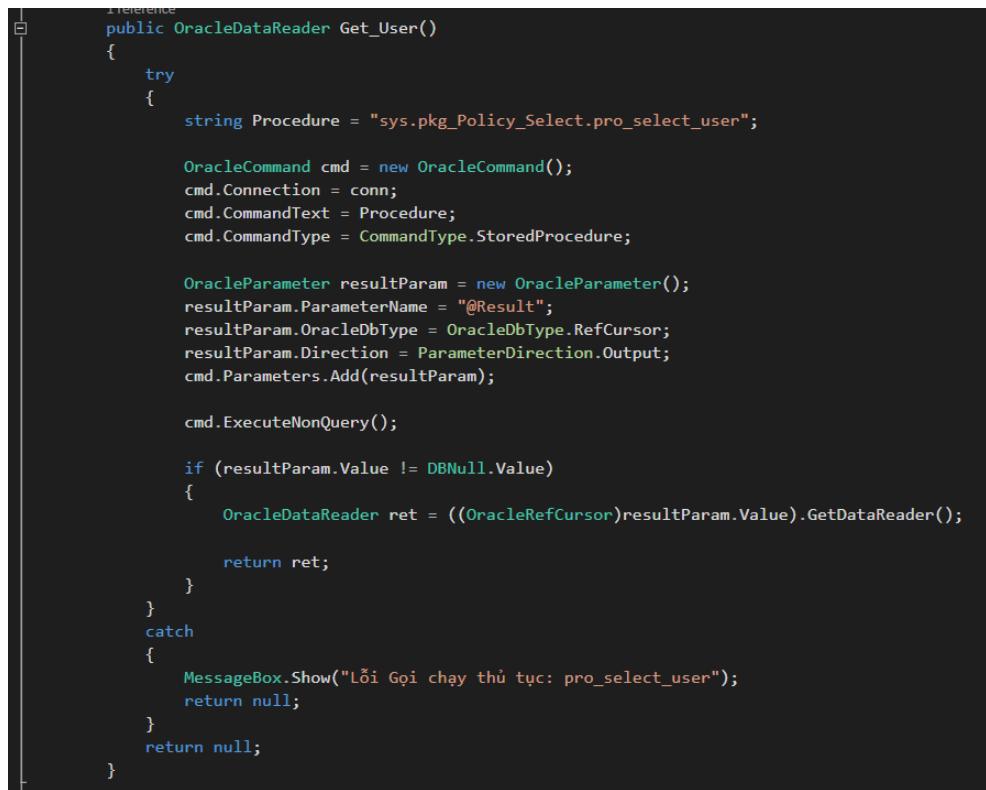
```

Policy_Function.cs  ✘ X
Bai4_1
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 Bai4_1
13 {
14     public class Policy_Function
15     {
16         OracleConnection conn;
17
18         public Policy_Function(OracleConnection conn)
19         {
20             this.conn = conn;
21         }
22     }

```

Hình: 29

- Hàm gọi chạy thủ tục truy vấn User



```

public OracleDataReader Get_User()
{
    try
    {
        string Procedure = "sys.pkg_Policy_Select.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: 30

- Hàm gọi chạy thủ tục truy vấn Function

```
public OracleDataReader Get_Function_User(string userowner, string type)
{
    try
    {
        string Procedure = "sys.pkg_Policy_Select.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);

        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: 31

- Hàm gọi chạy thủ tục truy vấn Table

```
public OracleDataReader Get_Table_User(string userowner)
{
    try
    {
        string Procedure = "sys.pkg_Policy_Select.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();

        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: 32

- Hàm gọi chạy thủ tục truy vấn Column của Table

```
public DataTable Get_Table_Column(string usershchema, string tablename)
{
    try
    {
        string Procedure = "sys.pkg_Policy_Select.pro_select_columns";

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

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

        OracleParameter TableName = new OracleParameter();
        TableName.ParameterName = "@tablename";
        TableName.OracleDbType = OracleDbType.Varchar2;
        TableName.Value = tablename.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();
            DataTable data = new DataTable();
            data.Load(ret);
            return data;
        }
    }
    catch
    {
        MessageBox.Show("Lỗi Gọi chạy thủ tục: pro_select_columns");
        return null;
    }
    return null;
}
```

Hình: 33

- Hàm gọi chạy thủ tục truy vấn Policy

```
public DataTable Get_Policy(string PF_owner)
{
    try
    {
        string Procedure = "sys.pkg_Policy_Select.pro_select_policy";

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

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

        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_policy");
        return null;
    }
    return null;
}
```

Hình: 34

- Hàm gọi chạy thủ tục Add Policy

```

public bool Pro_Add_Policy(string object_schema, string object_name,
    string policy_name, string policy_function, string statement_types, string sec_relevant_cols)
{
    try
    {
        string Procedure = "Pro_addPolicy";

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

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

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

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

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

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

        if (!sec_relevant_cols.Equals(""))
        {
            OracleParameter SecRelevantCols = new OracleParameter();
            SecRelevantCols.ParameterName = "@sec_relevant_cols";
            SecRelevantCols.OracleDbType = OracleDbType.Varchar2;
            SecRelevantCols.Value = sec_relevant_cols;
            SecRelevantCols.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(SecRelevantCols);
        }

        cmd.ExecuteNonQuery();

        return true;
    }
    catch
    {
        MessageBox.Show("Lỗi Gọi chạy thủ tục: Pro_addPolicy\n Hãy kiểm tra lại hàm hoặc tên Policy!!");
        return false;
    }
}

```

Hình: 35

- Hàm gọi chạy thủ tục Drop Policy

```
    public bool Pro_Drop_Policy(string object_schema, string object_name, string policy_name)
    {
        try
        {
            string Procedure = "Pro_dropPolicy";

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

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

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

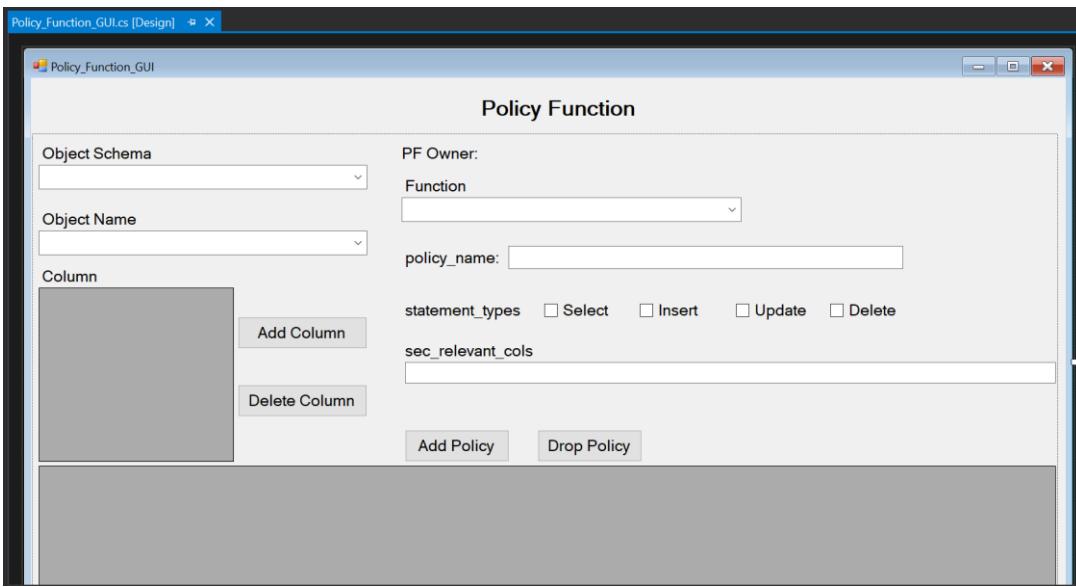
            OracleParameter PolicyName = new OracleParameter();
            PolicyName.ParameterName = "@policy_name";
            PolicyName.OracleDbType = OracleDbType.Varchar2;
            PolicyName.Value = policy_name;
            PolicyName.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(PolicyName);

            cmd.ExecuteNonQuery();

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

Hình: 36

- Tạo Form Policy_Function_GUI thiết kế giao diện quản lý Policy như sau



Hình: 37

- Khai báo thư viện và các biến cần thiết

```

10
11  using Oracle.ManagedDataAccess.Client;
12  using Oracle.ManagedDataAccess.Types;
13
14  namespace Bai4_1
15  {
16      public partial class Policy_Function_GUI : Form
17      {
18          OracleConnection conn;
19          Policy_Function pf;
20          string pf_owner;
21
22          List<string> column = new List<string>();
23          public Policy_Function_GUI()
24          {
25              InitializeComponent();
26              CenterToScreen();
27              conn = Database.Get_Connect();
28              pf = new Policy_Function(conn);
29              pf_owner = Database.User;
30
31              Load_Combobox_User();
32              Load_Combobox_function(pf_owner);
33              Load_Policy(pf_owner);
34              lb_PF.Text += pf_owner;
35          }
36

```

Hình: 38

- Hàm Load dữ liệu từ hàm truy vấn lên Combobox

```

    void Load_Combobox_User()
    {
        OracleDataReader read = pf.Get_User();
        if (read == null) return;
        while (read.Read())
        {
            cmb_object_schema.Items.Add(read[0].ToString());
        }
        read.Close();
        cmb_object_schema.SelectedIndex = 0;
    }

```

Hình: 39

- Hàm Load Policy vào DataGridView

```

    void Load_Policy(string pf_owner)
    {
        dtg_policy.DataSource = pf.Get_Policy(pf_owner);
    }

```

Hình: 40

- Hàm Load Combobox Function

```

    void Load_Combobox_function(string PF_Owner)
    {
        OracleDataReader read = pf.Get_Function_User(PF_Owner, "FUNCTION");
        if (read == null) return;
        while (read.Read())
        {
            cmb_function.Items.Add(read[0].ToString());
        }
        read.Close();
        cmb_function.SelectedIndex = 0;
    }

```

Hình: 41

- Hàm Load Combobox Table theo Schema

```

    void Load_Table_User(string user)
    {
        cmb_object_name.Items.Clear();
        OracleDataReader read = pf.Get_Table_User(user);
        if (read == null) return;
        while (read.Read())
        {
            cmb_object_name.Items.Add(read[0].ToString());
        }
        read.Close();

        if (cmb_object_name.Items.Count == 0)
            cmb_object_name.Items.Add("");
        cmb_object_name.SelectedIndex = 0;
    }

```

Hình: 42

- Hàm Load Column theo Table

```
    1 reference
    void Load_Table_Column(string UserSchema, string TableName)
    {
        dtg_column.DataSource = pf.Get_Table_Column(UserSchema, TableName);
    }
```

Hình: 43

- Hàm Add Column: ghi column được thêm vào textbox

```
    1 reference
    string Add_Column(int row)
    {
        string col = dtg_column.Rows[row].Cells[0].Value.ToString();

        if(!column.Any(item => item == col))
            column.Add(col);

        string sec_relevant_cols = "";
        foreach (string c in column)
        {
            if (sec_relevant_cols.Equals(""))
            {
                sec_relevant_cols += c;
            }
            else
            {
                sec_relevant_cols += " , " + c;
            }
        }

        return sec_relevant_cols;
    }
```

Hình: 44

- Hàm xóa column khỏi textbox

```
    1 reference
    string Delete_Column(int row)
    {
        string col = dtg_column.Rows[row].Cells[0].Value.ToString();

        if (column.Any(item => item == col))
            column.Remove(col);

        string sec_relevant_cols = "";
        foreach (string c in column)
        {
            if (sec_relevant_cols.Equals(""))
            {
                sec_relevant_cols += c;
            }
            else
            {
                sec_relevant_cols += " , " + c;
            }
        }

        return sec_relevant_cols;
    }
```

Hình: 45

- Hàm lấy statement types được áp dụng trong Policy

```
    string Get_statement_types()
    {
        List<string> statement_types_list = new List<string>();
        if (cb_select.Checked)
            statement_types_list.Add("SELECT");
        if (cb_insert.Checked)
            statement_types_list.Add("INSERT");
        if (cb_update.Checked)
            statement_types_list.Add("UPDATE");
        if (cb_delete.Checked)
            statement_types_list.Add("DELETE");

        string statement_types = "";
        int i = 0;
        foreach (string t in statement_types_list)
        {
            if (i == 0)
            {
                statement_types += t;
            }
            else
            {
                statement_types += ", ";
                statement_types += t;
            }
            i++;
        }
        return statement_types;
    }
```

Hình: 46

- Sự kiện chọn combobox object owner (user)

```
    private void cmb_object_owner_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (cmb_object_schema.SelectedItem != null)
        {
            string user = cmb_object_schema.SelectedItem.ToString();
            Load_Table_User(user);
        }
    }
```

Hình: 47

- Sự kiện chọn combobox object name (table)

```
1 reference
private void cmb_object_name_SelectedIndexChanged(object sender, EventArgs e)
{
    if (cmb_object_name.SelectedItem != null)
    {
        string UserSchema = cmb_object_schema.SelectedItem.ToString();
        string TableName = cmb_object_name.SelectedItem.ToString();

        Load_Table_Column(UserSchema, TableName);

        column.Clear();
        txt_sec_relevant_cols.Text = "";
    }
}
```

Hình: 48

- Sự kiện click button: Add Column và Delete Column

```
1 reference
private void btn_addcolumn_Click(object sender, EventArgs e)
{
    if (dtg_column.CurrentCell != null)
    {
        int row = dtg_column.CurrentCell.RowIndex;
        txt_sec_relevant_cols.Text = Add_Column(row);
    }
}

1 reference
private void btn_deletecolumn_Click(object sender, EventArgs e)
{
    if (dtg_column.CurrentCell != null)
    {
        int row = dtg_column.CurrentCell.RowIndex;
        txt_sec_relevant_cols.Text = Delete_Column(row);
    }
}
```

Hình: 49

– Nút Add Policy

```
private void btn_add_policy_Click(object sender, EventArgs e)
{
    string statement_types = Get_statement_types();
    string object_schema = cmb_object_schema.SelectedItem.ToString();
    string object_name = cmb_object_name.SelectedItem.ToString();
    string policy_name = txt_policy_name.Text;
    string policy_function = cmb_function.SelectedItem.ToString();
    string sec_relevant_cols = txt_sec_relevant_cols.Text;
    if (statement_types.Equals(""))
    {
        MessageBox.Show("Bạn chưa chọn điều kiện cho Policy Function!!!");
        return;
    }
    if (object_name.Equals(""))
    {
        MessageBox.Show("Mục object_name trống!!!");
        return;
    }
    if (policy_name.Equals(""))
    {
        MessageBox.Show("Chưa nhập tên Policy!!"); txt_policy_name.Focus();
        return;
    }
    if (pf.Pro_Add_Policy(object_schema, object_name, policy_name, policy_function, statement_types, sec_relevant_cols))
    {
        MessageBox.Show("Tạo Policy Function thành công");
        Load_Policy(pf_owner);
    }
}
```

Hình: 50

– Nút Drop Policy

```
private void btn_drop_policy_Click(object sender, EventArgs e)
{
    int row = dtg_policy.CurrentCell.RowIndex;

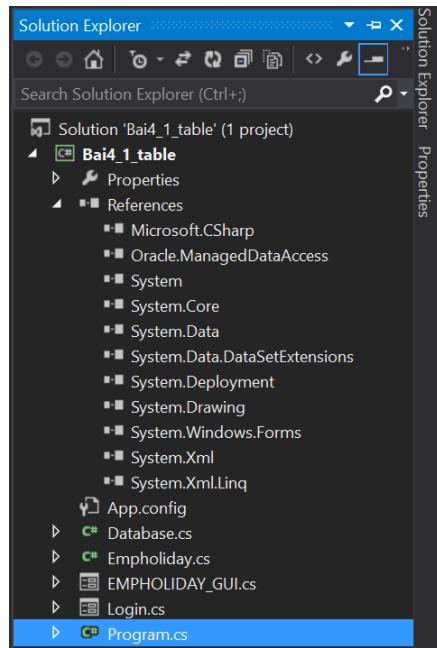
    string object_schema = dtg_policy.Rows[row].Cells[0].Value.ToString();
    string object_name = dtg_policy.Rows[row].Cells[1].Value.ToString();
    string policy_name = dtg_policy.Rows[row].Cells[2].Value.ToString();

    if (pf.Pro_Drop_Policy(object_schema, object_name, policy_name))
    {
        MessageBox.Show("Xóa Policy Function thành công");
        Load_Policy(pf_owner);
    }
}
```

Hình: 51

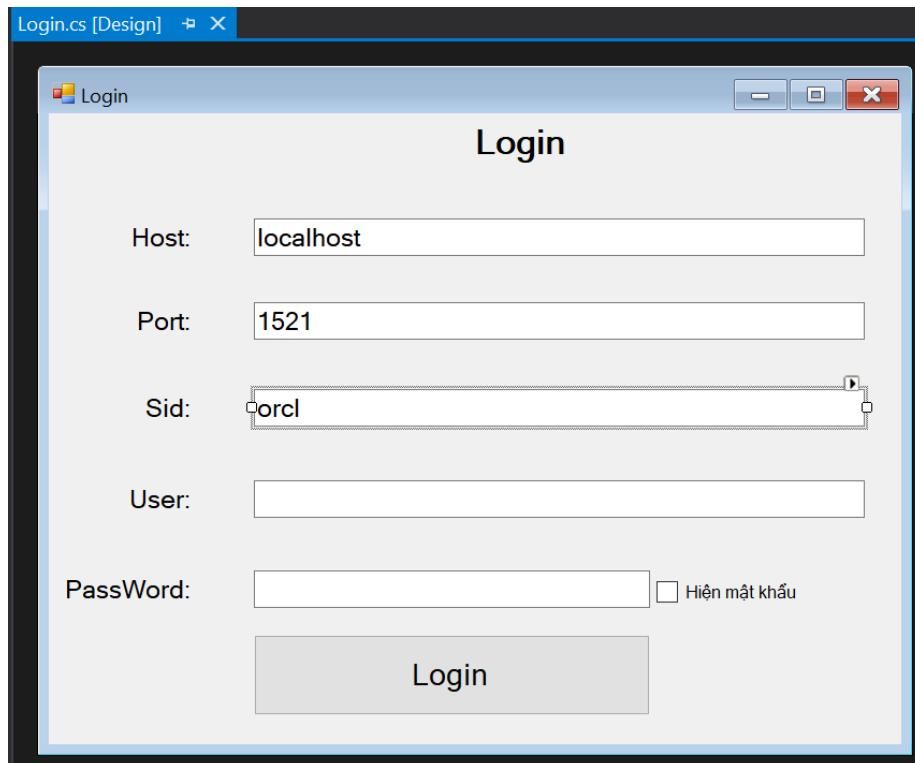
1.2.5. Hướng dẫn tạo form thao tác trên bảng Empholiday

- Tạo Project Bai4_1_table



Hình: 52

- Sử dụng lại Class Database và Form Login



Hình: 53

- Chính sửa lại nút Login

```

    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 EMPHOLIDAY_GUI().Show();
                this.Hide();
            }
            else
            {
                Check_Status(user);
                return;
            }
        }
    }
}

```

Hình: 54

- Tạo Class Empholiday

```

Empholiday.cs  ✎ X
C#\Bai4_1_table
Empholiday.cs  ✎ Bai4_1_table.Empholiday  ✎ Select_Empholid
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 Bai4_1_table
13 {
14     public class Empholiday
15     {
16         OracleConnection conn;
17
18         public Empholiday(OracleConnection conn)
19         {
20             this.conn = conn;
21             OracleCommand cmd = new OracleCommand("ALTER SESSION SET NLS_DATE_FORMAT = '\'DD-MM-YYYY\'", conn);
22             cmd.ExecuteNonQuery();
23         }
24     }
}

```

Hình: 55

- Viết hàm gọi chạy thủ tục Select

```
public DataTable Select_Empholiday()
{
    try
    {
        string Procedure = "scott.pro_select_Empholiday";

        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 (OracleException ex)
    {
        MessageBox.Show("Truy vấn thất bại: " + ex.Message);
        return null;
    }
    return null;
}
```

Hình: 56

- Viết hàm gọi chạy thủ tục Insert

```
1 reference
public int Insert_Empholiday(int empno_in, string name_in, string holiday_in)
{
    try
    {
        string Procedure = "scott.pro_insert_Empholiday";

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

        OracleParameter empno = new OracleParameter();
        empno.ParameterName = "@empno_in";
        empno.OracleDbType = OracleDbType.Int32;
        empno.Value = empno_in;
        empno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(empno);

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

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

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

        cmd.ExecuteNonQuery();

        if (int.Parse(ex.Value.ToString()) == 0)
        {
            MessageBox.Show("0 dòng được thêm");
            return -1;
        }

        return 0;
    }
    catch(OracleException ex)
    {
        MessageBox.Show("Thêm thất bại: " + ex.Message);
        return -1;
    }
}
```

Hình: 57

- Viết hàm gọi chạy thủ tục Update

```

    public int Update_Empholiday(int empno_in, string name_in, string holiday_in)
    {
        try
        {
            string Procedure = "scott.pro_update_Empholiday";

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

            OracleParameter empno = new OracleParameter();
            empno.ParameterName = "@empno_in";
            empno.OracleDbType = OracleDbType.Int32;
            empno.Value = empno_in;
            empno.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(empno);

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

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

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

            cmd.ExecuteNonQuery();

            if (int.Parse(ex.Value.ToString()) == 0)
            {
                MessageBox.Show("0 dòng được cập nhật");
                return -1;
            }

            return 0;
        }
        catch(OracleException ex)
        {
            MessageBox.Show("Sửa thất bại: " + ex.Message);
            return -1;
        }
    }
}

```

Hình: 58

- Viết hàm gọi chạy thủ tục Delete

```
    public int Delete_Empholiday(int empno_in)
    {
        try
        {
            string Procedure = "scott.pro_delete_Empholiday";

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

            OracleParameter empno = new OracleParameter();
            empno.ParameterName = "@empno_in";
            empno.OracleDbType = OracleDbType.Int32;
            empno.Value = empno_in;
            empno.Direction = ParameterDirection.Input;
            cmd.Parameters.Add(empno);

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

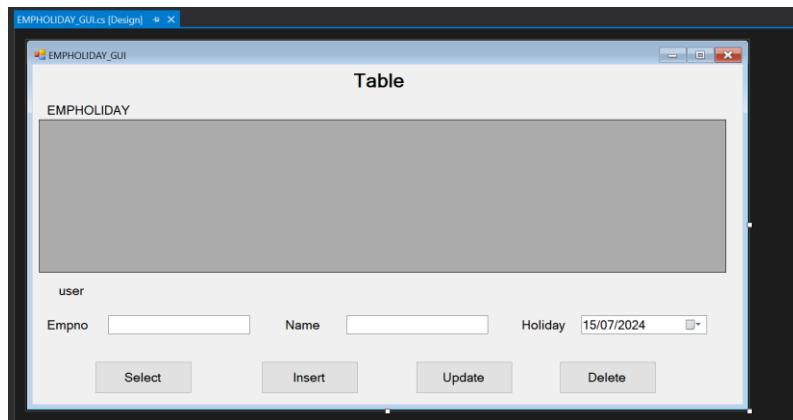
            cmd.ExecuteNonQuery();

            if (int.Parse(ex.Value.ToString()) == 0)
            {
                MessageBox.Show("0 dòng được xóa");
                return -1;
            }

            return 0;
        }
        catch(OracleException ex)
        {
            MessageBox.Show("Xóa thất bại:" + ex.Message);
            return -1;
        }
    }
}
```

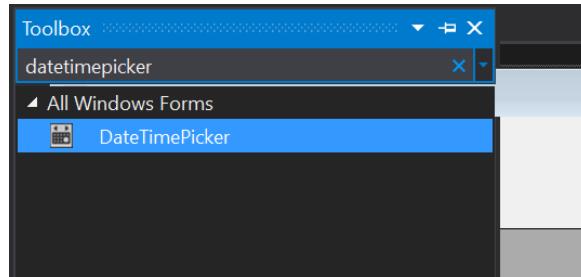
Hình: 59

- Tạo Form Empholiday GUI



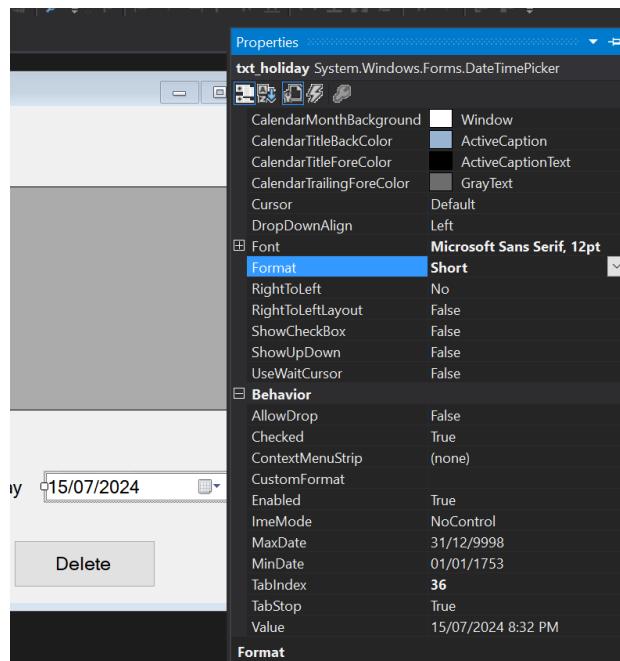
Hình: 60

- Sử dụng DateTimePicker để chọn dữ liệu ngày/tháng/năm cho cột Holiday



Hình: 61

- Chọn Format thành Short



Hình: 62

- Khai báo thư viện và biến

```

EMPHOLIDAY_GUI.cs  X  EMPHOLIDAY_GUI.cs [Design]
Bai4_1_table
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 using Oracle.ManagedDataAccess.Client;
11
12
13 namespace Bai4_1_table
14 {
15
16     public partial class EMPHOLIDAY_GUI : Form
17     {
18         OracleConnection conn;
19         Empholiday empholiday;
20         public EMPHOLIDAY_GUI()
21         {
22             InitializeComponent();
23             CenterToScreen();
24
25             conn = Database.Get_Connect();
26             empholiday = new Empholiday(conn);
27             lb_user.Text = Database.User;
28         }
29     }

```

Hình: 63

- Thiết lập sự kiện chọn dòng dữ liệu trong bảng

```

private void dtg_table_SelectionChanged(object sender, EventArgs e)
{
    int row = dtg_table.CurrentCell.RowIndex;

    txt_empno.Text = dtg_table.Rows[row].Cells[0].Value.ToString();
    txt_name.Text = dtg_table.Rows[row].Cells[1].Value.ToString();
    txt_holiday.Text = dtg_table.Rows[row].Cells[2].Value.ToString();
}

```

Hình: 64

- Thiết lập nút Select

```

private void btn_select_Click(object sender, EventArgs e)
{
    dtg_table.DataSource = empholiday.Select_Empholiday();
}

```

Hình: 65

– Thiết lập nút Insert

```
1 reference
private void btn_insert_Click(object sender, EventArgs e)
{
    int empno = int.Parse(txt_empno.Text);
    string name = txt_name.Text;
    string holiday = txt_holiday.Text;

    int i = empholiday.Insert_Empholiday(empno, name, holiday);
    if (i != -2)
    {
        MessageBox.Show("Thêm thành công");
    }
}
```

Hình: 66

– Thiết lập nút Update

```
1 reference
private void btn_update_Click(object sender, EventArgs e)
{
    int empno = int.Parse(txt_empno.Text);
    string name = txt_name.Text;
    string holiday = txt_holiday.Text;

    int i = empholiday.Update_Empholiday(empno, name, holiday);
    if (i != -2)
    {
        MessageBox.Show("Sửa thành công");
    }
}
```

Hình: 67

– Thiết lập nút Delete

```
1 reference
private void btn_delete_Click(object sender, EventArgs e)
{
    int empno = int.Parse(txt_empno.Text);

    int i = empholiday.Delete_Empholiday(empno);
    if (i != -2)
    {
        MessageBox.Show("Xóa thành công");
    }
}
```

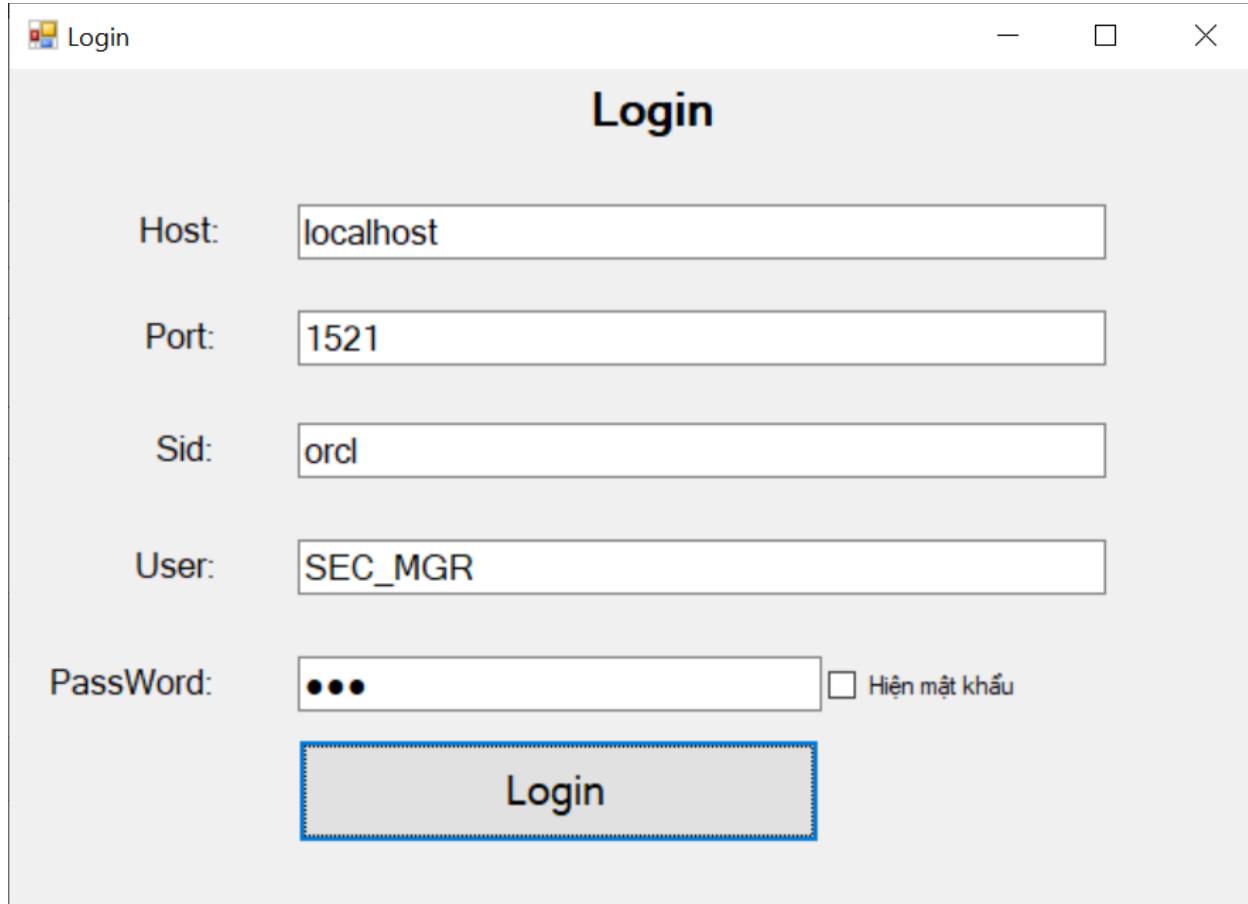
Hình: 68

- Sự kiện đóng form

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

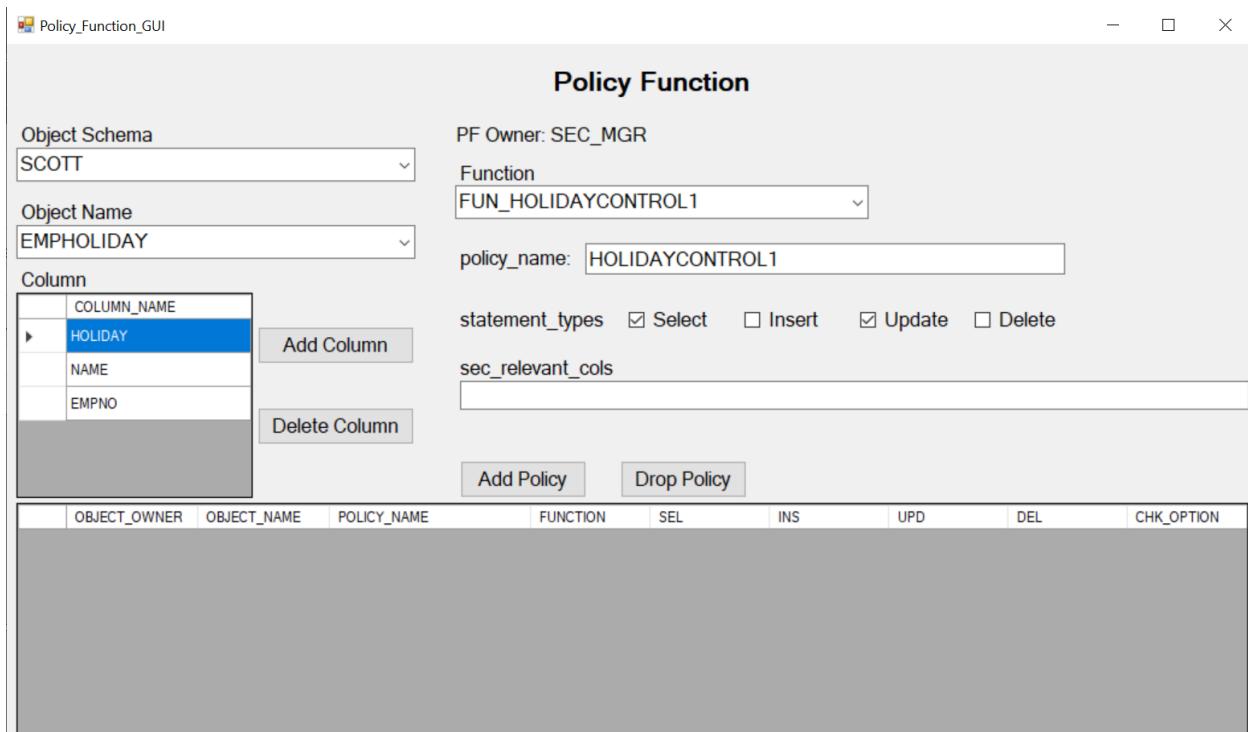
Hình: 69

- Chạy Bai4_1 để tạo Policy Function. Đăng nhập bằng user SEC_MGR



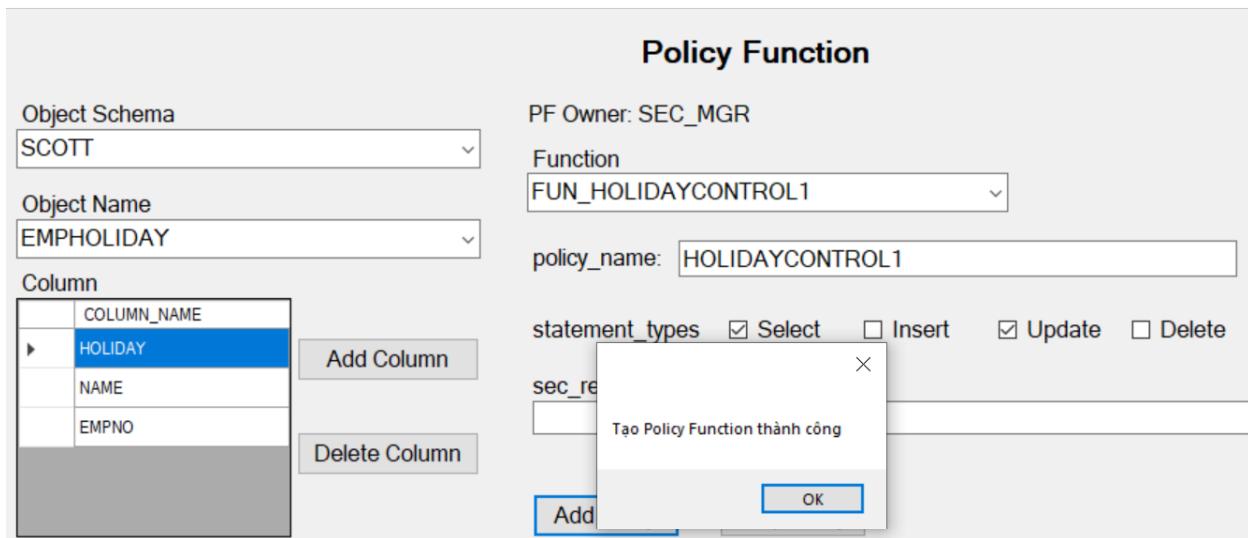
Hình: 70

- Chọn các Schema, Table, Function và đặt tên cho Policy như hình



Hình: 71

Nhấn vào nút Add Policy để thêm một Policy mới. Nếu thêm thành công sẽ nhận thông báo như hình



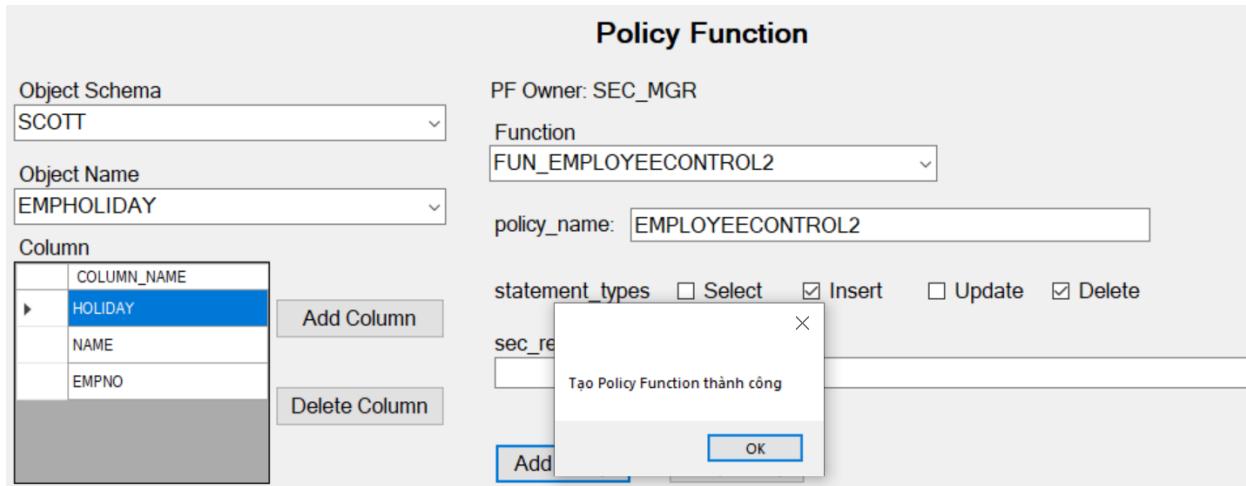
Hình: 72

- Sau khi tạo thành công Policy được truy vấn lên bảng

	OBJECT_OWNER	OBJECT_NAME	POLICY_NAME	FUNCTION	SEL	INS	UPD	DEL	CHK_OPTION
▶	SCOTT	EMPHOLIDAY	HOLIDAYCONTRO...	FUN_HOLIDAYCONTRO...	YES	NO	YES	NO	YES

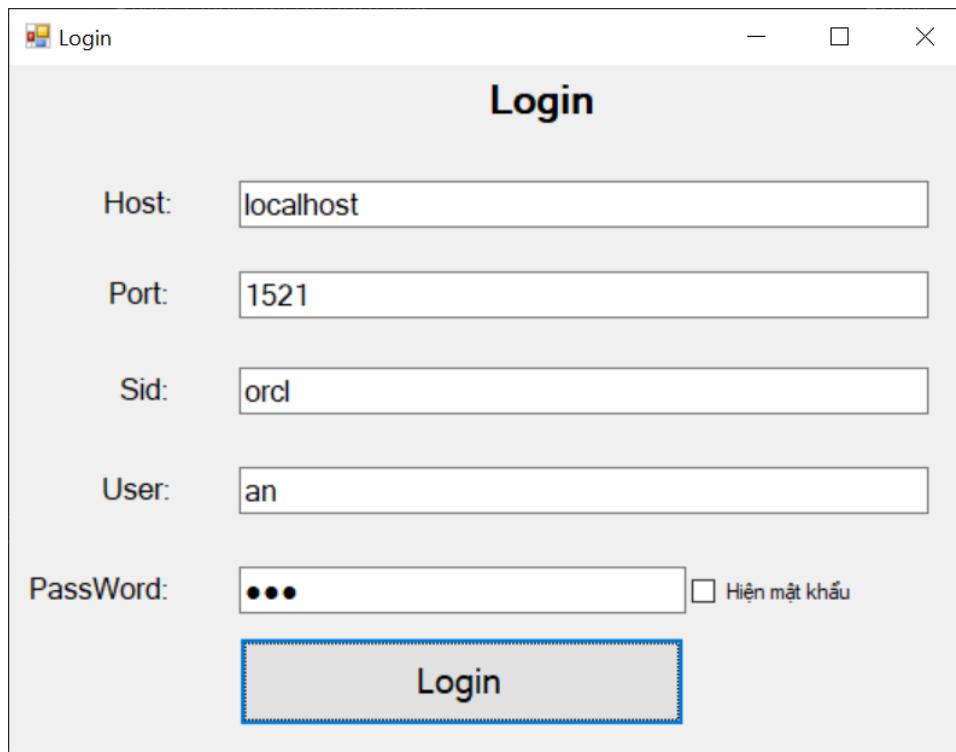
Hình: 73

- Thêm tiếp Policy Holidaycontrol2



Hình: 74

- Chạy Bai4_1_table để test Policy vừa tạo. Đăng nhập bằng user An



Hình: 75

- Giao diện sau khi đăng nhập thành công

The screenshot shows a Windows application window titled "EMPHOLIDAY_GUI". The main title bar has standard minimize, maximize, and close buttons. Below the title bar, the word "Table" is centered in a light gray header area. Underneath this, the word "EMPHOLIDAY" is displayed in blue. The main content area is a large gray rectangular box. At the bottom of the window, there is a row of buttons and input fields. On the left, the word "an" is typed. To the right of "an" are three input fields: "Empno" with a text box containing "2", "Name" with a text box containing "An", and "Holiday" with a date picker set to "16/07/2024". Below these fields are four buttons: "Select", "Insert", "Update", and "Delete".

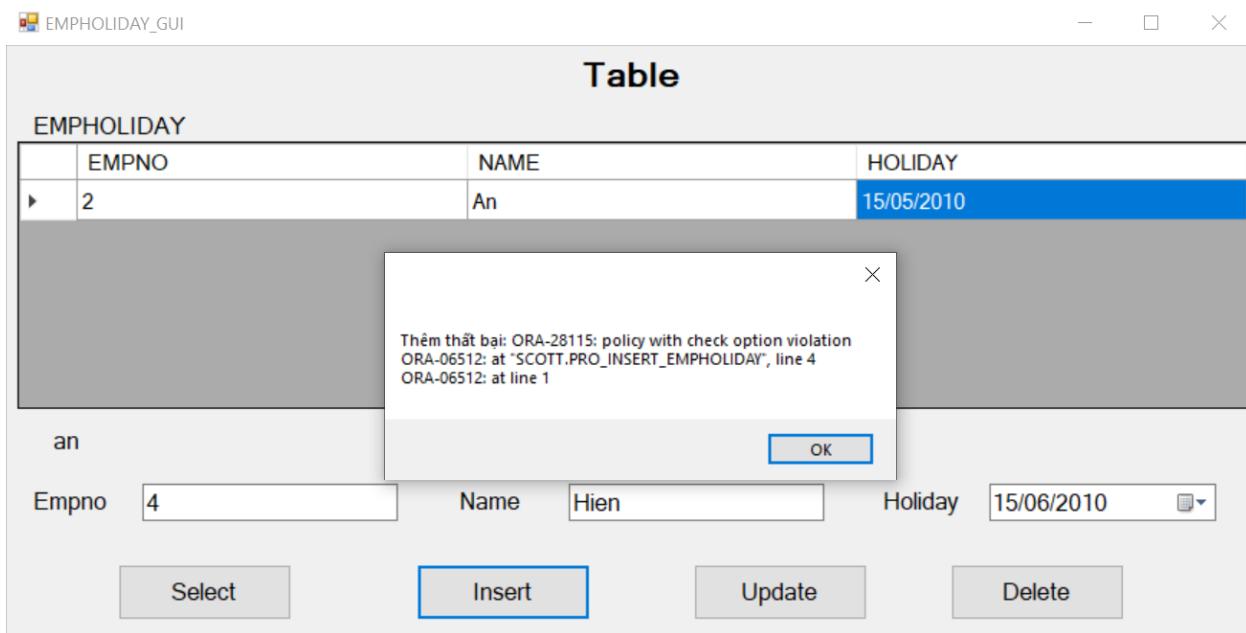
Hình: 76

- An thực hiện truy vấn dữ liệu chỉ thấy được dữ liệu của mình

This screenshot shows the same application window as the previous one, but now it displays a single row of data in the table. The table has columns labeled "EMPNO", "NAME", and "HOLIDAY". The first row contains the values "2", "An", and "15/05/2010". The "EMPNO" cell is highlighted with a blue background. The rest of the interface is identical to the previous screenshot, with the word "an" at the top, input fields for Empno, Name, and Holiday, and buttons for Select, Insert, Update, and Delete.

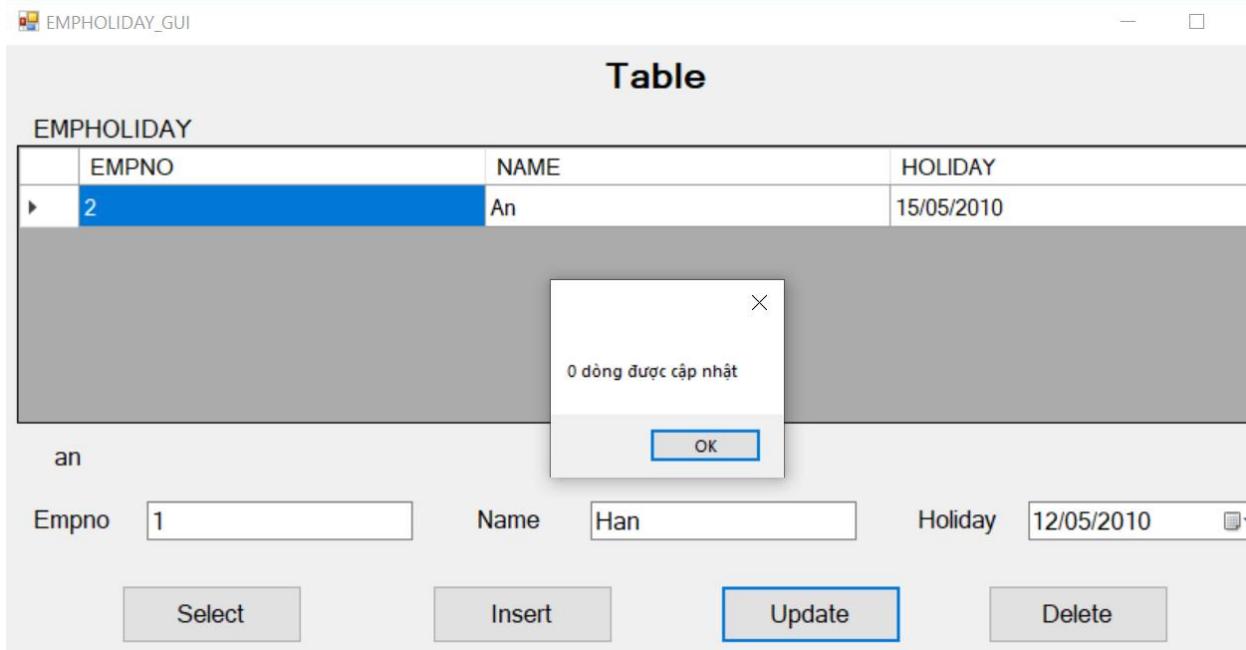
Hình: 77

- Thêm một dòng dữ liệu mới. Kết quả thất bại



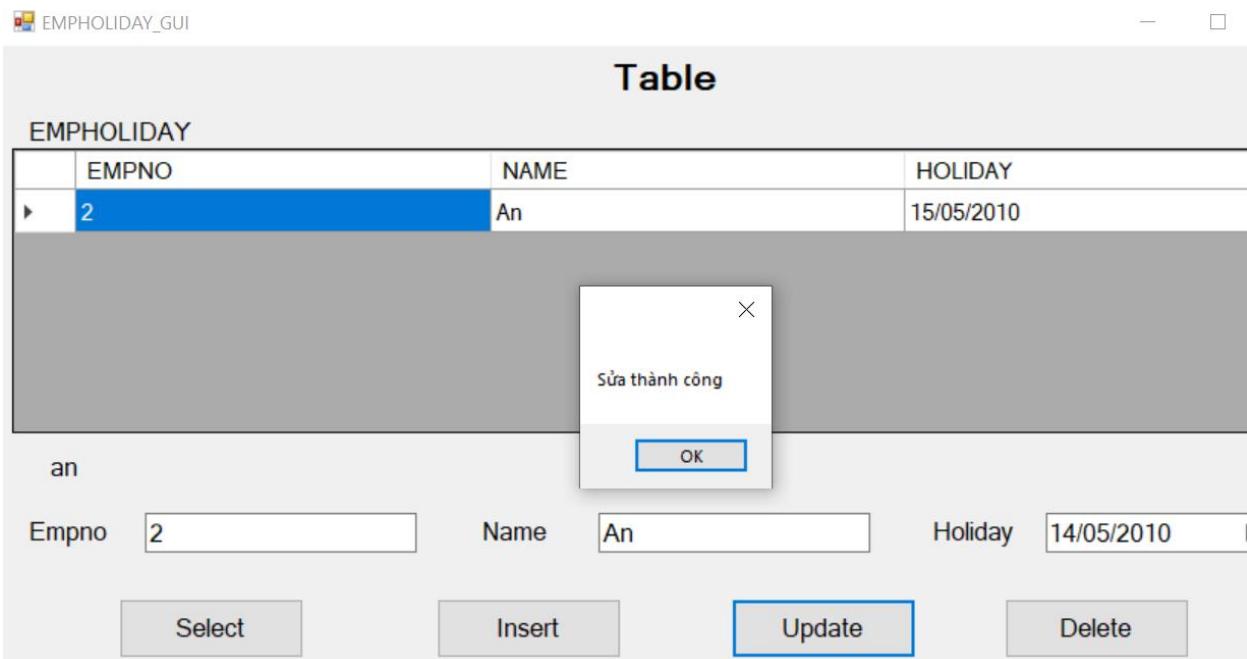
Hình: 78

- An sửa dữ liệu của người khác: không cập nhật được



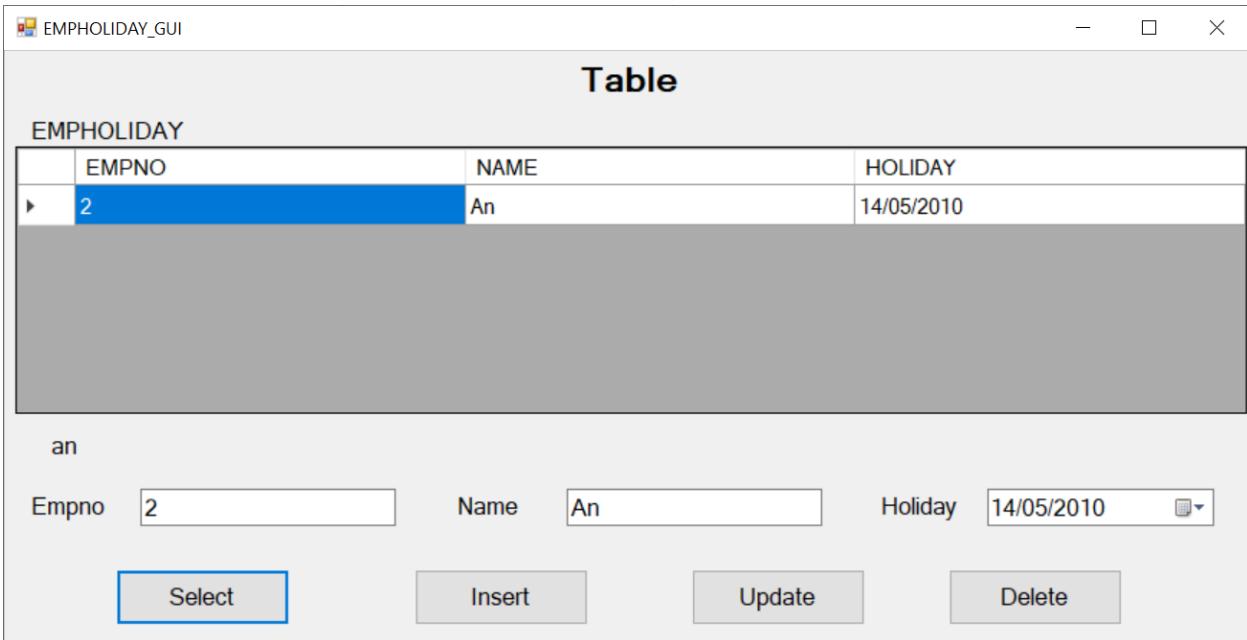
Hình: 79

- An sửa thành công dữ liệu của chính mình



Hình: 80

- Kết quả sau khi sửa



Hình: 81

- An không được xóa được dữ liệu

Table			
EMPHOLIDAY			
	EMPNO	NAME	HOLIDAY
▶	2	An	14/05/2010
X			
0 dòng được xóa			
<input type="button" value="OK"/>			
an <input type="text" value="2"/> Empno <input type="text" value="An"/> Name <input type="text" value="14/05/2010"/> Holiday			
<input type="button" value="Select"/> <input type="button" value="Insert"/> <input type="button" value="Update"/> <input type="button" value="Delete"/>			

Hình: 82

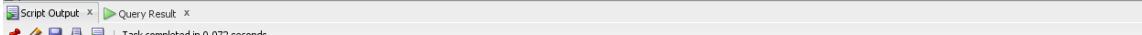
1.3. Thu không được xem hay chỉnh sửa bất kỳ thông tin nào.

- Sửa lại function fun_HolidayControl1 và fun_HolidayControl2 ở câu 1.1 thêm điều kiện của Thu là 1=0 như vậy khi Thu thực hiện bất cứ hành động nào đến bảng EMPHOLIDAY thì điều kiện 1=0 sẽ được thêm vào và điều kiện này không thỏa mãn với bảng nên Thu sẽ không được xem hay chỉnh sửa bất kỳ thông tin nào (hình 83,84)

```

56  --connect SEC_MGR--
57  create or replace function fun_HolidayControl1(p_schema in varchar2, p_object in varchar2)
58  return varchar2
59  is
60  begin
61  if user = 'AN' then
62      return 'Name = ''An'''';
63  elsif user = 'THU' then
64      return '1 = 0';
65  end if;
66 end;

```



Function FUN_HOLIDAYCONTROL1 compiled

Hình: 83

```

272 | create or replace function fun_HolidayControl2(p_schema in varchar2, p_object in varchar2)
273 | return varchar2
274 | is
275 | begin
276 |   if user = 'AN' then
277 |     return '1=0';
278 |   elsif user = 'THU' then
279 |     return '1 = 0';
280 |   end if;
281 | end;

```

Script Output X | Task completed in 0.123 seconds

Function FUN_HOLIDAYCONTROL2 compiled

Hình: 84

- Tiến hành test trên Sql Devlopper
- Tao User Thu

```

--connect SYS
create user Thu identified by 123;
grant create session to Thu;
grant select, insert, update, delete on scott.empholiday to Thu;
grant execute on scott.Pro_Select_EMPHOLIDAY to Thu;
grant execute on scott.Pro_Insert_EMPHOLIDAY to Thu;
grant execute on scott.Pro_Update_EMPHOLIDAY to Thu;
grant execute on scott.Pro_Delete_EMPHOLIDAY to Thu;

```

Hình: 85

- Sử dụng user Thu truy vấn đến bảng Empholiday: không truy vấn được dòng dữ liệu nào

```

68 | --connect Thu--test policy function--
69 | select * from SCOTT.empholiday;
70 |

```

Script Output X | Query Result X | All Rows Fetched: 0 in 0.01 seconds

EMPNO	NAME	HOLIDAY

Hình: 86

- Thu thêm một dòng dữ liệu mới: không được phép

```
256 | insert into scott.empholiday values(4, 'Hien', '15/5/2010');

Query Result x Script Output x
Task completed in 0.037 seconds

Error starting at line : 256 in command -
insert into scott.empholiday values(4, 'Hien', '15/5/2010')
Error report -
ORA-28115: policy with check option violation

Hình: 87

Thu sửa dữ liệu bất kì: không sửa được

258 | update SCOTT.empholiday set holiday = '1/2/2021' where empno = 1;

Query Result x Script Output x
Task completed in 0.053 seconds

0 rows updated.
```

Hình: 87

- Thu sửa dữ liệu bất kì: không sửa được

```
258 update SCOTT.empholiday set holiday = '1/2/2021' where empno = 1;
Query Result | Script Output | Task completed in 0.053 seconds
0 rows updated.

Thu tiên hành xóa dữ liệu: không xóa được

260 delete SCOTT.empholiday where EmpNo = 1;
Query Result | Script Output | Task completed in 0.051 seconds
0 rows deleted.
```

Hình: 88

- Tiến hành test Policy Function trên Form C#
 - Đăng nhập vào User Thu và nhấn nút Select: không có dữ liệu được truy vấn

EMPHOLIDAY

Table

	EMPNO	NAME	HOLIDAY			
Thu						
Empno	<input type="text"/>	Name	<input type="text"/>	Holiday	18/07/2024	<input type="button" value="▼"/>
	<input type="button" value="Select"/>	<input type="button" value="Insert"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>		

Hình: 89

- Thu tiền hành điền dữ liệu và thêm dữ liệu: Thông báo không thêm được

EMPHOLIDAY

EMPNO	NAME	HOLIDAY
Thu		
Empno	4	Name: Hien Holiday: 15/05/2010
<input type="button" value="Select"/> <input style="outline: 2px solid blue; border-radius: 5px; padding: 2px 10px;" type="button" value="Insert"/> <input type="button" value="Update"/> <input type="button" value="Delete"/>		

Thêm thất bại: ORA-28115: policy with check option violation
ORA-06512: at "SCOTT.PRO_INSERT_EMPHOLIDAY", line 4
ORA-06512: at line 1

Hình: 90

- Thu tiền hành sửa dữ liệu: thông báo không có dòng nào được sửa

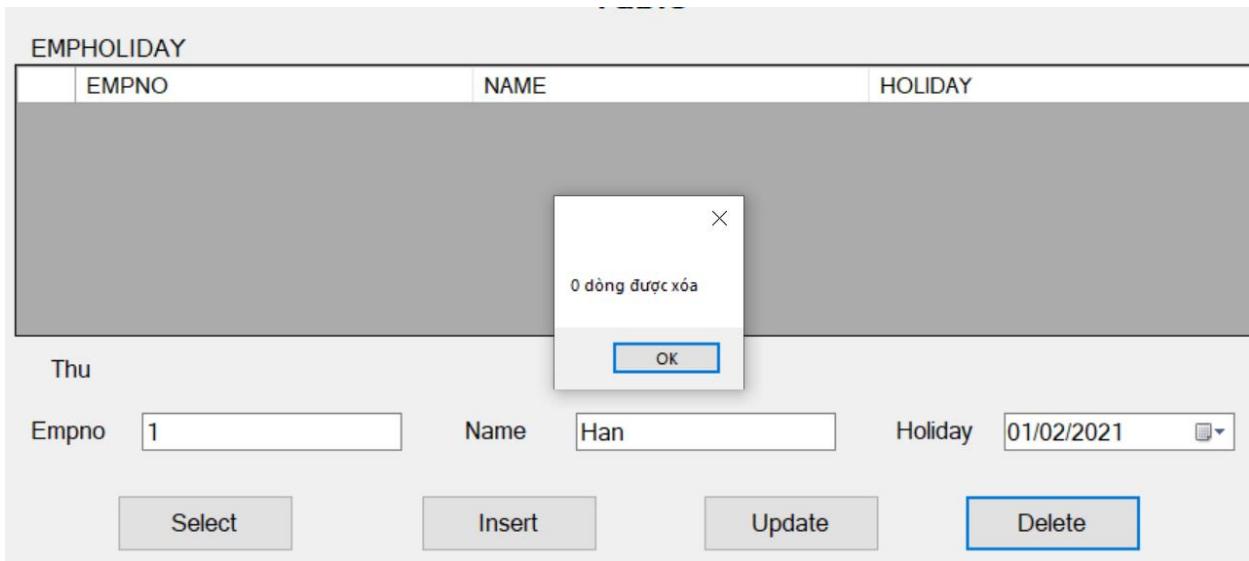
EMPHOLIDAY

EMPNO	NAME	HOLIDAY
Thu		
Empno	1	Name: Han Holiday: 01/02/2021
<input type="button" value="Select"/> <input type="button" value="Insert"/> <input style="outline: 2px solid blue; border-radius: 5px; padding: 2px 10px;" type="button" value="Update"/> <input type="button" value="Delete"/>		

0 dòng được cập nhật

Hình: 91

- Thu tiến hành xóa một dòng dữ liệu: thông báo không có dòng nào được xóa



Hình: 92

1.4. Han được quyền xem tất cả các thông tin nhưng chỉ chỉnh sửa (Insert, Update, Delete) được Holiday nào mà ngày lớn hơn hay bằng ngày hiện tại (không chỉnh sửa được ngày ở quá khứ).

- Chính sửa lại function fun_HolidayControl1 dùng cho chính sách Select

```

73 --connect SEC_MGR--
74 create or replace function fun_HolidayControl1(p_schema in varchar2, p_object in varchar2)
75 return varchar2
76 is
77 begin
78   if user = 'AN' then
79     return 'Name = ''An'''';
80   elsif user = 'THU' then
81     return '1 = 0';
82   elsif user = 'HAN' then
83     return '';
84   end if;
85 end;

```

Script Output X | Query Result X
✓ ✖ ✖ ✖ | Task completed in 0.087 seconds

Function FUN_HOLIDAYCONTROL1 compiled

Hình: 93

- Chính sửa lại function fun_HolidayControl1 cho điều kiện không ảnh hưởng đến Han

```

307 create or replace function fun_HolidayControl2(p_schema in varchar2, p_object in varchar2)
308 return varchar2
309 is
310 begin
311   if user = 'AN' then
312     return '1=0';
313   elsif user = 'THU' then
314     return '1 = 0';
315   elsif user = 'HAN' then
316     return '';
317   end if;
318 end;
319

```

Function FUN_HOLIDAYCONTROL2 compiled

Hình: 94

- Tạo tiếp function cho trường hợp Insert, Update, Delete với điều kiện không ảnh hưởng đến An và Thu và điều kiện của Han là ngày sửa đổi phải lớn hơn hoặc bằng ngày hiện tại

```

345 --connect SEC_MGR--
346 create or replace function fun_HolidayControl3(p_schema in varchar2, p_object in varchar2)
347 return varchar2
348 is
349 begin
350   if user = 'HAN' then
351     return 'Holiday >= to_char(sysdate, ''DD-MM-YYYY'')';
352   else
353     return '';
354   end if;
355 end;
356

```

Function FUN_HOLIDAYCONTROL3 compiled

Hình: 95

- Tạo policy function HolidayControl3 cho Insert, Update, Delete

```

337 --connect SEC_MGR--
338 begin
339   dbms_rls.add_policy
340   (
341     object_schema => 'SCOTT',
342     object_name => 'EMPHOLIDAY',
343     policy_name => 'HolidayControl3',
344     policy_function => 'fun_HolidayControl3',
345     statement_types => 'INSERT, UPDATE, DELETE',
346     update_check => TRUE
347   );
348 end;

```

Script Output | Task completed in 0.044 seconds

PL/SQL procedure successfully completed.

Hình: 96

- Vào user Han truy vấn vào bảng empholiday

```

135 --connect Han--test policy function--
136 select * from SCOTT.empholiday;

```

Script Output | Query Result | All Rows Fetched: 3 in 0.003 seconds

EMPNO	NAME	HOLIDAY
1	Han	01/08/2023
2	An	01/01/2020
3	Thu	26/08/2009

Hình: 97

- Thử trường hợp update dòng dữ liệu cũ và mới đều thỏa điều kiện

```

138 update SCOTT.empholiday set holiday = '2/8/2023' where EmpNo = 1;

```

Script Output | Query Result | Task completed in 0.046 seconds

1 row updated.

Hình: 98

- Trường hợp dòng dữ liệu thỏa điều kiện nhưng dòng cập nhật không đạt điều kiện

```

140 update SCOTT.empholiday set holiday = '1/1/2023' where EmpNo = 1;

```

Script Output | Query Result | Task completed in 0.048 seconds

Error starting at line : 139 in command -
update SCOTT.empholiday set holiday = '1/1/2023' Where EmpNo = 1
Error report -
ORA-28115: policy with check option violation

Hình: 99

- Trường hợp cả dòng dữ liệu cũ và cập nhật đều không thỏa điều kiện

```
142| update SCOTT.empholiday set holiday = '2/8/2023' Where EmpNo = 2;
```

Script Output | Query Result | Task completed in 0.05 seconds
0 rows updated.

Hình: 100

- Đăng nhập User Han trên form

Table			
EMPHOLIDAY			
	EMPNO	NAME	HOLIDAY
▶	1	Han	02/01/2010
	2	An	15/05/2010
	3	Thu	26/08/2009

Han

Empno Name Holiday

Hình: 101

- Thêm dòng dữ liệu mới nhưng với ngày không thỏa điều kiện

Table			
EMPHOLIDAY			
	EMPNO	NAME	HOLIDAY
▶	1	Han	02/01/2010
	2	An	15/05/2010
	3	Thu	26/08/2009
	4	Han	01/07/2024

Thêm thất bại: ORA-01843: not a valid month
ORA-06512: at "SCOTT.PRO_INSERT_EMPHOLIDAY", line 4
ORA-06512: at line 1

OK

Empno Name Holiday

Hình: 102

- Thêm một dòng mới với Holiday thỏa điều kiện

Table

EMPHOLIDAY

	EMPNO	NAME	HOLIDAY
▶	1	Han	02/01/2010
	2	An	15/05/2010
	3	Thu	26/08/2009

Thêm thành công

OK

Empno	<input type="text" value="4"/>	Name	<input type="text" value="Ngoc"/>
Holiday	<input type="text" value="20/07/2024"/>		

Select
Insert
Update
Delete

Hình: 103

- Thực hiện truy vấn lại

EMPHOLIDAY

	EMPNO	NAME	HOLIDAY
▶	1	Han	02/01/2010
	2	An	15/05/2010
	3	Thu	26/08/2009
	4	Ngoc	20/07/2024

Han

1

Han

02/01/2010

Select
Insert
Update
Delete

Hình: 104

- Cập nhật với dữ liệu cũ không thỏa điều kiện

The screenshot shows a database table named 'EMPHOLIDAY' with columns 'EMPNO', 'NAME', and 'HOLIDAY'. A row with Empno 3, Name 'Thu', and Holiday '26/08/2009' is selected. An update dialog box is open, containing the message '0 dòng được cập nhật' (0 rows updated) and an 'OK' button. Below the table, there are input fields for 'Empno' (3), 'Name' (Thu), and 'Holiday' (20/07/2024). At the bottom are buttons for 'Select', 'Insert', 'Update' (highlighted in blue), and 'Delete'.

Hình: 105

- Cập nhật với dữ liệu mới không thỏa điều kiện

The screenshot shows the same 'EMPHOLIDAY' table. A row with Empno 4, Name 'Ngoc', and Holiday '16/07/2024' is selected. An update dialog box is open, displaying an error message: 'Sửa thất bại: ORA-28115: policy with check option violation ORA-06512: at "SCOTT.PRO_UPDATE_EMPHOLIDAY", line 4 ORA-06512: at line 1'. Below the table, there are input fields for 'Empno' (4), 'Name' (Ngoc), and 'Holiday' (16/07/2024). At the bottom are buttons for 'Select', 'Insert', 'Update' (highlighted in blue), and 'Delete'.

Hình: 106

- Cập nhật thành công khi cả dữ liệu mới và cũ đều thỏa điều kiện

EMPHOLIDAY

	EMPNO	NAME	HOLIDAY
1	Han	02/01/2010	
2	An	15/05/2010	
3	Thu	26/08/2009	
▶ 4	Ngoc	20/07/2024	

Sửa thành công

Han

Empno Name Holiday

Hình: 107

- Xóa dòng dữ liệu không thỏa điều kiện

EMPHOLIDAY

	EMPNO	NAME	HOLIDAY
1	Han	02/01/2010	
2	An	15/05/2010	
▶ 3	Thu	26/08/2009	
4	Ngoc	19/07/2024	

0 dòng được xóa

Han

Empno Name Holiday

Hình: 108

- Xóa dòng dữ liệu thỏa điều kiện

EMPHOLIDAY			
	EMPNO	NAME	HOLIDAY
	1	Han	02/01/2010
	2	An	15/05/2010
	3	Thu	26/08/2009
▶	4	Ngoc	19/07/2024

Han

Empno Name Holiday

Select Insert Update Delete

Hình: 109

Như vậy ta cần tạo 2 policy function để giải quyết được yêu cầu của đề bài

2. Bài tập 2

Hoàn thiện chính sách bảo mật ở bài số 1 (HolidayControl) để đảm bảo khi exception xảy ra sẽ không bộc lộ thông tin nhạy cảm của chính sách bảo mật này.

- Trường hợp nếu user không có trong chính sách xuất hiện dòng lỗi được thông báo về policy function thay vì trả về 1 bảng trống

```

1 --connect HR--|
2 select * from SCOTT.empholiday;

```

ORA-28112: Failed to execute policy function
28112. 00000 - "Failed to execute policy function"
*Cause: The policy function has one or more error during execution.
*Action: Check the trace file and correct the errors.
Error at Line: 2 Column: 21

Hình: 110

- Policy function gặp phải lỗi ở trường hợp không có chuỗi trả về hoặc chuỗi trả về ở function gây ra lỗi cho câu lệnh SQL.
- Khắc phục bằng cách thêm else vào khối if..elsif để khi user không thỏa điều kiện muốn truy vấn thì sẽ mặc định là không được (hình 111)

```

4 |--connect SEC_MGR--
5 |create or replace function fun_HolidayControl1(p_schema in varchar2, p_object in varchar2)
6 |return varchar2
7 |
8 |begin
9 |  if user = 'AN' then
10 |    return 'Name = ''An'''';
11 |  elsif user = 'THU' then
12 |    return '1 = 0';
13 |  elsif user = 'HAN' then
14 |    return '';
15 |  else
16 |    return '1 = 0';
17 |  end if;
18 |end;

```

Function FUN_HOLIDAYCONTROL1 compiled

Hình: 111

- Dùng HR để truy vấn lại

```

1 |--connect HR--
2 |select * from SCOTT.empholiday;
3 |

```

Hình: 112

- Cách làm tương tự với policy HolidayControl2

```

20 |create or replace function fun_HolidayControl2(p_schema in varchar2, p_object in varchar2)
21 |return varchar2
22 |
23 |begin
24 |  if user = 'AN' then
25 |    return '1=0';
26 |  elsif user = 'THU' then
27 |    return '1 = 0';
28 |  elsif user = 'HAN' then
29 |    return '';
30 |  else
31 |    return '1 = 0';
32 |  end if;
33 |end;

```

Function FUN_HOLIDAYCONTROL2 compiled

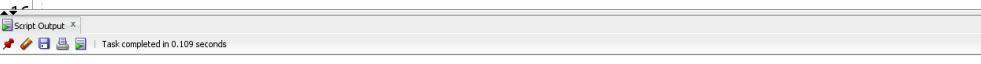
Hình: 113

3. Bài tập 3

Chỉnh sửa lại chính sách bảo mật ở câu 2, cho phép An xem được thông tin EmpNo và Name của các nhân viên khác trong bảng EmpHoliday nhưng chỉ xem được ngày nghỉ của chính mình.

- Do yêu cầu An có thể xem được tất cả dữ liệu nên chỉnh sửa ở chính sách HolidayControl1 không ảnh hưởng đến An bằng cách trả về chuỗi rỗng

```
1 --connect SEC_MGR--
2 create or replace function fun_HolidayControl1(p_schema in varchar2, p_object in varchar2)
3 return varchar2
4 is
5 begin
6  if user = 'AN' then
7    return '';
8  elsif user = 'THU' then
9    return '1 = 0';
10 elsif user = 'HAN' then
11   return '';
12 else
13   return '1 = 0';
14 end if;
15 end;
```

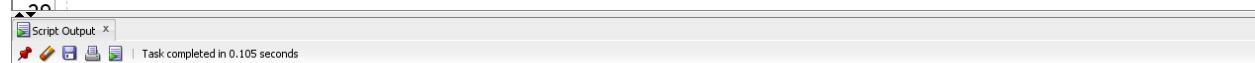


Function FUN_HOLIDAYCONTROL1 compiled

Hình: 114

- Tạo một chính sách mới ràng buộc cột Holiday cho An mà không ảnh hưởng đến các user khác

```
18 --connect SEC_MGR--
19 create or replace function fun_HolidayControl4(p_schema in varchar2, p_object in varchar2)
20 return varchar2
21 is
22 begin
23  if user = 'AN' then
24    return 'Name = ''An'''';
25  else
26    return '';
27  end if;
28 end;
```



Function FUN_HOLIDAYCONTROL4 compiled

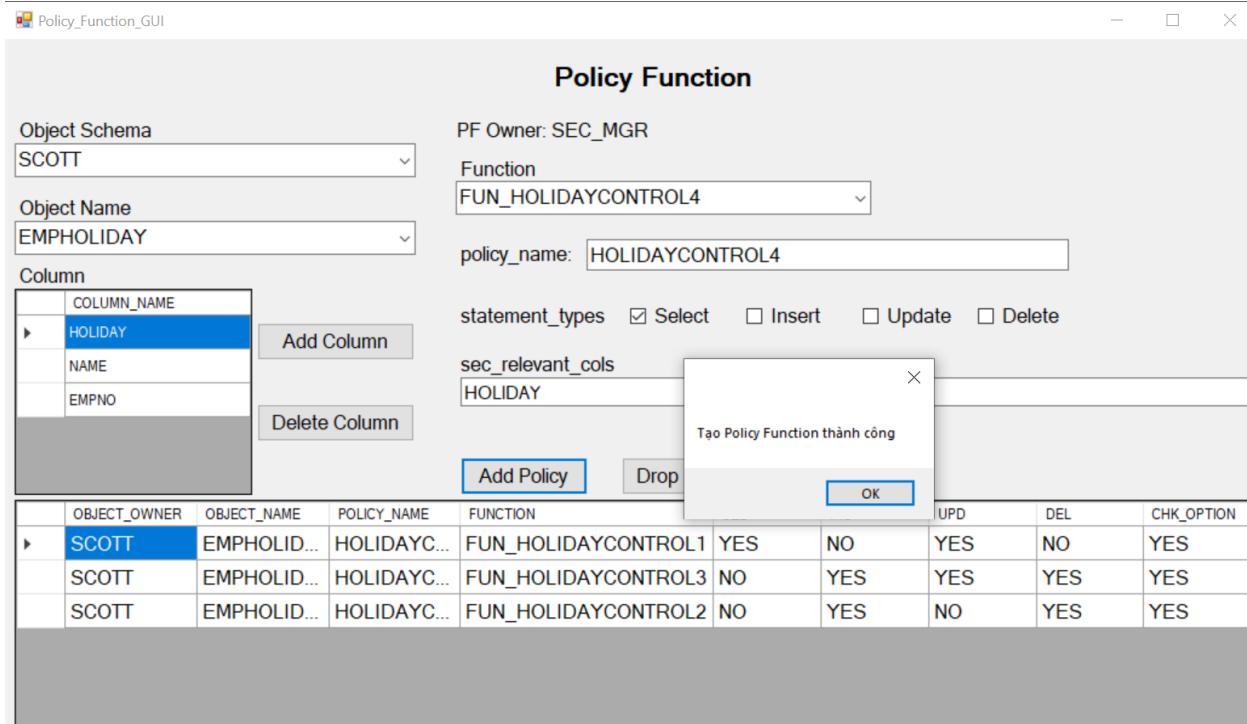
Hình: 115

- Tao policy

```
30 --connect SEC_MGR--  
31 begin  
32     dbms_rls.add_policy  
33     (  
34         object_schema => 'SCOTT',  
35         object_name => 'EMPHOLIDAY',  
36         policy_name => 'HolidayControl4',  
37         policy_function => 'fun_HolidayControl4',  
38         statement_types => 'Select',  
39         sec_relevant_cols => 'Holiday'  
40     );  
41 end;  
42 /
```

PL/SQL procedure successfully completed.

Hình: 116



Hình: 117

- Test trên user An: An được xem EmpNo và Name của các nhân viên khác

```
42 --connect An--
43 select empno, name from SCOTT.empholiday;
```

EMPNO	NAME
1	Han
2	An
3	Thu

Hình: 118

- Khi An xem ngày nghỉ thì chỉ xem được thông tin của mình

```
45 select empno, name, holiday from SCOTT.empholiday;
```

EMPNO	NAME	HOLIDAY
1	An	01/01/2020

Hình: 119

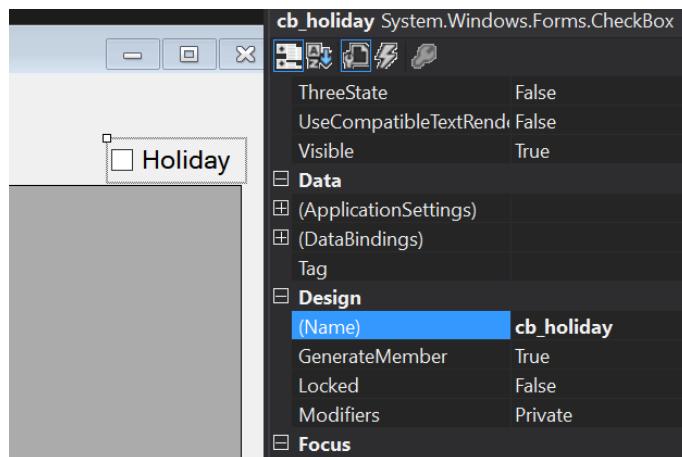
- Thay đổi thủ tục Select để truy vấn theo Holiday như chính sách vừa tạo

```
43 --connect SCOTT--
44 --Select--
45 create or replace procedure Pro_Select_EMPHOLIDAY(cur out sys_refcursor, Holiday_check in number)
46 is
47 begin
48   if Holiday_check = 1 then
49     open cur for select * from empholiday;
50   else
51     open cur for select EmpNo, Name from empholiday;
52   end if;
53 end;
```

Procedure PRO_SELECT_EMPHOLIDAY compiled

Hình: 120

- Thêm CheckBox Holiday vào form



Hình: 121

- Sửa hàm gọi chạy thủ tục Select

```
1 reference
public DataTable Select_Empholiday(int holiday_check)
{
    try
    {
        string Procedure = "scott.pro_select_Empholiday";

        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 Holiday_Check = new OracleParameter();
        Holiday_Check.ParameterName = "@Result";
        Holiday_Check.OracleDbType = OracleDbType.Int16;
        Holiday_Check.Value = holiday_check;
        Holiday_Check.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(Holiday_Check);

        cmd.ExecuteNonQuery();

        if (resultParam.Value != DBNull.Value)
        {
            OracleDataReader ret = ((OracleRefCursor)resultParam.Value).GetDataReader();
            DataTable data = new DataTable();
            data.Load(ret);
            return data;
        }
    }
    catch (OracleException ex)
    {
        MessageBox.Show("Truy vấn thất bại: " + ex.Message);
        return null;
    }
    return null;
}
```

Hình: 122

- Thêm thiết lập CheckBox vào hàm khai báo

```
3 references
public partial class EMPHOLIDAY_GUI : Form
{
    OracleConnection conn;
    Empholiday empholiday;
    1 reference
    public EMPHOLIDAY_GUI()
    {
        InitializeComponent();
        CenterToScreen();

        conn = Database.Get_Connect();
        empholiday = new Empholiday(conn);
        lb_user.Text = Database.User;

        cb_holiday.Checked = true;
    }
}
```

Hình: 123

- Sửa hàm Select

```
1 reference
private void btn_select_Click(object sender, EventArgs e)
{
    int holiday_check;
    if (cb_holiday.Checked)
    {
        holiday_check = 1;
    }
    else
    {
        holiday_check = 0;
    }

    dtg_table.DataSource = empholiday.Select_Empholiday(holiday_check);
}
```

Hình: 124

- Thay đổi sự kiện chọn bảng

```

private void dtg_table_SelectionChanged(object sender, EventArgs e)
{
    try
    {
        int row = dtg_table.CurrentCell.RowIndex;

        txt_empno.Text = dtg_table.Rows[row].Cells[0].Value.ToString();
        txt_name.Text = dtg_table.Rows[row].Cells[1].Value.ToString();

        if (cb_holiday.Checked)
            txt_holiday.Text = dtg_table.Rows[row].Cells[2].Value.ToString();
    }
    catch
    {
        return;
    }
}

```

Hình: 125

- Thêm sự kiện Check của CheckBox

```

1 reference
private void cb_holiday_CheckedChanged(object sender, EventArgs e)
{
    txt_holiday.Enabled = cb_holiday.Checked;
}

```

Hình: 126

- Đăng nhập vào An và truy vấn kèm cột Holiday

Table		
EMPHOLIDAY		
EMPNO	NAME	HOLIDAY
2	An	14/05/2010

Holiday

An

Empno Name Holiday

Select Insert Update Delete

Hình: 127

- Truy vấn không kèm cột Holiday

Table

EMPHOLIDAY

Holiday

	EMPNO	NAME
▶	1	Han
	2	An
	3	Thu
	4	Han

An

Empno Name Holiday

Select Insert Update Delete

Hình: 128

4. Bài tập 4

Từ chính sách HolidayControl ở câu 3, thiết lập quyền không ảnh hưởng và đảm bảo sự giám sát đối với chính sách này cho user Han. Thực thi một số thay đổi dữ liệu và xem kết quả.

- Cấp quyền để Han không bị ảnh hưởng bởi policy function

```

1 --connect SEC_MGR--
2 grant exempt access policy to Han;

```

Script Output x | Task completed in 0.073 seconds

Grant succeeded.

Hình: 129

- Audit quyền EXEMPT ACCESS POLICY

```

23 --connect SEC_MGR--
24 audit exempt access policy by access;

```

Script Output x | Task completed in 0.04 seconds

Audit succeeded.

Hình: 130

- Đăng nhập vào Han và thử Select, Update trên bảng empholiday

```
7 --connect Han-
8 select * from SCOTT.empholiday;
```

EMPNO	NAME	HOLIDAY
1	Han	02/08/2023
2	An	01/01/2020
3	Thu	26/08/2009

Hình: 131

```
10 update SCOTT.empholiday set holiday = '1/9/2022' where empno = 1;
```

1 row updated.

Hình: 132

- Vì SEC_MGR chưa có quyền xem trên bảng dba_audit_trail => dùng SYS để xem kết quả giám sát

```
12 --connect SYS-
13 select username, owner, action_name, obj_name, priv_used from dba_audit_trail;
```

USERNAME	OWNER	ACTION_NAME	OBJ_NAME	PRIV_USED
HAN	SCOTT	SELECT	EMPHOLIDAY	EXEMPT ACCESS POLICY
HAN	SCOTT	UPDATE	EMPHOLIDAY	EXEMPT ACCESS POLICY

Hình: 133

5. Bài tập 5

Cho bảng có cấu trúc như sau thuộc schema của sec_manager:

Employee (empno, ename, email, salary, deptno)

- empno (number) : mã số nhân viên
- ename (varchar2) : tên nhân viên
- email (varchar2) : email của nhân viên
- salary (number): lương nhân viên
- deptno (number) : mã số phòng ban của nhân viên

Lưu ý:

- Tên của nhân viên (ename) chính là username mà nhân viên đó dùng để log in vào hệ thống. (Sinh viên có thể dùng hàm USER trả về username của người dùng hiện tại)
- Sinh viên phải viết cả policy function và các lệnh gán policy function cho table employee.

- Sinh viên có thể viết 1 hay nhiều policy function để hiện thực chính sách trên.
- Các policy function tạo ra thuộc schema của user sec_manager và user sec_manager là người gán các policy function cho employee.

Hãy dùng kỹ thuật Row-level Security bảo vệ cho bảng employee theo chính sách được mô tả bên dưới

```

1 | --connect SEC_MGR-
2 | create table Employee
3 | (
4 |     empno number primary key,
5 |     ename varchar2(100),
6 |     email varchar2(50),
7 |     salary number,
8 |     deptno number
9 | );

```

Table EMPLOYEE created.

Hình: 134

```

| insert into employee values(1, 'Abel', 'Abel@gmail.com', 2500, 1);
| insert into employee values(2, 'Ande', 'Ande@gmail.com', 3200, 2);
| insert into employee values(3, 'Ernst', 'Ernst@gmail.com', 11000, 1);
| insert into employee values(4, 'Khoo', 'Khoo@gmail.com', 7400, 1);
| insert into employee values(5, 'Perkins', 'Perkins@gmail.com', 6500, 2);
| insert into employee values(6, 'Seo', 'Seo@gmail.com', 24000, 2);
| insert into employee values(7, 'Tobias', 'Tobias@gmail.com', 6500, 2);
| insert into employee values(8, 'Patel', 'Patel@gmail.com', 24000, 1);

```

Hình: 135

- Tạo User có tên là Employee dùng để tạo thủ tục thao tác đến bảng Employee

```

2 | --connect SYS--
3 | create user Employee identified by 123;
4 | grant create session to Employee;
5 | grant all privileges to Employee;

```

Hình: 136

- Tạo thủ tục Select

```

5  --connect Employee--
6  create or replace procedure pro_select_employee(cur out sys_refcursor)
7  is
8  begin
9      open cur for
L      select * from SEC_MGR.employee;
2 end;

```

Hình: 137

- Tạo thủ tục Insert

```

--Insert--
3  create or replace procedure pro_insert_employee(empno_in in number, ename_in in varchar2,
4                                              email_in in varchar2, salary_in in number,
5                                              deptno_in in number, ex out number)
6  is
7  begin
8      insert into SEC_MGR.employee values(empno_in, ename_in, email_in, salary_in, deptno_in);
9      ex := SQL%ROWCOUNT;
end;

```

Hình: 138

- Tạo thủ tục Update

```

--Update--
3  create or replace procedure pro_update_employee(empno_in in number, ename_in in varchar2,
4                                              email_in in varchar2, salary_in in number,
5                                              deptno_in in number, ex out number)
6  is
7  begin
8      update SEC_MGR.employee set ename = ename_in, email = email_in, salary = salary_in, deptno = deptno_in
9      where empno = empno_in;
10     ex := SQL%ROWCOUNT;
end;

```

Hình: 139

- Tạo thủ tục Delete

```

--Delete--
3  create or replace procedure pro_delete_employee(empno_in in number, ex out number)
4  is
5  begin
6      delete SEC_MGR.employee where empno = empno_in;
7      ex := SQL%ROWCOUNT;
end;

```

Hình: 140

- Gán quyền không ảnh hưởng đến Policy Function cho User SEC_MGR

```
--connect SYS--
grant exempt access policy to SEC_MGR;
```

Hình: 141

- Chuẩn bị 2 user đại diện cho 2 phòng khác nhau để test
- Abel phòng 1

```
--connect SYS--
--Abel--
create user Abel identified by 123;
grant create session to Abel;
grant select, insert, update, delete on Employee.EMPLOYEE to Abel;
grant execute on Employee.Pro_Select_EMPLOYEE to Abel;
grant execute on Employee.Pro_Insert_EMPLOYEE to Abel;
grant execute on Employee.Pro_Update_EMPLOYEE to Abel;
grant execute on Employee.Pro_Delete_EMPLOYEE to Abel;
```

Hình: 142

- Seo phòng 2

```
--Seo--
create user Seo identified by 123;
grant create session to Seo;
grant select, insert, update, delete on Employee.EMPLOYEE to Seo;
grant execute on Employee.Pro_Select_EMPLOYEE to Seo;
grant execute on Employee.Pro_Insert_EMPLOYEE to Seo;
grant execute on Employee.Pro_Update_EMPLOYEE to Seo;
grant execute on Employee.Pro_Delete_EMPLOYEE to Seo;
```

Hình: 143

5.1. Nhân viên thuộc phòng ban này không được phép xem hay chỉnh sửa (update) bất kỳ thông tin nào của những nhân viên thuộc phòng ban khác.

- Function của chính sách 5.1 trả về điều kiện là phòng của nhân viên (hình 144)

```

31 --connect SEC_MGR--
32 create or replace function fun_EmployeeControl1(p_schema in varchar2, p_object in varchar2)
33 return varchar2
34 is
35   f_deptno number;
36 begin
37   select deptno into f_deptno from SEC_MGR.employee where upper(ename) = upper(user);
38
39   return 'deptno = ' || f_deptno;
40
41   exception when others then
42     return '1 = 0';
43 end;

```

Script Output | Task completed in 0.082 seconds

Function FUN_EMPLOYEECONTROL1 compiled

Hình: 144

- Tạo chính sách có tên EmployeeControl1 (hình 145, 146)

```

95 --connect SEC_MGR--
96 begin
97   dbms_rls.add_policy
98   (
99     object_schema => 'SEC_MGR',
100    object_name => 'Employee',
101    policy_name => 'EmployeeControl1',
102    policy_function => 'fun_EmployeeControl1',
103    statement_types => 'SELECT, UPDATE',
104    update_check => TRUE
105  );
106 end;

```

Script Output | Task completed in 0.268 seconds

PL/SQL procedure successfully completed.

Hình: 145

Policy Function

Object Schema	PF Owner: SEC_MGR																																				
SEC_MGR																																					
Object Name	Function																																				
EMPLOYEE	FUN_EMPLOYEECONTROL1																																				
Column	policy_name: EMPLOYEECONTROL1																																				
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>COLUMN_NAME</td></tr> <tr><td>► MANAGER</td></tr> <tr><td>DEPTNO</td></tr> <tr><td>SALARY</td></tr> <tr><td>EMAIL</td></tr> <tr><td>ENAME</td></tr> </table>	COLUMN_NAME	► MANAGER	DEPTNO	SALARY	EMAIL	ENAME	<input type="checkbox"/> statement_types <input checked="" type="checkbox"/> Select <input type="checkbox"/> Insert <input checked="" type="checkbox"/> Update <input type="checkbox"/> Delete																														
COLUMN_NAME																																					
► MANAGER																																					
DEPTNO																																					
SALARY																																					
EMAIL																																					
ENAME																																					
	sec_relevant_c Tạo Policy Function thành công <input type="button" value="OK"/>																																				
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr><th>OBJECT_OWNER</th><th>OBJECT_NAME</th><th>POLICY_NAME</th><th>FUNCTION</th><th>SEL</th><th>INS</th><th>UPD</th><th>DEL</th><th>CHK_OPTION</th></tr> </thead> <tbody> <tr><td>► SCOTT</td><td>EMPHOLID...</td><td>HOLIDAYC...</td><td>FUN_HOLIDAYCONTROL4</td><td>YES</td><td>NO</td><td>NO</td><td>NO</td><td>YES</td></tr> <tr><td>SCOTT</td><td>EMPHOLID...</td><td>HOLIDAYC...</td><td>FUN_HOLIDAYCONTROL1</td><td>YES</td><td>NO</td><td>YES</td><td>NO</td><td>YES</td></tr> <tr><td>SCOTT</td><td>EMPHOLID...</td><td>HOLIDAYC...</td><td>FUN_HOLIDAYCONTROL2</td><td>NO</td><td>YES</td><td>NO</td><td>YES</td><td>YES</td></tr> </tbody> </table>	OBJECT_OWNER	OBJECT_NAME	POLICY_NAME	FUNCTION	SEL	INS	UPD	DEL	CHK_OPTION	► SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLIDAYCONTROL4	YES	NO	NO	NO	YES	SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLIDAYCONTROL1	YES	NO	YES	NO	YES	SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLIDAYCONTROL2	NO	YES	NO	YES	YES	
OBJECT_OWNER	OBJECT_NAME	POLICY_NAME	FUNCTION	SEL	INS	UPD	DEL	CHK_OPTION																													
► SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLIDAYCONTROL4	YES	NO	NO	NO	YES																													
SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLIDAYCONTROL1	YES	NO	YES	NO	YES																													
SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLIDAYCONTROL2	NO	YES	NO	YES	YES																													

Hình: 146

- Ví dụ lấy nhân viên Abel thuộc phòng 1 truy vấn vào phòng 2

```

60 --connect Abel--
61 select * from SEC_MGR.employee where Deptno = 2;

```

Script Output: X | Query Result: X | All Rows Fetched: 0 in 0.005 seconds

EMPNO	ENAME	EMAIL	SALARY	DEPTNO

Hình: 147

- Abel muốn sửa thông tin của nhân viên thuộc phòng 2:

```

63 update SEC_MGR.employee set salary = 7000 where empno = 2;

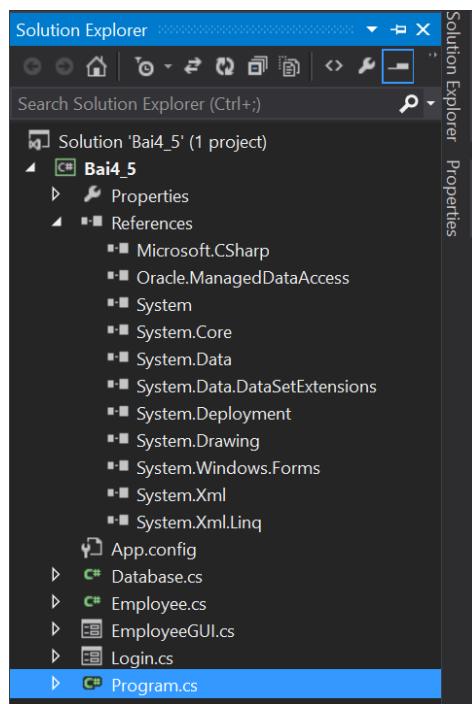
```

Script Output: X | Query Result: X | Task completed in 0.03 seconds

0 rows updated.

Hình: 148

- Tạo form thao tác trên bảng Employee



Hình: 149

```

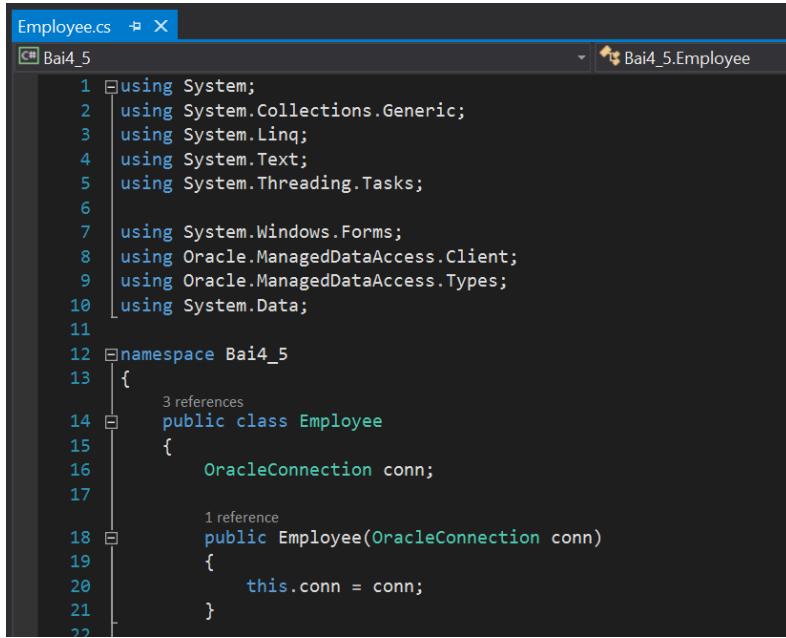
private void btn_login_Click(object sender, EventArgs e)
{
    string host = txt_host.Text;
    string port = txt_port.Text;
    string sid = txt_sid.Text;
    string user = txt_user.Text;
    string pass = txt_password.Text;

    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 EmployeeGUI().Show();
            this.Hide();
        }
        else
        {
            Check_Status(user);
            return;
        }
    }
}

```

Hình: 150

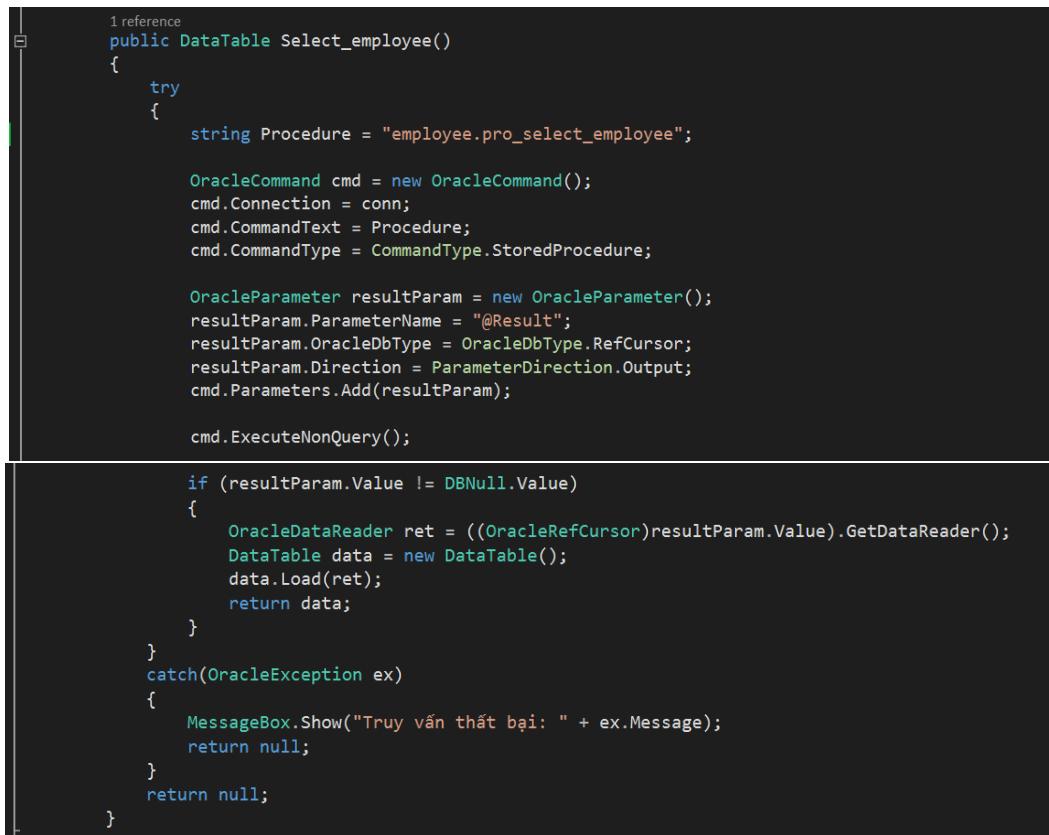
– Class Employee



```
Employee.cs  ✘ X
Bai4_5
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  using System.Windows.Forms;
8  using Oracle.ManagedDataAccess.Client;
9  using Oracle.ManagedDataAccess.Types;
10 using System.Data;
11
12 namespace Bai4_5
13 {
14     public class Employee
15     {
16         OracleConnection conn;
17
18         public Employee(OracleConnection conn)
19         {
20             this.conn = conn;
21         }
22     }
}
```

Hình: 151

– Select



```
1 reference
public DataTable Select_employee()
{
    try
    {
        string Procedure = "employee.pro_select_employee";

        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(OracleException ex)
    {
        MessageBox.Show("Truy vấn thất bại: " + ex.Message);
        return null;
    }
    return null;
}
```

Hình: 152

– Insert

```
public int Insert_Employee(int empno_in, string ename_in, string email_in, int salary_in, int deptno_in)
{
    try
    {
        string Procedure = "employee.pro_insert_employee";

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

        OracleParameter empno = new OracleParameter();
        empno.ParameterName = "@empno_in";
        empno.OracleDbType = OracleDbType.Int32;
        empno.Value = empno_in;
        empno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(empno);

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

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

        OracleParameter salary = new OracleParameter();
        salary.ParameterName = "@salary_in";
        salary.OracleDbType = OracleDbType.Int32;
        salary.Value = salary_in;
        salary.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(salary);

        OracleParameter deptno = new OracleParameter();
        deptno.ParameterName = "@deptno_in";
        deptno.OracleDbType = OracleDbType.Int32;
        deptno.Value = deptno_in;
        deptno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(deptno);

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

        cmd.ExecuteNonQuery();

        if (int.Parse(ex.Value.ToString()) == 0)
        {
            MessageBox.Show("0 dòng được thêm");
            return -1;
        }
        return 0;
    }
    catch(OracleException ex)
    {
        MessageBox.Show("Thêm thất bại: " + ex.Message);
        return -1;
    }
}
```

Hình: 153

- Update

```
1 reference
public int Update_Employee(int empno_in, string ename_in, string email_in, int salary_in, int deptno_in)
{
    try
    {
        string Procedure = "employee.pro_update_employee";

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

        OracleParameter empno = new OracleParameter();
        empno.ParameterName = "@empno_in";
        empno.OracleDbType = OracleDbType.Int32;
        empno.Value = empno_in;
        empno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(empno);

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

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

        OracleParameter salary = new OracleParameter();
        salary.ParameterName = "@salary_in";
        salary.OracleDbType = OracleDbType.Int32;
        salary.Value = salary_in;
        salary.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(salary);

        OracleParameter deptno = new OracleParameter();
        deptno.ParameterName = "@deptno_in";
        deptno.OracleDbType = OracleDbType.Int32;
        deptno.Value = deptno_in;
        deptno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(deptno);

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

        cmd.ExecuteNonQuery();

        if (int.Parse(ex.Value.ToString()) == 0)
        {
            MessageBox.Show("0 dòng được sửa");
            return -1;
        }

        return 0;
    }
    catch(OracleException ex)
    {
        MessageBox.Show("Sửa thất bại: " + ex.Message);
        return -1;
    }
}
```

Hình: 154

– Delete

```
1 reference
public int Delete_Employee(int empno_in)
{
    try
    {
        string Procedure = "employee.pro_delete_employee";

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

        OracleParameter empno = new OracleParameter();
        empno.ParameterName = "@empno_in";
        empno.OracleDbType = OracleDbType.Int32;
        empno.Value = empno_in;
        empno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(empno);

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

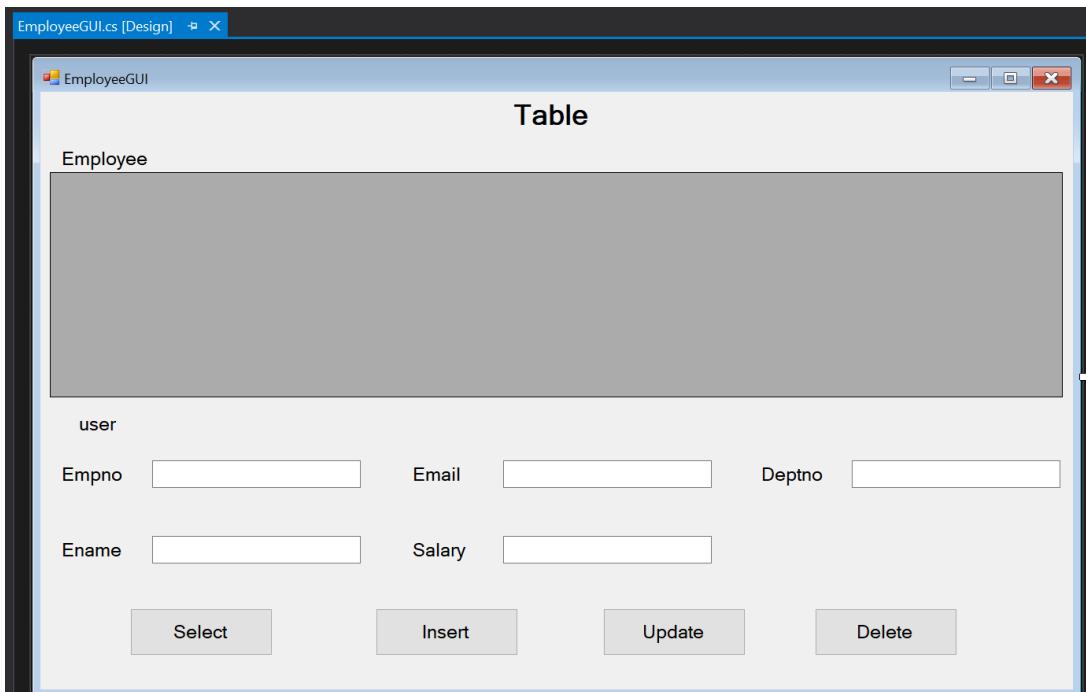
        cmd.ExecuteNonQuery();

        if (int.Parse(ex.Value.ToString()) == 0)
        {
            MessageBox.Show("0 dòng được xóa");
            return -1;
        }

        return 0;
    }
    catch(OracleException ex)
    {
        MessageBox.Show("Xóa thất bại: " + ex.Message);
        return -1;
    }
}
```

Hình: 155

- Thiết kế giao diện bảng Employee



Hình: 156

- Khai báo thư viện cần thiết

```

EmployeeGUI.cs  ✎ X EmployeeGUI.cs [Design]
Bai4_5
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 Bai4_5
14 {
15     public partial class EmployeeGUI : Form
16     {
17         OracleConnection conn;
18         Employee employee;
19         public EmployeeGUI()
20         {
21             InitializeComponent();
22             CenterToScreen();
23
24             conn = Database.Get_Connect();
25             employee = new Employee(conn);
26             lb_user.Text = Database.User;
27         }
28     }
}

```

Hình: 157

- Cài đặt sự kiện chọn bảng

```
1 reference
private void dtg_table_SelectionChanged(object sender, EventArgs e)
{
    int row = dtg_table.CurrentCell.RowIndex;

    txt_empno.Text = dtg_table.Rows[row].Cells[0].Value.ToString();
    txt_ename.Text = dtg_table.Rows[row].Cells[1].Value.ToString();
    txt_email.Text = dtg_table.Rows[row].Cells[2].Value.ToString();
    txt_salary.Text = dtg_table.Rows[row].Cells[3].Value.ToString();
    txt_deptno.Text = dtg_table.Rows[row].Cells[4].Value.ToString();
}
```

Hình: 158

- Nút select

```
1 reference
private void btn_select_Click(object sender, EventArgs e)
{
    dtg_table.DataSource = employee.Select_employee();
}
```

Hình: 159

- Nút Insert

```
1 reference
private void btn_insert_Click(object sender, EventArgs e)
{
    int empno = int.Parse(txt_empno.Text);
    string ename = txt_ename.Text;
    string email = txt_email.Text;
    int salary = int.Parse(txt_salary.Text);
    int deptno = int.Parse(txt_deptno.Text);

    int i = employee.Insert_Employee(empno, ename, email, salary, deptno);
    if (i == 0)
    {
        MessageBox.Show("Thêm thành công");
    }
}

1 reference.
```

Hình: 160

– Nút Update

```
1 reference
private void btn_update_Click(object sender, EventArgs e)
{
    int empno = int.Parse(txt_empno.Text);
    string ename = txt_ename.Text;
    string email = txt_email.Text;
    int salary = int.Parse(txt_salary.Text);
    int deptno = int.Parse(txt_deptno.Text);

    int i = employee.Update_Employee(empno, ename, email, salary, deptno);
    if (i == 0)
    {
        MessageBox.Show("Sửa thành công");
    }
}
```

Hình: 161

– Nút Delete

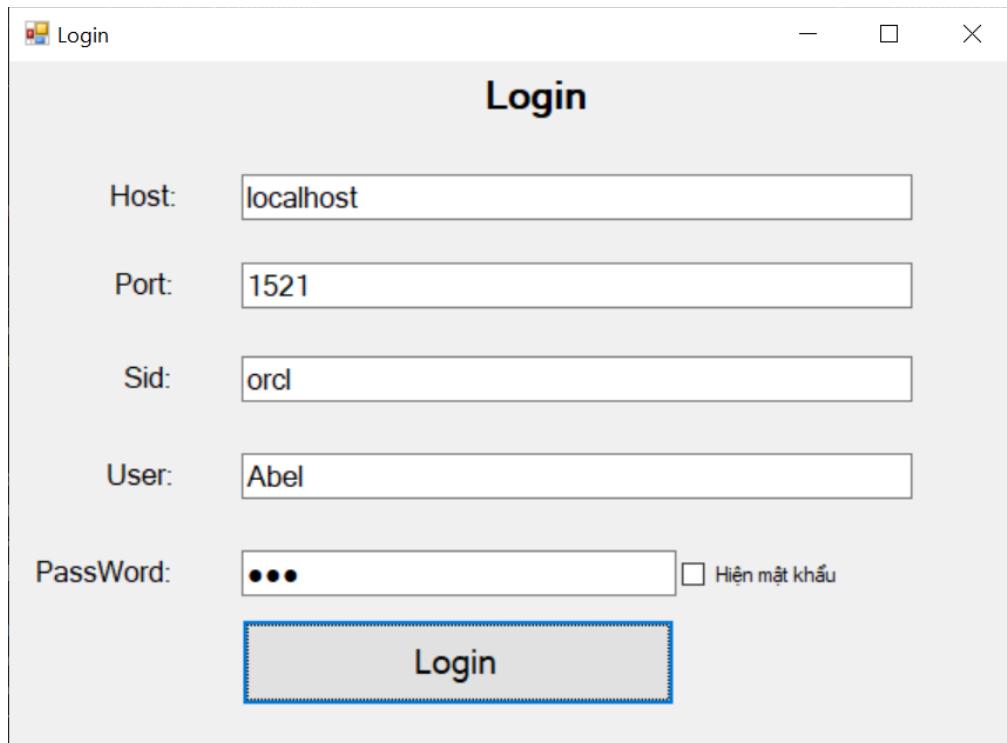
```
1 reference
private void btn_delete_Click(object sender, EventArgs e)
{
    int empno = int.Parse(txt_empno.Text);

    int i = employee.Delete_Employee(empno);
    if (i == 0)
    {
        MessageBox.Show("Xóa thành công");
    }
}

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

Hình: 162

- Đăng nhập vào Abel



Hình: 163

- Thực hiện Select

	EMPNO	ENAME	EMAIL	SALARY	DEPTNO
▶	1	Abel	Abel@gmail.com	2500	1
	3	Ernst	Ernst@gmail.com	11000	1
	4	Khoo	Khoo@gmail.com	7400	1
*	8	Patel	Patel@gmail.com	24000	1

Abel

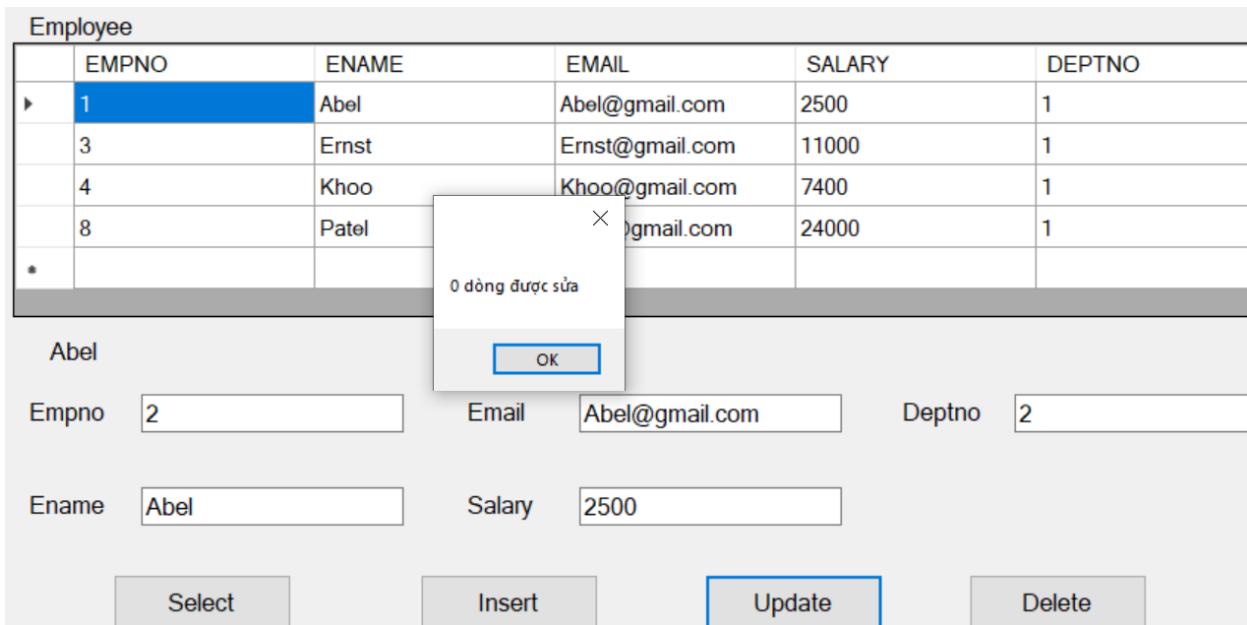
Empno Email Deptno

Ename Salary

Select Insert Update Delete

Hình: 164

- Cập nhật nhân viên phòng khác



Hình: 165

5.2. Các nhân viên được phép xem (select) các thông tin của những người trong cùng phòng ban.

Chỉnh sửa lại chính sách ở 5.1 thành điều kiện xem và tạo một chính sách để quản lý việc Update của nhân viên

- Xóa chính sách 5.1 cũ

```

72 --connect SEC_MGR--
73 begin
74   DBMS_RLS.drop_policy
75   (
76     object_schema => 'SEC_MGR',
77     object_name => 'Employee',
78     policy_name => 'EmployeeControl1'
79   );
80 end;
81

```

Script Output X | Query Result X
Task completed in 0.177 seconds

PL/SQL procedure successfully completed.

Hình: 166

- Chính sửa lại chính sách EmployeeControl1 ở câu 5.1 chỉ quản lý việc select

```

82 --connect SEC_MGR--
83 begin
84   dbms_rls.add_policy
85   (
86     object_schema => 'SEC_MGR',
87     object_name => 'Employee',
88     policy_name => 'EmployeeControl1',
89     policy_function => 'fun_EmployeeControl1',
90     statement_types => 'SELECT'
91   );
92 end;

```

Script Output | Query Result | Task completed in 0.08 seconds

PL/SQL procedure successfully completed.

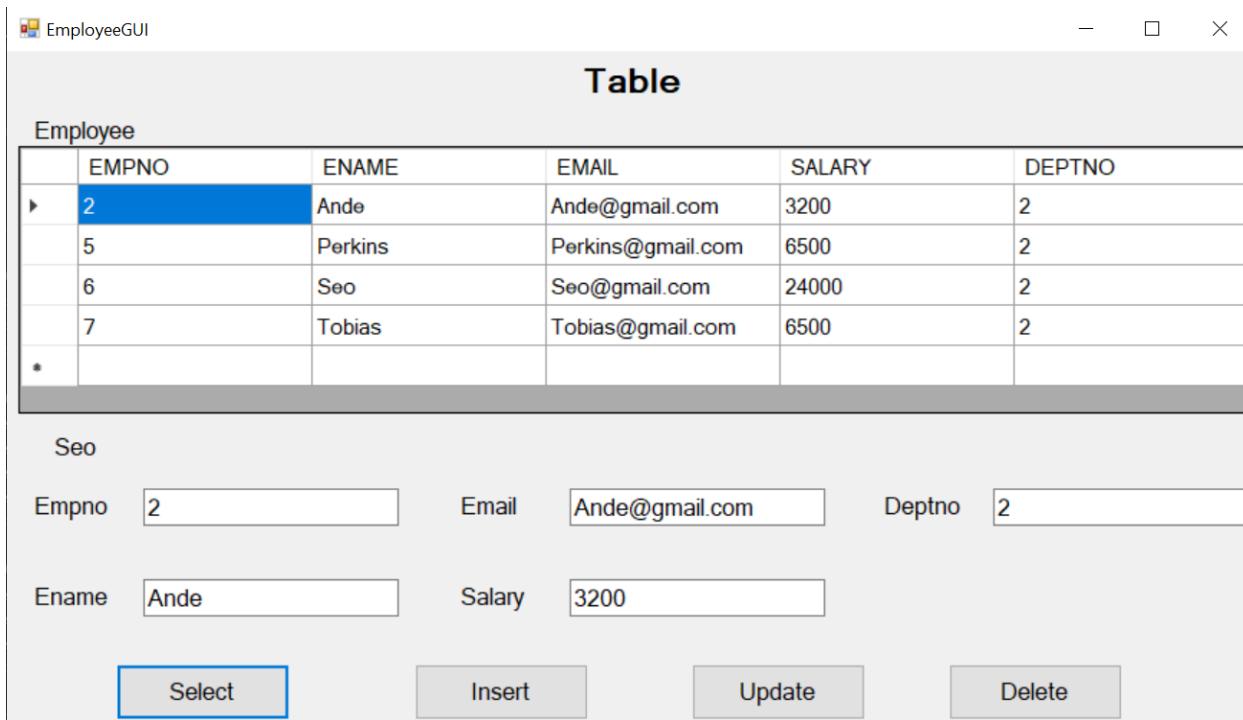
Hình: 167

- Tạo lại Policy Function bằng Form

OBJECT_OWNER	OBJECT_NAME	POLICY_NAME	FUNCTION	SEL	INS	UPD	DEL	CHK_OPTION
SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLI...	NO	YES	YES	YES	YES
SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLI...	YES	NO	NO	NO	NO
SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLI...	YES	NO	NO	NO	YES

Hình: 168

- Đăng nhập vào Seo truy vấn lại thông tin



Hình: 169

5.3. Nhân viên không được phép insert/delete trên bảng.

- Tạo function để quản lý việc insert/delete

```

87 --connect SEC_MGR--
88 create or replace function fun_EmployeeControl2(p_schema in varchar2, p_object in varchar2)
89 return varchar2
90 is
91 begin
92     return '1 = 0';
93 end;
94

```

Function FUN_EMPLOYEECONTROL2 compiled

Hình: 170

- Tạo policy EmployeeControl2 với statement_types => 'INSERT, DELETE'

```

95  --connect SEC_MGR--
96  begin
97      dbms_rls.add_policy
98      (
99          object_schema => 'SEC_MGR',
100         object_name => 'Employee',
101        policy_name => 'EmployeeControl2',
102        policy_function => 'fun_EmployeeControl2',
103        statement_types => 'INSERT, DELETE'
104    );
105 end;
106

```

Script Output | Query Result | Task completed in 0.046 seconds

PL/SQL procedure successfully completed.

Hình: 171

Policy Function

Object Schema: SEC_MGR
PF Owner: SEC_MGR
Function: FUN_EMPLOYEECONTROL2
policy_name: EMPLOYEECONTROL2

	COLUMN_NAME	Add Column	Delete Column
▶	DEPTNO		
	SALARY		
	EMAIL		
	ENAME		
	EMPNO		

statement_types: Select Insert Update Delete

Tạo Policy Function thành công

OK

OBJECT_OWNER	OBJECT_NAME	POLICY_NAME	FUNCTION	SEL	INS	UPD	DEL	CHK_OPTION
▶ SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLI...	NO	YES	YES	YES	YES
SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLI...	YES	NO	NO	NO	NO
SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLI...	YES	NO	NO	NO	YES
SEC_MGR	EMPLOYEE	EMPLOYEE...	FUN_EMPL...	YES	NO	NO	NO	YES

Hình: 172

- Đăng nhập vào user Abel
- Thực hiện xóa một nhân viên khác

Table					
Employee					
	EMPNO	ENAME	EMAIL	SALARY	DEPTNO
	1	Abel	Abel@gmail.com	2500	1
▶	3	Ernst	Ernst@gmail.com	11000	1
	4	Khoo	Khoo@gmail.com	7400	1
	8	Patel		24000	1
*					

Abel

Empno Email Deptno

Ename Salary

Select Insert Update Delete

Hình: 173

- Thực hiện thêm một nhân viên

Table					
Employee					
	EMPNO	ENAME	EMAIL	SALARY	DEPTNO
	1	Abel	Abel@gmail.com	2500	1
▶	3			11000	1
	4			7400	1
	8			24000	1
*					

Thêm thất bại: ORA-28115: policy with check option violation
ORA-06512: at 'EMPLOYEE.PRO_INSERT_EMPLOYEE', line 5
ORA-06512: at line 1

Abel

Empno Email Deptno

Ename Salary

Select Insert Update Delete

Hình: 174

5.4. Nhân viên chỉ có thể update thông tin email của bản thân mình. Những thông tin cá nhân còn lại không được phép chỉnh sửa.

Sử dụng Column Sensitive VPD trên cột email để quản lý việc cập nhật cột này của nhân viên. Theo yêu cầu của bài thì ta cần có 2 chính sách: 1 chính sách cho cột email được cập nhật đúng; 1 chính sách ngăn chặn việc cập nhật trên các cột khác trên bảng Employee

- Tạo function cho chính sách update cột email

```
108 --connect SEC_MGR--  
109 create or replace function fun_EmployeeControl3(p_schema in varchar2, p_object in varchar2)  
110 return varchar2  
111 is  
112 begin  
113     return q'[upper(ename) = '[' || upper(user) || q']]';  
114 end;  
Script Output X | Task completed in 0.214 seconds
```

Function FUN_EMPLOYEECONTROL3 compiled

Hình: 175

- Tạo chính sách EmployeeControl3 (hình 176,177)

```
116 --connect SEC_MGR--  
117 begin  
118     dbms_rls.add_policy  
119     (  
120         object_schema => 'SEC_MGR',  
121         object_name => 'Employee',  
122         policy_name => 'EmployeeControl3',  
123         policy_function => 'fun_EmployeeControl3',  
124         sec_relevant_cols => 'email',  
125         statement_types => 'UPDATE'  
126     );  
127 end;  
Script Output X | Task completed in 0.232 seconds  
PL/SQL procedure successfully completed.
```

Hình: 176

Policy Function

Object Schema SEC_MGR	PF Owner: SEC_MGR Function FUN_EMPLOYEECONTROL3 policy_name: EMPLOYEECONTROL3																														
Column <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <th>COLUMN_NAME</th> <th>Add Column</th> </tr> <tr> <td>DEPTNO</td> <td></td> </tr> <tr> <td>SALARY</td> <td></td> </tr> <tr> <td>EMAIL</td> <td>Add P</td> </tr> <tr> <td>ENAME</td> <td></td> </tr> <tr> <td>EMPNO</td> <td></td> </tr> </table>		COLUMN_NAME	Add Column	DEPTNO		SALARY		EMAIL	Add P	ENAME		EMPNO																			
COLUMN_NAME	Add Column																														
DEPTNO																															
SALARY																															
EMAIL	Add P																														
ENAME																															
EMPNO																															
statement_types <input type="checkbox"/> Select <input type="checkbox"/> Insert <input checked="" type="checkbox"/> Update <input type="checkbox"/> Delete sec_relevant_cols <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td>EMAIL</td> <td>X</td> </tr> </table>		EMAIL	X																												
EMAIL	X																														
Tạo Policy Function thành công																															
<input type="button" value="OK"/>																															
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>SEL</th> <th>INS</th> <th>UPD</th> <th>DEL</th> <th>CHK_OPTION</th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>YES</td> <td>YES</td> <td>YES</td> <td>YES</td> </tr> <tr> <td>YES</td> <td>NO</td> <td>NO</td> <td>NO</td> <td>NO</td> </tr> <tr> <td>YES</td> <td>NO</td> <td>NO</td> <td>NO</td> <td>YES</td> </tr> <tr> <td>NO</td> <td>YES</td> <td>NO</td> <td>YES</td> <td>YES</td> </tr> <tr> <td>YES</td> <td>NO</td> <td>NO</td> <td>NO</td> <td>YES</td> </tr> </tbody> </table>		SEL	INS	UPD	DEL	CHK_OPTION	NO	YES	YES	YES	YES	YES	NO	NO	NO	NO	YES	NO	NO	NO	YES	NO	YES	NO	YES	YES	YES	NO	NO	NO	YES
SEL	INS	UPD	DEL	CHK_OPTION																											
NO	YES	YES	YES	YES																											
YES	NO	NO	NO	NO																											
YES	NO	NO	NO	YES																											
NO	YES	NO	YES	YES																											
YES	NO	NO	NO	YES																											

Hình: 177

- Về function chặn việc update trên các cột còn lại ta có thể dùng fun_EmployeeControl2 của chính sách chặn insert/delete, ta chỉ cần tạo chính sách mới.
- Lưu ý Column Sensitive VPD sẽ được kích hoạt nếu ta có sử dụng đến cột thỏa điều kiện trong chính sách cho dù nó ở đâu trong câu lệnh, vì vậy ta không nên lấy cả cột primary key mà để lại như một cột để tham chiếu điều kiện chuẩn (hình 178, 179)

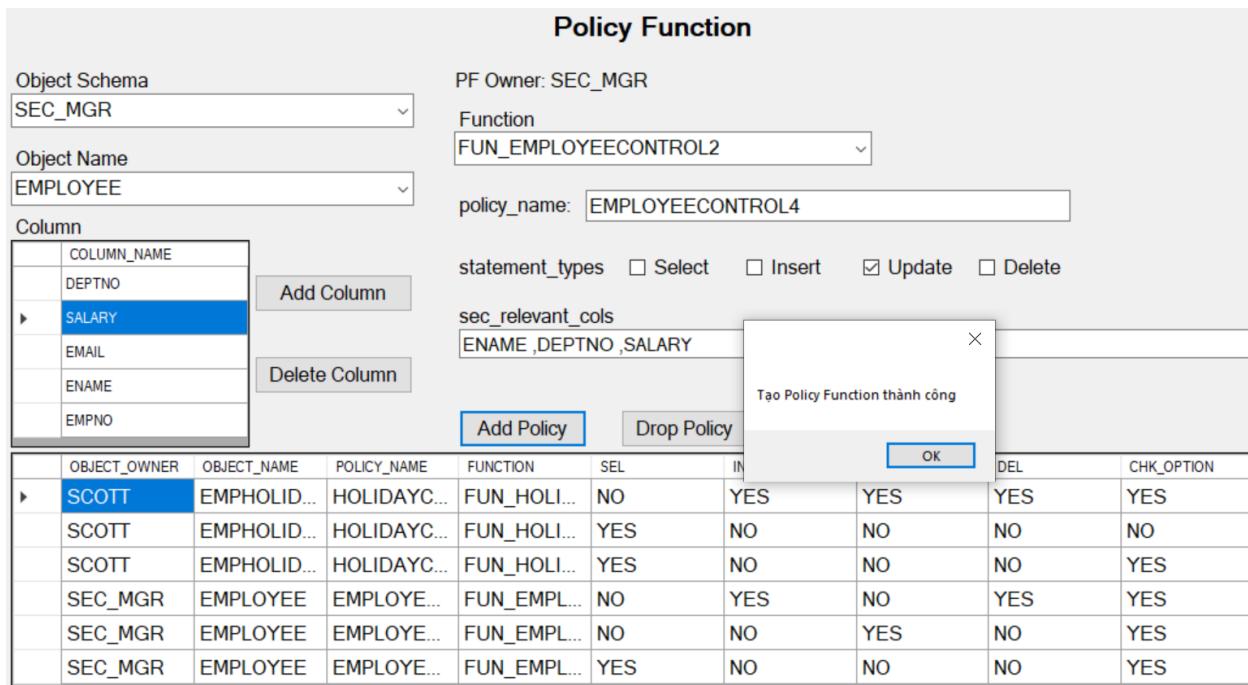
```

129 --connect SEC_MGR--
130 begin
131   dbms_rls.add_policy
132   (
133     object_schema => 'SEC_MGR',
134     object_name => 'Employee',
135     policy_name => 'EmployeeControl4',
136     policy_function => 'fun_EmployeeControl2',
137     sec_relevant_cols => 'ename, salary, deptno',
138     statement_types => 'UPDATE'
139   );
140 end;
  
```

Script Output | Task completed in 0.038 seconds

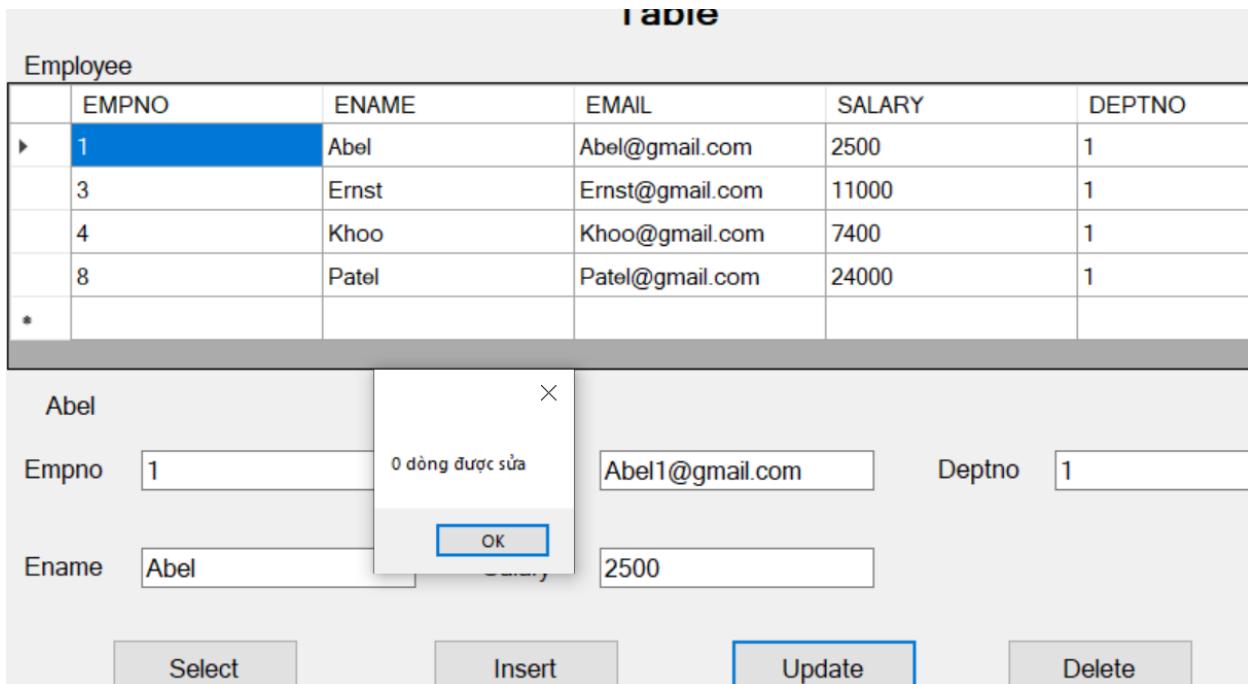
PL/SQL procedure successfully completed.

Hình: 178



Hình: 179

- Vì đoạn lệnh cập nhật trong form Table là đoạn lệnh cập nhật tất cả cột cho dù chỉ có một cột thay đổi dữ liệu cho nên khi câu lệnh được thực thi thì có 2 chính sách được kích hoạt tạo ra điều kiện chặn cập nhật ở EmployeeControl4, điều kiện cập nhật ở EmployeeControl3 nên ta có thông báo như sau



Hình: 180

- Cách khắc phục phù hợp với bài là sửa lại câu lệnh cập nhật chỉ cập nhật trên cột email và kèm điều kiện mã nhân viên nếu nhân viên chọn dữ liệu của nhân viên khác

```

203 --connect Employee--
204 --Update Email--
205 create or replace procedure pro_update_Email_employee(empno_in in number, email_in in varchar2, ex out number)
206 is
207 begin
208     update SEC_MGR.employee set email = email_in
209     where empno = empno_in;
210     ex := SQL%ROWCOUNT;
211 end;

```

Script Output: X | Task completed in 0.083 seconds

Procedure PRO_UPDATE_EMAIL_EMPLOYEE compiled

Hình: 181

- Gán quyền cho user

```

212 --connect SYS--
213 grant execute on Employee.Pro_Update_Email_EMPLOYEE to Abel;
214 grant execute on Employee.Pro_Update_Email_EMPLOYEE to Seo;
215

```

Script Output: X | Task completed in 0.09 seconds

Grant succeeded.

Grant succeeded.

Hình: 182

– Viết hàm Update Email

```
1 reference
public int Update_Email_Employee(int empno_in, string email_in)
{
    try
    {
        string Procedure = "employee.pro_update_email_employee";

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

        OracleParameter empno = new OracleParameter();
        empno.ParameterName = "@empno_in";
        empno.OracleDbType = OracleDbType.Int32;
        empno.Value = empno_in;
        empno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(empno);

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

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

        cmd.ExecuteNonQuery();

        if (int.Parse(ex.Value.ToString()) == 0)
        {
            MessageBox.Show("0 Email được sửa");
            return -1;
        }

        return 0;
    }
    catch (OracleException ex)
    {
        MessageBox.Show("Sửa Email thất bại: " + ex.Message);
        return -1;
    }
}
```

Hình: 183

- Thêm Buttun Update Email

Table

Employee

user

Empno	<input type="text"/>	Email	<input type="text"/>	Deptno	<input type="text"/>
Ename	<input type="text"/>	Salary	<input type="text"/>		

Select Insert Update Update Email Delete

Hình: 184

- Cài đặt sự kiện Click

```
1 reference
private void btn_update_email_Click(object sender, EventArgs e)
{
    int empno = int.Parse(txt_empno.Text);
    string email = txt_email.Text;

    int i = employee.Update_Email_Employee(empno, email);
    if (i == 0)
    {
        MessageBox.Show("Sửa Email thành công");
    }
}
```

Hình: 185

- Trường hợp nhân viên cập nhật email của mình (hình 186, 187)

Table

Employee

	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	
▶	1	Abel	Abel@gmail.com	2500	1	
	3	Ernst	Ernst@gmail.com	11000	1	
	4	Khoo	Khoo@gmail.com	7400	1	
*	8	Patel	Patel@gmail.com	24000	1	
Sửa Email thành công						
Abel						
Empno		<input type="text" value="1"/>	Email	<input type="text" value="Abel2@gmail.com"/>	Deptno	<input type="text" value="1"/>
Ename		<input type="text" value="Abel"/>	Salary	<input type="text" value="2500"/>		
<input type="button" value="Select"/>		<input type="button" value="Insert"/>	<input type="button" value="Update"/>	<input style="border: 2px solid blue; background-color: #e0f2e0; color: black; font-weight: bold; font-size: 10pt; padding: 2px 10px; border-radius: 5px; font-family: inherit; font-style: inherit; font-variant: inherit; font-weight: inherit; line-height: inherit; margin-right: 10px;" type="button" value="Update Email"/>	<input type="button" value="Delete"/>	

Hình: 186

Table

Employee

	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	
▶	1	Abel	Abel2@gmail.com	2500	1	
	3	Ernst	Ernst@gmail.com	11000	1	
	4	Khoo	Khoo@gmail.com	7400	1	
*	8	Patel	Patel@gmail.com	24000	1	
Abel						
Empno		<input type="text" value="1"/>	Email	<input type="text" value="Abel2@gmail.com"/>	Deptno	<input type="text" value="1"/>
Ename		<input type="text" value="Abel"/>	Salary	<input type="text" value="2500"/>		
<input style="border: 2px solid blue; background-color: #e0f2e0; color: black; font-weight: bold; font-size: 10pt; padding: 2px 10px; border-radius: 5px; font-family: inherit; font-style: inherit; font-variant: inherit; font-weight: inherit; line-height: inherit; margin-right: 10px;" type="button" value="Select"/>		<input type="button" value="Insert"/>	<input type="button" value="Update"/>	<input type="button" value="Update Email"/>	<input type="button" value="Delete"/>	

Hình: 187

- Trường hợp nhân viên cố tình cập nhật email của người khác (hình 188)

Table

Employee

EMPNO	ENAME	EMAIL	SALARY	DEPTNO
1	Abel	Abel@gmail.com	2500	1
3	Ernst	Ernst@gmail.com	11000	1
4	Khoo	Khoo@gmail.com	7400	1
8	Patel	Patel@gmail.com	24000	1
*				

Abel

Empno: 3 Email: Ernst2@gmail.com Deptno: 1

Ename: Ernst Salary: 11000

Select Insert Update **Update Email** Delete

0 Email được sửa

OK

Hình: 188

6. Cho bảng có cấu trúc như sau thuộc schema của sec_manager:

Employee (empno, ename, email , salary, deptno, manager) gồm các chi tiết:

- empno (number) : mã số nhân viên
- ename (varchar2) : tên nhân viên
- email (varchar2) : email của nhân viên
- salary (number): lương nhân viên
- deptno (number) : mã số phòng ban của nhân viên
- manager(number): mã số người quản lý của phòng ban mà nhân viên thuộc về

Lưu ý:

- Tên của nhân viên (ename) chính là username mà nhân viên đó dùng để log in vào hệ thống (Sinh viên có thể dùng hàm USER trả về username của người dùng hiện tại)
- Sinh viên phải viết cả policy function và các lệnh gán policy function cho table employee.
- Sinh viên có thể viết 1 hay nhiều policy function để hiện thực chính sách trên.
- Các policy function tạo ra thuộc schema của user sec_manager và user sec_manager là người gán các policy function cho employee

Hãy dùng kỹ thuật Row-level Security bảo vệ cho bảng employee theo chính sách được mô tả

Hướng dẫn

- Tạo bảng

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL script:

```
1 create table Employee
2 (
3     empno number primary key,
4     ename varchar2(100),
5     email varchar2(50),
6     salary number,
7     deptno number,
8     manager number,
9     constraint fk_emp foreign key(manager) references Employee(empno)
10 );
```

In the bottom-right pane, the output window displays the message "Table EMPLOYEE created.".

Hình: 189

- Thêm dữ liệu cho bảng

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL script:

```
14
15 insert into employee values(1, 'Abel', 'Abel@gmail.com', 2500, 1, null);
16 insert into employee values(2, 'Ande', 'Ande@gmail.com', 3200, 2, null);
17 insert into employee values(3, 'Ernst', 'Ernst@gmail.com', 11000, 1, 1);
18 insert into employee values(4, 'Khoo', 'Khoo@gmail.com', 7400, 1, 1);
19 insert into employee values(5, 'Perkins', 'Perkins@gmail.com', 6500, 2, 2);
20 insert into employee values(6, 'Seo', 'Seo@gmail.com', 24000, 2, 2);
21 insert into employee values(7, 'Tobias', 'Tobias@gmail.com', 6500, 2, 2);
22 insert into employee values(8, 'Patel', 'Patel@gmail.com', 24000, 1, 1);
23
```

In the bottom-right pane, the output window displays the message "1 row inserted." followed by four identical messages: "1 row inserted.", each corresponding to one of the eight rows inserted.

Hình: 190

– Tạo thủ tục select

```
--connect Employee--  
create or replace procedure pro_select_employee(cur out sys_refcursor)  
is  
begin  
    open cur for  
        select * from SEC_MGR.employee;  
end;
```

Hình: 191

– Tạo thủ tục Insert

```
--Insert--  
create or replace procedure pro_insert_employee(empno_in in number, ename_in in varchar2,  
email_in in varchar2, salary_in in number, deptno_in in number, manager_in in number default null, ex out number)  
is  
begin  
    insert into SEC_MGR.employee values(empno_in, ename_in, email_in, salary_in, deptno_in, manager_in);  
    ex := SQL%ROWCOUNT;  
end;
```

Hình: 192

– Tạo thủ tục Update

```
--Update--  
create or replace procedure pro_update_employee(empno_in in number, ename_in in varchar2,  
email_in in varchar2, salary_in in number, deptno_in in number, manager_in in number default null, ex out number)  
is  
begin  
    update SEC_MGR.employee set ename = ename_in, email = email_in,  
    salary = salary_in, deptno = deptno_in, manager = manager_in where empno = empno_in;  
    ex := SQL%ROWCOUNT;  
end;
```

Hình: 193

– Tạo thủ tục Delete

```
--Delete--  
create or replace procedure pro_delete_employee(empno_in in number, ex out number)  
is  
begin  
    delete SEC_MGR.employee where empno = empno_in;  
    ex := SQL%ROWCOUNT;  
end;
```

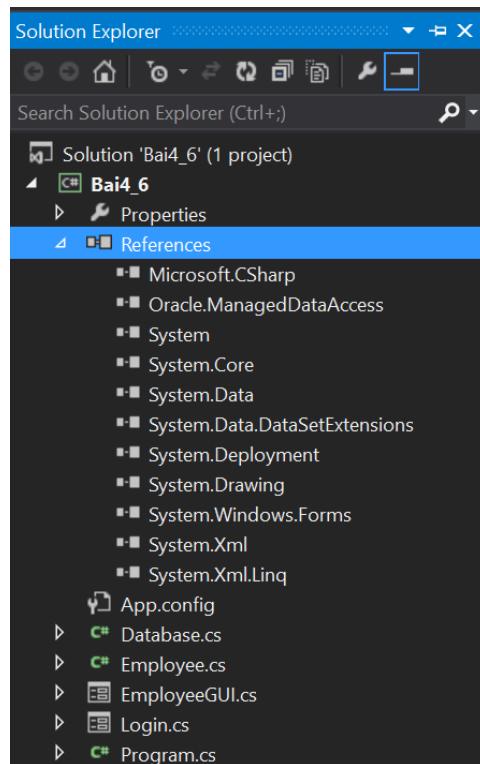
Hình: 194

- Tạo User và gán quyền

```
--connect SYS--
--Abel--
create user Abel identified by 123;
grant create session to Abel;
grant select, insert, update, delete on SEC_MGR.EMPLOYEE to Abel;
grant execute on Employee.Pro_Select_EMPLOYEE to Abel;
grant execute on Employee.Pro_Insert_EMPLOYEE to Abel;
grant execute on Employee.Pro_Update_EMPLOYEE to Abel;
grant execute on Employee.Pro_Delete_EMPLOYEE to Abel;
--Seo--
create user Seo identified by 123;
grant create session to Seo;
grant select, insert, update, delete on SEC_MGR.EMPLOYEE to Seo;
grant execute on Employee.Pro_Select_EMPLOYEE to Seo;
grant execute on Employee.Pro_Insert_EMPLOYEE to Seo;
grant execute on Employee.Pro_Update_EMPLOYEE to Seo;
grant execute on Employee.Pro_Delete_EMPLOYEE to Seo;
```

Hình: 195

- Form thao tác trên bảng



Hình: 196

- Sử dụng lại form Login và sửa lại nút Login



```

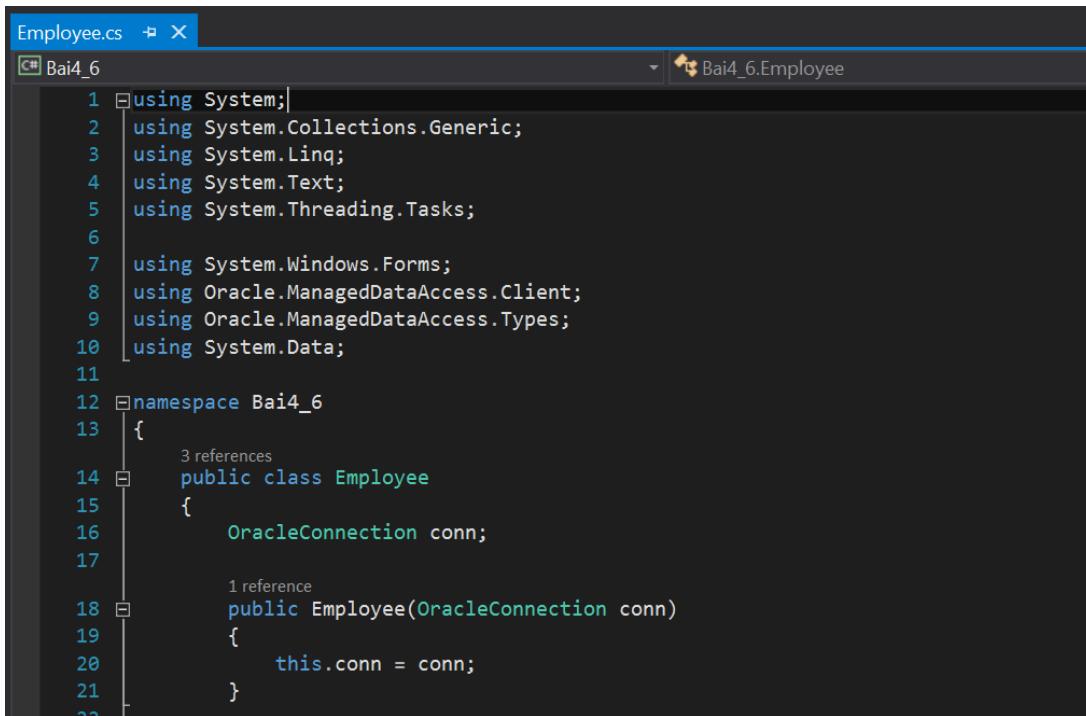
1 reference
private void btn_login_Click(object sender, EventArgs e)
{
    string host = txt_host.Text;
    string port = txt_port.Text;
    string sid = txt_sid.Text;
    string user = txt_user.Text;
    string pass = txt_password.Text;

    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 EmployeeGUI().Show();
            this.Hide();
        }
        else
        {
            Check_Status(user);
            return;
        }
    }
}

```

Hình: 197

- Tạo Class Employee



```

Employee.cs  ✎ ×
C# Bai4_6
namespace Bai4_6
{
    using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Text;
    using System.Threading.Tasks;
    using System.Windows.Forms;
    using Oracle.ManagedDataAccess.Client;
    using Oracle.ManagedDataAccess.Types;
    using System.Data;
}

public class Employee
{
    OracleConnection conn;

    public Employee(OracleConnection conn)
    {
        this.conn = conn;
    }
}

```

Hình: 198

– Hàm Select

```
    ↑ reference
public DataTable Select_employee()
{
    try
    {
        string Procedure = "employee.pro_select_employee";

        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(OracleException ex)
    {
        MessageBox.Show("Truy vấn thất bại: " + ex.Message);
        return null;
    }
    return null;
}
```

Hình: 199

– Hàm Insert (hình 200, 201)

```
1 reference
public int Insert_Employee(int empno_in, string ename_in, string email_in,
                           int salary_in, int deptno_in, int manager_in)
{
    try
    {
        string Procedure = "employee.pro_insert_employee";

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

        OracleParameter empno = new OracleParameter();
        empno.ParameterName = "@empno_in";
        empno.OracleDbType = OracleDbType.Int32;
        empno.Value = empno_in;
        empno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(empno);

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

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

        OracleParameter salary = new OracleParameter();
        salary.ParameterName = "@salary_in";
        salary.OracleDbType = OracleDbType.Int32;
        salary.Value = salary_in;
        salary.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(salary);

        OracleParameter deptno = new OracleParameter();
        deptno.ParameterName = "@deptno_in";
        deptno.OracleDbType = OracleDbType.Int32;
        deptno.Value = deptno_in;
        deptno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(deptno);

        OracleParameter manager = new OracleParameter();
        manager.ParameterName = "@manager_in";
        manager.OracleDbType = OracleDbType.Int32;
        manager.Direction = ParameterDirection.Input;
        if (manager_in != -1)
        {
            manager.Value = manager_in;
        }
        cmd.Parameters.Add(manager);
```

Hình: 200

```

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

        cmd.ExecuteNonQuery();

        if (int.Parse(ex.Value.ToString()) == 0)
        {
            MessageBox.Show("0 dòng được thêm");
            return -1;
        }
        return 0;
    }
    catch (OracleException ex)
    {
        MessageBox.Show("Thêm thất bại: " + ex.Message);
        return -1;
    }
}

```

Hình: 201

- Hàm Update

```

public int Update_Employee(int empno_in, string ename_in, string email_in,
                           int salary_in, int deptno_in, int manager_in)
{
    try
    {
        string Procedure = "employee.pro_update_employee";

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

        OracleParameter empno = new OracleParameter();
        empno.ParameterName = "@empno_in";
        empno.OracleDbType = OracleDbType.Int32;
        empno.Value = empno_in;
        empno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(empno);

        OracleParameter ename = new OracleParameter();
        ename.ParameterName = "@ename_in";
        ename.OracleDbType = OracleDbType.Varchar2;
        ename.Value = ename_in;
        ename.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(ename);
    }
}

```

Hình: 202

```

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

        OracleParameter salary = new OracleParameter();
        salary.ParameterName = "@salary_in";
        salary.OracleDbType = OracleDbType.Int32;
        salary.Value = salary_in;
        salary.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(salary);

        OracleParameter deptno = new OracleParameter();
        deptno.ParameterName = "@deptno_in";
        deptno.OracleDbType = OracleDbType.Int32;
        deptno.Value = deptno_in;
        deptno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(deptno);

        OracleParameter manager = new OracleParameter();
        manager.ParameterName = "@manager_in";
        manager.OracleDbType = OracleDbType.Int32;
        manager.Direction = ParameterDirection.Input;
        if (manager_in != -1)
        {
            manager.Value = manager_in;
        }
        cmd.Parameters.Add(manager);

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

        cmd.ExecuteNonQuery();

        if (int.Parse(ex.Value.ToString()) == 0)
        {
            MessageBox.Show("0 dòng được sửa");
            return -1;
        }

        return 0;
    }
    catch (OracleException ex)
    {
        MessageBox.Show("Sửa thất bại: " + ex.Message);
        return -1;
    }
}

```

Hình: 203

– Hàm Delete

```
1 reference
public int Delete_Employee(int empno_in)
{
    try
    {
        string Procedure = "employee.pro_delete_employee";

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

        OracleParameter empno = new OracleParameter();
        empno.ParameterName = "@empno_in";
        empno.OracleDbType = OracleDbType.Int32;
        empno.Value = empno_in;
        empno.Direction = ParameterDirection.Input;
        cmd.Parameters.Add(empno);

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

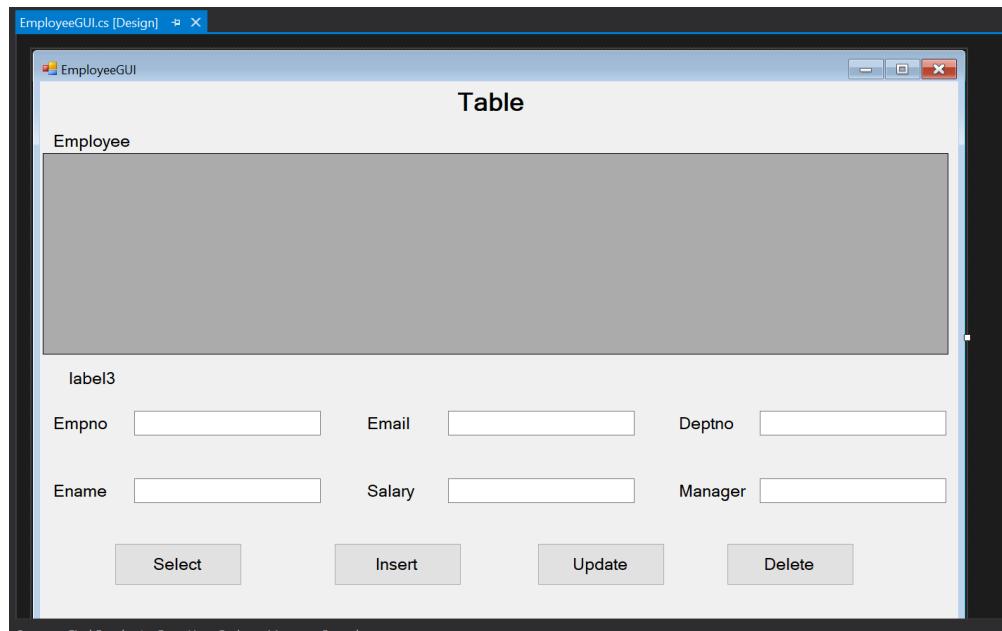
        cmd.ExecuteNonQuery();

        if (int.Parse(ex.Value.ToString()) == 0)
        {
            MessageBox.Show("0 dòng được xóa");
            return -1;
        }

        return 0;
    }
    catch (OracleException ex)
    {
        MessageBox.Show("Xóa thất bại: " + ex.Message);
        return -1;
    }
}
```

Hình: 204

- Thiết kế giao diện bảng Employee



Hình: 205

- Khai báo các biến cần thiết

```

EmployeeGUI.cs  ✎ X EmployeeGUI.cs [Design]
C# Bai4_6
  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 Bai4_6
14  {
15      public partial class EmployeeGUI : Form
16      {
17          OracleConnection conn;
18          Employee employee;
19          public EmployeeGUI()
20          {
21              InitializeComponent();
22              CenterToScreen();
23
24              conn = Database.Get_Connect();
25              employee = new Employee(conn);
26          }
27      }

```

Hình: 206

- Cài đặt sự kiện chọn bảng

```
1 reference
private void dtg_table_SelectionChanged(object sender, EventArgs e)
{
    int row = dtg_table.CurrentCell.RowIndex;

    txt_empno.Text = dtg_table.Rows[row].Cells[0].Value.ToString();
    txt_ename.Text = dtg_table.Rows[row].Cells[1].Value.ToString();
    txt_email.Text = dtg_table.Rows[row].Cells[2].Value.ToString();
    txt_salary.Text = dtg_table.Rows[row].Cells[3].Value.ToString();
    txt_deptno.Text = dtg_table.Rows[row].Cells[4].Value.ToString();
    txt_manager.Text = dtg_table.Rows[row].Cells[5].Value.ToString();
}
```

Hình: 207

- Cài đặt nút Select

```
1 reference
private void btn_select_Click(object sender, EventArgs e)
{
    dtg_table.DataSource = employee.Select_employee();
}
```

Hình: 208

- Cài đặt nút Insert

```
1 reference
private void btn_insert_Click(object sender, EventArgs e)
{
    int empno = int.Parse(txt_empno.Text);
    string ename = txt_ename.Text;
    string email = txt_email.Text;
    int salary = int.Parse(txt_salary.Text);
    int deptno = int.Parse(txt_deptno.Text);
    int manager = -1;
    if (!txt_manager.Text.Equals(""))
        manager = int.Parse(txt_manager.Text);

    int i = employee.Insert_Employee(empno, ename, email, salary, deptno, manager);
    if (i == 0)
    {
        MessageBox.Show("Thêm thành công");
    }
}
```

Hình: 209

- Cài đặt nút Update

```

    private void btn_update_Click(object sender, EventArgs e)
    {
        int empno = int.Parse(txt_empno.Text);
        string ename = txt_ename.Text;
        string email = txt_email.Text;
        int salary = int.Parse(txt_salary.Text);
        int deptno = int.Parse(txt_deptno.Text);
        int manager = -1;
        if (!txt_manager.Text.Equals(""))
            manager = int.Parse(txt_manager.Text);

        int i = employee.Update_Employee(empno, ename, email, salary, deptno, manager);
        if (i == 0)
        {
            MessageBox.Show("Sửa thành công");
        }
    }
}

```

Hình: 210

- Cài đặt nút Delete

```

    1 reference
    private void btn_delete_Click(object sender, EventArgs e)
    {
        int empno = int.Parse(txt_empno.Text);

        int i = employee.Delete_Employee(empno);
        if (i != -2)
        {
            MessageBox.Show("Xóa thành công");
        }
    }
}

```

Hình: 211

- Sự kiện đóng form

```

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

```

Hình: 212

6.1. Nhân viên hay quản lý thuộc phòng ban này không được phép xem hay chỉnh sửa (Insert, Update, Delete) bất kỳ thông tin nào của những nhân viên thuộc phòng ban khác.

- Tạo function cho chính sách

```

34 --connect SEC_MGR--
35 create or replace function fun_EmployeeControl1(p_schema in varchar2, p_object in varchar2)
36 return varchar2
37 is
38   f_deptno number;
39 begin
40   select deptno into f_deptno from SEC_MGR.employee where upper(ename) = user;
41
42   return 'deptno = ' || f_deptno;
43
44 exception when others then
45   return '1 = 0';
46 end;

```

Function FUN_EMPLOYEECONTROL1 compiled

Hình: 213

- Lệnh tạo chính sách EmployeeControl1

```

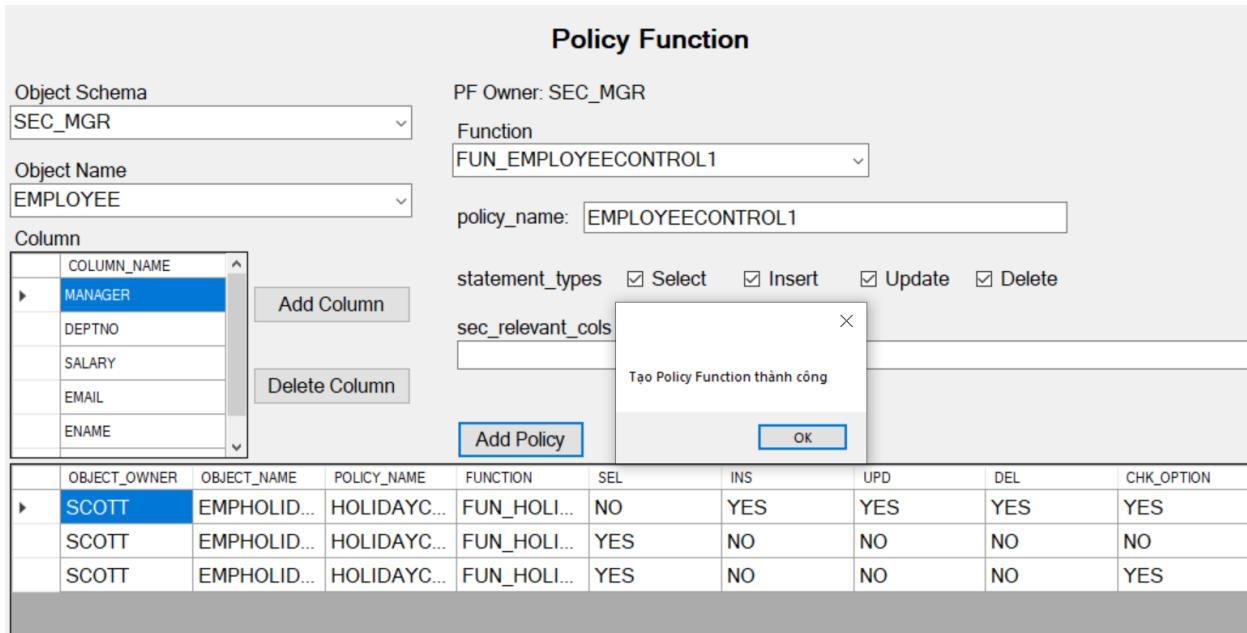
48 --connect SEC_MGR--
49 begin
50   dbms_rls.add_policy
51   (
52     object_schema => 'SEC_MGR',
53     object_name => 'Employee',
54     policy_name => 'EmployeeControl1',
55     policy_function => 'fun_EmployeeControl1',
56     statement_types => 'SELECT, INSERT, UPDATE, DELETE',
57     update_check => TRUE
58   );
59 end;

```

PL/SQL procedure successfully completed.

Hình: 214

- Tạo Policy bằng form



Hình: 215

- Test chính sách trên user của quản lý

```

61 --connect Abel--
62 select * from SEC_MGR.employee where Deptno = 2;

```

Script Output | Query Result | All Rows Fetched: 0 in 0.021 seconds

EMPNO	ENAME	EMAIL	SALARY	DEPTNO	MANAGER
-------	-------	-------	--------	--------	---------

Hình: 216

- Đăng nhập vào form và thực hiện truy vấn

label3

Empno Email Deptno

Ename Salary Manager

Hình: 217

- Thực hiện Update

```
64 update SEC_MGR.employee set salary = 7000 where empno = 2;
Script Output X | Query Result X
Task completed in 0.043 seconds
0 rows updated.
```

Hình: 218

Abel

Empno Email Deptno

Ename Salary Manager

Hình: 219

6.2. Nhân viên thuộc phòng ban nào chỉ được xem (select) thông tin của các nhân viên thuộc cùng phòng ban với mình ngoại trừ lương (salary). Mỗi nhân viên chỉ có thể xem lương của bản thân họ.

- Ở chính sách này điều kiện trả về của function không được gọi các hàm xử lý chuỗi nên ta truy vấn mã nhân viên rồi đặt làm điều kiện trả về

```

66 --connect SEC_MGR--
67 create or replace function fun_EmployeeControl2(p_schema in varchar2, p_object in varchar2)
68 return varchar2
69 is
70   n number;
71   f_empno number;
72 begin
73   select count(*) into n from SEC_MGR.employee where upper(ename) = user and manager is not null;
74
75   select empno into f_empno from SEC_MGR.employee where upper(ename) = user;
76
77 if n = 1 then
78   return 'empno = ' || f_empno;
79 else
80   return '1 = 0';
81 end if;
82
83 exception when others then
84   return '1 = 0';
85 end;

```

Hình: 220

- Tạo chính sách EmployeeControl2 quản lý việc Select trên cột lương(salary). Thêm biến đầu vào sec_relevant_col_opt= DBMS_RLS.all_rows: truy vấn tất cả các cột trừ cột không được truy vấn (trả về rỗng hoặc null).

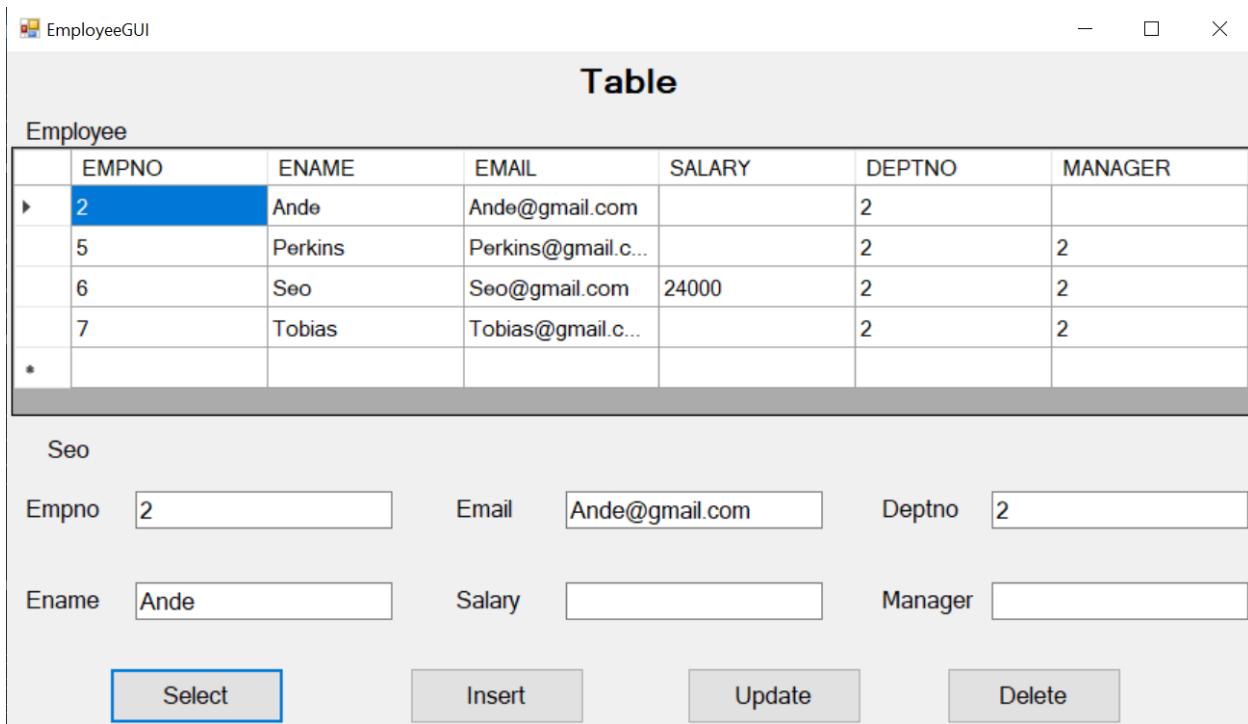
```

82 --connect SEC_MGR--
83 begin
84   dbms_rls.add_policy
85   (
86     object_schema => 'SEC_MGR',
87     object_name => 'Employee',
88     policy_name => 'EmployeeControl2',
89     policy_function => 'fun_EmployeeControl2',
90     statement_types => 'SELECT',
91     sec_relevant_cols => 'salary',
92     sec_relevant_cols_opt => DBMS_RLS.all_rows
93   );
94 end;

```

Hình: 221

- Đăng nhập user của nhân viên vào form và xem kết quả



Hình: 222

6.3. Nhân viên không có quyền chỉnh sửa (insert, update, delete) bất cứ thông tin gì, kể cả thông tin của chính nhân viên đó.

- Tạo function trả về điều kiện chặn thao tác “1 =0”

```

101 --connect SEC_MGR--
102 create or replace function fun_EmployeeControl3(p_schema in varchar2, p_object in varchar2)
103 return varchar2
104 is
105   n number;
106 begin
107   select count(*) into n from SEC_MGR.employee where upper(ename) = user and manager is not null;
108
109  if n = 1 then
110    return '1 = 0';
111  else
112    return '1 > 1';
113  end if;
114
115  exception when others then
116    return '1 = 0';
117 end;

```

Script Output X | Query Result X
Task completed in 0.083 seconds

Function FUN_EMPLOYEECONTROL3 compiled

Hình: 223

- Tạo chính sách EmployeeControl3

```

119 --connect SEC_MGR--
120 begin
121   dbms_rls.add_policy
122   (
123     object_schema => 'SEC_MGR',
124     object_name => 'Employee',
125     policy_name => 'EmployeeControl3',
126     policy_function => 'fun_EmployeeControl3',
127     statement_types => 'Insert, Update, Delete'
128   );
129 end;
130

```

Script Output X | Query Result X
X X | Task completed in 0.051 seconds

PL/SQL procedure successfully completed.

Hình: 224

Policy Function

Object Schema	PF Owner: SEC_MGR							
SEC_MGR	Function							
Object Name	FUN_EMPLOYEECONTROL3							
EMPLOYEE	policy_name: EMPLOYEECONTROL3							
Column	statement_types <input type="checkbox"/> Select <input checked="" type="checkbox"/> Insert <input checked="" type="checkbox"/> Update <input checked="" type="checkbox"/> Delete							
<table border="1"> <tr><th>COLUMN_NAME</th></tr> <tr><td>MANAGER</td></tr> <tr><td>DEPTNO</td></tr> <tr><td>SALARY</td></tr> <tr><td>EMAIL</td></tr> <tr><td>ENAME</td></tr> </table>	COLUMN_NAME	MANAGER	DEPTNO	SALARY	EMAIL	ENAME	sec_re Tạo Policy Function thành công	
COLUMN_NAME								
MANAGER								
DEPTNO								
SALARY								
EMAIL								
ENAME								
	Add OK							
OBJECT_OWNER	OBJECT_NAME	POLICY_NAME	FUNCTION	SEL	INS	UPD	DEL	CHK_OPTION
SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLI...	NO	YES	YES	YES	YES
SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLI...	YES	NO	NO	NO	NO
SCOTT	EMPHOLID...	HOLIDAYC...	FUN_HOLI...	YES	NO	NO	NO	YES
SEC_MGR	EMPLOYEE	EMPLOYEE...	FUN_EMPL...	YES	YES	YES	YES	YES
SEC_MGR	EMPLOYEE	EMPLOYEE...	FUN_EMPL...	YES	NO	NO	NO	NO

Hình: 225

- Đăng nhập vào user nhân viên tiến hành cập nhật và thêm dữ liệu

I adie

	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	MANAGER
▶	2	Ande	Ande@gmail.com		2	
	5	Perkins	Perkins@gmail.c...		2	2
	6				2	2
*	7				2	2
Thêm thất bại: ORA-28115: policy with check option violation ORA-06512: at "EMPLOYEE.PRO_INSERT_EMPLOYEE", line 5 ORA-06512: at line 1						
<input type="button" value="OK"/>						
Seo	Empno	9	Email	Seo2@gmail.com	Deptno	2
Ename	Ename	Seo	Salary	123	Manager	
<input type="button" value="Select"/>		<input type="button" value="Insert"/>	<input type="button" value="Update"/>		<input type="button" value="Delete"/>	

Hình: 226

I table

	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	MANAGER
	2	Ande	Ande@gmail.com		2	
	5	Perkins	Perkins@gmail.c...		2	2
▶	6	Seo	Seo@gmail.com	24000	2	2
*	7	Tobias	Tobias@...		2	2
0 dòng được sửa						
<input type="button" value="OK"/>						
Seo	Empno	6	Email	Seo@gmail.com	Deptno	2
Ename	Ename	Seo	Salary	25000	Manager	2
<input type="button" value="Select"/>		<input type="button" value="Insert"/>	<input type="button" value="Update"/>		<input type="button" value="Delete"/>	

Hình: 227

- Xóa dữ liệu

Table

Employee

	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	MANAGER
	2	Ande	Ande@gmail.com		2	
	5	Perkins	Perkins@gmail.c...		2	2
▶	6	Seo	Seo@gmail.com	24000	2	2
	7	Tobias	Tobias@...		2	2
*						

0 dòng được xóa

OK

Seo

Empno Email Deptno

Ename Salary Manager

Select Insert Update Delete

Hình: 228

6.4. Chỉ có người quản lý từng phòng ban được phép select, insert, update, delete tất cả các thông tin của các nhân viên thuộc phòng ban mình quản lý.

- Vì hai chính sách EmployeeControl2 và EmployeeControl3 đều viết cho nhân viên nên ta có thể chỉnh sửa chính sách dành cho cả nhân viên và quản lý.
- Chính sửa function của chính sách EmployeeControl2

```

132 --connect SEC_MGR--
133 create or replace function fun_EmployeeControl2(p_schema in varchar2, p_object in varchar2)
134 return varchar2
135 is
136   n number;
137   f_empno number;
138 begin
139   select count(*) into n from SEC_MGR.employee where upper(ename) = user and manager is not null;
140
141   select empno into f_empno from SEC_MGR.employee where upper(ename) = user;
142
143   if n = 1 then
144     return 'empno = ' || f_empno;
145   else
146     return '';
147   end if;
148
149   exception when others then
150     return '1 = 0';
151 end;

```

Script Output | Query Result | Task completed in 0.077 seconds

Function FUN_EMPLOYEECONTROL2 compiled

Hình: 229

- Chính sửa function của chính sách EmployeeControl3

```

153 --connect SEC_MGR--
154 create or replace function fun_EmployeeControl3(p_schema in varchar2, p_object in varchar2)
155 return varchar2
156 is
157   n number;
158 begin
159   select count(*) into n from SEC_MGR.employee where upper(ename) = user and manager is not null;
160
161 if n = 1 then
162   return '1 = 0';
163 else
164   return '';
165 end if;
166
167 exception when others then
168   return '1 = 0';
169 end;
170

```

Script Output: X | Query Result: X
Task completed in 0.046 seconds

Function FUN_EMPLOYEECONTROL3 compiled

Hình: 230

- Đăng nhập vào user của quản lý và kiểm tra kết quả

Table						
Employee						
	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	MANAGER
▶	1	Abel	Abel@gmail.com	2500	1	
	3	Ernst	Ernst@gmail.com	11000	1	1
	4	Khoo	Khoo@gmail.com	7400	1	1
*	8	Patel	gmail.com	24000	1	1

Thêm thành công

OK

Abel

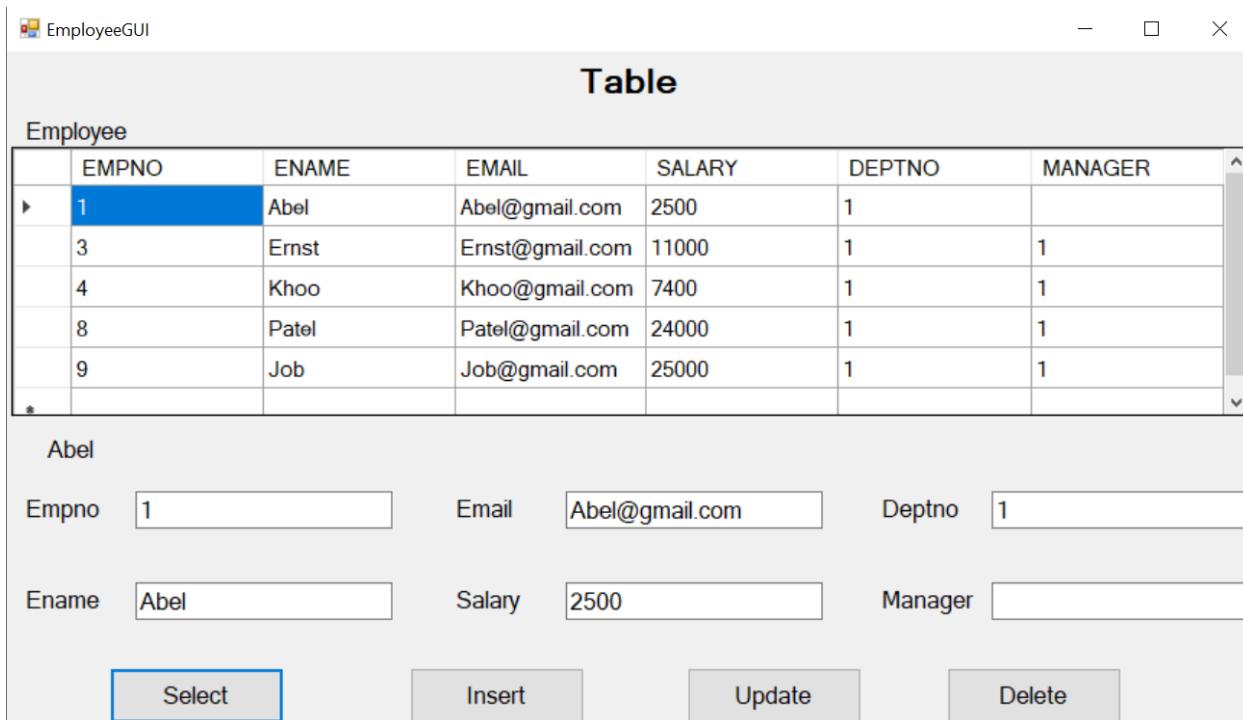
Empno 9 Email Job@gmail.com Deptno 1

Ename Job Salary 25000 Manager 1

Select Insert Update Delete

Hình: 231

- Truy vấn các nhân viên trong phòng

EmployeeGUI

Table

Employee

	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	MANAGER
▶	1	Abel	Abel@gmail.com	2500	1	
	3	Ernst	Ernst@gmail.com	11000	1	1
	4	Khoo	Khoo@gmail.com	7400	1	1
	8	Patel	Patel@gmail.com	24000	1	1
*	9	Job	Job@gmail.com	25000	1	1
*						

Abel

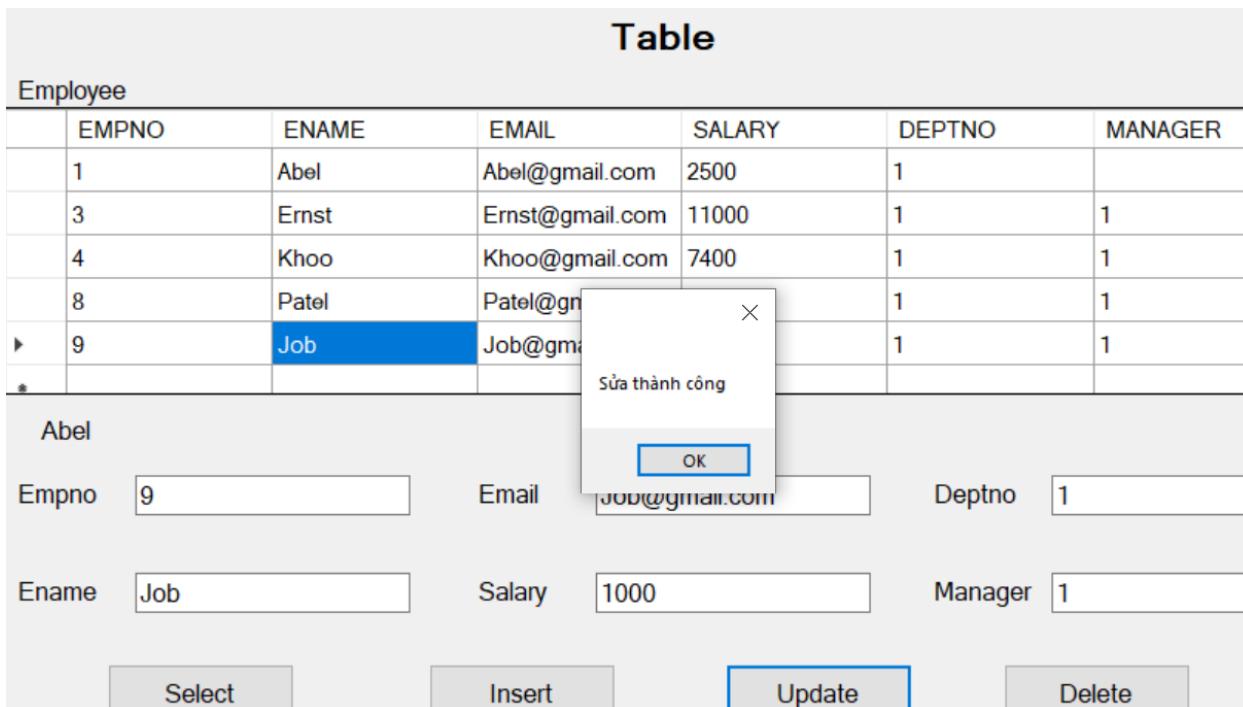
Empno Email Deptno

Ename Salary Manager

Select **Insert** **Update** **Delete**

Hình: 232

- Quản lý cập nhật thông tin nhân viên

EmployeeGUI

Table

Employee

	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	MANAGER
	1	Abel	Abel@gmail.com	2500	1	
	3	Ernst	Ernst@gmail.com	11000	1	1
	4	Khoo	Khoo@gmail.com	7400	1	1
	8	Patel	Patel@gmail.com		1	1
▶	9	Job	Job@gmail.com		1	1
*						

Abel

Empno Email Deptno

Ename Salary Manager

Select **Insert** **Update** **Delete**

Sửa thành công

OK

Hình: 233

- Quản lý xóa nhân viên

Table

Employee						
	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	MANAGER
	1	Abel	Abel@gmail.com	2500	1	
	3	Ernst	Ernst@gmail.com	11000	1	1
	4	Khoo	Khoo@gmail.com	7400	1	1
	8	Patel	Patel@gmail.com	24000	1	1
▶	9	Job	Job@gmail.com	1000	1	1
*						

Abel

Empno	<input type="text" value="9"/>	Email	<input type="text" value="Job@gmail.com"/>	Deptno	<input type="text" value="1"/>
Ename	<input type="text" value="Job"/>	Salary	<input type="text" value="1000"/>	Manager	<input type="text" value="1"/>

Xóa thành công

×

Hình: 234

- Truy vấn lại dữ liệu

Table

Employee						
	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	MANAGER
▶	1	Abel	Abel@gmail.com	2500	1	
	3	Ernst	Ernst@gmail.com	11000	1	1
	4	Khoo	Khoo@gmail.com	7400	1	1
	8	Patel	Patel@gmail.com	24000	1	1
*						

Abel

Empno	<input type="text" value="1"/>	Email	<input type="text" value="Abel@gmail.com"/>	Deptno	<input type="text" value="1"/>
Ename	<input type="text" value="Abel"/>	Salary	<input type="text" value="2500"/>	Manager	<input type="text"/>

Hình: 235

- Đăng nhập vào tài khoản nhân viên và sửa thử thông tin (vì đã có chính sách chặn thao tác Insert, Delete ở yêu cầu trước nên chỉ thao tác trên trường hợp Update)

Table

Employee

	EMPNO	ENAME	EMAIL	SALARY	DEPTNO	MANAGER
	2	Ande	Ande@gmail.com		2	
	5	Perkins	Perkins@gmail.c...		2	2
▶	6	Seo	Seo@gmail.com	24000	2	2
*	7	Tobias	Seo@gmail.c...		2	2

X

0 dòng được sửa

OK

Seo					
Empno	<input type="text" value="6"/>	Email	<input type="text" value="Seo@gmail.com"/>	Deptno	<input type="text" value="2"/>
Ename	<input type="text" value="Seo"/>	Salary	<input type="text" value="100"/>	Manager	<input type="text" value="2"/>
<input type="button" value="Select"/>		<input type="button" value="Insert"/>	<input type="button" value="Update"/>	<input type="button" value="Delete"/>	

Hình: 236