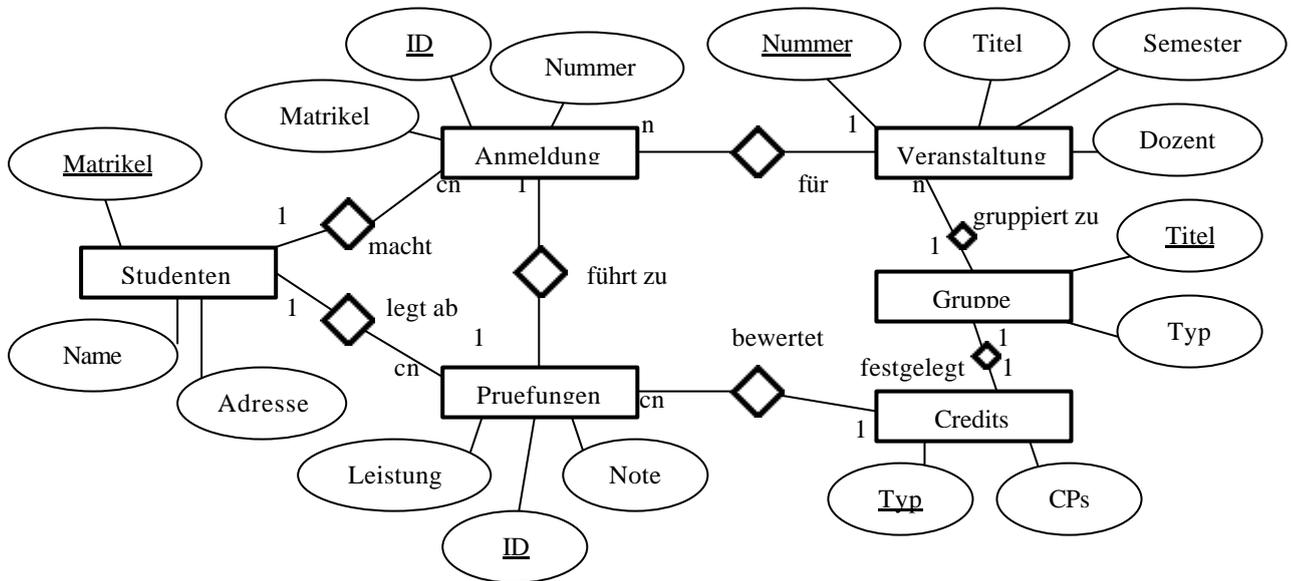


Datenbankensysteme – Aufgabenblatt 2

Aufgabe 1:

Es folgt das ER-Modell unter Berücksichtigung der 1. – 3. NF:



ANNAHMEN:

- Relation Studenten:
 1. erfasst werden nur Diplomstudenten im Hauptstudium
 2. es wird nur die erste polizeilich gemeldete Adresse gespeichert
 3. es kann Studenten mit gleichen Namen geben, auch Studenten mit gleicher Adresse
 4. aber: nicht Studenten mit gleichem Namen und gleicher Adresse
 5. die Matrikelnummer ist eindeutig
- Relation Lehre (normalisiert in: Veranstaltung, Vergleichbare, Credits):
 1. es kann keine Veranstaltung geben, bei der Titel, Semester und Dozent gleich sind => Eindeutigkeit der Nummer
 2. vergleichbare Veranstaltungen haben gleiche Creditpoints
- Relation Leistungen (normalisiert in: Anmeldung, Pruefungen):
 1. Die ID ist eindeutig (nur eine Anmeldung pro ID)
 2. Die ID in Anmeldungen ist die gleiche wie in Pruefungen => die Daten „durchlaufen“ die Relationen (dabei: nach Anmeldung immer Prüfung)
 3. Die Leistung ist ein boolean-Wert (bestanden oder nicht), die Note wird explizit eingegeben (leider sind – in Access – hier nur unrealistische ganze Zahlen möglich) => durch den boolean-Wert können die Bonus- / Maluspunkte in vordefinierten Unterabfragen (Bonus und Malus) berechnet werden
- Mengenaussagen:
 - Ein Student meldet sich 0 bis n –mal an
 - Ein Student macht 0 bis n Prüfungen
 - Eine Anmeldung führt zu einer Prüfung
 - Für eine Veranstaltung können n Prüfungsanmeldungen existieren
 - Mehrere Veranstaltungen werden zu einem Typ zusammengefasst
 - Ein Veranstaltungstyp hat einen festgelegten Creditpoint-Satz
 - Dieser Creditpoint-Satz gibt die Bewertungen mehrerer Prüfungen an

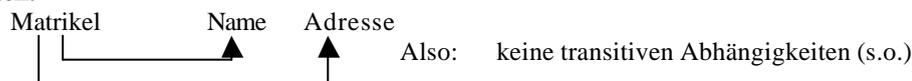
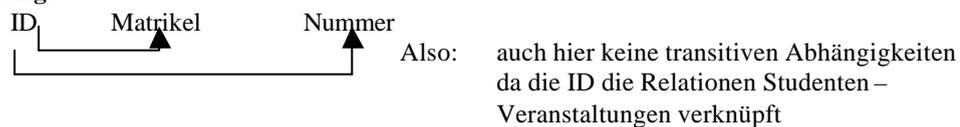
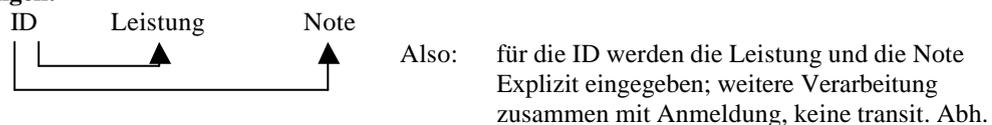
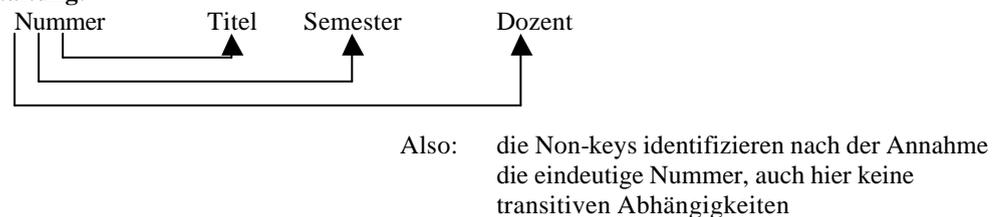
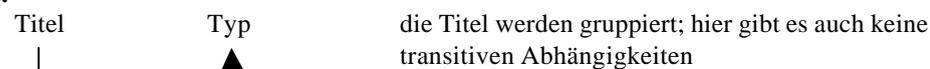
Übung 2

DATENDIKTIONÄR:

Relation	Attribute	Datentyp	Primärschlüssel	Fremdschlüssel
Studenten	Matrikel Name Adresse	Integer (not null) VarChar VarChar	Matrikel	
Anmeldung	ID Matrikel Nummer	Integer (not null) Integer (not null) Integer (not null)	ID	Matrikel (Studenten) Nummer (Veranstalt.)
Pruefungen	ID Leistung Note	Integer (not null) Boolean (not null) Integer	ID	ID (Anmeldung)
Veranstaltung	Nummer Titel Semester Dozent	Integer (not null) VarChar (not null) Integer (not null) VarChar (not null)	Nummer	Titel (Gruppe)
Gruppe	Titel Typ	VarChar (not null) Integer (not null)	Titel	Typ (Credits)
Credits	Typ CDs	Integer (not null) Integer (not null)	Typ	CDs (Pruefungen)

Aufgabe 2:

Hier die Beweise, daß die Relationen in der 3.NF sind (\Rightarrow 2.NF \Rightarrow 1.NF):

✓ **Studenten:**✓ **Anmeldung:**✓ **Pruefungen:**✓ **Veranstaltung:**✓ **Gruppe:**✓ **Credits:**

Übung 2

Beweise für BCNF, 4.NF und 5.NF:

Wie aus den Annahmen und den obigen Beweisen hervorgeht, gibt es immer nur einattributive Kandidatenschlüssel in allen Relationen:

Studenten: Matrikel

Anmeldung = Pruefungen: ID

Veranstaltung = Nummer

Gruppe = Titel

Credits = Typ

Von diesen einattributigen Kandidatenschlüsseln sind jeweils mehrere Attribute funktional abhängig

=> Jeder dieser Kandidatenschlüssel ist eine Determinate

=> alle Relationen sind in der Boyce-Codd-NF

Für die 4. NF:

Studenten: keine mehrwertigen Abhängigkeiten (eindeutige Identifikation über Kandidatenschl.)

Anmeldung: keine mehrwertigen Abhängigkeiten (eindeutige Identifikation über Kandidatenschl.)

Pruefungen: keine mehrwertigen Abhängigkeiten (eindeutige Identifikation über Kandidatenschl.)

Veranstaltungen: nach Annahmen ergibt sich, daß die Nummer eindeutig durch die Kombination der drei anderen Attribute zugewiesen wird => keine mehrwertigen Abhängigkeiten

Gruppe: einzige Relation mit mehrwertigen Abhängigkeiten; aber keine unabhängige mehrwertige Abhängigkeiten innerhalb des Schlüssels => in 4. NF

Credits: keine mehrwertigen Abhängigkeiten

Für die 5. NF werden nur die möglicherweise betroffenen Relationen betrachtet:

Pruefungen: hier könnte eine arithmetrische Verknüpfung zwischen Leistung und Note bestehen: In der Annahme wurde allerdings festgelegt, daß die Werte explizit eingegeben werden => unter Annahme in 5. NF

Oberrelation Lehre:

Durch die Aufteilung in die drei Teilrelationen ist eine arithmetrische Verknüpfung zwischen Attributen überflüssig geworden.

Aufgabe 3: SQL – Abfragen

Hier sind die realisierten, im weiteren Verlauf verwendeten Tabellen:

Studenten:

Matrikel	Name	Adresse
1	Hans Müller	Marsweg 2
2	Hans Müller	Highway 34
3	Frieda Busch	Badallee 23
4	Gerd Weih	Badallee 23
5	Herbert Bach	zu Hause 3
6	Hertha Berlin	Grünstraße 2
7	Gustav Breuz	Dresalweg 6
8	Aladin Memet	Halsstraße 32

Übung 2

Anmeldung:

ID	Matrikel	Nummer
1	1	1
2	1	4
3	1	3
4	2	4
5	3	1
6	7	1
7	4	2
8	8	8
9	6	1
10	7	9

Prüfungen:

ID	Leistung	Note
1	Nein	5
2	Ja	4
3	Ja	2
4	Ja	1
5	Nein	5
6	Ja	3
7	Ja	4
8	Nein	5
9	Ja	1
10	Ja	2

Veranstaltungen:

Nummer	Titel	Semester	Dozent
1	Softwareentwicklung	W 99	Lenz
2	Programmierung	S 00	Suhl
3	Netzwerke	W 00	Lenz
4	Softwareentwicklung	S 00	Lenz
5	Programmierungsumgebungen	W 01	Suhl
6	Projektmanagement	S 00	Suhl
7	Software-Engeniering	W 99	Suhl
8	Datenbanken	S 99	Lenz
9	Datenbanken	W 00	N. N.

Gruppe:

Titel	Typ
Datenbanken	Datenadministration
Netzwerke	Netze
Programmierungsumgebungen	Software
Programmierung	Software
Projektmanagement	Management
Software-Engeniering	Software
Softwareentwicklung	Software

Credits:

Übung 2

SS 2000, Lenz / Neiling

Michael Stollberg, # 344 94 10

Typ	CDs
Datenadministration	15
Management	7
Netze	20
Software	10

Als nächstes werden die beiden Hilfsabfragen Bonus und Malus generiert. Hier wird - dank meiner persönlichen Unkenntnis des Creditpointsystems, festgelegt, daß eine bestandene Prüfung mit den CDs der Veranstaltung positiv, eine nicht bestandene Prüfung negativ bewertet wird.

ANMERKUNG: Die eckigen Klammern werden von Access automatisch beim Zwischenspeichern erzeugt !!
Auf grund der NF- Realisation ergibt sich eine etwas komplexere Abfrageverschachtelung:

Bonus1:

```
SELECT [Pruefungen].[ID] AS BonusID, [Note]
FROM Pruefungen
WHERE Leistung=True;
```

BonusMatrikel:

```
SELECT [Anmeldung].[Matrikel], [Anmeldung].[ID] AS BonusID, [Bonus1].[Note]
FROM Anmeldung, Bonus1
WHERE [Anmeldung].[ID]=[Bonus1].[BonusID];
```

BonusName:

```
SELECT DISTINCT [Studenten].[Matrikel], [Name], [Adresse], [Nummer], [Note]
FROM Studenten, Anmeldung, BonusMatrikel
WHERE [Studenten].[Matrikel]=[BonusMatrikel].[Matrikel] And
[Anmeldung].[ID]=[BonusMatrikel].[BonusID];
```

BonusNameTitel:

```
SELECT [Matrikel], [Name], [Adresse], [Titel], [Note]
FROM Veranstaltung, BonusName
WHERE [BonusName].[Nummer]=[Veranstaltung].[Nummer];
```

BonusNameTyp:

```
SELECT [Matrikel], [Name], [Adresse], [BonusNameTitel].[Titel], [Typ],
[Note]
FROM Gruppe, BonusNameTitel
WHERE [BonusNameTitel].[Titel]=[Gruppe].[Titel];
```

Bonus:

```
SELECT [Matrikel], [Name], [Adresse], [Titel], [Note], [CDs]
FROM Credits, BonusNameTyp
WHERE [BonusNameTyp].[Typ]=[Credits].[Typ];
```

=> Ergebnis:

Matrikel	Name	Adresse	Titel	Note	CDs
1	Hans Müller	Marsweg 2	Netzwerke	2	20
1	Hans Müller	Marsweg 2	Softwareentwicklung	4	10
2	Hans Müller	Highway 34	Softwareentwicklung	1	10
4	Gerd Weih	Badallee 23	Programmierung	4	10
6	Hertha Berlin	Grünstraße 2	Softwareentwicklung	1	10
7	Gustav Breuz	Dresalweg 6	Softwareentwicklung	3	10
7	Gustav Breuz	Dresalweg 6	Datenbanken	2	15

Übung 2

SS 2000, Lenz / Neiling

Michael Stollberg, # 344 94 10

Die Abfrageschachtelung für Malus ist identisch, => Ergebnis Malus:

Matrikel	Name	Adresse	Titel	Note	Malus
1	Hans Müller	Marsweg 2	Softwareentwicklung	5	-10
3	Frieda Busch	Badallee 23	Softwareentwicklung	5	-10
8	Aladin Memet	Halsstraße 32	Datenbanken	5	-15

Erste Abfrage: Welche Studenten mit wie viel Malus-Punkten haben sich zur Prüfung 'Softwareentwicklung' im WS 99/00 angemeldet?

Unterabfrage Aufgabe1_1:

```
SELECT Matrikel, Name, Adresse
FROM Studenten
WHERE Matrikel IN
(SELECT Matrikel
FROM Anmeldung
WHERE Nummer =
(SELECT Nummer
FROM Veranstaltung
WHERE Titel = 'Softwareentwicklung' AND Semester = 'W 99'));
```

Ergebnis-Abfrage:

```
SELECT Aufgabel_1.Matrikel, Aufgabel_1.Name, Aufgabel_1.Adresse, sum(Malus)
AS Maluspunkte
FROM Aufgabel_1 LEFT JOIN Malus ON Aufgabel_1.Matrikel = Malus.Matrikel
GROUP BY Aufgabel_1.Matrikel, Aufgabel_1.Name, Aufgabel_1.Adresse;
```

=> Ergebnis:

Matrikel	Name	Adresse	Maluspunkte
1	Hans Müller	Marsweg 2	-10
3	Frieda Busch	Badallee 23	-10
6	Hertha Berlin	Grünstraße 2	
7	Gustav Breuz	Dresalweg 6	

Zweite Abfrage: Bonuspunkte von Hans Müller

BEACHTTE: es gibt 2 Hans Müller (daher zwei Matrikelnummern) !!

Ergebnis-Abfrage:

```
SELECT Matrikel, Name, Adresse, Titel, CDs AS Bonuspunkte
FROM Bonus
WHERE Name = 'Hans Müller' ;
```

=> Ergebnis:

Matrikel	Name	Adresse	Titel	Bonuspunkte
1	Hans Müller	Marsweg 2	Netzwerke	20
1	Hans Müller	Marsweg 2	Softwareentwicklung	10
2	Hans Müller	Highway 34	Softwareentwicklung	10

Dritte Abfrage: vergleichbare Veranstaltungen zu 'Softwareentwicklung'

Ergebnis-Abfrage:

```
SELECT Titel
FROM Gruppe
WHERE Typ = 'Software' ;
```

=> Ergebnis:

Titel
Softwareentwicklung
Programmierung
Programmierungsumgebungen
Software-Engeniering