

Übungen

Aufgabe 1

- Wir haben folgende Ausprägung der Relation Studenten:

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

1. Finde ein Attribut oder Menge von Attributen, die kein Kandidatschlüssel sein kann, bzgl. dieser Ausprägung
2. Können wir einen Kandidatschlüssel für eine Relation basierend auf eine gültige Ausprägung finden?

Aufgabe 1 - Lösung

1. Nicht-Kandidatschlüssel: Name (oder 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. Wir müssen immer die Semantik des Schemas berücksichtigen.

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 (oder mehrere) in einem Film

Die Primärschlüssel sind unterstrichen.

Schreibe folgenden Anfragen in SQL und/oder relationale Algebra

Aufgabe 2

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

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

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

2. Finde die Titel der Filme produziert nach 2000 oder bevor 1950(SQL+rel.A.)

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

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

Aufgabe 2

3. Finde **das Produktionsjahr** des Films und **die Namen** der Schauspieler, die "Fletcher Christian" in einer Produktion des Films "Mutiny on the Bounty" spielen (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

4. Finde die Namen der Schauspieler die sowohl 'Superman' als auch 'Clark Kent' gespielt haben (SQL + rel. Alg)

```
SELECT DISTINCT S1.Name
FROM Schauspieler S1 INNER JOIN Casting C1
  ON C1.SchauspielerID = S1.ID
  INNER JOIN Casting C2
    ON C2.SchauspielerID = S1.ID
WHERE C1.Charakter = 'Superman'
      AND C2.Charakter = 'Clark Kent'
```


Aufgabe 2

4. Relationale Algebra:

$\Pi_{\text{Schauspieler.Name, Schauspieler.Id}} (\text{Schauspieler}$
 $\bowtie_{\text{Schauspieler.ID} = \text{Casting.SchauspielerID}}$
 $\sigma_{\text{Casting.Character} = \text{'Superman'}} (\text{Casting})$
 \cap

$\Pi_{\text{Schauspieler.Name, Schauspieler.Id}} (\text{Schauspieler}$
 $\bowtie_{\text{Schauspieler.ID} = \text{Casting.SchauspielerID}}$
 $\sigma_{\text{Casting.Character} = \text{'Clark Kent'}} (\text{Casting})$

Aufgabe 2

5. Finde die Namen der Schauspieler die sowohl 'Superman' als auch 'Clark Kent' gespielt haben in derselben Produktion des Films (SQL)

```
SELECT DISTINCT S.Name
FROM Schauspieler S INNER JOIN Casting C1
    ON C1.SchauspielerID = S.ID
    INNER JOIN Casting C2
    ON C2.SchauspielerID = S.ID
WHERE C1.Charakter = 'Superman'
    AND C2.Charakter = 'Clark Kent'
    AND C1.FilmID = C2.FilmID
```

Aufgabe 2

6. Finde die Namen der Schauspieler die zwei unterschiedliche Charaktern in derselben Filmproduktion gespielt haben (SQL)

```
SELECT DISTINCT S.Name
FROM Schauspieler S INNER JOIN Casting C1
    ON C1.SchauspielerID = S.ID
    INNER JOIN Casting C2
    ON C2.SchauspielerID = S.ID
WHERE C1.Charakter <> C2.Charakter
    AND C1.FilmId = C2.FilmID
```

Aufgabe 2

7. Finde die Paare 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
INNER JOIN Filme F2 ON F1.Titel = F2.Titel
INNER JOIN Casting C2 ON C2.FilmID = F2.ID
INNER JOIN Schauspieler S2 ON C2.SchauspielerID = S2.ID
WHERE F1.Jahr <> F2.Jahr
      AND C1.Charakter = C2.Charakter
      AND S1.ID < S2.ID
```