

## SQL Blatt 2- Beispiel: Kontoführung

Ausgangspunkt ist wieder das aus den vorherigen Blättern bekannte Beispiel der Kontoführung mit folgendem relationalen Schema:

Kategorie: (Bezeichnung:string, Klassifikationskriterium:string)  
Kunde: (Vorname:string, Name:string, Gebdat:date, Bezeichnung:string)  
Konto: (Kontonr: integer, Betrag:numeric, Gebühr:numeric)  
Festgeldkonto: (Kontonr: integer, Fälligkeit:date, Zinssatz: numeric)  
Girokonto: (Kontonr: integer, Dispolimit:numeric, Telebanking:boolean)  
Kontobewegung: (Datum:date, Vorname:string, Name:string, Gebdat:date, Kontonr:integer, Betrag: numeric, Bewegungsart: string)  
Kontoauszug: (Erstelldat:date, Kontonr:integer, Startdat:date, Gesamt:numeric)  
besitzt: (Vorname:string, Name:string, Gebdat:date, Kontonr:integer)  
listet: (Erstelldat:date, Kontonr:integer, Datum:date, Vorname:string, Name:string, Gebdat:date)

### Aufgabe 1: SQL - Gruppierung

Formulieren Sie folgende Anfragen unter Verwendung der group-by-Klausel:

1. Das Gesamtguthaben eines jeden Kunden.

Ergebnis: Kundenvorname, -nachname, Geburtsdatum, Gesamtguthaben

```
select    b.Name, b.Vorname, b.Gebdat, sum(k.Guthaben)
from      besitzt b, Konto k
where     b.Kontonr = k.Kontonr
group by b.Name, b.Vorname, b.Gebdat
```

2. Kunden, die im November 2003 mehr als 2000 Euro abgehoben haben

Ergebnis: Kundenvorname, -nachname, Geburtsdatum, abgehobener Betrag

```
select    b.Name, b.Vorname, b.Gebdat, sum(kb.Betrag)
from      Kontobewegung kb, besitzt b
where     kb.Datum between '2003-11-01' and '2003-11-30' and
           Bewegungsart = 'Auszahlung' and b.Kontonr = kb.Kontonr
group by b.Name, b.Vorname, b.Gebdat
having    sum(kb.Betrag) >2000
```

## Aufgabe 2: Unteranfragen

Formulieren Sie folgende Anfragen unter Verwendung von subqueries:

1. Kunden der Kategorie U18 (unter 18 Jahre), die ihr Dispolimit überzogen haben.

Ergebnis: Kundenvorname, -nachname, Geburtsdatum, Kontonummer

Mögliche Lösung, die so aber nicht von DB2 verarbeitet wird (wegen **in tmp**):

```
select    b.Name, b.Vorname, b.Gebdat, b.Kontonr
from      besitzt b, Kunde k,
           (select ko.Kontonr
            from Girokonto g, Konto ko
            where g.Kontonr = ko.Kontonr and
                g.Dispolimit > ko.Betrag) tmp
where     k.Bezeichnung = U18 and
           k.Name = b.Name and
           k.Vorname = b.Vorname and
           k.Gebdat = b.Gebdat and
           b.Kontonr in tmp
```

Alternative Formulierung für DB2:

```
select    b.Name, b.Vorname, b.Gebdat, b.Kontonr
from      besitzt b, Kunde k
where     k.Bezeichnung = 'U-18' and
           k.Name = b.Name and
           k.Vorname = b.Vorname and
           k.Gebdat = b.Gebdat and
           b.Kontonr in (select ko.Kontonr from Girokonto g, Konto ko
                          where g.Kontonr = ko.Kontonr and g.Dispolimit > ko.Betrag)
```

2. Gesucht wird die Anzahl der Kunden, die für jede Kategorie eingetragen sind.

Ergebnis: Kategoriebezeichnung, Anzahl Kunden

Achtung:

Es gibt unterschiedliche Anfragen, wenn nur diejenigen Kategorien ausgegeben werden, für die tatsächlich Kunden eingetragen sind, oder wenn alle existierenden Kategorien ausgegeben werde. Formulieren Sie beide Anfragen.

Kategorien, in die tatsächlich Kunden eingetragen sind:

```
select    Bezeichnung, count(*)
from      Kunde
group by Bezeichnung
```

Alle Kategorien:

```
select    Bezeichnung,
           (select count(*) as Anzahl
            from Kunde k
            where kat.Bezeichnung = k.Bezeichnung)
from      Kategorie kat
group by Bezeichnung
```

### Aufgabe 3: Nullwerte

Folgende Anfragen sollen diejenigen Kategorien liefern, für die kein Kunde eingetragen ist:

**A1:** `select * from Kategorie  
where Bezeichnung not in  
(select Bezeichnung  
from Kunde)`

**A2:** `select * from Kategorie kat  
where not exists  
(select * from Kunde k  
where k.Bezeichnung=kat.Bezeichnung)`

Wenn es nun einen Kunden gibt, für den keine Kategorie eingetragen wurde, dann liefern die beiden Anfragen unterschiedliche Ergebnisse. Warum?

Beispiel

Kategorie	
Bezeichnung	Klass.Kriterium
U18	Unter 18 Jahre
privat	Privatkunde
gk	Großkunde

Kunde (Gebdat o.E.d.A. weggelassen)		
Name	Vorname	Bezeichnung
Meier	Florian	NULL
Müller	Erich	privat
Huber	Max	gk

**A1:** A1 liefert ein leeres Ergebnis.

Die Unteranfrage evaluiert im Beispiel zu (NULL, privat, gk).

Damit lautet die where-Bedingung von A1:

**where** Bezeichnung not in (NULL, privat, gk)

Dies evaluiert zu:

U18 not in (NULL, privat, gk) -> not (U18 in (NULL, privat, gk)) ->

not (U18=NULL or U18=privat or U18=gk) -> not (Unknown or false or false) ->

not Unknown -> Unknown

Die Kategoriebezeichnung U18 wird nicht mit in das Ergebnis aufgenommen (da Anfrage unknown liefert). Alle anderen Kategoriebezeichnungen werden ebenfalls nicht in das Ergebnis mit aufgenommen, da für sie ja Kunden eingetragen sind.

A1 liefert also ein leeres Ergebnis.

**A2:** A2 liefert als Ergebnis U18.

Wir betrachten den kritischen Fall kat.Bezeichnung =U18

Die where-Bedingung der Unteranfrage evaluiert für U18 zu

(U18=NULL or U18=privat or U18=gk) -> unknown.

Damit wird das Tupel Florian Meier nicht in das Ergebnis der Unteranfrage von A2 übernommen, die where-Bedingung (der not exists-Ausdruck) evaluiert zu TRUE !

U18 wird damit in das Gesamtergebnis übernommen.

Die Kategorien gk und privat sind nicht Bestandteil des Ergebnisses, da für sie die Unteranfrage von A2 Tupel liefert und damit die where-Bedingung zu false evaluiert.