

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

namespace Caissa.Persistence
{
    public class TestDB
    {
        public int test_id
        {
            get;
            set;
        }

        #region Insert Test
        public void Insert(Test test)
        {
            string storedprocedure = "sp_InsertTest";

            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_tes_title", test.Title,
                DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_tes_view_title",
                test.ViewTitle, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_tes_description",
                test.Description, DbType.String, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_tes_start_date",
                test.StartDate, DbType.Date, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_tes_end_date", test.EndDate,
                DbType.Date, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_sys_id", test.System,
                DbType.Int32, ParameterDirection.Input));
            command.Parameters.Add(Mapped.Parameter("p_tes_id", "", DbType.Int32,
                ParameterDirection.Output));
            command.ExecuteNonQuery();

            int id = 0;

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

            connection.Close();

            test_id = id;
        }
    }
}
```

```
}
#endregion

#region Update Test
public void Update(Test test)
{
    string storedprocedure = "sp_UpdateTest";

    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_tes_id", test.Id,
        DbType.Int32, ParameterDirection.Input));
    command.Parameters.Add(Mapped.Parameter("p_tes_title", test.Title,
        DbType.String, ParameterDirection.Input));
    command.Parameters.Add(Mapped.Parameter("p_tes_view_title",
        test.ViewTitle, DbType.String, ParameterDirection.Input));
    command.Parameters.Add(Mapped.Parameter("p_tes_description",
        test.Description, DbType.String, ParameterDirection.Input));
    command.Parameters.Add(Mapped.Parameter("p_tes_start_date",
        test.StartDate, DbType.Date, ParameterDirection.Input));
    command.Parameters.Add(Mapped.Parameter("p_tes_end_date", test.EndDate,
        DbType.Date, ParameterDirection.Input));
    command.ExecuteNonQuery();

    connection.Close();
}
#endregion

#region Delete Test
public void Delete(int test_id)
{
    string storedprocedure = "sp_DeleteTest";

    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_tes_id", test_id,
        DbType.Int32, ParameterDirection.Input));
    command.ExecuteNonQuery();

    connection.Close();
}
#endregion

#region Select Test
public DataSet Select(int test_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 tes_id, tes_title, tes_view_title, tes_description,
tes_start_date, tes_end_date, sys_id,"
+ " FORMAT((select
+ " sum(asi.asi_cov_loc) / count(asi.asi_id)"
+ " from tbl_assessment_item_test ait"
+ " inner join tbl_assessment_item asi on (ait.asi_id =
asi.asi_id)"
+ " where ait.tes_id = tes_id"
+ " and ait.ait_active = 0)"
+ " ,3) as tes_cov_tot"
+ " from tbl_test"
+ " where tes_id = ?test_id"
+ " and tes_active = 0;";

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

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

connection.Close();

return ds;
}
#endregion

#region Select Assessment Items
public DataSet SelectAssessmentItems(int test_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 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,"
+ " asi.sou_id, sou.sou_title, ait.ait_id"
+ " from tbl_assessment_item_test as ait"

```

```
+ " inner join tbl_assessment_item as asi on (ait.asi_id =  
    asi.asi_id)"  
+ " inner join tbl_source as sou on (asi.sou_id = sou.sou_id)"  
+ " where ait.tes_id = ?test_id"  
+ " and asi.asi_active = 0"  
+ " and ait.ait_active = 0;";  
  
command = Mapped.Command(sql, connection);  
command.Parameters.Add(Mapped.Parameter("?test_id", test_id));  
  
adapter = Mapped.Adapter(command);  
adapter.Fill(ds);  
  
connection.Close();  
  
    return ds;  
}  
#endregion  
  
public TestDB()  
{  
    //  
    // TODO: Add constructor logic here  
    //  
}  
}
```