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:

Zeige Datensätze 0 - 24 (72 insgesamt, Die Abfrage dauerte 0.0006 Sekunden.)

```
select * from schueler,lehrer
```

1 > >> | Alles anzeigen | Anzahl der Datensätze: 25 Zeilen filtern: Diese Tabelle

+ Optionen

name	vorname	klasse	telefon	name	vorname	klehrer_in	fach
Margarine	Carl	8a	7625653	Einstein	Albert	7a	Physik
Margarine	Carl	8a	7625653	Newton	Isaac	6a	Mathe
Margarine	Carl	8a	7625653	Lovelace	Ada	6a	Informatik
Margarine	Carl	8a	7625653	Noether	Emmi	6b	Mathe
Margarine	Carl	8a	7625653	Gauss	Carl-Friedrich	6a	Mathe
Margarine	Carl	8a	7625653	Freud	Sigmund	6a	Psychologie
Schockokeks	Marie	7a	87216625	Einstein	Albert	7a	Physik
Schockokeks	Marie	7a	87216625	Newton	Isaac	6a	Mathe
Schockokeks	Marie	7a	87216625	Lovelace	Ada	6a	Informatik
Schockokeks	Marie	7a	87216625	Noether	Emmi	6b	Mathe
Schockokeks	Marie	7a	87216625	Gauss	Carl-Friedrich	6a	Mathe
Schockokeks	Marie	7a	87216625	Freud	Sigmund	6a	Psychologie
Erdbeermarmelade	Alan	7b	454261	Einstein	Albert	7a	Physik
Erdbeermarmelade	Alan	7b	454261	Newton	Isaac	6a	Mathe
Erdbeermarmelade	Alan	7b	454261	Lovelace	Ada	6a	Informatik
Erdbeermarmelade	Alan	7b	454261	Noether	Emmi	6b	Mathe
Erdbeermarmelade	Alan	7b	454261	Gauss	Carl-Friedrich	6a	Mathe
Erdbeermarmelade	Alan	7b	454261	Freud	Sigmund	6a	Psychologie
Schinkenwurst	Hermann	6a	872662625	Einstein	Albert	7a	Physik

Sinnvolle Datensätze bilden

Sinnvoll sind in diesem Fall nur die Datensätze, bei denen in der Schülertabelle die `klassenlehrer_id` des zugehörigen Klassenlehrers mit der `lehrer_id` in der `lehrer` Tabelle übereinstimmt.

```
SELECT * FROM `lehrer`,`schueler`
WHERE schueler.klassenlehrer_id=lehrer.lehrer_id
```

Wenn man Felder verschiedener Tabellen in einer Abfrage verwendet, muss man diese in der Form `Tabelle.Attribut` angeben. Das Ergebnis sieht jetzt so aus:

Zeige Datensätze 0 - 11 (12 insgesamt, Die Abfrage dauerte 0.0005 Sekunden.)

```
SELECT * FROM `lehrer`,`schueler` WHERE schueler.klassenlehrer_id=lehrer.lehrer_id
```

Mes

Alles anzeigen | Anzahl der Datensätze: 25 Zeilen filtern: Diese Tabelle durchsuchen

+ Optionen

lehrer_id	name	vorname	klehrer_in	fach	schueler_id	name	vorname	klasse	telefon	klassenlehrer_id
3	Lovelace	Ada	6a	Informatik	1	Margarine	Carl	8a	7625653	3
2	Newton	Isaac	6a	Mathe	2	Schockokeks	Marie	7a	87216625	2
2	Newton	Isaac	6a	Mathe	3	Erdbeermarmelade	Alan	7b	454261	2
1	Einstein	Albert	7a	Physik	4	Schinkenwurst	Hermann	6a	872662625	1
1	Einstein	Albert	7a	Physik	5	Gummibärchen	Karl	7a	78265635	1
2	Newton	Isaac	6a	Mathe	6	Eisbein	Hildegard	7a	7288393	2
3	Lovelace	Ada	6a	Informatik	7	Kartoffelchips	Gabriele	6a	928893484	3
4	Noether	Emmi	6b	Mathe	8	Apfelsaft	Lieschen	7b	991823764	4
4	Noether	Emmi	6b	Mathe	9	Aprikosenjoghurt	Klaus-Gustav	7b	9288374	4
4	Noether	Emmi	6b	Mathe	10	Kaffepulver	Ingeborg	6a	209384	4
2	Newton	Isaac	6a	Mathe	11	Schokocreme	Lisa-Marie	6a	923867346	2
2	Newton	Isaac	6a	Mathe	12	Cornfkakes	Crunchie	7b	92848724t36	2

Das macht Sinn, das ist eine Liste aller Lehrer und Schüler, und zwar passend, so das die Schüler bei Ihren Klassenlehrern mitgelistet werden.

Allgemeiner 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
```

(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.

Gib pro Kunde aus, wie viele Artikel er insgesamt bestellt hat. Die Liste soll absteigend sortiert nach der Anzahl der Bestellungen ausgegeben werden.

Lösungsvorschlag

```
SELECT Knachname,SUM(BAnzahl),COUNT(*) FROM kunden,bestellt
```

```
WHERE kunden.KNr = bestellt.KNr
GROUP BY KNachname
ORDER BY COUNT(*) ASC;
```

1. 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.
2. Gib pro Kunde aus, wie viel Geld er insgesamt ausgegeben hat.
3. Gib die Nummern und Namen der Kundenpaare aus, die den gleichen Artikel bestellt haben.
4. Zusatzaufgabe: Gib die Liste aller Kunden aus, die noch nie einen Artikel im Webshop bestellt haben.[2)]

Lösungen

Zu den Lösungen

Material

04-sql-joins.odp	1.4 MiB	26.11.2020 08:39
04-sql-joins.pdf	681.6 KiB	26.11.2020 08:39
auswahl_080.png	254.2 KiB	21.10.2020 17:12
auswahl_081.png	120.7 KiB	21.10.2020 17:29
auswahl_082.png	98.3 KiB	21.10.2020 17:28
schule500_sus_keine_keys.zip	10.5 KiB	26.11.2020 08:38
schule500_sus_keys.zip	11.8 KiB	26.11.2020 08:38
schule_klein.sql.zip	1.1 KiB	26.11.2020 08:38

1)
Das kartesische Produkt der Tabellen

2)
Tipp: `WHERE attribute NOT IN` (SQL-Klausel).

From:
https://info-bw.de/ -

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:datenbanken:joins:start?rev=1643269214

Last update: **27.01.2022 07:40**