Seminar 7

Aufgabe 6

In der folgenden Relation speichern wir die Betreuer für jeden Student. Ein Student kann aber gleichzeitig mehrere Hauptfächer studieren (um am Ende mehrere Diplomen zu kriegen, z.B. Mathematik und Informatik). Dafür muss er aber auch mehrere Diplomarbeite schreiben.

(StudentId, Hauptfach, Betreuer)

(100, Info, ‘B1’)

(100, Economie, ‘B2’)

1. Welche sind die Kandidatschlüssel?

StudentId – kann nicht KS sein, weil nicht eindeutig

(StudentId, Hauptfach) -> Betreuer Ja

(StudentId, Hauptfach) - minimal? Ja

**KS : (StudentId, Hauptfach)**

**(StudentId, Betreuer ) - KS**

Betreuer -> Hauptfach

Betreuer -> StudentId ? Nein

2

1. Ist die Relation in 1NF, 2NF, 3NF, BCNF? Erkläre.

1NF – Ja, die Attr sind atomar

2NF – Ja

Nichtschlüsselattribute (nicht prim): keine

3NF – Ja

Betreuer -> Hauptfach (Hauptfach – prim)

BCNF - Nein

Betreuer -> Hauptfach - verletzt BCNF

**Aufgabe 7**

**Sei die Schema R(A, B, C, D, E) mit folgenden fkt. Abh.:**

**F = {AB → CDE, AC → BDE, B → C, C → B, C → D, B → E}**

1. **Finde alle Kandidatschlüssel der Relation R.**
2. **Berechne die kanonische Überdeckung von F.**
3. **Ist R in BCNF? Erkläre.**
4. **Finde eine verlustlose BCNF Zerlegung von R.**
5. **Ist die Zerlegung von Punkt 4. abhängigkeitsbewahrend(toate dependentele initiale se regasesc si in descompunerea in BCNF)? Erkläre.B->**
6. **Ist R in 3NF? Erkläre.**
7. **Berechne mithilfe des Synthesealgorithmus eine 3NF Zerlegu2ng von R.**

1.

A+, B+, C+ , D+, E+ , AB, …. , BC, BD,…

Bem. A wird von keiner Attributenmenge bestimmt => A gehört zu dem KS

(Cheie candidat at cand o litera/combinatie de litere implica restul literelor)

A += A

**AB += ABCDE**

**AC += ACBDE**

AD += AD

AE += AE

ABC – nicht minimal (C redundant)

ADE += ADE

**KS: AB, AC**

**2.**

**Linksreduktion:**

AB → CDE, AC → BDE, B → C, C → B, C → D, B → E

AB → CDE <=> AB->C , AB-> D, AB ->E

B+ = BCDE (B->C, C->D, B->E)

C += CBDE (C->B, C->D, B->E)

**Rechtsreduktion:**

B → CDE, C →B**D**E, B → C, C → B, C → D, B → E

B-> C , B->CDE

C+ = CDE (fara C->B)

(fara C->D ) C+ = CBED

(fara C->E ) C+ = CBDE

**“Leere” Abh.**

B → CDE, C →B

**Vereinigungregel**

**FC = {**B → CDE, C →B**}**

Ergebnis 2: B → C, C →BDE,

Ergebnis 3: B → CE, C →BD,

1. **Ist R in BCNF?**

C-> B , C- nicht Superschl - verletzt BCNF

1. **Zerlegung in BCNF**

Alpha -> beta verletzt BCNF, R1 = R-beta, R2 = alpha + beta

C->B

R1(**A, C**, D, E) , AC-> DE, C->D (C->D verletzt BCNF)

… (weiter mit Zerlegung)

R2(**B**, C) , C->B , B->C , KS: B und separat C

Met II

**AB → CDE, AC → BDE, B → C, C → B, C → D, B → E**

B-> C – verletzt BCNF

R1(**A, B**, D, E), AB-> DE, B->E

B->E verletzt BCNF

R11 (**A,B**,D), AB->D

R12 (**B**, E) , B->E

R2(B, C) , B->C, C->B , KS: B und C - ist in BCNF

Zerlegung: R11, R12 und R2

Verlustlos – Ja

Abhängigkeitsbewahrend - Nein

**6. 3NF ?**

(cautam o relatie in care o non-cheie implica o cheie)

**FC = {**B → CDE, C →B**}**

C->D verletzt 3NF

**7.**

Synthesealg:

1. se calculeaza kanonische Uberdeckung
2. Se creeaza cate o relatie pt fiecare dependenta
3. Daca niciuna din cheile candidat nu sunt incluse in relatii, adaugam inca o relatie – ex. R3
4. Verificam daca exista o relatie inclusa in alta relatie – R2 dispare

**FC = {**B → CDE, C →B**}**

R1(B, C, D, E) , F1 = {B->CDE , C->B }, KS: **B** (B+=BCDE), **C** (C+=CBDE) (relativ la F1)

R2(B,C) F2 = {C->B, B->C} --stergem pentru ca R2 este inclus in R1

R3(A,B)