**Warehouse**

**base**

**Insert into sql server**

SET IDENTITY\_INSERT base\_geography\_zone ON

delete from base\_geography\_zone

INSERT INTO base\_geography\_zone (id, "name") VALUES

(1, N'جابر الاحمد'),

(2, N'تيماء'),

(5, N'القصر'),

(6, N'النسيم'),

(7, N'الواحه'),

(9, N'العيون'),

SET IDENTITY\_INSERT base\_geography\_zone ON

**Custom (Save changes) Button - ASPXGridView**

Place our ASPxButton control on the page, set the AutoPostBack property to false, and handle the client-side Click event to call the [ASPxClientGridView.UpdateEdit](https://documentation.devexpress.com/AspNet/DevExpressWebASPxGridViewScriptsASPxClientGridView_UpdateEdittopic.aspx) method. Alternatively, it's possible to utilize the approach from the [ASPxGridView - New values are not saved on an external button callback when SettingsEditing.Mode is Batch](https://www.devexpress.com/Support/Center/p/Q553776.aspx) thread. Note that there is no capability to prevent a callback if data wasn't changed. We're working on this task in the context of the  [ASPxGridView / Batch Editing - Provide the capability to determine whether or not grid cells have been modified since the last saving action](https://www.devexpress.com/Support/Center/p/Q465699.aspx) thread.

**Hide (Save changes) button**

Aassign a CSS class to the grid's StatusBar:

<Styles>

<StatusBar CssClass="statusBar">

</StatusBar>

</Styles>

Then, add the following style on the page:

.statusBar a:first-child

{

display: none;

}

**Order and Order\_line (Master-Detail)**

How insert a default value (order\_id)

protected void ASPxGridView1\_RowInserting(object sender, DevExpress.Web.Data.ASPxDataInsertingEventArgs e)

{

usersDS.InsertParameters["age"].DefaultValue = "99";

**How to insert Data in SQL server via SqlDataSource in ASP.NET**

How to insert Data in SQL server via SqlDataSource in ASP.NET

1. First of all, take a new website project.
2. Insert two lables, two Textboxes and a Button.
3. Rename thier ID like = lblname,lblFName, txtName, txtFName and btnInsert.
4. Create database in sqlServer 2005 named EMPDB
5. Create the Table emp having 3 fields:
   * ID = variable type int is identity = YES
   * Name = varchar(50)
   * F\_Name = varachar(50)
6. Insert from toolbox->data->sqlDataSource.
7. Rename sqlDataSource to SqlDS.
8. Now double click on button and write the following code:

Hide   Copy Code

protected void btnInsert\_Click(object sender, EventArgs e)

{

SqlDS.InsertCommandType = SqlDataSourceCommandType.Text;

SqlDS.InsertCommand = "Insert into emp (Name,F\_Name) VALUES (@Name, @F\_Name)";

SqlDS.InsertParameters.Add("Name", txtName.Text);

SqlDS.InsertParameters.Add("F\_Name", txtFName.Text);

SqlDS.Insert();

txtName.Text = "";

txtFName.Text = "";

}

AccessDataSource1.InsertParameters["ProductName"].DefaultValue = ASPxFormLayout1.GetNestedControlValueByFieldName("ProductName").ToString();

AccessDataSource1.InsertParameters["UnitPrice"].DefaultValue = ASPxFormLayout1.GetNestedControlValueByFieldName("UnitPrice").ToString();

AccessDataSource1.InsertParameters["UnitsOnOrder"].DefaultValue = ASPxFormLayout1.GetNestedControlValueByFieldName("UnitsOnOrder").ToString();

AccessDataSource1.Insert();

# How to: Create and Execute an SQL Statement that Returns a Single Value

SqlConnection sqlConnection1 = new SqlConnection("Data Source=.;Initial Catalog=mew\_jahra\_mssql;Integrated Security=True");

SqlCommand cmd = new SqlCommand();

cmd.CommandText = "SELECT max(id) FROM base\_material";

cmd.CommandType = System.Data.CommandType.Text;

cmd.Connection = sqlConnection1;

int myval;

sqlConnection1.Open();

myval =Int16.Parse(cmd.ExecuteScalar().ToString());

sqlConnection1.Close();

return (myval);

**Tags**

علامات

**Anything can be applied a tag.**

**Table structure**

|  |  |
| --- | --- |
| **tag\_member\_rel** | |
| **tag\_id** | **member\_id** |
| 3 | 15 |
| 4 | 16 |

|  |  |  |
| --- | --- | --- |
| **tags** | | |
| **id** | **name** | **table** |
| 3 | قسم التشغيل | partner |
| 4 | معاملات مهمة | transaction |

**Notes**

ملاحظات

**Anything can be applied a tag.**

**Table structure**

**TO BE CONTINUED**

|  |  |
| --- | --- |
| **tag\_member\_rel** | |
| **tag\_id** | **member\_id** |
| 3 | 15 |
| 4 | 16 |

|  |  |  |
| --- | --- | --- |
| **notes** | | |
| **id** | **text** | **table** |
| 3 | تم تحويل المعاملة إلى محمود | partner |
| 4 | معاملات مهمة | transaction |

**base**

**hr**

* **base\_hr\_nationality**
* **base\_hr\_degree**
* **base\_hr\_dept**
* **base\_hr\_sector**
* **base\_hr\_title**

**warehouse**

* **base\_hr\_warehouse**
* **base\_warehouse\_material**

**SQL**

|  |  |  |
| --- | --- | --- |
| **Data type** | **Range** | **Storage** |
| **bigint** | -2^63 (-9,223,372,036,854,775,808) to 2^63-1 (9,223,372,036,854,775,807) | 8 Bytes |
| **int** | -2^31 (-2,147,483,648) to 2^31-1 (2,147,483,647) | 4 Bytes |
| **smallint** | -2^15 (-32,768) to 2^15-1 (32,767) | 2 Bytes |
| **tinyint** | 0 to 255 | 1 Byte |

**Date and Time**

|  |  |
| --- | --- |
| **SQL Server data type** | **Description** |
| **date** | The **date** data type has a range of January 1, 01 through December 31, 9999 with an accuracy of 1 day. The default value is January 1, 1900. The storage size is 3 bytes. |
| **time** | The **time** data type stores time values only, based on a 24-hour clock. The **time** data type has a range of 00:00:00.0000000 through 23:59:59.9999999 with an accuracy of 100 nanoseconds. The default value is 00:00:00.0000000 (midnight). The **time** data type supports user-defined fractional second precision, and the storage size varies from 3 to 6 bytes, based on the precision specified. |
| **datetime2** | The **datetime2** data type combines the range and precision of the **date** and **time** data types into a single data type.  The default values and string literal formats are the same as those defined in the **date** and **time** data types. |
| **datetimeoffset** | The **datetimeoffset** data type has all the features of **datetime2** with an additional time zone offset. The time zone offset is represented as [+|-] HH:MM. HH is 2 digits ranging from 00 to 14 that represent the number of hours in the time zone offset. MM is 2 digits ranging from 00 to 59 that represent the number of additional minutes in the time zone offset. Time formats are supported to 100 nanoseconds. The mandatory + or - sign indicates whether the time zone offset is added or subtracted from UTC (Universal Time Coordinate or Greenwich Mean Time) to obtain the local time. |

**Git first step**

### … create a new repository on the command line

echo # mew >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/ehabmosilhy/mew.git

git push -u origin master

### …or push an existing repository from the command line

git remote add origin https://github.com/ehabmosilhy/mew.git

git push -u origin master

**Use git with proxy**

git config --global http.proxy http://proxyuser:proxypwd@proxy.server.com:8080

git config --global https.proxy https://proxyuser:proxypwd@proxy.server.com:8080

**If you decide at any time to reset this proxy and work without (no proxy):**

git config --global --unset http.proxy

git config --global --unset https.proxy

**Finally, to check the currently set proxy;**

git config --global --get http.proxy

git config --global --get https.proxy