

Datenbanksysteme 2, 6. Übung

Stored Procedures mit PL/SQL

Generelles zu dieser Übung

Diese Übung baut auf dem Datenmodell auf, welches Sie in der letzten Übung erstellt hatten. Sie sollten also zunächst sicherstellen, dass dieses Datenmodell noch in Ihrer Datenbank eingerichtet ist. Weiterhin wird die MovieDB von dboracleserv benötigt. Bitte prüfen Sie, ob Sie Zugriff haben. Die können Sie z.B. mit folgendem Befehl machen:

```
SELECT * FROM moviedb.movie;
```

Falls dieser Zugriff nicht funktioniert, wenden Sie sich bitte entweder an Andreas Holitschke oder an einen der Dozenten während der Übungsstunden.

Beachten Sie die Hinweise unten auf dem Übungsblatt. Nutzen Sie die Übungen, um Fragen zu stellen!

Aufgabe 6.1 Kopieren von Daten aus der Movie-DB

Ziel der Aufgabe ist es, Daten aus der Movie-DB in ihr lokales Schema zu übernehmen. Sie sollen dazu PL/SQL-Code implementieren, so dass Sie eine Stored Procedure zur Verfügung haben, mit der Sie jederzeit Daten aus der Movie-DB in Ihr Schema kopieren können.

Die Methode soll folgendermaßen aussehen:

```
CREATE OR REPLACE
PROCEDURE dbs2_import_movies(p_title VARCHAR2,
                             p_type CHAR,
                             p_year NUMBER)
AUTHID CURRENT_USER IS
    -- TODO: hier kommt Ihr Code
END;
```

Im Detail sollen folgende Schritte implementiert werden:

1. Alle in MovieDB vorkommenden Genres werden in Ihre Genre-Tabelle kopiert, falls Sie noch nicht vorhanden sind. Falls in Ihrer Tabelle bereits zusätzliche Genres eingepflegt sind, bleiben diese Genres bestehen. Genres werden anhand des Namens identifiziert. Für die neu eingefügten Genres müssen sie neue IDs aus einer Sequenz vergeben.
2. Über die Parameter werden die Filme identifiziert, die übertragen werden sollen: `p_title` ist ein String, über den Filme mit passendem Titel gesucht werden. Es wird nach Teilstrings gesucht und Groß/Kleinschrift wird ignoriert. Mit `p_type` kann die Suche auf einen Typ eingeschränkt werden; mit `p_year` wird die Suche auf ein Jahr eingeschränkt. Beide Parameter können auch NULL sein, dies bedeutet, dass die Kriterien ignoriert werden sollen.
3. Es werden maximal 30 Filme übertragen, um zu verhindern, dass bei zu offenen Suchkriterien große Datenmengen übertragen werden.

4. Es werden alle Informationen aus der Movie-Tabelle für die ausgewählten Filme übertragen (Id, Titel, Typ, Jahr). Fügen Sie die Filme mit ihrer Original-ID ein; wir treffen die Annahme, dass diese IDs noch nicht vergeben sind (sonst wird es zu kompliziert).
5. Dann werden alle Personen, die Schauspieler in den in ihrem Datenmodell eingetragenen Filmen spielen, in ihre Personen-Tabelle übertragen. Falls eine/r der Schauspieler/innen schon existiert, wird er/sie nicht erneut übertragen, d.h. Sie müssen prüfen, welche der benötigten Schauspieler/innen noch nicht in der Datenbank stehen.
6. Übertragen Sie jetzt alle fehlenden Charaktere aus den in ihrer lokalen Datenbank eingetragenen Filmen.
7. Verfahren Sie entsprechend mit den Genres.
8. Zum Abschluss müssen Sie noch die Sequenz, aus der neue IDs gezogen werden, so einstellen, dass die GUI aus dem letzten Aufgabenblatt weiterhin arbeitet. Dazu müssen Sie die höchste vergebene ID ermitteln und die Sequenz höher einstellen.

Hinweise:

- Verwenden Sie die Original-IDs der MovieDB und vergeben Sie keine neuen IDs für die Objekte (Filme, Schauspieler, etc.). Wir treffen überall die Annahme, dass das nicht mit lokal vergebenen IDs kollidiert.
- Alle Schritte lassen sich über SQL-Statements der Form "INSERT INTO ... SELECT ..." lösen, d.h. Sie erstellen für jeden Schritt ein SELECT-Statement, welches die neuen Datensätze zusammenstellt. Dieses muss ggf. über ein Konstrukt wie z.B. "WHERE ... NOT IN (SELECT ...)" feststellen, welche Datensätze nicht schon in der Zieltabelle vorhanden sind, um ein erneutes Einfügen zu verhindern. Es sind also keine Schleifen nötig!
- Eine Sequenz mit einem Startwert können Sie aus PL/SQL heraus folgendermaßen neu einrichten (dynamisches SQL):

```
EXECUTE IMMEDIATE 'DROP SEQUENCE dbs2_ids';  
EXECUTE IMMEDIATE 'CREATE SEQUENCE dbs2_ids START WITH ' || nextid;
```

Testen Sie Ihre Prozedur mit verschiedenen Aufrufen, z.B.:

```
EXECUTE dbs2_import_movies('Star Wars', 'C', NULL);
```

oder

```
EXECUTE dbs2_import_movies('Star Wars', NULL, 1977);
```