

Gruppe 1:

1. Erstellen Sie ein Entity Relationship Diagramm für die folgende Situation, wobei assoziative Objekttypen verwendet werden, falls dies sinnvoll erscheint:

Es soll ein Unternehmen beschrieben werden, das Segelboote verleiht. Dabei werden pro Kunde eine interne Kunden Id (eindeutig), die Mitgliedsnummer im Verband der Segler (eindeutig), der Kundename, die Adresse und die Kommunikationsverbindung gespeichert (jeweils als ein String), ferner die Qualifikation als Segler (als Integer). Pro Boot werden eine interne Id (eindeutig), der Bootsname (eindeutig) sowie eine Beschreibung (als String) gespeichert. Ferner soll festgehalten werden, welcher Kunde für welchen Zeitraum (Beginn, Ende) ein Boot ausleiht bzw. ausgeliehen hat.

Objekttypen:

Kunde, Attribute:

- Kunden Id (Primärschlüssel)
- Mitgliedsnummer (Alternativschlüssel)
- Kundename
- Adresse
- Kommunikationsverbindung
- Qualifikation

Boot, Attribute:

- Boot Id (Primärschlüssel)
- Bootsname (Alternativschlüssel)
- Beschreibung

Beziehung Kunde leiht Boot aus bzw. hat Boot ausgeliehen: 0..* - 0..* (Historie!)
=> assoziativer Objekttyp Ausleihe

Attribute:

- Kunden Id (Fremdschlüssel)
- Boot Id (Fremdschlüssel)
- Beginn
- Ende

Primärschlüssel: KundenId, BootId, Beginn

2. Ein Tennisverein will seine Punktspielaktivitäten in einer Datenbank festhalten. Ein Mitglied ohne große Kenntnisse in der Datenmodellierung schlägt vor, die Daten in einer Tabelle abzulegen, in der jedes Spiel jedes Spielers im Rahmen eines Punktspiels (jedes Einzel, jedes Doppel für beide Spieler, also zweifach) abgelegt ist und die folgende Attribute hat:

Mitgliedsnummer des Spielers (vom Verband vergeben, eindeutig pro Spieler)

Name des Spielers,

Geburtsdatum des Spielers,

Position (in der Mannschaft laut Meldeliste)

Position (beim Wettkampf)

Einsatzart (Einzel, Doppel)

Mannschaftsnummer (vom Verband vergeben, eindeutig pro Mannschaft)

Altersklasse der Mannschaft

Spieltag

Ergebnis (gewonnen / verloren)

Bsp.:

Spieler 10000, Hans Schmidt, geb. 1.7.1950, gemeldet an Position 1, gespielt an Position 1, im Einzel, in Mannschaft 1001, Altersklasse Herren 50, am 5.7.2003, gewonnen

Spieler 10000, Hans Schmidt, geb. 1.7.1950, gemeldet an Position 3, gespielt an Position 1, im Einzel, in Mannschaft 1000, Altersklasse Herren 40, am 6.7.2003, gewonnen

Spieler 10000, Hans Schmidt, geb. 1.7.1950, gemeldet an Position 3, gespielt an Position 2, im Doppel, in Mannschaft 1000, Altersklasse Herren 40, am 6.7.2003, verloren

Spieler 20000, Dieter Schneider, geb 7.1.1960, gemeldet an Position 9, gespielt an Position 2, im Doppel, in Mannschaft 1000, Altersklasse Herren 40, am 6.7.2003, verloren

Gehen Sie davon aus, dass ein Spieler an einem Spieltag nur in einer Mannschaft spielen darf (Einzel und / oder Doppel).

Begründen Sie, dass die Attributkombination (Mitgliedsnummer des Spielers, Spieltag, Einsatzart) Primärschlüssel der o.g. Tabelle ist.

Dies ist genau die Umsetzung der Regel, dass ein Spieler an einem Spieltag nur ein Einzel und ein Doppel spielen darf.

Welche funktionalen Abhängigkeiten sehen Sie?

Funktionale Abhängigkeit vom Primärschlüssel

(1) Mitgliedsnummer des Spielers =>

Name des Spielers,
Geburtsdatum des Spielers,

(2) Mannschaftsnummer =>

Altersklasse der Mannschaft

(3) Mitgliedsnummer Spieler, Mannschaftsnummer =>

Position (des Spielers in der Mannschaft laut Meldeliste)

Geben Sie eine möglichst einfache Definition für die zweite und dritte Normalform. Ausgehend davon, welche Verstöße gegen die zweite und dritte Normalform sehen Sie?

2NF: Kein Nichtschlüsselattribut darf von einem Teil eines Schlüsselkandidaten abhängen

3NF: Kein Nichtschlüsselattribut darf von einer Attributkombination mit anderen Nichtschlüsselattributen abhängen

(1) => Verstoß gegen 2NF

(2) => Verstoß gegen 3NF

(3) => Verstoß gegen 3NF

Wie können Sie das Datenmodell normalisieren?

Tabelle Spieler, Attribute:

- Mitgliedsnummer des Spielers
- Name des Spielers
- Geburtsdatum des Spielers

Tabelle Mannschaft, Attribute:

- Mannschaftsnummer
- Altersklasse der Mannschaft

Tabelle Meldeposition:

- Mitgliedsnummer des Spielers
- Mannschaftsnummer
- Position (des Spielers in der Mannschaft laut Meldeliste)

In Originaltabelle verbleiben die Attribute:

- Mitgliedsnummer des Spielers
- Position (beim Wettkampf)
- Einsatzart (Einzel, Doppel)
- Mannschaftsnummer
- Spieltag
- Ergebnis

3. Gegeben sei das Universitätsdatenmodell aus der Vorlesung. Formulieren Sie folgende Abfragen in der relationalen Algebra (ohne Subqueries!) und in SQL:

- Bestimmen Sie sämtliche Daten zu Studenten im ersten Semester.
- Bestimmen Sie Personalnummer und Name der Assistenten von Professor Sokrates.
- Bestimmen Sie Matrikelnummer und Name sowie die Note in der Prüfung für sämtliche Studenten, die eine Prüfung zur Vorlesung Wissenschaftstheorie abgelegt haben.
- Bestimmen Sie die Matrikelnummern zu Studenten, die keine Prüfung zur Vorlesung Wissenschaftstheorie abgelegt haben.
- Bestimmen Sie die Vorlesungsnummer aller Vorlesungen, die im ersten Semester gehört werden können.

Formulieren Sie folgende Abfragen in SQL:

- Bestimmen Sie die Personalnummer und die Anzahl der Assistenten für alle Professoren (die Assistenten haben).
Bonus: Wie sieht es aus, wenn auch Professoren ohne Assistenten zu berücksichtigen sind?
- Bestimmen Sie sämtliche Daten aller Professoren, sortiert zunächst nach Rang und dann nach Namen.

Relationale Algebra:

- Restriktion; Studenten; semester = 1

- ii) Restriktion; Professoren; name='Sokrates' => Res1
Inner Join; Res1, Assistenten; Res1.persnr = Assistenten.boss
- iii) Restriktion; Vorlesungen; titel='Wissenschaftstheorie' => Res1
Inner Join; Res1, prüfen; Res1.vorlnr = prüfen.vorlnr => Res2
Inner Join; Res2, Studenten; Res2.matnr = Studenten.matnr => Res3
Projektion; Res3; matnr, name
- iv) Projektion; Studenten; matnr => Res1
Restriktion; Vorlesungen; titel = Wissenschaftstheorie => Res2
Inner Join: Res2, prüfen; Res2.vorlnr = prüfen.vorlnr => Res3
Projektion; Res3; matnr => Res4
Differenz; Res1 – Res4
- v) Bestimmung der Vorlesungen, die keine Vorgänger haben:
Projektion; Vorlesungen; vorlnr => Res1
Projektion; voraussetzen; nachfolger => Res2
Differenz: Res1 – Res2 => Res3

SQL:

- i)

```
select *
from Studenten
where semester = 1;
```
- ii)

```
select a.*
from Professoren p, Assistenten a
where p.persnr = a.boss
and p.name='Sokrates';
```
- iii)

```
select s.matnr, s.name
from Vorlesungen v, prüfen p, Studenten s
where v.vorlnr = p.vorlnr
and p.matnr = s.matnr
and v.title = 'Wissenschaftstheorie'
```
- iv)

```
select matnr
from Studenten
minus / except
select s.distinct s.matnr
from Vorlesungen v, prüfen p, Studenten s
where v.vorlnr = p.vorlnr
and p.matnr = s.matnr
and v.title = 'Wissenschaftstheorie'
```
- v)

```
select vorlnr
from Vorlesungen
minus / except
select nachfolger
from voraussetzen
```
- vi)

```
select persnr, count(*)
from Assistenten
group by persnr
```

Bonus:

```

Select p.persnr, count(a.persnr)
From Professoren p left outer join Assistenten a
    On p.persnr = a.boss
Group by p.persnr

```

```

vii) select *
from Professoren
order by rang, name desc

```

4. Welche Schichten werden in der Drei Schichten - Architektur von Datenbanksystemen betrachtet?

Was versteht man unter physischer und logischer Datenunabhängigkeit?

Wie begründet man in der Drei Schichten – Architektur, dass physische und logische Datenunabhängigkeit erreicht werden?

Physische Schicht

Logische Schicht

Anwendungsschicht

Physische Datenunabhängigkeit:

Unabhängigkeit der Anwendungen von der physischen Schicht

Begründung: Abschirmung der Anwendungsschicht von der physischen Schicht durch die logische Schicht

Logische Datenunabhängigkeit:

Unabhängigkeit der Anwendungen von der logischen Schicht

Begründung: Abschirmung der Anwendungsschicht von der logischen Schicht durch Zugriffsschicht

5. Es gibt drei Möglichkeiten, eine 0..1 – 0..1 – Beziehung in einer relationalen Datenbank abzubilden, z.B. Mitarbeiter – Dienstwagen:

- i) Eine Tabelle mit allen Attributen
- ii) Zwei Tabellen Mitarbeiter und Dienstwagen, Fremdschlüssel Mitarbeiter Id in Tabelle Dienstwagen
- iii) Zwei Tabellen Mitarbeiter und Dienstwagen, Fremdschlüssel Dienstwagen Id in Tabelle Mitarbeiter
- iv) Drei Tabellen, Mitarbeiter, Dienstwagen und Assoziation “Welcher Mitarbeiter fährt welchen Dienstwagen?”

Welche dieser Varianten wenden Sie an, wenn es viele Mitarbeiter ohne Dienstwagen gibt und

- a. ein Dienstwagen grundsätzlich einem Mitarbeiter zugeordnet ist
- b. Dienstwagen grundsätzlich als Pool Car verwendet werden?

Zuordnung

- a. – ii) Der Dienstwagen bezieht sich i.a. auf einen Mitarbeiter, also ist FK adäquate Abbildung.
- b. – iv) Es gibt viele Mitarbeiter ohne Dienstwagen und Dienstwagen ohne fest zugeordneten Mitarbeiter, also sind weder ii) noch iii) adäquate Abbildungen.

