

# SQL

- [Daten per PowerShell in eine Microsoft SQL Datenbank importieren](#)

# Daten per PowerShell in eine Microsoft SQL Datenbank importieren

## Einleitung

In diesem Beitrag geht es kurz darum, wie wir mit einem **PowerShell Skript**, Daten in eine **Microsoft SQL Datenbank** importieren können. Dies verwenden wir, wenn wir z.B. Daten aus einer Abfrage heraus, in eine Datenbank importieren möchten, um aus dieser z.B. Auswertungen zu fahren.

## Durchführung

Um die Daten nun in eine Datenbank zu importieren, brauchen wir natürlich ein entsprechendes Skript. Für diesen Vorgang habe ich eine Funktion geschrieben, welche wir in vorhandene Skripte einfach einbauen können. Je nachdem, ob die **Windows NT Authentifizierung** oder die **SQL-Server Authentifizierung** gewählt wird, muss die entsprechende Funktion verwendet werden:

## Windows NT Authentifizierung

```
function Run-SQL {  
    Param(  
        [string]$sqlServer,  
        [string]$sqlDatabase,  
        [string]$sqlQuery  
    )  
  
    $connectionParameter = "Data Source=$sqlServer;Integrated Security=SSPI;Initial Catalog=$sqlDatabase"  
    $sqlConnection = New-Object System.Data.SqlClient.SqlConnection($connectionParameter)  
    $sqlCommand = New-Object System.Data.SqlClient.SqlCommand($sqlQuery, $sqlConnection)  
    $sqlConnection.Open()  
    $sqlDataAdapter = New-Object System.Data.SqlClient.SqlDataAdapter $sqlCommand  
    $dataSet = New-Object System.Data.DataSet  
    $sqlDataAdapter.Fill($dataSet) | Out-Null  
    $sqlConnection.Close()
```

```
$dataSet.Tables  
}
```

## SQL-Server Authentifizierung

```
function Run-SQL {  
    Param(  
        [string]$sqlServer,  
        [string]$sqlDatabase,  
        [string]$sqlQuery,  
        [string]$sqlUser,  
        [string]$sqlPassword  
    )  
  
    $connectionParameter = "Data Source=$sqlServer;UserID=$sqlUser;Password=$sqlPassword;Initial  
Catalog=$sqlDatabase"  
    $sqlConnection = New-Object System.Data.SqlClient.SqlConnection($connectionParameter)  
    $sqlCommand = New-Object System.Data.SqlClient.SqlCommand($sqlQuery, $sqlConnection)  
    $sqlConnection.Open()  
    $sqlDataAdapter = New-Object System.Data.SqlClient.SqlDataAdapter $sqlCommand  
    $dataSet = New-Object System.Data.DataSet  
    $sqlDataAdapter.Fill($dataSet) | Out-Null  
    $sqlConnection.Close()  
    $dataSet.Tables  
}
```

Jetzt kann in dem PowerShell Skript, mit Aufruf der Funktion und den entsprechenden Parametern, eine Abfrage an den **SQL-Server** getätigt werden. Ein Aufruf der Funktion könnte dann z.B. so aussehen:

```
Run-SQL -sqlHost "<IP SQL-Server>" -sqlDatabase "<Datenbank Name>" -sqlQuery "INSERT INTO  
<Tabellenname> (<Spalten Name>) VALUES (<WERTE>);"
```

Sobald der SQL-Befehl auf die eigenen Bedürfnisse angepasst ist, und der Benutzer Zugriff auf die Tabelle hat, werden Daten erfolgreich in die Datenbank geschrieben.