Code Versionierung mittels Edition Based Redefinition und Apex

Sven-Uwe Weller
syntegris information solutions GmbH
Neu-Isenburg

Schlüsselworte
EBR, APEX, Software Life Cycle, Continuous Integration Testing

Einleitung

Oracle 11g hat als eines der Highlights „Edition Based Redefinition“ (EBR) eingeführt. Damit ist es möglich Applikationen auf eine neue Version upzugraden, ohne das eine für den Enduser spürbare Downtime entsteht. In der Praxis wird dieses Feature jedoch kaum eingesetzt.

Der Vortrag stellt eine Verwendung von EBR vor, die deutlich von der Oracle Intention abweicht. Jedoch teilweise überraschende Vorteile und neue Möglichkeiten eröffnet. Die Erkenntnisse stammen aus einem realen Apex-Projekt und wurden innerhalb des letzten Jahres gewonnen und erfolgreich getestet. Ein Highlight welches sich im Laufe des Projektes herausgestellt hat, ist die Einsparung einer kompletten eigenen Testdatenbank durch den Einsatz von EBR und Apex.

Es wird gezeigt wie man EBR einschaltet und verwendet. Wie die Konfiguration zu erfolgen hat und wie man eine Apex Anwendung so einrichten kann, dass dynamisch zwischen verschiedenen Editions gewechselt werden kann.

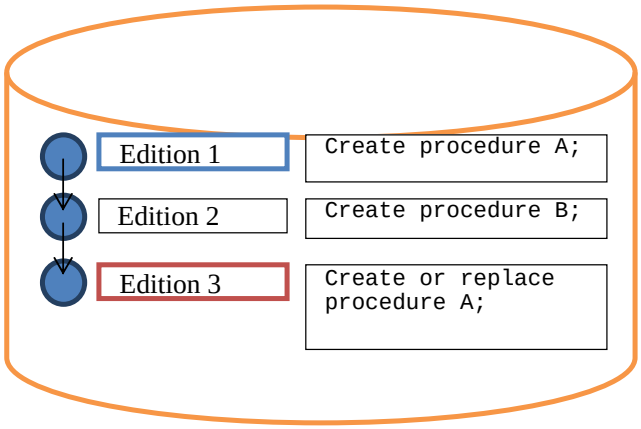
Idee

Eine Anforderung im Projekt war es, dass für verschiedene Mandanten mit unterschiedlichen Programm Versionen gearbeitet werden kann. Dabei bezieht sich Programm Version hauptsächlich auf berechnungsintensiven PL/SQL Code, der in der Datenbank in Packages abgelegt ist. Die Oberfläche (APEX) war damit nicht gemeint. Eine vergleichbare Zielstellung kann in verschiedenen Projekten vorkommen. Vor allem dort, wo aus Dokumentations- oder Beweisgründen es wichtig sein kann, auch inzwischen veraltete Versionen einer Software nochmals einzusetzen.

Die Idee ist, dass die alten PL/SQL Code Versionen in jeweils einer eigenen Edition gehalten werden. Ein neues (PLSQL-)Release wird jeweils in eine neue Edition eingespielt. Die alten Editions werden nach dem Applikationsupgrade nicht gedroppt, sondern bleiben bestehen. EBR wird damit also nicht als kurzzeitige Hilfe zum downtimelosen Applikationsupgrade verwendet, sondern als dauerhafte Funktion zum Umschalten zwischen den Versionen.

Beispiel:
Mandant 1 arbeitet mit Edition3 und kann Prozedur B (vererbt aus 2) und die neue Version von Prozedur A verwenden.

Ein anderer Mandant arbeitet mit Edition 1 und kann nur die alte Version von Prozedur A verwenden.



Umsetzung

Damit EBR genutzt werden kann müssen in der Datenbank einige wenige Aktionen laufen.

- Datenbank generell für EBR aktivieren
- Edition anlegen
- Default Edition setzen
- Schema für Edition freischalten

```
ALTER USER edition_test ENABLE EDITIONS;
CREATE EDITION release_v1 AS CHILD OF ORA$BASE;
ALTER DATABASE DEFAULT EDITION = release_v1;
GRANT USE ON EDITION release_v1 TO edition_test;
```

- Prüfen der Edition
- Umschalten der Edition

```
SELECT SYS_CONTEXT('USERENV', 'SESSION_EDITION_NAME') AS edition FROM dual;
ALTER SESSION SET EDITION = release_v1;
```

Restriktionen

Nicht alle Objekte sind editionierbar!

Editionierbare Objekte	Nicht Editionierbare Objekte
Packages, Funktions, Procedures	Tabellen, Daten
Trigger	Sequences
Types, Type Bodies	Indexe
(editionable) Views	Materialized Views
Libraries	Database Links
Synonyme	Public Synonyme
Virtual Columns	Function Based Indexe

Die Anzahl der Editionen ist begrenzt.

Daten lassen sich nur mit sehr hohem Aufwand editionierbar halten. Dies ist mittels Cross-Edition-Triggern prinzipiell möglich. Jedoch führt die Verwendung von Cross-Edition-Triggern zu zu Performanceverlusten innerhalb der Applikation. Eine permanente Verwendung von Cross_Editio-Trigger ist deshalb nicht zu empfehlen.

EBR als Dauerkonzept funktioniert lt. Ansicht des Autors deshalb nur, wenn die Daten und auch die Tabellen nicht versioniert werden müssen. D.h. es sind durchaus Erweiterungen an Tabellen möglich, diese sollten möglichst weitgehend abwärtskompatibel sein. Kurz gesagt neue Spalten ja, Spalten löschen eher nein.

Vorteile

- Paralleles Entwickeln und Testen auf einer Datenbank! ➔ Einsparung einer kompletten Testinstanz!

- Dynamisches Umschalten von Mandanten (oder Anwendern) auf alten oder neuen Programmcode.
- Umgang mit Editions wird geübt. ➔ Downtimeloses Applikationsupgrade möglich. Vor allem in Kombination mit Apex.

Erfahrungen

In der Praxis ergeben sich einige Dinge, die man beim Umgang mit Editions beachten sollte. Dies beginnt bei der Vergabe des Editon Namens, über die Verwendung von Editions mit dem Oracle SQL Developer oder anderen Tools bis hin zu Performance Effekten. Diese Erfahrungen werden in Form von Tipps im Vortrag näher erklärt und ggf. Lösungsmöglichkeiten aufgezeigt.

Ausblick

Die Verwendung von EBR kann potentiell auch dazu dienen mehrere unterschiedliche Apex Versionen in der gleichen Datenbank zu unterstützen. Das Apex interne Mapping mittels Public Synonymen müsste auf Private synonyme umgestellt werden. Dies wurde jedoch bisher nicht getestet.

Kontaktadresse:
Sven-Uwe Weller
syntegris information solutions GmbH
Hermannstr. 54-56
D-63263 Neu-Isenburg

Telefon:	+49 (0) 6102 298668
Mobil:	+49 (0) 160 7465015
Fax:	+49 (0) 6102 558806
E-Mail	sven.weller@syntegris.de
Internet:	www.syntegris.de