

Migration von Fremddatenbanken mit dem SQL Developer 3.0

KlausWessolowski / Detlef Kassow
Oracle Deutschland B.V. & Co. KG

Schlüsselworte:

Migration, Fremddatenbank, SQL Developer, SQL Server, Sybase, DB2, Teradata, Access

Einleitung

Im Rahmen der Maßnahmen zur Kostensenkung, Steigerung der Datenqualität oder Einführung eines zentralen Datenmodells, steht in vielen Unternehmen eine Konsolidierung von Datenbanken verschiedener Anbieter im Unternehmen an. Oracle bietet mit dem SQL Developer ein kostenfreies Werkzeug zur einfachen Übernahme von Daten aus ‚non-Oracle‘ Datenbanken. Das mag dazu verleiten, die Daten aus einem anderen System mal eben so nebenbei zu migrieren. Dabei wird oft übersehen, dass meistens nicht nur Daten sondern auch Logik in Form von Triggern und Prozeduren in Datenbanken enthalten sind. Diese werden wiederum von Anwendungen genutzt, die bei einem Wechsel der Datenbank mehr oder weniger angepasst werden müssen. Der Aufwand für eine Datenbankmigration wird daher oft unterschätzt. Egal ob es sich um die Migration eines Microsoft Access-Systems oder eines komplexen Hostsystems handelt, es gelten immer die Regeln eines IT-Projektes mit genauen Abschätzungen, Testszenarien und festgelegten Zeitabläufen.

Der Oracle SQL Developer

Der SQL Developer ist **das** kostenlose Entwicklungswerkzeug für Oracle Entwickler. Er ist mit einer graphischen Oberfläche ausgestattet und beinhaltet viele Features für die Entwicklung von SQL- und PL/SQL-Programmen. Schon die Grundinstallation einer Oracle 10g/11g Datenbank beinhaltet dieses Werkzeug, doch man sollte unbedingt immer die neueste Version¹ herunterladen, da das Produkt ständig erweitert wird. Die zurzeit aktuelle Version ist 3.0.

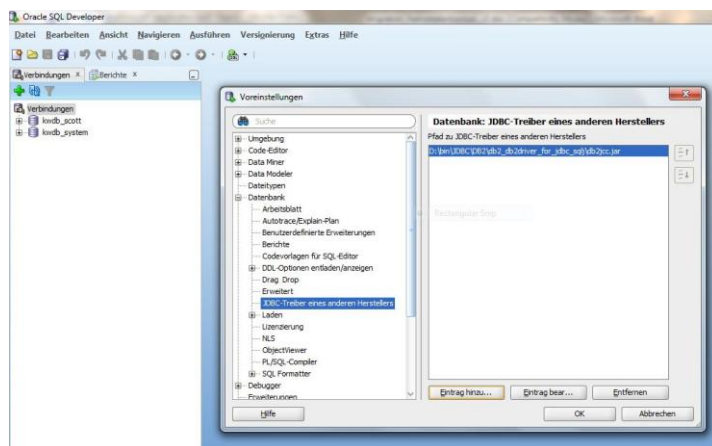


Abb.1: JDBC Treiber einbinden

Mit dem SQL Developer kann aber nicht nur gegen Oracle Datenbanken, sondern auch gegen die folgenden Datenbanken entwickelt werden:

- Microsoft SQL Server,
- Sybase Adaptive Server,
- Mysql und
- IBM DB2.

Man benötigt dafür allerdings einen JDBC-Treiber des entsprechenden Anbieters.

Für Microsoft SQL Server oder Sybase Adaptive Server benötigt man den gleichen JDBC-Treiber, er kann hier² heruntergeladen werden.

¹ <http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

² <http://jtds.sourceforge.net/>

Unter „Extras“ →“Voreinstellungen“ →“Datenbank“ →“JDBC-Treiber eines anderen Herstellers“ kann der gewünschte Treiber eingebunden werden. Abbildung 1 stellt dies dar.

Zurzeit lassen sich mit dem SQL Developer Microsoft Access (man benötigt allerdings eine englische Access Version), Microsoft SQL Server, Sybase Adaptive Server, Mysql, Teradata und IBM DB2 Datenbanken migrieren. Es werden die Tabellen, Views, Daten, Benutzer und programmierten Objekte (Stored Procedures/Functions und Trigger) migriert. Bei den programmierten Objekten kann man bei Sybase und Microsoft mehr als 90% der Funktionalität konvertieren.

Migratoinsanalyse

Um abschätzen zu können, wie groß der Aufwand einer Migration ist, wird oft versucht, mit einer Teilmenge der Daten oder Sourcen („repräsentativer Querschnitt“) hoch zu rechnen. Leider muss man in der Praxis häufig feststellen, dass dann bei der praktischen Umsetzung Programm- oder Datenbankstrukturen auftauchen, die vorher überhaupt nicht bekannt waren. Das Ergebnis ist eine neue Ermittlung der tatsächlichen Aufwände, was in der Regel zur Verschiebung des geplanten Projektablaufes und zu zusätzlichen Kosten führt.

Der SQL Developer bietet Möglichkeiten eine Abschätzung über Umfang, Dauer und Kosten zu ermitteln. Hat man direkten Zugriff auf das Fremdsystem, kann man mit dem SQL Developer direkt eine Verbindung zu der Datenbank herstellen. Hat man keinen Zugriff, kann der SQL Developer scripte erstellen, die dann auf dem Quellsystem laufen und die nötige Kataloginformation extrahieren. Diese kann dann in den SQL Developer mittels ‚Offline capture‘ importiert werden. Nun kann ein Report erzeugt werden, der einen ersten Überblick über Machbarkeit und Aufwand bietet.

Verbindung zur Datenbank eines anderen Herstellers

Der nächste Schritt ist das Anlegen einer Verbindung zum Fremdsystem. Der einfachste Weg ist ein Klick auf das große grüne Kreuz auf der oberen linken Seite des SQL Developers.

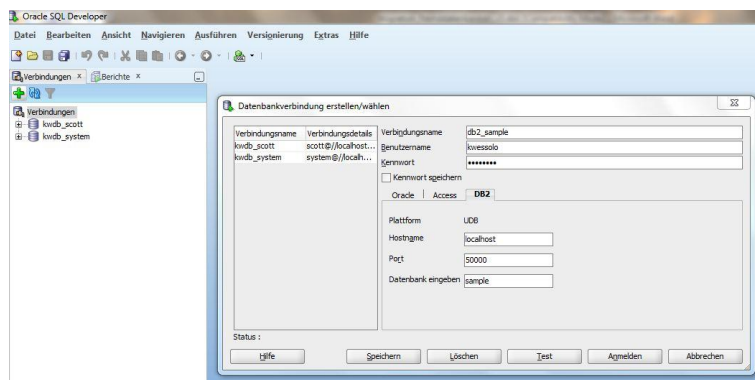


Abb. 2 SQL Developer Aufbau der Verbindung

Danach erscheint der in Abb. 2 dargestellte Bildschirm. Wenn der entsprechende JDBC Treiber wie oben beschrieben geladen ist, erhält man auch einen weiteren Tabulator für die Verbindung zu diesem System.

Mit „Anmelden“ wird dann eine neue Verbindung angelegt, die auch auf der linken Seite erscheint. Nun kann man beispielsweise über die Navigationsleiste Objekte des Fremdsystems ansehen.

Somit hat man auch die Möglichkeit, einen Select-Befehl gegen die Oracle Datenbank und die Fremddatenbank absetzen und die Ergebnisse zu vergleichen.

Anlegen des Migrationsrepositorys

Um eine Migration durchführen zu können, benötigt man ein Oracle Schema für das Migrations- Repository. In diesem Repository befinden sich die Daten über die Migrationsstrukturen. SQL Developer lädt die Metadaten der Fremddatenbank in das Repository. Der Besitzer des Repository muss die Rechte für das Anlegen von Tabellen, Views und Stored Procedures haben. In diesem Beispiel wurde das Schema „MIGRATIONS“ angelegt. Klickt man nun mit der rechten Maustaste auf das Schema kann man es als Migrations Repository

verknüpfen, existiert noch kein Repository in dem Schema wird es erstellt. Hier kann es auch geleert (mit truncate) oder ganz gelöscht werden.

Die Migration

Der Migrations-Assistent (Abb. 3) ist in der neuen Version verbessert worden, er ist jetzt wesentlich flexibler und kann z.B. an jeder Stelle beendet werden. Er wird durch „Extras“ → „Migrieren“ → „Migrieren“ aufgerufen. Wurde vorher kein Repository angelegt, kann man das auch hier tun, wurde eins „verknüpft“, wird dieses genutzt.

Im nächsten Step werden jetzt ein oder mehrere Datenbanken (bei Oracle werden dies meistens Schemata) einer Fremddatenbank (Instanz) eines anderen Herstellers analysiert. Dabei wird die Katalog-Information ausgelesen und in das Repository geladen. Hat man eine Verbindung zur Quelldatenbank, dann kann das ‚online‘ geschehen, hat man die Daten als ‚Offline Capture‘ vorliegen, kann man aus diesen Dateien die Information einlesen.

Zum Erstellen der offline capture files kann SQL Developer unter „Extras“ → „Migrieren“ → „Datenbank-Capture-Scripte-erstellen“ scripte erzeugen, die gegen die Fremddatenbank laufen müssen.

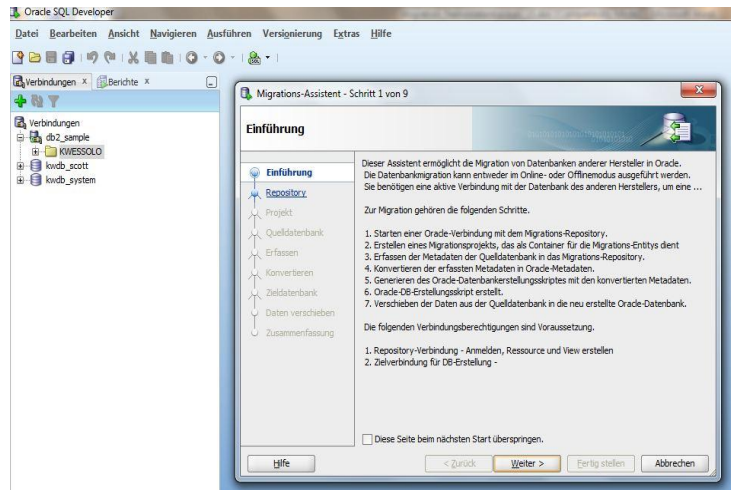


Abb.3: Migrations Assistent

Wenn der Captureprozess erfolgreich beendet wurde, kann man mit „Ansicht“ → „Migrationsprojekte“ ein Fenster einblenden (Abbildung 4 links unten), in dem man durch die eingelesenen Schemata navigieren kann und die Reports, grafische als auch tabellarische, online ansehen kann.

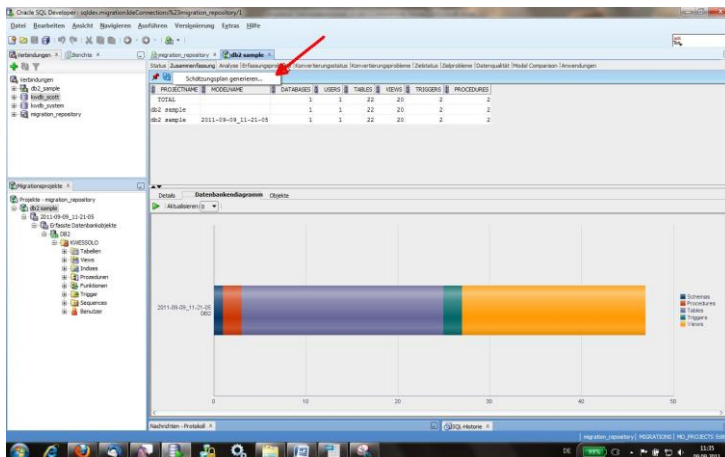


Abb. 4: Ausschnitt mit den Erfassungsfenstern

Man hat jetzt die Möglichkeit, sich die einzelnen erfassten Objekte anzusehen und eventuell auch schon Anpassungen vorzunehmen oder Objekte zu löschen, die nicht migriert werden sollen. Es ist auch möglich, mehrere Modelle in einem Repository zu speichern.

Außerdem kann eine Excel-Datei mit den Ergebnissen erstellt werden. Dazu muss die etwas unauffällige Fläche „Aktionen“ angeklickt werden (Abb.4, roter Pfeil) oder mit einem Klick der rechten Maustaste auf das Migrationsprojekt.

Konvertierung des Modells

Durch anklicken (rechte Maustaste) eines erfassten Modells kann die Konvertierung in das Oracle Modell durchgeführt werden. Der Assistent erscheint und führt den Dialog an der entsprechenden Stelle fort. Bevor der Konvertierungslauf durchgeführt wird, hat man die Möglichkeit, das Mapping der Datentypen auf die Oracle

Datenbank festzulegen. Das Protokoll der Konvertierung erscheint im Nachrichtenfenster. Es besteht jetzt die Möglichkeit, sich jedes Element anzusehen und eventuell Änderungen vorzunehmen.

Nur um Missverständnissen vorzubeugen: Es ist noch nichts migriert worden. Die Strukturen liegen bis jetzt nur im Repository, und es sind noch keinerlei Objekte in der Oracle Datenbank angelegt worden. Erst durch erneutes (rechts) klicken auf ein konvertiertes Modell kann die Generierung angestoßen werden. Für die Migration mehrerer Datenbanken (Schemata) muss der Migrations-User erweiterte Rechte haben (create user etc.)

Generierung des Oracle Modells

Auch bei der Generierung hat man die Wahl zwischen „online“ und „offline Generierung“. Online generiert die Datenbank Objekte sofort in eine Oracle Datenbank.

Der vorsichtige Migrator wird sich meistens für die offline Variante entscheiden und das generierte script neugierig inspizieren. Er kann es nach eigenen Bedürfnissen anpassen (z.B. Schema- und Tablespace-Namen) und in einer Textdatei speichern.

Außerdem kann man auch einzelne Objekte auswählen und in der Datenbank erzeugen lassen. Große Stored Procedures werden in aller Regel so migriert und getestet.

Wenn man das generierte Script ausführlich studiert, wird man über ein PL/SQL-Package mit dem Namen [db]_utilities³ stolpern. Da die String-, Datums-, Zeit- und mathematischen Funktionen im jeweiligen SQL-Dialekt der Hersteller mit anderen Namen belegt sind, wird dieses Package mitgeliefert. Es bildet die entsprechenden Funktionen mit Oracle PL/SQL nach. Sie verhalten sich genau so wie die Funktionen der Fremddatenbank, und man kann sie auch stand-alone generieren. D.h. man nimmt einfach die Package-Definitionen heraus. So können Frontend-Applikationen, die via SQL auf die Datenbank zugreifen und diese Funktionen benutzen, quasi unverändert bleiben (es müssen nur die Oracle-Treiber-Definitionen im Programm geändert werden).

Datenmigration

Nachdem die Strukturen erfolgreich migriert wurden, können nun auch die Daten migriert werden. Der SQL Developer bietet zwei Optionen an: online über JDBC oder offline über das Erzeugen von Entlade- und Ladeskripten. Man bedient sich bei der zweiten Option im Falle von Microsoft und Sybase Datenbanken des Bulkcopy-Tools in Kombination mit dem Oracle SQL*Loader. Die erste Option eignet sich für kleine Datenmengen, z.B. wenn nur Testdaten migriert werden sollen. Die zweite Option wird in aller Regel benutzt, um die produktiven Daten zu migrieren.

Für eine Online-Migration muss eine Verbindung zum Zielschema in der Oracle Datenbank eingerichtet sein. Jetzt mit der rechten Maustaste das konvertierte Modell anklicken und die Option „Daten verschieben“ wählen. Der Assistent wird eingeblendet und man muss nur noch wählen ob man offline oder online die Daten verschieben will sowie die Ursprungs- und Zielverbindung auswählen. In einem Fenster wird dann der Status und die Anzahl der Fehler beim Transport angezeigt.

Hat der Migrator sich für die Offline-Migration entschieden, werden die Skripte für Unload und Load in das Ausgabeverzeichnis generiert. Man muss nur noch den Namen des Servers und des Benutzers mit Password editieren und kann dann mit dem Entladen beginnen. In oracle_ctl.bat findet man neben den Aufrufen des SQL*Loaders auch zwei Aufrufe von SQL-Skripten. Diese werden benötigt, um vor dem Laden die Constraints und Trigger abzuschalten und nach dem Ladelauf wieder zu aktivieren. Um Probleme bei der Zeichenkonvertierung zu vermeiden, sollte man noch die entsprechenden NLS-Settings für Zeichensatz und Datumsformat anpassen.

³ [db] ist der Prefix für das source system, z.B. „DB2_utilities“.

Migration Scratch Editor

Sehr häufig kommt es vor, dass man nur einzelne Objekte wie eine T-SQL-Stored Procedure nach PL/SQL migrieren will. Hier bietet der SQL Developer den sogenannte Migration Scratch-Editor. Unter „Extras“ → „Migrieren“ → „Scratch-Editor für Übersetzung“ wird er aktiviert. Es erscheinen zwei neue Fenster im SQL-Developer, links für den Quellcode der Fremddatenbank und rechts die dazugehörige Ausgabe in PL/SQL. Man kopiert den gewünschten Quellcode in das linke Fenster und klickt auf das Symbol für das Übersetzen (>>) und erhält die eine Ausgabe wie in Abbildung. 5.

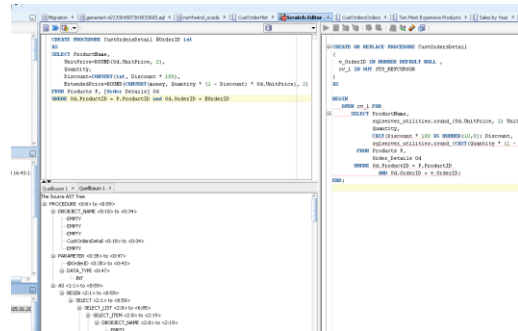


Abb. 5: Migration Scratch-Editor

Das Fenster unter dem Original zeigt den sogenannten AST Tree an und ist nur als Information für die Generierung gedacht. Auf der linken Seite erscheint das erzeugte PL/SQL-Equivalent. Dieses kann dann in eine Skript-Datei kopiert werden oder direkt über eine Verbindung in Oracle angelegt werden. Aber der Scratch-Editor kann noch viel mehr! Klickt man auf das Symbol „Diff Editor für Übersetzung“, erhält man eine direkte Anzeige, welcher Befehl in T-SQL mit welchem in Oracle PL/SQL korrespondiert. Gerade bei großen Objekten ist das eine wichtige Hilfe.

Testen, Testen ...

Schon während der Konvertierung gehören ausgedehnte Tests zur Migration. Der Testaufwand beträgt in aller Regel mehr als 50% der gesamten Projektlaufzeit. Da die Migrationstools immer weiter verbessert werden, wird dieser Anteil in Zukunft eher weiter ansteigen. Zu einem guten Pflichtenheft gehören immer Testszenerien und Informationen dazu, von wem die Tests durchgeführt werden sollen. Durch gute Tests können Tuning-Maßnahmen, die sonst nur während des späteren Betriebes durchgeführt werden können, schon vorher in Angriff genommen werden.

Die Inbetriebnahme

Vor der Inbetriebnahme des Systems muss unbedingt eine Fallback-Lösung vorhanden sein. Wenn beim Start des neuen Systems Fehler auftreten, die den weiteren Betrieb verhindern, muss auf das alte System zurückgegriffen werden können. Weitere Tuning-Schritte können sich beispielsweise ergeben, da für andere Datenbanken optimierte Abfragen unter Oracle einen anderen Ablaufplan haben und deshalb an Oracle angepasst werden müssen.

Online Quellen

Software:

<http://www.oracle.com/global/de/community/platform/index.html>

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

<http://jtds.sourceforge.net/>

Kontaktadresse:

Detlef Kassow
Oracle Deutschland B.V. & Co. Kg.
Kühnehöfe 5
D-22761 Hamburg

Telefon: +49 (0) 40-89091184
Fax: +49 (0) 40-89091250
E-Mail: detlef.kassow@oracle.com

Klaus Wessolowski
Oracle Deutschland B.V. & Co. Kg.
Hambornerstr. 51
D-40472 Düsseldorf

Telefon: +49 (0) 211-74839-324
Fax: +49 (0) 211-74839-15
E-Mail: klaus.wessolowski@oracle.com