

Tutorial

MySQL + C#

Objectivo:

O objectivo deste documento é oferecer suporte à ligação da linguagem de programação C#, da Microsoft, ao sistema gestor de base de dados MySQL sem utilizar um controlador ODBC como intermediário.

Motivação:

O uso do MySQL pode ser interessante para aqueles que pretendem aliar a linguagem C# a um sistema gestor de base de dados gratuito com suporte de transacções. Qualquer uma das ferramentas (compilador, SGBD e driver) referidas neste documento são de uso gratuito, e como tal representam uma boa alternativa ao uso de Microsoft SQL Server, Microsoft Access e outros. O Microsoft Visual Studio .NET está disponível, como é sabido, na biblioteca da LEIC:

Por outro lado, para o objectivo concerto do projecto de ARGE, a inexistência de um SGBD digno desse nome nas instalações dos laboratórios (devido a problemas de segurança como por exemplo worms) e sendo a uma alternativa o MS Access (não suporta transacções entre outras limitações), torna-se problemático garantir a coerência de dados quando acedidas por threads e/ou processos diferentes.

Instalação:

A instalação resume-se à inclusão de uma biblioteca de programação no projecto e depende do IDE que se está (ou não) a utilizar. De seguida descreve-se a inclusão da referida biblioteca no MS Visual Studio .NET

1. Obter uma cópia do controlador (biblioteca DLL) em <http://mega.ist.utl.pt/~gedl/ARGE/publico/MySQL+CSharp.zip> (aproximadamente 81Kb);

2. Copiar os dois ficheiros (driver e compressor de dados, não sendo o uso deste último contemplado neste documento) para a estrutura de directórios do projecto;
3. Incluir a directiva “[using](#) ByteFX.Data.MySQLClient;” no código e adicionar a referência ao projecto:
 - a. Menu view → source Explorer;
 - b. Botão direito no nome do projecto;
 - c. Add reference;
 - d. Browse e apontar para o ficheiro ByteFX.Data.dll;

Configuração:

Após concluída a instalação do driver, é necessário configurar o ambiente de trabalho de modo a que os PCs do laboratório tenham acesso à vossa base de dados no MEGA.

Por omissão o MySQL apenas aceita ligações vindas do “localhost” ou seja, de dentro da máquina onde está instalado. É o que sucede no MEGA. Para configurar e terem acesso à base de dados a que têm direito no ~~m~~ega devem seguir os seguintes passos:

1. Fazer login no MEGA e escrever “mysqlreset”. Este comando terá um output semelhante a:

```
[gedl@grace gedl]$ mysqlreset
login: gedl
database: gedl
password: xslq3563
[gedl@grace gedl]$
```

Isto quer dizer que a vossa base de dados no MySQL se chama “gedl” e o par nome/senha é gedl/xslq3563. Este comando pode ser executado inúmeras vezes sem se perder o conteúdo da base de dados.

2. Após garantir a configuração do MySQL é necessário contornar a limitação acima referida, deste apenas aceitar ligações a partir do localhost. Como tal vai ser criado um túnel seguro (SSH) entre a porta 3306 da máquina local e a porta 3306 do MEGA, para que este considere que as ligações que recebe vêm de dentro. Nos PCs dos laboratórios está disponível o SSH2.EXE, um cliente de linha comandos. Devem:

a. Executar o seguinte comando:

```
$ssh2 -l <login> mega -L 3306:localhost:3306
```

b. Após a execução do comando anterior fica aberta uma shell normal mas simultaneamente foi criado um túnel (ou ponte) SSH. Configurem a vossa shell para não desligar ao fim de 30 minutos ou então ponham um programa a correr (por exemplo o Pine).

Neste momento tudo deve estar a funcionar. Para testar (e talvez utilizar como ferramenta visual de edição da base de dados) recomendo o MySQLFront, conhecido para os alunos que tiveram EP e que está disponível em

http://mega.ist.utl.pt/~ic-ep/docs/material/software/MySQL-Front_2.5_Setup.exe

Utilização:

Este pequeno exemplo mostra como fazer uma ligação à base de dados no MySQL. Sobre esta ligação podem ser executados os comandos “IDbCommand” e as classes “IDataReader” e outras que o C# disponibiliza e que estão documentadas pela Microsoft. Para mudar de SGBD para o Microsoft SQL Server, por exemplo, (aconselhado) deverá ser apenas necessário mudar a ligação “MySQLConnection”.

ANTEÇÃO: não é garantido que assim seja, pois nunca experimentei.

O exemplo pressupõe a existência da tabela criada através da seguinte Query:

```
CREATE TABLE utilizador
(
  nome varchar (32),
  senha varchar (32)
);
```

Devem também modificar os parâmetros da ligação conforme cada caso.

Código:

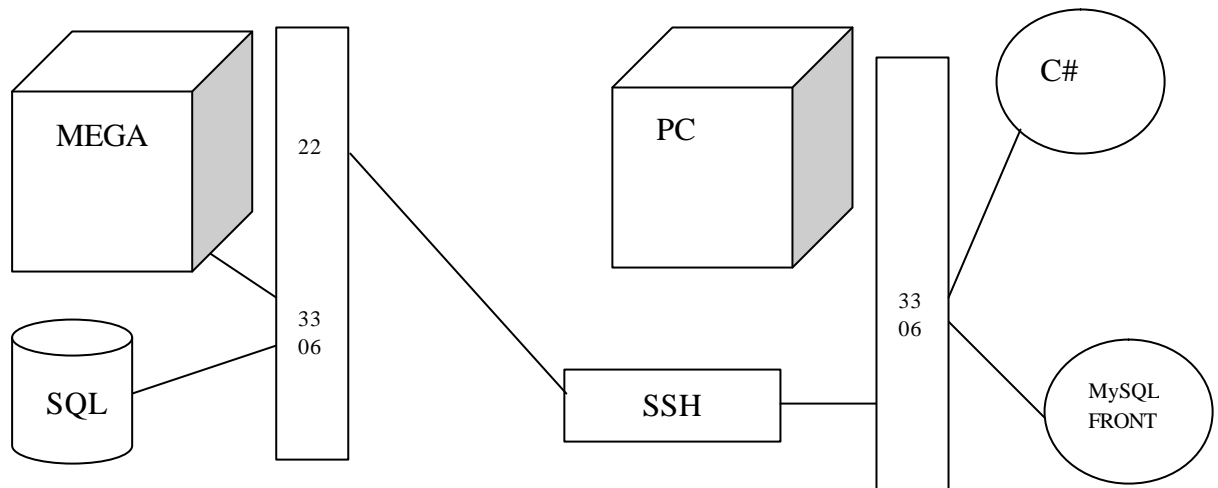
```
using System;

using System.Data;
using ByteFX.Data.MySQLClient;

public class Test
{
    public static void Main(string[] args)
    {
        string connectionString =
            "Server=localhost;" +
            "Database=gedl;" +
            "User ID=gedl;" +
            "Password=ixual599;";
        IDbConnection dbcon;
        dbcon = new MySqlConnection(connectionString);
        dbcon.Open();
        IDbCommand dbcmd = dbcon.CreateCommand();
        string sql =
            "SELECT nome, senha " +
            "FROM utilizador";
        dbcmd.CommandText = sql;
        IDataReader reader = dbcmd.ExecuteReader();
        while(reader.Read())
        {
            string FirstName = (string) reader["firstname"];
            string LastName = (string) reader["lastname"];
            Console.WriteLine("Name: " + FirstName + " " +
                LastName);
        }
        reader.Close();
        dbcmd.Dispose();
        dbcon.Close();
    }
}
```

Conclusão:

A configuração final é a representada no seguinte diagrama (peço desculpa pela qualidade do desenho). As máquinas envolvidas podem ser as que quiserem:



Como tal todos os acessos à base de dados devem ser dirigidos à porta 3306 da máquina local (localhost). O túnel SSH pode, obviamente, ser aberto para qualquer máquina onde esteja a correr um servidor MySQL

Notas:

Existem inúmeras soluções alternativas desde outros drivers para o C# a outros métodos de fazer túneis. Este documento apenas apresenta UMA solução que penso ser adequada.

É possível que o driver, o MySQL, o Visual Studio, o SSH ou mesmo o Windows tenham bugs. Claro que isto não está previsto no documento.

IMPORTANTE: a versão MySQL disponível no MEGA não suporta chaves estrangeiras, pelo que qualquer tipo de integridade referencial deve ser garantida na camada de negócio. Podem, claro, instalar o MySQL 4.0 na vossa área (se a quota chegar).

Se alguém souber como usar o PostgreSQL ou o Sybase, faça-se ouvir.

Algum erro ou sugestão deve ser enviado para gedl@mega.ist.utl.pt e para o corpo docente.