

Informatik Übung 5

1. Aufgabe

Relation: Bezeichnet die Beziehungen zwischen zwei Tabellen, jedoch auch eine Tabelle da diese ja nichts anderes ist als eine Zusammenführung mehrerer Elementen.

Tupel: Ein Datensatz, bzw. eine Zeile in einer Tabelle.

Attribut: Eine Eigenschaft, bzw. Element welches in einem Datensatz enthalten ist.

Tabelle: Stellt die Relationen mehrerer Attribute dar, ist also eine Sammlung von vielen Datensätzen.

Schlüssel: Durch die Schlüssel werden die Relationen eindeutig bezeichnet, damit man diese später identifizieren kann. So ist ein Schlüssel z.B. eine Nummerierung der Datensätze/Relationen.

Primärschlüssel: Durch den Primärschlüssel wird eben genau jeder Datensatz eindeutig bezeichnet, der Primärschlüssel kann aus einem oder mehreren Schlüsseln bestehen, wichtig ist das die Bezeichnung schlussendlich eindeutig ist.

Fremdschlüssel: Ein Fremdschlüssel ist ein Sekundärschlüssel der Tabelle, welcher in einer anderen Tabelle der Primärschlüssel ist. So ein Schlüssel verweist somit auf einen Wert oder zusätzliche Informationen welche in einer anderen Tabelle gespeichert sind.

Mehrbenutzerbetrieb: Damit mehrere Benutzer zeitgleich an einer Datenbank arbeiten können, verwaltet diese auch den Mehrbenutzerbetrieb. Dies beinhaltet Berechtigungen, Sperren von Datensätzen (z.B. Zeitgleiches ändern der Datensätze) oder log files.

Transaktion: bezeichnet die Durchführung mehrerer Operationen welche als eine Einheit bezeichnet werden. Verlangt wird vor allem, dass die Einheit vollständig oder gar nicht ausgeführt wird.

ACID-Prinzip: Muss beim Ausführen einer Transaktion erfüllt sein:

- Atomicity: ganz oder gar nicht ausgeführt
- Consistency: nach der Ausführung muss der Datenbestand in einer Konsistenten Form sein
- Isolation: mehrere gleichzeitige Transaktionen dürfen einander nicht beeinflussen
- Durability: müssen dauerhaft in der Datenbank bestehen bleiben

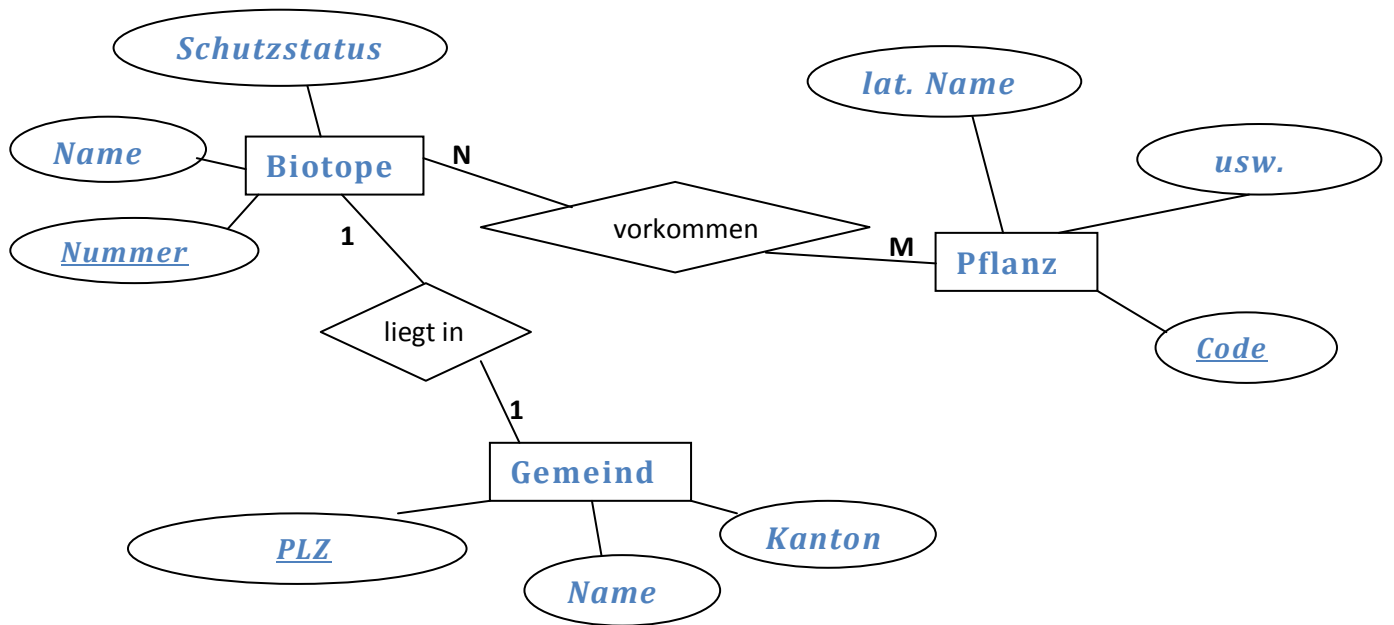
2. Aufgabe

Das Objekt „Student/in“ beinhaltet als Attribute die Namen und Legi-Nr. der Studenten. Diese besitzen eine n:1 beziehung zu den Studiengängen, heisst einem jeden Studiengang können mehrere Studenten zugeordnet werden, aber jedem Studenten nur ein Studiengang, diese Beziehung wird zusätzlich noch durch das Semester beschrieben.

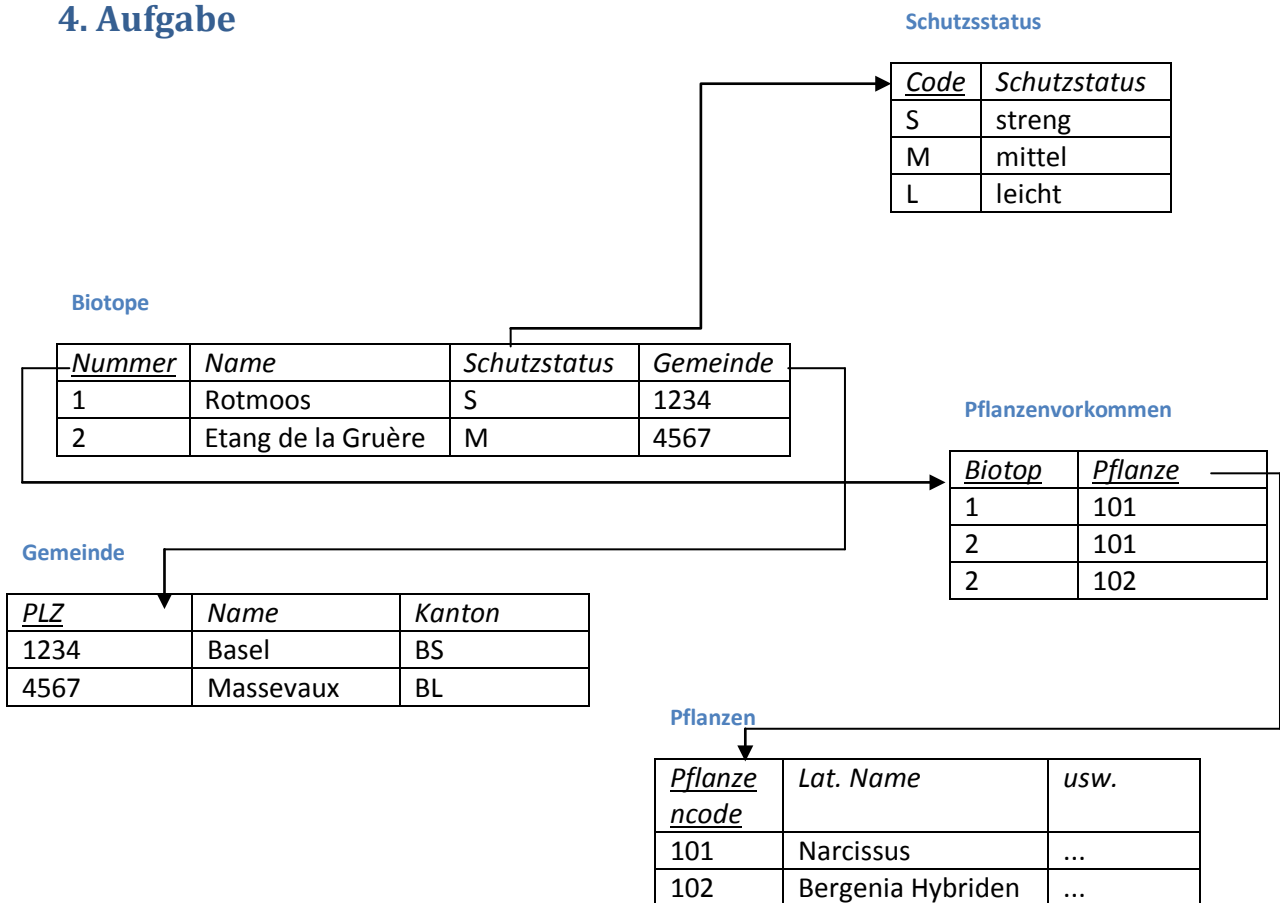
Dazu besteht zwischen dem Studiengang, mit den Attributen Bezeichnung und Departement, eine m:n Beziehung zu den Lehrveranstaltungen mit den Attributen LV_Name und LV_Nummer.

Dies heisst ein Studiengang kann mehrere Lehrveranstaltungen beinhalten und eine Lehrveranstaltung kann durch mehrerer Studiengänge besucht werden.

3. Aufgabe



4. Aufgabe



5. Aufgabe

1. Normalform: Jedes Attribut einer Tabelle muss einen atomaren Wertebereich haben. Wenn nicht erfüllt müssen die nicht atomaren Attribute in weitere Attribute aufgespalten werden.

2. Normalform: Die 1. NF muss erfüllt sein, dazu muss jedes Attribut von allen Primärschlüsseln abhängig sein, und nicht nur von einem Teil. Falls nicht erfüllt muss die Tabelle in mehrere thematisch einheitliche Tabellen aufgespalten werden.

3. Normalform: die 2. NF muss erfüllt sein, dazu darf kein Attribut transitiv von einem Primärschlüssel abhängen, heisst alle Attribute müssen 'direkt' von allen Primärschlüsseln abhängen. Falls nicht erfüllt muss auch hier die Tabelle weiter in mehrere Tabellen aufgespalten werden.

Tabelle: Ist in keiner Normalform, da Strasse (Nummer und Namen im gleichen Feld) nicht atomar ist. (Könnte jedoch je nach DB auch so belassen werden, somit wäre die Tabelle in der 2. NF, da alle Informationen zwar zur Adresse gehören, der Namen der Ortschaft jedoch von der PLZ abhängig, und somit transitiv vom Hauptschlüssel.)

Relationenmodell aus Aufgabe 4: Ist schon vollständig normalisiert...

Kunden

<u>Nr.</u>	Name	Vorname	Strasse	StrNr.	PLZ
1	Muster	Max Moritz	Demoweg	23	8044
2	Flattertier	Fabiola	Nachtgasse	13	1234
3	Hacker	Hugo Hannibal	Schrottweg	56	8044

Gemeinden

<u>PLZ</u>	Name
8044	Zürich
1234	Flederhausen

6. Aufgabe

The screenshot shows the phpMyAdmin interface for a database named 'test'. The left sidebar shows the database structure with 'GEMEINDEN' and 'KUNDEN' tables. The main area displays the following SQL commands:

```
ALTER TABLE KUNDEN ADD PRIMARY KEY ( nr ); # MySQL lieferte ein leeres Resultat zurück (d.ansp:h. null Zeilen).
ALTER TABLE GEMEINDEN ADD PRIMARY KEY ( plz ); # MySQL lieferte ein leeres Resultat zurück (d.ansp:h. null Zeilen).
ALTER TABLE KUNDEN ADD FOREIGN KEY ( plz ) REFERENCES GEMEINDEN( plz ); # MySQL lieferte ein leeres Resultat zurück (d.ansp:h. null Zeilen).
```

Below the commands, the same SQL statements are repeated in a larger font for readability:

```
alter table KUNDEN add primary key(nr);# MySQL lieferte ein leeres Resultat zurück (d.ansp:h. null Zeilen).
alter table GEMEINDEN add primary key(plz);# MySQL lieferte ein leeres Resultat zurück (d.ansp:h. null Zeilen).
alter table KUNDEN add foreign key(plz) references GEMEINDEN(plz);# MySQL lieferte ein leeres Resultat zurück (d.ansp:h. null Zeilen).
```

7. Aufgabe

The screenshot shows the phpMyAdmin interface with the SQL tab selected. A green message bar at the top states: "Ihr SQL-Befehl wurde erfolgreich ausgeführt." Below this, the executed SQL code is displayed:

```

INSERT INTO GEMEINDEN( plz, name )
VALUES ( 1234, 'Fliederhausen' ) ;# 1 Zeile(n) betroffen.
INSERT INTO GEMEINDEN( plz, name )
VALUES ( 3983, 'Mörel' ) ;# 1 Zeile(n) betroffen.
INSERT INTO KUNDEN( nr, name, vorname, strasse, strnr, plz )
VALUES ( 1, 'Muster', 'Max Moritz', 'Demoweg', 23, 8044 ) ;# 1 Zeile(n) betroffen.
INSERT INTO KUNDEN( nr, name, vorname, strasse, strnr, plz )
VALUES ( 2, 'Flattertier', 'Fabiola', 'Nachtgasse', 13, 1234 ) ;# 1 Zeile(n) betroffen.
INSERT INTO KUNDEN( nr, name, vorname, strasse, strnr, plz )
VALUES ( 3, 'Hacker', 'Hugo Hannibal', 'Schrottweg', 56, 8044 ) ;# 1 Zeile(n) betroffen.
    
```

Below the code, the text "SQL-Befehl(e) in Datenbank test ausführen:" is followed by the same SQL code block.

Bei erneutem einfügen des gleichen Eintrages, wird ein Fehler zurückgegeben, da dieser schon identisch vorhanden ist und darum nichts am aktuellen Eintrag geändert werden kann.

8. Aufgabe

The screenshot shows the phpMyAdmin interface with the SQL tab selected. A green message bar at the top states: "Zeige Datensätze 0 - 1 (2 insgesamt, die Abfrage dauerte 0.0034 sek.)". Below this, the executed SQL code is displayed:

```

SELECT *
FROM KUNDEN
WHERE plz =8044
LIMIT 0 , 30
    
```

The result is displayed in a table view with the following columns: nr, name, strasse, strnr, plz, vorname. The data is as follows:

nr	name	strasse	strnr	plz	vorname
1	Muster	Demoweg	23	8044	Max Moritz
3	Hacker	Schrottweg	56	8044	Hugo Hannibal

Below the table, the text "Alle auswählen / Auswahl entfernen markierte:" is followed by icons for selecting, deleting, and refreshing the table.

The screenshot also shows the second part of the task, where the SQL query is modified to include the GEMEINDEN table:

```

SELECT *
FROM KUNDEN
JOIN GEMEINDEN
WHERE KUNDEN.plz = GEMEINDEN.plz
LIMIT 0 , 30
    
```

The result is displayed in a table view with the following columns: nr, name, strasse, strnr, plz, vorname, plz, name. The data is as follows:

nr	name	strasse	strnr	plz	vorname	plz	name
1	Muster	Demoweg	23	8044	Max Moritz	8044	Zürich
2	Flattertier	Nachtgasse	13	1234	Fabiola	1234	Fliederhausen
3	Hacker	Schrottweg	56	8044	Hugo Hannibal	8044	Zürich