

14.11.2023

Übungsblatt 5 – Lösungsvorschlag

Diskussionsteil (im PS zu lösen; keine Abgabe nötig)

- a) ☐ ★ Was ist der Unterschied zwischen einem NATURAL JOIN und einem EQUIJOIN?

Lösung



Der EQUIJOIN enthält nur Join-Bedingungen mit Gleichheitszeichen. Ein NATURAL JOIN ist ein EQUIJOIN der die Join-Bedingung aus den gegebenen Relationen automatisch aufstellt (gleiche Spaltennamen = gleiche Werte). Beim NATURAL JOIN ist jedes Join-Attribut in der resultierenden Relation nur einmal enthalten.

- b) ☐ ★ Gegeben sei ein Relationenmodell mit Relationenschemata $R(\underline{id}, name, length)$ und $S(\underline{id}, from_date, to_date, length)$ mit folgenden Tupeln:

R		
id	name	length
0	Nationalfeiertag	16
1	Allerheiligen	13
2	Mariä Empfängnis	18
3	Heiliger Abend	14
4	Weihnachten	11
5	Stefanitag	10
6	Silvester	9

S			
id	from_date	to_date	length
1	2019-11-01	2019-11-03	3
4	2019-12-23	2020-01-06	15

Bestimmen und vergleichen Sie die Ergebnisse der Relationenalgebra-Ausdrücke

$$R \bowtie S \text{ und } R \bowtie_{R.id=S.id} S.$$

Lösung



Hinweis



Unter

dbis-uibk.github.io/relax/calc/gist/e0b27dcd0fa7e9657529be136fd09aff

können Sie die Datenbank in RelaX öffnen.

a) Das Ergebnis des NATURAL JOINS $R \bowtie S$ ist leer ($R \text{ join } S$)

b) Das Ergebnis des EQUIJOINS ($R \text{ join } R.id=S.id S$) ist:

R.id	name	R.length	S.id	from_date	to_date	S.length
1	Allerheiligen	13	1	2019-11-01	2019-11-03	3
4	Weihnachten	11	4	2019-12-23	2020-01-06	15

- c) ☐ ★ Gegeben sei ein Relationenmodell mit Relationenschemata Drivers(driverID, firstname, lastname, team, points) mit folgenden Tupeln:

driverID	firstname	lastname	team	points
44	Lewis	Hamilton	Mercedes	338
33	Max	Verstappen	Red Bull Racing	212
77	Valtteri	Bottas	Mercedes	274
5	Sebastian	Vettel	Ferrari	212
27	Nico	Hülkenberg	Renault	35
16	Charles	Leclerc	Ferrari	221
3	Daniel	Ricciardo	Renault	42
23	Alexander	Albon	Red Bull Racing	64

- a) Bestimmen Sie $\gamma_{\text{team}; \text{MAX}(\text{points}) \rightarrow \text{top}, \text{COUNT}(\text{driverID}) \rightarrow \text{numDrivers}}(\text{Drivers})$.
b) Was wird mit $\gamma; \text{AVG}(\text{points}) \rightarrow \text{avg}(\text{Drivers})$ bestimmt?

Hinweis



Unter

dbis-uibk.github.io/relax/calc/gist/bb4cec3e91daa1d18539ae56e9855774

können Sie die Datenbank „Drivers“ in RelaX öffnen.

Lösung



- a) Resultat Aggregation der Tabelle Drivers:

Drivers.team	top	numDrivers
'Mercedes'	338	2
'RedBullRacing'	212	2
'Ferrari'	221	2
'Renault'	42	2

RelaX-Abfrage:

`gamma team; MAX(points)->top, COUNT(driverID)->numDrivers(Drivers)`

- b) Es wird über die gesamte Tabelle der Durchschnitt des Attributs points berechnet:

avg
174.75

RelaX-Abfrage:

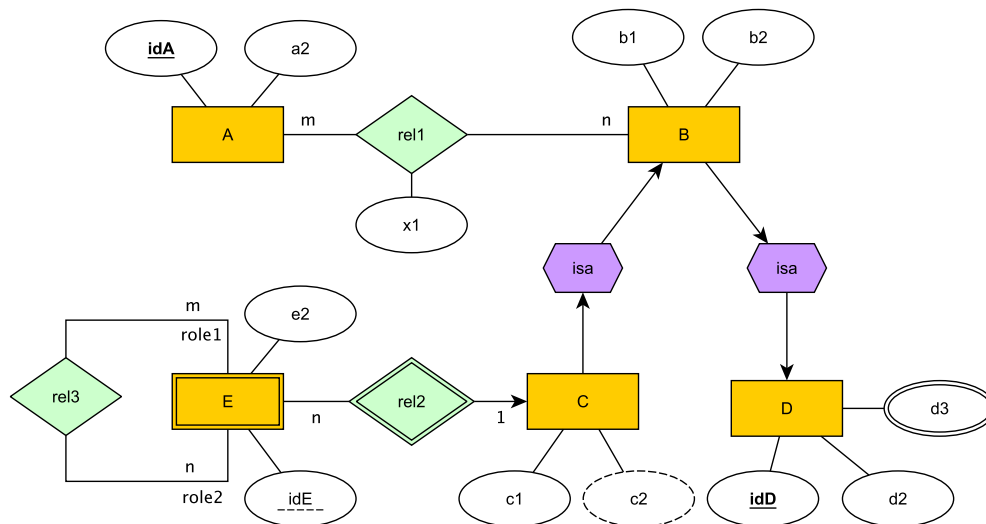
`gamma ; AVG(points)->avg(Drivers)`

Hausaufgabenteil (Zuhause zu lösen; Abgabe nötig)

Aufgabe 1 (Relationales Mapping)

[3 Punkte]

Erstellen Sie zu dem gegebenen ER-Diagramm ein passendes Relationalschema.



Abgabe



mapping.pdf

Lösung



```
A(idA, a2)
D(idD, d2)
D_d3(idD, d3)
B(idD, b1, b2)
C(idD, c1, c2)
E(idD, idE, e2)
rel1(idA, idD, x1)
rel3(idD-role1, idE-role1, idD-role2, idE-role2)
```

Aufgabe 2 (Relationale Algebra (Theorie))

[3 Punkte]

Gegeben sei das folgende relationale Modell mit den Relationen R und S:

- a) 0.5 Punkte Berechnen Sie manuell $R \bowtie S$.

Abgabe



2a_result.txt

R(A,B):	A	B	S(B,C,D):	B	C	D
	1	a		c	5	6
	7	t		a	7	8
	2	g		t	8	9
	4	c				
	9	t				

Lösung



A	B	C	D
1	a	7	8
7	t	8	9
4	c	5	6
9	t	8	9

- b) 0.5 Punkte Berechnen Sie manuell $R \bowtie_{R.A=S.C} S$.

Abgabe



2b_result.txt

Lösung



A	R.B	S.B	C	D
7	t	a	7	8

- c) 0.5 Punkte Berechnen Sie manuell $(\sigma_{B='t'} R) \bowtie_{R.A \geq S.C} S$.

Abgabe



2c_result.txt

Lösung



A	R.B	S.B	C	D
7	t	c	5	6
7	t	a	7	8
9	t	c	5	6
9	t	a	7	8
9	t	t	8	9

- d) 1.5 Punkte Berechnen Sie manuell den folgenden Relationalen Algebra Ausdruck.

$$(\rho_X(S \bowtie_{C=G} (\rho_{E \leftarrow B, F \leftarrow C, G \leftarrow D} S))) \bowtie_{X.C=Y.C \wedge X.G=Y.D} (\rho_Y(S))$$

Abgabe



2d_result.txt

Lösung



Der linke Teil ($\rho_X (S \bowtie_{C=G} (\rho_{E \leftarrow B, F \leftarrow C, G \leftarrow D} S)))$)

X	B	C	D	E	F	G
t	t	8	9	a	7	8

wird mit der Join-Bedingung $X.C = Y.C \wedge X.G = Y.D$ mit Relation S gejoint – Joinattribute, die die Join-Bedingung erfüllen, sind fett markiert –, die davor in Y umbenannt wird:

X.B	X.C	X.D	(X.)E	(X.)F	(X.)G	Y.B	Y.C	Y.D
t	8	9	a	7	8	t	8	9
t	8	9	a	7	8	a	7	8

Dementsprechend ist das Endergebnis eine leere Relation.

Aufgabe 3 (Relationale Algebra Abfragen)

[4 Punkte]

Gegeben sei das folgende Relationenschema eines Informationssystems, das Daten über Schachturniere verwaltet:

Club (id, name, country, address, zip)

Ranking (id, title)

Player (id, firstname, lastname, birthday, email, rankId, clubId)

Tournament (id, name, country, start_date)

Participation (playerId, tournamentId, finalRank)

Match (tournamentId, player1Id, player2Id, duration, result)

Erstellen Sie auf Basis dieses Relationenschemas die folgenden Anfragen in relationaler Algebra. Sie können dazu RelaX verwenden. Überlegen Sie sich jedoch trotzdem, wie man die Operationen „händisch“ berechnen würde. Mit folgendem Link ist RelaX inklusive des für diese Aufgabe benötigten Schemas erreichbar:

<https://dbis-uibk.github.io/relax/calc/gist/6f0a6c87c4ab5ba9a3eccabbe1837d55>

- a) 0.5 Punkte Geben Sie Vor- und Nachnamen aller Spieler*innen aus, die vor dem Jahr 1945 geboren wurden.

Abgabe



3a_query.txt

3a_result.txt

Lösung



Query:

`pi firstname, lastname (sigma birthday < '1945-01-01' Player)`

Resultat:

Player.firstname Player.lastname

'Alea'	'Cline'
'Savannah'	'Mcintosh'
'Maris'	'Wooten'
'Malcolm'	'Schultz'

4 Tupel gesamt

- b) 0.5 Punkte Geben Sie die E-Mail-Adressen aller Spieler*innen aus, die den Rang Grand Master oder Professional haben.

Abgabe



3b_query.txt

3b_result.txt

Lösung



Query:

```
pi email (Player join (rankId = Ranking.id)
(sigma title='Grand Master' or title='Professional' Ranking))
```

Resultat:

Player.email

```
'quis.massa.Mauris@aliquameros.co.uk'
'nisl.Quisque@mattisvelit.edu'
'magna.Sed@vestibulum.ca'
'egestas.Fusce.aliquet@enimcommodohendrerit.org'
'eu@Integeridmagna.net'
'neque.Sed@Phasellus.net'
'Suspendisse.tristique.neque@tortorInteger.org'
'torquent.per@enimnectempus.org'
'bibendum@Maecenasornare.co.uk'
'luctus@orci.org'
```

68 Tupel gesamt

- c) 0.5 Punkte Finden Sie die Ergebnisse aller Turniere, die in Dänemark ('Denmark') stattgefunden haben und geben Sie jeweils die id der Spieler*innen und das Ergebnis aus.

Abgabe



3c_query.txt

3c_result.txt

Lösung

Query:

```
pi player1Id, player2Id, result (Match
  join (Match.tournamentId = Tournament.id) (sigma country='Denmark'
    Tournament))
```

Resultat:

Match.player1Id	Match.player2Id	Match.result
68	43	'0-1'
5	29	'1-0'
81	48	'1-0'
19	94	'0-1'

4 Tupel gesamt

- d) 0.5 Punkte Finden Sie das Turnier, bei dem ein Spiel nur 3 Minuten gedauert hat.
Hinweis: die Dauer von Spielen ist (leider) ein String, d.h. es muss ein exakter Stringvergleich durchgeführt werden. Das Zeitformat ist HH:MM.

Abgabe

3d_query.txt

3d_result.txt

Lösung

Query:

```
pi Tournament.name ((sigma duration='00:03' Match)
  join (tournamentId = Tournament.id) Tournament)
```

Resultat:

Tournament.name
'Tournament #79'

1 Tupel gesamt

- e) 1 Punkt Geben Sie den Namen aller Turniere aus, an denen die Spielerin 'Aimee Lamb' teilgenommen hat.

Abgabe

3e_query.txt

3e_result.txt

Lösung



Query:

```
pi Tournament.name ((sigma firstname='Aimee' and lastname='Lamb' Player)
  join (Player.id=playerId) Participation
  join (Tournament.id = tournamentId) Tournament)
```

Resultat:


```
Tournament.name
'Tournament #89'
'Tournament #4'
```


2 Tupel gesamt

- f) 1 Punkt Geben Sie Vor- und Nachnamen aller Spieler*innen aus, die bei einem Turnier, das nach dem Juli 2016 begonnen hat, den 1. Platz erreicht haben.

Abgabe



 3f_query.txt

 3f_result.txt

Lösung



Query:

```
pi firstname, lastname (Player join (Player.id = Participation.playerId)
  (sigma finalRank=1 Participation)
  join (Participation.tournamentId = Tournament.id)
  (sigma start_date >= '2016-08-01' Tournament))
```

Resultat:

Player.firstname	Player.lastname
'Dale'	'Salas'
'Rose'	'Lopez'

2 Tupel gesamt

Wichtig: Laden Sie bitte Ihre Lösung in OLAT hoch und geben Sie mittels der Ankreuzliste auch unbedingt an, welche Aufgaben Sie gelöst haben. Die Deadline dafür läuft am Vortag des Proseminars um 16:00 ab.