

1 Zerlegungsalgorithmus

1. Wandeln Sie das folgende relationale Schema mithilfe des Zerlegungsalgorithmus in BCNF um.

$\mathcal{R} : [A, B, C, D, E, F] \quad BA \rightarrow FE \quad F \rightarrow AB \quad A \rightarrow CD$

Zuerst bestimmen Sie die Kandidatenschlüssel und identifizieren Sie alle FDs, die gegen die BCNF verstoßen. Bitte geben Sie für jeden Schritt die Zwischenergebnisse an.

2. Ist Ihre Zerlegung abhängigkeiterhaltend?

2 Erstellen von Tabellen

Betrachten Sie das folgende relationale Schema, das offensichtlich nicht die Standard-Situation an der TU Wien beschreibt.

Wir nehmen an, dass Tutor:innen für eine oder mehrere Lerngruppen verantwortlich sind, Student:innen individuell (nicht pro Gruppe) Lösungen für Übungsblätter einreichen und individuelle Noten in Form der Anzahl der erreichten Punkte pro Blatt erhalten. Einige der Tutor:innen sind erfahrener (senior) als andere.

student: {[sid: int, firstname: string, lastname: string, semester: int, birthdate: date]}

tutor: {[tid: int, firstname: string, lastname: string, issenior: boolean]}

studygroup: {[gid: int, tid \rightarrow tutor, weekday: string, room: string, starttime: time]}

exercisheet: {[eid: int, maxpoints: int]}

handsin: {[sid \rightarrow student, eid \rightarrow exercisheet, achievedpoints: int]}

member: {[sid \rightarrow student, gid \rightarrow studygroup]}

Bitte formulieren Sie geeignete SQL-Statements zur Erstellung dieser 6 Tabellen unter Verwendung von:

- Sequence number generators, deren Werte automatisch erhöht werden, wenn Daten eingefügt werden
- Geeigneten Datentypen
- Primär- und Fremdschlüsseln

Hinweis: *Es gibt Unterschiede in der Implementierung des Datentyps "time", z.B. kennt DB2 und MySQL "time" als Datentyp, PostgreSQL kennt "time without time zone" und "time with time zone", usw. Daher werden diese Alternativen in den Lösungen als äquivalent betrachtet.*

3 Abfragen von Tabellen

Übersetzen Sie die folgenden Abfragen in äquivalente SQL-Statements, die auf den oben erstellten Tabellen ausgeführt werden können.

1. Finden Sie die Vor- und Nachnamen aller Studierenden, die sich derzeit im zweiten, dritten oder vierten Semester befinden.
2. Finden Sie alle verschiedenen Vornamen von Studierenden und Tutor:innen, deren Nachnamen mit einem 'P' beginnen. *Hinweis: Sie können testen, ob das Attribut test mit 'A' beginnt, indem Sie "test LIKE 'A%'" verwenden.*
3. Finden Sie heraus, wie erfahren (senior) der/die Tutor:in der Übungsgruppe 1 ist.
4. Finden Sie die IDs aller Studierenden, die mindestens 10 Punkte für Übungsblatt 1 erreicht haben.
5. Geben Sie die IDs aller Studierenden und deren erreichten Punkte für Übungsblatt 1 in absteigender Reihenfolge nach der Anzahl der erreichten Punkte aus.
6. Finde die IDs aller Studierenden, die vor dem 01.03.1992 geboren wurden und denselben Vornamen wie ein:e Tutor:in haben.

4 Tabellenmanipulation

1. Füllen Sie die oben erstellten Tabellen, indem Sie mindestens ein gültiges Tupel pro Tabelle einfügen.
2. Formulieren Sie ein SQL-Statement, das die Vornamen aller Studierenden auf 'Jens' setzt, deren Nachname 'Doe' ist.
3. Löschen Sie alle Studierenden aus der Tabelle student, deren Vorname 'Tom' ist.