

Von Datenbank zu Datenbank

Um ein IT-System auf ein anderes zu überführen, muss in der Regel eine Migration durchgeführt werden. Der Weg zu Oracle wird durch Hilfsprogramme für die Portierung vereinfacht. **Detlef Kassow**

Auf einen Blick

Konsolidierung und Modernisierung sind Schlagwörter, die auch aufgrund des Zwanges, Kosten einzusparen, im Mittelpunkt stehen. Dieser Artikel soll Administratoren einen Überblick verschaffen, wie eine Migration zur Oracle-Datenbank in der Praxis vereinfacht durchgeführt werden kann.

■ Plattform

Oracle 11g/11g R2

■ Technik/Anwendung

Datenbank-Administration

■ Voraussetzungen

Oracle Datenbank 11g oder höher

■ Autor

Detlef Kassow ist seit 1996 bei Oracle Deutschland beschäftigt. Schwerpunkte seiner Tätigkeit sind die Beratung von Kunden und Partnern bei der Migration von Datenbank- und Programmsystemen nach Oracle sowie die Mitarbeit bei der Entwicklung von Werkzeugen zur Migration.

Oracle entwickelt seit mehr als 15 Jahren Tools für die Migration von anderen Datenbanksystemen nach Oracle. Der primäre Fokus liegt in diesem Artikel auf der Konvertierung der Datenbank, da Ausführungen zur Frontend-Migration den Rahmen sprengen würden. Doch bevor Sie sich selbst mit der Praxis beschäftigen, zunächst etwas Theorie über die Durchführung von Migrationsprojekten. Der Aufwand für eine Datenbankmigration wird häufig unterschätzt, deshalb scheitern viele Migrationsvorhaben. Ganz gleich 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. Immer wieder wird gefordert, die Migration quasi „nebenbei“ zu den täglich anfallenden Arbeiten durchzuführen. Dieser Ansatz führt dann aber in aller Regel dazu, dass die Migration niemals zu Ende gebracht wird.

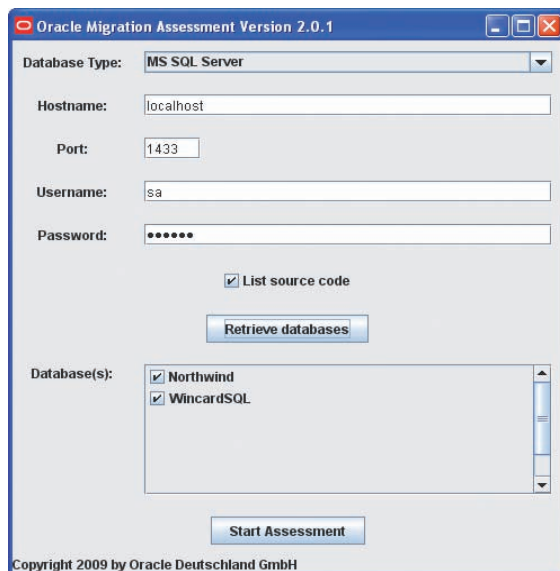
Schritt 1: Die Analyse

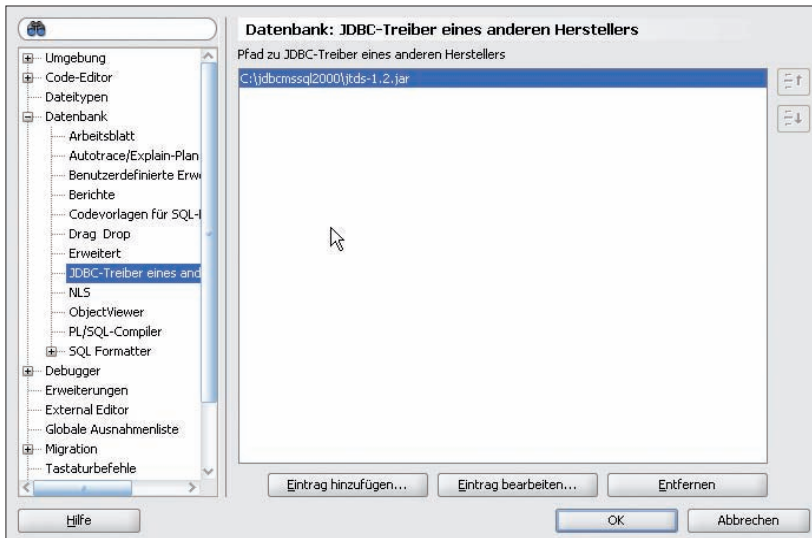
Um abzuschätzen, wie groß der Aufwand einer Migration ist, benötigen Sie zur Analyse sämtliche (!) Quellen und Zugriff auf alle relevanten Datenbanksysteme. Immer wieder wird versucht, eine Abschätzung durchzuführen, indem nur ein Teil der Programme zur Verfügung gestellt wird (repräsentativer Querschnitt), um dann das ermittelte Ergebnis mit einem Faktor zu multiplizieren. Leider müssen Sie in der Praxis aber häufig feststellen, dass dann bei der praktischen Umsetzung Programm- oder Datenbankstrukturen auftauchen, die vorher 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. Entscheidend ist bei der Analyse auch, was

eigentlich zu migrieren ist. Problematisch sind Projekte, in denen die Datenbank und das Frontend komplett und gleichzeitig zu migrieren sind. Wenn möglich, sollte im ersten Schritt immer nur die Datenbank ausgetauscht und das alte Frontend beibehalten werden. So kann relativ genau analysiert werden, ob die Oracle-Datenbank die gleichen Ergebnisse liefert wie das alte System. Eventuell auftretende Fehler lassen sich dann der Datenbank zuordnen und sind wesentlich leichter zu lokalisieren. Danach kann das Frontend ausgetauscht oder erweitert werden.

Doch wie kommt man an die benötigten Fakten für die Migration einer Datenbank? Eine Möglichkeit besteht darin, Fragebögen zu entwerfen, in denen die Anzahl der Tabellen, Views und programmierten Objekte eingetragen wird, und sich manuell alle Quellen aus der Datenbank herauszuziehen. Je mehr Datenbanken zu migrieren sind, desto ungenauer werden jedoch in aller Regel die Angaben in den Fragebögen. Um diesem Dilemma zu entfliehen, entwickelte Oracle Consulting und Sales Consulting das Werkzeug Oracle Migration Assessment Tool (OMAT) (Bild 1), das kostenlos zum Download angeboten wird [1]. OMAT ist ein Java-basiertes Programm, mit dem zurzeit Datenbanken von Microsoft SQL Server, Sybase Adaptive Server und IBM DB2 (Unix und Windows) analysierbar sind. Die Ergebnisse des Analyselaufes sind ein HTML-Dokument mit Informationen zur Anzahl der vorhandenen Datenbankobjekte und zu Besonderheiten der Migration (z.B. Namenskonventionen) sowie die Sourcecodes der Views, Stored Procedures/Functions und Trigger. Durch den Einsatz dieses Tools haben Sie sofort ein sehr genaues Mengengerüst und den gesamten Sourcecode, der sich in der Datenbank befindet. So lässt sich relativ einfach der Aufwand für die Migration der Datenbank abschätzen. Ferner haben Sie damit ein Dokument, das Sie einem Pflichtenheft beifügen können. Mitunter verwenden Benutzer diesen HTML-Report sogar als Dokumentation für ihr Datenbanksystem. Ein Ergebnis der Analyse kann aber auch sein, dass keine (!) Migration durchgeführt wird. Eine Migration lohnt sich nur, wenn der manuelle Anteil an der Konvertierung nicht höher als 15 Prozent der gesamten Aufwände ist. Liegt er höher, ist in vielen Fällen eine Neuprogram-

Die OMAT-Eingabemaske (Bild 1)





Einbindung des JDBC-Treibers im SQL Developer (Bild 2)

mierung sinnvoller. Außerdem wird ein System, das vorher schon ein schlechtes Datenbankdesign hatte, durch eine Migration nicht besser. Dann migrieren Sie eventuell nur Teile, obgleich auch hier in aller Regel nur ein Redesign auf Basis der alten Lösung hilft.

Migration in der IT

Der Begriff „Migration“ beschreibt in der Informationstechnologie die Portierung und einhergehend damit den Wechsel einer wichtigen Software, den Datenaustausch aus einer Umgebung in eine andere sowie das Umstellen einer Hardware auf eine neue Plattform. Die Migration umfasst die Teilbereiche Software-, Daten-, Anwendungs- und Hardware-Migration, die in der Regel eng miteinander verbunden sind. Bei der Datenmigration wird ein altes (Datenbank-)System durch ein neues oder alternatives (Datenbank-)System ersetzt, wobei nicht nur die ursprünglichen – gegebenenfalls gefilterten – Daten, sondern auch darauf aufsetzende Anwendungen übernommen und angepasst werden müssen, um sie anschließend in das neue Zielsystem zu laden und entsprechend den jeweiligen Anforderungen weiterzuarbeiten. Die Migration selbst wird häufig durch ETL-Tools (*Extract, Transform, Load*) oder durch SQL-Skripte realisiert.

[am]

Schritt 2: Die Konvertierung

Oracle bietet für die Migration von Microsoft-SQL-Server-, MySQL-, Microsoft-Access-, Sybase-Adaptive-Server- und IBM-DB2-Datenbanken die im SQL Developer integrierte Migration Workbench an. Auch dieses Werkzeug kann kostenlos heruntergeladen werden. Oracle-Partner liefern weitere kostenpflichtige Lösungen für die Migration von Mainframe-, Informix- oder Adabas/Natural-Datenbanken. Schon während der Konvertierung gehören ausgedehnte Tests zur Migration. Der Testaufwand beträgt in aller Regel mehr als 50 Prozent der gesamten Projektlaufzeit. Da die Migrationstools immer weiter verbessert werden, wird dieser Anteil in Zukunft eher weiter ansteigen. Zu einem guten Pflichtenheft gehören immer Testszenarien und Informationen dazu, von wem die Tests durchzuführen sind. Durch gute Tests können Tuningmaßnahmen, die sonst nur während des späteren Betriebes durchführbar sind, schon vorher in Angriff genommen werden.

Schritt 3: 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 Tuningschritte können sich beispielsweise ergeben, da für andere Datenbanken optimierte Abfragen unter Oracle einen anderen Ablaufplan haben und anzupassen sind.

Der Oracle SQL Developer

Der SQL Developer ist das kostenlose Entwicklungswerkzeug für Oracle-Entwickler. Es ist mit einer grafischen Oberfläche ausgestattet und bietet 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 Version herunterladen [2], da das Produkt ständig erweitert wird. Die zurzeit aktuelle Version ist 2.1. Mit dem SQL Developer kann nicht nur für Oracle-Datenbanken, sondern auch gegen die folgenden Datenbanken entwickelt werden: Microsoft SQL Server, Sybase Adaptive Server, MySQL und IBM DB2. Dazu benötigen Sie allerdings einen JDBC-Treiber des entsprechenden Anbieters. Für Microsoft SQL Server oder Sybase Adaptive Server brauchen Sie den JDBC-Treiber, den Sie unter [3] erhalten. Über *Extras/Voreinstellungen/Datenbank/JDBC-Treiber eines anderen Herstellers* binden Sie den gewünschten Treiber ein (Bild 2). Der nächste Schritt ist dann das Anlegen einer Verbindung zum SQL Server. Die einfachste Lösung ist ein Klick auf das große grüne Kreuz auf der oberen linken Seite des SQL Developer. Mit *Anmelden* wird im folgenden Dialog eine neue Verbindung angelegt, die auch auf der linken Seite dieses Dialoges erscheint. Der Standardport für den SQL Server ist 1433, und Sie benötigen auch das Recht seitens des SQL Server, eine entfernte Verbindung aufbauen zu dürfen. Nun können Sie beispielsweise einen *Select*-Befehl gegen den SQL-Server absetzen und sich die Ergebnisse, wie in Bild 3 dargestellt, ansehen. Jetzt haben Sie bei Migrationen die Möglichkeit, denselben Befehl gegen den SQL Server und Oracle abzusetzen und die Ergebnismengen zu vergleichen.

Die Migration Workbench

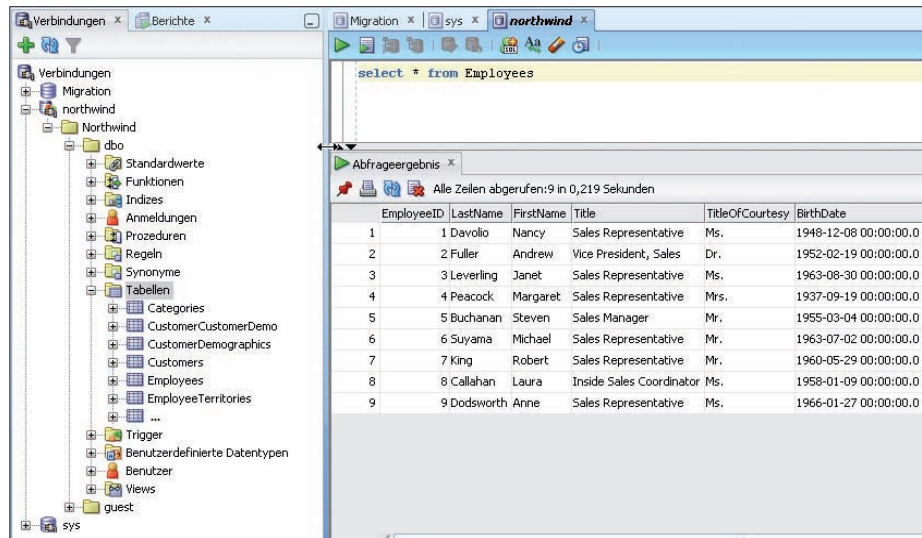
Bis vor einigen Jahren war die Oracle Migration Workbench (OMWB) ein eigenständiges Produkt der Oracle Migration Technology Group. Mittlerweile ist sie fester Bestandteil des SQL Developer. Damit lassen sich Datenbanken von MS Access (man benötigt eine englische (!) Access-Version), MS SQL Server, Sybase Adaptive Server, MySQL und IBM DB2 migrieren. Es werden die Tabellen, Views, Daten, Benutzer und programmierten Objekte (Stored Procedures/ Functions und Trigger) migriert. Bei den programmierten Objekten lassen sich bei Sybase und Microsoft mehr als 90 Prozent der Funktionalität konvertieren. Es wird wohl niemals ein Tool geben, das 100 Prozent migrieren kann, da es immer Unterschiede zwischen den Anbietern geben wird, die eine Migration verhindern. Bei

spielsweise hat Oracle eine komplett andere Funktionalität bei Triggern implementiert. Durch den Einsatz sogenannter Rowlevel-Trigger (zünden bei jeder Veränderung einer Zeile), einer Technik, die es in Sybase- und Microsoft-Datenbanken nicht gibt, benötigt man nicht die Inserted- und Deleted-Cursor wie bei Microsoft und Sybase mit entsprechenden T-SQL-Konstrukten.

Aufgrund der problematischen Locking-Architektur bei COMMITTED-READ-Betrieb wird bei Microsoft und Sybase sehr häufig mit temporären Tabellen und Resultsets gearbeitet. Dadurch werden Lesesperren auf der Ergebnismenge vermieden. Da Oracle keine Lesesperren kennt, kann in vielen Fällen mit Views statt mit Stored Procedures gearbeitet werden, was den Administrationsaufwand sehr stark reduziert. In einigen Projekten wurden mehr als 1.500 Stored Procedures durch wesentlich weniger Views ersetzt, was außerdem die Performance deutlich verbesserte. Migration bedeutet nicht immer eine 1:1-Umsetzung, denn eine Oracle-Datenbank, die sich verhält wie eine SQL-Server-Datenbank, verzichtet auf viele Leistungsmerkmale. In solchen Fällen sind manuelle Anpassungen vorzunehmen, bei der die OMWB gute Dienste leistet. Um eine Migration durchzuführen, benötigen Sie ein Oracle Schema für das sogenannte „Migrations Repository“. In diesem Repository befinden sich die Daten über die Migrationsstrukturen. Die OMWB lädt die Metadaten der Fremddatenbank in der ersten Phase 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 Migration angelegt. Nun muss wie oben gezeigt eine Verbindung zu diesem Schema erzeugt werden. Der nächste Schritt ist dann das Anlegen des Repository in diesem Schema. Unter Extras/Migrieren/Repository Management/Repository Anlegen legen Sie das Repository Ihrer Wahl an. Ist dies erfolgt, ändert sich das Fenster des SQL Developer. Auf der linken Seite erscheinen zwei neue Teilbereiche: Erfasste Modelle und Konvertierte Modelle.

Erfassen des Modells

Der erste Schritt der Migration ist das Erfassen (Capture) der Fremddatenbank. Exemplarisch soll hier die Northwind-Datenbank mit dem Schema DBO migriert werden. Dazu klicken Sie mit der rechten Maustaste auf dieses Schema, und es erscheint ein Fenster mit Schema erfassen als Option. Diese wählen Sie aus und das Modell wird erfasst, das heißt, Oracle lädt die Meta-



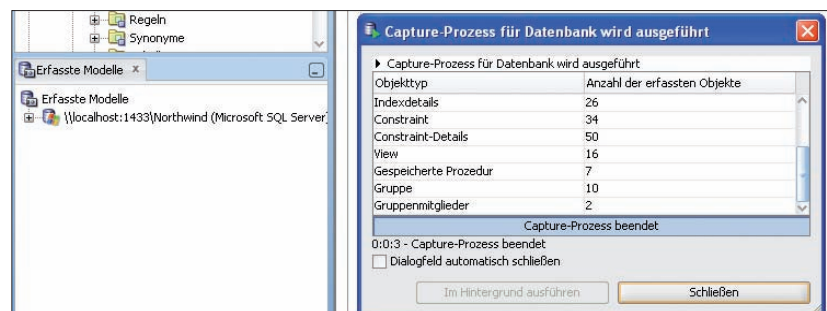
SQL-Developer-Select gegen SQL Server absetzen (Bild 3)

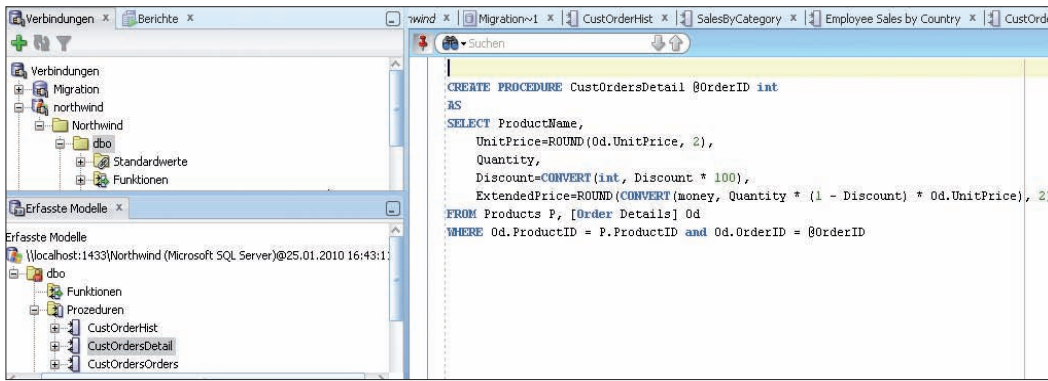
daten des SQL Server in das Repository. Es erscheint ein weiteres Fenster mit der Statistik der erfassten Objekte, und auch im Fenster für Erfasste Objekte wird ein Eintrag generiert (Bild 4). Wie in Bild 5 zu erkennen ist, haben Sie jetzt die Möglichkeit, sich die einzelnen erfassten Objekte anzusehen und eventuell auch schon Anpassungen vorzunehmen oder Objekte zu löschen, die nicht zu migrieren sind. Nach dem Erfassen hat die OMWB alle Informationen im Repository gespeichert, die nötig sind, um die Datenbankobjekte zu migrieren. Es ist auch möglich, mehrere Modelle in einem Repository zu speichern. Diese werden alle in den Fenstern angezeigt.

Konvertierung des Modells

Bevor der Konvertierungslauf durchgeführt wird, legen Sie das Mapping der Datentypen vom SQL Server auf die Oracle-Datenbank fest. Dazu klicken Sie mit der rechten Maustaste auf das erfasste Modell. Daraufhin erscheint ein Fenster mit der Option Datenzuordnung festlegen. Nach der Wahl sehen Sie das in Bild 6 dargestellte Fenster mit den Standardeinstellungen zur Transformation der Datentypen. Sie passen die Datentypen an, wenn sie nicht Ihren Wünschen entsprechen, oder fügen auch neue Regeln hinzu. Wählen Sie für den Konvertierungslauf mit einem Rechtsklick auf das erfasste Modell die Option Zu Oracle-Modell konvertieren. Nach-

Ausschnitt mit den Erfassungsfenstern (Bild 4)





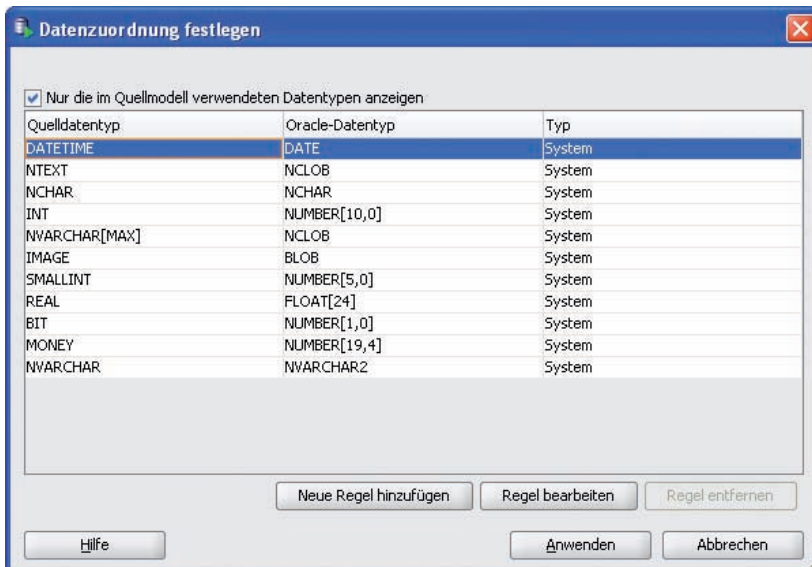
Ausschnitt mit Objektbaum im Erfassungsfenster (Bild 5)

dem das Statusfenster geschlossen wurde, erhalten Sie im Log-Fenster des SQL Developer Informationen zur Konvertierung. Außerdem besteht 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.

Generierung

Mit der rechten Maustaste klicken Sie für die tatsächliche Generierung das konvertierte Modell an und wählen die Option *Generieren*. Als Ergebnis wird ein Migrationsskript angelegt, das in einem SQL-Arbeitsblatt des SQL Developer angezeigt wird. Dieses Skript kann angepasst (z.B. Schema- und Tablespace-Namen) und in einer Textdatei gespeichert werden. Außerdem können Sie auch einzelne Objekte auswählen und in der Datenbank erzeugen lassen. Große Stored Procedures werden in aller Regel so migriert und getestet. Wenn Sie das generierte Skript ausführlich studieren, werden Sie über ein PL/SQL-Package mit dem Namen *sqlserver_utilities* stolpern. Da die String-, Datums-, Zeit- und ma-

Fenster mit der Datenzuordnung (Bild 6)



thematischen 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. Diese verhalten sich genau so wie die namensgleichen SQL-Server-Funktionen, und Sie können sie auch

stand-alone generieren, das heißt, Sie nehmen 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). Unter *Extras/Migrieren* finden Sie die Option *Schnellmigration*. Hier werden Sie, wie **Bild 7** zeigt, mit einem Assistenten in mehreren Schritten durch die Migration geführt. In der Praxis wählen Sie diese Option nur, wenn es sich um eine kleine Datenbank mit wenigen Stored Procedures handelt.

Datenmigration

Nachdem die Strukturen erfolgreich migriert wurden, sind auch die Daten zu migrieren. Der SQL Developer bietet zwei Optionen an: direkt über JDBC aus dem Tool heraus oder indirekt über das Erzeugen von Entlade- und Ladeskripten. Bei der zweiten Option bedienen Sie sich im Falle von Microsoft- und Sybase-Datenbanken auch des Bulkcopy-Tools in Kombination mit dem Oracle SQL*Loader. Die erste Option eignet sich nur für kleine Datenmengen, beispielsweise, wenn Testdaten zu migrieren sind. Die zweite Option wird in aller Regel benutzt, um die produktiven Daten zu migrieren. **Bild 8** zeigt, wie unter *Extras/Voreinstellungen/Migration* individuelle Einstellungen für die Datenmigration vorgenommen werden.

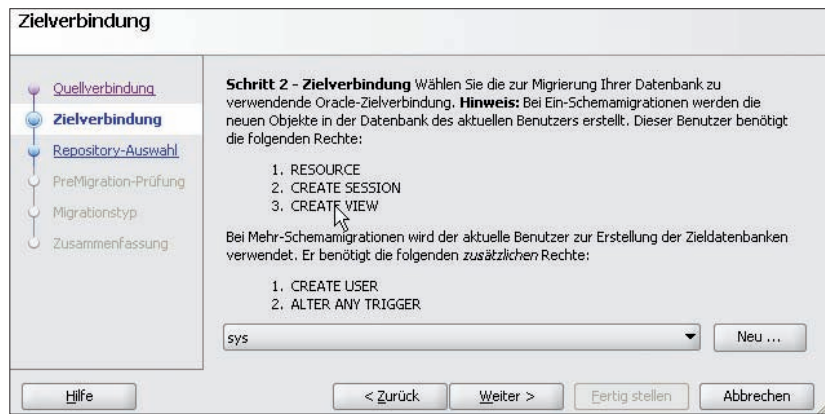
Für eine Online-Migration muss eine Verbindung zum Zielschema in der Oracle-Datenbank eingerichtet sein. Dazu klicken Sie mit der rechten Maustaste das gewünschte, konvertierte Modell an und wählen dann die Option *Daten verschieben*. Danach geben Sie nur noch die Ursprungs- und die Zielverbindung ein, und die Datenmigration startet. In einem gesonderten Fenster werden dann der Status und die Anzahl der Fehler beim Transport angezeigt. Auch die Generierung der Skripte für die Offline-Migration wird in ähnlicher Weise durchgeführt. Mit der rechten Maustaste klicken Sie das konvertierte Modell an und wählen dann die Option *Skripte für Datenverschiebung generieren*. Es er-

scheint ein Fenster, in dem Sie den Speicherort im Dateisystem wählen. In einem weiteren Fenster wird der Status der Generierung angezeigt. Neben einem Verzeichnis *Oracle* werden zwei Batch-Files im ausgewählten Verzeichnis erzeugt: *unload_script.bat* und *oracle_ctl.bat* (für Linux/Unix werden auch *.sh*-Dateien angelegt). *Unload_script.bat* enthält die Bulkcopy-Aufrufe für den SQL Server. Sie müssen nur noch den Namen des Servers und des Benutzers mit Password editieren und beginnen dann mit dem Entladen. In *oracle_ctl.bat* finden Sie neben den Aufrufen des SQL*Loader 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, sollten Sie die entsprechenden NLS-Settings für Zeichensatz und Datumsformat anpassen.

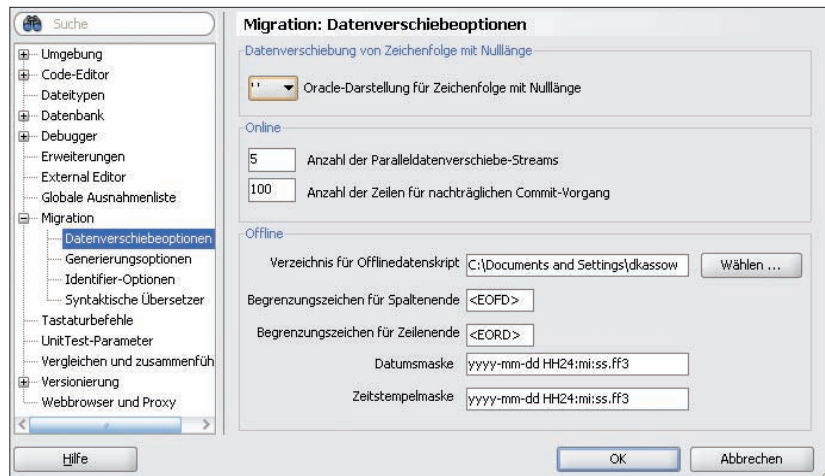
Migration Scratch-Editor

Sehr häufig kommt es vor, dass Sie nur einzelne Objekte wie eine T-SQL-Stored-Procedure nach PL/SQL migrieren wollen. Der Aufwand, das mit der OMWB durchzuführen, ist dafür viel zu groß. Aus diesem Grunde ist im SQL Developer der sogenannte Migration Scratch-Editor implementiert, den Sie mit dem Befehl *Extras/Migrieren/Scratch-Editor für Übersetzung* aktivieren. Es erscheinen zwei neue Fenster im SQL-Developer: eines links für den Quellcode der Fremddatenbank und rechts die dazugehörige Ausgabe in PL/SQL. Sie kopieren den gewünschten Quellcode in das linke Fenster, klicken auf das Symbol für das Übersetzen (>>) und erhalten die in **Bild 9** wiedergegebene Ausgabe.

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-Äquivalent. 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. Klicken Sie auf das Symbol *Diff Editor für Übersetzung*, erhalten Sie eine direkte Anzeige, welcher Befehl in T-SQL mit welchem in Oracle PL/SQL korrespondiert. Gerade bei großen Objekten ist das eine wichtige Hilfe. Natürlich benötigen Sie während einer Migration eine Übersicht über die durchgeführte Migration, Fehlermeldungen und Mengengerüste.



Dialogfenster zur Schnellmigration (Bild 7)



Optionsmaske zur Datenmigration (Bild 8)

Unter *Ansicht/Berichte/Migrationsberichte* generieren und exportieren Sie Berichte.

Damit haben Sie grundlegende Informationen zur Migration nach Oracle kennengelernt.

[am]

- [1] Oracle Migration Assessment Tool; www.oracle.com/global/de/community/platform/index.html
- [2] Oracle SQL Developer; www.oracle.com/technology/software/products/sql/index.html
- [3] The jTDS Project; <http://jtds.sourceforge.net>
- [4] Community News der Business Unit Server Technology; www.oracle.com/global/de/community/platform/index.html

Der Migration Scratch-Editor (Bild 9)

