

Datenbanken für 13 LK

gtslehrer @ freewebs.com

10. April 2006

Inhaltsverzeichnis

1	Allgemein	2
1.1	Definition DB, DBS, DBMS, Schichten	2
1.2	Drei-Schichten-Modell (3-Tier-Architecture)	4
1.3	Modellierung der Datenbank	5
2	Drei Beispiele	6
2.1	Verbindungsdatenbank der Deutschen Telekom	6
2.2	Bibliothek	6
2.3	Halter und Fahrzeug	7
2.3.1	Übung Telekom	7
2.3.2	Übung Bibliothek	7
3	Eine Tabelle	8
3.1	Grundlagen	8
3.2	Datentypen	8
3.3	Schreibweisen (Notationen)	9
3.4	Schlüssel an einer Tabelle	10
3.5	Übung	11
4	SQL-Abfragen an Einer Tabelle	12
4.1	Verbindlicher Sprachumfang, Teil 1	12
4.2	NULL	12
4.3	SQL	13
4.4	Beispieltabelle	13
4.5	Projektion	13
4.6	Selektion	13
4.7	SELECT - zieht Daten aus Datenbank	13
4.7.1	Mehrzeilige Tabellen	13
4.7.2	Einzeilige Tabelle	14

4.7.3	AS - benennt Spalten um	14
4.7.4	FROM - wählt Tabelle(n)	14
4.7.5	WHERE - Auswahlkriterien	14
4.7.6	ORDER BY - sortiert	14
4.8	Übung am grmsrv	15
5	Weitere SQL-Befehle - nicht verbindlich	16
5.1	INSERT - fügt Tupel ein	16
5.2	UPDATE - Ändert Spalten	16
5.3	DELETE - Löscht Tupel	16
5.4	CREATE TABLE	17
5.5	DROP TABLE - löscht Tabellen UNWIDERRUFLICH	17
5.6	GRANT - gewährt Zugriffsrechte	17
5.7	REVOKE - zieht Zugriffsrechte zurück	17
6	Entity Relationship (ER)	18
6.1	Notation	18
7	Kardinalitäten und Rollen	19
7.1	Rollen	19
7.2	Schreibweise	19
7.3	Kardinalitäten	19
7.3.1	1:1-Beziehung	20
7.3.2	1:n-Beziehung	20
7.3.3	n:m-Beziehung	20
7.4	Beispiel: ER-Modell für Halter und Fahrzeug	21
7.5	Übung ER	23
7.5.1	Fahrzeug und Halter	23
7.5.2	PLZ	23
7.5.3	Bibliothek	23
7.5.4	Bibliothek II	23
7.5.5	ER-Modell für Beispiel Telekom	23
7.5.6	Lawine	24
7.5.7	Ibäi	24
7.5.8	Fahrdienst	24
7.5.9	Diabetes	24
7.5.10	Fußballturnier	25
7.5.11	Schule	25
8	Vom ER-Modell zur Tabellenstruktur	26
8.1	Tabellen und Beziehungen erzeugen	26

8.2	Beziehungen in SQL modellieren Teil 1 - mit WHERE	26
8.3	Übung	27
8.3.1	Lehrmittel	27
9	Referentielle Integrität (RI) Teil 1 - verbindlich	29
9.1	0. Anomalie: Leere Datenbank	29
9.2	Einfüge-Anomalie : INSERT	29
9.2.1	Lösungsmöglichkeiten	29
9.3	Lösch-Anomalie : DELETE	29
9.3.1	Lösungsmöglichkeiten	30
9.4	Änderungs-Anomalie : UPDATE	30
9.4.1	Lösungsmöglichkeiten	30
9.5	Übung	30
9.5.1	Schüler gegen access	30
9.5.2	Schüler und Hobbies	30
10	Referentielle Integrität (RI) Teil 2 - optional	31
10.1	RI bei real existierenden DBMS	31
10.2	Handling	31
10.3	Pros and Cons	31
10.4	RI in SQL	31
11	Tabellen multiplizieren	32
11.1	Beispieltabellen	32
11.2	Multiplikation - Kartesisches Produkt	32
11.2.1	Übung	32
11.2.2	Lösung	32
12	Joins	34
12.1	Joins, die uns interessieren	34
12.2	Übung	35
12.2.1	Syntax	35
12.2.2	Schülerhobbies 1:n	35
12.2.3	Schülerhobbies n:m	35
12.2.4	Telekom	36
12.3	Joins, die uns nicht interessieren	37
13	Datenschutz	38
13.1	Einstiegsübungen	38
13.1.1	Atomstaat	38
13.1.2	Festplatte 1	38

13.1.3 Festplatte 2	38
13.1.4 Traffic	38
13.1.5 Ingenieur 1	38
13.1.6 Praktikant	38
13.1.7 Programmierer P.	39
13.1.8 Programmierer Q.	39
13.1.9 Schülerin S.	39
13.1.10 Rasterfahndung	39
13.1.11 Schüler T.	39
13.1.12 Lehrer Z.	39
13.1.13 Chaostage	39
13.1.14 Raubkopien	40
13.1.15 Schülerbibliothek	40
13.1.16 Buslinie	40
13.1.17 Lauscher	40
13.2 Unterricht	42
13.3 Definitionen	44
13.4 BDSG	44
13.5 LDSG	45
13.6 Regeln	45
13.7 Anlage zum BDSG: Die "Zehn Gebote"	46
13.8 Rechte des Betroffenen	46
13.9 Datenschutzkontrolle	47
13.10 Strafvorschriften	47
14 Codd'sche Normalformen	50
14.1 Zweck der Normalisierung	50
14.2 Startpunkt	50
14.3 1. Codd'sche Normalform	51
14.4 2. Codd'sche Normalform	52
14.5 3. Codd'sche Normalform	53
14.6 4. Codd'sche Normalform (optional)	54
14.7 5. Codd'sche Normalform (optional)	54
14.8 Denormalisierung (optional)	55
14.9 Ein dickeres Beispiel	56
14.10 Übung	58
14.10.1 Promo KG	58
14.10.2 Fahrzeug	58
14.10.3 Büromaterial	58
14.10.4 Fußballverein	58

<i>INHALTSVERZEICHNIS</i>	5
14.11 Lösung Promo KG	59
15 History: Modellierung vergangener Ereignisse (optional)	61
15.1 Beispiel	61
16 Sich selbst referenzierende Entities (optional)	62
16.1 Ungleichberechtigte Referenz	62
16.2 Gleichberechtigte Referenz	62

Disclaimer

Wissen ist zum Teilen da. Ich teile mein Wissen mit Ihnen, lieber Kollege.

Ich bin aber nicht perfekt. Unter gtslehrer@yahoo.com nehme ich dankbar Ihre Verbesserungsvorschläge entgegen.

*

Legal Blurb: Alle Informationen in diesem Dokument sind falsch, unvollständig, irreführend, irrelevant und / oder funktionieren einfach nicht.

Wenn Sie es trotzdem benutzen, und es geht dabei etwas kaputt, ist das Ihr Problem, nicht meins.

*

Bitte teilen Sie meine Web-Adresse nicht Ihren Schülern mit.

1 Allgemein

1.1 Definition DB, DBS, DBMS, Schichten

Beispiel 1. In einer Firma mit 1000 Mitarbeitern sollen die Mitarbeiter mittels Rechnern verwaltet werden. Gehalt und Urlaubsgeld sollen automatisch überwiesen werden.

Lösung 1a. Es wird eine extrem große DOS-Textdatei erzeugt, die gleichmäßig auf alle Rechner verteilt wird. Bei Änderungen wird die Datei auf sämtliche anderen Rechner neu verteilt.

Probleme 1a.

- Jeder Mitarbeiter der Personalabteilung kann alles über jeden anderen Mitarbeiter sehen.
- Das kann auch jeder Nicht-Personaler, sofern er Zugang zu einem dieser Rechner erlangt.
- Lesen, Ändern und Schreiben der Datei geht 999999mal gut und einmal schief. Anschließend ist die Datei defekt und unbrauchbar.
- Die Datei ist anfällig für Eingabefehler. Wer will all die Mayers und Meyers auseinanderhalten.
- Wenn zwei Mitarbeiter gleichzeitig ändern, wird nur eine geänderte Version überleben, d.h. die Änderungen des einen Mitarbeiters gehen verloren.

Lösung 1b. Man reserviert einen speziellen Rechner mit einem speziellen Programm darauf. Dieses kommuniziert mittels einer strukturierten Sprache, die die Daten-Struktur betrifft, mit der Außenwelt. Es beherrscht Nutzerrechte, kann die Daten in (für andere unzugänglichen) Dateien ablegen, macht automatisch Backups und gibt seine Arbeit an einen befreundeten Rechner ab, kurz bevor es abstürzt oder jemand das Stromkabel zieht.

Ein DBMS (Datenbank-Management-System)

- verwaltet mehrere Datenbanken;
- verwaltet Nutzerrechte;
- verwaltet zeitnahe Zugriffe auf dieselbe Datenbasis;
- organisiert die Daten (für den Benutzer unsichtbar) in Dateien;
- stellt Schnittstellen zur Kommunikation bereit, zB
 - proprietäre Software, wo jeder Datenban-User einzeln bezahlt werden muß;
 - textconsolen-Interface;
 - Interface zu einem Webserver zur betriebssystem-unabhängigen Kommunikation mittels Browser;
 - Software-Schnittstelle (API) zum Aufsetzen von Konzept-schicht und proprietären Interfaces.

Eine DB (Datenbank) ist eine Sammlung von Daten, zumeist über Objekte aus der realen Welt.

Ein DBS (Datenbanksystem) ist ein DBMS mit darin enthaltenen DBn.

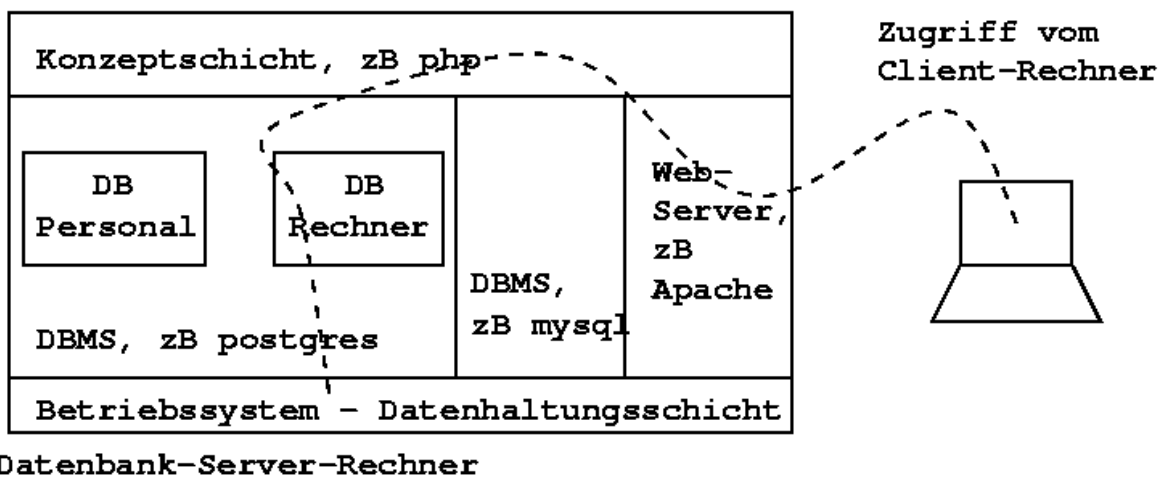
Eine Datenhaltungsschicht ist das Aufteilen der Daten in Dateien.

Ein Betriebssystem schreibt und liest Dateien nach Maßgabe des Users oder DBMS auf Speichermedien (zB Festplatten).

Eine Konzeptschicht ist zwischen DBMS und User geschaltet. Sie modelliert Randbedingungen, die vom DBMS nicht geprüft werden können oder sollen, und kann userfreundliche Fehlermeldungen erzeugen.

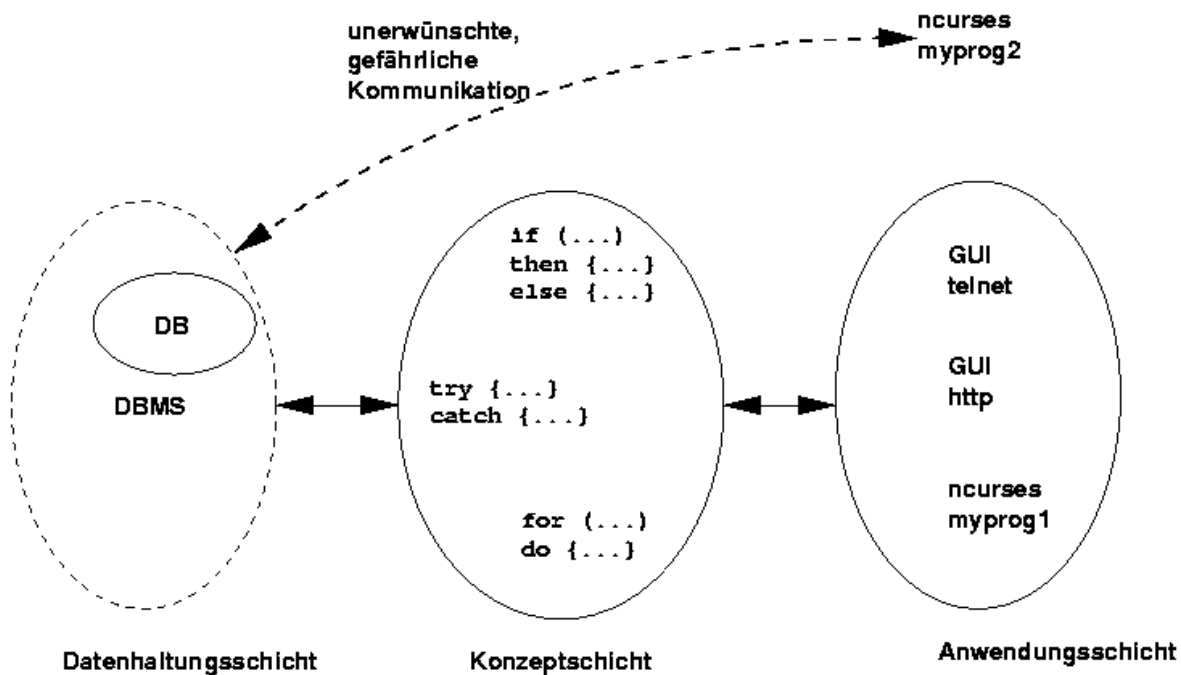
Beispiel: Wenn das Geburtsdatum eines neuen Mitarbeiters in der Zukunft liegt, kann das DBMS das nicht abprüfen.

(Moderne können, aber die Fehlermeldung lautet: "INSERT failed, value personal.gebjahr out of range.")



1.2 Drei-Schichten-Modell (3-Tier-Architecture)

Datenbank-Regeln heißen Konzept. Das Konzept sitzt in der Konzept-Schicht, die zwischen User und DBS angeordnet ist.



Für Zugriffe auf Datenbanken wird normalerweise ein 3-Schichten-Modell angewandt.

Untere Schicht: Datenhaltung. Sie umfaßt OS, DBMS, FS und Kommunikationskanäle mit der Mittleren Schicht.

Mittlere Schicht: Konzeptschicht. Hier steht drin, was wie ablaufen soll (Ref. Integrität, Reihenfolge, ...) und hat Kommunikationskanäle nach oben und unten.

Oberste Schicht: Anwendungsschicht. User-Interaktion.

1.3 Modellierung der Datenbank

Grundsätzlich können Daten in verschiedenen Konstrukten eingeordnet werden:

- Baum (tree);
- Hash (Name-Wert-Paare);
- Tabelle (relation).

Wir verwenden das relationale (tabellengestützte) Modell, d.h. wir werden unsere Daten in Tabellen aufbewahren.

2 Drei Beispiele

2.1 Verbindungsdatenbank der Deutschen Telekom

In dieser Datenbank stehen die Verbindungsdaten, kodiert wie folgt:

Vom Anschluß 069-123456 zum Anschluß 089-654321 wurde am 13.12.2002 von 23:55:13 Uhr bis 23:59:11 ein Gespräch geführt.

no	from	to	start	end
1024	069123456	089654321	2002.12.13.23.55.13	2002.12.13.23.59.11

Weitere wichtige Verbindungsdaten wie Tarif, Tarifklasse des Anrufers, Adresse usw. gehen aus weiteren, hier nicht gezeigten Tabellen (`kunden` und `tarife`) hervor.

Es sollte möglich sein, aus diesen 3 Tabellen die Monatsrechnung für Dezember 2002 für alle Kunden herzustellen.

Beachte die Verwendung eines Timestamp.

- Datum und Uhrzeit wurden NICHT auseinandergerissen, damit man die Gesprächsdauer ohne Schmerzen berechnen lassen kann.
- Most Significant First. Jahr kommt vor Monat, Monat kommt vor Tag, usw.

Diese Datenbank MUSS laufen, Tag und Nacht, 24 h am Tag, 365,25 Tage im Jahr. Sie darf nie etwas vergessen, verwechseln oder falsch berechnen.

2.2 Bibliothek

Eine Bibliothek hat 6 Bücher und 6 Kunden. Die Benutzer kommen gelegentlich vorbei, leihen Bücher aus (Leihfrist = 4 Wochen oder 2 Wochen) und/oder bringen sie zurück. Die Bibliothek gibt keinen Kredit.

Die Datenbank soll jederzeit sagen können,

- welche Bücher ausgeliehen sind;
- welche Bücher über die Leihfrist hinaus von wem ausgeliehen sind;
- wer gerade wie viele Bücher hat.

Bei Rückgabe soll sie ggf. eine Rechnung ausgeben:

- Buch defekt: Wert ersetzen.
- Leihfrist überzogen: 0,50 Euro pro Tag.

2.3 Halter und Fahrzeug

Ein Fahrzeughalter kann 0..viele Fahrzeuge halten. Jedes Fahrzeug wird von genau einem Halter gehalten.

2.3.1 Übung Telekom

Was brauchen Sie noch, um eine Rechnung schreiben zu können?

2.3.2 Übung Bibliothek

Was brauchen Sie noch, um Mahnungen schreiben zu können?

3 Eine Tabelle

3.1 Grundlagen

Eine Tabelle besteht aus Zeilen und Spalten.

Tabellen werden woanders auch "Klassen" oder "Relationen" genannt.

Spalten werden auch "Attribute" genannt.

Zeilen werden auch "Datensätze" oder "Tupel" oder "Objekt" genannt.

Die Anzahl der Zeilen ist dynamisch. Von 0 bis viele.

Anzahl, Reihenfolge und Datentyp der Attribute werden beim Anlegen der Tabelle festgelegt.

Jede Tabelle und jede Spalte hat einen "sprechenden" Namen, der keine Leerzeichen und keine Rechenzeichen enthalten darf und im Singular steht.

Regel 0a: Prozeßdaten sind verboten.

Prozeßdaten sind veränderliche Daten, die aus unveränderlichen Daten berechnet werden können.

Beispiel Es ist nicht sinnvoll, das Alter eines Kunden aufzunehmen - es ändert sich jedes Jahr. Besser ist das Geburtsdatum.

Regel 0b: Redundanz ist verboten.

Redundanz ist, wenn dieselbe Information an zwei verschiedenen Stellen gespeichert wird.

Beispiel

```
name  vorname anrede
angst anton   anton angst
```

Falls anton seinen namen ändert, muß an mehr als einer Stelle geändert werden.

3.2 Datentypen

Man hat sich bemüht, Datenbanken zu normen. Verschiedene Hersteller kümmern sich allerdings in unterschiedlichem Ausmaß um diese Normen.

Die unten angegebenen Datentypen-Grenzen sind üblich, aber möglicherweise nicht immer korrekt:

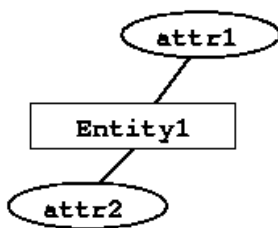
int	-2147483648 bis 2147483647
int unsigned	0 bis 4294967295
decimal(X,Y)	Dezimalzahl mit X Vorkommastellen und Y Nachkommastellen
text	string <= 65535 Zeichen
time	HH:MM:SS
date	YYYY-MM-DD
timestamp	YYYYMMDDHHMMSS

3.3 Schreibweisen (Notationen)

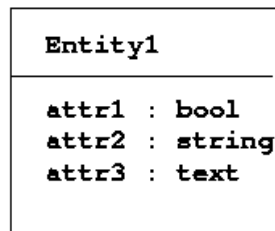
Tabelle mit Zeilen und Spalten:

```

+-----+
|          relation1          |
+-----+-----+-----+-----+
| attribut1 | attribut2 | .... | .... |
+-----+-----+-----+-----+
| ....     | ....     |     |     |
+-----+-----+-----+-----+
    
```



ER-Schreibweise



UML-Schreibweise

```

Entity1 : attr1, attr2, attr3
Entity1 (attr1, attr2, attr3);
    
```

Fließtext-Schreibweisen

3.4 Schlüssel an einer Tabelle

Jeder Datensatz sollte einen eindeutigen Bezeichner haben: einen Schlüssel. Wenn man mit diesem Schlüssel in die Tabelle hineingeht, darf nur EIN Datensatz dabei herauskommen.

Definition: Ein "Schlüssel" identifiziert einzelne Tupel.

Ein Schlüssel aus mehr als einem Attribut heißt "zusammengesetzter Schlüssel" (composite key).
Das Gegenteil heißt "nicht zusammengesetzt".

Beispiel: Personalausweis-Nummer ist eindeutiges Attribut.

Eine Koordinate (Länge/Breite) ist ein zusammengesetzter Schlüssel.

In Datenbank-Klausuren werden gern (Vorname/Nachname) als zusammengesetzter Primärschlüssel verwendet, damit man die 2. Codd'sche Normalform demonstrieren kann.

Schlüssel sollen minimal sein.
"Minimal" bedeutet: kein Attribut kann gestrichen werden, ohne dass die Eindeutigkeit verloren geht.

Definition: Ein Schlüsselkandidat ist ein Schlüssel mit minimaler Anzahl Attribute.
Eine Tabelle kann mehrere Schlüsselkandidaten haben.

Anders gesagt: Ein Schlüsselkandidat ist etwas, das man *nehmen könnte*, um einzelne Zeilen zu identifizieren.

Ein "Primärschlüssel" ist ein Schlüsselkandidat, der zur eindeutigen Identifizierung jeder Zeile *benutzt* wird.

Anders gesagt: Ein Primärschlüssel ist das, das man *wirklich nimmt*, um einzelne Zeilen zu identifizieren.

Ein "Schlüsselattribut" ist ein Attribut, das Teil mindestens eines Schlüsselkandidaten ist.
Alle anderen Attribute sind "Nicht-Schlüsselattribute".

3.5 Übung

Beantworten Sie die folgenden Fragen schriftlich und ohne ins Skript zu gucken:

- Was ist ein Primärschlüssel?
- Was ist ein Schlüsselkandidat?
- Wie heißt das Gegenteil eines zusammengesetzten Schlüssels, und wodurch zeichnet es sich aus?

4 SQL-Abfragen an Einer Tabelle

4.1 Verbindlicher Sprachumfang, Teil 1

1. SQL-Schlüsselworte sind groß geschrieben.
2. Kleingeschriebenes soll durch etwas Sinnvolles ersetzt werden.
3. Elemente in eckigen Klammern sind optional.
4. Ein | (bar; senkrechter Strich) heißt: wähle genau eines.
5. SQL-Anweisungen werden durch ein ; (Semikolon) abgeschlossen.

```
SELECT expr1 [AS attrX] [, ...exprN]
      [FROM table1 [, table2, ...]]
      [WHERE qual]
      [ORDER BY attrX [ASC|DESC] , ...attrN ];
```

Beispiele für expr :

attr1

table4.attr1 AS tab4attr1, attr2 * attr3 AS stückzahl

sum(attr1), min(attr2), max(attr3), count(attr1), avg(table4.attr3)

Funktionen:

sum	berechnet	Summe (nur für Zahlentypen)
min	berechnet	Minimum
max	berechnet	Maximum
count	berechnet	Anzahl
avg	berechnet	Mittelwert (nur für Zahlentypen)

Beispiele für qual:

table1.attr1 = table2.attr3

(attr2 * attr3 >= attr4 / attr1) AND (attr3 > 17.0345)

4.2 NULL

NULL ist SQL für "Hier steht nichts drin" oder "Ich weiß genau, daß ich es NICHT weiß."

4.3 SQL

SQL (Structured Query Language, Strukturierte Abfragesprache) ist eine genormte Sprache. Sie dient zum Ändern und Abfragen von Datenbanken.

Die Hersteller von DBMS halten sich unterschiedlich stark an die Norm. Manche Befehle lassen sie weg, andere fügen sie dazu, bei wieder anderen ändern sie die Syntax. Die grundlegenden Befehle sind aber bei allen sehr ähnlich.

4.4 Beispieltabelle

```

tausend
+-----+
a|      b| c
+-----+
0|25086| 0
1| 6053|11
2|31640|22
...
(999 rows)

```

4.5 Projektion

**Projektion ist, wenn man nur ausgewählte *Spalten* der Tabelle betrachtet.
Die ausgewählten Spalten stehen zwischen SELECT und FROM.**

4.6 Selektion

**Selektion ist, wenn man nur ausgewählte *Zeilen* der Tabelle betrachtet.
Die ausgewählten Zeilen stehen hinter WHERE.**

4.7 SELECT - zieht Daten aus Datenbank

4.7.1 Mehrzeilige Tabellen

```

SELECT expr1 [AS attr1], ...exprN
      [FROM mytable]
      [WHERE qual]
      [ORDER BY attr1 [ASC|DESC] , ...attrN ];

```

expr steht für expression, Ausdruck, Rechenaufgabe.

Variablen sind die Attribute (Spalten). Konstante Zahlen bzw. Strings sind ebenfalls erlaubt.
Verknüpfungen für Zahlen: + - */ ><

4.7.2 Einzeilige Tabelle

SQL stellt u.a. die Funktionen SUM, COUNT, MAX, MIN, AVG zur Verfügung. Wenn sie auf eine Tabellenspalte angewendet werden, kommt genau eine Zeile dabei heraus.

Man darf sie mischen, solange für jede Projektion nur 1 Zeile als Ergebnis herauskommt:

```
SELECT    MAX(a), AVG(a), 2*MIN(a)
FROM      mytable;
```

4.7.3 AS - benennt Spalten um

Falls der Spaltenname unpassend oder undefiniert ist, darf man hier einen neuen angeben. Dieser darf bei weiteren Abfragen benutzt werden:

```
SELECT    a AS Maximum,
          Maximum*Maximum,
          Maximum/a AS eins
FROM      mytable;
```

4.7.4 FROM - wählt Tabelle(n)

Selbsterklärend.

4.7.5 WHERE - Auswahlkriterien

Bedingungen dürfen geklammert und mit AND, NOT und OR verknüpft werden.
In Bedingungen werden üblicherweise Attribute gegen etwas Konstantes geprüft.

**SELECTs dürfen geschachtelt werden. Oder:
Eine einzeilige, einspaltige Tabelle ist ebenfalls eine Konstante.**

Beispiel: Alle Staaten mit mehr Einwohnern als Nigeria:

```
SELECT * FROM staat
WHERE einwohner > (
  SELECT einwohner FROM staat
  WHERE name = 'Nigeria';
);
```

4.7.6 ORDER BY - sortiert

Sortiert. ASC = ascending = aufsteigend, DESC = descending = absteigend.

4.8 Übung am `grmsrv`

Browsen Sie zum `grmsrv`, Abteilung SQL-Abfrage.

Machen Sie alles mit der Tabelle `tausend`, was erlaubt ist. Achten Sie besonders auf Überläufe und Definitionslücken.

Untersuchen Sie dabei auch, ob beim Dividieren zweier Integer-Zahlen merkwürdige Sachen passieren.

5 Weitere SQL-Befehle - nicht verbindlich

```
INSERT INTO class_name [(attr1, ...attrN)]
    VALUES (expr1,..exprN) |
    [WHERE qual]
```

```
UPDATE class_name SET attr1 = expr1, ...attrN = exprN
    [WHERE qual];
```

```
DELETE FROM class_name [WHERE qual];
```

```
CREATE TABLE class_name
    (attr1 type1 [DEFAULT expression] [NOT NULL] [PRIMARY KEY], ...attrN
```

```
DROP TABLE table;
```

```
GRANT what ON rel1, ...relN TO whom;
    what is a comma-separated list of
    ALL | SELECT | INSERT | UPDATE | DELETE | RULE
    whom is PUBLIC | GROUP group | username
```

```
REVOKE what ON rel1, ...relN FROM whom;
    what is a comma-separated list of
    ALL | SELECT | INSERT | UPDATE | DELETE | RULE
    whom is PUBLIC | GROUP group | username
```

5.1 INSERT - fügt Tupel ein

```
INSERT INTO class_name [(attr1, ...attrN)]
    VALUES (expr1,..exprN) |
    [WHERE qual]
```

Reihenfolge, Anzahl und Datentyp der Werte müssen stimmen, sofern die Attribute auf NOT NULL gesetzt sind.

5.2 UPDATE - Ändert Spalten

```
UPDATE class_name SET attr1 = expr1, ...attrN = exprN
    [WHERE qual];
```

Wenn man WHERE vergißt und weder PRIMARY KEY noch UNIQUE gesetzt sind, wird die gesamte Spalte geändert.

5.3 DELETE - Löscht Tupel

```
DELETE FROM class_name [WHERE qual];
```

**Wenn man WHERE vergißt, wird die gesamte Tabelle geleert.
So etwas wie UNDELETE gibt es nicht.**

5.4 CREATE TABLE

```
CREATE TABLE class_name
    (attr1 type1 [DEFAULT expression] [NOT NULL] [PRIMARY KEY], ...attrN
```

5.5 DROP TABLE - löscht Tabellen UNWIDERRUFLICH

```
DROP TABLE table;
```

So etwas wie UNDROP gibt es nicht.

5.6 GRANT - gewährt Zugriffsrechte

```
GRANT what ON rel1, ...relN TO whom;
```

what is a comma-separated list of
{ ALL | SELECT | INSERT | UPDATE | DELETE | RULE }

whom is { PUBLIC | GROUP group | username }

5.7 REVOKE - zieht Zugriffsrechte zurück

```
REVOKE privileg1, ...privilegN ON rel1, ...relN FROM
{ PUBLIC | GROUP group | username }
    privilege is { ALL | SELECT | INSERT | UPDATE | DELETE | RU-
LE }
```

6 Entity Relationship (ER)

Mit einer Tabelle allein lassen sich keine redundanzfreien Datenbanken erstellen.

Also braucht man pro DB mehrere Tabellen sowie deren Beziehungen (Relations, Kardinalitäten, Zuordnungen).

Die Relationen und ihre Zuordnungen zeichnet man als Entity-Relationship-Diagramme.

Entity, engl.: Seiendes, Wesen, Ding, Objekt. Hat Eigenschaften (Attribute).
 Relationship, engl.: Verwandtschaft, Zusammenhang, Zugehörigkeit, Zuordnung.

6.1 Notation

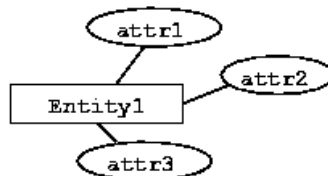
Wir verwenden zunächst die ER-Schreibweise, weil da noch Kardinalitäten dazugehören. Später verwenden wir dann UML, das spart Papier.

Entity1 (attr1, attr2, attr3)

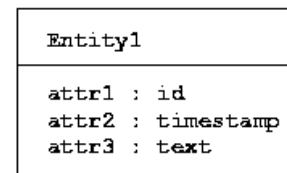
oder

Entity1 : (attr1, attr2, attr3) ;

Fließtext-Notation



ER-Notation



UML-Notation

Datentypen = optional

7 Kardinalitäten und Rollen

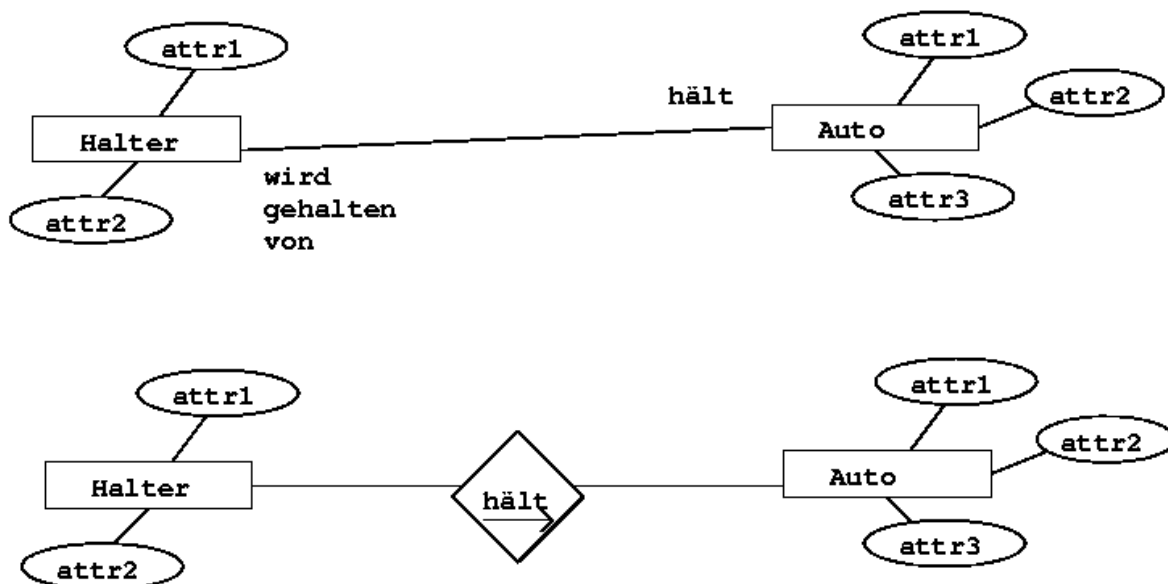
7.1 Rollen

Zwischen zwei Entitäten besteht fast immer ein Zusammenhang.

Die Beschreibung des Zusammenhangs heißt Rolle.

Häufige Zusammenhänge sind *hat*, *kennt*, *besitzt*, *benutzt*.

7.2 Schreibweise



Bei der Rauten-Schreibweise ist die Leserichtung nicht festgelegt.

Eine nicht-standardisierte Verbesserung wäre das Anzeigen der Leserichtung durch ein Pfeilchen.

7.3 Kardinalitäten

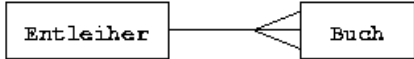
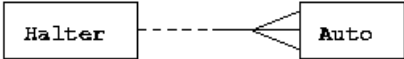
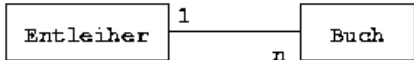
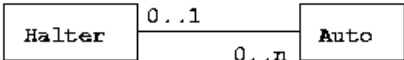
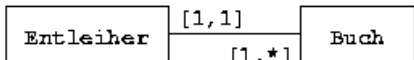
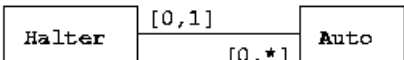
Die zahlenmäßigen Beziehungen der Entitäten untereinander heißen "Kardinalitäten".

Beispiele:

- Jeder Mensch hat eine oder mehr Staatsangehörigkeiten.
- Jeder Mensch hat 0 bis viele Autos angemeldet.

- Jedes Auto ist auf keine oder 1 Person angemeldet.
- Ein entliehenes Buch ist von genau einem Benutzer entliehen worden.

Hierfür gibt es in der Literatur verschiedene Schreibweisen. Wir verwenden Schreibweise 2.

Name	1 bis 1	1 bis viele	0 bis 1	0 bis viele
Beispiel	entliehenes Buch hat 1 Entleiher	Entleiher hat 1 bis viele Bücher	Auto hat 0 oder 1 Halter	Halter hat 0 bis viele Autos
Schreibweise 1				
Schreibweise 2				
Schreibweise 3				

Hinweis 1 : Statt n wird auch * oder oo benutzt, und umgekehrt.

Hinweis 2 : Die Beziehung 1 bzw. [1,1] wird häufig nicht hingeschrieben.

7.3.1 1:1-Beziehung

Beispiel: Personummer und Führerscheinnummer.

Eine 1:1-Beziehung soll aufgelöst werden, so daß beide Entities zu einer zusammengelegt werden.

7.3.2 1:n-Beziehung

Die lassen wir stehen. Typischerweise zeigt dabei ein Nicht-Primär-Schlüssel der Kardinalität n in der Kind-Tabelle auf den Primärschlüssel mit Kardinalität 1 in der Vater-Tabelle.

7.3.3 n:m-Beziehung

Kennzeichen: Ein A kann mehrere B haben, und ein B kann mehrere A haben.

Schreibweise:

- eine Raute mit Rollenbezeichnung in einer n:m-Beziehung;

- eine n:m-Beziehung (durchgezogen) mit einer Hilfsentity (gestrichelt an die durchgezogene Beziehung angehängt und gerne A_B genannt);
- Entity A und Entity B mit je einer 1:n-Beziehung gebunden an Entity A_B.

Beim Übergang zum Tabellenschema MÜSSEN wir eine Tabelle aus der Hilfsentity basteln.

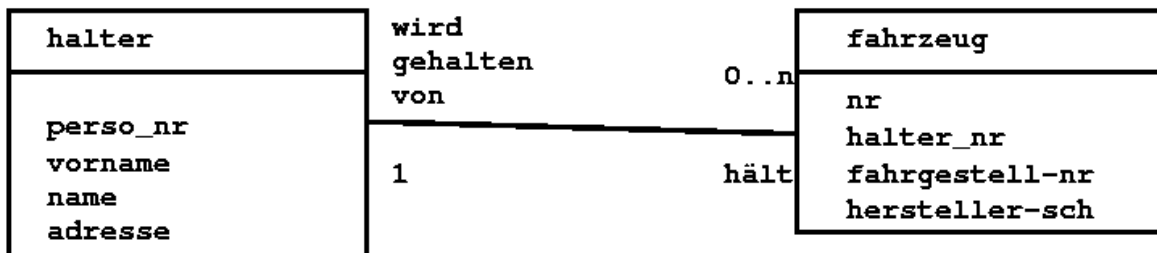
7.4 Beispiel: ER-Modell für Halter und Fahrzeug

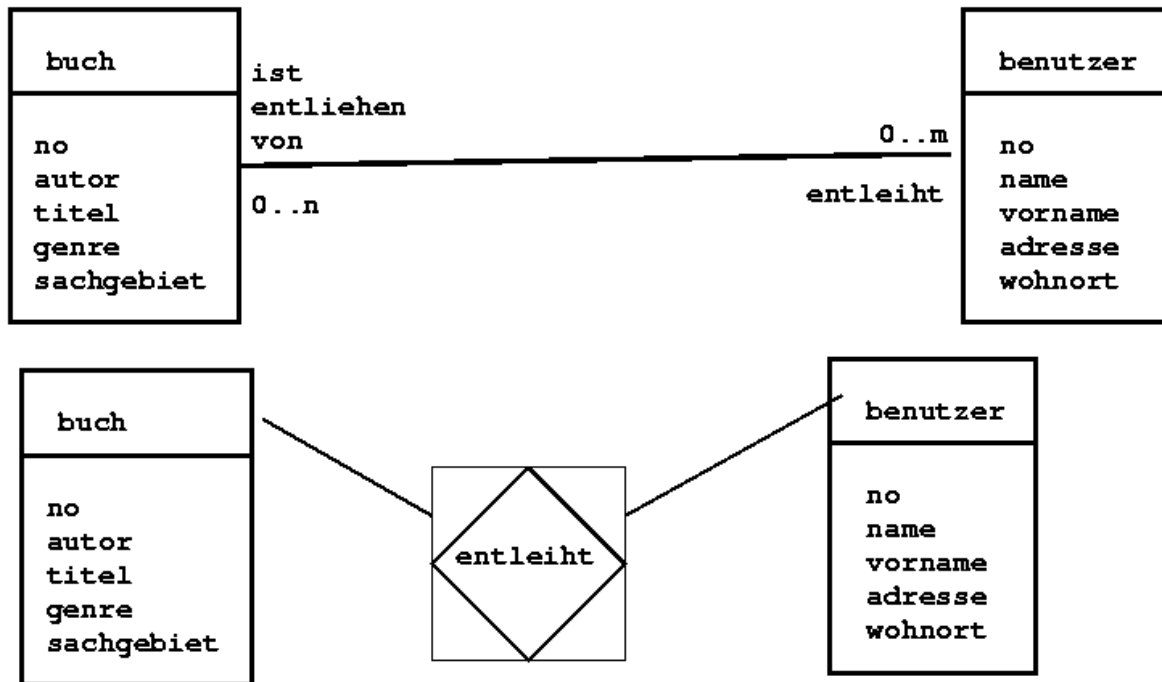
marke	modell	kennz.	vorname	nachname	strasse	ort
opel	omega	of-aa11	lutz	lastig	seeufer 3	87653 of
vw	golf2	of-bb12	martha	mies	bergstr 4	24784 of
opel	rekord	of-cc77	lutz	lastig	seeufer 3	87653 of

Das ist zunächst mal eine prima Tabelle, aus der zu jedem Fahrzeug der Halter hervorgeht, und mittels SQL auch umgekehrt. Das ER-Modell dafür ist schnell gemalt.

Leider ist die Tabelle redundant, denn lutz steht zweimal darin. Redundanz ist aber verboten.

Ergo müssen wir die Liste aufteilen in eine Entity halter und Entity fahrzeug. Zwischen beiden besteht die Beziehung "hält" bzw. "wird gehalten von".





7.5 Übung ER

Formen Sie den geschilderten Sachverhalt in ein ER-Modell um.

Begründen Sie stichwortartig und kurz, warum Sie sich für welche Kardinalitäten entschieden haben.

7.5.1 Fahrzeug und Halter

Gehen Sie davon aus, daß sich nichts ändert. Niemand zieht um oder heiratet. Niemand legt Autos still oder verkauft sie. Niemand parkt falsch.

7.5.2 PLZ

Im ER-Diagramm zu Halter-und-Fahrzeug kann man die Adresse aufteilen in (Strasse, Nr., PLZ, Ort).

Zeichnen Sie verschiedene ER-Modelle dafür und diskutieren Sie, welches optimal ist. Achten Sie dabei auf ungewöhnliche Hausnummern und Redundanzen.

7.5.3 Bibliothek

Es ändert sich nie was.

7.5.4 Bibliothek II

In der Bibliothek sind die Nutzer entweder Schüler oder Lehrer. Lehrer können im Gegensatz zu Schülern für einen bestimmten Titel verantwortlich sein.

Basteln Sie das ER-Modell.

7.5.5 ER-Modell für Beispiel Telekom

Zeichnen Sie das ER-Modell für die Telekom-Verbindungs-Datenbank, mit Attributen, Rollen und Kardinalitäten.

Berücksichtigen Sie dabei folgende Vereinfachungen:

- Es gibt 3 Tarife: super, billig, normal.
- Sie haben verschiedene Grundgebühr und verschiedene euro-pro-sec. Ansonsten sind sie gleich.
- Sie ändern sich NIE.
- Ein Gespräch kostet tags, nachts, feiertags, usw. gleichviel.
- Die Gespräche sind vorsortiert: von 069-XXXX nach 069-YYYY, also nur Ortsgespräche.
- Kunden wechseln nie den Tarif, den Wohnort, die Tel-Nr. oder den Namen.

7.5.6 Lawine

Eine Lawine hat ein Dorf in den Alpen mit nur 1 Zufahrtsstraße verschüttet. Beschreiben Sie, wie man mit Datenbanken Leben retten kann.

7.5.7 Ibäi

Niemand heiratet. Es gibt keine Power-Seller und keine Sofort-Kaufen-Angebote.

7.5.8 Fahrdienst

Oberst Reinschleimer von der Schweizer Armee möchte jederzeit wissen, wo wer oder was aus dem Fahrdienst gerade ist.

Der Fahrdienst besteht aus Fahrern mit Führerschein A, B oder C. Die Fahrzeuge fahren n Gegenstände mit Beschreibung von einem Ort zum anderen. Bei Ankunft rufen sie in der Zentrale an. Dann laden sie ab, nehmen neue Gegenstände auf, rufen nochmals an und fahren anschließend weiter. Gegenstände können Gefahrgut sein. Jedes Fahrzeug hat vorgeschriebene Führerscheinklassen, Nummernschild, Zulässige Nutzlast, TÜV-Datum, Anzahl Räder. Das Kompanie-Dreirad ist führerscheinfrei.

Fahrten haben Start und Ziel. Beendete Fahrten werden gelöscht.

SQL:

Welche Fahrzeuge sind gerade unterwegs?

Wo ist Fahrer Müller?

Mit was ist er unterwegs?

Wer transportiert gerade Gefahrgut nach Bern?

7.5.9 Diabetes

Sie müssen als Hiwi Geld verdienen und unterschreiben beim Leerstuhl für Diabetes einen 3-Monats-Vertrag. Darin ist festgehalten, daß Sie die 3. Normalform nachweisen müssen.

Die Einsatzbesprechung findet zwischen zwei Operationen statt und dauert 2 Minuten:

„Schön, daß Sie bei uns sind. Papiere gibts bei der Sekretärin. Nächste Woche brauche ich eine Datenbank für unsere Diabetesambulanz. Diabetikerbetreuung geht so: Die Patienten kommen regelmäßig alle Vierteljahr zur ambulanten Untersuchung. Natürlich auch etwa alle 2 Jahre statinär, zur Stoffwechseleinstellung, aber das brauchen wir nicht. Ich bekomme dann ein Formular, Name, Alter, Geschlecht, Sie wissen schon, Stammdaten halt. Hausarzt. Viele Patienten sind ja beim selben Hausarzt. Da könnte ich Geschichten erzählen... Pro Untersuchung Blutdruck, Puls, Gewicht, Anthropometrie halt, Laborwerte, HBA1c-Wert, Blutgefälle, Natrium, Kalium, LSD, Eythsch, THC und Restalkohol. Sie werden von insgesamt nur 6 Klinikärzten betreut. Pro Patient nur ein Arzt, nicht so wie an der Uni FFH! Da könnte ich Geschichten erzählen... Natürlich wird die Insulintherapie des Patienten eingestellt. Wir sind damit schulmedizinisch führend, als Ergebnis der DGHUIDX-Studie. Die Patienten spritzen bis zu 7mal pro Tag. Keine festen Dosen, Sie wissen das sicher. Wir sagen ihnen nur, wieviel Insulin sie pro gegessener BE spritzen sollen, und empfehlen soundsoviel BE pro Tag. Noch Fragen? Nein? Ah! Mein nächster Patient, ein seltener Fall von hypochondri-

tischer Enzephalopathitis. Bis nächste Woche! Ihr Vortrag vor den Kollegen der Uni FFH ist Montag, 16:00. Hals- und Beinbruch!"

Erstellen Sie ein Formular zur Patientenaufnahme.

Erfinden Sie einen ambulanten Untersuchungsbericht.

Modellieren Sie die Datenbank.

Zählen Sie alle Patienten.

Zählen Sie die Patienten aus Bottrop, deren THC-Wert höher ist als der Restalkohol-Wert.

Finden Sie diejenigen Patienten, deren Eythsch-Wert im letzten Jahr kontinuierlich angestiegen ist, und schreiben Sie sie an: Empfehlung, mit der Behandlung an die Uni FFH zu wechseln, wegen Kapazitätsproblemen hier.

7.5.10 Fußballturnier

Eine unbekannte Anzahl Mannschaften n spielt jede gegen jede. Jede Mannschaft besteht aus 11 Spielern und einer unbekanntem Anzahl Ersatzspielern. Vom VfB Zwickau sind drei Mannschaften gekommen. Es sind $n/2$ Fußballplätze vorhanden, die dicht beieinanderliegen.

SQL:

Aus welchen Orten kommen Spieler, die mit Vornamen Andreas heißen?

Erstellen Sie den Spielplan. Zwischen 2 Spielen ist 1 h Pause.

7.5.11 Schule

Ein Schulleiter möchte Lehrer, Schüler, Räume, Kurse und den Stundenplan in einer Datenbank modellieren.

Die Schule hat nur Doppelstunden. Montag die 1-2 heißt A1. Montag die 5-6 heißt A3. Samstag die 3-4 heißt F2.

Betrachten Sie den Mathematik-Unterricht der Klasse 181 in der 11. Klasse als Kurs. Schüler sind immer in Kursen, aber nicht immer in Klassen.

Machen Sie mal. Und denken Sie sich ein paar sinnvolle und nützliche Abfragen aus.

8 Vom ER-Modell zur Tabellenstruktur

8.1 Tabellen und Beziehungen erzeugen

Nachdem das ERM steht, wollen wir die passenden Tabellen erzeugen:

- Jede Entity wird eine Tabelle.
- Jede Entity bekommt spätestens jetzt einen Primärschlüssel. Falls irgend möglich, soll er zusammengesetzt sein, das zeigt, daß Sie es verstanden haben.

Die Verknüpfungslinien zwischen den Tabellen weisen immer mit einem Ende auf einen Schlüssel und mit dem anderen Ende auf einen Nicht-Schlüssel.

8.2 Beziehungen in SQL modellieren Teil 1 - mit WHERE

Die 1:1- und 1:n-Verknüpfungen realisieren wir durch einen zusätzlichen Befehl : WHERE. Damit das immer klappt, vergeben wir für jede Tabelle einen nicht zusammengesetzten Primärschlüssel (zB namens id).

Bisher haben wir nur die Gleichheit von Zellen derselben Tabelle geprüft. Aber mit verschiedenen Tabellen funktioniert das genauso:

```
... WHERE tab1.colx = tab2.coly
```

8.3 Übung

8.3.1 Lehrmittel

Eine Schule hat Lehrmittel (zB Bücher, Modelle oder Düsentriebwerke). Jedes Lehrmittel gehört zu einem Klassensatz. Die Lehrmittel werden an Klassen verliehen. Der Schulleiter möchte sehen können, was wann bei wem war. Jedes Lehrmittel wird an einer bestimmten Stelle (zB Regal13, UnterDerKellerTreppe) gelagert, wenn es nicht ausgeliehen ist. Wenn Lehrer ABC für die FSMT einen Satz "C++ für Dummies" beschafft und Lehrer DEF einen für das BVJ, werden diese getrennt behandelt. Die Lehrer sind schulformübergreifend tätig. Der Zustand der Lehrmittel wird nicht festgehalten.

Sie sind verpflichtet, Fragen zur Aufgabenstellung zu stellen, wenn sich welche ergeben sollten.

Sie sind aufgefordert, Ihre Kritik möglichst konstruktiv zu gestalten.

Falsch: "Die Aufgabenstellung ist doof. - Ich versteh Aufgabe 13 nicht. - Wie geht das?"

Besser: "Die Reihenfolge von Aufgabe 2 und Aufgabe 4 sollte umgekehrt werden. - Aufgabe 5 beinhaltet folgende Lösungsalternativen: a) [...] b) [...]. Welche hätten Sie gern? - Zu Aufgabe 14 habe ich im Kontextmenü einen hochinteressanten Eintrag gefunden. Darf ich ihn am Beamer vorführen? - Mein Leidensdruck ist nicht hoch genug. Welchem Mitschüler darf ich die Lösung der anstehenden Aufgaben näherbringen? - Zur Lösung von Aufgabe 17 habe ich in google, der Windos-Hilfe, der Winword-Hilfe und der Abszeß-Hilfe keine Anhaltspunkte gefunden. Welchen Mitschüler soll ich anzapfen?"

Noch besser: Denken Sie sich's aus.

Aufgaben

- Erstellen Sie das ERM.
- Erstellen Sie das Tabellen-und-Verknüpfungen-Schaubild.
- Modellieren Sie es in `access`. Schalten Sie überall Referentielle Integrität an. Überlegen und begründen Sie schriftlich, ob Sie Änderungs- und Löschweitergabe aktivieren.
- Schreiben Sie 5 Datensätze in jede Tabelle.
- Speichern Sie auf `grmsrv`.
- Versuchen Sie, in der erstellten Datenbank die RI zu brechen. Achten Sie dabei auf Veränderungen im Tabellen-und-Verknüpfungen-Schaubild. Halten Sie Ihre Ergebnisse schriftlich fest.
- Erstellen Sie die folgenden Abfragen, entweder im Visual Selector (QueryByExample, auch: "Entwurfs-Ansicht") oder in SQL:
 - Sag die Titel aller real existierenden Bücher.
 - Sag alle Lehrer und die Titel der Bücher, für die er verantwortlich ist.
 - Welcher Lehrer hütet den Klassensatz "C+ für Dummies"?
 - Welche realen Bücher sind gerade ausgeliehen?
 - Welche realen Bücher sind gerade von Klasse 4711 ausgeliehen?
 - Wo befindet sich das Buch "Lehrbuch der Gummitechnik" von Dr. Dr. Latex? Es ist schulweit nur 1 Exemplar vorhanden.
- Ersetzen Sie in Ihren Abfragen `INNER JOIN [...]` durch `WHERE tab1.ps = tab2.fk`. Prüfen Sie das Ergebnis.

- Vergeben Sie Benutzerrechte.
 - User `anyone` (ohne Passwort) darf die realen Bücher, aber nicht die Ausleihvorgänge durchsuchen.
 - User `viewer` darf alles ansehen, aber nichts ändern.
- Stellen Sie Ihre Datenbank online. Schreiben Sie `dbname` und IP Ihres Rechners an die Tafel. Greifen Sie von einem anderen Rechner aus darauf zu.
- Lassen Sie für Ihre Abfragen, für `INSERT`- und `UPDATE`-Vorgänge Formulare erstellen.
- Heben Sie diejenigen Aufgaben farbig hervor, die man komplett mit Stift und Papier lösen kann, und meditieren Sie kurz über Inhalte der anstehenden 4-h-Klausur.

9 Referentielle Integrität (RI) Teil 1 - verbindlich

Definition:

Referentielle Integrität liegt in einer Datenbank vor, wenn jeder Fremdschlüssel-Wert gültiger Primärschlüssel in einer anderen Tabelle ist.

Referenzen gehen immer von einem PS (Primärschlüssel) zu einem FK (Foreign Key).

Die Tabelle mit dem PS heißt "Vater" (engl.: parent) der Referenz.

Die Tabelle mit dem FK heißt "Kind" (engl.: child) der Referenz.

Falls mindestens ein Tupel des Kindes keinen entsprechenden Tupel beim Vater hat, heißt das Kind "Waise" (engl.: orphan).

Stellen Sie in den UML-Diagrammen den Vater etwas höher.

Beispiel: buch und buchb aus der Bibliothek.

9.1 0. Anomalie: Leere Datenbank

(Hab ich erfunden.)

Eine leere Datenbank besitzt RI.

9.2 Einfüge-Anomalie : INSERT

Die Einfüge-Anomalie tritt auf, wenn man ein Kind ohne Vater erzeugt.

9.2.1 Lösungsmöglichkeiten

- Eingabe zurückweisen. Ausgabe: "RI verletzt!"
- Zunächst Eintrag in buchb verlangen.
- Eingabe zulassen, anschließend Eintrag in buchb verlangen.

9.3 Lösch-Anomalie : DELETE

Die Lösch-Anomalie tritt auf, wenn man einen Vater löscht, der noch Kinder hat.

9.3.1 Lösungsmöglichkeiten

- **ABORT:** Löschung zurückweisen. Ausgabe: "RI verletzt!"
- **CASCADE:** Automatisch die zugehörigen Waisen löschen. Vielleicht Meldung oder Rückfrage.
- Löschen zulassen. Anschließend Ausgabe: "RI verletzt, Chem12 verwaist."

9.4 Änderungs-Anomalie : UPDATE

Die Änderungs-Anomalie tritt auf, wenn wir ein Kind ohne Vater erzeugen:

- durch Ändern der Vater-ID des Kindes
oder
- durch Ändern des Primärschlüssels des Vaters.

9.4.1 Lösungsmöglichkeiten

- **ABORT:** Änderung zurückweisen. Ausgabe: "RI verletzt!"
- **CASCADE:** Automatisch die FK der zugehörigen Kinder ändern. Vielleicht Meldung oder Rückfrage.
- Änderung zulassen. Anschließend Ausgabe: "RI verletzt, Chem12 verwaist."

9.5 Übung

9.5.1 Schüler gegen access

Übertragen Sie die bisher gefundenen ERM nach `access`.
Modellieren Sie die Beziehungen mit RI und Weitergabe.
Versuchen Sie anschließend, RI hineinzubringen.

9.5.2 Schüler und Hobbies

Erzeugen Sie zu jedem Sachverhalt das ERM und das Tabellenschema. Implementieren Sie in `access`. Versuchen Sie anschließend, die RI zu brechen.

- a) An einer Schule gibt es Schüler. Manche haben ein Hobby. Kein Hobby kommt doppelt vor.
- b) An einer Schule gibt es Schüler. Manche haben ein Hobby. Manche Hobbies kommen doppelt vor.
- c) An einer Schule gibt es Schüler. Manche haben viele Hobbies. Manche Hobbies kommen doppelt vor. Manche Hobbies benötigen ein oder mehrere Sportgeräte.

10 Referentielle Integrität (RI) Teil 2 - optional

10.1 RI bei real existierenden DBMS

DBMS	kann RI seit ca.
access	1992 (ver 1.1 - default = AUS)
postgres	2001
ms-sql	2000
mysql	2001

10.2 Handling

Mit einer System-Tabelle (wie sonst, DBMS können nur Tabellen):

```
references
=====
child   |   parent
-----+-----
tab1.fk |   tab2.ps
tab3.fk |   tab2.ps
...     |   ...
```

10.3 Pros and Cons

Vorteile RI:

- RI ist wirklich da.

Nachteile RI:

- RI macht langsam. RI ist manchmal unnütz.
- Datenbanken mit Waisen funktionieren auch irgendwie.

10.4 RI in SQL

```
CREATE TABLE buch
  (VARCHAR signatur PRIMARY KEY,
   VARCHAR isbn
    REFERENCES buch(isbn)      // !!!!! not RI-capable RDBMS ignore this
  );                          // SILENTLY !!!!!
```

11 Tabellen multiplizieren

11.1 Beispieltabellen

T1		T2	
a	b	c	d
a1	b1	c1	d1
a2	b2	c2	d2

11.2 Multiplikation - Kartesisches Produkt

Bei der Multiplikation von Tabellen werden

- alle Spalten nebeneinander hingeschrieben UND
- jeder Tupel mit jedem anderen kombiniert.

Dabei entsteht eine neue Tabelle. Ihre Spaltenanzahl ist die Summe der Ausgangsspalten. Ihre Zeilenanzahl ist das *Produkt* der Ausgangs-Tupel.

11.2.1 Übung

Bilden Sie a) T1 mal T2. b) T2 mal T1. c) T1 mal T1. d) T2 mal T2. e) T1 mal T2 mal T1.

11.2.2 Lösung

```
grimmer=> select * from T1, T2;
```

```
a |b |c |d
---+---+---+---
a1|b1|c2|d2
a2|b2|c2|d2
a1|b1|c1|d1
a2|b2|c1|d1
(4 rows)
```

```
grimmer=> select * from T1 as w, T2, T1 as e;
```

```
a |b |c |d |a |b
---+---+---+---+---+---
a1|b1|c2|d2|a1|b1
a2|b2|c2|d2|a1|b1
a1|b1|c1|d1|a1|b1
a2|b2|c1|d1|a1|b1
a1|b1|c2|d2|a2|b2
```

```
a2|b2|c2|d2|a2|b2
a1|b1|c1|d1|a2|b2
a2|b2|c1|d1|a2|b2
(8 rows)
```

12 Joins

Ein Join ist das Verbinden von zwei Relationen zu einer neuen Tabelle.

12.1 Joins, die uns interessieren

Left Outer Join, Left Join Mit einem Left Join wird eine sogenannte linke Inklusionsverknüpfung erstellt. Linke Inklusionsverknüpfungen schließen alle Datensätze aus der ersten (linken) Tabelle ein, auch wenn keine entsprechenden Werte für Datensätze in der zweiten Tabelle existiert.

Right Outer Join, Right Join Mit einem Right Join wird eine sogenannte rechte Inklusionsverknüpfung erstellen. Rechte Inklusionsverknüpfungen schließen alle Datensätze aus der zweiten (rechten) Tabelle ein, auch wenn keine entsprechenden Werte für Datensätze in der ersten Tabelle existiert.

Full Outer Join, Full Join Eine Kombination von Left Outer Join und Right Outer Join.

Inner Join, Equivalent Join **Def1:** Verbindet Datensätze aus zwei Tabellen, sobald *ein* gemeinsames Feld dieselben Werte enthält. **Def2:** Verbindet Datensätze aus zwei Tabellen anhand des ersten gemeinsamen Feldes.

INNER JOIN verknüpft zwei Tabellen mit einer Bedingung.
Auch mit INNER JOIN kann man eine 1:n-Verknüpfung zweier Tabellen in SQL modellieren.
Sie sieht üblicherweise so aus:
`SELECT * FROM child INNER JOIN father ON (father.pk = child.fk) ;`

Dabei kommt wieder eine Tabelle heraus, die natürlich mit einem weiteren INNER JOIN verknüpft werden darf, usw.

12.2 Übung

12.2.1 Syntax

```
SELECT [ ... ]          (* gibt eine Tabelle *)
FULL OUTER JOIN table2
  ON qual
;
```

12.2.2 Schülerhobbies 1:n

Erzeugen Sie eine Datenbank mit den beiden folgenden Tabellen:

schueler				hobby	
id	vorname	nachname	hid	id	name
1	anton	a	2	1	angeln
2	berta	b	3	2	baseball
3	cäsar	c	4	3	cegeln
4	doro	d	2	4	drachenfliegen
5	emil	e	3	5	eislaufen

Schreiben und benutzen Sie SQL-Anweisungen für folgende Aufgaben:

- Geben Sie alle Schüler, die ein Hobby haben, mit ihren Hobbies aus.
- Geben Sie alle Schüler mit Hobbies aus - auch die, die kein Hobby haben.
- Geben Sie alle Schüler mit Hobbies aus - auch Hobbies, die kein Schüler hat.
- Geben Sie alle Schüler und alle Hobbies aus.
- Geben Sie alle Schüler ohne Hobbies aus.
- Geben Sie alle Hobbies ohne Schüler aus.
- Welches ERM steht hinter dieser Datenbank?

12.2.3 Schülerhobbies n:m

Löschen Sie die Spalte `schüler.hid`. Erzeugen Sie noch eine Tabelle namens `schüler_hobby`:

```
sh
id sid hid
  2  2  2
  3  2  4
  4  3  4
  5  4  3
```

Erzeugen Sie keine Verknüpfungen, und finden Sie:

- alle Schüler, die ein Hobby haben, mit ihren Hobbies;
- alle Schüler mit Hobbies — auch die, die kein Hobby haben.
- alle Schüler mit Hobbies — auch Hobbies, die kein Schüler hat.
- Erzeugen Sie nun Verknüpfungen. Finden Sie anschließend alle Schüler mit ihren Hobbies. Sehen Sie sich die Syntax des Inner Join in Access an.

12.2.4 Telekom

Die Rechnung für Dezember wird berechnet wie folgt:

Betrag = Grundgebühr + Anzahl der Sekunden im Dezember * Sekundenpreis.

- a) Geben Sie für alle Kunden Name und Rechnungsbetrag aus - auch für die, die im Dezember gar nicht telefoniert haben.
- b) Versuchen Sie einen Einzelverbindungs nachweis.

12.3 Joins, die uns nicht interessieren

Cross Join entspricht dem kartesischen Produkt; häufig nutzlos

Natural Join Verknüpft die beiden Tabellen über die Gleichheit *aller* gleichlautenden Spalten. Gleichlautende Spalten werden im Ergebnis nur einmal angezeigt.

Wenn die Tabellen keine gleichlautenden Spalten haben, dann wird der Natural Join zum Cross Join.

Wenn es nur eine gleichlautende Spalte gibt, dann ist der Natural Join ein Inner Join mit anschließender Projektion, bei der gleichnamige Spalten ausgeblendet werden.

Union Join Ähnlich dem Full Outer Join werden Datensätze beider Tabellen aufgenommen. Sie werden aber nicht über eine Bedingung verknüpft.

Semi-Join (Tino Hempel weiß es auch nicht, kann nicht so wichtig sein)

Theta Join, Non-Equivalent-Join Der Theta Join ist eine Verallgemeinerung des Inner Join. Während beim Inner Join die Gleichheit des Inhalts zweier Attribute verglichen wird, wird beim Theta Join der Inhalt der Attribute i und j mit einer beliebigen Formel $\text{Theta}(i,j)$ verglichen, etwa $i = j$ (i gleich j ; InnerJoin), $i < j$ (i kleiner j), $i \leq j$ (i kleiner oder gleich j), $i > j$ (i größer j) usw.

Self-Join Der Self-Join ist ein beliebiger Join, bei dem nicht zwei verschiedene Tabellen benutzt werden, sondern zweimal dieselbe Tabelle.

13 Datenschutz

13.1 Einstiegsübungen

Ermitteln Sie zu jedem Sachverhalt, was Sie legitim bzw. illegitim finden.

Überlegen Sie, wessen Interessen höher einzustufen sind.

Markieren Sie, welche Geschichten nach Ihrer Meinung erfunden sind.

13.1.1 Atomstaat

Ein Land will eine neue Technologie mit unwägbaren Risiken einführen. Mit 1g der dabei entwickelten Substanz kann man eine Stadt von 100000 Einwohnern vergiften.

13.1.2 Festplatte 1

Lehrer Mayer findet im Rechnerschrott eine Festplatte mit Verzeichnissen, die aussehen wie Schülernamen. Er liest die Dateien mit Genuß und stellt die pikantesten Stellen anonymisiert ins Internet.

13.1.3 Festplatte 2

Lehrer Mayer findet im Rechnerschrott eine Festplatte mit Verzeichnissen, die aussehen wie Lehrerkürzel. Er kann die Dateien aber nicht lesen, da sie im netware-Dateisystem abgelegt sind.

Er besorgt sich eine Demo-Version von Netware 6.1 und liest die Dateien anschließend abends vor dem Zubettgehen.

13.1.4 Traffic

Lehrer X schneidet sämtlichen Internetverkehr seiner Schule mit und läßt ihn sich automatisch in sein privates Postfach bei t-online schicken.

13.1.5 Ingenieur 1

Eine Firma sucht für einen vertrauensvollen Posten einen Ingenieur und enthält entsprechende Bewerbungen. Ein Mitarbeiter der Firma wird beauftragt, anhand des Internets Details über 5 Bewerber herauszufinden.

Bei einem findet er heraus, daß er in Offenbach geboren wurde. Seine Mutter war Portugiesin. Der Bewerber hat qualifizierte Reparaturhinweise für den Golf 1 und Bungee-Springseile gegeben. Er findet auch einen gecachten Newsgroup-Beitrag von Nemo.Nostrum@offenbach.de zum Thema Verschleierung von Alkoholmißbrauch, aus dem u.a. hervorgeht, daß die Mutter des Autors Spanisch sprach.

13.1.6 Praktikant

Eine Firma sucht Praktikanten und gibb deren Bewerbungsunterlagen zum Screening an einen bekannten Softwarekonzern weiter.

13.1.7 Programmierer P.

Programmierer P. schreibt seine Diplomarbeit über Datenmißbrauch durch bundesdeutsche Geheimdienste und tauscht seine Gedanken darüber über Email aus. Eines Morgens stehen zwei Herren ohne Ausweis bei ihm in des Professors Büro und ersuchen beide dringend, das Thema der Diplomarbeit zu ändern.

13.1.8 Programmierer Q.

Programmierer Q. lebt in einem Dritte-Welt-Land und schickt einem Kollegen per Fax einen bissigen Kommentar zu seinem Steuerbescheid. 2 Wochen später wird er mit einem Nackenschuß im Gebirge aufgefunden. Die Behörden schließen die Akten über diesem offenkundigen Selbstmord.

13.1.9 Schülerin S.

Schülerin S. liest in der Schule über ihren Webmail-Account die Email einer Freundin, die soeben das Ergebnis ihres AIDS-Testes erfahren hat.

13.1.10 Rasterfahndung

Im Rahmen der Terroristenfahndung in den 70er Jahren wurden alle Kunden von Elektrizitätswerken polizeilich erfasst, wenn ihre Stromrechnungen sehr niedrig waren. Man versuchte damit, Wohnungen, die von Terroristen nur gelegentlich benutzt worden waren, ausfindig zu machen. Im Rahmen dieser 'Rasterfahndung' gelangten viele Bürger in diese Polizeidatei, die sich im Ausland oder Krankenhaus aufhielten, oder die Zweitwohnungen besaßen. Erst durch das Eingreifen der Datenschutzbeauftragten wurde die Datei gelöscht.

13.1.11 Schüler T.

Die Schüler einer Schule möchten innerhalb einer Schule ihre eigene Homepage haben, wissen aber nicht, wie das geht.

Schüler T. stellt ihnen ein entsprechendes bash-Skript zur Verfügung, das ihm nebenbei ermöglicht, sämtlichen Datenverkehr des fraglichen Schülers abzulauschen.

13.1.12 Lehrer Z.

Lehrer Z. stöpselt einen Rechner ins Schulnetz, schneidet 1 Monat lang sämtlichen Datenverkehr innerhalb der Schule mit (dann ist die Festplatte voll) und nimmt den Rechner anschließend wieder mit nach Hause.

13.1.13 Chaostage

In einer norddeutschen Hauptstadt wurde in den 80er Jahren eine Datei über sämtliche Punks sowie Personen aus deren Umfeld erstellt: mit Bild, Eltern, Lebensort, Lebensgewohnheiten, Kontakte, Weltanschauung, Gewaltbereitschaft etc. Da dies verboten war und in einer Behörde nicht sein kann, was nicht sein darf, existiert die Datei juristisch gesehen nicht. Deshalb gibt es bis heute (2003) auch keinen Gerichtsbeschuß zu ihrer Vernichtung.

13.1.14 Raubkopien

Die Schule beschafft für den Unterricht eine Lizenz für ein Textverarbeitungssystem, das auf einem frei zugänglichen Rechner installiert wird. Kundige Schüler und Lehrer kopieren sich die Software vom Rechner.

Nachdem ein bekannter Software-Konzern die Schulen schriftlich auf die Strafbarkeit von Raubkopien aufmerksam gemacht hat, wird auf allen Rechnern der Schule freie Software aufgespielt.

Dadurch spart die Schule einen 5-stelligen Euro-Betrag im Jahr. Viele Lehrer müssen aber von der Schulleitung zum Besuch von Kursen über Freie Software nachhaltig motiviert werden.

13.1.15 Schülerbibliothek

Die Schülerbibliothek verwendet ein Datenbanksystem zur Speicherung der Bücherdaten und der Ausleihvorgänge. Da die Bibliothekarin vom Förderverein der Schule bezahlt wird, fordert dieser von ihr einen Nachweis der erbrachten Arbeitsleistung und der Ausleihvorgänge.

Die Schulleitung möchte außerdem von der Bibliothekarin wissen, inwieweit die verwendeten Mittel für neue Bücher auch sinnvoll eingesetzt und die gekauften Bücher auch ausgeliehen wurden.

Die Datenschutzbeauftragte der Schule weist die Bibliothekarin darauf hin, daß Ausleihvorgänge nur während der Leihzeit ausleiherbezogen gespeichert werden dürfen. Anschließend seien sie zu löschen. Die Bibliothekarin weigert sich, den Bibliotheksrechner zur Arbeitskontrolle einsetzen zu lassen. Da die Ausleihvorgänge bisher nur personenbezogen gespeichert sind, muß das Programm umgeschrieben werden.

13.1.16 Buslinie

Eine große Firma baute in einer wirtschaftlich schwachen Region ein Werk. Es wurden hauptsächlich Frauen aus den umgebenden Dörfern beschäftigt. Deshalb bot das Werk eine eigene Buslinie an, um den Frauen die Arbeit im Werk zu ermöglichen. Bei der Einstellung gaben die Frauen auf Anfrage die Arbeitsstelle ihrer Ehepartner an.

Nach einigen Jahren musste das Werk aus wirtschaftlichen Gründen Stellen abbauen. Man fand per Computer anhand der Arbeitsstellen der Männer schnell heraus, dass die meisten Frauen auf den Werksbus angewiesen waren, da sich ein Zweitwagen für sie wahrscheinlich nicht rentiert hätte. Die Werksleitung ließ daraufhin die Werkslinie einstellen, die meisten betroffenen Frauen mussten von sich aus die Stelle kündigen. Die Kosten für Abfindungen usw., die bei einer regulären Entlassung angefallen wären, wurden damit gespart.

13.1.17 Lauscher

Ein Arbeitgeber belauscht über den zentralen Proxy die Internet-Aktivitäten seiner Arbeitnehmer. Was könnte er hierbei NICHT ausspionieren?

EMAIL:

- Freundeskreis (Greenpeace Aktivisten, Politiker, meine Tochter?)
- Eventuelle Kontakte zu Konkurrenz bzw. neuen Arbeitgeber
- Allgemeine Stimmungslage
- Arbeitsmoral/Einstellung seiner Arbeit gegenüber (auch wenn nicht ausdrücklich angesprochen!)

FORUM:

- Interessen
- Politische und Gesellschaftliche Einstellung
- Vertritt er unsere Firma gebührend (wenn z.B. die Branche in Spiegel Forum angegriffen wird)
- Akzeptanz/Zufriedenheit mit Firmen EDV-Lösungen

RECHERCHEN/FINANCIAL SERVICES:

- Finanzielle Lage (nicht so sehr mengenmäßig als qualitativ: Optionen (risikoreich) oder Bundesanleihen (risiko"armer))
- Berufliche Pläne
- Eigene Projekte, die eventuell mit meinen kollidieren oder meinem Untergebenen zuviel Profil geben könnte.

13.2 Unterricht

Schüler der 13 stehen dem Thema Datenschutz (wie anderen betriebs- oder gesellschaftswissenschaftlichen Themen auch) in der Regel unsensibel gegenüber. Deshalb ist eine andere Vorgehensweise erforderlich.

Geplanter Unterrichtsverlauf

Die Beispiele sollen und werden die Schüler stark emotional ansprechen. Anhand einer Diskussion oder durch Moderationstechnik lassen sich die Grundfunktion des Datenschutzes, sowie die funktionalen Elemente des Datenschutzes, wie Zulässigkeitsregeln, Pflichten der verarbeitenden Stelle, Rechte des Betroffenen etc., erarbeiten.

Nach einer Erläuterung des Aufbaus des LDSG können die Schüler den Fall hinsichtlich der Beeinträchtigung von Persönlichkeitsrechte und der funktionalen Elemente analysieren und bewerten. Für diese Phase bietet sich Gruppenarbeit an.

In einem abschließenden Unterrichtsgespräch präsentieren die Arbeitsgruppen ihre Ergebnisse.

Die Schüler sollen

- erkennen, daß es keine belanglosen personenbezogenen Daten gibt und deswegen die Verarbeitung personenbezogener Daten grundsätzlich Beschränkungen unterliegen muß
- erkennen, daß die Menschenwürde und die Persönlichkeitsrechte durch die automatische Datenverarbeitung bedroht werden können;
- erkennen und darlegen können, daß Datenschutz für die Sicherung der Grundrechte von zentraler Bedeutung ist;
- die Grundbegriffe des Datenschutzes : informationelle Selbstbestimmung, Zweckbindung, Datengeheimnis, Verbotsprinzip, kennen und anwenden können;
- wissen, welche Pflichten der Datenverarbeiter, welche Rechte der Betroffene hat, wer die Einhaltung der Datenschutzbestimmungen kontrolliert;
- bereit sein, für die Beachtung des Datenschutzes einzutreten und selbst mit den Daten anderer verantwortlich umzugehen. (Endlich in der Informationstechnik auch ein Ziel aus dem affektiven Bereich!)

13.3 Definitionen

Datenschutz ist

- 1. Schutz von Daten vor Verfälschung oder Löschung;**
- 2. Schutz vor Mißbrauch personenbezogener Daten.**

Definition: Personenbezogene Daten

Personenbezogene Daten sind Einzelangaben, die sich auf eine natürliche Person beziehen und etwas über deren Verhältnisse aussagen, wie z.B. Kenntnisse, Alter, Beruf, Gesundheitszustand, Vermögens- verhältnisse, Straftaten etc., Beziehungen zu Personen oder Beziehungen zu Sachen, wie z.B. Grundstücke und Wertgegenstände.

Definition: Datenverarbeitung ist das

- Speichern
 - Übermitteln
 - Verändern
 - Löschen
- personenbezogener Daten in oder aus Dateien oder Karteien.**

Akten, Aktensammlungen oder Bücher werden nicht als Dateien angesehen.

13.4 BDSG

Das Bundesdatenschutzgesetz gilt für alle natürlichen und juristischen Personen. Es enthält

- Zulässigkeitsregeln:
legen fest, unter welchen Voraussetzungen die Verarbeitung von Daten zulässig ist.
- Rechte des Betroffenen:
kann er gegenüber den datenverarbeitenden Stellen geltend machen.

- Kontrolle:

Das BDSG sieht ein System von Kontrollinstitutionen (Datenschutzbeauftragte, Aufsichtsbehörden) vor, die die Einhaltung der Vorschriften überwachen.

- Strafvorschriften

Das Gesetz enthält Bußgeldvorschriften und Strafbestimmungen, um Gesetzesverletzungen ahnden zu können.

13.5 LDSG

Für Landesbehörden, Kreis- und Gemeindendeckverwaltungen und insbesondere auch die Schulen, gehen die jeweiligen Landesdatenschutzgesetze (LDSG). Diese LDSG stimmen inhaltlich weitgehend mit dem BDSG überein.

Die DSG beziehen sich nur auf personenbezogene Daten. Anonymisierte Daten von ausschließlich statistischem Wert oder z.B. Unternehmensdaten, die nicht zugleich persönliche Daten eines Firmeninhabers sind, werden durch die Datenschutzgesetze nicht betroffen.

13.6 Regeln

1. Zulässigkeitsregeln, Pflichten der verarbeitenden Stelle

Persönliche Daten dürfen nur unter der Voraussetzung erfaßt werden, daß das BDSG oder LDSG es zuläßt oder andere Rechtsvorschriften es gestatten oder der Betroffene eingewilligt hat.

2. Verbotprinzip

Bei der Verarbeitung von personenbezogenen Daten ist alles verboten, was nicht ausdrücklich erlaubt ist. (Beweislastumkehr)

Das ist in der Rechtsprechung ungewöhnlich.

3. Prinzip der Erforderlichkeit

Grundidee: Die Verarbeitung von Daten ist nur zulässig, wenn diese zur Erfüllung der Aufgaben erforderlich ist.

4. Interessenabwägung

Der Zweck eines Vertrages oder ein berechtigtes Interesse kann Grundlage einer berechtigten Datenverarbeitung sein, sofern dadurch nicht schutzwürdige Belange des Betroffenen beeinträchtigt werden.

5. Zweckbindung

Die Zweckbindung besagt, daß die Daten nur zu dem Zweck verarbeitet werden dürfen, zu dem sie ursprünglich erhoben wurden.

6. Datengeheimnis

In Ergänzung zu sonstigen Geheimhaltungsvorschriften (Arztgeheimnis, Postgeheimnis, Personalaktengeheimnis, usw.) gilt für die Verarbeitung personenbezogener Daten

ein Datengeheimnis. Die in der Datenverarbeitung beschäftigten Personen sind auf dieses Datengeheimnis zu verpflichten.

7. Meldepflicht

Werden personenbezogene Daten zum Zweck der Übermittlung verarbeitet, so ist die Aufnahme dieser Tätigkeit bei der Aufsichtsbehörde meldepflichtig. Ebenso sind öffentliche Stellen in Baden-Württemberg grundsätzlich gegenüber dem/der Landesbeauftragten für Datenschutz meldepflichtig, wenn sie automatisierte Verfahren zur Verarbeitung personenbezogener Daten anwenden.

13.7 Anlage zum BDSG: Die "Zehn Gebote"

Es sind technische und organisatorische Maßnahmen zu treffen, um die Ausführung der Vorschriften des Gesetzes zu gewährleisten.

- Zugangskontrolle
- Abgangskontrolle
- Speicherkontrolle
- Benutzerkontrolle
- Zugriffskontrolle
- Übermittlungskontrolle
- Eingabekontrolle
- Auftragskontrolle
- Transportkontrolle
- Organisationskontrolle

S. Folie am Ende des Kapitels.

13.8 Rechte des Betroffenen

Das BDSG räumt dem Betroffenen folgende Rechte ein:

1. Recht auf Wissen um die Datenspeicherung (Benachrichtigung)
2. Recht auf Auskunft über den Inhalt der gespeicherten Daten
3. Recht auf Berichtigung,
4. Recht auf Sperrung,
5. Recht auf Löschung der Daten.

S. Folie am Ende des Kapitels.

13.9 Datenschutzkontrolle

1. Der Betroffene hat das Recht auf Auskunft, Berichtigung, Sperrung, Löschung.
2. Bußgelder, Strafandrohungen nach BDSG und. LDSG können verhängt werden.
3. Gerichte entscheiden in Streitfällen
4. Die Medien veröffentlichen krasse Verletzungen des Datenschutzes.

Wie der Fall von XP zeigt, interessiert sich die Bevölkerung nicht dafür.

13.10 Strafvorschriften

Das BDSG droht Geldstrafe oder Freiheitsstrafe bis zu einem Jahr, bei Bereicherungs- oder Schädigungsabsicht Freiheitsstrafe bis zu zwei Jahren.

Verletzungen bestimmter Pflichten, die das Gesetz auferlegt, können als Ordnungswidrigkeiten geahndet und mit Bußgeldern bis zu 50.000 DM belegt werden.

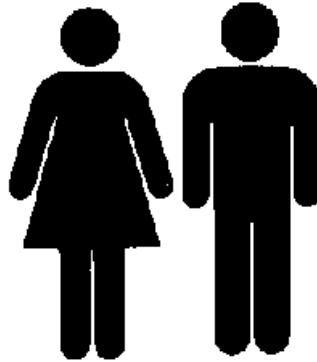
Rechte des einzelnen Bürgers

1. Einwilligungs- und Benachrichtigungsrecht

Jeder Bürger muß über die Verarbeitung seiner Daten informiert werden und mit der Speicherung einverstanden sein

(Ausnahme: wenn es eine Rechtsvorschrift bzw. ein Gesetz vorsieht)

§ 4, § 33 BDSG



2. Auskunftsrecht

Der Bürger kann Auskunft verlangen über seine gespeicherten Daten, deren Herkunft und den Zweck der Speicherung
§ 19, § 35 BDSG

3. Berichtigungsrecht

Personenbezogene Daten sind zu löschen, wenn sie unrichtig sind.

§ 20, § 35 BDSG

4. Recht auf Löschung

Der Bürger hat das Recht auf Löschung seiner Daten, wenn ihre Speicherung unzulässig oder nicht mehr notwendig ist.

§ 20, § 35 BDSG

5. Recht auf Sperrung

Der Bürger darf Daten sperren lassen, wenn sie unrichtig sind bzw. die Richtigkeit von ihm bestritten wird.

§ 20, § 35 BDSG

Die 10 Pflichten von DV-Anwendern

§ 6 Abs.1 Satz 1 BDSG: Werden personenbezogene Daten automatisch verarbeitet, sind zur Ausführung der Vorschriften dieses Gesetzes Maßnahmen zu treffen, die je nach Art der zu schützenden personenbezogenen Daten geeignet sind.



1. Zugangskontrolle

Unberechtigten ist der Zugang zur DV-Anlage, mit der personenbezogene Daten verarbeitet werden, zu verwehren.



2. Abgangskontrolle

Personen, die in der DV tätig sind, sind daran zu hindern, daß sie Datenträger entfernen



3. Speicherkontrolle

Die unbefugte Eingabe, sowie Kenntnissnahme, Veränderung oder Löschung ist zu verhindern.



4. Benutzerkontrolle

Die Nutzung des Datenverarbeitungssystems durch Unbefugte ist zu verhindern.



5. Zugriffskontrolle

Der Zugriff der Zugangsberechtigten ist nur auf die für ihn relevanten Daten einzugrenzen



6. Übermittlungskontrolle

Es muß überprüfbar sein, an wen und wie personenbezogene Daten übermittelt werden.



7. Eingabekontrolle

Es muß überprüft und festgestellt werden können, von wem und wann personenbezogene Daten eingegeben worden sind.



8. Auftragskontrolle

Die personenbezogenen Daten dürfen nur so verarbeitet werden, wie es vom Auftraggeber vorgegeben wurde.



9. Transportkontrolle

Bei der Übermittlung und beim Transport müssen Daten bzw. Datenträger vor unbefugtem Lesen, Verändern und Löschen gesichert sein.



10. Organisationskontrolle

Die Organisation einer Behörde oder eines Betriebes muß geeignet sein, daß alle Datenschutzbestimmungen erfüllt werden können.

14 Codd'sche Normalformen

So schön die Regeln sind: Bei größeren Projekten verliert man schnell den Überblick.

14.1 Zweck der Normalisierung

Gründe für die Normalisierung

- Vermeidung von unerwünschten Anomalien (Lösch-, Einfüge- und Änderungsanomalie);
- Vermeidung von überflüssigen Informationen (Redundanz);
- Zwang zum systematischen Entwurf einer Datenbank;
- bessere Übersichtlichkeit für Benutzer und Programmierer

Zum Glück gibt es ein mathematisches Verfahren, mit dem man die Datenbank aufdröseln (auf Konsistenz überprüfen) kann: die Normierung nach Codd (1970).

14.2 Startpunkt

Wir beginnend dabei mit einer SEHR GROSSEN Tabelle, die ALLE zu verwaltenden Informationen enthält.

Beispiel (ohne History)

```

halter-auto
name      strasse  ort      herst  modell  kennz
lutz kist seestr. 3 65432 of vw    golf   of-ab 3
                                opel  kadett of-bc 5
karl most torweg 4 65432 of opel  manta  of-de 4

```

14.3 1. Codd'sche Normalform

Definition:

Ein Wert ist "atomar", wenn er aus nur einem Element besteht = keine innere Struktur besitzt.

Eselsbrücke: "Ich würde diesen Wert in eine einzelne Variable schreiben." \Leftarrow ist atomar.
 "Ich würde diese Werte in ein Feld (array) oder einen Datensatz (record) oder in einen semikolon-getrennten String schreiben." \Leftarrow ist NICHT atomar.

Eine Relation ist in der 1. Normalform, wenn alle Attribute atomar sind.

Dieser Aufbau ist nach der 1. Codd'schen Normalform nicht zulässig. Wir müssen aus jeder Zeile drei neue machen, damit die Relation wenigstens die erste NF erfüllt.

Unser Beispiel - Lösung:

```

halter-auto
name      strasse  ort      herst modell kennz
lutz kist seestr. 3 65432 of vw    golf   of-ab 3
lutz kist seestr. 3 65432 of opel kadett of-bc 5
karl most torweg 4 65432 of opel  manta  of-de 4
  
```

Bei der 1. Normalenform können Zeilen oder Spalten dazukommen.

Noch'n Beispiel

```

+-----+-----+-----+
| name   | adresse           | handy   |
+-----+-----+-----+
| anton gas | Seestr. 13       | 0171-12345 |
|          | 65432 Offenbach | 0172-54321 |
+-----+-----+-----+
  
```

14.4 2. Codd'sche Normalform

Definition:

Attribut A ist "funktional abhängig" von B, wenn gilt: aus A folgt B.

Definition:

B ist "voll funktional abhängig" vom Primärschlüssel, wenn der Primärschlüssel zusammengesetzt ist und B nicht bereits aus Teilen des Primärschlüssels folgt.

Eine Relation ist in der 2. Normalform, wenn sie in der 1. Normalform ist und jedes Attribut voll funktional vom *gesamten* Primärschlüssel abhängt.

Oder:

Kein Attribut darf nur von einem Teil eines (zusammengesetzten) Primärschlüssels abhängen.

Hinweis: Wenn die Relation bereits in der ersten Normalform ist und der Primärschlüssel nicht zusammengesetzt ist, dann ist die 2. NF erfüllt.

Bei der 2. Normalform können Tabellen dazukommen. Deren Tupel sind dann von einem Teil des Primärschlüssels einer anderen Tabelle abhängig.

Unser Beispiel - Lösung:

Zunächst müssen wir den Primärschlüssel raten.

Der Primärschlüssel ist wahrscheinlich das KFZ-Kennzeichen. Dieser ist nicht zusammengesetzt, ergo ist 2. NF erfüllt.

14.5 3. Codd'sche Normalform

**"Transitiv abhängig" bedeutet zB:
Aus X folgt Y und Aus Y folgt Z und Aus Y folgt NICHT X.
In diesem Beispiel ist Z über Y von X transitiv abhängig.**

Eine Relation ist in der 3. Normalform, wenn sie

- 1. in der 2. Normalform ist und**
- 2. die Nicht-Schlüssel-Attribute funktional unabhängig voneinander sind.**

Oder: Kein Nicht-Schlüssel-Attribut darf von einem anderen Nicht-Schlüssel-Attribut abhängen.

Auch bei der 3. Normalform können Tabellen entstehen.

Unser Beispiel - Lösung Strasse und Ort hängen vom Namen ab!

Modell hängt von Hersteller ab! (Es sei denn, Ford baut demnächst das Modell "Käfer"...)

Ergo:

halter (name, strasse, ort)

modell (modell, hersteller)

anmeldung (name, modell)

14.6 4. Codd'sche Normalform (optional)

Eine Relation ist in der 4. Normalform, wenn sie in der 3. Normalform ist und keine paarweise auftretenden mehrwertigen Abhängigkeiten vorhanden sind.

Beispiel:

Die Disposition muß informiert werden, an welchem Ort im Lager welcher Artikel mit welcher Auftragsnummer liegt, damit sie den Lastwagen entsprechend beladen kann.

disposition		
artikelno	lager	auftragnr
4	s4	3
4	n4	3
4	s4	7
4	n4	7
5	o1	8
5	s1	8
5	l1	8
5	o1	0
5	s1	0
5	l1	0

Es sind paarweise auftretende mehrwertige Abhängigkeiten vorhanden.

Damit die Relation der 4. Normalform genügt, muß sie in artikelno-lager und artikelno-auftragnr aufgespalten werden.

14.7 5. Codd'sche Normalform (optional)

**Eine Relation ist in der 5. Normalform, wenn sie in der 4. Normalform ist und sie sich nicht durch Kombination einfacherer Relationen mit unterschiedlichen Schlüsseln bilden läßt.
Oder: Wenn man sie nicht in noch kleinere Relationen auseinanderreißen kann, ohne dabei wenigstens teilweise dieselben Schlüssel zu benutzen.**

Selbsterklärend.

14.8 Denormalisierung (optional)

Nach der kompletten Normalisierung kann man das erhaltene Modell 1:1 implementieren, und es wird funktionieren.

Wenn man dabei allerdings sehr viele Tabellen erhalten hat, wird es langsam.

Sofern man nur statische Daten (die sich nicht ändern) hat, KANN man über eine Denormalisierung nachdenken. Man sollte sich dabei aber bewußt machen, warum man wo gegen welche Normalisierungsregel verstoßen hat.

14.9 Ein dickeres Beispiel

auftrag							
id	kunde	kunde-id	artikel	artikel-id	anzahl	hersteller	herst-id
13	johann mueller	005	kondom	222	7	gabriel	112
			tesafilm	227	3	luzifer	113
			fussball	229	1	michael	114
14	carsten meyer	007	locher	223	1	gabriel	112

Ist nicht in 1. NF, da nicht-atomare Werte vorhanden sind. Lösung:

auftrag							
id	kunde	kunde-id	artikel	artikel-id	anzahl	hersteller	herst-id
13	johann mueller	005	kondom	222	7	gabriel	112
13	johann mueller	005	tesafilm	227	3	luzifer	113
13	johann mueller	005	fussball	229	1	michael	114
14	carsten meyer	007	locher	223	1	gabriel	112

Nun müssen wir den Primärschlüssel raten. Nach genügend langer Diskussion ergibt sich als einzige Möglichkeit: id - artikel-id.

Ist nicht in 2. NF, da Nicht-Schlüssel-Attribute von Teilen des Primärschlüssels abhängen. Lösung:

artikel			
artikel-id	artikel	hersteller	herst-id
222	kondom	gabriel	112
227	tesafilm	luzifer	113
229	fussball	michael	114
223	locher	gabriel	112

auftrag-kunde			auftrag-artikel		
id	kunde	kunde-id	id	artikel-id	anzahl
13	johann mueller	005	13	222	7
14	carsten meyer	007	13	227	3
			13	229	1
			14	223	1

Ist nicht in 3. NF, da Nicht-Schlüssel-Attribut hersteller von Nicht-Schlüssel-Attribut herst-id abhängt. Lösung:

hersteller		artikel		
hersteller	herst-id	artikel-id	artikel	herst-id
gabriel	112	222	kondom	112
luzifer	113	227	tesafilm	113
michael	114	229	fussball	114
		223	locher	112

**Am liebsten würde ich die Tabellen nennen wie folgt:
artikel - hersteller - auftrag (id, kunde) - posten**

14.10 Übung

14.10.1 Promo KG

PersNr	Name	Vorname	AbtNr	Abteilung	ProjektNr	Beschreibung	Zeit
1	Lorenz	Sophia	1	Personal	2	Verkaufspromotion	83
2	Hohl	Tatjana	2	Einkauf	3	Konkurrenzanalyse	29
3	Willschrein	Theodor	1	Personal	1,2,3	Kundenumfrage, Verkaufspromotion, Konkurrenzanalyse	140, 92, 110
4	Richter	Hans-Otto	3	Verkauf	2	Verkaufspromotion	67
5	Wiesenland	Brunhilde	2	Einkauf	1	Kundenumfrage	160

Normalisieren Sie.

14.10.2 Fahrzeug

Normalisieren Sie Fahrzeug im Buch, S. 93.

14.10.3 Büromaterial

Normalisieren Sie Büromaterial im Buch, S. 93.

14.10.4 Fußballverein

Normalisieren Sie Fußballverein im Buch, S. 93.

14.11 Lösung Promo KG

1. Normalform

=====

Eine Relation befindet sich in der ersten Normalform, wenn alle ihre Attribut nur einfache Attributwerte aufweisen, d. h. wenn jeder Kreuzungspunkte von Zeile und Spalte höchstens einen Wert besitzt (Bezeichnung: atomar).

Die Redundanz nimmt zu. Der bisherige Primärschlüssel verliert seine Eindeutigkeit, es muss ein Neuer gefunden werden.

Festlegung: PersNr und ProjektNr ist der neue Primärschlüssel.

2. Normalform

=====

Eine Relation befindet sich in der zweiten Normalform, wenn

1. sie in der ersten Normalform ist und
2. jedes Nicht-Schlüssel-Attribut vom gesamten Identifikationsschlüssel funktional abhängig ist, nicht aber bereits von Teilen des Gesamtschlüssels.

Bemerkung:

- o Es muss also stets geprüft werden, ob man aus einem Teilschlüssel auf Nichtschlüsselattribute folgern kann.
- o Bei der Erzeugung der 2. NF entstehen i. d. R. mehrere Relationen!

Beispiel:

Aus dem Teilschlüssel ProjektNr lässt sich bereits eindeutig auf Beschreibung schließen. Aus PersNr lässt sich eindeutig auf Name, Vorname, AbtNr und Abteilung schließen. Somit sind zwei neue Relationsschemen Projekt und Personal zu erzeugen und das verbleibende Schema Firma so zu überarbeiten, dass die Attribute Beschreibung, Name, Vorname, AbtNr und Abteilung gelöscht werden.

3. Normalform

=====

- o Es muss also stets geprüft werden, ob man aus einem Nichtschlüsselattribut auf weiteres Nichtschlüsselattribute folgern kann.
- o Bei der Erzeugung der 3. NF entstehen weitere Relationen!

Aus der PersNr. folgt Name, Vorname, AbtNr und die Abteilung. Aber

aus der AbtNr. folgt bereits die Abteilung! Also werden die Attribute AbtNr. und Abteilung in eine neue Relation gebracht.

15 History: Modellierung vergangener Ereignisse (optional)

Je nachdem ob die Datenbank nur den Ist-Zustand oder auch die Vergangenheit beschreiben soll, sind verschiedene Konzepte nötig.

In unseren bisherigen ER-Modellen wird nur die Gegenwart modelliert.

Problem: Wenn ein böser Mensch falsch parkt, erwischt wird und anschließend sein Auto abmeldet, gibt es keine Möglichkeit, auf seine Adresse zu kommen.

Anders gesagt: Was jede Kartei auf Papier leistet, nämlich eine Aufzeichnung der Vergangenheit, das leistet unsre Datenbank *nicht*.

Lösung: Was vorher das Attribut `attr1` war, wird abgelöst und in eine Entity `Eattr1` verwandelt. Diese hat außer dem Attribut `attr1` die Attribute `owner` und `seit`.

`owner` sagt, zu wem `Eattr1` gehört.

`seit` sagt, seit wann sein Zustand beobachtet wird.

15.1 Beispiel

Vorher:

```
patient
=====
name    vorname risiko
mueller hans    raucher
```

Nachher:

```
patient
=====
id name    vorname
17 mueller hans
```

```
risiko
=====
pid  seit      risiko
17   01.01.1990 raucher
17   01.01.2001 nichtraucher
```

Übung: Zeichnen Sie zu beiden Fällen die ER-Modelle.

16 Sich selbst referenzierende Entities (optional)

16.1 Ungleichberechtigte Referenz

Bei der ungleichberechtigten Referenz haben beide Enden der Referenz verschiedene Rollennamen.

Beispiel: Chefs In angestellter gibt es die 1:n-Referenz "ist Chef von". Also muß jeder Angestellte eine CID (Chef-ID) haben.

Achtung, der Chef vom Ganzen hat keinen Chef.

Lösung: Entweder Cid darf NULL werden ODER Der Chef wird sein Chef.

Beispiel: Arzt In mensch gibt es die n:m-Referenz "behandelt". Dabei ist der eine Mensch Arzt und der andere Patient.

mensch	behandlung
=====	=====
Mid ...	BehandlID ArztID PatientID Datum ...
-----+-----	-----+-----+-----+-----+-----

16.2 Gleichberechtigte Referenz

Bei der gleichberechtigten Referenz haben beide Enden der Referenz gleiche Rollennamen.

Beispiel Menschen heiraten. Aufgabe: Finden Sie das ERM. Hinweise:

- Die Richtung der Referenz ist - nun ja - beidseits.
- Die Kardinalität ist n:m.

Falsch: Mid-1 und Mid-2 sind gleichberechtigt und dürfen nicht in verschiedenen Spalten stehen.

mensch	heirat
=====	=====
Mid ...	Hid Datum-start Datum-end Mid-1 Mid-2
-----+-----	-----+-----+-----+-----+-----

Richtig:

mensch		heirat			gatte		
=====		=====			=====		
Mid	...	Hid	Datum-start	Datum-end	Gid	Hid	Mid
-----+	-----	-----+	-----+	-----	-----+	-----+	-----