Tutorium 3
Relationale Algebra
Big Data Engineering

Prof. Dr. Jens Dittrich

bigdata.uni-saarland.de

16./17. Mai 2022

Verbesserung Übungsblätter - Häufige Fehler

Aufgabe 1:

- Doppeldeutigkeiten aufgrund von fehlenden Umbenennungen.
- Zu frühe Projektionen, sodass Attribute später fehlen.
- Fehlende Klammerung.
- a) 2.: Fehlende Gruppierung über Klausur ('durchschnittlichen Noten *je* Klausur')
- a) 5.: Fehlende Projektion auf [korrigieren] vor der Gruppierung auf Schüler*in. Dies ist problematisch, da bei einem Join mit unterrichten ein Tupel aus korrigieren mit mehreren Tupeln aus unterrichten konkateniert werden kann, sodass die Ergebnisse verfälscht werden.

Aufgabe 2:

- 4.: Fehlende Gruppierung über max(rank) bei movies und anschließendem Join mit movies über rank = max(rank).

Wiederholung - Frage 1

Frage

Beschreiben Sie die 3 Hauptschritte, die bei einer Gruppierung γ durchgeführt werden. Halten Sie hierbei den Namen des Operators für angemessen?

Wiederholung - Frage 1

Lösung

- **Gruppierung**: Zuerst werden alle Tupel der Relation nach dem Gruppierungsschlüssel $[G]$ horizontal partitioniert. Das heißt alle Tupel, die bzgl. $[G]$ gleich sind, enden in der gleichen Gruppe. Wichtig ist hierbei, dass alle Attribute aus $[G]$ gleich sein müssen und nicht nur einzelne.
- **Aggregation**: Anschließend werden für jede dieser Gruppen die zugehörigen Aggregatfunktionen berechnet. Dabei werden jeweils alle Tupel einer Gruppe betrachtet, **allerdings nicht Tupel aus anderen Gruppen**.
- **Projektion**: Im letzten Schritt wird für jede dieser Gruppen genau ein Tupel erzeugt, dass die Attribute aus $[G]$, sowie die in (2.) berechneten Aggregate enthält.

Da die eigentliche Gruppierung nur einer von 3 Schritten darstellt, ist der Name dieses Operators irreführend.

Wiederholung - Frage 2

Frage

Was ist das Ergebnis des folgenden Ausdrucks?

$\gamma_{\text{rank,max(year)}}$ movies

(A):

max(year)
2000
1998

(B):

rank	max(year)
5.0	2000
3.0	1998

(C):

rank	max(year)
5.0	2000
5.0	2000
3.0	1998

(D):

rank	max(year)
5.0	2000
3.0	2000

movies

id	year	rank
1	2000	5.0
2	1998	3.0
3	1965	5.0

Wiederholung - Frage 2

movies

id	year	rank
1	2000	5.0
2	1998	3.0
3	1965	5.0

Frage

Was ist das Ergebnis des folgenden Ausdrucks?

$\gamma_{\text{rank,max(year)}}$ movies

Lösung

Die richtige Antwort lautet (B):

rank	max(year)
5.0	2000
3.0	1998

Wiederholung - Frage 3

Frage

Was ist das Ergebnis des folgenden Ausdrucks?

$\gamma_{\text{name, year}}$ movies

(A):

name	year
The Avengers	2012
Gladiator	2000, 1992

(B):

name	year
The Avengers	2012
Gladiator	2000
Gladiator	1992

movies

id	name	year	rank
1	The Avengers	2012	8.1
2	Gladiator	2000	8.2
3	Gladiator	1992	6.5

Wiederholung - Frage 3

movies

id	name	year	rank
1	The Avengers	2012	8.1
2	Gladiator	2000	8.2
3	Gladiator	1992	6.5

Frage

Was ist das Ergebnis des folgenden Ausdrucks?

$\gamma_{\text{name, year}}$ movies

Lösung

Die richtige Antwort lautet (B):

name	year
The Avengers	2012
Gladiator	2000
Gladiator	1992

Wiederholung - Frage 4

Frage

Betrachten Sie den folgenden relationalen Ausdruck, der basierend auf dem unten stehenden Ausschnitt eines Relationenschemas das durchschnittliche Geburtsjahr aller Personen berechnen soll, die schon mal in einem Auto gesessen haben. Fällt Ihnen ein Problem mit diesem Ausdruck auf? Wie würden sie es beheben?

$$\gamma_{\text{avg(Geburtsjahr)}}(\text{Personen} \bowtie_{\text{ID} = \text{Person\_ID}} \text{sitzen\_in})$$

$$[\text{sitzen\_in}] : \{[\underline{\text{Person\_ID:(Personen}\rightarrow\text{ID}), \text{ Zeitpunkt: time}}, \text{ Auto\_ID:(Autos}\rightarrow\text{ID})]\}$$

$$[\text{Personen}] : \{[\underline{\text{ID: int}}, \text{ Geburtsjahr: int}]\}$$

Wiederholung - Frage 4

$$\gamma_{\text{avg(Geburtsjahr)}}(\text{Personen} \bowtie_{\text{ID} = \text{Person\_ID}} \text{sitzen\_in})$$

$$[\text{sitzen\_in}] : \{[\underline{\text{Person\_ID:(Personen}\rightarrow\text{ID}), \text{ Zeitpunkt: time}}, \text{Auto\_ID:(Autos}\rightarrow\text{ID})]\}$$

$$[\text{Personen}] : \{[\underline{\text{ID: int}}, \text{ Geburtsjahr: int}]\}$$

Lösung

Das Problem ist, dass Person_ID nicht das einzige Schlüsselattribut in [sitzen_in] ist. Daher werden die Tupel von Personen, die schon zu unterschiedlichen Zeitpunkten in einem Auto gesessen haben, entsprechend oft beim Join mit Tupeln aus [sitzen_in] konkateniert. Dies hat zur Folge, dass die gleiche Person bei der Gruppierung mehrmals betrachtet wird, wodurch die Ergebnisse verfälscht werden. Um dies zu lösen, muss man vor der Gruppierung eine Projektion auf [Personen] einführen, da so aufgrund der Mengensemantik für jede Person ein einziges Tupel entstehen wird:

$$\gamma_{\text{avg(Geburtsjahr)}}(\pi_{[\text{Personen}]}(\text{Personen} \bowtie_{\text{ID} = \text{Person\_ID}} \text{sitzen\_in}))$$

Aufgabe 1

Frage

In den folgenden Aufgaben werden Sie die Gruppierung '$\gamma_{\text{name,avg(rank),max(year)}}$ movies' am Beispiel der unten stehenden Relation schrittweise von Hand durchführen

movies

id	name	year	rank
1	The Avengers	2012	8.1
2	Gladiator	2000	8.2
3	Gladiator	1992	6.5
4	Frozen	2010	6.1
5	Independance Day	1993	6.5
6	Frozen	2013	7.4
7	Independance Day	1996	7.0
8	Terminator	1984	8.1

Aufgabe 1.1

Frage

Bilden Sie zunächst die Horizontale Partitionierungen für den Gruppierungsschlüssel [name] der folgenden Relation.

movies

id	name	year	rank
1	The Avengers	2012	8.1
2	Gladiator	2000	8.2
3	Gladiator	1992	6.5
4	Frozen	2010	6.1
5	Independance Day	1993	6.5
6	Frozen	2013	7.4
7	Independance Day	1996	7.0
8	Terminator	1984	8.1

Aufgabe 1.1

Frage

Bilden Sie die Horizontale Partitionierungen für den Gruppierungsschlüssel [name] der folgenden Relation.

Lösung

- Terminator: {(8, Terminator, 1984, 8.1)}
- Gladiator: {(2, Gladiator, 2000, 8.2), (3, Gladiator, 1992, 6.5)}
- Independance Day: {(5, Independance Day, 1993, 6.5), (7, Independance Day, 1996, 7.0) }
- Frozen: {(4, Frozen, 2010, 6.1), (6, Frozen, 2013, 7.4)}
- The Avengers: {(1, The Avengers, 2012, 8.1)}

Aufgabe 1.2

Frage

Berechnen Sie nun für jede der gebildeten Partitionen die Aggregatsfunktionen avg(rank) und max(year).

movies

id	name	year	rank
1	The Avengers	2012	8.1
2	Gladiator	2000	8.2
3	Gladiator	1992	6.5
4	Frozen	2010	6.1
5	Independance Day	1993	6.5
6	Frozen	2013	7.4
7	Independance Day	1996	7.0
8	Terminator	1984	8.1

Aufgabe 1.2

Frage

Berechnen Sie nun für jede der gebildeten Partitionen die Aggregatsfunktionen avg(rank) und max(year).

Lösung

- Terminator: avg(rank) = 8.1 und max(year) = 1984
- Gladiator: avg(rank) = 7.35 und max(year) = 2000
- Independance Day: avg(rank) = 6.75 und max(year) = 1996
- Frozen: avg(rank) = 6.75 und max(year) = 2013
- The Avengers: avg(rank) = 8.1 und max(year) = 2012

Aufgabe 1.3

Frage

Vollenden Sie nun die Gruppierung, indem sie die finale Ausgaberelation der Gruppierung angeben.

movies

id	name	year	rank
1	The Avengers	2012	8.1
2	Gladiator	2000	8.2
3	Gladiator	1992	6.5
4	Frozen	2010	6.1
5	Independance Day	1993	6.5
6	Frozen	2013	7.4
7	Independance Day	1996	7.0
8	Terminator	1984	8.1

Aufgabe 1.3

Frage

Vollenden Sie nun die Gruppierung, indem sie die finale Ausgaberelation der Gruppierung angeben.

Lösung

$\gamma_{\text{name,avg(rank),max(year)}}$ movies :=
{(Terminator, 8.1, 1984), (Gladiator, 7.35, 2000), (Independance Day, 6.75, 1996), (Frozen, 6.75, 2013), (The Avengers, 8.1, 2012)}

Aufgabe 2

Hinweis

Benutzen Sie folgendes Relationenschema für Aufgabe 2.

[Personen] : {[PID:int, Name:varchar, Wohnort: varchar, Geburtsjahr: int]}

[Sänger*innen] : {[SID:(Personen→PID), Künstler*innenname:varchar, Genre:varchar]}

[Musiklabels] : {[MID:int, Name:varchar, Kapital:varchar, Gründungsjahr: int]}

[Songs] : {[SongID:int, Label:(Musiklabels→MID), Titel: varchar,
Veröffentlichtungsdatum: date]}

[singen_live] : {[Song:(Songs→SongID), Sänger*in:(Sänger*innen→SID), Datum:date,
Arena:string]}

Aufgabe 2.1

Frage

Übersetzen Sie folgende umgangssprachliche Anfrage in relationale Algebra:

(a) Das Kapital der Musiklabels, die schon mindestens 10 Lieder veröffentlicht haben.

(b) Die Arena, in denen schon mal live ein Song gesungen wurde, der von den ältesten Musiklabels veröffentlicht wurden.

Aufgabe 2.1

Lösung

(a) $R := \sigma_{\text{count}(*) \geq 10} \left(\gamma_{\text{Label,count}(*)} \text{ Songs}\right)$
$\pi_{\text{Kapital}} \left(\text{Musiklabels} \bowtie_{\text{MID} = \text{Label}} R\right)$

(b) $R_1 := \gamma_{\text{min(Gründungsjahr)}} \text{ Musiklabels}$
$R_2 := \text{Musiklabels} \bowtie_{\text{Gründungsjahr} = \text{min(Grüdungsjahr)}} R_1$
$\pi_{\text{Arena}} \left(\text{singen\_live} \bowtie_{\text{Song} = \text{SongID}} \left(R_2 \bowtie_{\text{MID} = \text{Label}} \text{Songs}\right)\right)$

Aufgabe 2.2

Frage

Übersetzen Sie folgende Ausdrücke der relationalen Algebra in natürliche Sprache:

(a) $R := \sigma_{\text{count(*)} = 3}\ (\gamma_{\text{Song, Datum,count(*)}}\ (\sigma_{\text{Arena} = \text{'BigEvents'}}\ \text{singen\_live}))$
$\pi_{\text{Veröffentlichungsdatum}}\ (\text{Songs} \bowtie_{\text{SongID} = \text{Song}}\ R)$

(b) $R_1 := \gamma_{\text{Sänger*in,min(Datum), max(Datum)}}\ \text{singen\_live}$
$R_2 := \sigma_{\text{min(Datum)} > 17.10.2006 \wedge \text{max(Datum)} < 20.12.2019}\ R_1$
$R_3 := \pi_{\text{Genre}}\ (R_2 \bowtie_{\text{Sänger*in} = \text{SID}}\ \text{Sänger*innen})$

Aufgabe 2.2

Lösung

(a) Das Veröffentlichungsdatum der Songs, die schon von genau 3 Sänger*innen am gleichen Tag in der Arena mit dem Namen 'BigEvent' gesungen wurden.

(b) Das Genre der Sänger*innen, deren erster Auftritt nach dem 17.10.2006 und letzter Auftritt vor dem 20.12.2019 war.

Aufgabe 3

Frage

In den folgenden Aufgaben werden sie eine Co-Group-Join, d.h. einen Equi-Join unter Zuhilfenahme einer Co-Gruppierung, anhand der unten stehenden Relationen schrittweise von Hand durchführen.

directors

id	first_name	last_name
1	James	Cameron
2	Sam	Raimi
3	James	Wan
4	Sam	Mendes
5	George	Lucas

actors

a_id	f_name	l_name
1	George	Clooney
2	Sam	Rockwell
3	James	McAvoy
4	Anthony	Mackey
5	Anthony	Hopkins

Aufgabe 3.1

Frage

Sei $p_{directors}(d) := d.first\_name$ und $p_{actors}(a) := a.f\_name$. Bestimmen Sie zunächst die Horizontalen Partitionierung der einzelnen Relationen gemäß den angegeben Partitionierungsfunktionen.

directors

id	first_name	last_name
1	James	Cameron
2	Sam	Raimi
3	James	Wan
4	Sam	Mendes
5	George	Lucas

actors

a_id	f_name	l_name
1	George	Clooney
2	Sam	Rockwell
3	James	McAvoy
4	Anthony	Mackey
5	Anthony	Hopkins

Aufgabe 3.1

Frage

Sei $p_{directors}(d) := d.first\_name$ und $p_{actors}(a) := a.f\_name$. Bestimmen Sie zunächst die Horizontalen Partitionierung der einzelnen Relationen gemäß den angegeben Partitionierungsfunktionen.

Lösung

- **directors**:
  - James: {(1, James, Cameron), (3, James, Wan)}
  - Sam: {(2, Sam, Raimi), (4, Sam, Mendes)}
  - George: {(5, George, Lucas)}
- **actors**:
  - James: {(3, James, McAvoy)}
  - Anthony: {(4, Anthony, Mackey), (5, Anthony, Hopkins)}
  - Sam: {(2, Sam, Rockwell)}
  - George: {(1, George, Clooney)}

Aufgabe 3.2

Frage

Bilden Sie nun die Co-Gruppierung, in dem gemäß Sie des Schemas der Co-Gruppierung beide Partitionierungen zusammenfügen. Hierfür müssen Sie die Partitionen beider Relationen nehmen und diejenigen kombinieren, deren Gruppierungsschlüssel gleich sind. Anschließend entsteht für jeden möglichen Schlüssel d mit Schema $[D]$ ein neues Tupel mit dem Schema $[d : D, directors : [directors], actors : [actors]]$. Taucht ein Gruppierungschlüssel lediglich in einer der beiden Relationen auf, so ist das entsprechende Tupelelement der anderen Relation die leere Menge.

directors

id	first_name	last_name
1	James	Cameron
2	Sam	Raimi
3	James	Wan
4	Sam	Mendes
5	George	Lucas

actors

a_id	f_name	l_name
1	George	Clooney
2	Sam	Rockwell
3	James	McAvoy
4	Anthony	Mackey
5	Anthony	Hopkins

Aufgabe 3.2

Lösung

$\Gamma_{p_{directors}, p_{actors}}(directors, actors) :=$
{
(James, {(3, James, Wan), (1, James, Cameron)}, {(4, James, MacAvoy)}),
(Sam, {(2, Sam, Raimi), (4, Sam, Mendes)}, {(2, Sam, Rockwell)}),
(George, {(5, George, Lucas)}, {(1, George, Clooney)}),
(Anthony, {}, {(4, Anthony, Mackey), (5, Anthony, Hopkins)})
}

Aufgabe 3.3

Frage

Erstellen Sie nun das Ergebnis des Equi-Joins, indem für jedes Tupel der Co-Gruppierung das Kreuzprodukt der letzten beiden Tupelelemente bilden.

directors

id	first_name	last_name
1	James	Cameron
2	Sam	Raimi
3	James	Wan
4	Sam	Mendes
5	George	Lucas

actors

a_id	f_name	l_name
1	George	Clooney
2	Sam	Rockwell
3	James	McAvoy
4	Anthony	Mackey
5	Anthony	Hopkins

Aufgabe 3.3

Frage

Erstellen Sie nun das Ergebnis des Equi-Joins, indem für jedes Tupel der Co-Gruppierung das Kreuzprodukt der letzten beiden Tupelelemente bilden.

Lösung

directors $\bowtie_{\text{first\_name} = \text{f\_name}}$ actors :=
{
(1, James, Cameron, 3, James, McAvoy), (3, James, Wan, 3, James, McAvoy),
(2, Sam, Raimi, 2, Sam, Rockwell), (4, Sam, Mendes, 2, Sam, Rockwell),
(5, George, Lucas, 1, George, Clooney)
}

Aufgabe 4

Frage

Betrachten Sie folgendes Relationenschema, welches einen einfachen gerichteten Graphen repräsentieren soll.

$$[\text{vertices}] : \{[\underline{\text{ID:int}}]\}$$
$$[\text{edges}] : \{[\underline{\text{parent:(vertices}\rightarrow\text{ID), child:(vertices}\rightarrow\text{ID)}}]\}$$

(a) Geben Sie nun einen Ausdruck in relationaler Algebra an, der alle Pfade der Länge 2 ausgibt.

(b) Geben Sie einen Ausdruck in relationaler Algebra an, der alle Pfade der Länge 3 ausgibt.

(c) Können Sie einen Ausdruck definieren, welcher für beliebige, aber feste $n \in \mathbb{N}$, alle Pfade der Länge n ausgibt?

Eine mögliche Ausgabe sollte die IDs aller Knoten des Pfades enthalten.

Aufgabe 4

Lösung

(a) $R_1 := \rho_{\text{start\_p} \leftarrow \text{parent}, \text{start\_c} \leftarrow \text{child}}\ \text{edges}$
$R_2 := \rho_{\text{end\_p} \leftarrow \text{parent}, \text{end\_c} \leftarrow \text{child}}\ \text{edges}$
$R_3 := R_1 \bowtie_{\text{start\_c}=\text{end\_p}} R_2$
$\pi_{\text{start\_p, start\_c, end\_c}}\ R_3$

(b) $R_1 := \rho_{\text{start\_p} \leftarrow \text{parent}, \text{start\_c} \leftarrow \text{child}}\ \text{edges}$
$R_2 := \rho_{\text{middle\_p} \leftarrow \text{parent}, \text{middle\_c} \leftarrow \text{child}}\ \text{edges}$
$R_3 := \rho_{\text{end\_p} \leftarrow \text{parent}, \text{end\_c} \leftarrow \text{child}}\ \text{edges}$
$R_4 := (R_1 \bowtie_{\text{start\_c}=\text{middle\_p}} R_2) \bowtie_{\text{middle\_c}=\text{end\_p}} R_3$
$\pi_{\text{start\_p, start\_c, middle\_c, end\_c}}\ R_4$

(c) Nein, in relationaler Algebra ist dies nicht möglich, da es keine Rekursion gibt.