

Übungen

# Aufgabe 1

- Wir haben folgende Ausprägung der Relation Studenten:

SID	Name	Email	Age	Note
2833	Jones	<a href="mailto:jones@scs.ubbcluj.ro">jones@scs.ubbcluj.ro</a>	19	9
2877	Smith	<a href="mailto:smith@scs.ubbcluj.ro">smith@scs.ubbcluj.ro</a>	20	8
2976	Jones	<a href="mailto:jones@math.ubbcluj.ro">jones@math.ubbcluj.ro</a>	21	10
2765	Mary	<a href="mailto:mary@math.ubbcluj.ro">mary@math.ubbcluj.ro</a>	22	7.7
3000	Dave	<a href="mailto:dave@cs.ubbcluj.ro">dave@cs.ubbcluj.ro</a>	18	5.5
3010	Smith	<a href="mailto:smith2@scs.ubbcluj.ro">smith2@scs.ubbcluj.ro</a>	20	7
3020	Sam	<a href="mailto:sam@scs.ubbcluj.ro">sam@scs.ubbcluj.ro</a>	19	9.5

1. Finde ein Attribut oder Menge von Attributen, die kein Kandidatschlüssel sein können, bzgl. dieser Ausprägung
2. Können wir einen Kandidatschlüssel für die Relation Studenten finden, wenn wir wissen dass diese Ausprägung gültig ist?

# Aufgabe 1 - Lösung

1. Nicht-Kandidatschlüssel: Name, Age

Note kann bzgl. dieser Ausprägung nicht als Nicht-Kandidatschlüssel identifiziert werden. In dieser Ausprägung ist Note eindeutig, obwohl wir wissen, dass es normalerweise nicht eindeutig ist.

2. Nein, wir können den Kandidatschlüssel basierend auf einer Ausprägung nicht bestimmen.

# Aufgabe 2

Die Datenbank enthält folgende Relationen:

- Schauspieler(ID, Name, Geburtsjahr)
- Filme(ID, Titel, Jahr), wobei Jahr = wann der Film produziert wurde
- Casting(FilmId, SchauspielerID, Charakter) – ein Schauspieler spielt ein Charakter in einem Film

Die Primärschlüssel sind unterstrichen.

Schreibe folgenden Anfragen in SQL und/oder relationale Algebra

# Aufgabe 2

1. Finde die Titeln der Filme produziert zwischen 1950 und 2000 (SQL+rel.A.)

```
SELECT [DISTINCT] Titel FROM Filme  
WHERE Jahr > 1950 AND Jahr < 2000
```

$$\Pi_{\text{Titel}} (\sigma_{\text{Jahr} > 1950 \wedge \text{Jahr} < 2000} (\text{Filme}))$$

# Aufgabe 2

3. Finde **die Namen** der Schauspieler, die “Fletcher Christian” spielen in einer Produktion des Films “Mutiny on the Bounty” und **das Produktionsjahr** des Films (SQL + rel.A.)

```
SELECT S.Name, F.Jahr
FROM Schauspieler S INNER JOIN Casting C
    ON S.ID = C.SchauspielerID
    INNER JOIN Filme F ON C.FilmId = F.ID
WHERE C.Charakter = 'Fletcher Christian'
    AND F.Titel = 'Mutiny on the Bounty'
```

# Aufgabe 2

## 3. Relationale Algebra:

$\rho_S(\text{Schauspieler})$

$\rho_C(\sigma_{\text{Casting.Character} = \text{'Fletcher Christian'}}(\text{Casting}))$

$\rho_F(\sigma_{\text{Filme.Titel} = \text{'Mutiny on the Bounty'}}(\text{Filme}))$

$\Pi_{S.Name, F.Jahr} (S \bowtie_{S.ID = C.SchauspielerID} C$   
 $\bowtie_{C.FilmID = F.ID} F)$

# Aufgabe 2

7. Finde die Paaer von unterschiedlichen Schauspielernamen, die denselben Charakter in unterschiedlichen Filmproduktionen desselben Films gespielt haben (SQL)

```
SELECT S1.Name, S2.Name
FROM Schauspieler S1 INNER JOIN Casting C1
  ON C1.SchauspielerID = S1.ID
INNER JOIN Filme F1 ON C1.FilmID = F1.ID,
Schauspieler S2 INNER JOIN Casting C2
  ON C2.SchauspielerID = S2.ID
INNER JOIN Filme F2 ON C2.FilmID = F2.ID
WHERE F1.Titel = F2.Titel
      AND F1.Jahr <> F2.Jahr
      AND C1.Charakter = C2.Charakter
      AND S1.Name <> S2.Name
```



# Aufgabe 2

8. Finde die Namen der Schauspieler, die zusammen mit Victoria Abril in einem Film gespielt haben und die geboren wurden, nachdem Victoria in dem ersten Film gespielt hat (SQL)

```
SELECT S1.Name FROM Schauspieler S1 INNER JOIN Casting C1
    ON S1.ID = C1.SchauspielerID ,
    Schauspieler S2 INNER JOIN Casting C2
    ON S2.ID = C2.SchauspielerID
WHERE S2.Name = 'Victoria Abril' AND
    C2.FilmId = C1.FilmID AND S1.ID <> S2.ID
AND S1.Geburtsjahr >
    (SELECT MIN(F.Jahr) FROM Filme F
    INNER JOIN Casting C ON C.FilmID = F.ID
    INNER JOIN Schauspieler S ON C.SchauspielerID = S.ID
    WHERE S.Name = 'Victoria Abril')
```

# Aufgabe 2

9. Finde die Namen der Schauspieler die 'Superman' oder 'Clark Kent' gespielt haben. (SQL + rel. Alg.)

```
SELECT DISTINCT S.Name
FROM Schauspieler S, Casting C
WHERE C.SchauspielerID = S.ID
      AND (C.Charakter = 'Superman'
      OR C.Charakter = 'Clark Kent')
```

# Aufgabe 2

## 9. Relationale Algebra:

$\Pi_{\text{Schauspieler.Name}} ((\text{Schauspieler} \bowtie$

$\sigma_{\text{Casting.Charakter} = \text{'Superman'} \vee \text{Casting.Character} = \text{'Clark Kent'}} (\text{Casting}))$

# Aufgabe 3

Die Datenbank enthält folgende Relationen:

- Lieferanten (LID, Lname, Adresse)
- Produkte (PID, Pname, Farbe)
- Katalog (LID, PID, Preis)

Die Primärschlüssel sind unterstrichen.

Katalog enthält die Preise für Produkte von unterschiedlichen Lieferanten.

Geben sie an, was die folgenden Anfragen ausgeben.

# Aufgabe 3

1.  $\pi_{\text{LName}}(\pi_{\text{LID}}((\sigma_{\text{Farbe} = \text{'rot'}} \text{Produkte}) \bowtie (\sigma_{\text{Preis} < 100} \text{Katalog})) \bowtie \text{Lieferanten})$

Finde die Lieferantennamen, die ein rotes Produkt liefern, das weniger als 100 kostet.

2.  $\pi_{\text{LName}}(\pi_{\text{LID}}((\sigma_{\text{Farbe} = \text{'rot'}} \text{Produkte}) \bowtie (\sigma_{\text{Preis} < 100} \text{Katalog}) \bowtie \text{Lieferanten}))$

Diese Anfrage gibt nichts aus. Nachdem wir auf das Attribut SID projizieren, ist das das einzige Attribut in der Menge. Dann können wir nicht mehr auf Lname projizieren.

# Aufgabe 3

5.  $(\pi_{\text{LID, PID}} \text{ Katalog}) \div \pi_{\text{PID}} \text{ Produkte}$

$R_1 \div R_2$  enthält alle  $x$ , sodass für jeder  $y$  in  $R_2$ , es gibt ein  $xy$  in  $R_1$ .  
Finde alle LIDs der Lieferanten die jeder Produkt liefern.

6.  $\rho_{R_1}(\text{Katalog})$  ,  $\rho_{R_2}(\text{Katalog})$

$$\pi_{R_1.PID}(\sigma_{R_1.PID = R_2.PID \wedge R_1.LID \neq R_2.LID} (R_1 \times R_2))$$

Finde die PIDs der Produkte, die von wenigstens zwei Lieferanten geliefert worden.

# Aufgabe 4

Um die Bücher in der Bibliothek zu speichern brauchen wir:

*BuchID, ISBN, Kategorie, SerialNr, KopieNr, Titel, Author*

Jedes Buch (jede Kopie) hat eine eindeutige Nummer (BuchID) den wir benutzen können um die Anleihe des Buches zu speichern. Diese Nummer ist einem Buch zugeordnet als ein Barcode, hat aber keine besondere Bedeutung.

ISBN ist eindeutig für eine Edition eines Buches.

Jedes Buch gehört zu einer Kategorie. In einer Kategorie haben die Bücher unterschiedliche SerialNr. Wenn wir mehrere Kopien von demselben Buch haben, dann werden diese durch den BuchID unterschiedet.

Welche der folgenden fkt. Abh. gelten? (Nicht aus dem Text, sonder logisch in dem beschriebenen Kontext) Gebe eine kurze Erklärung dafür.

# Aufgabe 4

1. BuchID  $\rightarrow$  ISBN, Kategorie, SerialNr, KopieNr

**Ja:** jedes Buch hat ein eindeutiges BuchId (BuchID ist ein Kandidatschlüssel)

2. ISBN  $\rightarrow$  BuchID

**Nein:** es kann mehrere Kopien von demselben Buch geben mit demselben ISBN

3. ISBN  $\rightarrow$  Kategorie, SerialNr

**Ja:** diese Bedingung erzwingt den Bibliothekar konsistent zu sein



# Aufgabe 4

4. ISBN  $\rightarrow$  KopieNr

**Nein:** : es kann mehrere Kopien von demselben Buch geben mit demselben ISBN

5. ISBN  $\rightarrow$  Titel, Author

**Ja:** Ein ISBN ist eindeutig für ein Buchedition, also es gibt nicht unterschiedliche Titel und Autoren für denselben ISBN

6. KopieNr  $\rightarrow$  ISBN, BuchID

**Nein:** die KopieNr kann nicht eindeutig das Buch identifizieren

# Aufgabe 5

Die Ausprägung einer Relation  $S(A, B, C)$  enthält folgende Tupeln:

(1,2,3)

(4,2,3)

(5,3,3)

1. Welche der folgenden fkt. Abh. können für die Relation  $S$  nicht gelten?

a)  $A \rightarrow B$

b)  $BC \rightarrow A$

c)  $B \rightarrow C$

2. Könnt ihr fkt. Abh. identifizieren, die für die Relation  $S$  gelten?

# Aufgabe 5 - Lösung

1. Welche der folgenden fkt. Abh. können für die Relation S nicht gelten?

a)  $A \rightarrow B$  —

b)  $BC \rightarrow A$  - kann nicht gelten wegen der Tupeln (1,2,3) und (4,2,3) :  
unterschiedliche Werte für A mit denselben Werten für BC

c)  $B \rightarrow C$  —

2. Könnt ihr fkt. Abh. identifizieren, die für die Relation S gelten?

**Nein.** Wenn wir sagen, dass die fkt. Abh. für die ganze Relation gilt, so muss diese für alle gültige Ausprägungen gelten.