**ADO.NET(BASICS)**

**SqlDataReader**

1.Connection String string cs =ConfigurationManager.ConnectionStrings[“DBCS”].ConnectionString ( is in Web Config file)

2. SqlConnection con = new SqlConnection(cs);

3.SqlCommand cmd = new SqlCommand(“Sql statement”,con);

4.Open the connection con.Open();

5.Execute the command cmd.ExecuteReader(), cmd.ExecuteScalar() , cmd.ExecuteNoNQuery

6.Close the connection con.Close() (use the finally block to close the connection)

**Store Procedure with out put parameters**

SqlCommand cmd = new SqlCommand(“spAddEmployee”,con);

Cmd.CommandType = System.Data.CommandType.StoredProcedure;

Cmd.Parameter.AddWithValue(“@Name”,txtName.Text); ……….

SqlParameter output = new SqlParameter();

outputParam.ParameterName =”@Employee”;

outputParam.SqlDbType = System.Data.SqlDbType.Int;

outputParam.Direction = System.Data.ParameterDirection.Output;

cmd.Parameters.Add(outputParam);

con.Open();

cmd.ExecuteNonQuery();

string EmpId = outputParam.Value.ToString();

Label1.Text = “Employee ID = “ + EmpId;

**SqlDataAdapter**

1.SqlConnection

2.SqlDataAdapter da = new SqlDataAdapter(“Sql Statements or StoredProcedures”,con);

3.DataSet ds = new DataSet();

4.da.Fill(ds); ( this fills the data Set with the data from the database , we don’t have to open the con)

5.We can use dataset to load 2 or more tables into the dataset

|  |  |
| --- | --- |
| GridView1.DataSource=ds.Tables[0]; GridView1.DataBind(); | GridView1.DataSource=ds.Tables[1]; GridView1.DataBind(); |

**Caching DataSet**

1.SqlConnection---2.SqlDataAdapter ---3.DataSet ds..---4.da.Fill(ds);---5.Cache[“Data”]=ds;(the data is stored in the Cache using the name”Data”, we can now use the Cache to load data from it)

**SqlCommandBuilder**

1.SqlDataAdapter ---2.da.SelectCommand = new SqlCommand(“select \_Query”,con);

3.SqlCommandBuilder bc =new SqlCommandBuilder(da);

**DataSet RejectChanges() and AcceptChanges()—(**everty row that is present DataTable of DataSet has RowState property

**StronglyTyped DataSets**

Add New Item🡪 DataSet File.xsd;

View-> Server Explorer(Displays the tables in the database we use to create strongly typed DataSet ,we only need to drag and drop the table on to the designer(in the dataset we made)

**Load xml data into sql server table using SqlBulkCopy**

Ex:

1.Create a empty project.2Create a xml file (with the data we want to add to the Sql Tables)

3.Create 2 tables in Sql ; 4.Create a web form -> pass in the connection string to the database

5.Insert a button and implement it’s functionality.

Using(SqlConnection con = new SqlConnections(cs)){DataSet ds = new DataSet();

ds.ReadXml(Server.MapPath(“~/Data.xml); DataTable dtDep =ds.Tables[“Department”];

DataTable dtEmp =ds.Tables[“Employee”];con.Open();

Using(SqlBulkCopy bc = new SqlBulkCopy(con)){bc.DestinationTableName =”Departments”;

bc.ColumnMappings.Add(“ID”,”ID”);....(the same for the columns we want)..bc.writeToServer(dtDep);}

Using(SqlBulkCopy bc = new SqlBulkCopy(con)){bc.DestinationTableName =”Employees”;

bc.ColumnMappings.Add(“ID”,”ID”);....(the same for the columns we want)bc.writeToServer(dtEmp);}

(Column mappings are not required if the column names in the source and destination tables are the same)

**SqlBulkCopy NotifyAfter**(SqlRowsCopied event )BatchSize property specifies the number of rows in a batch that will be copied to the destination table.The default batchsize is 1(we can set it to any number and once the reader has read that number the rows will be sent to the database as a single batch to perform bulkCopy operation.

**Transaction in Ado.Net**

To open the transaction :

SqlTransaction tran = con.BeginTransaction();

Try{ --Command --Execute command tran.Commit();}catch{tran.Rollback();}

**DataSet AcceptChanges() and RejectChanges()**

To understand accept and reject changes methods better we need to understand RowStates and RowVersions.

|  |  |
| --- | --- |
| RowState | Unchanged;Added;Modified;Deleted;Detached |
| DataRowVersion | Current(doesn’t exist for rows with a RowState of deleted);  Default(for Added,Modified,Unchanged row is Current,for Deleted is Original and for Detached is proposed) |