

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

namespace Caissa.Persistence
{
    public class ToolDB
    {
        public int tool_id
        {
            get;
            set;
        }

        #region Insert Tool
        private void Insert(Tool tool)
        {
            string storedprocedure = "sp_InsertTool";

            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_too_title", tool.Title,
                DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_too_description",
                tool.Description, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_too_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_too_id")
                        id = Convert.ToInt32(p.Value);
            }

            connection.Close();

            tool_id = id;
        }
        #endregion

        #region Update Tool
        private void Update(Tool tool)
        {
            string storedprocedure = "sp_UpdateTool";

            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_too_id", tool.Id,           ↗
            DbType.Int32, ParameterDirection.Input));
        command.Parameters.Add(Mapped.Parameter("p_too_title", tool.Title,     ↗
            DbType.String, ParameterDirection.Input));
        command.Parameters.Add(Mapped.Parameter("p_too_description",          ↗
            tool.Description, DbType.String, ParameterDirection.Input));
        command.ExecuteNonQuery();

        connection.Close();
    }
    #endregion

    #region Delete Tool
    private void Delete(int tool_id)
    {
        string storedprocedure = "sp_DeleteTool";

        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_too_id", tool_id,          ↗
            DbType.Int32, ParameterDirection.Input));
        command.ExecuteNonQuery();

        connection.Close();
    }
    #endregion

    #region Select Tool
    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 too_id, too_title, too_description"
            + " from tbl_tool"
            + " where too_active = 0;";

        command = Mapped.Command(sql, connection);
```

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

        connection.Close();

        return ds;
    }
    #endregion

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