ORDIX AG

Datenübertragung von SQL Server und MySQL/MariaDB nach Oracle mit PowerShell

Andreas Jordan

DOAG Konferenz + Ausstellung
Raum Helsinki, 14:00 Uhr
21.11.2023, Nürnberg

Andreas Jordan

ORDIX AG

- Principal Consultant und Teamleiter

- Microsoft SQL Server
- Oracle Database
- PowerShell

- Sprecher auf den IT-Tagen 2015, 2022, 2023
- Sprecher auf der DOAG Konferenz & Ausstellung 2022, 2023

- https://blog.ordix.de/andreas-jordan

Agenda

ORDIX AG

- Warum PowerShell?
- Welche Voraussetzungen müssen erfüllt sein?
- Drei Arten der Datenübertragung
- Demo
- Transfer aus der Datenbank heraus starten

Warum PowerShell?

ORDIX AG

- PowerShell 5.1 und die PowerShell ISE sind auf jedem Windows System verfügbar.
  - Übertragungen zwischen SQL Server und Oracle sind damit auf jeden Fall möglich.
- PowerShell 7 und Visual Studio Code können auf Windows und Linux einfach installiert werden.
  - Wird für Übertragungen von MySQL oder MariaDB aus empfohlen.
- Durch die Nutzung von .NET ist es eine vollwertige objektorientierte Programmiersprache.
  - Schleifen
  - Fehlerbehandlung
  - Logging
- Ich kann meine Skripte Zeile für Zeile ausführen.

Welche Voraussetzungen müssen erfüllt sein?

ORDIX AG

- PowerShell basiert auf .NET und kann daher für .NET bereitgestellte Bibliotheken verwenden.
- Viele Datenbankhersteller, aber auch die Community, stellen passende Bibliotheken bereit.
- Die Bibliotheken sind in den Clients der Hersteller oder in NuGet-Paketen enthalten.

Datenbanksystem	PowerShell 5.1	PowerShell 7
MS SQL Server	Alles bereits in .NET enthalten	Alles bereits in .NET enthalten
Oracle	Oracle.ManagedDataAccess.dll NuGet: Oracle.ManagedDataAccess Client: ODP.NET\managed\common	Oracle.ManagedDataAccess.dll NuGet: Oracle.ManagedDataAccess.Core
MySQL / MariaDB	devart.com: dotConnect for MySQL	MySqlConnector.dll NuGet: MySqlConnector

Wie wird die DLL in PowerShell verwendet?

ORDIX AG

```
Add-Type -Path '…\Oracle.ManagedDataAccess.dll'
```

- Diese eine Zeile fügt die in der DLL enthaltenen .NET-Klassen zur PowerShell-Sitzung hinzu.

- Ein paar beispielhafte Klassen:
  - Oracle.ManagedDataAccess.Client.OracleConnection
  - Oracle.ManagedDataAccess.Client.OracleCommand
  - Oracle.ManagedDataAccess.Client.OracleDataAdapter

Eigene Befehle als Wrapper für die .NET-Klassen

ORDIX AG

- Direkter Zugriff auf die .NET-Klassen ist möglich, der Code wird aber unübersichtlich.
- Eigene Befehle bilden eine Schnittstelle zwischen PowerShell-Code und .NET-Klassen.
- Ich verwende:
  - Connect-SqlInstance / Connect-OraInstance / Connect-MyInstance
  - Invoke-SqlQuery / Invoke-OraQuery / Invoke-MyQuery
  - Get-TableInformation
  - Get-TableReader
  - Write-Table
- Den Quellcode in meinem GitHub Repository können Sie gerne als Basis verwenden.

Drei Arten der Datenübertragung

ORDIX AG

- Invoke-Query
  Invoke-Query

- Invoke-Query
  Write-Table

- Get-TableInformation
  Get-TableReader
  Write-Table

Invoke-Query / Invoke-Query

ORDIX AG

```
$data = Invoke-SqlQuery -Connection $sqlConnection -Query "SELECT * FROM Badges"
Invoke-OraQuery -Connection $oracleConnection -Query "TRUNCATE TABLE badges"
foreach ($row in $data) {
    $invokeQueryParams = @{
        Connection      = $oracleConnection
        Query           = "INSERT INTO badges VALUES (:Id, :Name, :UserId, :CreationDate)"
        ParameterValues = @{
            Id           = $row.Id
            Name         = $row.Name
            UserId       = $row.UserId
            CreationDate = $row.CreationDate
        }
    }
    Invoke-OraQuery @invokeQueryParams
}
```

Invoke-Query / Write-Table

ORDIX AG

```
$data = Invoke-SqlQuery -Connection $sqlConnection -Query "SELECT * FROM Badges"

Write-OraTable -Connection $oracleConnection -Table badges -Data $data -TruncateTable
```

Get-TableInformation + Get-TableReader / Write-Table

ORDIX AG

```
$tableInformation = Get-SqlTableInformation -Connection $sqlConnection -Table Badges
$dataReader = Get-SqlTableReader -Connection $sqlConnection -Table Badges

$invokeWriteTableParams = @{
    Connection          = $oracleConnection
    Table               = 'badges'
    DataReader          = $dataReader
    DataReaderRowCount = $tableInformation.Rows
    TruncateTable       = $true
}
Write-OraTable @invokeWriteTableParams
```

Demo

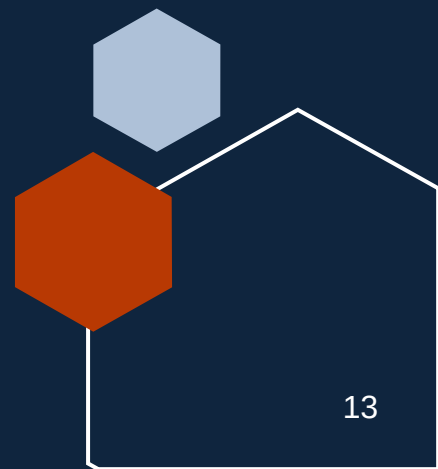
ORDIX AG

Schauen wir uns das doch mal live an…

Transfer aus der Datenbank heraus starten

ORDIX AG

- SQL Server: Der SQL Server Agent kann PowerShell-Skripte ausführen.
- Oracle: Mit DBMS_SCHEDULER.CREATE_JOB lassen sich auch PowerShell-Skripte nutzen.
- Problem hierbei: Wie werden die Zugangsdaten für die Zieldatenbanken verwaltet?



Lernen Sie doch PowerShell…

ORDIX AG

Vielen Dank für Ihre Aufmerksamkeit

https://github.com/andreasjordan/PowerShell-for-DBAs/tree/main/DOAG2023

ORDIX AG

Aktiengesellschaft für Softwareentwicklung, Schulung, Beratung und Systemintegration

Zentrale Paderborn
Karl-Schurz-Straße 19a
33100 Paderborn
Tel.: 05251 1063-0
Fax: 0180 1 67349 0

Seminarzentrum Wiesbaden
Kreuzberger Ring 13
65205 Wiesbaden
Tel.: 0611 77840-00

info@ordix.de
https://www.ordix.de/