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

namespace Caissa.Persistence
{
    public class AssessmentItemDB
    {
        public int assessment_item_id
        {
            get;
            set;
        }

        public int assessment_item_test_id
        {
            get;
            set;
        }

        public int assessment_item_tool_id
        {
            get;
            set;
        }

        #region Insert Assessment Item
        public void Insert(AssessmentItem item)
        {
            string storedprocedure = "sp_InsertAssessmentItem";

            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_asi_initials",
              item.Initials, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_title", item.Title,
              DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_description",
              item.Description, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_expected_result",
              item.ExpectedResult, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_procedure",
              item.Procedure, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_div_dm", item.DivDM,
              DbType.Double, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_div_pp", item.DivPP,
              DbType.Double, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_cov_loc", item.CovLoc,
              DbType.Double, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sou_id", item.Source,
              DbType.Int32, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_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_asi_id")
                        id = Convert.ToInt32(p.Value);
            }

            connection.Close();

            assessment_item_id = 0;
        }
        #endregion

        #region Update Assessment Item
        public void Update(AssessmentItem item)
        {
            string storedprocedure = "sp_UpdateAssessmentItem";

            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_asi_id", item.Id, 
              DbType.Int32, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_initials", 
              item.Initials, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_title", item.Title, 
              DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_description", 
              item.Description, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_expected_result", 
              item.ExpectedResult, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_procedure", 
              item.Procedure, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_div_dm", item.DivDM, 
              DbType.Double, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_div_pp", item.DivPP, 
              DbType.Double, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_asi_cov_loc", item.CovLoc, 
              DbType.Double, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sou_id", item.Source, 
              DbType.Int32, ParameterDirection.Input));
            command.ExecuteNonQuery();

            connection.Close();
        }
        #endregion

        #region Delete Assessment Item
        public void Delete(int item_id)
        {
            string storedprocedure = "sp_DeleteAssessmentItem";

            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_asi_id", item_id,
              DbType.Int32, ParameterDirection.Input));
            command.ExecuteNonQuery();

            connection.Close();
        }
        #endregion

        #region Select Assessment Item
        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 asi.asi_id, asi.asi_initials, asi.asi_title,
              asi.asi_description, asi.asi_expected_result,"
                + " asi.asi_procedure, FORMAT(asi.asi_div_dm,3) as asi_div_dm,
                  FORMAT(asi.asi_div_pp,3) as asi_div_pp,"
                + " FORMAT(asi.asi_cov_loc,3) as asi_cov_loc, sou.sou_id,
                  sou.sou_title"
                + " from tbl_assessment_item as asi"
                + " inner join tbl_source as sou on (asi.sou_id = sou.sou_id)"
                + " where asi.asi_active = 0"
                + " and asi.asi_id not in ("
                + "     select ait.asi_id from tbl_assessment_item_test ait"
                + "     where ait.ait_active = 0"
                + " );";

            command = Mapped.Command(sql, connection);

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

            connection.Close();
            return ds;
        }
        #endregion

        #region Bind To Tool
        private void BindToTool(int item_id, int tool_id)
        {
            string storedprocedure = "sp_BindAssessmentItemToTool";

            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_asi_id", item_id,
              DbType.Int32, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_too_id", tool_id,
              DbType.Int32, ParameterDirection.Input));
            command.ExecuteNonQuery();

            int id = 0;

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

            connection.Close();

            assessment_item_tool_id = id;
        }
        #endregion

        #region Bind To Test
        public void BindToTest(int item_id, int test_id)
        {
            string storedprocedure = "sp_BindAssessmentItemToTest";

            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_asi_id", item_id,
              DbType.Int32, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_tes_id", test_id,
              DbType.Int32, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_ait_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_ait_id")
                        id = Convert.ToInt32(p.Value);
            }

            connection.Close();

            assessment_item_test_id = id;
        }
        #endregion

        #region Unbind From Test
        public void UnbindFromTest(int assessment_item_test_id)
        {
            string storedprocedure = "sp_UnbindAssessmentItemFromTest";

            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_ait_id",                     ↩
              assessment_item_test_id, DbType.Int32, ParameterDirection.Input));
            command.ExecuteNonQuery();

            connection.Close();
        }
        #endregion

        #region Unbind From Tool
        public void UnbindFromTool(int assessment_item_tool_id)
        {
            string storedprocedure = "sp_UnbindAssessmentItemFromTool";

            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_ito_id",                     ↩
              assessment_item_tool_id, DbType.Int32, ParameterDirection.Input));
            command.ExecuteNonQuery();

            connection.Close();
        }
        #endregion

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