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

namespace Caissa.Persistence
{
    public class SecurityPropertyDB
    {
        public int security_property_id
        {
            get;
            set;
        }

        #region Insert Security Property
        private void Cadastrar(SecurityProperty property)
        {
            string storedprocedure = "sp_InsertSecurityProperty";

            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_sep_title", property.Title,
              DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sep_description",
              property.Description, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sep_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_sep_id")
                        id = Convert.ToInt32(p.Value);
            }

            connection.Close();

            security_property_id = id;
        }
        #endregion

        #region Update Security Property
        private void Update(SecurityProperty property)
        {
            string storedprocedure = "sp_UpdateSecurityProperty";

            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_sep_id", property.Id,
              DbType.Int32, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sep_title", property.Title,
              DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sep_description",
              property.Description, DbType.String, ParameterDirection.Input));
            command.ExecuteNonQuery();

            connection.Close();
        }
        #endregion

        #region Delete Security Property
        private void Delete(int property_id)
        {
            string storedprocedure = "sp_DeleteSecurityProperty";

            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_sep_id", property_id,
              DbType.Int32, ParameterDirection.Input));
            command.ExecuteNonQuery();

            connection.Close();
        }
        #endregion

        #region Select Security Property
        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 sep_id, sep_title, sep_description"
                    + " from tbl_security_property"
                    + " where sep_active = 0"
                    + " order by sep_title;";
```

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

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

            connection.Close();

            return ds;
        }
        #endregion

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