

برمجة صلاحيات المستخدمين

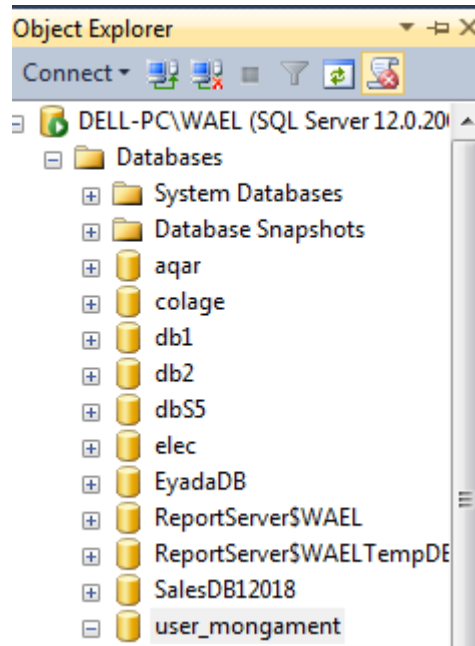
باستخدام لغة البرمجة C# 2012 وقواعد البيانات SQL Server 2014

سنتعلم في هذا الدرس النقاط والمواضيع التالية

- انشاء قاعدة البيانات اذا لم تكن موجودة وتسميتها
- انشاء الجداول الخاصة بالمستخدمين والصلاحيات
- ربط العلاقات بين الجداول مع ضبط مجموعة من الخصائص اثناء الربط
- ادخال بيانات القوائم في جدول القوائم وبيانات الواجهات في جدول الواجهات
- ادخال بيانات المستخدمين والتأكد بيانات جدول الصلاحيات
- برمجة واجهة المستخدم (اضافة - حذف - تعديل)

أولاً إنشاء قاعدة البيانات في برنامج SQL Server 2014

بعد إنشاء قاعدة البيانات باسم user_mongament تأتي الخطوة التالية الى إنشاء الجداول



ثانياً إنشاء الجداول

١- إنشاء جدول المستخدمين بالخصائص التالية: وتسميته t_users

	Column Name	Data Type	Allow Nulls
PK	id_user	int	<input type="checkbox"/>
	name_user	nvarchar(30)	<input checked="" type="checkbox"/>
	name_user_login	nvarchar(30)	<input checked="" type="checkbox"/>
	password_user	nvarchar(50)	<input checked="" type="checkbox"/>

٢- إنشاء جدول القوائم بالخصائص التالية: وتسميته t_menu

	Column Name	Data Type	Allow Nulls
PK	id_menu	int	<input type="checkbox"/>
	name_menu	nvarchar(30)	<input checked="" type="checkbox"/>

٣- انشاء جدول الواجهات بالخصائص التالية مع مراعاة اضافة رقم القائمة كمفتاح اجنبي :
وتسمية الجدول t_form

Column Name	Data Type	Allow Nulls
id_form	int	<input type="checkbox"/>
name_form	nvarchar(30)	<input checked="" type="checkbox"/>
id_menu	int	<input checked="" type="checkbox"/>

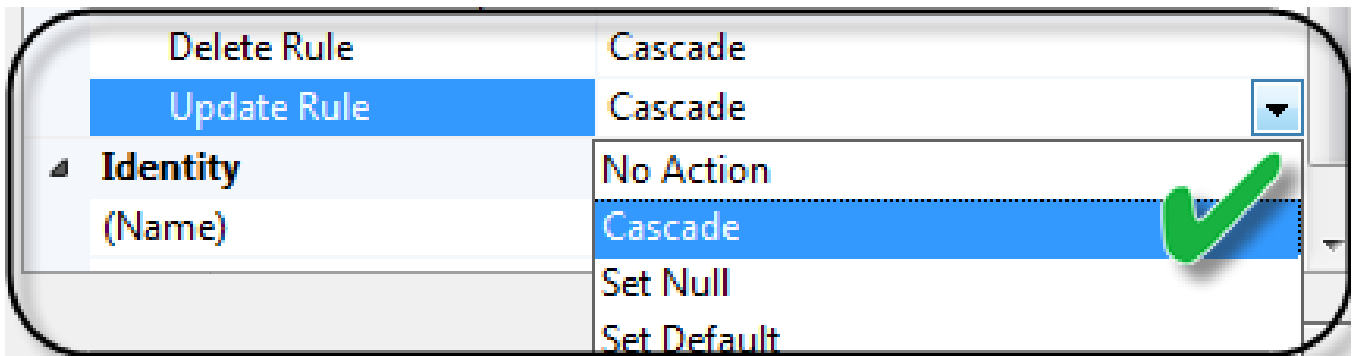
٤- سنضطر الى انشاء جدول العمليات ونقوم باضافة رقم النموذج (الواجهة) ورقم المستخدم كمفاتيح اجنبية في جدول العمليات (الصلاحيات) بحيث رقم الواجهة يحدد رقم القائمة بحكم ان رقم القائمة هو مفتاح اجنبي في جدول الواجهات وكذلك رقم المستخدم سيحدد اسم المستخدم وبالتالي سيتم تحديد الصلاحيات للمستخدم بناء على وجود رقمه كمفتاح اجنبي في جدول العمليات (الصلاحيات) . كالتالي : وتسميته t_opration

Column Name	Data Type	Allow Nulls
op_id_user	int	<input checked="" type="checkbox"/>
op_id_form	int	<input checked="" type="checkbox"/>
show_data	bit	<input checked="" type="checkbox"/>
insert_data	bit	<input checked="" type="checkbox"/>
update_data	bit	<input checked="" type="checkbox"/>
delete_data	bit	<input checked="" type="checkbox"/>

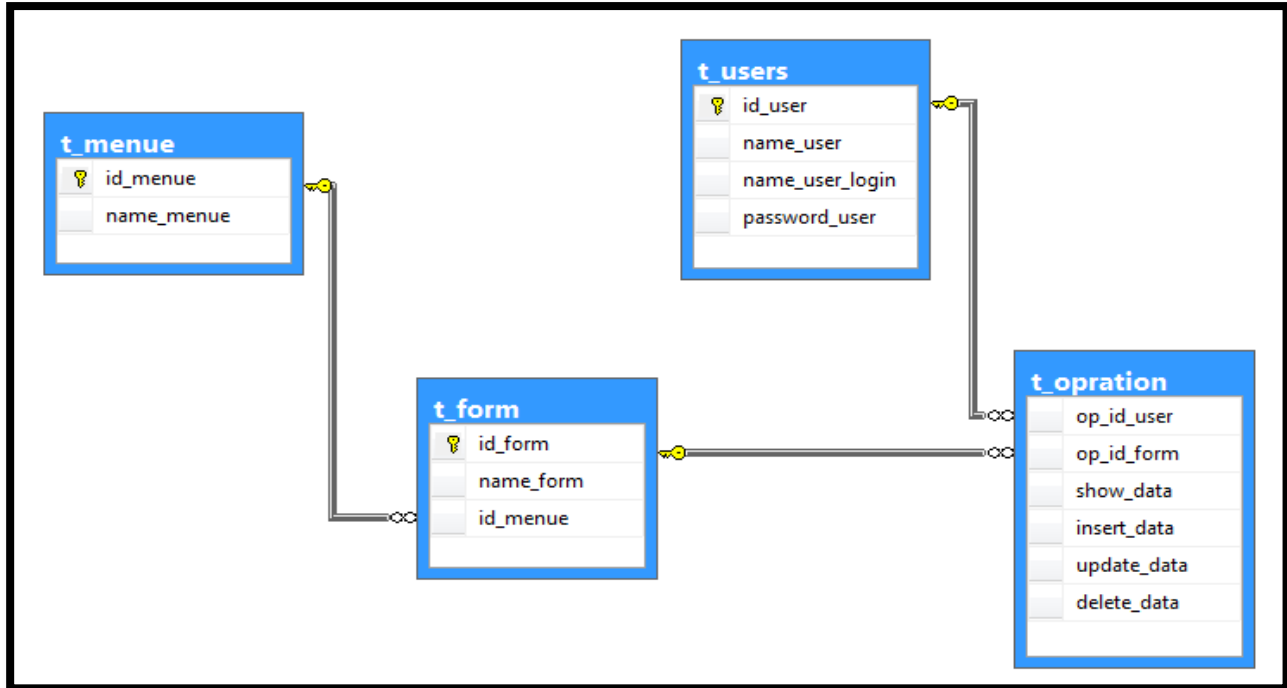
ثالثا ربط العلاقات بين الجداول مع مراعاة خاصية INSERT And UPDATE

INSERT And UPDATE Specific	
Delete Rule	No Action
Update Rule	No Action

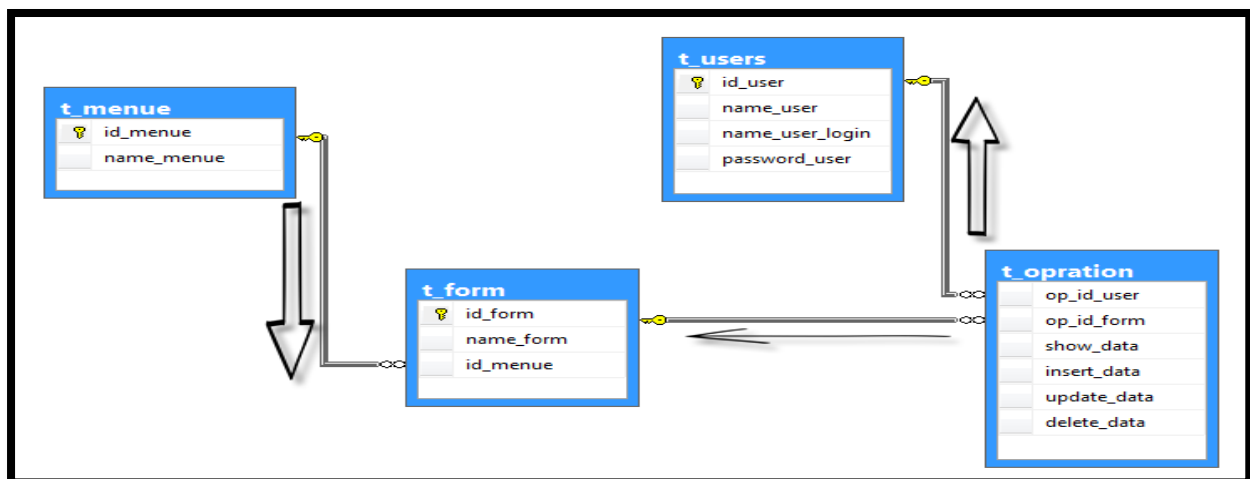
هذه الخاصية تفيد في عمليتي التعديل او الحذف بحيث اذا تم تفعيلها بالقيمة Cascade معنى ذلك اذا تم حذف مستخدم فانه يتم حذف جميع صلاحياته واذا تم تعديل بيانات المستخدم او تعديل صلاحياته فانه يتم ذلك على جميع الصلاحيات الخاصة بهذا المستخدم كالتالي :



المخطط العلائقي الخاص بربط الجداول بالعلاقات :



لاحظ ان عملية الربط بين المفاتيح تتم كالتالي رقم المستخدم (op_id_user) في جدول العمليات يتم سحبه وافلاته على رقم المستخدم (id_user) في جدول المستخدم وكذلك رقم الواجهة (op_id_form) في جدول العمليات يتم اسقاط على رقم الواجهة (op_id_form) في جدول العمليات على رقم الواجهة في جدول الواجهات (id_form) كما توضح الاسهم التاليه:



رابعا : تعبئة جدول الواجهات بالبيانات

قبل تعبئة الجدول بالبيانات (اسماء الواجهات في برنامجك) لابد من

١. انشاء الواجهة الرئيسة للبرنامج وبيانات القوائم التي تحتويها

٢. معرفة عدد الواجهات واسماءها

لنفترض ان لديك خمس واجهات (البيانات الاساسية - العمليات - التقارير)

بحيث ان واجهة البيانات الاساسية تحتوي قائمة :

(بيانات الكلية - بيانات الاقسام - بيانات المواد - بيانات المدرسين - بيانات القاعات)

وواجهة العمليات تحتوي قائمة فيها عمليات (توزيع مواد _ سحب مادة)

وواجهة التقارير تحتوي قائمة فيها تقرير (جدول عام - جدول اقسام - جدول شخصي)

البيانات الاساسية

بيانات الكلية

بيانات الاقسام

بيانات المواد

بيانات المدرسين

بيانات القاعات

العمليات

توزيع مواد

سحب مواد

التقارير

جدول عام

جدول اقسام

جدول شخصي

خامساً تعبئة الجدول بالبيانات :

نذهب الان الى جدول القوائم ونقوم بتعبئته بالبيانات التالية :

	id_menu	name_menu
	1	البيانات الاساسية
	2	العمليات
	3	التقارير

نذهب الى جدول الواجهات وتعبئته بالبيانات التالية :

	id_form	name_form	id_menu
▶	1	بيانات الكلية	1
	2	بيانات الاقسام	1
	3	بيانات المدرسين	1
	4	بيانات العواد	1
	5	بيانات القاعات	1
	6	توزيع العواد	2
	7	سحب العواد	2
	8	جدول عام	3
	9	جدول الاقسام	3
	10	جدول شخصي	3

تلاحظ تم ادخال رقم القائمة في الجدول السابق وذلك يعني اننا نحدد محتويات القائمة بناء

على رقمها فمثلا القائمة الاولى (رقم 1) تحتوي على بيانات الكلية والاقسام و...

بعد تعبئة بيانات المستخدمين لن نقوم بتعبئة جدول بيانات المستخدمين او حتى جدول

الصلاحيات لان الفكرة هنا عند ادخال مستخدم جديد لا بد من ادخال صلاحياته في جدول

العمليات وهذا لن يتم الا من خلال البرمجة

سادسا : البرمجة

الخطوة الاولى نقوم بانشاء نموذج (واجهة) يحتوي على الادوات التالية كما هو موضح

قم بتصميم الواجهة التالية مع الادوات المطلوبه بشكل دقيق

اولا نضبط بروفيدر الاتصال - المسار واسم السيرفر واسم القاعدة

```
SqlConnection cn = new SqlConnection(  
@"Data  
Source=.\WAEL;InitialCatalog=user_mongament;Integrated  
Security=true");
```

وفي حدث klick لزر الدخول في النموذج اعلاه نقوم بكتابة الكود التالي :

```
string sql = "select * from t_users where name_user_login =
'" + txt_input_name.Text + "' and password_user =
'" + txt_pass.Text + "'";
```

بعد ضبط الخصائص التالية

الاسم البرمجي	الاداة
txt_input_name.Text	TextBox1.Text
txt_pass_name.Text	TextBox2.Text

نعمل استعلام يجلب اسم المستخدم و كلمة السر فقط القيمة التي تساوي القيمة المدخلة في مربعات النص ثم بعد ذلك القيمة ستكون مخزنة في DataTable ونخزن القيمة الاولى في DataTable في متغير تم تعريفه مسبقا في Program واسم هذا المتغير هو user_id وذلك فقط من اجل عرضه في رساله بعد الضغط على زر الدخول وفيها اسم المستخدمون ثم نقوم بعرض النموذج التالي والكود التالي يوضح ذلك

```
//-----
SqlCommand cmd = new SqlCommand(sql, cn);
cn.Open();
SqlDataReader dr = cmd.ExecuteReader();
dr.Read();
if (dr.HasRows)
{
    Program.user_id = dr[0].ToString();
    MessageBox.Show("ok" +
Program.user_id.ToString());
    FrmMainAll frm = new FrmMainAll();
    frm.ShowDialog();
    this.Close();
}
else
    MessageBox.Show("not login");
cn.Close();
//-----
```


أولاً النموذج الرئيس : FrmMain_user

بعد انزال الادوات على النموذج نقوم بضبط الخصائص لجميع الادوات الظاهرة عليه وتعريف مكتبة التعامل مع SQLServer ومقبض الاتصال الذي يحوي اسم السيرفر واسم القاعدة ونكتب الكود التالي :

١- برمجة زر الاضافة

```
try {
    string sql_user_validiati = "insert into
t_users(id_user,name_user,name_user_login,password_user) values(" +
Convert.ToInt32(txt_id.Text) + "," + txt_name.Text + "','" +
txt_input_name.Text + "','" + txt_pass.Text + "') " + "insert into t_opration
(op_id_form,op_id_user,show_data,insert_data,update_data,delete_data) select
id_form," + Convert.ToInt32(txt_id.Text) + ",1,0,1,0 from t_form";
    SqlCommand cmd = new SqlCommand(sql_user_validiati, cn);
    cn.Open();
    cmd.ExecuteNonQuery();
    cn.Close();
    MessageBox.Show("بنجاح جديد مستخدم اضافة تم");
}
catch(Exception ex)
{
    MessageBox.Show(ex.Message);
}

rev();
```

شرح الكود :

اضافة مستخدم

جملة الاستعلام `insert into` استخدمناها مرتين في نفس الجملة حيث ان `insert into` الاولى لادخال بيانات المستخدم والجملة الثانية هي لادخال بيانات جدول العمليات الذي شرحناه سابقا وكذلك جملة `select` وذلك لاختذ رقم النموذج من اجل التحكم بالصلاحيات ،

اثناء برمجة الصلاحيات نكون قد ميزنا هذا المستخدم انه يمكن ان يكون له صلاحيات الوصول للنماذج او عمليات الاضافة والحذف والتعديل التابعة للنماذج

٢- برمجة زر التعديل

تعديل بيانات المستخدم

```
try
    {
        string sql_user_validiati = "update t_users set name_user
        ='"+txt_name.Text+"',name_user_login='"+txt_input_name.Text+"',password_user=
        '"+txt_pass.Text+"' where id_user = "+
        dataGridViewX1.CurrentRow.Cells[0].Value;
        SqlCommand cmd = new SqlCommand(sql_user_validiati, cn);
        cn.Open();
        cmd.ExecuteNonQuery();
        cn.Close();
        MessageBox.Show("بنجاح المستخدم بيانات تعديل تم"+txt_name.Text);
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    rev();
```

٣- برمجة دالة التحديث للبيانات في داتا جرايد (rev() :

```
public void rev()
    {
        String sql_rev=" select * from t_users ";
        SqlDataAdapter da = new SqlDataAdapter("sql_rev ", cn);
        DataTable dt = new DataTable();
        da.Fill(dt);
        dataGridViewX1.DataSource = dt;
    }
```

حذف بيانات المستخدم

٤- برمجة زر الحذف :

```

try
{
    string sql_user_validiati = "delete t_users where id_user =
" + dataGridViewX1.CurrentRow.Cells[0].Value;
    SqlCommand cmd = new SqlCommand(sql_user_validiati, cn);
    cn.Open();
    cmd.ExecuteNonQuery();
    cn.Close();
    MessageBox.Show("بنجاح المستخدم بيانات تعديل تم" + txt_name.Text);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}

rev();

```

٥- برمجة اداة DataGridView :

اولا في حدث dataGridViewX1_CellClick وذلك من اجل عرض بيانات المستخدم في الحقول من اجل تعديلها لو حذفها وكما نعرف نريد البيانات تظهر مشفرة من اجل سرية كلمة المرور

```

try
{
    SqlDataAdapter da = new SqlDataAdapter("select * from t_users
where id_user =" + Convert.ToInt32(dataGridViewX1.CurrentRow.Cells[0].Value),
cn);

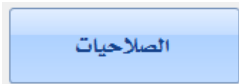
    DataTable dt = new DataTable();
    da.Fill(dt);
    //dataGridViewX1.DataSource = dt;
    txt_id.Text = dt.Rows[0][0].ToString();
    txt_name.Text = dt.Rows[0][1].ToString();
    txt_input_name.Text = dt.Rows[0][2].ToString();
    txt_pass.Text = dt.Rows[0][3].ToString();
}
catch { MessageBox.Show("فارغة قيمة عرض لايمكن"); }

```

٦- برمجة اداة DataGridView : في حدث DataGridViewX1_CellFormatting

```
if (e.ColumnIndex == 3 && e.Value != null)
{
    e.Value = new string('*', e.Value.ToString().Length);
}
```

وذلك من اجل عرض البيانات مشفرة بعلامة النجمة من اجل سرية وامن كلمة المرور



٧- برمجة زر الصلاحيات

```
FrmMain_validati frm = new FrmMain_validati();
Close();
frm.ShowDialog();
```

الانتقال الى نموذج الصلاحيات FrmMain_validati

الحذف	التعديل	الاضافة	العرض	اسم الواجهة	رقم الواجهة
[Empty Table Area]					

رقم المستخدم	اسم المستخدم
[Empty Table Area]	

خروج
منح الصلاحيات

جميع الصلاحيات

جميع الصلاحيات

جميع الصلاحيات

جميع الصلاحيات

ثانياً النموذج الفرعي : FrmMain_validati

الادوات المستخدمة على هذا النموذج

ملاحظات	الوظيفة	الاداة
	تليحث عن مستخدم	TextBox1
تحتوي على عمودين رقم المستخدم ، اسم المستخدم	عرض بيانات المستخدمين	DataGraidVeiw2
	تنفيذ منح الصلاحيات بعد تفعيلها من DataGraidVeiw1	Button1
	الخروج	Button1
تحتوي على 6 اعمدة تخص الواجهات و العمليات وكل خلية فيها عبارة CheckBox وقيمتها صفر او واحد	عرض اسماء النماذج مع العمليات لكل نموذج	DataGraidVeiw1
تمنح الصلاحيات - تحكم كامل	لمنح كامل الصلاحيات	Button,3,4,5,6

١- برمجة DataGridView2 التي تعرض أسماء المستخدمين ويتم تحميل البيانات فيها أثناء

الاقلاع ونكتب الكود في حدث load او InitializeComponent

الكود :

```
SqlDataAdapter da1 = new SqlDataAdapter("SELECT
id_user,name_user from t_users", cn);
DataTable dt1 = new DataTable();
da1.Fill(dt1);
dataGridViewX2.DataSource = dt1;
```

برمجة DataGridView2 في حدث CellClick أثناء النقر عليها

```
try
{
    dataGridViewX1.Enabled = true;
    buttonX3.Enabled = true;
    buttonX4.Enabled = true;
    buttonX5.Enabled = true;
    buttonX6.Enabled = true;
    SqlDataAdapter da = new SqlDataAdapter("SELECT
t_opration.op_id_user, t_opration.op_id_form, t_form.name_form,
t_opration.show_data, t_opration.insert_data, t_opration.update_data,
t_opration.delete_data FROM
t_form INNER JOIN t_opration ON
t_form.id_form = t_opration.op_id_form where t_opration.op_id_user= "+
dataGridViewX2.CurrentRow.Cells[0].Value, cn);
    DataTable dt = new DataTable();
    da.Fill(dt);
    dataGridViewX1.Rows.Clear();
    for (int i = 0; i < dt.Rows.Count; i++)
    {
        dataGridViewX1.Rows.Add(dt.Rows[i][1].ToString(),
dt.Rows[i][2].ToString(), dt.Rows[i][3].ToString(),
dt.Rows[i][4].ToString(), dt.Rows[i][5].ToString(),
dt.Rows[i][6].ToString());
    }
}
catch { MessageBox.Show("صلاحياته لعرض مستخدم يوجد لا"); }
```

لماذا استخدمه ١0646 جملة for لعرض البيانات في dataGridViewX1؟؟

بالامكان ان نكتب مباشرة dataGridViewX1.DataSurce=dt ولكن هذا لايفيد فسر ذلك؟؟

٢- برمجة DataGridView التي تعرض أسماء الواجهات والعمليات الأربعة

(العرض - الأضافة - التعديل - الحذف) ويتم تحميل البيانات فيها أثناء الأقلاب ونكتب

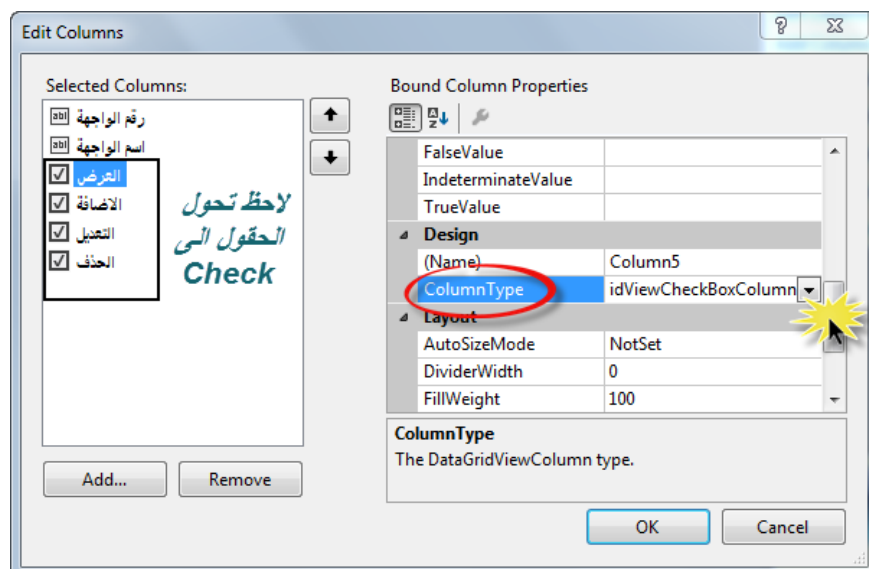
الكود في حدث load او InitializeComponent

الكود :

```

SqlDataAdapter da = new SqlDataAdapter("SELECT
t_opration.op_id_user, t_opration.op_id_form,
t_form.name_form, t_opration.show_data,
t_opration.insert_data, t_opration.update_data,
t_opration.delete_data FROM          t_form INNER JOIN
t_opration ON t_form.id_form = t_opration.op_id_form", cn);
        DataTable dt = new DataTable();
        da.Fill(dt);
        for (int i = 0; i < dt.Rows.Count; i++)
        {
            dataGridViewX1.Rows.Add(dt.Rows[i][1].ToString(),
                dt.Rows[i][2].ToString(),
            dt.Rows[i][3].ToString(),
                dt.Rows[i][4].ToString(),
            dt.Rows[i][5].ToString(),
                dt.Rows[i][6].ToString());
        }
    
```

نقوم بضبط اداة dataGridViewX1 لحقوق الصلاحيات الأربعة كالتالي



٣- برمجة Button1 الذي يحفظ منح وتعديل الصلاحيات

اولا نعرف المتغيرات :

```
public int op_id_users;
public int op_id_forms;
```



```
op_id_users=Convert.ToInt32(dataGridViewX2.CurrentRow.Cells[0].Value.ToString());
try
{
    for (int j = 0; j < dataGridViewX1.Rows.Count; j++)
    {
        op_id_forms =Convert.ToInt32
(dataGridViewX1.Rows[j].Cells[0].Value.ToString());

        string sh =
dataGridViewX1.Rows[j].Cells[2].Value.ToString() == "True" ? "1" : "0";
string ins = dataGridViewX1.Rows[j].Cells[3].Value.ToString() == "True" ?
"1" : "0";

        string upd =
dataGridViewX1.Rows[j].Cells[4].Value.ToString() == "True" ? "1" : "0";
        string del =
dataGridViewX1.Rows[j].Cells[5].Value.ToString() == "True" ? "1" : "0";

string sql = "update t_opration set show_data='" + sh + "',insert_data='" +
ins + "',update_data='" + upd + "',delete_data='" + del + "'where op_id_user
=" + Convert.ToInt32(op_id_users) + "and op_id_form=" +
Convert.ToInt32(op_id_forms);

        SqlCommand cmd = new SqlCommand(sql, cn);
        cn.Open();
        cmd.ExecuteNonQuery();
        cn.Close();

    }
    MessageBox.Show("الصلاحيات منح تم");
}
catch (Exception ex) { MessageBox.Show(ex.Message); }
```


٤- برمجة Button3,4,5,6 الذي يمنح جميع الصلاحيات حسب عمود الصلاحيات في داتا جرايد

Button3

```
for (int j = 0; j < dataGridViewX1.Rows.Count; j++)
{
    dataGridViewX1.Rows[j].Cells[2].Value= "True";
}
```

Button4

```
for (int j = 0; j < dataGridViewX1.Rows.Count; j++)
{
    dataGridViewX1.Rows[j].Cells[3].Value= "True";
}
```

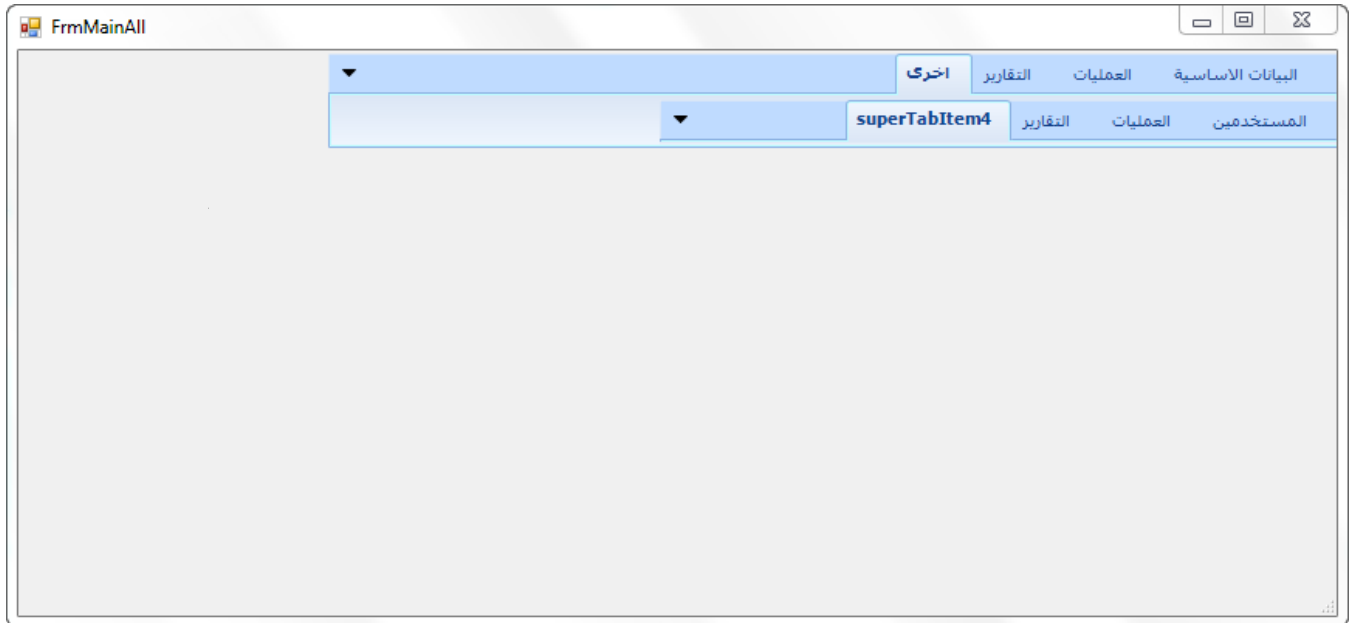
Button5

```
for (int j = 0; j < dataGridViewX1.Rows.Count; j++)
{
    dataGridViewX1.Rows[j].Cells[4].Value= "True";
}
```

Button6

```
for (int j = 0; j < dataGridViewX1.Rows.Count; j++)
{
    dataGridViewX1.Rows[j].Cells[5].Value= "True";
}
```

عند تنفيذ المشروع السابق : ليكن لدينا واجهة رئيسية ونعتبرها واجهة النظام كالتالي:



طبعا هذي الواجهة سيتم من خلالها التنقل بين النماذج

ولتنفيذ تجربة تعطيل او اخفاء احد عناصر القائمة لابد ان نقوم اولا بالتالي :

- 1- تنفيذ استعلام تجريبي من جدول العمليات وليكن استعلام يعرض قيمة الحقل show-data عندما يكون رقم المستخدم يساوي 1 ويعني ذلك اننا لابد من اخذ رقم المستخدم اثناء عملية تسجيل الدخول ويكون الاستعلام الذي ياخذ رقم المستخدم اثناء عملية الدخول كالتالي :

```
string sql = "select * from t_users where name_user_login = '" +
txt_input_name.Text + "' and password_user = '" + txt_pass.Text + "'";
SqlCommand cmd = new SqlCommand(sql, cn);
cn.Open();
SqlDataReader dr = cmd.ExecuteReader();
dr.Read();
if (dr.HasRows)
{
```

```
Program.user_id = dr[0].ToString();
```

لازم يكون معنا متغير عام عرفناه في البروجرام الرئيسي وهو Program.user_id

```
static class Program
{
    /// <summary>
    /// The main entry point for the application.
    public static string user_id;
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Frm_login());
    }
}
```

الآن سنأتي إلى كود الاستعلام الخاص بالواجهة الرئيسية والذي سنكتبه في حدث التحميل

```
SqlDataAdapter da = new SqlDataAdapter("SELECT show_data from t_opration
where op_id_user="+Convert.ToInt32(Program.user_id), cn);
DataTable dt = new DataTable();
da.Fill(dt);
// MessageBox.Show(dt.Rows[0][0].ToString());
```

```
if (dt.Rows[0][0].ToString() == "False")
{
    superTabItem1.Enabled = false;
}
```

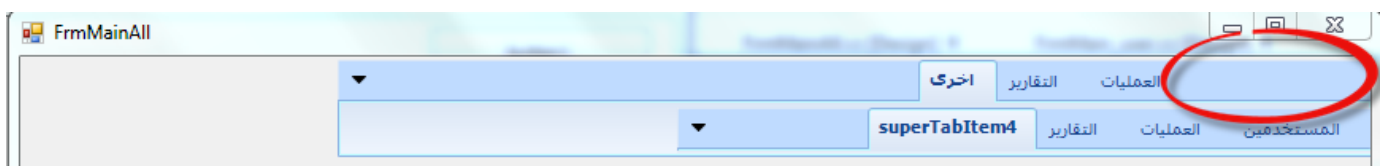
لاحظ جملة **where** تقارن الاستعلام الناتج مع رقم المستخدم الذي سجل الدخول

```
if (dt.Rows[0][0].ToString() == "False")
{
    superTabItem1.Enabled = false;
}
```

هذا الكود يفحص القيمة الموجودة في DataTable التي سيكون فيها قيمة وحدة فقط وهي ناتج استعلام عن **show_data** المخزنة في **dt.Rows[0][0]** هل قيمتها **False** أو **True** إذا كانت **False** يقوم بتجميد الاداة **superTabItem1.Enabled = false;** يعني غير مفعلة وعلى ضوء ذلك تنفذ الامر لجميع القوائم او الادوات الموجودة في النموذج الرئيسي او الفرعي



غير مفعلة يعني ظهر الاداة القائمة (البيانات الرئيسية) **superTabItem1.Enabled = false;** غير مفعلة



ملاحظة عند ما تأخذ رقم المستخدم وتخزنه في متغير اثناء تسجيل الدخول هذا سيخدمك كثيرا وخصوصا في موضوع حركات المستخدمين

الى هنا نكون قد وقفنا بفضل الله عزوجل في اتمام موضوع صلاحيات المستخدمين بناء على طلب طلاب المستوى الرابع قسم علوم وتقنية الحاسوب - كلية العلوم و الادارة بيت الفقيه وجميع الطلاب الذين طلبو تنفيذ لك بمختلف الجامعات على الصعيد العربي

اليمن جامعة الحديدة ٨-١٢-٢٠٢٠م

هذا المرجع مجاني لوجه الله الكريم لكل من يرغب وان شاء الله يكون صدقه جارية ولا تنسوننا من الدعاء
أ.وائل قاسم سطیح

Email: walwalqassim@gmail.com

Tel: +96733778123

الدرس مشروح بالفيديو بامكانك اقتناءه

الشكر موصول للذين ساندوني ودعموني بالمعلومات ولم يبخلو علينا وكانو سندا حقيقيا لنا وخص بالذكر هنا

أ.م.د.ظاهر الرشاحي

م.طارق السبيل

أ.وليد الشراعي

أ.يحيى مهدي

م.خالد السعداني

الحقيقة استفدت منهم كثيرا في بدايت مشواري للبرمجة المتقدمة باستخدام لغة البرمجة C# 2012 وقواعد البيانات SQL Server 2014

والفضل بعد الله عزوجل هو

للدكتور يحيى فلوس عميد الكلية

الذي منحني الثقة والدعم والتشجيع

ولا انسى استاذي الفاضل الدكتور جميل راشد الذي مازال يعطي ولا يبخل