„Versionsverwaltung“ für Datenbankobjekte - Edition Based Redefinition

Daniel Horwedel & Stefan Winkler
merlin.zwo InfoDesign GmbH & Co. KG
Bad Liebenzell

Schlüsselworte

Datenbank, Edition Based Redefinition, Codequalität, Versionierung

Einleitung

Bei der Aktualisierung von Anwendungen ergibt sich für die Entwickler häufig die Anforderung, dass diese ohne große Wartungsfenster erfolgen sollen.

Des Weiteren kann es sinnvoll sein, neue Versionen von Datenbankobjekten innerhalb eines bestehenden Schemas zu testen und die Auswirkungen auf abhängige Objekte zu analysieren.

Oracle bietet zur Lösung dieser Problematik auf der Datenbank-Ebene ein einfach nutzbares Werkzeug an: Edition Based Redefinition (EBR). Hiermit lassen sich mehrere Versionen eines Datenbankobjektes im Schema installieren und sessionabhängig verwenden, wodurch ein Update auf eine neue Version Ihrer Datenbankobjekte oder ein Test der neuen Version ohne Beeinträchtigung des laufenden Betriebs erfolgen kann.

Die Anwender haben in ihrer Session die alte Version der Datenbankobjekte zur Verfügung, während die neue Version installiert und getestet werden kann. Mit einem einfachen „ALTER SESSION“-Befehl kann anschließend die neue Version den Anwendern zur Verfügung gestellt werden.

Klassische Lösungswege

Eine gängige Möglichkeit zum Testen von Datenmodell-Änderungen stellt die Verwendung einer Schema-Kopie dar, mittels derer auf Basis von Produktivdaten der Impact einer Änderung der Datenstruktur ermittelt werden kann. Hierbei entsteht allerdings durch das Vorhalten mehrerer Testumgebungen ein gewaltiger Aufwand, wodurch im Endeffekt oftmals veraltete Testdaten zur Verfügung stehen oder die Testschemas unterschiedliche Versionsstände bei den enthaltenen Datenbankobjekten aufweisen.

Bei der Bereitstellung neuer Anwendungsversionen auf Produktivsystemen ensteht durch die Änderungen im Schema meistens eine Downtime, in der die Produktivität eingeschränkt ist, wodurch ebenfalls hohe Kosten entstehen können.

Versionierung der DB-Objekte als Lösungsweg

Zur Reduzierung des Aufwandes beim Betrieb mehrerer Test- und Entwicklungsumgebungen sowie der Durchführung von „Online Application Upgrades“, also der Bereitstellung neuer Anwendungsversionen ohne Downtime, bietet sich die „Versionierung“ der Datenbankobjekte mittels Edition Based Redefinition an. Hierdurch lassen sich mehrere Versionen eines

Datenbankobjektes innerhalb eines Schemas vorhalten – zwischen diesen Versionen kann im Session-Kontext einfach gewechselt werden.

Hierdurch wird der Aufwand für die Pflege und Bereitstellung von Testumgebungen deutlich reduziert: in vielen Fällen ist nun nur noch eine Testumgebung notwendig, in der direkt – über den Wechsel der in der Session aktiven „Version" der Datenbankobjekte – die Auswirkungen der Änderungen an den Objekten überprüft werden können.

Edition Based Redefinition steht in allen Editionen der Oracle-Datenbank seit der Version 11g Release 2 zur Verfügung.

Grenzen der Edition Based Redefinition

Leider wird der durch Edition Based Redefinition (EBR) gebotene Komfort durch einige Einschränkungen reduziert.

Mittels EBR können lediglich die Objekttypen Synonym, View, Function, Library, Package (Spec & Body), Procedure, Trigger sowie Type (Spec & Body) versioniert werden – eine Versionierung von Tabellen ist mittels EBR nicht möglich.

Im Gegensatz zu privaten Synonymen ist eine Editionierung von Public Synonyms nicht möglich, da sich die Edition immer auf ein konkretes Schema bezieht – bei einem Public Synonym handelt es sich allerdings Objekt, das im Schema PUBLIC angelegt wird, beim Schema PUBLIC handelt es sich allerdings um ein Schema, das systembedingt nicht editioniert werden kann.

Innerhalb eines editionierten Schemas dürfen Public Synonyme allerdings verwendet werden, solange diese nicht auf ein editioniertes Objekt verweisen.

Die Verwendung benutzerdefinierter Datentypen in Tabellen, durch die ein Verweis auf ein editioniertes Objekt erfolgt, ist ebenfalls nicht möglich, ebenso darf ein nicht-editioniertes Subprogram keine statische Referenz auf ein Subprogram haben, dessen Eigentümer-Schema editioniert ist.

Mit der Version 12c der Oracle-Datenbank sind einige dieser Einschränkungen entfallen, so sind Verweise von Materialized Views, Indexen und virtuellen Spalten auf editionierte Objekte möglich, hierbei muss dann explizit angegeben werden, auf welche Edition des Objektes zugegriffen werden soll.

Aktivierung von EBR

Zunächst wird mit dem folgenden Statement überprüft, ob EBR für das betreffende Schema bereits aktiviert wurde:

```
SELECT editions_enabled
  FROM dba_users
 WHERE username = 'MEINSCHEMA';
```

Für die Erzeugung einer neuen Edition wird das CREATE ANY EDITION-Recht benötigt, für das Löschen bestehender Editionen ist das DROP ANY EDITION-Recht erforderlicht. Diese Rechte werden dem Schema mittels folgendem Statement zugewiesen:

```
GRANT CREATE ANY EDITION, DROP ANY EDITION TO meinschema;
```

Bevor nun für das jeweilige Schema die Editionierung aktiviert wird, sollte zunächst überprüft werden, ob in diesem Schema Objekte vorhanden sind, die selbst nicht editionierbar sind, aber gleichzeitig von editionierbaren Objekten abhängen, wodurch eine Aktivierung der Editionierung zu Problemen führen kann.

Mittels folgendem SELECT-Statement kann überprüft werden, ob solche Objekte im Schema vorhanden sind:

```
SELECT d.owner#,
       u.name,
       d.name,
       d.namespace,
       d.stime
  FROM obj$ d,
       dependency$ dep,
       obj$ p, user$ u
 WHERE d.obj# = dep.d_obj#
   AND p.obj# = dep.p_obj#
   AND d.remoteowner is null
   AND p.owner# = (   SELECT user_id
                        FROM sys.dba_users
                       WHERE username = 'MEINSCHEMA'
                  )
   AND d.owner# = u.user#
   AND p.type# in (4,5,7,8,9,10,11,12,13,14,22,87)
   AND (
            (       u.type#              != 2
                AND bitand(u.spare1, 16) = 0
                AND u.user#              != p.owner#
            )
         OR
            (
                d.type# NOT IN (4,5,7,8,9,10,11,12,13,14,22,87)
            )
       )
;
```

Sollte dieser SELECT keine Daten zurückliefern, stellt die Aktivierung von EBR kein Problem dar. In der Version 11g stellen oftmals Materialized Views, die auf editionierbare Views zugreifen, ein Problem dar, da Materialized Views in dieser Datenbankversion nicht editioniert werden können. In der Version 12c muss hingegen lediglich angegeben werden, welche Edition der zugrundeliegenden Views verwendet werden soll.

Anschließend wird EBR mittels einem ALTER USER-Statement für das Schema aktiviert:

```
ALTER USER meinschema ENABLE EDITIONS;
```

Erzeugen und Löschen von Editionen

Nach der Aktivierung der Editionierung ist standardmäßig die sogenannte Root-Edition „ora$base“ vorhanden.

Auf dieser Edition basieren alle im Folgenden angelegten Editions, diese Edition lässt sich daher nicht löschen. Werden nun neue Editionen erstellt, so legt die Datenbank diese als Child-Editions unterhalb der Root-Edition an.

Eine neue Edition wird immer als schema-unabhängiges Objekt angelegt, zur Erstellung dient der folgende SQL-Befehl:

```
CREATE EDITION edition1 [AS CHILD OF ora$base];
```

Eine neue Edition erbt zum Zeitpunkt ihrer Erstellung immer die editionierten Objekte ihrer Parent-Edition – diese werden dazu in die neue Edition hineinkopiert.

Zum Löschen einer Edition – **mitsamt ihrer editionierten Objekte** – genügt ein simpler DROP-Befehl:

```
DROP EDITION edition1;
```

Objekte, die nicht editioniert sind, werden mittels dieses Befehls nicht gelöscht.

Wechsel zwischen den Editionen

Zunächst wird die aktuell in der Session aktivierte Edition mittels der Funktion SYS_CONTEXT ermittelt:

```
SELECT sys_context('userenv', 'session_edition_name') FROM dual;
```

Analog dazu ist die aktuell gültige bzw. neueste Edition zu ermitteln, dies geschieht über den Systemkontext „current_edition_name“.

Die Auswahl der zu nutzenden Edition erfolgt auf Session-Ebene und kann dementsprechend komfortabel per ALTER SESSION-Statement erfolgen:

```
ALTER SESSION SET EDITION = edition1;
```

Das Recht zum Zugriff auf die einzelnen Editionen muss dem Nutzer allerdings im Voraus mittels

```
GRANT USE ON EDITION edition1 TO meinschema;
```

zugewiesen werden. Sollte die Verwendung einer Edition für alle Nutzer ermöglicht werden, so wird durch das Grant-Statement dieses Recht einfach dem Schema PUBLIC zugewiesen.

Die standardmäßig zu verwendende Edition wird mittels

```
ALTER DATABASE DEFAULT EDITION = edition1
```

datenbankweit festgelegt.

Erzeugen, Bearbeiten und Löschen von DB-Objekten

Das Erzeugen, Bearbeiten und Löschen von Datenbank-Objekten erfolgt wie gewohnt, hierbei ist lediglich zu beachten, dass in der Datenbanksession die jeweils gewünschte Edition aktiviert ist – es werden ausschließlich die in der für die jeweilige Session aktiven Edition verfügbaren Objekte bearbeitet, erzeugt oder gelöscht.

Die Editionierung erfolgt vollständig transparent, so dass zum Entwicklungszeitpunkt – bis auf die Auswahl der Edition innerhalb der Session – keinerlei Besonderheiten beachtet werden müssen.

Editionierung von Tabellen

Mittels EBR lassen sich Tabellen leider nicht in Editionen verwalten. Hierfür bietet Oracle ein etwas abweichend aufgebautes Konzept an: Online Table Redefinition.

Hierbei werden nicht mehrere Editionen parallel vorgehalten, sondern lediglich die Migration von Tabellenobjekten in eine neue Version unterstützt. Vereinfacht formuliert handelt es sich hierbei um die Erzeugung eines Klons der zu verändernden Tabellen, der nun bearbeitet werden kann und bei Abschluss der Redefinition die Daten der ursprünglichen Tabelle enthält und diese Tabelle ersetzt.

Dieses abweichende Konzept verhindert damit leider eine durchgängige Versionierung der Tabellen mittels der Standard-EBR-Funktionalität, ermöglicht aber dennoch eine komfortable Migration der Tabellen bei der Aktualisierung des Datenbank-Schemas.

Ablauf der Tabellen-Redefinition

Vor der Tabellen-Redefinition muss zunächst überprüft werden, ob eine Redefinition grundsätzlich möglich ist. Hierzu stellt Oracle eine Überprüfungsfunktion zur Verfügung:

```
dbms_redefinition.can_redef_table('meinschema', 'meine_original_tabelle')
```

Anschließend wird eine Tabelle mit der *neuen* Struktur erzeugt, die später die ursprüngliche Tabelle ersetzen wird. Hierfür wird ein normales CREATE TABLE-Statement ohne jegliche Besonderheiten verwendet. Es ist lediglich zu beachten, dass genügend Speicherplatz zur Erzeugung einer Kopie der Ursprungstabelle zur Verfügung steht.

Mittels dbms_redefinition wird nun die Redefinition gestartet und die Inhalte der Ursprungstabelle in die neue Tabelle kopiert:

```
dbms_redefinition.start_redef_table('meinschema', 'meine_original_tabelle',
'meine_neue_tabelle')
```

Seit der Datenbank-Version 10g werden die den Tabellen zugehörigen Objekte wie Constraints, Trigger sowie Indizes automatisch mithilfe der Prozedur COPY_TABLE_DEPENDENTS mitkopiert und bei Bedarf auch kompiliert bzw. aktiviert.

Aus Performance-Gründen sollte vor dem Abschluss der Redefinition noch eine Synchronisation der Tabellen mittels der Prozedur dbms_redefinition.SYNC_INTERIM_TABLE erfolgen.

Mittels FINISH_REDEF_TABLE wird anschließend die Redefinition abgeschlossen, und die neue Tabelle ersetzt die Ursprungstabelle.

Seit Version 11g werden dazugehörige Objekte (mit Ausnahme von Triggern), die durch die Tabellenänderung von keiner strukturellen Änderung betroffen sind, automatisch rekompiliert.

Fazit

Mittels der Funktionalität “Edition Based Redefinition” stellt Oracle eine komfortable Möglichkeit zur transparenten Versionierung von Datenbankobjekten zur Verfügung.

Allerdings lässt sich diese Technik nicht durchgängig für alle Objekttypen anwenden, wodurch sich der Einsatzbereich im Alltag in der Regel auf die Editionierung von Stored Procedures und Views beschränken wird. Mit einigen Einschränkungen ist auch die Migration von Tabellen auf eine neue DDL-Version möglich – allerdings kann hierbei nicht innerhalb der Session zwischen verschiedenen Versionen gewechselt werden.

Insbesondere für den Test neuer Versionen von Stored Procedures und deren Auswirkungen auf andere Datenbankobjekte sowie das Online Application Upgrade stellt die Verwendung von Edition Based Redefinition ein praktisches Feature dar, das sich insbesondere in der Datenbank-Version 12c für die regelmäßige Verwendung eignet.

Kontaktadresse:

Daniel Horwedel
merlin.zwo InfoDesign GmbH & Co. KG
Karmelstraße 9
D-75378 Bad Liebenzell

Telefon: +49 (0) 7052 – 508 98 15
Fax: +49 (0) 7052 – 508 98 30
E-Mail daniel.horwedel@merlin-zwo.de
Internet: www.merlin-zwo.de