

Datenbankpraktikum
Wintersemester 2012/13
Übungsblatt 3

Abgabe über *UniWorX* bis **Mittwoch, den 05.12.2012, 9:00 Uhr** möglich.

Aufgabe 3-1 SQL in Java

Entwerfen und implementieren Sie ein Programm in Java mit grafischer Oberfläche, das (in einer Schleife) nach einer Unterkunfts-ID fragt (Schema **reise_tables_2012**) und

- eine Liste aller Buchungen zu dieser Unterkunft ausgibt (mit Buchungs-ID, Namen, Personenanzahl und Zeitraum) und
- eine Fehlermeldung ausgibt, falls die ID nicht existiert (dabei die Schleife aber nicht beendet).

Beim Aufbau der Verbindung zum Oracle-Server soll das Eintippen des Passwortes am Bildschirm nicht sichtbar sein. Achten Sie darauf, dass das Programm ordnungsgemäß beendet werden kann.

Aufgabe 3-2 Java gegen PL/SQL

Schreiben Sie zwei Programme, welche jeweils alle überbuchten Unterkünfte finden. Implementieren Sie eines in Java, das andere in PL/SQL. Ausgegeben werden soll für jeden Tag an dem eine Unterkunft überbucht ist: die ID, der Name und die Maximalkapazität der Unterkunft sowie die Anzahl der gebuchten Plätze. Vergleichen Sie die Laufzeit der beiden Programme.

Aufgabe 3-3 Trigger

Für das Buchungssystem (Schema **reise_tables_2012**) gilt folgende Integritätsbedingung:

Für eine Unterkunft dürfen an keinem Tag Buchungen für mehr Personen vorliegen als die maximale Kapazität der Unterkunft beträgt.

Überlegen Sie, welche Operationen (INSERT, UPDATE, DELETE) auf welchen Relationen zu einer Verletzung obiger Integritätsbedingung führen können. Entwerfen und implementieren Sie Trigger, die die Einhaltung obiger Integritätsbedingung überwachen. Programmieren Sie dazu ein Paket **trigger_functions**, welches die eigentliche Überprüfung über boolesche Funktionen enthält, so dass die Trigger nur noch den Aufruf dieser Funktionen und die dazugehörige Fehlererzeugung enthalten.

Hinweis: Um die Funktionalität eines Triggers testen zu können, erstellen Sie zunächst eine lokale Kopie der relevanten Relationen, um die notwendigen Update-Privilegien zu besitzen. Sämtliche Trigger sollen dann auf diesen lokalen Kopien arbeiten. Für diese Aufgabe kann die Verwendung der Lösung aus Aufgabe 2 sinnvoll sein.

Aufgabe 3-4 Dynamic SQL

Schreiben Sie ein PL/SQL-Package 'projection', das eine Prozedur *createLandTables*(land VARCHAR2) enthält. *createLandTables* soll für das angegebene Land *land* eine Tabelle nach folgendem Schema anlegen:

```
CREATE TABLE UNTERKUENFTE_<land> (  
    u_id NUMBER,  
    spalte VARCHAR2(255),  
    wert VARCHAR2(255)  
)
```

Zum Beispiel würde aus:

Tabelle 1: UNTERKUNFT

u_id	u_name	kapazitaet	liegt_in
...
30007	Kakaobaum Camping	602	399 (<i>Jena</i>)
...
30017	Kokkelstrauch Haus	9	578 (<i>New_York</i>)
30018	Engelwurz Hotel	151	308 (<i>Essen</i>)
...

für 'Deutschland' folgende Tabelle entstehen:

Tabelle 2: UNTERKUENFTE_Deutschland

u_id	spalte	wert
30007	u_name	Kakaobaum Camping
30007	kapazitaet	602
30007	liegt_in	399
30007	toiletten	7
30018	u_name	Engelwurz Hotel
...
30018	sterne	4
...

Anschließend soll die Prozedur für jede Unterkunft aus der Tabelle **reise_tables.2012.unterkunft**, die in diesem Land liegt, alle Felder in die erstellte Tabelle einfügen. Gehen Sie dabei für jede Unterkunft über jede Spalte und fügen Sie für jedes Wert-Spalten-Paar einen Datensatz in Ihre neue Tabelle ein. Vergessen Sie nicht die Spalten der abgeleiteten Relationen (HOTEL, CAMPINGPLATZ, FERIENWOHNUNG) ebenfalls als Wert-Spalten-Paar einzufügen. Gestalten Sie Ihre Prozedur so dynamisch, dass sie auch noch alle Werte in die neue Tabelle übernimmt falls neue Spalten in die Ausgangstabellen hinzugefügt werden.

Hinweis: Um herauszubekommen, welche Spalten beispielsweise in der Tabelle **reise_tables.2012.unterkunft** vorkommen, können Sie die Data-Dictionary-View ALL_TAB_COLUMNS verwenden.

```
SELECT column_name FROM all_tab_columns  
WHERE owner = 'REISE_TABLES_2012' AND table_name = 'UNTERKUNFT'
```

Bitte speichern Sie alle Lösungen so ab, dass sie bei der Abnahme getestet und besprochen werden können.

Die Abnahme des Übungsblattes findet in der Woche vom 03.12.2012 statt.