

**Integritätsbedingungen / Normalformen- 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: Ergänzung des Konto-Beispiels um Integritätsbedingungen**

1. Ergänzen sie alle Fremdschlüsselbeziehungen im SQL-Schema (ohne weitere Spezifizierung der Constraints)

Tabelle	Attribute	Parent
Kunde	Bezeichnung	Kategorie
Festgeldkonto	Kontonr	Konto
Girokonto	Kontonr	Konto
Kontobewegung	Kontonr	Konto
Kontobewegung	Vorname, Name, Gebdat	Kunde
Kontoauszug	Kontonr	Konto
besitzt	Kontonr	Konto
besitzt	Vorname, Name, Gebdat	Kunde
listet	Erstelldat, Kontonr	Kontoauszug
listet	Kontonr, Datum	Kontobewegung
listet	Vorname, Name, Gebdat	Kunde

Die vollständige Schemadefinition mit Fremdschlüsselbeziehungen kann im file uebblatt8-schema-constraints.sql überprüft werden.

2. Was bedeutet es für Änderungen und Löschvorgänge, wenn keine explizite Aussage zu Fremdschlüssel-Constraints getroffen wird?

Falls keine Angabe bei den Constraints steht, dann wird automatisch der Modus NO ACTION verwendet. Damit sind sämtliche Änderungs- und Löschvorgänge blockiert, wenn noch referenzierende Tupel bestehen.

3. Welche on delete Option wählt man für den Fremdschlüssel *Bezeichnung* in der Tabelle Kunde? Ergänzen Sie diese im Schema und erklären sie die Gründe dafür!

Die richtige Option ist ON DELETE SET NULL. Die Logik der Beziehung besagt ja, dass Kunden auch ohne Kategoriezuordnung bestehen können. Damit wird einfach die Referenz gelöscht, falls die Kategorie gelöscht wird.

```
CREATE TABLE mytable.Kunde
(Vorname VARCHAR(30) NOT NULL,
Name VARCHAR(30) NOT NULL,
Gebdat DATE NOT NULL,
Bezeichnung VARCHAR(30) REFERENCES mytable.Kategorie ON DELETE SET
NULL, PRIMARY KEY (Name, Vorname, Gebdat));
```

4. Welche on delete Option wählt man für die Fremdschlüsselattribute *Name*, *Vorname*, *Datum* in den Tabellen *besitzt*, *listet* und *Kontobewegung*? Formulieren Sie diese im Schema und erklären sie den Grund dafür!

Die richtige Option ist ON DELETE CASCADE.

Kontobewegung ist eine schwache Entität, deshalb besagt die Logik der Beziehung, dass eine Kontobewegung nicht ohne zugehörigen Kunden (und zugehöriges Konto) bestehen kann.

*Listet* und *besitzt* sind Beziehungsrelationen. Wenn ein Partner der n:m-Beziehung gelöscht wird, müssen auch die entsprechenden Tupel der Beziehungsrelation gelöscht werden (nicht aber der andere Beziehungspartner!)

```
CREATE TABLE mytable.Kontobewegung
(Datum Date DATE NOT NULL,
Name VARCHAR(30) NOT NULL,
Vorname VARCHAR(30) NOT NULL,
Gebdat DATE NOT NULL ,
Kontonr INTEGER NOT NULL,
Betrag NUMERIC, Bewegungsart VARCHAR(15),
PRIMARY KEY (Datum, Vorname, Name, Gebdat, Kontonr),
CONSTRAINT Kundenref FOREIGN KEY (Name, Vorname, Gebdat) REFERENCES
Kunde ON DELETE CASCADE ON UPDATE CASCADE);
```

*besitzt* und *listet* analog

ON UPDATE CASCADE: s. Teilaufgabe 1.5;  
in DB2 müsste RESTRICT or No Action verwendet werden

5. Welche on update Option würde man für die Fremdschlüsselbeziehungen aus 1.3 bzw. 1.4 wählen? Begründung!

Änderung der Kategoriebezeichnung:

Hier sollten Updates entweder propagiert oder verhindert werden, abhängig von der Anwendungslogik.

Wenn eine Änderung der Kategoriebezeichnung nicht dazu führen soll, dass Kunden keiner Kategorie mehr zugeordnet werden können, muss ON UPDATE CASCADE gewählt werden. Sonst geht auch NO ACTION.

Änderung der Kundenidentifikation:

Da es nicht erwünscht ist, dass durch eine Änderung der Kundenidentifikation die Konten des Kunden und die von ihm getätigten Kontobewegungen nicht mehr eindeutig zugeordnet werden können (s. auch Aufgabe 2.3), muß ON UPDATE CASCADE gewählt werden (bzw. ON UPDATE RESTRICT, falls wie in DB2 die Option CASCADE nicht zur Verfügung steht).

6. Ergänzen sie die Bedingung, dass für das Attribut Bewegungsart in der Tabelle Kontobewegung nur die Werte „ein“ und „aus“ enthalten sein dürfen. Versuchen Sie auch, diese Constraints nachträglich zu einer bestehenden Tabelle hinzuzufügen.

```
ALTER TABLE Kontobewegung
ADD CONSTRAINT Bewegung_korrekt CHECK (Bewegungsart in ('ein', 'aus'))
```

## Aufgabe 2: Updates in SQL

Versuchen Sie folgende Änderungen am Datenbestand aus Aufgabe 1 durchzuführen:

1. Alle Kontos mit telebanking sind gebührenfrei

Um diesen Update durchzuführen, müssen erst in der Relation Girokonto alle Konten gefunden werden, die telebanking einsetzen.

```
UPDATE Konto
SET    Gebühr = 0
WHERE Kontonr in (select Kontonr from Girokonto where Telebanking = TRUE);
```

2. Das Girokonto mit der Kontonummer 174266 soll aufgelöst (d.h. gelöscht) werden. Formulieren Sie das entsprechende delete-statement.

Was passiert, wenn folgende constraints gelten:

Create table Konto (Kontonr integer not null primary key, ....);

Create table Girokonto (Kontonr integer not null primary key, .....,  
constraint fk1 foreign key (Kontonr)  
references Konto on delete cascade );

Create table besitzt (... ,Kontonr integer not null primary key,  
constraint fk2 foreign key (Kontonr)  
references Konto on delete cascade );

Analoge constraints sollen für *listet*, *Kontobewegung* und *Kontoauszug* gelten.

```
DELETE FROM Konto WHERE Kontonr = 174266
```

Durch die definierten constraints on delete cascade werden auch die Einträge mit dieser Kontonummer aus den Relationen Girokonto und besitzt gelöscht. Semantik ist, dass kein Girokonto existieren kann, das nicht mit Betrag und Kontoführungsgebühr in der Kontorelation erfasst ist. Ebenso kann ein Kunde kein Konto besitzen, das nicht existiert.

Hinweis: Der Kunde wird beim Auflösen eines Kontos nicht gelöscht, da er noch andere Konten besitzen kann.

Beim Löschen eines Kontos werden außerdem noch alle für das Konto gespeicherten Kontobewegungen, Kontoauszüge und alle für das Konto existierenden Einträge in der Relation *listet* gelöscht:

Kontobewegung (... , constraint fk1 foreign key (Kontonr)  
references Konto on delete cascade, ...)

*listet* (... , constraint fk1 foreign key (Datum, Kontonr)  
references (Kontobewegung.Datum, Kontobewegung.Kontonr) on delete cascade, ...)

Löschen eines Kontos verursacht Löschen einer Kontobewegung. Löschen einer Kontobewegung verursacht Löschen des entsprechenden Eintrags in der Relation *listet*.

3. Ändern Sie den Namen des Kunden Erwin Müller (17.5.1935) in Erwin Mueller (17.5.1935).

Welche Tupel in welchen Tabellen werden dadurch beeinflusst?

Welche Konsequenz hat die Option *on update restrict* im Gegensatz zu *on update no action* in diesem Zusammenhang?

```
UPDATE Kunde SET Name = 'Mueller'  
WHERE Vorname = 'Erwin' and Name = 'Müller' and Gebdat = 19350517
```

Geändert wird das entsprechende Tupel in der Relation *Kunde*. Beeinflusst durch diese Änderung werden Tupel der Relationen *besitzt*, *listet* und *Kontobewegung*.

Je nach Optionen des entsprechenden *on update*-constraints geschieht folgendes:

On update cascade:	die entsprechenden Tupel in <i>besitzt</i> , <i>listet</i> und <i>Kontobewegung</i> werden ebenfalls geändert
On update restrict:	verweist die Relation (z.B. <i>besitzt</i> ) nicht mehr auf <b>exakt</b> das selbe Tupel wie vorher, wird die Update-Relation zurückgewiesen.
On update no action:	verweist die Relation (z.B. <i>besitzt</i> ) weiterhin auf <b>irgend ein</b> Tupel in <i>Kunde</i> , wird die Änderung erlaubt, sonst zurückgewiesen.

### Aufgabe 3: Funktionale Abhängigkeiten

Betrachten Sie die folgende Auftragsrelation:

**Auftrag** (ProduktNr, ProduktName, KundenNr, KundenName, Datum, Stückpreis, Anzahl, Nettopreis, Mehrwertsteuersatz, Bruttopreis)

Hinweis:

Beachten Sie, dass der Mehrwertsteuersatz vom Produkt abhängt (z.B. 8% für Bücher oder 16% für Luxusartikel)

1. Bestimmen Sie die nicht-trivialen funktionalen Abhängigkeiten in der Relation Auftrag.

$\{\text{ProduktNr}\} \rightarrow \{\text{ProduktName}, \text{Stückpreis}, \text{Mehrwertsteuersatz}\}$ $\{\text{KundenNr}\} \rightarrow \{\text{KundenName}\}$ $\{\text{ProduktNr}, \text{KundenNr}, \text{Datum}\} \rightarrow \{\text{Anzahl}\}$ $\{\text{Stückpreis}, \text{Anzahl}\} \rightarrow \{\text{Nettopreis}\}$ $\{\text{Nettopreis}, \text{Mehrwertsteuersatz}\} \rightarrow \{\text{Bruttopreis}\}$ $\{\text{ProduktName}\} \rightarrow \{\text{ProduktNr}\}$ $\{\text{KundenName}\} \rightarrow \{\text{KundenNr}\}$
--

2. Was sind die Schlüsselkandidaten?

Schlüsselkandidaten z.B.: $\{\text{ProduktNr}, \text{KundenNr}, \text{Datum}\}$ $\{\text{ProduktName}, \text{KundenName}, \text{Datum}\}$
---

3. In welcher Normalform ist diese Relation?

Begründen Sie, warum sie nicht in den entsprechenden höheren Normalformen ist.

1NF ja, da keine mehrwertigen Attribute 2NF nein, da nicht alle Nichtschlüssel-Attribute voll funktional abhängig von Schlüsselkandidaten, z.B. KundenName von KundenNR. Höhere NF nein, da 2NF nein.
---