SQL-Joins -- Grundlagen

Wie werden die Tabellen (und Entitäten) jetzt aber mit SQL Abfragen verbunden, so das die gewünschten „zusammengesetzten“ Informationen als Ergebnis der Abfrage vorliegen?

Abfragen über mehrere Tabellen

Bei Abfragen aus mehreren Tabellen wird ohne weitere Bedingung die Besziehung zwischen den Entitäten nicht berücksichtigt - es werden einfach alle Kombinationen aller Datensätze der beiden Tabellen gebildet und ausgegeben[1)]. Die meisten der so aggregierten Datensätze sind sinnfrei.

Beipiel: Die Tabelle `lehrer` hat 6 Datensätze, die Tabelle `schueler` 12. Die Abfrage

```
SELECT * FROM `lehrer`,`schueler`
```

liefert als Ergebnis eine Tabelle mit 72 Datensätzen:

WHERE schueler.klassenlehrer_id=lehrer.lehrer_id

Aufbau

Eine Klausel für SQL-Joins ist im Wesentlichen folgendermaßen aufgebaut:

```
SELECT [DISTINCT] { spalten | * }
FROM tabelle1, tabelle2, ...
WHERE tabelle1.spalte1=tabelle2.spalte1,
      tabelle1.spalte2=tabelle2.spalte2, ...
```

Um die Ausdrücke abzukürzen, kann man Tabellen mit mit dem Schlüsselwort AS mit Aliasen bezeichnen:

```
SELECT [DISTINCT] { spalten | * }
FROM tabelle1 AS t1, tabelle2 AS t2, ...
WHERE t1.spalte1=t2.spalte1,t1.spalte2=t2.spalte2, ...
```

Das Ergebnis eines Joins ist eine Tabelle mit allen Spalten, die in der SQL-Klausuel angegeben wurden. Die Spaltenbezeichnung muss eindeutig sein. Enthalten die Tabellen `tabelle1` und `tabelle2` gleichnamige Spalten wie `id`, muss die Spalte mit `tabelle1.id` eindeutig benannt sein.

Wird keine WHERE-Bedingung angegeben, wird das Kreuzprodukt der Tabellen gebildet, d.h. jeder Datensatz einer Tabelle wird mit jedem der anderen Tabelle(n) kombiniert.

Beispiele

Ausgabe aller Kunden mit den Nummern ihrer bestellten Artikel

```
SELECT k.KNr, k.KVorname, k.KNachname, b.ANr
FROM   Kunden AS k, Bestellt AS b
WHERE  k.KNr=b.KNr
```

Ausgabe aller Kunden mit den Namen ihrer bestellten Artikel

```
SELECT k.KNr, k.KVorname, k.KNachname, a.Name
FROM   Kunden AS k, Bestellt AS b, Artikel AS a
WHERE  k.KNr=b.KNr AND b.ANr=a.ANr
```

Aufgaben

(A1)

Sofern noch nicht geschehen: Importiere die Datenbank webshop in deinen mysql-Datenbankbereich.

Löse die folgenden Aufgaben innerhalb der Datenbank `webshop`, indem du das SQL-Abfragefenster verwendest. Speichere deine Lösungen in einer Textdatei/deinem Infoheft. Vermeide die Ausgabe von Duplikaten.

1. Gib die Liste mit den Namen aller jemals bestellen Artikel mitsamt ihres Bestelldatums aus.
2. Gib die Liste aller Kundennummern zusammen mit den Namen der Artikel, die unter dieser Nummer bestellt wurden, aus.
3. Gib alle Zeitpunkte aus, zu denen Thomas Tischler Artikel im Webshop bestellt hat.
4. Gib Sie die Namen aller Artikel aus, die Maximilian Mayer bestellt hat.
5. Gib die Namen aller Kunden aus, die mindestens einmal einen Artikel bestellt haben, absteigend sortiert nach dem Datum der letzten Bestellung.
6. Gib pro Kunde aus, wie viele Artikel er insgesamt bestellt hat. Die Liste soll absteigend sortiert nach der Anzahl der Bestellungen ausgegeben werden.
7. Gib die Namen der Artikel aus mitsamt der Information, wie oft sie insgesamt bestellt wurden. Die Liste soll absteigend sortiert nach der Anzahl der Bestellungen ausgegeben werden.
8. Gib pro Kunde aus, wie viel Geld er insgesamt ausgegeben hat.
9. Gib die Nummern und Namen der Kundenpaare aus, die den gleichen Artikel bestellt haben.
10. Zusatzaufgabe: Gib die Liste aller Kunden aus, die noch nie einen Artikel im Webshop bestellt haben.[2)]

[1)]

Das kartesische Produkt der Tabellen

[2)]

Tipp: `WHERE attribute NOT IN` (SQL-Klausel).

From:
https://wiki.qg-moessingen.de/ - **QG Wiki**

Permanent link:
https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:datenbanken:joins:start?rev=1603301411

Last update: **21.10.2020 19:30**