Verslag database koppelen met applicatie v0.1– Team Smokey

Laatst bijgewerkt: 15-5-2014  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Projectbegeleider: Sietse Dijks

Klas: RIO4-APO1F, RIO4-APO1E

Periode: 4

Project: Project FIFA

Inhoud

[Keuzes 2](#_Toc387909922)

[SQLite 2](#_Toc387909923)

[MySQL 2](#_Toc387909924)

[SQL 2](#_Toc387909925)

[SQL Compact 2](#_Toc387909926)

[Beste keuze 2](#_Toc387909927)

[Hoe koppel je een database met C#? 2](#_Toc387909928)

# Keuzes

Wij hebben opgezocht welke databases er gekoppeld kunnen worden met onze c# applicatie, en wat wij dachten dat voor ons doeleinde de beste keuze zou zijn. Over de beste keuze komen we later op terug in dit document.

De volgende database types kunnen worden gebruikt voor onze applicatie:

### SQLite

Erg handig voor voor beginnende programmeurs, makkelijk uit te lezen, geen installatie nodig.  
XML

XML is handig om data uit te lezen via code.

### MySQL

Handig voor een applicatie, maar veel te ingewikkeld voor wat wij op het moment nodig hebben.

### SQL

SQL is enorm handig om te gebruiken in C#, want er zijn genoeg tutorials voor om te volgen en het is makkelijk te gebruiken wegens functies in visual studio.

### SQL Compact

SQL compact is goed voor kleinere databases, zoals een die wij nodig hebben. SQL compact is een goede keuze om te gebruiken in een applicatie zoals onze.Access

Access is op zich een goede database, maar voor onze doeleinde is Access een te uitgebreide database om te gebruiken.

### CSV

In CSV word alles gescheden met een komma, op die manier is het makkelijk te onderscheiden wat bij elkaar hoort. Ook is het enorm makkelijk (net als bij SQL) om data uit de database te lezen.

# Beste keuze

De beste optie lijkt ons SQL omdat we daar al een VHT over hebben gehad, en al begrijpen hoe SQL werkt en het is gewoon heel fijn om mee te werken in visual studio omdat die al heel veel mogelijkheden bied.

# Hoe koppel je een database met C#?

Een database koppelen met C# is vrij makkelijk. Eerst moet er een connectie geopend worden met de database, wat één regeltje code is. Vervolgens kunnen we gegevens uit de database halen, en inladen in een DataGridView, als laatst sluiten we de connectie.

# Connection Strings

Om te kijken welke code je nodig hebt voor een database is <http://www.connectionstrings.com/> een enorm handige website om op te kijken, daar staan alle databases met de benodigde code erbij.

# Code SQL

Omdat wij SQL gaan gebruiken voor onze software is hieronder aangegeven hoe je een SQL connectie opent en daar data uit haalt vanuit C# en Visual Studio.

using System;

using System.Data.SqlClient;

class Program

{

static void Main()

{

//

// First access the connection string.

// ... This may be autogenerated in Visual Studio.

//

string connectionString = ConsoleApplication1.Properties.Settings.Default.ConnectionString;

//

// In a using statement, acquire the SqlConnection as a resource.

//

using (SqlConnection con = new SqlConnection(connectionString))

{

//

// Open the SqlConnection.

//

con.Open();

//

// The following code uses an SqlCommand based on the SqlConnection.

//

using (SqlCommand command = new SqlCommand("SELECT TOP 2 \* FROM Dogs1", con))

using (SqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

Console.WriteLine("{0} {1} {2}",

reader.GetInt32(0), reader.GetString(1), reader.GetString(2));

}

}

}

}

}