

Datenbankpraktikum
Wintersemester 2010/11
Übungsblatt 2

Aufgabe 2-1 SQL-Anfragen

Die Datenbank **dbprakt** enthält im Schema **reise_tables** eine Umsetzung des Buchungssystems gemäß dem E/R-Diagramm von Arbeitsblatt 1. (Einige Objekte und Beziehungen wurden vereinfacht, z.B. werden Kunden in **Buchung** und nicht in einer extra Tabelle gehalten). Die Relationen sind mit Daten gefüllt. Machen Sie sich mit Hilfe des Oracle-Data-Dictionary (z.B. Relation **all_tables**) und des **SQL*PLUS**-Befehls **describe** oder dem **SQL-Developer** mit den Definitionen der Relationen des Owners **reise_tables** vertraut und formulieren Sie dann folgende Anfragen:

- (a) Welche Städte haben mehr als 10 Millionen Einwohner?
- (b) Alle Städte die mit *Pa* beginnen.
- (c) Die Anzahl aller Unterkünfte in *Paris* und deren Gesamtkapazität.
- (d) Die Namen aller Städte die an einem See liegen zusammen mit den Namen dieser Seen.
- (e) Welche Unterkünfte haben für den Februar 2009 mehr als 2 Buchungen? (Hinweis: die *to_date*-Funktion könnte hier nützlich sein)
- (f) Alle Campingplätze bei denen (bei voller Buchung) sich höchstens 50 Gäste eine Toilette teilen müssen. (Hinweis: Versuchen Sie Division durch 0 durch Umgehen oder Abfangen dieses Falls zu vermeiden)
- (g) Bestimmen Sie die ID, den Namen und die Größe aller Ferienwohnungen mit überdurchschnittlicher Quadratmeterzahl, die in einer Hauptstadt in Europa liegen. Der Name der Stadt soll zusätzlich mit ausgegeben werden.

Aufgabe 2-2 View-Definition

Städte können in Landteilen liegen durch die keiner, einer oder mehrere Flüsse fließen. Erstellen Sie einen View **sgs** mit Attributen **st_id**, **st_name**, **fluesse** über den Relationen **stadt**, **landteil**, **gewaesser**, **fluss** der folgende Bedingungen erfüllt:

- je Tupel in **stadt** ein Ergebnistupel im View **sgs**
- Eintrag im Attribut **fluesse**:

fluesse = 'keine',	(falls die Stadt in einem Landteil ohne Flüsse liegt)
fluesse = gewaesser.g_name ,	(falls die Stadt in einem Landteil mit genau einem Fluss liegt)
fluesse = gewaesser.g_name '...!',	(wobei gewaesser.g_name der alphabetisch erste Fluss ist, falls die Stadt in einem Landteil mit mehreren Flüssen liegt)

Hinweise: Die drei Fälle können durch einzelne Select-Anweisungen gelöst werden, die mit Hilfe der Operation **union** in der View-Definition zusammengefasst werden können. Der alphabetisch erste Flussname aus einer Liste kann z.B. mit der Aggregatsfunktion **min** ermittelt werden.

Aufgabe 2-3 PL-SQL (Datumsfunktion)

- (a) Schreiben Sie eine PL/SQL-Funktion **day_calc**, die zu einem gegebenen Datum den Wochentag als Zeichenkette zurückliefert. Die Berechnung des Tages soll dabei ohne Benutzen von Hilfsrelationen auskommen. Sehen Sie auch eine Fehlerbehandlung vor, falls der Eingabe-String kein gültiges Datum repräsentiert. Achten Sie auf Jahr-2000-Fähigkeit!

Hinweis: Folgende SQL-Funktionen könnten nützlich sein: **to_char**, **to_date**.

- (b) Schreiben Sie einen oder ggf. mehrere SQL-Befehle, die zu einer gegebenen Unterkunft (ID oder Name) alle aktuell bestehenden Buchungen findet. Für jede Buchung soll neben den bestehenden Attributen der Name der Unterkunft mit ausgegeben werden. Für die Attribute VON (erster Tag der Buchung) und BIS (letzter Tag) soll zusätzlich der Wochentag (aus a)) ausgegeben werden.

Aufgabe 2-4 PL-SQL (Packages und Funktionen)

Im Schema **reise_tables** finden Sie eine Tabelle ZAHLEN (ID, WERT). Schreiben Sie ein Package calculate, dass folgende Funktion und Prozedur implementiert:

- *function median return number as ..*

Die Funktion soll den Median, der in ZAHLEN abgespeicherten Werte (WERT) berechnen. Der Median einer Multi-Menge von Zahlen ist das Element, das nach dem Sortieren an der mittleren Stelle des Ergebnisses zu finden ist. Falls die Multi-Menge eine gerade Kardinalität besitzt, ist das größere der beiden mittleren Elemente zurückzugeben. Zum Beispiel: Der Median von 3, 1, 8, 7, 5 ist 5 und der Median von 3, 1, 10, 6, 7, 1 ist 6.

- *procedure exp_std as ..*

Die Prozedur soll Erwartungswert und Standardabweichung aller in ZAHLEN abgespeicherten Werte (WERT) ausgeben.

Aufgabe 2-5 PL-SQL (Iteration)

Schreiben Sie eine SQL*PLUS-Batch-Datei **weg_find.sql**, die nach zwei Land-ID's fragt und berechnet, ob ein Landweg zwischen den beiden Ländern existiert. (Optional: Falls ja, geben sie die Länder aus, über die der kürzeste weg verläuft.)

Hinweise:

Eventuell benötigen Sie eine Hilfsrelation (siehe Pseudocode im Skript).

Ein Beispiel für Ein- und Ausgabe in PL/SQL:

```
SET SERVEROUTPUT ON
accept eingabe prompt 'Zahl eingeben'
DECLARE
BEGIN
    dbms_output.put_line('&eingabe');
END;
```

Bitte speichern Sie alle Lösungen so ab, dass Sie bei der Abnahme getestet und besprochen werden können. Bringen Sie SQL-Statements und PL/SQL-Funktionen dazu gegebenenfalls in digitaler Form und ausgedruckt zum Abnahmetermin mit.

Die Abnahme des Übungsblattes findet am 16.11.2009 statt.

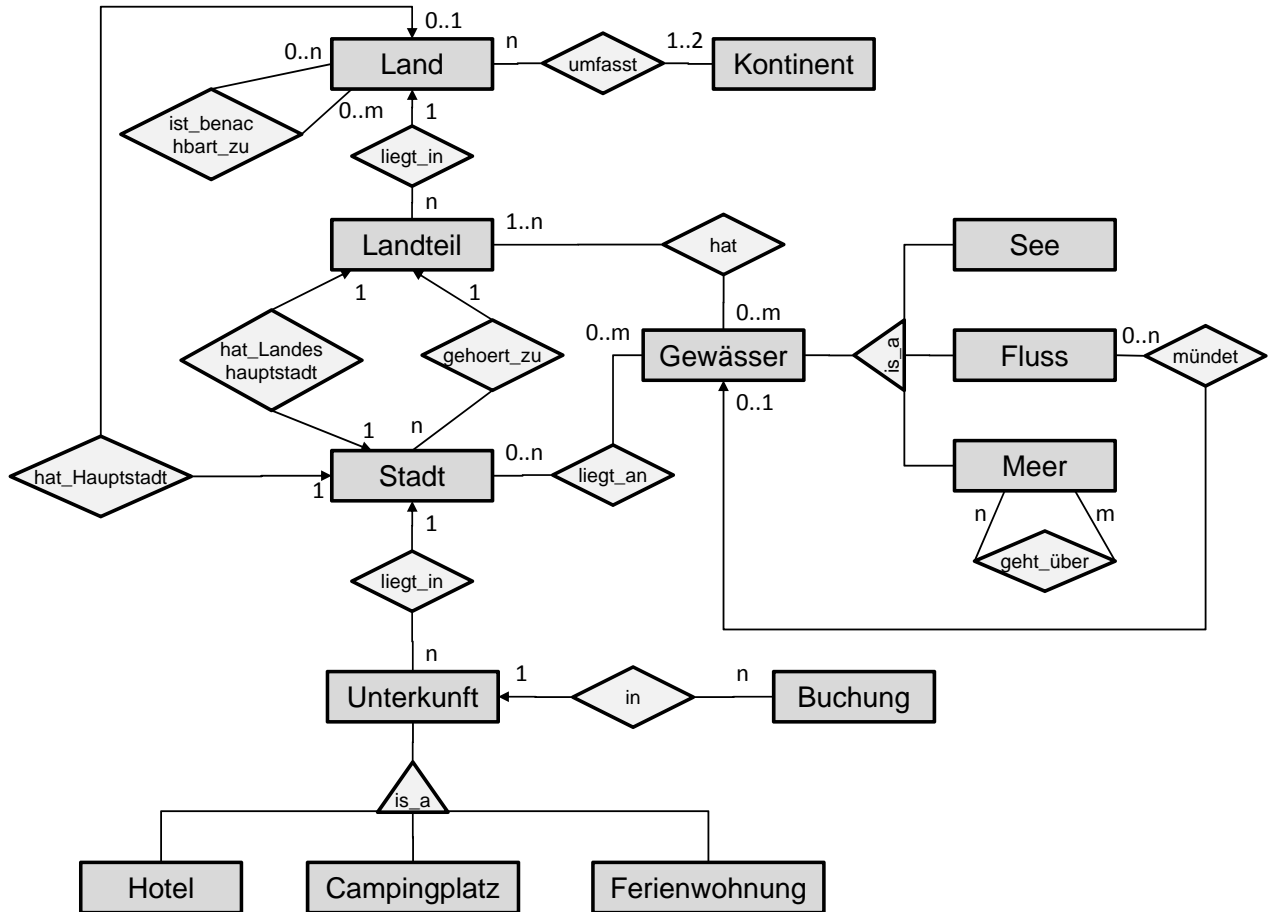


Abbildung 1: Arbeitsblatt 1