



TextDateiLesen

Der Abschnitt `#Connection` stellt die Verbindung zur Access-Datenbank her.

Im Abschnitt `#Tabelle Games` werden die beiden benötigten Zugriffsmechanismen `SELECT` und `INSERT` für ein so `OleDb-Adapter-Objekt $Adapter` formuliert und für jedes Feld ein eigenes Parameter-Objekt definiert. Der Einfachheit halber werden alle Felder als Text importiert. Die Umwandlung in Zahlen (beim Feld `Spiel`) erfolgen später in der Datenbank durch Änderung des Feldtyps.

An dieser Stelle sieht man auch den nicht ganz reibungsfreien Entwicklungsvorgang. Anfangs wurde nämlich die Tabelle der Spiele `Games` genannt, damit sie mit einer bereits bestehenden Fußball-Datenbank zusammenpasst. Dann aber, als dieser Artikel entstehen sollte, wurde die Tabelle auf das deutsche `Spiele` umgeändert. Auch das sollte nicht bestehen bleiben und wurde nach Fertigstellung der Datenbank auf die Einzahl `Spiel` geändert. Aus Termingründen war es dann aber nicht mehr möglich, die Einzahlbildung auf die anderen Tabellen auszudehnen und daher ist zum Beispielen die Tabelle `Tore` immer noch in der Mehrzahl geschrieben.

Dieses Adapter-Objekt `$Adapter` kommuniziert mit einer `$DataTable`, die die Spaltennamen und Spaltentypen vom Adapter in `$Adapter.Fill($DataTable)` übernimmt. Anfangs ist diese Tabelle leer, lediglich die Anzahl und die Namen der Spalten ist definiert.

Die Datei `alles.txt` wird in die Variable `$Alles` eingelesen, eine neue Datenzeile `$DataRow` angelegt.

Die `for`-Schleife `for($i=0; $i -lt $Alles.length; $i++)` liest alle Zeilen des Textes und überprüft den Aufbau der Zeilen.

Beginnt eine Zeile mit `####`, wir die Variable `$Game` hochgezählt, der mit Daten gefüllte Datensatz `$DataRow` an die Tabelle `$DataTable` angefügt und ein neuer, leerer Datensatz `$DataRow` erzeugt und alle Werte in diesem Datensatz mit der Funktion `ClearValues` gelöscht.

Beginnt der Datensatz mit `#Spiel#` wird zur Kontrolle ein Text ausgegeben.

Die Regular Expression `$regex = [regex] '#(.+)(.*)'` findet Zeilen, die den gewählten Aufbau `#key#value` hat und isoliert daraus die Variablen `$key` und `$value`.

Bei den Keys `aufstellung`, `austausch` und `teamchef` muss noch der Umstand gelöst werden, dass es bei den aktuellen Begegnungen immer zwei gibt: die der Heimmannschaft und die der Auswärtsmannschaft. Bei der Gastmannschaft wird ein `G` zum Feldnamen angehängt.

Diese Schleife füllt daher 716 Datensätze in die Datentabelle. Die Übertragung in die Datenbank erfolgt mit dem Kommando `$Adapter.Update($DataTable)`. Das genügt, denn wie die Tabelle in die Datenbank zu übertragen ist, wurde bereits in der Definition des Adapters genau festgelegt. Um diese konkreten Schreibvorgänge, die einfach eine schleifenartige Abarbeitung des `INSERT`-Kommandos ist, muss man sich nicht kümmern, das kann der Adapter ganz von allein.

```
#TextDateiLesen
#Connection
$DatabaseName = "s:\desktop\nationalmannschaft\österreich.accdb"
$ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=$DatabaseName;"
$Connection = New-Object System.Data.OleDb.OleDbConnection $ConnectionString
$Connection.Open()

#Tabelle Games
$SqlQuery = "SELECT * FROM Games"
$Command = New-Object System.Data.OleDb.OleDbCommand $SqlQuery,$Connection
$SqlInsert = "INSERT INTO Games ( Spiel, Begegnung, Ergebnis, Pause, Art, Details, Datum,
Schiedsrichter, Aufstellung, AufstellungG, Austausch, AustauschG, Teamchef, TeamchefG, Tore ) VALUES
(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
$CommandInsert = New-Object System.Data.OleDb.OleDbCommand $SqlInsert,$Connection
$Adapter = New-Object System.Data.OleDb.OleDbDataAdapter $Command
$Adapter.InsertCommand = $CommandInsert
$Parameter1 = New-Object System.Data.OleDb.OleDbParameter "@Spiel", Char, 255, "Spiel"
$Parameter2 = New-Object System.Data.OleDb.OleDbParameter "@Begegnung", Char, 255, "Begegnung"
$Parameter3 = New-Object System.Data.OleDb.OleDbParameter "@Ergebnis", Char, 255, "Ergebnis"
$Parameter4 = New-Object System.Data.OleDb.OleDbParameter "@Pause", Char, 255, "Pause"
$Parameter5 = New-Object System.Data.OleDb.OleDbParameter "@Art", Char, 255, "Art"
$Parameter6 = New-Object System.Data.OleDb.OleDbParameter "@Details", Char, 255, "Details"
$Parameter7 = New-Object System.Data.OleDb.OleDbParameter "@Datum", Char, 255, "Datum"
$Parameter8 = New-Object System.Data.OleDb.OleDbParameter "@Schiedsrichter", Char, 255,
"Schiedsrichter"
$Parameter9 = New-Object System.Data.OleDb.OleDbParameter "@Aufstellung", Char, 2550, "Aufstellung"
$Parameter10 = New-Object System.Data.OleDb.OleDbParameter "@AufstellungG", Char, 2550,
"AufstellungG"
$Parameter11 = New-Object System.Data.OleDb.OleDbParameter "@Austausch", Char, 255, "Austausch"
$Parameter12 = New-Object System.Data.OleDb.OleDbParameter "@AustauschG", Char, 255, "AustauschG"
$Parameter13 = New-Object System.Data.OleDb.OleDbParameter "@Teamchef", Char, 255, "Teamchef"
$Parameter14 = New-Object System.Data.OleDb.OleDbParameter "@TeamchefG", Char, 255, "TeamchefG"
$Parameter15 = New-Object System.Data.OleDb.OleDbParameter "@Tore", Char, 255, "Tore"
[void]$CommandInsert.Parameters.Add($Parameter1);
[void]$CommandInsert.Parameters.Add($Parameter2);
[void]$CommandInsert.Parameters.Add($Parameter3);
[void]$CommandInsert.Parameters.Add($Parameter4);
[void]$CommandInsert.Parameters.Add($Parameter5);
[void]$CommandInsert.Parameters.Add($Parameter6);
[void]$CommandInsert.Parameters.Add($Parameter7);
[void]$CommandInsert.Parameters.Add($Parameter8);
[void]$CommandInsert.Parameters.Add($Parameter9);
[void]$CommandInsert.Parameters.Add($Parameter10);
[void]$CommandInsert.Parameters.Add($Parameter11);
[void]$CommandInsert.Parameters.Add($Parameter12);
[void]$CommandInsert.Parameters.Add($Parameter13);
[void]$CommandInsert.Parameters.Add($Parameter14);
[void]$CommandInsert.Parameters.Add($Parameter15);
$DataTable = New-Object System.Data.DataTable
[void] $Adapter.Fill($DataTable)

$File_Alles = "S:\desktop\nationalmannschaft\alles.txt"
$Alles = Get-Content $File_Alles
$Game = 0
$KeyNames = @(
$DataRow = $DataTable.NewRow()
function ClearValues {
$DataRow["Spiel"]="" ; $DataRow["Begegnung"]="" ; $DataRow["Ergebnis"]="" ; $DataRow["Pause"]="" ;
$DataRow["Art"]="" ; $DataRow["Details"]="" ; $DataRow["Datum"]="" ; $DataRow["Schiedsrichter"]="" ;
$DataRow["Aufstellung"]="" ; $DataRow["AufstellungG"]="" ; $DataRow["Austausch"]="" ; $DataRow
["AustauschG"]="" ; $DataRow["Teamchef"]="" ; $DataRow["TeamchefG"]="" ; $DataRow["Tore"]=""
}
ClearValues

for($i=0; $i -lt $Alles.length; $i++) {
$Line = $Alles[$i];
#"Line:"+$Line
if ($Line.StartsWith("####")) {
$Game++

$DataTable.Rows.Add($DataRow)

$GameText = "Record:$Game "

$DataRow = $DataTable.NewRow()
ClearValues

# if ($Game -eq 5) { break }
continue
}
if ($Line.StartsWith("#Spiel#")) {
Write-Host $GameText"Spiel"($Line.Substring(7))
# if ($Line.Substring(7) -eq 1) { break }
}
$regex = [regex] '#(.+)(.*)'
$groups = $regex.Match($Line).Groups
$key = $groups[1].ToString().ToLower()
$value = $groups[2]

$value = $value.ToString().Trim(' ',',',';',';',';')

#Write-Host $key":"$value
switch ($key) {
"aufstellung" { if ($DataRow[$key] -eq "") { $DataRow[$key] = $value; }
else { $DataRow[$key+"G"] = $value; } }
"austausch" { if ($DataRow[$key] -eq "") { $DataRow[$key] = $value; }
else { $DataRow[$key+"G"] = $value; } }
"teamchef" { if ($DataRow[$key] -eq "") { $DataRow[$key] = $value; }
else { $DataRow[$key+"G"] = $value; } }
default { $DataRow[$key] = $DataRow[$key] + $value; }
}
}

$Adapter.Update($DataTable)
$Connection.Close()
```