

1. Es soll das Entity Relationship – Modell einer Bibliothek erstellt werden (wobei n:m Beziehungstypen durch assoziative Objekt Typen ersetzt werden). In diesem Modell sollen folgende Informationen abgelegt werden:

Leseausweisnummer (eindeutig), Name, Geburtsdatum und Anschrift für alle Benutzer

Signatur (eindeutig), Titel, Autor(en), Verlag und ISBN für alle Bücher

Für sämtliche Ausleihvorgänge und Reservierungen ausgeliehener Bücher soll das Datum der Ausleihe bzw. der Reservierung erfasst werden, für Reservierungen zusätzlich das Datum der Rückgabe des Buchs.

Objekttypen:

Benutzer, Attribute Leseausweisnummer (Primärschlüssel), Name, Geburtsdatum, Anschrift

Buch: Attribute Signatur (Primärschlüssel), Titel, Autor(en), Verlag, ISBN

Beziehungstypen:

ausleihen: 1:n – Beziehungstyp zwischen Benutzer und Buch, Attribut: Ausleihdatum. Umsetzung (ohne Betrachtung der Historie)

durch Fremdschlüssel Leseausweisnummer (Referenz auf Benutzer) und Attribut Ausleihdatum für Objekttyp Buch

oder

durch assoziativen Objekttyp Ausleihe: Attribute Leseausweisnummer (Fremdschlüssel => Benutzer), Signatur (Fremdschlüssel => Buch) (Signatur Primärschlüssel, falls keine Historie geführt wird), Ausleihdatum. (Das Ausleihdatum ist Teil des Primärschlüssels, falls die Historie mit betrachtet wird.)

reservieren: n:m – Beziehungstyp zwischen Benutzer und Buch, Attribute: Reservierungsdatum, Rückgabedatum

Assoziativer Objekttyp Reservierung: Attribute Leseausweisnummer (Fremdschlüssel => Benutzer), Signatur (Fremdschlüssel => Buch) (beide zusammen Primärschlüssel, falls keine Historie geführt wird und nur eine Reservierung pro Benutzer und Buch möglich ist), Reservierungsdatum, Rückgabedatum.

2. Gehen Sie von der Universitätsdatenbank aus der Vorlesung aus und formulieren Sie folgende Abfragen in der relationalen Algebra und in SQL:

Bestimmen Sie Personalnummer und Namen sämtlicher Assistenten.

Bestimmen Sie Personalnummer und Name sämtlicher Assistenten mit Fachgebiet Ideenlehre.

Bestimmen Sie Personalnummer und Namen der Assistenten von Professor Sokrates.

Bestimmen Sie die Matrikelnummer und den Namen der Studenten, die von Professor Sokrates geprüft wurden.

Bestimmen Sie Matrikelnummer und Namen der Studenten, die keine Prüfung abgelegt haben.

Formulieren Sie folgende Abfragen in SQL:

Bestimmen Sie pro Fachgebiet die Anzahl der Assistenten.

Bestimmen Sie die Daten der Studenten sortiert nach Semester (absteigend) und Namen (aufsteigend).

Relationale Algebra:

Projektion; Assistenten; Spalten Personalnummer, Name

Restriktion; Assistenten; Bedingung: Fachgebiet='Ideenlehre' => R1
Projektion; R1; Spalten PersNr, Name

Restriktion; Professoren; Bedingung name='Sokrates' => R1
Natural Join; R1, Assistenten; Bedingung: Assistenten.boss=R1.PersNr => R2
Projektion; R2; Spalten Personalnummer, Name

Restriktion; Professoren; Bedingung name='Sokrates' => R1
Natural Join; R1, pruefung; Bedingung: R1.PersNr=pruefung.persnr => R2
Projektion; R2; Spalte MatrNr => R3
Natural Join; R3, Studenten; Bedingung: R3.MatrNr=Studenten.MatrNr => R4
Projektion; R4; Spalten MatrNr, Name

Bestimmung der Studenten, die eine Prüfung abgelegt haben:
Natural Join; pruefung, Studenten; Bedingung: pruefung.MatrNr=Studenten.MatrNr
=> R1
Projektion; R1; Spalten MatrNr, Name
Bestimmung der Studenten, die keine Prüfung abgelegt haben:
Projektion; Studenten; Spalten MatrNr, Name => R2
Differenz; R2 \ R1

SQL:

```
Select PersNr, Name  
From Assistenten;
```

```
Select PersNr, Name  
From Assistenten  
Where fachgebiet='Ideenlehre';
```

```
Select a.Persnr, a.Name  
From Assistenten a, Professoren p  
Where a.boss=b.PersNr  
And b.Name='Sokrates';
```

```
Select s.MatrNr, s.Name  
From Studenten s, pruefung sp, Professoren p  
Where s.MatrNr=sp.MatrNr  
And sp.VorlNr=p.VorlNr  
And p.name='Sokrates';
```

```
Select matrnr, name  
From Studenten  
minus  
Select distinct s.MatrNr, s.Name  
From Studenten s, pruefung sp  
Where s.MatrNr=sp.MatrNr;
```

```
Select fachgebiet, count(*)
```

From Assistenten
Group by fachgebiet;

Select *
From Studenten
Order by semester desc, Name;

3. Gegeben sei eine Tabelle Tab mit Spalten (s1,s2,s3,s4,s5) und (s1,s2) als einzigem Schlüsselkandidaten.

Folgende funktionale Abhängigkeiten seien gegeben:

(s1,s2) -> s3

(s1,s2) -> s4

(s1,s2) -> s5

s1 -> s4

s1 -> s5

s4 -> s5

Ist die Tabelle in 2NF / 3NF? Was kann man zwecks Normalisierung tun?

s1 -> s4 ist eine funktionale Abhängigkeit eines Nichtschlüsselattributs (s4) von einem Teil des Primärschlüssels (s1,s2). Folglich ist 2NF nicht erfüllt. Damit ist 3NF ebenfalls nicht erfüllt.

3NF scheitert auch an der funktionalen Abhängigkeit des Nichtschlüsselattributs s5 vom Nichtschlüsselattribut s4.

Normalisierung:

Auflösung in Tabellen Tab1 (s1,s2,s3) und Tab2 (s1,s4,s5).

Tab1 ist in 2NF und in 3NF, da es nur ein Nichtschlüsselattribut gibt und damit Abhängigkeiten eines Nichtschlüsselattributs von einem anderen unmöglich sind.

Tab 2 ist in 2NF, da der Primärschlüssel s1 nur aus einem Attribut besteht, aber nicht in 3NF wegen der funktionalen Abhängigkeit s4 -> s5.

Normalisierung:

Tab21 (s1,s4), Tab22 (s4,s5).

Beide sind in 2NF, da der Primärschlüssel s1 bzw. s4 jeweils nur aus einem Attribut besteht.

Beide sind in 3NF, da es jeweils nur ein Nichtschlüsselattribut gibt.

Der Natural Join von Tab1, Tab21 und Tab22 ergibt Tab, also ist diese Zerlegung mit keinem Informationsverlust verbunden.

4. Erläutern Sie kurz den Begriff ACID für Transaktionen.

A – atomic; eine Transaktion wird ganz oder gar nicht ausgeführt.

C – konsistenzertretend; sind die von einer Transaktion veränderten Daten vor der Transaktion konsistent und ist die Transaktion korrekt, so sind die Daten nach der Transaktion in einem konsistenten Zustand.

I – isolated; während der Transaktion sieht der Benutzer die betroffenen Daten, als ob er alleine (ohne konkurrierende Zugriffe) arbeiten würde.

D- durable: Änderungen, die in einer Transaktion durchgeführt wurden, deren Commit vom Datenbanksystem bestätigt wurde, sind in der Datenbank festgeschrieben und stehen allgemein zur Verfügung.

5. Erläutern Sie die Begriffe Inner Join, Outer Join und Semi Join an einem selbstgewählten Beispiel auf Grundlage des Universitätsdatenmodells.

Betrachten wir die Tabellen Professoren und Assistenten.

Der Inner Join (=Natural Join) liefert die Daten der Professoren und der zugehörigen Assistenten bzw. umgekehrt. Daten von Professoren ohne Assistenten werden nicht

geliefert! Assistenten ohne Professoren gibt es nicht, würden aber auch nicht geliefert.

Der Left Outer Join Professoren left outer join Assistenten on Professoren.PersNr = Assistenten.Boss liefert zusätzlich die Daten der Professoren ohne Assistenten. Der Right Outer Join bzw. der Full Outer Join würden alternativ bzw. zusätzlich die Daten der Assistenten ohne Professoren liefern, was hier irrelevant ist.

Der Semi Join ist die Projektion des Inner Join auf eine der beteiligten Tabellen, d.h., es interessiert nur die Fragestellung, ob es zu einer Zeile dieser Tabelle eine zugehörige Zeile in der anderen Tabelle gibt, nicht, welche es gibt. Beispiel: Bestimme die Professoren, die Assistenten haben.

6. Welche Integritätsregeln gibt es im relationalen Datenmodell? Skizzieren Sie kurz den Sinn der Integritätsregeln.

Entity Integrity: Kein Teil eines Schlüsselkandidaten darf den Wert NULL annehmen. Dies ist eine Selbstverständlichkeit, denn der Schlüsselkandidat identifiziert eine Zeile einer Tabelle und muß daher zumindest bekannt sein.

Referentielle Integrität: Ein Fremdschlüsselwert kommt entweder als Wert des zugehörigen Schlüsselkandidaten vor, oder sämtliche Attribute sind NULL. Dies bedeutet, dass ein Fremdschlüsselwert zwar NULL sein darf (im Falle einer optionalen Beziehung), aber ansonsten keinen Wert annehmen darf, zu dem es kein übergeordnetes Objekt gibt.

Business Rules: Geschäftsregeln sollen in der Datenbank implementiert werden können. Damit sollen Anwendungen weniger komplex werden.