

## SpielerTabelleErstellen

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

Im Abschnitt `#Tabelle Spiele` werden die beiden Felder benötigten Felder `Spiel` und `Aufstellung` in den Daten-Adapter `$AdapterSpiel` eingelesen.

In den zweiten Daten-Adapter `$AdapterSpieler` wird die Tabellenstruktur der Tabelle `Spieler` aufgebaut.

`$Team` aus der Tabelle `Spiel` enthält die gesamte Aufstellung als Text. Diese Textzeile wird in die einzelnen Spieler in der Array-Variablen `$TeamA` zerlegt.

Jeder dieser Einzelteile enthält ein bestimmtes Spielerformat. Bei neueren Aufstellungen wird der Verein nach dem Spielernamen durch einen Schrägstrich / angegeben. Die Rückennummer ist eingeklammert. Die zusätzliche Kennzeichnungen bei der Rückennummer: `G` steht für *Goalie* und `C` für *Captain*.

Bei der älteren Schreibweise ist der Verein eingeklammert und muss daher anders behandelt werden wie bei den neuere Spielen.

Spielernamen können einteilig sein (nur Familienname) oder zweiteilig (Vorname und Familienname). Leider ist diese Systematik nicht ganz durchgängig und manchmal stehen Vor- und Nachname in der falschen Reihenfolge. Da bleibt dann oft nichts anders übrig, als diese punktuellen Fehler händisch zu korrigieren.

Jedenfalls wird versucht, den Spielernamen in die Felder `FName` und `VName` richtig einzutragen und gleichzeitig auch die Rückennummer in das Feld `Nummer`.

In der Tabelle `Spieler` stehen die wesentlichen Felder: `Spiel`, `FName`, `VName` und `Verein`. Jeder Datensatz ist ein Einsatz eines Spielers. Es gibt bis zum Spiel 727 9180 Einsätze. Diese jetzt importierte Tabelle `Spiel` wird später `Spieler-Spiel` heißen, weil daraus noch die Teiltabelle `Spieler-Verein` und dann daraus die Tabellen `Spieler` und `Verein` entstehen.

## SpielerTauschTabelleErstellen

(Kode siehe nächste Seite) Leider ist der Spielertausch in einem anderen Format angegeben als die Aufstellung. Das Feld `Austausch` in der Tabelle `Spieler` wird daher zuerst in einer Hilfstabelle `Spieler_Spiel_Tausch` importiert. Da die Vorgänge dem Import der Spieler prinzipiell gleichen, wird dieser Vorgang nicht weiter beschrieben. Bei diesem Import werden die Bestandteile des Austausches (`SpielerEin`, `SpielerAus`, `VereinEin`, `VereinAus`, `Minute`) in einzelnen Felder abgelegt. Dann werden mit Mitteln von Access diese Datensätze dieser Tabelle `Spieler_Spiel_Tausch` mit der Anfüge-Abfrage `SpielerTausch Neue Spieler` an die Tabelle `Spieler-Spiel` angehängt und das Feld von Minute auf den richtigen Zeitpunkt eingestellt. Die `bisMinute` ist 90. Danach wird bei jenen Spielern, die das Feld verlassen, die `bisMinute` auf denselben Wert eingestellt.

## Anhang

Die beiden Programme `SpielerTauschTabelleErstellen` und `ToreTabelleErstellen` sind in einem Anhang der PDF-Version dieser Ausgabe auf den Seiten 33 und 34 enthalten.

```
#SpielerTabelleErstellen
#Datenbank
$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 Spiele
$SqlSpielQuery = "SELECT Spiel, Aufstellung FROM Spiele ORDER BY SPIEL"
$CommandSpiel = New-Object System.Data.OleDb.OleDbCommand $SqlSpielQuery,$Connection
$AdapterSpiel = New-Object System.Data.OleDb.OleDbDataAdapter $CommandSpiel

$DataTableSpiel = New-Object System.Data.DataTable
[void] $AdapterSpiel.Fill($DataTableSpiel)
$DataTableSpiel.Length = $DataTableSpiel.Rows.Count

#Tabelle Spieler
$SqlSpielerQuery = "SELECT * FROM Spieler"
$SqlSpielerInsert = "INSERT INTO Spieler (Spiel, FName, VName, Verein, Spielposition, Captain, Lfd, Nummer) VALUES
(@Spiel, @FName, @VName, @Verein, @Spielposition, @Captain, @Lfd, @Nummer)";
$CommandSpieler = New-Object System.Data.OleDb.OleDbCommand $SqlSpielerQuery,$Connection
$AdapterSpieler = New-Object System.Data.OleDb.OleDbDataAdapter $CommandSpieler
$AdapterSpieler.ContinueUpdateOnError = $false
$CommandSpielerInsert = New-Object System.Data.OleDb.OleDbCommand $SqlSpielerInsert,$Connection
$AdapterSpieler.InsertCommand = $CommandSpielerInsert
$Parameter1 = New-Object System.Data.OleDb.OleDbParameter "@Spiel", Integer, 4, "Spiel"
$Parameter2 = New-Object System.Data.OleDb.OleDbParameter "@FName", Char, 255, "FName"
$Parameter3 = New-Object System.Data.OleDb.OleDbParameter "@VName", Char, 255, "VName"
$Parameter4 = New-Object System.Data.OleDb.OleDbParameter "@Verein", Char, 255, "Verein"
$Parameter5 = New-Object System.Data.OleDb.OleDbParameter "@Spielposition", Char, 255, "Spielposition"
$Parameter6 = New-Object System.Data.OleDb.OleDbParameter "@Captain", Boolean, 4, "Captain"
$Parameter7 = New-Object System.Data.OleDb.OleDbParameter "@Lfd", Integer, 4, "Lfd"
$Parameter8 = New-Object System.Data.OleDb.OleDbParameter "@Nummer", Integer, 4, "Nummer"
[void] $CommandSpielerInsert.Parameters.Add($Parameter1)
[void] $CommandSpielerInsert.Parameters.Add($Parameter2)
[void] $CommandSpielerInsert.Parameters.Add($Parameter3)
[void] $CommandSpielerInsert.Parameters.Add($Parameter4)
[void] $CommandSpielerInsert.Parameters.Add($Parameter5)
[void] $CommandSpielerInsert.Parameters.Add($Parameter6)
[void] $CommandSpielerInsert.Parameters.Add($Parameter7)
[void] $CommandSpielerInsert.Parameters.Add($Parameter8)

$DTSpieler = New-Object System.Data.DataTable
[void] $AdapterSpieler.Fill($DTSpieler)

for ($i=0; $i -lt $DataTableSpiel.Length; $i++) {
    $Spiel = $DataTableSpiel.Rows[$i]["Spiel"].ToString()
    $Team = $DataTableSpiel.Rows[$i]["Aufstellung"].ToString()
    $Spiel += " " + $Team
    $Split_Spieler = ' , ; '
    $Trim_Verein_Old = ' ( ) '
    $TeamA = $Team.ToString().Split($Split_Spieler)
    #TeamA
    for ($j=0; $j -lt $TeamA.Length; $j++) {
        $TeamA[$j] = $TeamA[$j].Trim()
    }
    for ($j=0; $j -lt $TeamA.Length; $j++)
    {
        $SpielerNummer = 0
        $Spieler = ""
        $Verein = ""
        $Spielposition = ""
        $Captain = $false

        #if ($TeamA.Length -eq 0) {} else { $Spieler }
        if ($Team.Contains("/") { # neuere Aufstellung mit Rückennummern
            $Spieler = ($TeamA[$j].Split('/'))[0].Trim(' ')
            if ($TeamA[$j].Split('/').Length -eq 2 {
                $Verein = ($TeamA[$j].Split('/'))[1].Trim(' ')
            }
        }
        if ($TeamA[$j].Contains("(")) { # Rückennummer ist angegeben
            $SpielerNummer = $Spieler.Substring(1,$Spieler.IndexOf(')-1).Trim()
            if ($SpielerNummer.Contains("C")) { $SpielerNummer = $SpielerNummer.Replace("C",""); $Captain=$true }
            if ($SpielerNummer.Contains("GK"))
                { $SpielerNummer = $SpielerNummer.Replace("GK",""); $Spielposition="GK" }
            $Spieler = $Spieler.Substring($Spieler.IndexOf('')+1).Trim()
        }
        } else { # alte Aufstellung
            if ($TeamA[$j].Contains("(")) { # Verein ist angegeben
                $Spieler = ($TeamA[$j].Split('(')) [0].Trim(' ')
                $Verein = ($TeamA[$j].Split('(')) [1].Trim($Trim_Verein_Old)
            } else { # Verein fehlt
                $Spieler = $TeamA[$j].Trim(' ')
            }
        }
        if ($j -eq 0) { $Spielposition="G" }
        $Out = $Spiel + " " + $Spieler + " -> " + $Verein + " -> " + $SpielerNummer
        if ($Captain -eq "true") {$Out += " C"}
        if ($Spielposition -ne "") {$Out += " " + $Spielposition}
        #Out

        $DataRow = $DTSpieler.NewRow()
        $DataRow["Spiel"] = $Spiel
        $SpielerTeilname = $Spieler.Split(' ')
        switch ($SpielerTeilname.Count)
        {
            2 { if ($SpielerTeilname[1].Length -lt 3) {
                $DataRow["FName"] = $SpielerTeilname[0]
                $DataRow["VName"] = $SpielerTeilname[1]
            } else {
                $DataRow["FName"] = $SpielerTeilname[1]
                $DataRow["VName"] = $SpielerTeilname[0]
            }
        }
        default {
            $DataRow["FName"] = $Spieler
            $DataRow["VName"] = ""
        }
    }
    $DataRow["Verein"] = $Verein
    $DataRow["Spielposition"] = $Spielposition
    $DataRow["Captain"] = $Captain
    $DataRow["Lfd"] = $j + 1
    $DataRow["Nummer"] = [int]$SpielerNummer

    $DTSpieler.Rows.Add($DataRow)
}

$AdapterSpieler.Update($DTSpieler)
$Connection.Close()
```