فصل نهم :
طریقه دستیابی و کار با داده ها در ASP.NET – قسمت دوم

مقدمه :

در این فصل مباحث تکمیلی کار با دیتابیس ها و ADO.NET مورد بررسی قرار خواهند گرفت مانند اجرای رویه های ذخیره شده ، انجام عملیات ریاضی روی ستون ها و غیره.

اجرای رویه های ذخیره شده :

استفاده از رویه های ذخیره شده در برنامه ها ، برنامه ای سریعتر و امن تر نسبت به حالتی که دستورات SQL به صورت مستقیم روی دیتابیس اجرا می شوند را ایجاد می کند. همانطور که در طی فصول پیشین در مورد نحوه ی ایجاد آنها صحبت شد ، رویه های ذخیره شده دستورات SQL و پیش کامپایل شده ای بوده و در حافظه ی سرور دیتابیس شما Cache خواهند شد. نحوه ی استفاده از آنها در ADO.NET به صورت یک مثال ارائه خواهد شد.

مثال ۱ :

یک بانک اطلاعاتی جدید به نام MyTestDB ایجاد کنید با دوجدول به صورت زیر. می خواهیم از این دو جدول گزارشی تهیه کنیم که به ازای هر ID تفاضل تعداد ورودی و تعداد فروخته شده به صورت یک ستون واحد در یک دیتا گرید نمایش داده شود. بهترین ، مطمئن ترین و سریعترین راه برای این نوع مثالها استفاده از رویه های ذخیره شده می باشد.

جدول tblEntry :

Design Table 'tblEntry' in 'myTestDB' on '(local)'

	Column Name	Data Type	Length	Allow Nulls
🔑	ID	int	4	
	Entry_No	int	4	

شکل ۱ – مشخصات جدول tblEntry در حال طراحی.

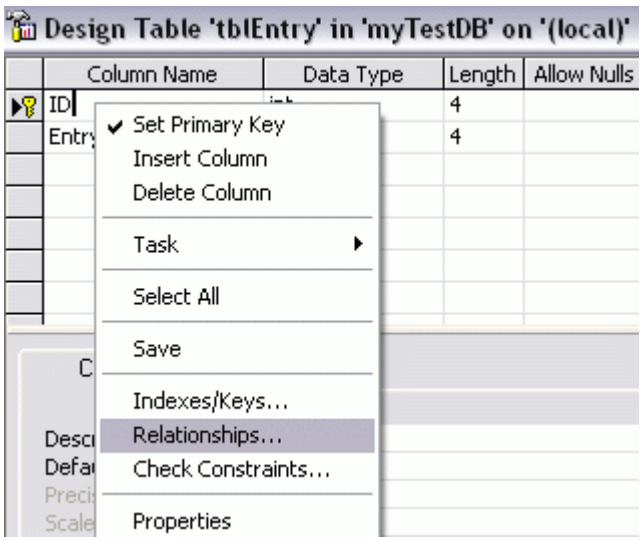
جدول tblSell :

Design Table 'tblSell' in 'myTestDB' on '(local)'

	Column Name	Data Type	Length	Allow Nulls
🔑	ID	int	4	
	Sell_No	int	4	

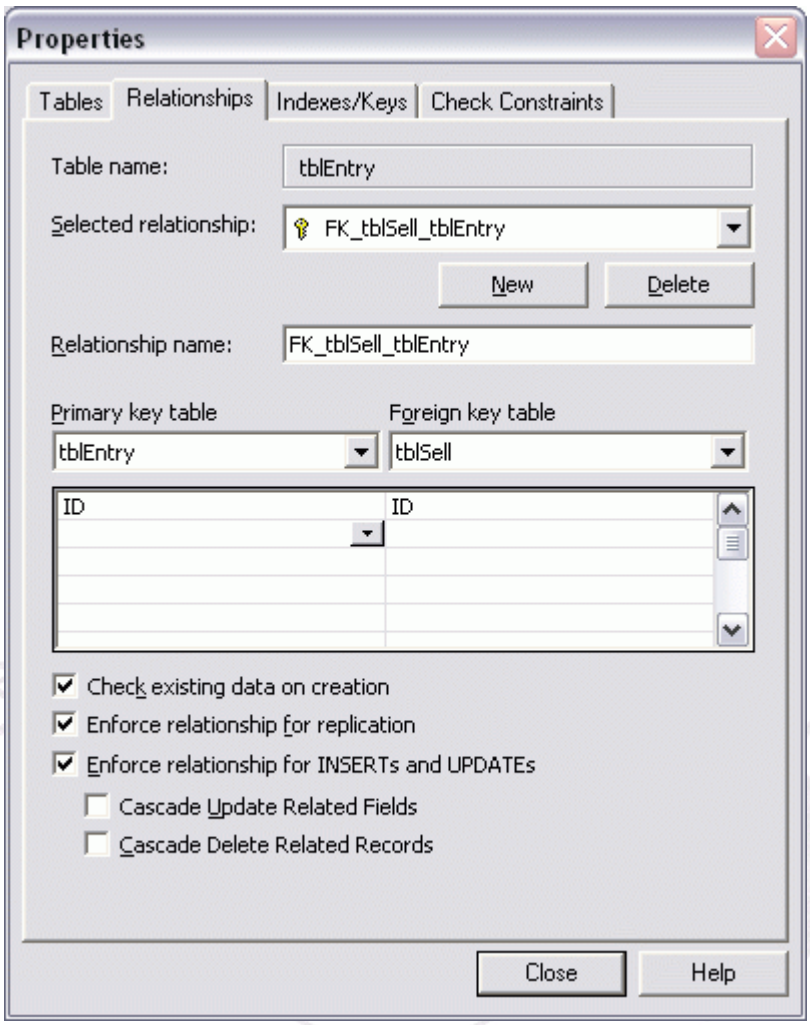
شکل ۲– مشخصات جدول tblSell در حالت طراحی.

در ادامه می خواهیم Foreign key را ایجاد نماییم . روی صفحه در حالت طراحی کلیک راست کنید و سپس گزینه ی Relationship را انتخاب نمایید (شکل ۳) :



شکل ۳ – انتخاب Relationship برای ایجاد قیودات بیشتر .

در صفحه ی ظاهر شده روی New کلیک کنید (شکل ٤) و تنظیمات صفحه را مانند شکل انجام دهید.



شکل ٤ – نحوه ی ایجاد Relationship بین فیلدها و ایجاد Foreign key .

و در آخر پس از بستن این صفحه دیالوگ ، صفحه ی نمایش ثبت تغییرات ظاهر خواهد شد (شکل ٥) .

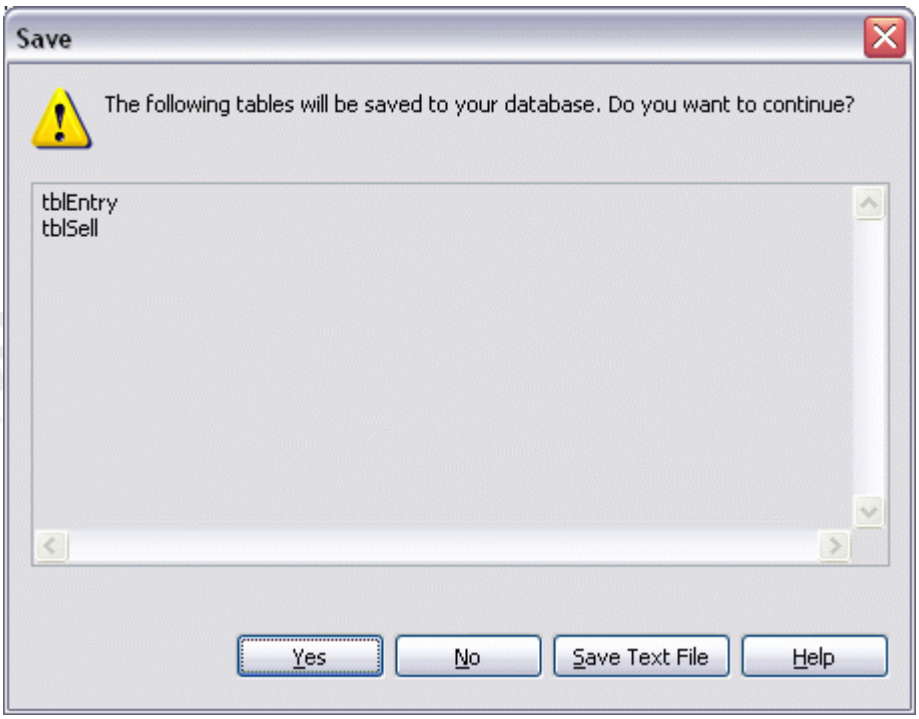
برای ایجاد رویه ذخیره شده ، در Query Analyzer رویه ذخیره شده زیر را ایجاد کنید.

```
Create Procedure rptDiff
As
select distinct myTestDB.dbo.tblEntry.ID ,myTestDB.dbo.tblEntry.Entry_No , myTestDB.dbo.tblSell.Sell_No,
(myTestDB.dbo.tblSell.Sell_No - myTestDB.dbo.tblEntry.Entry_No)  as final_result
from myTestDB.dbo.tblEntry LEFT OUTER JOIN  myTestDB.dbo.tblSell
   ON (myTestDB.dbo.tblEntry.ID = myTestDB.dbo.tblSell.ID)
Return
```

برای اینکه بتوان با دیتابیس فوق کار کرد می توان یک سری داده را خیلی سریع در محیط Enterprise manager وارد نمود.

برای تست کردن آن هم می توانید از دستور زیر استفاده کنید :

Exec rptDiff



شکل ۵ – صفحه ی تایید تغییرات انجام شده بر روی دیتابیس .

سپس نحوه ی استفاده از آن در ADO.NET و برنامه ما با استفاده از SqlDataAdapter به صورت زیر است :

```
private void Page_Load(object sender, System.EventArgs e)
   {
      SqlConnection sqlconnectionForum =  new
      SqlConnection("server=(local);uid=sa;pwd=;database=MyTestDB");

      SqlDataAdapter sqldataadapterSP =
          new SqlDataAdapter("rptDiff",sqlconnectionForum);
      sqldataadapterSP.SelectCommand.CommandType =
          CommandType.StoredProcedure ;

      DataSet datasetSP = new DataSet();
```

```
    sqldataadapterSP.Fill( datasetSP,"tblEntry" );

    DataGrid1.DataSource = datasetSP.Tables["tblEntry"].DefaultView;
    DataGrid1.DataBind();
}
```

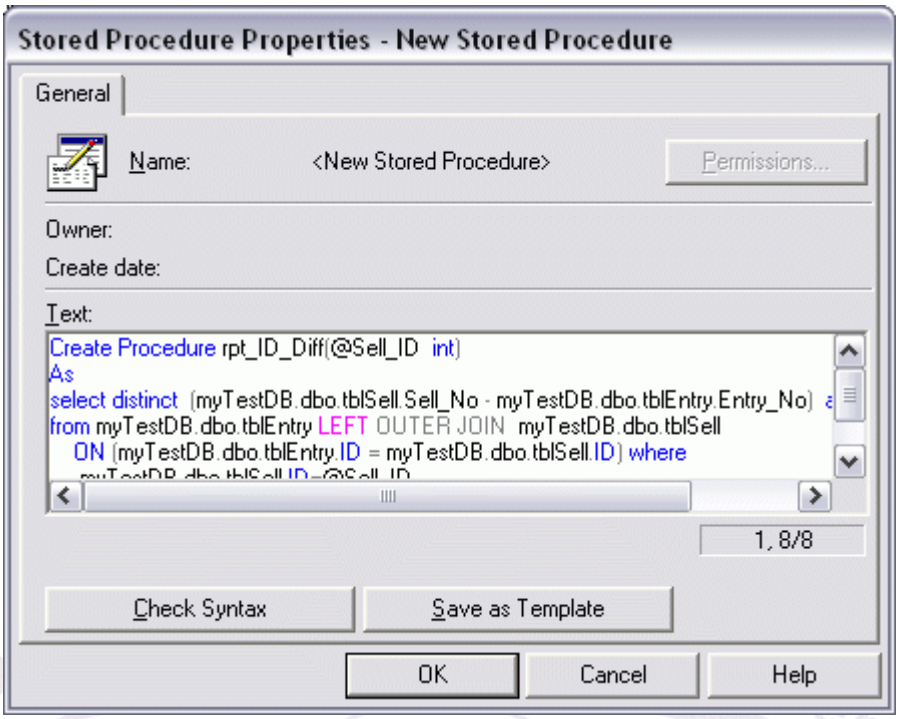
مثال ۲ :

در این مثال می خواهیم نحوه ی پاس کردن متغیرها و ورودی های فرم را بـه یـک رویـه ذخیـره شـده بررسی کنیم. فرض کنیم در دیتابیس MyTestDB که آنرا در مثال قبل ایجاد کردیم می خـواهیم بـا دادن ID تفاضل تعداد ورودی و تعداد فروخته شده آنرا بدست آوریم و روی صفحه نمایش دهیم :

ابتدا رویه ذخیره شده زیر را ایجاد نمایید :

برای اینکار علاوه بر Query Analyzer می توان از محیط Enterprise manager نیز استفاده کرد. در لیست درختی مربوط به MyTestDB روی گزینه ی Stored procedures کلیک نمایید تا تمام رویه های ذخیره شده مربوط به بانک اطلاعاتی خودتان را مشاهده نمایید . سپس روی صفحه آن کلیک راست نمایید و گزینه ی New Stored procedure را انتخاب نماییـد.درصـفحه ی بـاز شـده (شـکل ٦) عبـارت زیـر را نوشته و آنرا ذخیره کنید :

```
Create Procedure rpt_ID_Diff(@Sell_ID  int)
As
select distinct  (myTestDB.dbo.tblSell.Sell_No - myTestDB.dbo.tblEntry.Entry_No)  as final_result
from myTestDB.dbo.tblEntry LEFT OUTER JOIN  myTestDB.dbo.tblSell
   ON (myTestDB.dbo.tblEntry.ID = myTestDB.dbo.tblSell.ID) where
    myTestDB.dbo.tblSell.ID=@Sell_ID
Return
GO
```



شکل ٦ – نحوه ی ایجاد رویه ذخیره شده در محیط Enterprise manager .

برای تست کردن آن هم می توان در Query analyzer عبارت زیر را وارد کرد :

```
exec rpt_ID_diff 2
```

سپس در یک پروژه جدید به صورت زیر از آن استفاده خواهیم کرد :

```
private void Page_Load(object sender, System.EventArgs e)
   {
        SqlConnection   MyConnection = new
            SqlConnection("server=(local);uid=sa;pwd=;database=MyTestDB");

         //Calling the DisplayCustomers stored procedure
         SqlDataAdapter MyCommand = new
                     SqlDataAdapter("rpt_ID_Diff", MyConnection);
          MyCommand.SelectCommand.CommandType = CommandType.StoredProcedure;

          //Adding the SQL parameter
          MyCommand.SelectCommand.Parameters.Add(
                     new SqlParameter("@Sell_ID",SqlDbType.Int ));
          //Specifying the parameter value
          MyCommand.SelectCommand.Parameters["@Sell_ID"].Value = 3;
```

```
        DataSet DS = new DataSet();
        MyCommand.Fill(DS, "tblSell");
        // write & show final_result value
        Response.Write (    "final_result = "+
                        DS.Tables["tblSell"].Rows[0][0].ToString()  ) ;

    }
```

انجام یک سری از عملیات ریاضی روی فیلدها :

گاهی از اوقات لازم است تا برای مثال جمع کل عددهای موجود در یک ستون (فیلد) را محاسبه کنیم و یا میانگین ها و امثال اینگونه عملیات . یکی از راه حل ها آن بدین صورت است که کل اعداد فیلد را بخوانیم و سپس عملیات روی آن انجام دهیم و یا راه دیگر استفاده از دستورات مخصوص SQL برای اینگونه کارها می باشد. مثال زیر نحوه ی انجام اینگونه عملیات را بیان می کند.

مثال ۳ :

می خواهیم بزرگترین عدد موجود در فیلد myTestDB.dbo.tblSell.ID که در مثال اول ایجاد شد را بدست آوریم.

برای نوشتن این برنامه راه حل های زیادی وجود دارد که یکی از آنها در زیر بررسی خواهد شد. برای بدست آوردن یک مقدار از دیتابیس از متد ExecuteScalar مربوط به شیء SqlCommand استفاده می گردد.

```
private void Page_Load(object sender, System.EventArgs e)
{
  SqlConnection MyConnection = new
           SqlConnection("server=(local);uid=sa;pwd=;database=MyTestDB");
```

```
 SqlCommand newCmd = new
      SqlCommand("select MAX(tblSell.ID) from tblSell",MyConnection);

 MyConnection.Open();
   int intRes = (int)newCmd.ExecuteScalar();
 MyConnection.Close();

 Response.Write(intRes) ;
}
```

روشی دیگر برای فرستادن متغیرها به عبارات T-SQL :

در فصل قبل برای مثال برای انتخاب کردن یک سری رکورد از دیتابیس ازدستوراتی مانند زیر استفاده می کردیم :

```
strSQL = " select * from tbl1 where field1='"+ text1.text + "'";
```

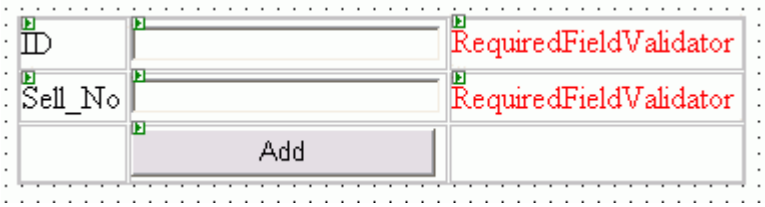
به جای اینکه Text1.text را بدین صورت به عبارت پاس کنیم می توان از پارامترها هم به صورت زیر استفاده نمود :

مثال ۴:

در این مثال می خواهیم با استفاده از روشی متفاوت نسبت به مثال یک ، محتویات تکست باکس ها را البته برای ورود اطلاعات به جدول myTestDB.dbo.tblEntry.ID مورد بررسی قرار دهیم.

فرم ورود اطلاعات را به شکل زیر آماده کرده و نام تکست باکس ها را به عباراتی معنا دار تبدیل نمایید (از جدولی با BorderSize=0 هم می توان برای مرتب کردن عناصر صفحه استفاده کرد) ، سپس

RequiredFieldValidators را به تکست باکس ها مرتبط نمایید و یک دکمه هم برای اضافه کردن اطلاعات به دیتابیس به صفحه اضافه نمایید و یک دیتاگرید برای نمایش آنها.



شکل ۷ – فرم ورود اطلاعات مربوط به مثال ٤ در حال طراحی.

```
private void btnAdd_Click(object sender, System.EventArgs e)
{
      SqlConnection sqlconnectionMyTestDB = new
      SqlConnection("server=(local);uid=sa;pwd=;database=MyTestDB");
      SqlDataAdapter sqldataadapterEntry = new
      SqlDataAdapter("select * from tblEntry", sqlconnectionMyTestDB);

      String insertCmd = "INSERT INTO tblEntry(id, Entry_no) VALUES( " +
                         "@Id, @Entry_No )";

      SqlCommand sqlcommandEntry = new SqlCommand(insertCmd,
                                              sqlconnectionMyTestDB);

      sqlcommandEntry.Parameters.Add(
                           new SqlParameter("@Id", SqlDbType.Int));
      sqlcommandEntry.Parameters["@Id"].Value = txtID.Text;

      sqlcommandEntry.Parameters.Add(
                         new SqlParameter("@Entry_No", SqlDbType.Int));
      sqlcommandEntry.Parameters["@Entry_No"].Value = txt_Entry_No.Text;


      sqlcommandEntry.Connection.Open();
      sqlcommandEntry.ExecuteNonQuery();
      sqlcommandEntry.Connection.Close();

      DataSet datasetEntry = new DataSet();
      sqldataadapterEntry.Fill(datasetEntry, "tblEntry");


      DataGrid1.DataSource=datasetEntry.Tables["tblEntry"].DefaultView;
      DataGrid1.DataBind();
      DataGrid1.Visible = true;

}
```

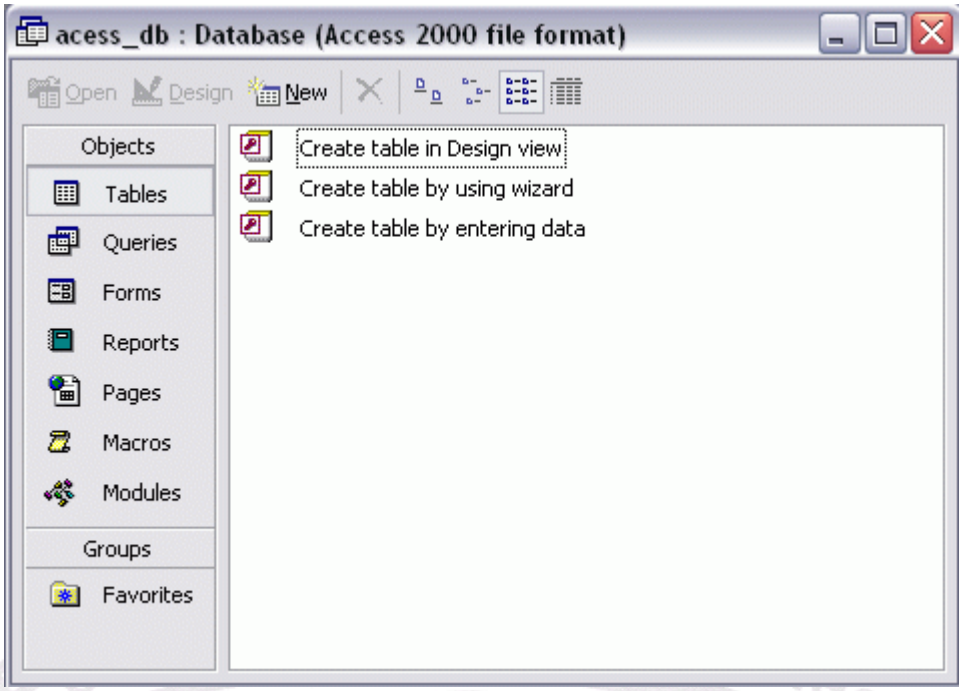
همانطور که ملاحظه می فرمایید این روش شبیه به روشی است که برای رویه های ذخیره شده ای که پارامتر می گرفتند بکار می رود.

استفاده از دیتابیس های OLE DB :

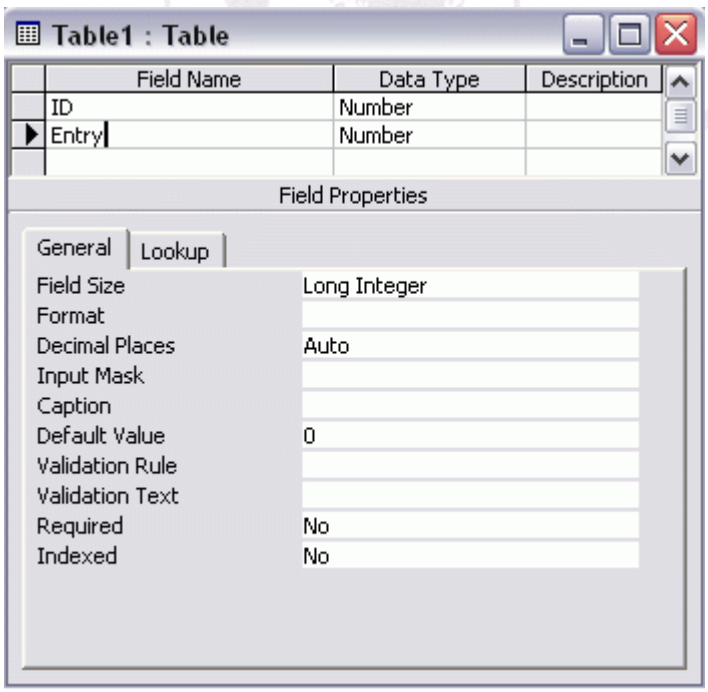
برای استفاده از دیتابیس های OLE DB مانند اکسس در ADO.NET باید از فضای نام System.Data.OleDb استفاده کرد. با استفاده از کلاس System.Data.OleDb.OleDbCommand می توان یک سری از دستورات SQL مانند Select ، Insert ، Update و Delete و همچنین اجرای رویه های ذخیره شده را انجام داد.

مثال ۵:

در مثال زیر قصد داریم یک سری از اطلاعات را درون بانک اطلاعاتی اکسس ذخیره کنیم.
ابتدا اکسس XP را باز کنید و سپس از منوی فایل گزینه ی New را انتخاب کنید و از پنل سمت راست صفحه روی گزینه ی Blank DataBase کلیک نمایید تا یک بانک جدید خالی برای مثال به نام access_db.mdb ایجاد شود. در صفحه ی دیالوگ باز شده (شکل ۸) روی آیتم Create Table in design View کلیک نمایید تا صفحه ی طراحی دیتابیس که شبیه محیط طراحی دیتابیس در SQL-Server است باز شود . سپس مطابق شکل زیر (شکل ۹) جدول را طراحی کنید. سپس پنجره را ببندید تا صفحه ی ذخیره کردن نام جدول (شکل ۱۰) ظاهر شود و نام tblEntry را وارد نمایید. سپس اکسس از شما در مورد ایجاد Primary key سوال می کند (شکل ۱۱).

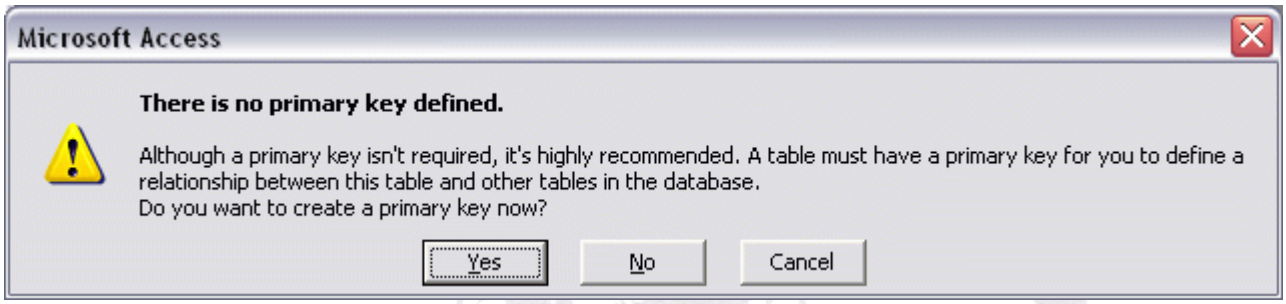


شکل ۸ – صفحه ی خواص دیتابیس خالی ایجاد شده در اکسس.



شکل ۹ – طراحی جدول جدید در اکسس.

شکل ۱۰- وارد کردن نام جدول جدید بانک اکسس.

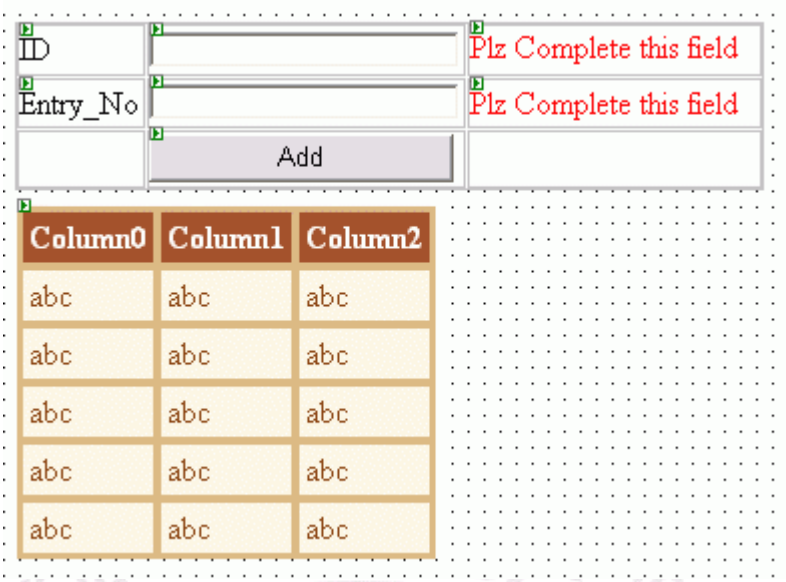


شکل ۱۱– صفحه ی تایید ایجاد فیلد Primary key .

آنرا تایید نکنید! چون یک فیلد پیش فرض درست می کند (البته این موضوع بستگی به دیتابیس شما هم دارد...). برای درست کردن Primary key همانند SQL-Server می توان عمل کرد (یعنی روی فیلد دلخواه می توان کلیک کرد و سپس از منوی ظاهر شده گزینه ی Primary key را انتخاب نمود.) یک Primary key روی فیلد ID درست کنید. کار خود را ذخیره کنید تا از این دیتابیس ایجاد شده در مثالهای بعدی استفاده نماییم .

یک پروژه ی ASP.NET دیگر باز کنید و سپس شکل ظاهری آنرا همانند مثال قبلی طراحی نمایید (شکل ۱۲) .

سپس با استفاده از کد زیر می توان رکوردهای جدید را به آن اضافه کرد.



شکل ۱۲ – شکل ظاهری مثال ۵ در حالت طراحی .

قبل از هر کاری فضای نام مربوطه را به برنامه ملحق می کنیم :

```
using System.Data.OleDb;
```

و در رخداد کلیک مربوط به دکمه Add خواهیم نوشت :

```
private void btnAdd_Click(object sender, System.EventArgs e)
{
      //g:\inetpub\wwwroot\classes\ch09\ex05\acess_db.mdb
      // or :
      //MapPath : Returns the physical file path that corresponds to
      //the specified virtual path on the Web server.
      String FilePath;
      FilePath = Server.MapPath("acess_db.mdb");

      // Connect to Database
      OleDbConnection cnAccess = new
            OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;"+
                        "Data Source="+ FilePath );
      cnAccess.Open();


      //Make the insert statement
      string sInsertSQL = "insert into tblEntry values(" +
                        txtID.Text + "," + txt_Entry_No.Text  + ")";
```

```
        //Make the OleDbCommand object
        OleDbCommand cmdInsert = new OleDbCommand(sInsertSQL,cnAccess);

        // This not a query so we do not expect any return data so use
        // the ExecuteNonQuery method
        cmdInsert.ExecuteNonQuery();

        // displaying data
        OleDbDataAdapter sqldataadapterEntry =
               new OleDbDataAdapter("select * from tblEntry",cnAccess);
        DataSet datasetEntry = new DataSet();
                    sqldataadapterEntry.Fill(datasetEntry, "tblEntry");


        DataGrid1.DataSource=datasetEntry.Tables["tblEntry"].DefaultView;
        DataGrid1.DataBind();
        DataGrid1.Visible = true;

}
```

در مثال فوق از همان روش قدیمی ASP برای مشخص کردن مسیر فیزیکی فایل mdb اکسس به صورت Server.MapPath استفاده کرده ایم . همانطور که ملاحظه می فرمایید همه چیز مانند قبل است فقط بجای Sql عبارت OleDb قرار گرفته است و تمام توضیحات آنها هم تکراری می باشد .

با استفاده از کلاس System.Data.OleDb.OleDbDataReader می توان مانند یک Recordset فقط خواندنی سیستم قبلی ADO استفاده کرد. در هر لحظه فقط یک رکورد را خواند.

مثال ۶:

در زیر مثالی را از نحوه ی استفاده از کلاس OleDbDataReader با هم مرور می کنیم .

یک Label روی فرم قرار دهید و سپس فضاهای نام زیر را به برنامه اضافه نمایید :

```
using System.Data.OleDb ;
using System.Text; // for StringBuilder
```

می خواهیم تک تک رکوردهای جدول tblEntry مربوط به بانک اطلاعاتی اکسس را که در طی مثال قبل ایجاد کرده ایم ، در برنامه خوانده و آنرا در یک جدول که خودمان با استفاده از تگهای HTML ایجاد می کنیم ، نمایش دهیم.

از فضای نام System.Text به این جهت در برنامه استفاده کرده ایم که می خواهیم از کلاس StringBuilder استفاده نماییم. روشی شیک (!) و کارآتر نسبت به علامت + در مورد جمع کردن string ها استفاده از این کلاس می باشد که در مثال زیر در عمل بکار گرفته شده است.

```
private void Page_Load(object sender, System.EventArgs e)
{
	String FilePath;
	FilePath = Server.MapPath("acess_db.mdb");

	// Connect to Database
	OleDbConnection cnAccess = new
		OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;"+
		"Data Source="+ FilePath );

	cnAccess.Open();

	//Make the select statement
	string sSelectSQL = "select * from tblEntry";
	//'Make the OleDbCommand object
	OleDbCommand cmdSelect = new OleDbCommand(sSelectSQL,cnAccess);

	//This query should return an OleDbDataReader so we use the
	//ExecuteReader method

	StringBuilder sbResults = new  StringBuilder();
	OleDbDataReader drEmp = cmdSelect.ExecuteReader();


	drEmp.Read();

	sbResults.Append ("<Table>");
	do
	{
		sbResults.Append ("<TR><TD>");
		sbResults.Append ( drEmp.GetInt32(0).ToString());
		sbResults.Append ("</TD><TD>");
		sbResults.Append ( drEmp.GetInt32(1).ToString() );
		sbResults.Append ("</TD><TR>");

	}while (drEmp.Read());
	sbResults.Append ("</Table>");

	lblResult.Text = sbResults.ToString();
}
```

کلاس OleDbDataAdapter :

همانطور که تابحال ملاحظه کرده اید Data Adapter بیانگر دستورات و اتصالاتی است که برای پیمایش دیتابیس بکار گرفته می شود . این کلاس سه خاصیت دستوری دارد که برای به روز رسانی دیتابیس مورد استفاده قرار می گیرد:

InsertCommand : بیانگر پرسجو یا رویه ذخیره شده ای است که برای اضافه کردن رکورد جدید به دیتابیس بکار برده می شود.

SelectCommad : بیانگر یک عبارت SQL است که برای انتخاب رکوردها از بانک اطلاعاتی بکار می رود.

DeleteCommand : بیانگر یک عبارت SQL است که برای حذف رکوردها از دیتاست بکار می رود.

کلاس های System.Data.DataSet و System.Data.DataTable و System.Data.DataRow و System.Data.DataColumn :

DataSet کلاسی است عمومی که بوسیله ی Net Framework. تهیه شده است. این کلاس بر روی سمت کلاینت برای ذخیره سازی داده ها به روشی که بسیار کاربردی تر و قوی تر است نسبت به ADO Recordset کاربرد دارد. علاوه بر این داده ها در DataSet به فرمت XML موجود بوده و بنابراین برای دستیابی و مدیریت آماده می باشند. فرمت XML آنرا برای کاربردهای وب بسیار مناسب ساخته و دستیابی Cross-Platform را ممکن می سازد. DataSet قابلیت ذخیره سازی از چندین جدول و حفظ ارتباطات بین آنها است. جداول در اشیاء DataTable ذخیره می شوند و DataRelation بیانگر اتباطات بین جداول است. در شیء های DataRow و DataColumn به ترتیب ، ردیف ها و ستون ها در یک جدول ذخیره می شوند (شکل ۱۳).

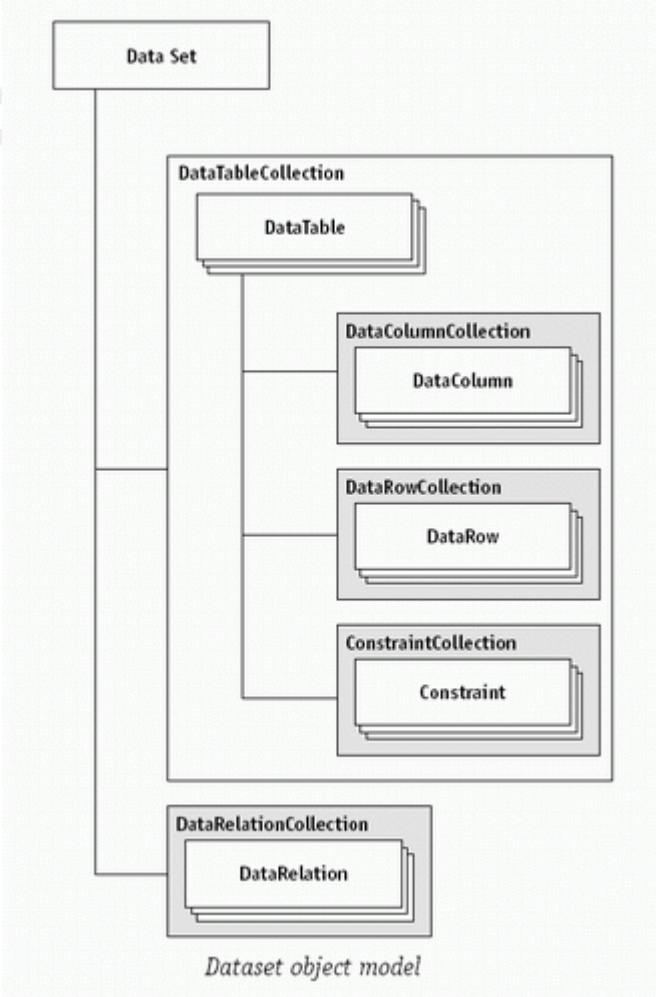
مثال ۷:

مثالی در مورد نحوه ی استفاده از اشیاء DataTable و روابط بین آنها .

در این مثال می خواهیم مثال قبل را با استفاده از اشیاء ذکر شده در قمست جاری باز نویسی کنیم .

یک Label روی فرم قرار دهید و سپس فضاهای نام زیر را به برنامه اضافه نمایید :

```
using System.Data.OleDb ;
using System.Text; // for StringBuilder
```



شکل ۱۳- مدل شیءایی DataSet .

```
private void Page_Load(object sender, System.EventArgs e)
{

      String FilePath;
      FilePath = Server.MapPath("acess_db.mdb");

      // Connect to Database
      OleDbConnection cnAccess = new
                        OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;"+
                        "Data Source="+ FilePath );

      cnAccess.Open();

      // Make the select statement
      string sSelectSQL = "select * from tblEntry";
      //Make the OleDbCommand object
      OleDbCommand cmdSelect = new OleDbCommand(sSelectSQL,cnAccess);
      OleDbDataAdapter daEmp = new OleDbDataAdapter(cmdSelect);

      DataSet dsEmp = new DataSet();

      StringBuilder sbResults = new StringBuilder();

      // Fill the data with the output of the cmdSelect command. Note
      // that the dataadapter is associated with the command. We use
      // the dataadapter to fill the dataset.
      daEmp.Fill(dsEmp, "tblEntry");
      PrintRows(dsEmp);
}
private void PrintRows(DataSet myDataSet)
{
      StringBuilder sbResult =new  StringBuilder();

      // Iterate through all the DataTables in the DataSet
      foreach( DataTable dtEmp in myDataSet.Tables )
         {
               sbResult.Append("<Table>");
                 // Iterate through all the DataRows in the DataTable
               foreach( DataRow drEmp in dtEmp.Rows )
               {
                 sbResult.Append("<TR>");
                 // Iterate through all the DataColumns in the DataRow
                       foreach (DataColumn dcEmp in dtEmp.Columns)
                       {
                             sbResult.Append("<TD>");
                             sbResult.Append(drEmp[dcEmp]);
                             sbResult.Append("</TD>");
                       }
                   sbResult.Append("</TR>");
                }
                sbResult.Append("</Table>");
            }

        lblResult.Text = sbResult.ToString();
}
```

تمرین :

۱- مثال ٤ را طوری تغییر دهید که بجای txtID یک Label قرار گیرد و با هر بار اضافه شدن یک آیتم به دیتابیس مقدار آن به صورت خودکار یکی اضافه شود .
۲- New Post مربوط به فوروم را با رویه های ذخیره شده پیاده سازی کنید.
۳- در جدول tblEntry دیتابیس MyTestDB که در این فصل ایجاد شد میانگین داده های موجود در فیلد Entry_No را روی صفحه نمایش دهید.
٤- جدول tblArea فوروم را با فیلدهایی مانند نام و آی دی بخش و شرح بخش جدید ایجاد نموده و سپس فرم ورود داده های آنرا خلق نمایید.
٥- جدول tblPosts فوروم را با فیلدهایی مانند نام شخص پست کننده ، عنوان پست ، محتوای پست ، تاریخ پست ، آی دی بخشی که در آن پست انجام می شود ایجاد نموده و سپس فرم ورود داده های آنرا خلق نمایید.