

Views

Abra o banco de dados

Crie uma View no banco de dados webheranca

```
CREATE VIEW `vw_todoscarros_com_piloto` AS
SELECT tbl_carro.car_codigo, tbl_carro.car_numero, tbl_carro.car_potencia,
tbl_carro.car_alturabico, tbl_carro.car_foto, tbl_carro.car_alturaaerofolio,
tbl_membro.mem_codigo, tbl_membro.mem_nome, tbl_membro.mem_contrato,
tbl_membro.mem_tipo, tbl_membro.mem_peso, tbl_membro.mem_altura
FROM tbl_carro
INNER JOIN tbl_membro
ON tbl_carro.mem_codigo=tbl_membro.mem_codigo
WHERE tbl_membro.mem_tipo = 1
ORDER BY tbl_carro.car_numero;
```

Abrir o Visual Studio

Abrir o projeto WebHeranca

Altere o método SelectAllWithPilots da classe CarroBD.cs

```
public DataSet SelectAllWithPilots()
{
    DataSet ds = new DataSet();

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    System.Data.IDataAdapter objDataAdapter;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command("SELECT * FROM vw_todoscarros_com_piloto",
objConexao);

    objDataAdapter = Mapped.Adapter(objCommand);
    objDataAdapter.Fill(ds);

    objConexao.Close();

    objCommand.Dispose();
    objConexao.Dispose();

    return ds;
}
```

Execute a aplicação.

Mais informações em <http://www.mysqltutorial.org/create-sql-views-mysql.aspx>

Stored Procedure

Stored Procedure para Consulta

Crie uma Stored Procedure no banco de dados

```
CREATE PROCEDURE `sp_carros_com_potencia_maior_que` (IN potencia INT)
BEGIN

    SELECT * FROM tbl_carro
    WHERE car_potencia >= potencia;

END
```

Em CarroBD.cs, crie um método para Mostrar os carros que tem a potência maior que um determinado valor.

```
public DataSet SelectAllByPower(int power)
{
    DataSet ds = new DataSet();

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    System.Data.IDataAdapter objDataAdapter;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command("sp_carros_com_potencia_maior_que", objConexao);
    objCommand.CommandType = CommandType.StoredProcedure;

    objCommand.Parameters.Add(Mapped.Parameter("?potencia", power));

    objDataAdapter = Mapped.Adapter(objCommand);
    objDataAdapter.Fill(ds);

    objConexao.Close();

    objCommand.Dispose();
    objConexao.Dispose();

    return ds;
}
```

Para fazer uso desse método, basta chamá-lo passando um valor:

```
CarroBD bd = new CarroBD();
DataSet ds = bd.SelectAllByPower(100);
```

Stored Procedure para verificar se um registro existe

```
CREATE PROCEDURE `sp_carroexistente` (IN numero INT, OUT codigo INT)
BEGIN

SELECT car_codigo INTO codigo
FROM tbl_carro
WHERE tbl_carro.car_numero = numero;

END
```

Para testar no MySQL, use:

```
set @codigo =15;
CALL sp_carroexistente (@codigo, @cidade);
SELECT @cidade;
```

Onde 15 é o Número do carro.

Para chamar essa Stored Procedure no software, serão necessárias algumas modificações

Altere o Mapped.cs

```
using System;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

//Importa funções do MySQL
using MySql.Data.MySqlClient;
//Trabalhar com Dataset
using System.Data;
//Permite visualizar o web.config
using System.Configuration;

namespace FATEC
{
    /// <summary>
    /// Summary description for Mapped
    /// </summary>
    public class Mapped
    {
        //Abrir conexao
        public static IDbConnection Connection()
        {
            MySqlConnection conn = new
            MySqlConnection(ConfigurationManager.AppSettings["strConexao"]);
            conn.Open();
            return conn;
        }
    }
}
```

```

//Executa comando no BD
public static IDbCommand Command(string query, IDbConnection conexao)
{
    IDbCommand comando = conexao.CreateCommand();
    comando.CommandText = query;
    return comando;
}

//Retorna um Adapter (SELECT)
public static IDataAdapter Adapter(IDbCommand comando)
{
    IDbDataAdapter adap = new MySqlDataAdapter();
    adap.SelectCommand = comando;
    return adap;
}

//Cria parametro da SQL
public static IDbDataParameter Parameter(string nome, object valor)
{
    return new MySqlParameter(nome, valor);
}

```

```

public static IDbDataParameter Parameter_StoredProcedureIN(string nome, object
valor)
{
    MySqlParameter parameter = new MySqlParameter(nome, valor);
    parameter.Direction = ParameterDirection.Input;
    return parameter;
}

```

```

public static IDbDataParameter Parameter_StoredProcedureOUT(string nome, TIPOS type)
{

```

```

    MySqlDbType tipo = 0;
    switch (type)
    {
        case TIPOS.INTEGER:
            tipo = MySqlDbType.Int32;
            break;
        case TIPOS.VARCHAR:
            tipo = MySqlDbType.VarChar;
            break;
        case TIPOS.TEXT:
            tipo = MySqlDbType.Text;
            break;
        default:
            break;
    }

```

```

    MySqlParameter parameter = new MySqlParameter(nome, tipo);
    parameter.Direction = ParameterDirection.Output;
    return parameter;
}

```

```

public static int GetValue(IDbCommand command, string param)
{
    MySql.Data.MySqlClient.MySqlCommand comando =
(MySql.Data.MySqlClient.MySqlCommand)command;

```

```

        int retorno = 0;
        if (comando.Parameters[param].Value != DBNull.Value)
        {
            retorno = Convert.ToInt32(comando.Parameters[param].Value);
        }
        return retorno;
    }
}

```

```

        public static IDbCommand Command_StoredProcedure(string query, IDbConnection
conexao)
        {
            IDbCommand comandi = conexao.CreateCommand();
            MySql.Data.MySqlClient.MySqlCommand comando =
(MySql.Data.MySqlClient.MySqlCommand) comandi;
            comando.CommandText = query;
            comando.CommandType = CommandType.StoredProcedure;
            return comando;
        }
}

```

```

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

```

```

public enum TIPOS
{
    INTEGER,
    VARCHAR,
    TEXT
}
}

```

Em CarroBD.cs, crie o método

```

public int IsExisteByNumber(int numero)
{
    DataSet ds = new DataSet();

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command_StoredProcedure("sp_carroexistente", objConexao);

    objCommand.Parameters.Add(Mapped.Parameter_StoredProcedureIN("?numero",
numero));
    objCommand.Parameters.Add(Mapped.Parameter_StoredProcedureOUT("?codigo",
TIPOS.INTEGER));

    objCommand.ExecuteNonQuery();

    int retorno = Mapped.GetValue(objCommand, "?codigo");

    objConexao.Close();

    objCommand.Dispose();
    objConexao.Dispose();
}

```

```
        return retorno;  
    }
```

Para executar o método, coloque o trecho a seguir em alguma página:

```
CarroBD bd = new CarroBD();  
int retorno = bd.IsExisteByNumber(15);  
Response.Write(retorno);
```

Execute a aplicação.