

```
using System;
using System.Data;
using Caissa.Classes;

namespace Caissa.Persistence
{
    public class SourceDB
    {
        public int source_id
        {
            get;
            set;
        }

        #region Insert Source
        public void Insert(Source source)
        {
            string storedprocedure = "sp_InsertSource";

            System.Data.IDbConnection connection;
            System.Data.IDbCommand command;

            connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

            command = Mapped.Command(storedprocedure, connection);
            command.CommandType = CommandType.StoredProcedure;
            command.Parameters.Add(Mapped.Parameter("p_sou_title", source.Title,
                DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sou_description",
                source.Description, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sou_author", source.Author,
                DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sou_external_link",
                source.ExternalLink, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sot_id", source.SourceType,
                DbType.Int32, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sou_id", "", DbType.Int32,
                ParameterDirection.Output));
            command.ExecuteNonQuery();

            int id = 0;

            foreach (var parameter in command.Parameters)
            {
                IDbDataParameter p = (IDbDataParameter)parameter;
                if (p.Direction == ParameterDirection.Output)
                    if (p.ParameterName == "p_fon_codigo")
                        id = Convert.ToInt32(p.Value);
            }

            connection.Close();

            source_id = id;
        }
        #endregion
    }
}
```

```
#region Update Source
public void Update(Source source)
{
    string storedprocedure = "sp_UpdateSource";

    System.Data.IDbConnection connection;
    System.Data.IDbCommand command;

    connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

    command = Mapped.Command(storedprocedure, connection);
    command.CommandType = CommandType.StoredProcedure;
    command.Parameters.Add(Mapped.Parameter("p_sou_id", source.Id,
        DbType.Int32, ParameterDirection.Input));
    command.Parameters.Add(Mapped.Parameter("p_sou_title", source.Title,
        DbType.String, ParameterDirection.Input));
    command.Parameters.Add(Mapped.Parameter("p_sou_description",
        source.Description, DbType.String, ParameterDirection.Input));
    command.Parameters.Add(Mapped.Parameter("p_sou_author", source.Author,
        DbType.String, ParameterDirection.Input));
    command.Parameters.Add(Mapped.Parameter("p_sou_external_link",
        source.ExternalLink, DbType.String, ParameterDirection.Input));
    command.Parameters.Add(Mapped.Parameter("p_sot_id", source.SourceType,
        DbType.Int32, ParameterDirection.Input));
    command.ExecuteNonQuery();

    connection.Close();
}
#endregion

#region Delete Source
public void Delete(int source_id)
{
    string storedprocedure = "sp_DeleteSource";

    System.Data.IDbConnection connection;
    System.Data.IDbCommand command;

    connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

    command = Mapped.Command(storedprocedure, connection);
    command.CommandType = CommandType.StoredProcedure;
    command.Parameters.Add(Mapped.Parameter("p_sou_id", source_id,
        DbType.Int32, ParameterDirection.Input));
    command.ExecuteNonQuery();

    connection.Close();
}
#endregion

#region Select Source
public DataSet Select()
{
    System.Data.IDbConnection connection;
    System.Data.IDbCommand command;
    System.Data.IDataAdapter adapter;
```

```

        DataSet ds = new DataSet();

        string sql = string.Empty;

        connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

        sql = "select sou.sou_id, sou.sou_title, sou.sou_description,"
            + " sou.sou_author, sou.sou_external_link, sot.sot_title,"
            + " FORMAT( "
                + " (select"
                + " sum(asi.asi_cov_loc)/count(asi.asi_id)"
                + " from tbl_assessment_item asi"
                + " where asi.sou_id = sou.sou_id)"
            + " ,3) as sou_cov_glo"
            + " from tbl_source as sou"
            + " inner join tbl_source_type as sot on (sou.sot_id = sot.sot_id)"
            + " where sou_active = 0;";

        command = Mapped.Command(sql, connection);

        adapter = Mapped.Adapter(command);
        adapter.Fill(ds);

        connection.Close();

        return ds;
    }
#endregion

#region Select Especific Source
public DataSet SelectSource(int source_id)
{
    System.Data.IDbConnection connection;
    System.Data.IDbCommand command;
    System.Data.IDataAdapter adapter;

    DataSet ds = new DataSet();

    string sql = string.Empty;

    connection = Mapped.Connection(Mapped.conexao.ServerExplorer);

    sql = "select sou.sou_id, sou.sou_title, sou.sou_description,"
        + " sou.sou_author, sou.sou_external_link, sot.sot_id,"
        + " sot.sot_title"
        + " from tbl_source as sou"
        + " inner join tbl_source_type as sot on (sou.sot_id = sot.sot_id)"
        + " where sou.sou_id = ?source_id"
        + " and sou.sou_active = 0;";

    command = Mapped.Command(sql, connection);
    command.Parameters.Add(Mapped.Parameter("?source_id", source_id));

```

```
        adapter = Mapped.Adapter(command);
        adapter.Fill(ds);

        connection.Close();

        return ds;
    }
#endregion

public SourceDB()
{
    //
    // TODO: Add constructor logic here
    //
}
}
```