

# Metadaten-Erstellung mit der Oracle BI Suite automatisieren

Jonas Strauß, Schütze Consulting AG

*Bei der Einführung von BI-Systemen wird der Aufwand für die Überführung komplexer, historisch gewachsener Datenstrukturen häufig unterschätzt. Dies führt dazu, dass eine termingerechte Ausführung des BI-Projekts und das geplante Budget oftmals nicht eingehalten werden können. Bei komplexen Migrationsprojekten auf die Oracle Business Intelligence Enterprise Edition (OBIEE) lohnt es sich deshalb, die Potenziale einer Automatisierung zu betrachten.*

Im Rahmen eines Migrationsprojekts der Firma Schütze Consulting AG wurden die Möglichkeiten der Automatisierung evaluiert und konnten erfolgreich erprobt werden. Die bei der Evaluation identifizierten Werkzeuge und Skripte ermöglichen dem Anwender die automatisierte und somit effiziente Erstellung von Repositories.

Die angewandte Vorgehensweise eignet sich insbesondere für folgende Anwendungsszenarien:

- Migration auf die Oracle BI Suite, die aufgrund der Einstellung des Supports für den Oracle Discoverer notwendig wird
- Neueinführung der Oracle BI Suite in historisch gewachsenen Datenstrukturen

Die Nutzung der Skripte und Werkzeuge der OBIEE können jedoch nicht nur bei einer Neueinführung, sondern auch bei einer Überarbeitung beziehungsweise Konsolidierung von Data-Warehouse-Strukturen zur Wartbarkeit und Verbesserung der BI-In-

frastruktur helfen. Im Folgenden werden das oben genannte Projekt und die verwendeten Automatisierungsmethoden dargestellt.

## Ausgangslage

Im Rahmen des betreffenden Projekts wurde die Migration umfangreicher Auswertungsbestände aus dem Oracle Discoverer zur Oracle BI Suite angestrebt. Aufgrund der Größe der Auswertungsbereiche wurde zu einem frühen Zeitpunkt über eine Automatisierung nachgedacht. Als Ziele der Automatisierung wurden die folgenden Vorteile erwartet:

- Einsparungen von Zeit und Aufwand
- Reduzierung von Fehlern durch manuelle Prozesse
- Reproduzierbarkeit von Migrationsschritten
- Fokussierung auf die komplexen Migrationsschritte durch Automatisierung der wiederkehrenden Aufgaben
- Durchsetzung einer einheitlichen Namenskonvention

Im ersten Schritt hat man daher die Möglichkeiten der skriptgesteuerten Konfiguration der Oracle BI Suite analysiert und eine entsprechende Methodik entwickelt.

## Methodik

Zu Beginn des Projekts wurde eine Überführung der bestehenden Metadaten aus dem Discoverer angestrebt. Aufgrund der projektspezifischen Datenstrukturen war eine direkte Überführung der Architektur mit großen Performance-Problemen verbunden. Aus diesem Grund hat man mit einer datenbankseitigen Transformation der breiten Auswertungstabellen in ein Star-Schema begonnen. Hierzu wurden PL/SQL-Routinen geschrieben, die diesen Schritt unterstützen. Im Ergebnis entstanden themenspezifische Faktentabellen sowie einheitliche Dimensionen.

Analog zu diesen themenbezogenen Stars wurden nun die Strukturen in der Oracle BI Suite aufgebaut. Skripte der Oracle BI Suite und programmierte PL/SQL-Routinen halfen, den folgende Prozess zu implementieren:

- Auslesen der Metadaten des Star-Schemas
- Transformierung in eine Repository-Datei
- Bereitstellung und Nachbereitung

Das Datenfluss-Diagramm in *Abbildung 1* zeigt den Verlauf der Daten innerhalb der vorgestellten Methodik sowie die Schnittstellen zur Datenbank und zum OBIEE-Server.

## Das Auslesen

Als Erstes wurden alle notwendigen Metadaten mit Unterstützung von Abfragen aus der Datenbank ausgelesen. Dies umfasste

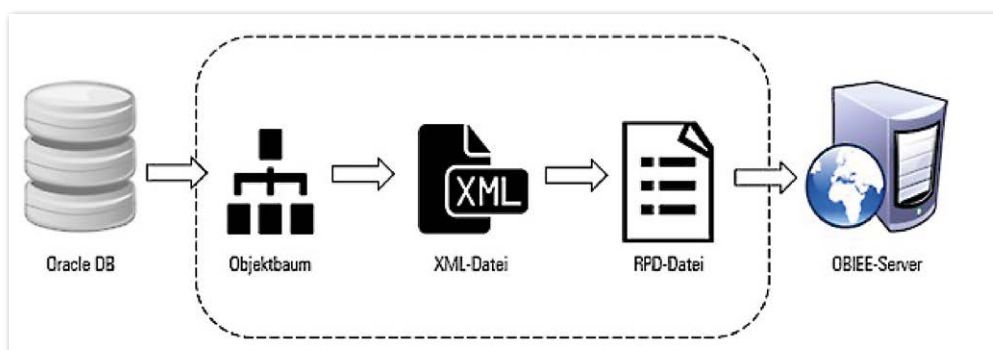


Abbildung 1: Datenfluss-Diagramm der vorgestellten Anwendungen

die Informationen zur Fakten- und Dimensionstabelle, die Primär- und Fremdschlüssel sowie die Tabellenspalten. Die einzelnen Faktentabellen der Star-Schemata konnte man anhand eines Prä- oder Suffix identifizieren. *Listing 1* stellt die im ersten Schritt identifizierten Faktentabellen dar. Anhand des Listings ist zu erkennen, dass die Faktentabellen mit einem Suffix „\_F“ gekennzeichnet sind, was eine eindeutige Identifizierung ermöglicht.

Nachdem alle Faktentabellen identifiziert sind, ist es möglich, die verbundenen Dimensionen über die Fremdschlüsselrelationen zu bestimmen. Dafür werden die Informationen mithilfe eines SQL-Skripts aus dem Data-Dictionary der Oracle-Datenbank über ein zweistufiges Verfahren kompiliert. Nach diesem Schritt lagen alle benötigten Informationen zu einem Star-Schema in tabellarischer Form vor und konnten beliebig ausgelesen und weiterverarbeitet werden.

### Die Transformation

Aus den gesammelten Informationen wurde mittels Java Architecture for XML Binding (JAXB) eine XML-Datei erzeugt. Dabei handelt es sich um eine Programmier-Schnittstelle zur automatischen Generierung von Java-Klassen aus einem XML-Schema.

Zunächst wurde zur besseren Überführung der Daten in die Struktur der OBIEE-XML eine Objektstruktur unter Verwendung von JAXB aufgebaut, die auf den im OBIEE zugrunde liegenden XML-Schemata basiert. Dieser Vorgang heißt auch „XML-Datenbindung“, da es sich um eine Methodik zur Abbildung von XML-Dokumenten in Objektstrukturen handelt. Die neu erzeugte JAXB-Objektstruktur beinhaltet nach diesem Schritt sämtliche Objekte, die zur Bestückung der drei Layer des BI

Administration Tools (physisch, Geschäftsmodell und Präsentation) notwendig sind.

Die einzelnen Klassen konnten nun in einen Workflow zum Befüllen der Objekte integriert werden. *Listing 2* zeigt beispielhaft das Befüllen des Objekts „PhysicalTableT“. Es enthält die physischen Tabellen-Informationen aus der Datenbank.

Die JAXB-Objekte ließen sich nach diesem Schritt ohne Beachtung einer Reihenfolge füllen. Anschließend konnte das XML-Objekt in eine XML-Datei umgewandelt und in einem vordefinierten Ordner abgelegt werden. Dieser Vorgang ist für jedes identifizierte Star-Schema wiederholbar. Pro Star-Schema war somit eine XML-Datei entstanden.

### Bereitstellen und Nachbereiten

Im Anschluss war es notwendig, die XML-Dateien in eine Repository-Datei umzuwandeln und anschließend auf dem OBIEE-Server zu veröffentlichen. Die Erstellung und Zusammenführung sowie das Deployment des Repository wurden unter Verwendung von Skripten, die der OBIEE bereitstellt, durchgeführt. Oracle stellt im XML-Bereich folgende Skripte zur Verfügung:

- *Biserverxmlgen*  
Generiert XML-Code aus einem bestehenden Repository
- *Biserverxmllexc*  
Erstellt oder ändert ein bestehendes Repository im Offline-Modus
- *Biserverxmlcli*  
Migriert eine XML-Datei mit dem Repository des Oracle BI Server

Mit dem Deployment auf dem OBIEE-Server ist der letzte Arbeitsschritt des Tools abgeschlossen. Im Anschluss steht das Repository für weitere Bearbeitungen über das BI Administration Tool zur Verfügung.

Das entstandene Repository ist als Fundament zu betrachten, da nur wiederkehrende beziehungsweise allgemeingültige Arbeitsschritte per Skript erledigt werden können. Dennoch ist die Repository-Datei für erste Auswertungen einsetzbar und kann direkt durch OBIEE Analytics verwendet werden. So lassen sich mit minimalem Customizing-Aufwand erste Probe-Auswertungen auf den eigenen Daten durchführen.

### Fazit

Eine Kombination aus Java- und PL/SQL-Skripten sowie den Skripten der Oracle BI Suite kann eine automatisierte Repository-Erstellung ermöglichen und so den Projektverlauf der BI-Einführung in vielerlei Hinsicht positiv beeinflussen. Die Entwicklungsaufwände werden durch die Automatisierung erheblich reduziert, wovon insbesondere kritische Migrationsprojekte mit einem umfangreichen Data Warehouse profitieren können. Dies führt darüber hinaus zu einer höheren Qualität des Online-Repository. Testdurchläufe der einzelnen Elemente können zu einem frühen Zeitpunkt erfolgen.

Nicht zuletzt bleibt zu erwähnen, dass der Erfolg eines BI-Projekts stets von der Akzeptanz seiner Anwender abhängt. Aufgrund einer integrierten Namenskonvention und somit der einheitlichen Basis sind Anwender durch den Aufbau per Skript in der Lage, sich schneller in den OBIEE-Strukturen zurechtzufinden.

Ein mit den Skripten und Werkzeugen von OBIEE erstelltes Repository sollte dabei als Grundgerüst betrachtet werden, auf das im weiteren Projektverlauf aufgebaut werden kann. Eine solche Lösung bildet dabei das Fundament der BI-Einführung, das mithilfe von individuell konfigurierbaren Namenskonventionen und der Automatisierung von Arbeitsabläufen den Projektverlauf leichter planbar und dadurch steuerbar macht. Aufwände können erheblich eingespart werden, wodurch mehr zeitliche Kapazitäten für die individuelle und zeitintensive Repository-beziehungsweise Berichtserstellung bleiben.

Eine automatisierte Repository-Erstellung kann somit letztlich in erheblichem Maße zum Projekterfolg eines Migrationsprojekts beitragen und diesen positiv beeinflussen. Die Schütze Consulting AG hat die im Artikel beschriebenen Werkzeuge in einer Java-Lösung gebündelt und wendet dieses Tool erfolgreich in ihren Projekten an.

Jonas Strauß

[j.strauss@schuetze-consulting.ag](mailto:j.strauss@schuetze-consulting.ag)

```
Select fact.table_name
FROM dba_tables fact
WHERE SUBSTR(fact.TABLE_NAME, LENGTH(fact.TABLE_NAME) - 1) = '_F';
```

Listing 1

```
String rOwner = "DB_Schema_NAME";
String rTableName = "TABLE_NAME";
String rDataBaseName = "DB_NAME";
// PhysicalTable
final PhysicalTableT table = new PhysicalTableT();
table.setName(rTableName);
table.setParentName("\"\" + rDataBaseName + "\"\".\".\"\" + rOwner + "\"\"");
physicalTable.add(table);
```

Listing 2