

This document was created by:

José Sahle Netto	
------------------	--

# Version History

---

Date	Version	Revision History	Author
17/07/2013	1.0		José Sahle Netto

## Abastract

This paper aims to present the features of **SHLStudio** and plug-in **Stored Procedures And C# Code Generator**.

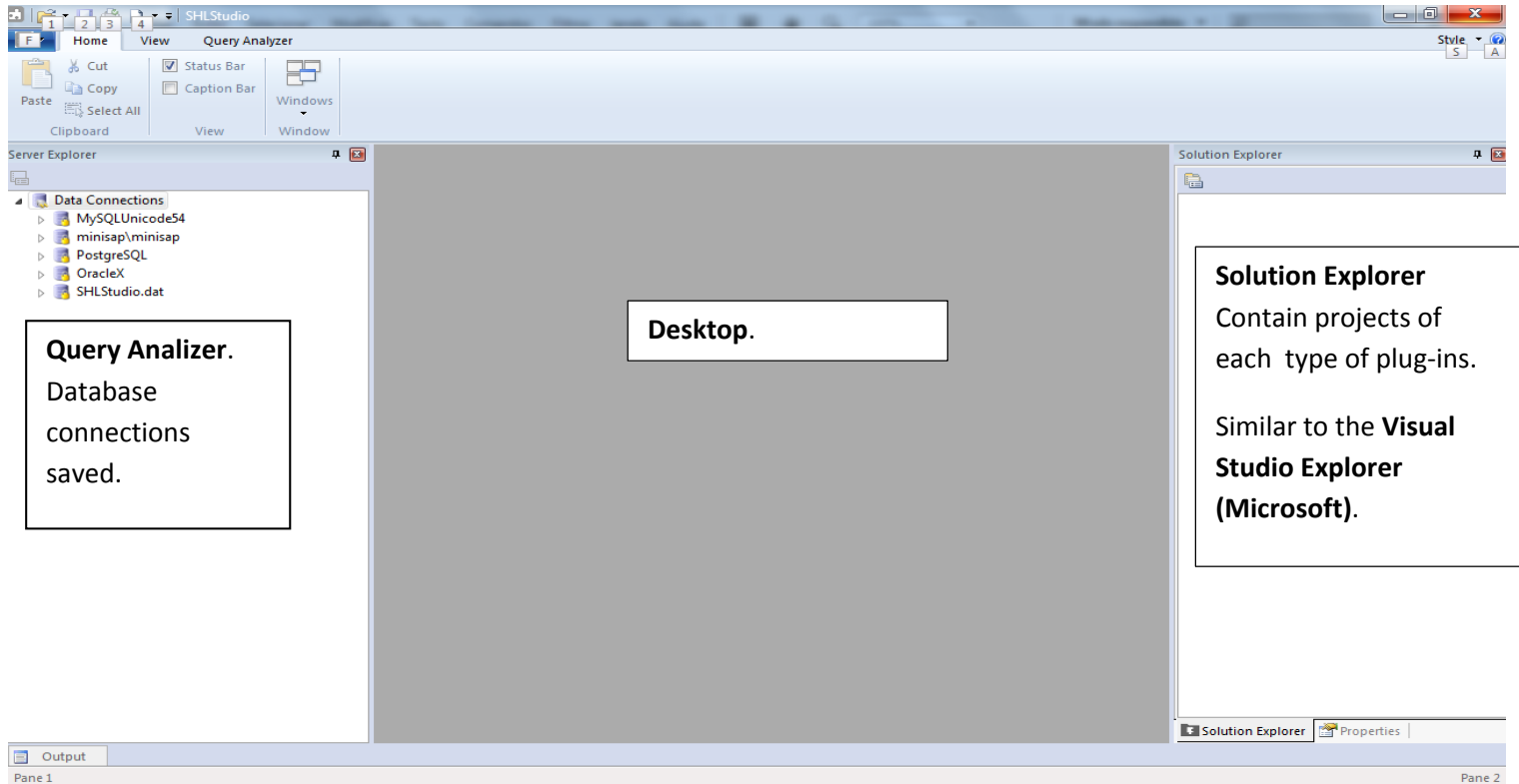
The design **SHL Studio** is written in C++ using MFC classes.

This project is in testing phase, which I'm testing and correcting faults.

# SHLStudio

The **SHLStudio** system is an IDE that has the purpose of docking functionality via plug-ins.

The system has a built-in **Query Analyzer**.



Indicated at left area are the connections made to the databases (MS-SQL Server, Oracle, MySQL, PostGeSQL, and any other via OLE DB).

Have a workspace, where **script** editors will be opened. This site will also be used to open the features of the **plug-ins**.

Finally, on the right is the **SOLUTION EXPLORER**, which will manage the projects of each type of plug-in. These projects will be grouped within a similar solution to the **Microsoft Visual Studio**.

# Query Analyzer

The screenshot displays the Query Analyzer window in SHLStudio. The SQL window contains the following queries:

```
1 SELECT * FROM GP_ROTINA
2 SELECT * FROM GP_DLL
```

The results pane shows the following data:

	idRotina	Nome
<b>Executed query</b>		
SELECT * FROM GP_ROTINA		
<b>Result</b>		
1	2	TJBAProjudi
2	4	Distribuidor-PainelAutomator
3	6	SaneamentoPilotoPCES
4	7	TJSudesteAutomator
5	8	TJNorteAutomator
6	9	TJNordesteAutomator
7	10	TJCentroOesteAutomator
8	11	TJSulAutomator
9	15	Gerador de Código
10	16	Gerador de Código
11	17	Gerador de Código
12	18	Gerador de Código

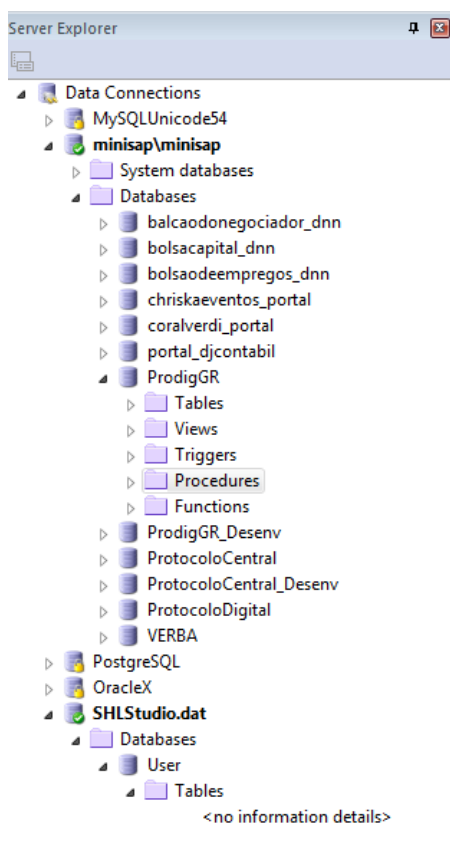
The interface also includes a Server Explorer on the left showing the database structure, and a Solution Explorer on the right.

The **Query Analyzer** has many similar to **SQL Server Management Studio** functionality, but extended to any databases supported by OLE DB.

	idRotina	Nome	
Executed query			
SELECT * FROM GP_ROTINA			
Result			
1	2	TJBAProjudi	Robo Projudi
2	4	Distribuidor-PainelAutomator	Robo Importação Fila 09
3	6	SaneamentoPilotoPCES	Saneamento do diretório Piloto PCES
4	7	TJSudesteAutomator	TJAutomator dos estados da região Sudeste
5	8	TJNorteAutomator	TJAutomator dos estados da região Norte
6	9	TJNordesteAutomator	TJAutomator dos estados da região Nordeste
7	10	TJCentroOesteAutomator	TJAutomator dos estados da região Centro-Oeste
8	11	TJSulAutomator	TJAutomator dos estados da região Sul
9	15	Gerador de Código	Rotina geradora de códigos
10	16	Gerador de Código	Rotina geradora de códigos
11	17	Gerador de Código	Rotina geradora de códigos
12	18	Gerador de Código	Rotina geradora de códigos

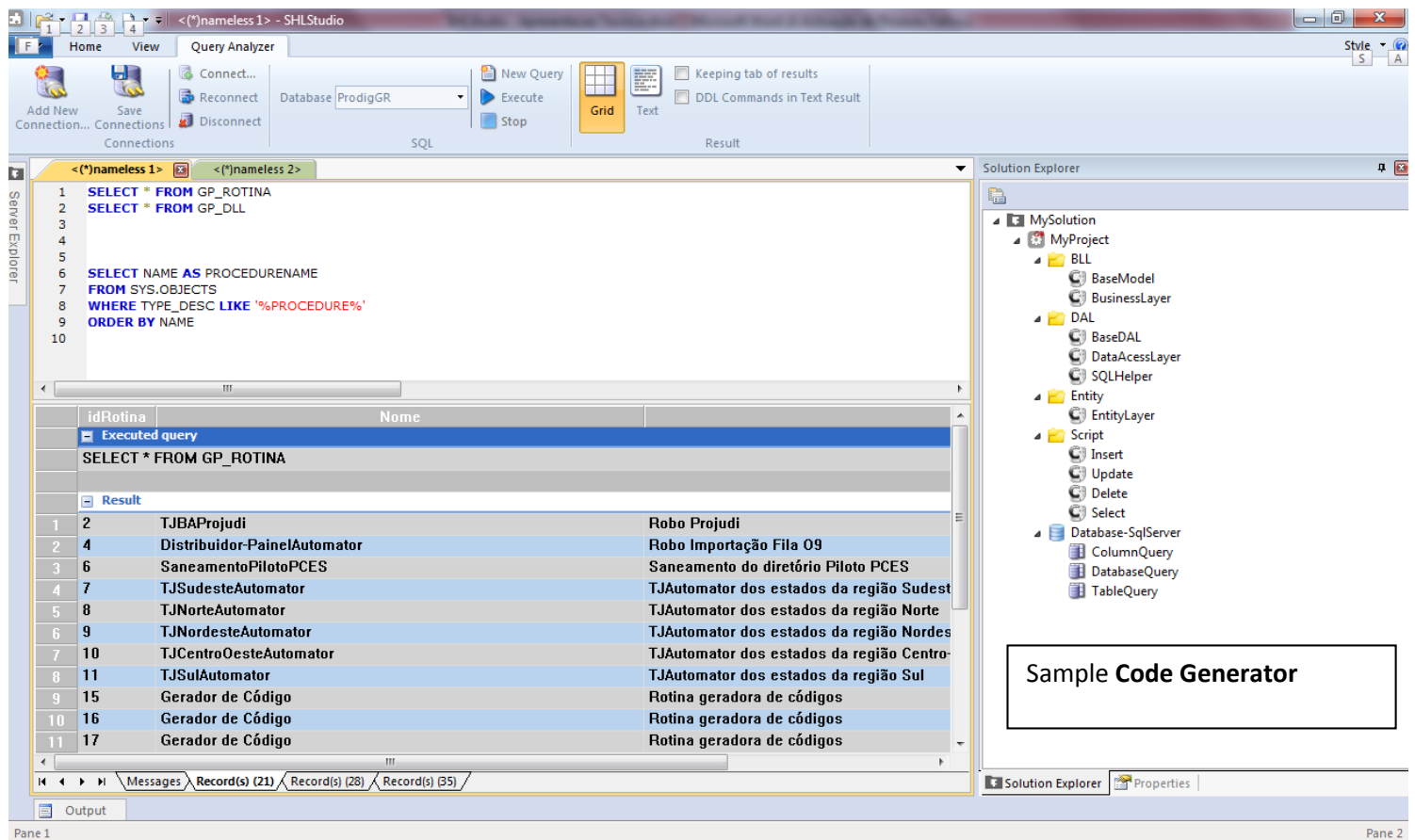
Messages | Record(s) (21) | Record(s) (28)

Unlike **SQL Management Studio** when there are more than one recordset are not shown below each other. In these cases the results will be presented in separate tabs. Another importante detail is the on top of the result grid, there is the sentence that generated the result, so with multiple queries, it is easy identify the both, query and result.



In the **Server Explorer** has additional information as:

- Tables
- Views
- Triggers
- Functions
- Procedures

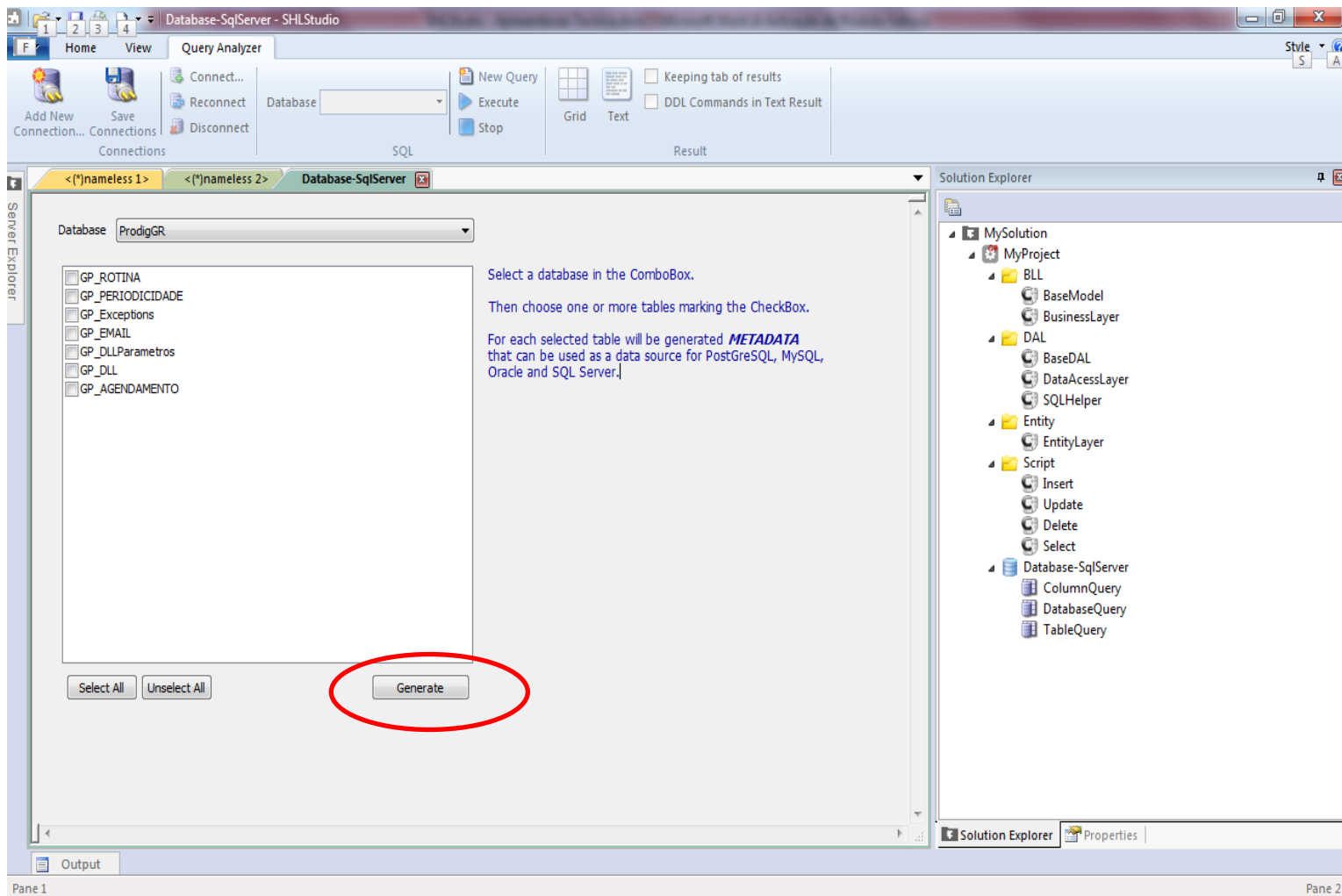


The ***Solution Explorer*** has the design of a particular plug-in. Within the solution will be possible to have diferentes desings plug-ins.

# C# Code Gerator

The generator *plugin* for **C# code** is a routine that generates C# code for the three (3) main layers (*entity*, *business* and *data*). This plug-in also generates **stored procedures** used in *data layer*.

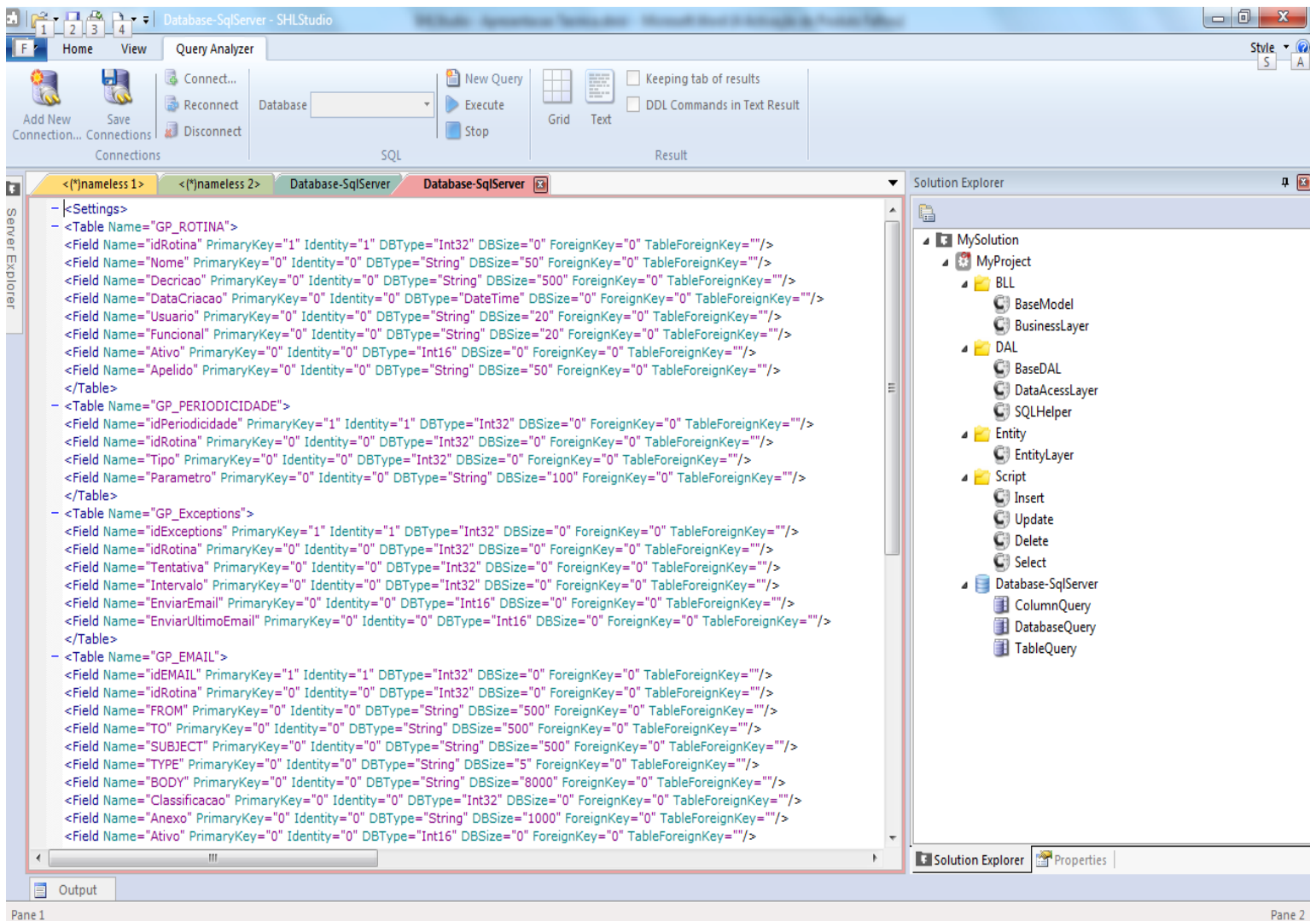
You can extend the functionality of the Code Generator because it generates based on **databases** and **tables** using tags in replacement of ***template files***.



The **Database-SqlServer** entity is actually a metadata generated from a connection with a particular database, which can be (Oracle, PostgreSQL, MySQL and MS-SQL Server).

From these connections generates a metadata representing the database, which is an XML file, and can be edited externally, although in most cases there is no need to do.



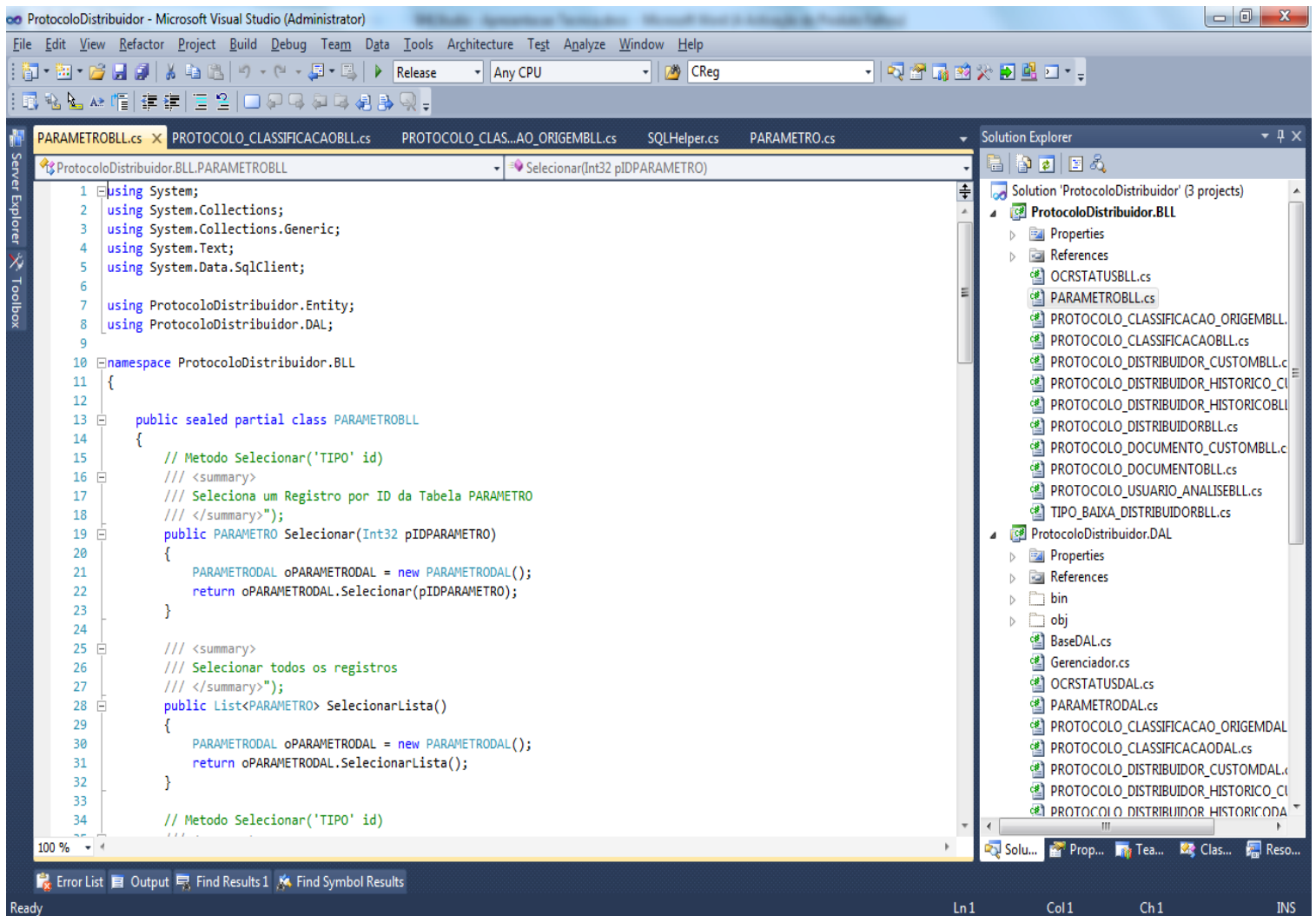


Above is the result of the XML metadata generated from a database and its tables previously selected.

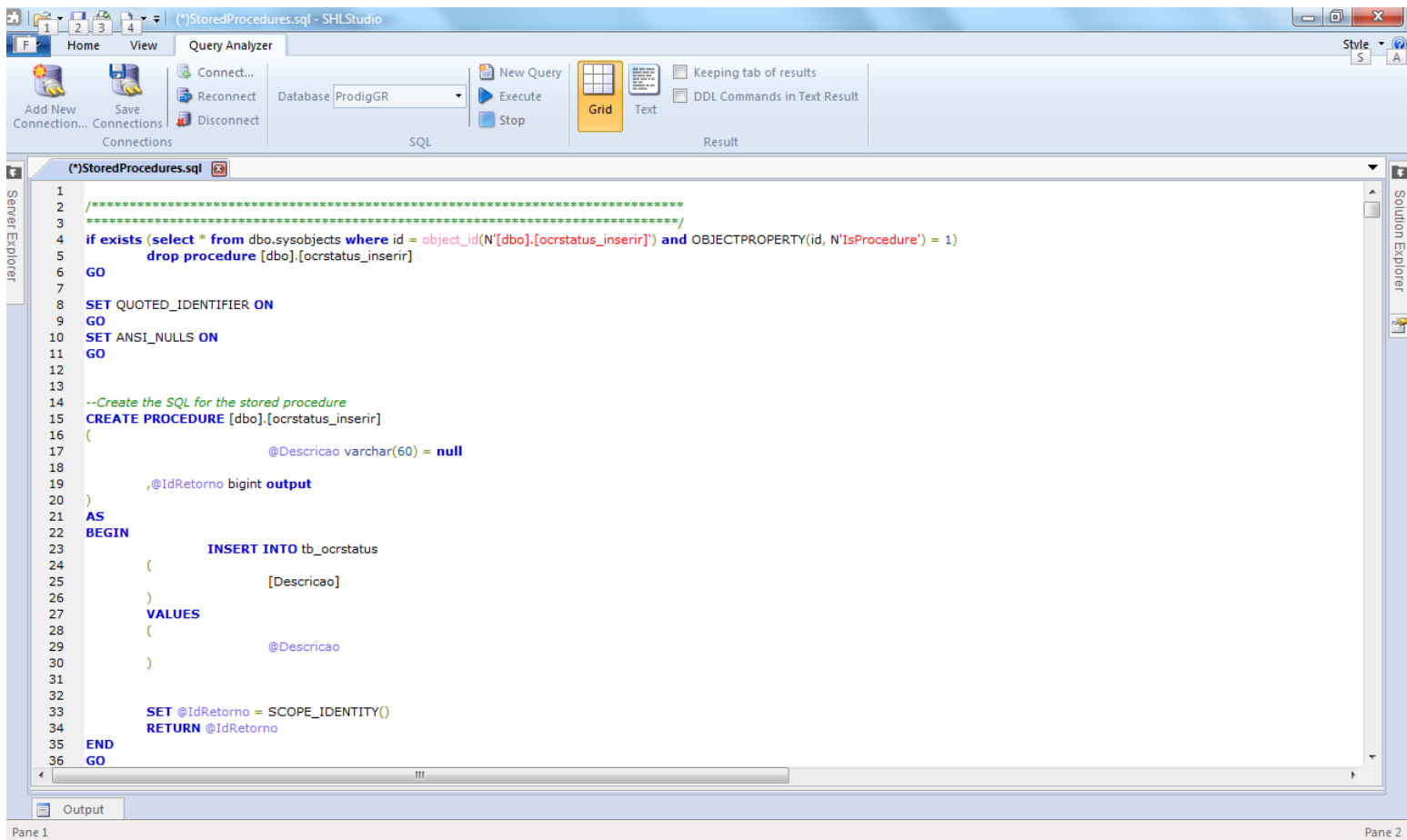


# Example of Generated Code

Below an example of code generated from the plug-in **Code Generantor**.



The classes listed in this project were created by the **Code Generator**, which did not take even 1 minute, cosidering the selection of tables and the “order” to generate the classes.



**Stored procedures** were also generated at the same time.

Totaling 12 selected tables and resulting in 36 C# files (entity, business and data), and 60 stored procedures (insert, update and delete queries).