

Datenbankprogrammierung (Oracle)

Aufgabe 1: SQL

Gegeben sei die folgende Relation:

```
messwerte(datum, wert_1, wert_2, wert_3, wert_4, wert_5, wert_6)
```

Die Spalten `wert_1` bis `wert_6` enthalten numerische Messwerte für den in `datum` angegebenen Tag. Bearbeiten Sie die folgenden Aufgaben:

- Erstellen Sie zunächst ihre eigene Relation `messwerte` und füllen Sie diese mit geeigneten Daten. `datum` ist Primärschlüssel.
- Geben Sie für jeden Tag den höchsten und den niedrigsten Messwert aus.
- Geben Sie für jeden Tag `wert_1` aus und die Differenz von `wert_1` und dem Vortageswert von `wert_1`.

Aufgabe 2: SQL: Formatierung

Von Übungsblatt 4 kennen Sie bereits die Tabellen eines Gebrauchtwagenhändlers. Bearbeiten Sie nun folgende Aufgaben.

- Geben Sie die 5 teuersten Autos aus. Verwenden Sie `rownum`.
- Machen Sie sich mit `fetch only` vertraut. Infos gibt es z.B. unter <https://oracle-base.com/articles/12c/row-limiting-clause-for-top-n-queries-12crl>
Geben Sie anschließend mit `fetch only` die 5 teuersten Autos aus.
- Geben Sie für jedes Auto aus `USED_CARS_SOLD` die Nationalität des Herstellers aus, aufsteigend sortiert nach ID. Die Nationalitäten der einzelnen Automarken erfahren Sie im Internet, z.B. bei <http://www.wikipedia.de>. Der Beginn der Ergebnisrelation soll wie folgt aussehen:

CARS	make	country	color	price	age
	BMW	Germany	black	6482	8
	Fiat	Italy	blue	5444	12
	BMW	Germany	white	7041	5
	Peugeot	France	green	4516	11

Aufgabe 3: DAX (Subqueries, Oracle-SQL)

In der Tabelle *DAX* finden Sie alle Tageskurse des *Deutsche Aktienindexes* von Juli 1988 bis Juni 2004. Beantworten Sie die folgenden Aufgaben:

- An wie vielen Tagen wurde ein Kurs veröffentlicht?
- Was war der Höchststand des Dax? An welchem Tag wurde dieser erreicht?
- Geben Sie die 10 Tage mit den 10 höchsten Dax-Ständen aus.
- Geben Sie den durchschnittlichen Kurs jedes Jahres aus. Ergänzen Sie die Abfrage, so dass zusätzlich ausgegeben wird, an wie vielen Tagen der Kurs höher war und an wie vielen niedriger.
- Es soll nun für jeden Tag ausgegeben werden, ob der DAX im Vergleich zum letzten Kurs gestiegen oder gefallen ist. Betrachten Sie dazu die Funktion LAG.

LAG (<SPALTE>, <ANZAHL>) OVER (ORDER BY <SORTIERUNG>)

Diese Funktion bietet Zugriff auf Attributwerte vorheriger Tupel. Die Parameter haben dabei folgende Semantik:

- anzuzeigende <SPALTE>
- <ANZAHL> der Zeilen, die zurückgeblickt wird
- <SORTIERUNG> des Rückblicks

(Die Funktion LEAD bietet dieselbe Funktionalität für einen Blick auf nachfolgende Tupel.)

- Geben Sie die 10 Tage aus, an denen der Dax am stärksten gestiegen ist und mit einer weiteren Anfrage die 10 Tage, an denen er am stärksten gefallen ist.

Aufgabe 4: Berichte

Die Tabelle *BESTELLUNGEN* von Übungsblatt 2 kennen Sie bereits. Erstellen Sie eine nach Monaten gegliederte Umsatzaufstellung für das Jahr 2008. Das Ergebnis soll die Tabellenform rechts aufweisen.

Eine mögliche Strategie zur Lösung der Aufgabe könnte wie folgt sein:

- Sie benötigen eine Zeile für jeden Monat. Überlegen Sie sich eine Abfrage, die (unabhängig von Tabelle *BESTELLUNGEN*) ein Tupel für jeden Monat des Jahres 2008 zurückliefert.
- Für jeden Monat müssen Sie nun die vorhandenen Umsätze zusammenzählen bzw. 0 ausgeben.
- Eine passende Formatierung des Datums erreichen Sie mit Hilfe der Funktion
`TO_CHAR(datum, 'YYYY-MM')`.

UMSATZ	Monat	Betrag
	2008-01	0
	2008-02	0
	2008-03	0
	2008-04	500
	2008-05	0
	2008-06	700
	2008-07	500
	2008-08	700
	2008-09	800
	2008-10	1400
	2008-11	0
	2008-12	0