

PROFICY HISTORIAN E SQL LINKED SERVER

Informações de produção consolidadas através de indicadores e armazenadas em historiadores de processo, como o **Proficy Historian**, muitas vezes devem ser integradas com os sistemas de gestão das empresas, permitindo assim, uma melhor visibilidade, controle e planejamento das operações.

Através da ferramenta OLEDB do Proficy Historian, os dados armazenados no servidor poderão ser acessados por aplicativos externos, como é o caso da integração com o banco de dados Microsoft SQL Server através da funcionalidade *Linked Server*.

Essa “ponte de comunicação” entre servidores SQL e OLE DB facilita a estruturação de consultas e a geração de relatórios de diferentes fontes de dados.

Nesse Faça Fácil serão apresentadas as etapas de configuração necessárias para integrar os dados do Proficy Historian com o banco de dados SQL

1. Configuração do Linked Server

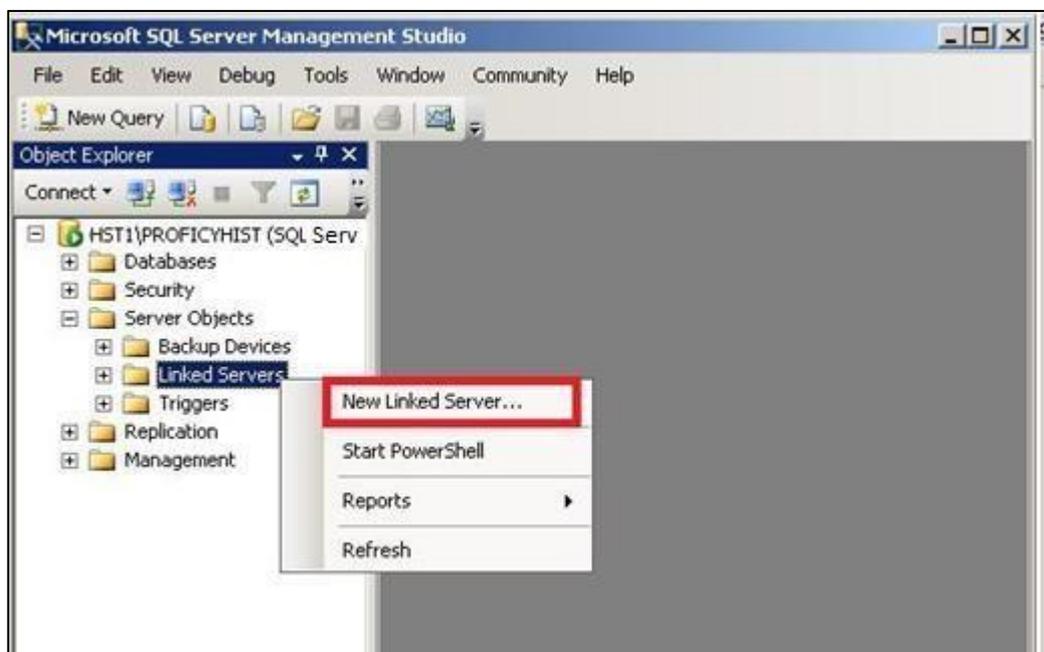
1.1 A partir do menu Iniciar, selecione All Programs > SQL Server Management Studio.



1.2 Clique em **Connect** para abrir o Banco SQL.

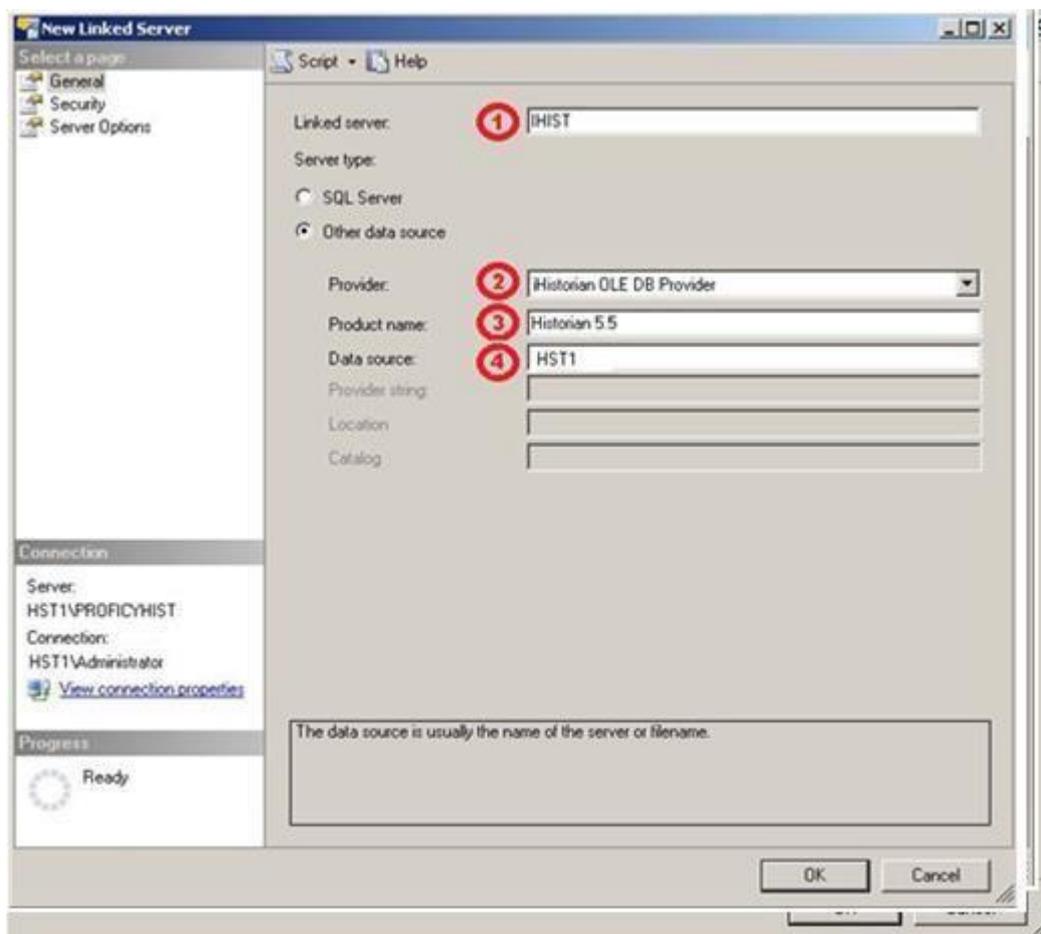


1.3 Abra a pasta **Server Objects** e clique com o botão direito do mouse na pasta **Linked Servers**. Selecione a opção **New Linked Server....**

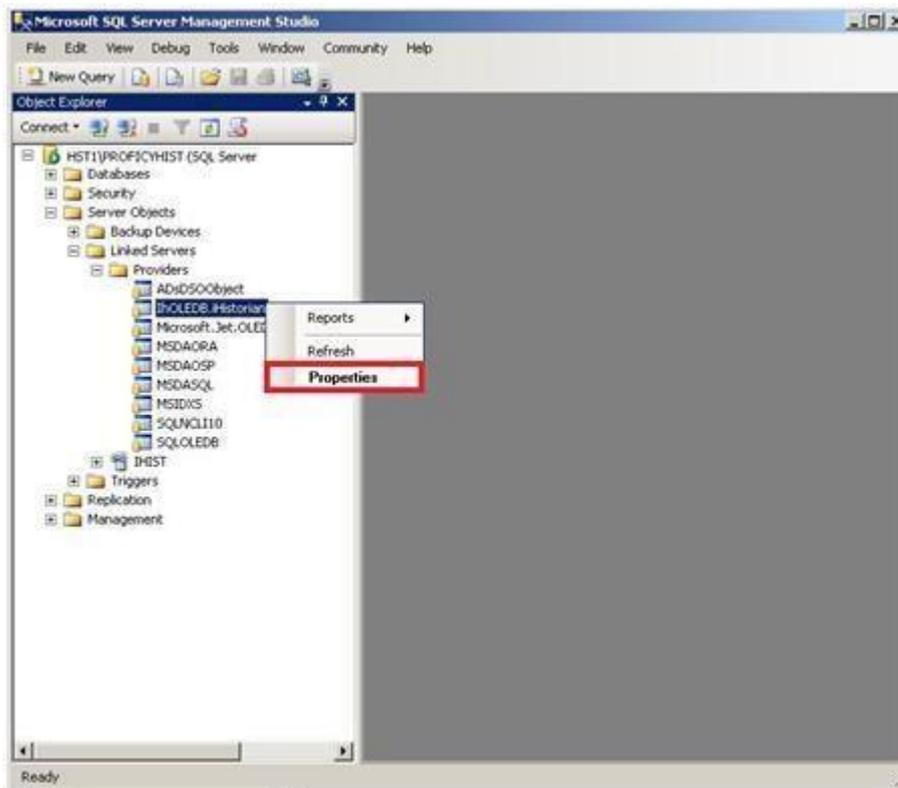


1.4 Na configuração do novo **Linked Server**, configure os seguintes campos:

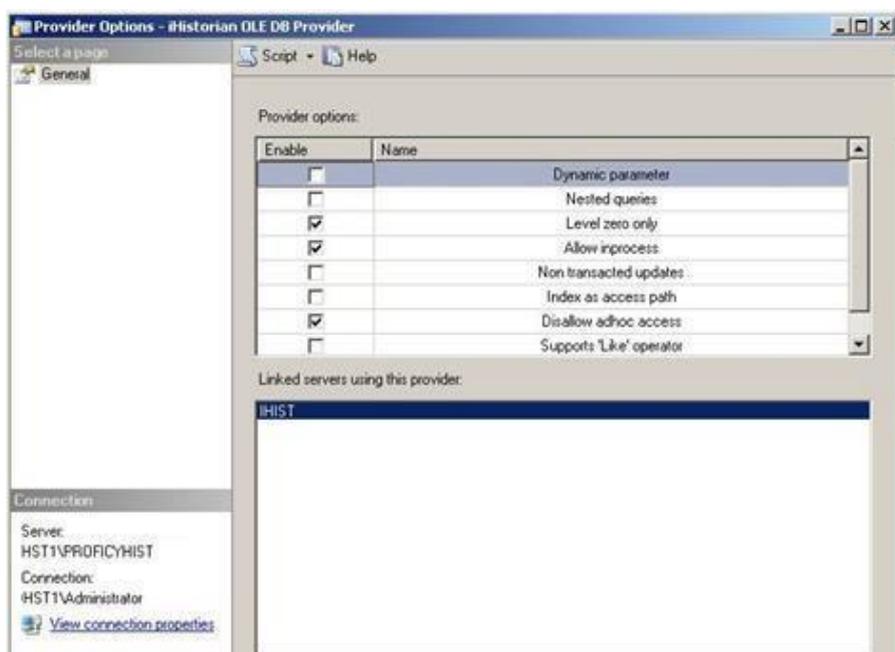
- (a) Digite o nome do Linked Server, como IHIST;
- (b) Selecione o iHistorian OLE DB Provider na lista de Provider;
- (c) Digite o nome do Produto, como por exemplo **Historian 5.5**;
- (d) Digite o nome do Servidor Historian no campo Data Source;
- (e) Clique em OK para finalizar a configuração.



1.5 Expanda a pasta **Providers**, localizada dentro da pasta **Linked Servers**. Clique com o botão direito do mouse no Provider **HOLEDB.iHistorian** e selecione a opção **Properties**.



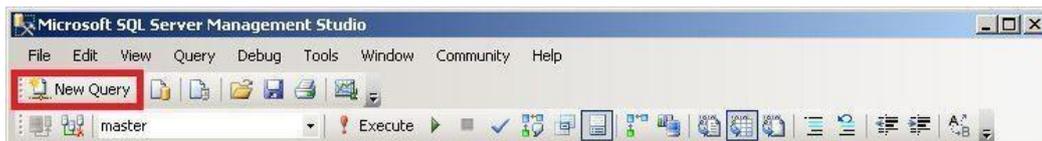
1.6 Habilite as seguintes opções para o *provider* selecionado:



2. Testando o acesso ao Linked Server

Após o término da configuração, a conexão poderá ser testada através da ferramenta *SQL Express*.

2.1 Selecione a opção **New Query**, para criar uma nova consulta.



2.2 Digite os seguintes comandos na *Query*:

Exemplo 1: Consulta para retornar a lista de tags configuradas no Servidor Historian.

```
SELECT * FROM OPENQUERY(iHist,'SELECT * FROM ihTags')
```

Exemplo 2: Executa uma consulta complexa para extrair os valores de um determinado tag do Historiador.

```
SELECT TagName, TimeStamp, Value, Quality FROM OPENQUERY  
(iHist,  
SET  
StartTime=Today,  
EndTime=Now,  
IntervalMilliseconds=1Hour,  
SamplingMode=Calculated,  
CalculationMode=Maximum  
SELECT * FROM ihRawData WHERE TagName LIKE *TESTE1.F_CV')
```

| | TagName | TimeStamp | Value | Quality |
|---|-------------|-----------------------------|---------------------|---------|
| 1 | TESTE1.F_CV | 2014-07-15 01:00:00.0000000 | 0.34180730581283569 | 100 |
| 2 | TESTE1.F_CV | 2014-07-15 02:00:00.0000000 | 0.34180730581283569 | 100 |
| 3 | TESTE1.F_CV | 2014-07-15 03:00:00.0000000 | 0.34180730581283569 | 100 |
| 4 | TESTE1.F_CV | 2014-07-15 04:00:00.0000000 | 0.34180730581283569 | 100 |
| 5 | TESTE1.F_CV | 2014-07-15 05:00:00.0000000 | 0.34180730581283569 | 100 |
| 6 | TESTE1.F_CV | 2014-07-15 06:00:00.0000000 | 0.34180730581283569 | 100 |

Dessa maneira, os dados de processo armazenados no Proficy Historian passam a estar disponíveis no banco de dados relacional, e poderão ser relacionadas com as informações já contidas em diferentes tabelas do SQL

3. Habilitando o acesso remoto ao Linked Server.

3.1. A partir do Menu Iniciar > All Programs, abra o SQL Server Management Studio.

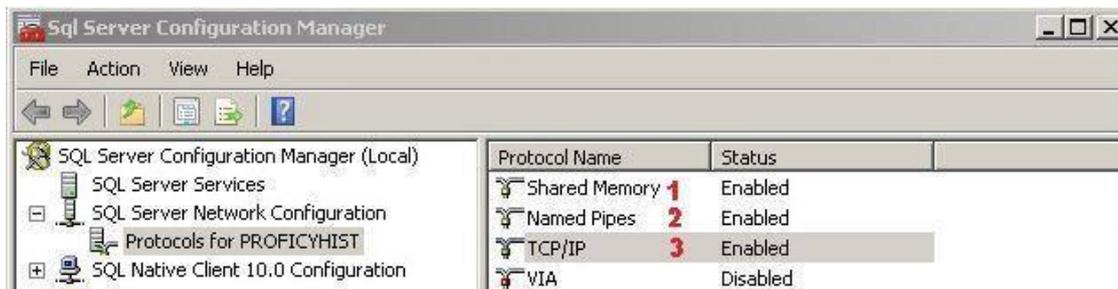


3.2. Na opção **SQL Server Services**, verificar se os serviços **SQL Server** e **SQL Server Browser** estão iniciados e em modo automático:

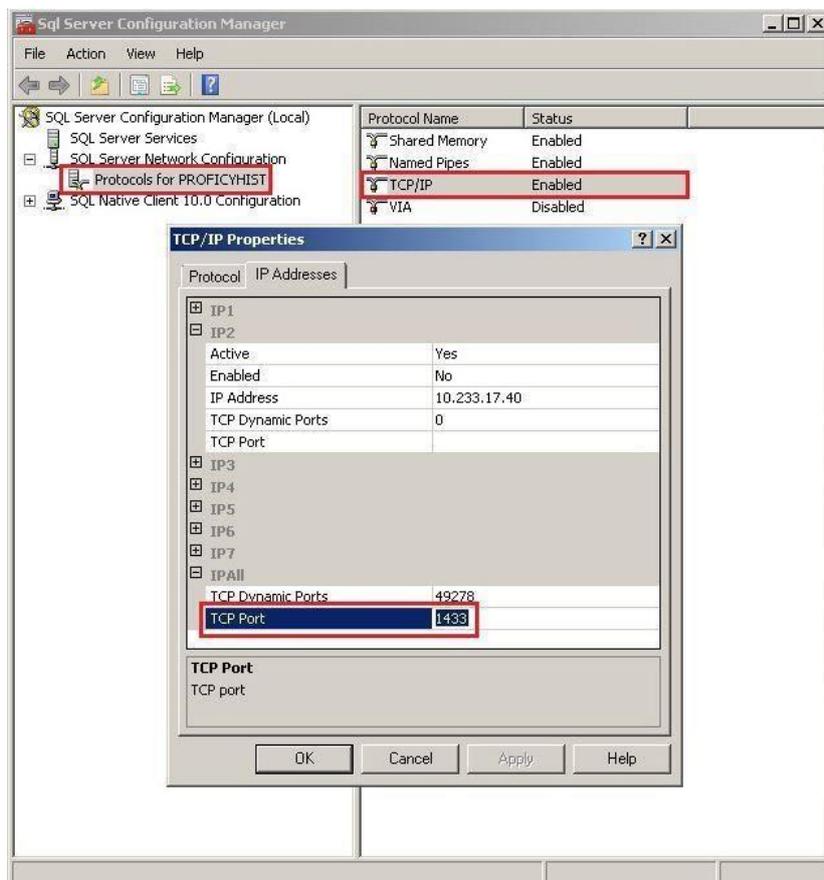


3.3. Selecionar a configuração de **Protocols for PROFICYHIST** dentro de **SQL Server Network Configuration**.

- (a) Habilitar o protocolo Shared Memory; (b) Habilitar o protocolo Named Pipes;
(c) Habilitar o protocolo TCP/IP.



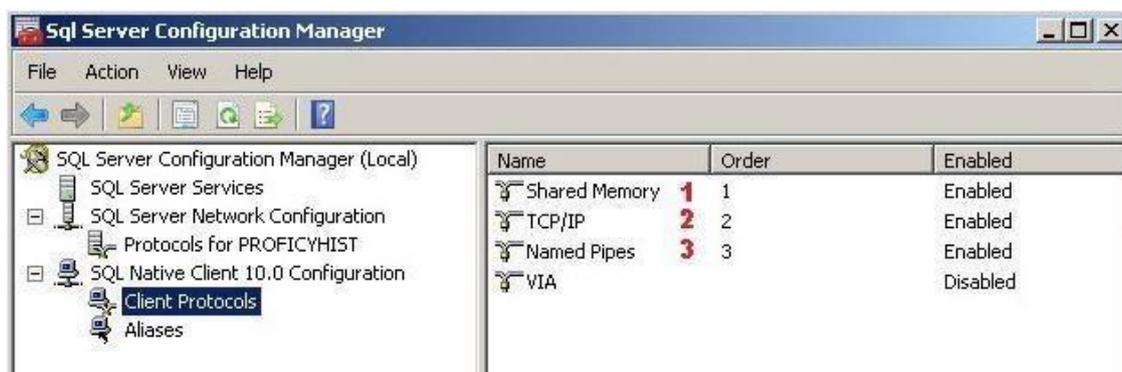
3.4. Clique com o botão direito do mouse no protocolo TCP/IP, selecione as propriedades e configure a opção **TCP Port** para **1433** dentro de **IPAll**:



3.5. Selecionar a configuração de **Client Protocols** dentro de **SQL Native Client 10.0 Configuration**.

(a) Habilitar o Protocolo Shared Memory; (b) Habilitar o Protocolo Named Pipes;

(c) Habilitar o Protocolo TCP/IP.



Com essa configuração adicional, será possível o desenvolvimento de aplicativos externos para acesso às informações do banco de dados relacional. O banco de dados passa a ser a ponte de comunicação entre a rede de automação e a rede corporativa.

Abaixo, segue o exemplo de um código desenvolvido em linguagem JAVA no NetBeans, cuja consulta retorna informações do Proficy Historian através do Linked Server.

```
/*
 *      To change this license header, choose License Headers in Project Properties.
 *      To change this template file, choose Tools | Templates
 *      and open the template in the editor.
 */
package testemssql;

import java.sql.Connection; import java.sql.DriverManager; import java.sql.PreparedStatement; import
java.sql.ResultSet;

public class TesteMSSql {

    /**
     *      @param args the command line arguments
     */
    @SuppressWarnings({"CallToPrintStackTrace", "BroadCatchBlock", "TooBroadCatch", "UseSpecificCatch"})
    public static void main(String[] args) {
        // TODO code application logic here
        String connectionUrl = "jdbc:sqlserver://10.233.17.40\\PROFICYHIST:1433;user=sa;password=Pr0f1cyhist;";
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            Connection con = DriverManager.getConnection(connectionUrl);
```

```
PreparedStatement pst = con.prepareStatement("SELECT * FROM OPENQUERY(iHist,'SELECT * FROM  
ihTags')");    ResultSet rs = pst.executeQuery();  
  
while(rs.next()){  
    System.out.println("Tagname: "+ rs.getString(1));  
    System.out.println("Description: "+ rs.getString(2));  
    System.out.println("EGU: "+ rs.getString(3));  
    System.out.println("-----");  
}  
} catch (Exception ex) {  
    System.out.println(">>>> " + ex);    ex.printStackTrace();  
}  
}  
}
```

