

Lösung - Übungsblatt 2

(Modellierung von Datenquellen)

Fabian Panse

panse@informatik.uni-hamburg.de

Universität Hamburg



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG



Aufgabe 1: GaV mit Nebenbedingungen

Student(MNr, Name, Sem, SG)

IC: Sem > 5

S1(MNr, Name, Sem)

IC: SG = 'ITMC'

S2(MNr, Name, Sem, SG)

IC: Sem < 8

S3(MNr, Name, Sem)

IC: Sem > 12

S4(MNr, Name, SG)

S5(MNr, Name, Sem)

IC: SG = 'ITMC' oder SG = 'MCI'

Aufgabe 1: GaV mit Nebenbedingungen

Student(MNr, Name, Sem, SG)

IC: Sem > 5

```
CREATE VIEW Student AS
SELECT  MNr, Name, Sem, 'ITMC'
FROM    S1
WHERE   Sem > 5
UNION
SELECT  MNr, Name, Sem, SG
FROM    S2
WHERE   Sem > 5 (AND Sem < 8)
UNION
SELECT  MNr, Name, Sem, NULL AS SG
FROM    S3
WHERE   Sem > 12
UNION
SELECT  MNr, Name, Sem, NULL AS SG
FROM    S5
WHERE   Sem > 5
```

S1(MNr, Name, Sem)

IC: SG = 'ITMC'

S2(MNr, Name, Sem, SG)

IC: Sem < 8

S3(MNr, Name, Sem)

IC: Sem > 12

S4(MNr, Name, SG)

S5(MNr, Name, Sem)

IC: SG = 'ITMC' oder SG = 'MCI'

Aufgabe 1: GaV mit Nebenbedingungen

Student(MNr, Name, Sem, SG)

IC: Sem > 5

```
CREATE VIEW Student AS
SELECT  MNr, Name, Sem, 'ITMC'
FROM    S1
WHERE   Sem > 5
UNION
SELECT  MNr, Name, Sem, SG
FROM    S2
WHERE   Sem > 5 (AND Sem < 8)
UNION
SELECT  MNr, Name, Sem, NULL AS SG
FROM    S3
WHERE   Sem > 12
UNION
SELECT  MNr, Name, Sem, NULL AS SG
FROM    S5
WHERE   Sem > 5
```

S1(MNr, Name, Sem)

IC: SG = 'ITMC'

S2(MNr, Name, Sem, SG)

IC: Sem < 8

S3(MNr, Name, Sem)

IC: Sem > 12

S4(MNr, Name, SG)

S5(MNr, Name, Sem)

IC: SG = 'ITMC' oder SG = 'MCI'

- Für S4 kann die globale IC nicht geprüft werden
- ⇒ Quelle generell nicht nutzbar

Aufgabe 1: GaV mit Nebenbedingungen

Student(MNr, Name, Sem, SG)

IC: Sem > 5

```
CREATE VIEW Student AS
SELECT  MNr, Name, Sem, 'ITMC'
FROM    S1
WHERE   Sem > 5
UNION
SELECT  MNr, Name, Sem, SG
FROM    S2
WHERE   Sem > 5 (AND Sem < 8)
UNION
SELECT  MNr, Name, Sem, NULL AS SG
FROM    S3
WHERE   Sem > 12
UNION
SELECT  MNr, Name, Sem, NULL AS SG
FROM    S5
WHERE   Sem > 5
```

S1(MNr, Name, Sem)

IC: SG = 'ITMC'

S2(MNr, Name, Sem, SG)

IC: Sem < 8

S3(MNr, Name, Sem)

IC: Sem > 12

S4(MNr, Name, SG)

S5(MNr, Name, Sem)

IC: SG = 'ITMC' oder SG = 'MCI'

- Für S4 kann die globale IC nicht geprüft werden
⇒ Quelle generell nicht nutzbar
- Für S5 kann die lokale IC nicht modelliert werden
⇒ Quelle oft nicht brauchbar

Aufgabe 2: SQL → Datalog

SQL-Anfrage:

```
SELECT p1.Name AS N, f.Titel AS T
FROM Film f, Person p1, Person p2, Spielt s1, Spielt s2
WHERE f.Titel = s1.Film
AND f.Titel = s2.Film
AND p1.PID = s1.Person
AND p2.PID = s2.Person
AND p2.Ort = 'Hamburg'
AND f.Jahr = 2015;
```

Aufgabe 2: SQL → Datalog

SQL-Anfrage:

```
SELECT p1.Name AS N, f.Titel AS T
FROM Film f, Person p1, Person p2, Spielt s1, Spielt s2
WHERE f.Titel = s1.Film
AND f.Titel = s2.Film
AND p1.PID = s1.Person
AND p2.PID = s2.Person
AND p2.Ort = 'Hamburg'
AND f.Jahr = 2015;
```

Datalog-Anfrage:

```
q(N, T):- Film(T, Reg, G, J),
          Person(PID1, N, A1, O1), Person(PID2, N2, A2, O2),
          Spielt(T, PID1, R1), Spielt(T, PID2, R2),
          O2='Hamburg',J=2015
```

Aufgabe 2: SQL → Datalog

SQL-Anfrage:

```
SELECT p2.Name AS N
FROM Film f, Person p1, Person p2, Spielt s
WHERE f.Regisseur = p1.PID
AND (p1.Name = 'S. Leone' OR p1.Name = 'S. Kubrik')
AND f.Titel = s.Film
AND p2.PID = s.Person;
```


Aufgabe 2: SQL → Datalog

SQL-Anfrage:

```
SELECT p2.Name AS N
FROM Film f, Person p1, Person p2, Spielt s
WHERE f.Regisseur = p1.PID
AND (p1.Name = 'S. Leone' OR p1.Name = 'S. Kubrik')
AND f.Titel = s.Film
AND p2.PID = s.Person;
```

Datalog-Anfrage:

q(N):- Film(T, PID1, G, J),
Person(PID1, N1, A1, O1), Person(PID2, N, A2, O2),
Spielt(T, PID2, R1), N1='S. Leone'

q(N):- Film(T, PID1, G, J),
Person(PID1, N1, A1, O1), Person(PID2, N, A2, O2),
Spielt(T, PID2, R1), N1='S. Kubrik';

Aufgabe 2: SQL → Datalog

Erläuternde Information:

Obige SQL Anfrage kann man umformen in:

```
SELECT  p2.Name AS N
FROM    Film f, Person p1, Person p2, Spielt s
WHERE   f.Regisseur = p1.PID
AND     p1.Name = 'S. Leone'
AND     f.Titel = s.Film
AND     p2.PID = s.Person

UNION

SELECT  p2.Name AS N
FROM    Film f, Person p1, Person p2, Spielt s
WHERE   f.Regisseur = p1.PID
AND     p1.Name = 'S. Kubrik'
AND     f.Titel = s.Film
AND     p2.PID = s.Person;
```

Aufgabe 3: Mappings - Korrespondenzen (Quelle S1)

Tabellen-Ebene: S1.Buch - G.Buch

Aufgabe 3: Mappings - Korrespondenzen (Quelle S1)

Tabellen-Ebene: S1.Buch - G.Buch

Attribut-Ebene: 1:1

- S1.Buch.Titel - G.Buch.Titel
- S1.Buch.Genre - G.Buch.Genre
- S1.Buch.UVP - G.Preis.NeuPreis
- S1.Autor.Name - G.Buch.Autor
- S1.Verlag.Name - G.Buch.Verlag

Aufgabe 3: Mappings - Korrespondenzen (Quelle S1)

Tabellen-Ebene: S1.Buch - G.Buch

Attribut-Ebene: 1:1

- S1.Buch.Titel - G.Buch.Titel
- S1.Buch.Genre - G.Buch.Genre
- S1.Buch.UVP - G.Preis.NeuPreis
- S1.Autor.Name - G.Buch.Autor
- S1.Verlag.Name - G.Buch.Verlag

Attribut-Ebene: 1:n

- S1.Buch.ISBN - {G.Buch.ISBN10,G.Buch.ISBN13}
- S1.Autorenschaft.ISBN - {G.Buch.ISBN10,G.Buch.ISBN13}

Aufgabe 3: Mappings - Korrespondenzen (Quelle S2)

Tabellen-Ebene: S2.Book - G.Buch, S2.Read - G.Bewertung

Aufgabe 3: Mappings - Korrespondenzen (Quelle S2)

Tabellen-Ebene: S2.Book - G.Buch, S2.Read - G.Bewertung

Attribut-Ebene: 1:1

- S2.Book.Title - G.Buch.Titel
- S2.Book.Author - G.Buch.Autor
- S2.Read.Rating - G.Bewertung.AnzSterne

Aufgabe 3: Mappings - Korrespondenzen (Quelle S2)

Tabellen-Ebene: S2.Book - G.Buch, S2.Read - G.Bewertung

Attribut-Ebene: 1:1

- S2.Book.Title - G.Buch.Titel
- S2.Book.Author - G.Buch.Autor
- S2.Read.Rating - G.Bewertung.AnzSterne

Attribut-Ebene: 1:n und n:1

- S2.Book.ISBN13 - {G.Buch.ISBN10,G.Buch.ISBN13}
- {S2.User.Firstname,S2.User.Name} - G.Bewertung.Nutzer

Aufgabe 3: Mappings - Korrespondenzen (Quelle S2)

Tabellen-Ebene: S2.Book - G.Buch, S2.Read - G.Bewertung

Attribut-Ebene: 1:1

- S2.Book.Title - G.Buch.Titel
- S2.Book.Author - G.Buch.Autor
- S2.Read.Rating - G.Bewertung.AnzSterne

Attribut-Ebene: 1:n und n:1

- S2.Book.ISBN13 - {G.Buch.ISBN10,G.Buch.ISBN13}
- {S2.User.Firstname,S2.User.Name} - G.Bewertung.Nutzer

Anmerkung: Die Tatsache, dass S2.Read.UID - G.Bewertung.AnzSterne keine korrekte Korrespondenz ist, kann man zwar nicht anhand der Labels oder der Instanzdaten erkennen, aber anhand der Struktur (Fremdschlüsselbeziehungen)

Aufgabe 3: Mappings - GaV

```
CREATE VIEW G.Buch AS
SELECT   skolemBuch(b.ISBN) AS BuchID,
          CASE WHEN LENGTH(b.ISBN)<13
            THEN b.ISBN ELSE NULL END AS ISBN10,
          CASE WHEN LENGTH(b.ISBN)>=13
            THEN b.ISBN ELSE NULL END AS ISBN13,
          b.Titel, NULL AS AnzSeiten, a.BuchAutor AS Autor,
          v.Name AS Verlag, b.Genre AS Genre
FROM    S1.Buch b, S1.Verlag v,
          (SELECT   b.ISBN,
              GROUP_CONCAT(a1.Name, ', ') AS BuchAutor
          FROM    S1.Buch b, S1.Autor a1, S1.Autorenschaft a2
          WHERE   a2.Autor = a1.ID AND a2.Buch = b.ISBN
          GROUP BY b.ISBN) AS a
WHERE    b.Verlag = v.ID
AND     b.ISBN = a.ISBN
UNION   ...           (next slide)
```

Aufgabe 3: Mappings - GaV

```
SELECT skolemBuch(b.ISBN13) AS BuchID,  
        NULL AS ISBN10, b.ISBN13 AS ISBN13,  
        b.Title AS Titel, NULL AS AnzSeiten,  
        b.Author AS Autor,  
        NULL AS Verlag, NULL AS Genre  
FROM S2.Book b;
```

Aufgabe 3: Mappings - GaV

```
CREATE VIEW G.Preis AS  
SELECT skolemBuch(b.ISBN) AS BuchID,  
        'Euro' AS Waehrung,  
        b.UVP AS NeuPreis  
FROM    S1.Buch b;
```

```
CREATE VIEW G.Bewertung AS  
SELECT skolemBewertung(b.ISBN13, r.UID) AS BID,  
        skolemBuch(b.ISBN13) AS BuchID,  
        r.Rating AS AnzSterne  
        CONCAT (u.Firstname,' ',u.Name) AS Nutzer,  
        'Bookscrobbler' AS Quelle  
FROM    S2.Book b, S2.User u, S2.Read r  
WHERE   b.BID = r.BID  
AND    u.ID = r.UID;
```

Aufgabe 3: Mappings - GaV

Anmerkung: Die Funktion GROUP_CONCAT ist MySQL spezifisch, jedoch besitzen die meisten großen DBS eine ähnliche Funktion:

Beispiele:

- **Postgres:** STRING_AGG(a1.Name, ',')
- **DB2 und Oracle:** LISTAGG(a1.Name, ',') WITHIN GROUP
- **MS SQL Server:** STUFF
(andere Art der Nutzung als GROUP_CONCAT)

Aufgabe 3: Mappings - GaV

Die vorgestellte Lösung kann unter folgendem Link ausprobiert werden:

<http://sqlfiddle.com/#!9/4cb0f7/1>

Aufgabe 3: Mappings - LaV (Quelle S1)

```
CREATE VIEW S1.Buch AS  
SELECT CASE WHEN b.ISBN13 IS NOT NULL  
        THEN b.ISBN13 ELSE b.ISBN10 END AS ISBN,  
        b.Titel, NULL AS Datum, p.NeuPreis AS UVP, b.Genre,  
        NULL AS Format, NULL AS Verlag  
FROM G.Buch b, G.Preis p  
WHERE b.BuchID = p.BuchID;
```

```
CREATE VIEW S1.Verlag AS  
SELECT NULL AS ID, Verlag AS Name, NULL AS Adresse  
FROM G.Buch;
```

Aufgabe 3: Mappings - LaV (Quelle S1)

```
CREATE VIEW S1.Buch AS  
SELECT  CASE WHEN b.ISBN13 IS NOT NULL  
          THEN b.ISBN13 ELSE b.ISBN10 END AS ISBN,  
          b.Titel, NULL AS Datum, p.NeuPreis AS UVP, b.Genre,  
          NULL AS Format, NULL AS Verlag  
FROM    G.Buch b, G.Preis p  
WHERE   b.BuchID = p.BuchID;
```



```
CREATE VIEW S1.Verlag AS  
SELECT  NULL AS ID, Verlag AS Name, NULL AS Adresse  
FROM    G.Buch;
```

Anmerkung:

Die Tabellen Autor und Autorenschaft können nur mit Hilfe einer komplexen Methode welche eine Liste aus Strings in Attributwerte verschiedener Tuple zerlegt definiert werden. Daher sind diese hier nicht beschrieben.

Aufgabe 3: Mappings - LaV (Quelle S2)

```
CREATE VIEW S2.Book AS  
SELECT NULL AS BID,  
        ISBN13, Titel AS Title,  
        NULL AS Year, Autor AS Author  
FROM   G.Buch;
```

```
CREATE VIEW S2.User AS  
SELECT NULL AS ID,  
        SUBSTRING_INDEX (Autor, ' ', -1) AS Name  
        SUBSTRING_INDEX (Autor, ' ', 1) AS Firstname,  
FROM   G.Buch;
```

```
CREATE VIEW S2.Read AS  
SELECT NULL AS UID, NULL AS BID, AnzSterne AS Rating  
FROM   G.Bewertung;
```

Aufgabe 3: Mappings - LaV

- Problem: In LaV erschweren Aggregationsfunktionen wie CONCAT oder SUBSTRING die Anfrageplanung drastisch

Aufgabe 3: Mappings - LaV

- Problem: In LaV erschweren Aggregationsfunktionen wie CONCAT oder SUBSTRING die Anfrageplanung drastisch
 - Lösung: Auslagerung in Wrapper
- ⇒ Jeder Wrapper stellt ein virtuelles lokales Schema ohne Aggregationsfunktionen bereit mit dem der Mediator arbeitet

Aufgabe 3: Mappings - LaV

- Problem: In LaV erschweren Aggregationsfunktionen wie CONCAT oder SUBSTRING die Anfrageplanung drastisch
 - Lösung: Auslagerung in Wrapper
- ⇒ Jeder Wrapper stellt ein virtuelles lokales Schema ohne Aggregationsfunktionen bereit mit dem der Mediator arbeitet
- Die Aggregation geschieht dann lokal (pro Quelle) innerhalb des jeweiligen Wrappers
 - Das virtuelle Schema des Wrappers ist quasi eine View definiert auf den eigentlichen Quellrelationen
- ⇒ Ein einfaches Query Unfolding ist ausreichend

Aufgabe 3: Mappings - LaV

- Problem: In LaV erschweren Aggregationsfunktionen wie CONCAT oder SUBSTRING die Anfrageplanung drastisch
 - Lösung: Auslagerung in Wrapper
- ⇒ Jeder Wrapper stellt ein virtuelles lokales Schema ohne Aggregationsfunktionen bereit mit dem der Mediator arbeitet
- Die Aggregation geschieht dann lokal (pro Quelle) innerhalb des jeweiligen Wrappers
 - Das virtuelle Schema des Wrappers ist quasi eine View definiert auf den eigentlichen Quellrelationen
 - ⇒ Ein einfaches Query Unfolding ist ausreichend
 - Gleiches gilt für schematische Heterogenität, die aus einer unterschiedlichen Modellierung von Objektbeziehungen resultiert (z.B. Autorenschaft einmal als 1:n und einmal als n:m Beziehung)

Aufgabe 3: Mappings - LaV

Die vorgestellte Lösung kann unter folgendem Link ausprobiert werden:

<http://sqlfiddle.com/#!9/1d46d>

Aufgabe 4a: Ausführbarer Anfrageplan

Aufgabe 4a: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$

Aufgabe 4a: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$

Aufgabe 4a: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$
- Anfrage bindet Variablen G und J
 $\Rightarrow AD_{\text{Film}} = \{\text{'bfff'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$

Aufgabe 4a: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$
- Anfrage bindet Variablen G und J
 $\Rightarrow AD_{\text{Film}} = \{\text{'bfff'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$

- Es gibt kein Subgoal mit einem Adornment 'ff...f'
 \Rightarrow Ende der Schleife

Aufgabe 4a: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$
- Anfrage bindet Variablen G und J
 $\Rightarrow AD_{\text{Film}} = \{\text{'bfff'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$
- Es gibt kein Subgoal mit einem Adornment 'ff...f'
 \Rightarrow Ende der Schleife
- EP ist leer
 \Rightarrow Kein ausführbarer Plan vorhanden

Aufgabe 4b: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$

Aufgabe 4b: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$,
 $AD_{\text{FbG}} = \{\text{'fb'}\}$

Aufgabe 4b: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$,
 $AD_{\text{FbG}} = \{\text{'fb'}\}$
- Anfrage bindet Variablen G und J $\Rightarrow AD_{\text{Film}} = \{\text{'bfff'}\}$ und
 $AD_{\text{FbG}} = \{\text{'ff'}\}$

Aufgabe 4b: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$,
 $AD_{\text{FbG}} = \{\text{'fb'}\}$
- Anfrage bindet Variablen G und J $\Rightarrow AD_{\text{Film}} = \{\text{'bfff'}\}$ und
 $AD_{\text{FbG}} = \{\text{'ff'}\}$

- Schritt 1:
 - Wähle $\text{FilmByGenre}(T, G)$
 - Setze $EP = \{\text{FilmByGenre}\}$
 - Setze $AD_{\text{Film}} = \{\text{'ffff'}\}$ und $AD_{\text{Spielt}} = \{\text{'fff'}\}$

Aufgabe 4b: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$,
 $AD_{\text{FbG}} = \{\text{'fb'}\}$
- Anfrage bindet Variablen G und J $\Rightarrow AD_{\text{Film}} = \{\text{'bfff'}\}$ und
 $AD_{\text{FbG}} = \{\text{'ff'}\}$

- Schritt 2:
 - Wähle $\text{Film}(T, \text{Reg}, G, J)$ (alternativ auch $\text{Spielt}(T, \text{PID}, R)$)
 - Setze $EP = \{\text{FilmByGenre, Film}\}$
 - Keine Änderungen der Adornments

Aufgabe 4b: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$,
 $AD_{\text{FbG}} = \{\text{'fb'}\}$
- Anfrage bindet Variablen G und J $\Rightarrow AD_{\text{Film}} = \{\text{'bfff'}\}$ und
 $AD_{\text{FbG}} = \{\text{'ff'}\}$

- Schritt 3:
 - Wähle $\text{Spielt}(T, \text{PID}, R)$
 - Setze $EP = \{\text{FilmByGenre, Film, Spielt}\}$
 - Setze $AD_{\text{Person}} = \{\text{'ffff'}\}$

Aufgabe 4b: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$,
 $AD_{\text{FbG}} = \{\text{'fb'}\}$
- Anfrage bindet Variablen G und J $\Rightarrow AD_{\text{Film}} = \{\text{'bfff'}\}$ und
 $AD_{\text{FbG}} = \{\text{'ff'}\}$

- Schritt 4:
 - Wähle Person(PID, N, A, O)
 - Setze $EP = \{\text{FilmByGenre, Film, Spielt, Person}\}$
 - Keine Änderungen der Adornments

Aufgabe 4b: Ausführbarer Anfrageplan

- $EP \leftarrow \emptyset$
- $AD_{\text{Film}} = \{\text{'bffb'}\}$, $AD_{\text{Person}} = \{\text{'bfff'}\}$, $AD_{\text{Spielt}} = \{\text{'bff'}\}$,
 $AD_{\text{FbG}} = \{\text{'fb'}\}$
- Anfrage bindet Variablen G und J $\Rightarrow AD_{\text{Film}} = \{\text{'bfff'}\}$ und
 $AD_{\text{FbG}} = \{\text{'ff'}\}$

- Schritt 4:
 - Wähle Person(PID, N, A, O)
 - Setze $EP = \{\text{FilmByGenre, Film, Spielt, Person}\}$
 - Keine Änderungen der Adornments

- EP enthält alle Subgoals
 $\Rightarrow EP$ ist ein ausführbarer Plan für die betrachtete Anfrage