

DBMS_METADATA - Flexibler Umgang mit DDL

Robert Marz
its-people
Frankfurt

Schlüsselworte

DBMS_METADATA, Transformation, XML, DDL, Konfigurationsmanagement

Einleitung

Die Definitionen von Schemadaten über mehrere Systeme hinweg synchron zu halten und zu versionieren ist schwerer als den berühmten Sack Flöhe zu hüten.

DBMS_METADATA erweist sich für diese Aufgaben als gut versteckte aber schlecht dokumentierte Perle, die weit mehr kann, als man ihr auf den ersten Blick ansieht.

Das Paket dient als Basis für die Datapump und kann weit mehr, als nur DDLs im Klartext für Tabellendefinitionen zu erzeugen:

Unter anderem kann zwischen Script- und XML-Format gewählt, Filter gesetzt und Transformationen durchgeführt werden.

Die Skripte für abhängige Objekte können gleich mit exportiert werden.

Während die Entwickler oftmals nur die Differenzskripte zwischen ihren Releases pflegen und liefern, erzeugt dbms_metadata die Initialskripte automatisch, um die Konfigurationsmanager zufrieden zu stellen.

Dieser Vortrag beleuchtet folgende Themen:

- Erzeugen von Quelltext für Schema Revisionen
- Von einfach zu komplex: Umgang mit den PL/SQL APIs
- Erzeugen von Schema- und Objekt-DDLs mit Änderungen

- Löschen, modifizieren und wiederanlegen von Datenbankobjekten und deren Abhängigkeiten

Wurzeln

Das Package dbms_metadata gehört seit Oracle 9i zum Standardlieferungsumfang der Datenbank.

Es wird beständig weiterentwickelt und im Funktionsumfang verbessert.

Seit Oracle 10g ist dbms_metadata das Fundament der Datapump. So ist es kaum verwunderlich, dass sich Parameter und Optionen der beiden Features sehr ähnlich sind.

Wie der Name schon vermuten lässt, kann das Package die Metadaten, also die Internas der Datenbank beschreiben. Intern arbeitet das Programm XML-basiert. Diese XML lässt sich auch speichern und später wieder laden.

Genau wie die Datapump, garantiert auch dbms_metadata Aufwärtskompatibilität, so dass sich XML, das mit der Version 9i erzeugt wurde, auch noch in 11gR2 verarbeiten lässt.

Das XML selbst bekommt der Anwender nur selten zu Gesicht, da es in der Regel vor der Ausgabe mit XLST-Stylesheets in ausführbares DDL umgewandelt wird.

Das PL/SQL-API von `dbms_metadata` bietet Funktionen, mit denen man durch die Datenbank browsen kann und sich via SQL-Abfrage einen Überblick über die vorhandenen Objekte verschaffen kann.

Denn vollen Funktionsumfang kann man allerdings nur aus einem PL/SQL-Programm heraus nutzen.

Anwendungsbeispiele

Das Package lässt sich sehr vielfältig einsetzen:

Konfigurationsmanagement

Die Entwickler arbeiten auf dem Stand des letzten Releases.

Alle Änderungen werden in Form von Patches mit Alter-Skripten ausgeliefert.

Nach der Installation der Software, wird mit `dbms_metadata` das DDL für den Neuaufbau des kompletten Schemas erzeugt. `Dbms_metadata` erzeugt die Befehle in der richtigen Reihenfolge, so dass keine Kompilierungsfehler auftreten. Alle Abhängigkeiten wie Constraints und Grants werden mitgeneriert.

Anschließend werden die DDL-Anweisungen mit `utl_file` ins Dateisystem geschrieben und eingchecked.

Partitionieren von Tabellen

Um Tabellen automatisiert in partitionierte umzuwandeln, werden zunächst die betroffenen Tabellen mittels `create table as select` samt Daten kopiert. Dabei werden die Partitionen schon mit angelegt.

Die kopierten Tabellen bekommen temporäre Namen.

Mittels `dbms_metadata` werden anschließend alle abhängigen DDLs der Originaltabellen, also zum Beispiel Constraints und Grants erzeugt.

Jetzt werden die Quelltabellen gelöscht und die neuen Tabellen auf die alten Namen umbenannt.

Nach dem ausführen der abhängigen DDLs sehen die neuen Tabellen aus wie die Alten, nur dass sie jetzt über Partitionen verfügen.

Umziehen von Schemas

Das Umziehen oder in diesem Falle kopieren von Schemas ist mit Hilfe von `dbms_metadata` einfach zu bewerkstelligen.

Zunächst werden das XML der Objekte mit `dbms_metadata.fetch` erzeugt, dann mithilfe von `dbms_metadata.modify` die Schema- und Tablespacenamen angepasst.

Mit `dbms_metadata.put` kann das so veränderte XML direkt ausgeführt und so die Objekte im neuen Schema angelegt werden.

Zum Schluss werden noch die Daten kopiert und der Vorgang ist abgeschlossen.

Ermitteln von Unterschieden

Seit Oracle 11g kann `dbms_metadata` auch Objekte vergleichen und die Unterschiede als speichern.

Das XML kann anschließend in Alter-Anweisungen gewandelt und ausgeführt werden.

Damit lassen sich nachträglich auch Skripte für erstellen, die eine Version eines Schemas in die nächste überführen, z.B. für ein Deinstall.

Details

Das API von `dbms_metadata` ist sehr mächtig, aber auch sehr komplex.

Einige Aspekte erschließen sich aus der Dokumentation im „PL/SQL Packages and Types Reference“ nicht sofort...

Sicherheit

User können `dbms_metadata` für ihre eigenen Objekte aufrufen. Von fremden Objekten werden lediglich die grants zurückgeliefert.

Ein Aufruf für Objekte, die dem aktuellen Schema nicht gehören, liefert keine Daten zurück.

Schemas, die über die `select_catalog_role` verfügen, dürfen alle Objekte abfragen.

Objekte, die Kennwörter beinhalten, wie DB Links, User oder Rollen, werden ohne diese beschrieben.

Wenn die Kennwörter mit erzeugt werden sollen, muss der aufrufende User SYS sein, oder über die Rolle `exp_full_database` verfügen.

Filter

Wenn Metadaten für mehrere Objekte abgefragt werden, kann man die Abfrage mit Filtern einschränken. Filter können einfach sein, indem zum Beispiel auf Schema- Objekt oder TablespaceName eingeschränkt wird. Es sind aber auch komplexe Filter möglich, die in Form von SQL-Prädikaten (where-Klausel-Ausdrücke) Ex- und Inklusionen definieren. Mittels in- und (not) exists-Operatoren sind auch komplexe unterabfragen möglich.

Zur Kontrolle kann liefert die Funktion `get_query` das erzeugte SQL zurück.

Performance

Umfangreiche Aufrufe können schon einmal ein wenig Zeit in Anspruch nehmen.

Mit einigen einfachen Tricks kann man diese jedoch auf ein Minimum reduzieren.

Wenn die Definitionen von vielen Objekten erzeugt werden soll, lohnt sich diese „sortenrein“ zu erzeugen, also z.B. erst alle Tabellen, dann alle Indizes, dann alle Views, usw.

Seit Oracle 11g können auch mehrere Definitionen auf einmal pro Fetch-Aufruf abgeholt werden. Mittels der Funktion `set_count()` kann die Anzahl festgelegt werden.

Generell ist es sinnvoll, Objekte die viel Speicher belegen – wie zum Beispiel XML-Objekte oder LOBs – nur einmal auf Package-Spezifikationsebene zu deklarieren und nicht in jedem einzelnen Unterprogramm des Packages immer wieder.

Fazit

Mit der Einführung von dbms_metadata ist die Zeit von verschachtelten SQL-Abfragen, um DDL für Datenbankobjekte zu erzeugen, vorbei.

DBMS_METADATA ist schon beeindruckend, wenn es in SQL-Abfragen eingesetzt wird, entfaltet sein ganzes Potential aber erst im PLSQL-Block und im Zusammenspiel mit dbms_metadata_diff.

Die Quelltexte zu den in diesem Vortrag gezeigten Demonstrationen können nach der DOAG 2012 Konferenz von unserem its-people-Blog heruntergeladen werden:

<http://www.its-people.de/blog>

Kontaktadresse:

Robert Marz

its-people Hochtaunus GmbH
Lyoner Straße 44-48

60528 Frankfurt am Main

Telefon: +49 (69) 247521-00
Fax: +49 (69) 247521-021
E-Mail: robert.marz@its-people.de
Internet: www.its-people.de