

Aufgabe 2: noch mal Kaufhaus...

Gegeben ist wiederum die Kaufhaus-Datenbank (Tabellen siehe oben). Formulieren Sie nachfolgende Anfragen in SQL!

- (a) Welche Artikel (Artikelnummer, Abteilungsname) werden in den Abteilungen angeboten? Die Ausgabe soll absteigend nach der Artikelnummer sortiert werden. Bei gleicher Artikelnummer sollen die betroffenen Abteilungen alphabetisch aufgelistet werden.

richtig:

Viel zu kompliziert. Geht ohne Joins

```
1 SELECT Artikel.ArtNr AS Artikelnummer, Abteilungsname
2 FROM Artikel, Bestand WHERE
3     Bestand.ArtNr = Artikel.ArtNr
4 ORDER BY Artikel.ArtNr DESC, Abteilungsname ASC;
```

Musterlösung:

```
1 SELECT ArtNr, Abteilungsname
2 FROM Bestand
3 ORDER BY ArtNr DESC, Abteilungsname;
```

- (b) Wie viele verschiedene Waren werden in der Lebensmittelabteilung verkauft?

Musterlösung:

```
1 SELECT COUNT(*)
2 FROM Bestand
3 WHERE Abteilungsname = 'Lebensmittel';
```

- (c) Wie viele verschiedene Waren werden in den einzelnen Abteilungen verkauft?

Musterlösung:

```
1 SELECT Abteilungsname, COUNT(*)
2 FROM Bestand
3 GROUP BY Abteilungsname;
```

- (d) Wie viel kostet der billigste, wie viel der teuerste Artikel?

Musterlösung:

```
1 SELECT MIN(Verkaufspreis), MAX(Verkaufspreis)
2 FROM Artikel;
```

- (e) Gib die Namen aller Abteilungen aus, deren Gesamtinventar an Artikel kleiner als 100 ist!

Musterlösung:

```

1  SELECT Abteilungsname
2  FROM Bestand
3  GROUP BY Abteilungsname
4  HAVING COUNT(Vorrat) < 100;

```

- (f) Gesucht sind Bezeichnung und Verkaufspreis aller in der Datenbank gespeicherten Artikel. Die Ausgabe soll alphabetisch aufgelistet werden. Bei gleicher Bezeichnung sollen die teureren Artikel zuerst aufgelistet werden.

Musterlösung:

```

1  SELECT Bezeichnung, Verkaufspreis
2  FROM Artikel
3  ORDER BY Bezeichnung, Verkaufspreis DESC;

```

- (g) Gib für alle Artikel, von denen (unabhängig von der Abteilung) noch mindestens 130 Exemplare vorrätig sind, die Artikelnummer und den aktuellen Vorrat aus!

falsch:

Derselbe Artikel kann in mehreren Abteilungen angeboten werden.

```

1  SELECT ArtNr, Vorrat
2  FROM Bestand
3  WHERE Vorrat >= 130;

```

Musterlösung:

```

1  SELECT ArtNr, SUM(Vorrat)
2  FROM Bestand
3  GROUP BY ArtNr
4  HAVING SUM(Vorrat) >= 130;

```