Proseminar Datenbanksysteme

Universität Innsbruck - Institut für Informatik

Y. Ipek, M. Mayerl, M. Vötter, E. Zangerle

31.10.2019

Übungsblatt 4 – Lösungsvorschlag

Diskussionsteil (im PS zu lösen; keine Abgabe nötig)

a) ★★ Gegeben seien folgende Relationen, die einen Blogpost (`Post`) und dazugehörige Tags (`Tag`) beinhalten, wobei Tags dazu verwendet werden, um Blogposts verschiedenen Themen zuzuordnen:

`Post(PostID, Headline, Author)`

PostID	Headline	Author
1	My favourite recipes	Mary Potter
2	Setting up a Linux VM	Jane Doe
3	First Steps in Python	Bob Smith
4	Travels 2019	John Doe
5	Introduction to SEO	Anne Johnson
6	Knitting a scarf	William Gold
7	Getting ready for my first marathon	Alicia Silverstone

`Tag(PostID, Tag)`

PostID	Tag
1	cooking
1	recipe
1	diy
3	coding
3	hacking
4	usa
4	roadtrip
4	travels
6	wool
6	diy

Lösung ✔

RelaX zu dieser Datenbank: http://dbis-uibk.github.io/relax/calc.htm?data=gist:65141c587bdf3dd124e5ae27a93c85b8

Berechnen Sie das Ergebnis folgender Abfragen:

a) $\sigma_{\text{Headline}=\text{“Travels 2019”}}\text{Post}$

Lösung ✔

ID	Headline	Author
4	Travels 2019	John Doe

RelaX-Abfrage:

```
sigma Headline='Travels 2019'(Post)
```

b) $Post \bowtie_{Post.ID=Tag.PostID} Tag$

Lösung ✔

Post.PostID	Post.Headline	Post.Author	Tag.PostID	Tag.Tag
1	My favourite recipes	Mary Potter	1	cooking
1	My favourite recipes	Mary Potter	1	recipe
1	My favourite recipes	Mary Potter	1	diy
3	First Steps in Python	Bob Smith	3	coding
3	First Steps in Python	Bob Smith	3	hacking
4	Travels 2019	John Doe	4	usa
4	Travels 2019	John Doe	4	roadtrip
4	Travels 2019	John Doe	4	travels
6	Knitting a scarf	William Gold	6	wool
6	Knitting a scarf	William Gold	6	diy

RelaX-Abfrage:

```
Post join Post.PostID=Tag.PostID Tag
```

c) $\pi_{Headline,Tag}((\sigma_{Author=\text{“MaryPotter”}} Post) \times Tag)$

Lösung ✔

Post.Headline	Tag.Tag
My favourite recipes	cooking
My favourite recipes	recipe
My favourite recipes	diy
My favourite recipes	coding
My favourite recipes	hacking
My favourite recipes	usa
My favourite recipes	roadtrip
My favourite recipes	travels
My favourite recipes	wool

RelaX-Abfrage:

```
pi Headline, Tag ((sigma Author='Mary Potter'(Post)) x Tag)
```

[a] Hier und bei weiteren Beispielen, die Strings beinhalten kann es beim Herauskopieren der Lösung dazu kommen, dass das Leerzeichen nach dem String nicht mitkopiert wird und RelaX deshalb einen Syntaxfehler erkennt. Sie können dies lösen, indem Sie einfach das Leerzeichen nach dem String-Ende (vor (Post)) einfügen.

d) $\sigma_{Post.PostID<4}(Post \bowtie_{Post.PostID=Tag.PostID} Tag)$

Lösung ✔

Post.PostID	Headline	Author	Tag.PostID	Tag
1	My favourite recipes	Mary Potter	1	cooking
1	My favourite recipes	Mary Potter	1	recipe
1	My favourite recipes	Mary Potter	1	diy
2	Setting up a Linux VM	Jane Doe	null	null
3	First Steps in Python	Bob Smith	3	coding
3	First Steps in Python	Bob Smith	3	hacking

RelaX-Abfrage:

```
sigma Post.PostID < 4 (Post left outer join Post.PostID=Tag.PostID Tag)
```

e) $(\sigma_{Tag=\text{“diy”}} Tag) \rtimes Post$

Lösung ✔

PostID	Headline	Author
1	My favourite recipes	Mary Potter
6	Knitting a scarf	William Gold

RelaX-Abfrage:

```
(sigma Tag='diy'(Tag)) right semi join Post
```

b) ★ Für die Ausführung der Operationen $\cup$, $-$, $\cap$, $\div$ müssen die Schemata der beiden Relationen ident sein - wieso gilt dies nicht für z.B. Joins?

Lösung ✔

Die Schemata müssen ident sein, da die Operationen $\cup$, $-$, $\cap$, $\div$ auf Zeilenbasis berechnet werden (es werden einzelne Zeilen vergleichen; dies ist nur bei gleichem Schema möglich). Bei Joins hingegen wird der Join nur auf zwei Spalten berechnet, das Schema ist unerheblich für diese Operation.

c) ★ Falls die in der vorigen Aufgabe festgestellte Bedingung für die gegebenen Relationen nicht gilt: wie können Sie die Ausführung dieser Operatoren trotzdem ermöglichen?

Lösung ✔

Dies kann durch Projektionen erreicht werden, die so auf die Relationen angewendet werden, dass diese danach das gleiche Schema aufweisen.

Hausaufgabenteil (Zuhause zu lösen; Abgabe nötig)

Aufgabe 1 (Relationale Algebra: Basics) [4 Punkte]

Gegeben sei das folgende relationale Modell:

R(A,B,C)

A	B	C
0	8	c
1	19	b
2	14	z
3	15	e
4	15	t
5	7	f
6	9	x

S(X, A, Z)

X	A	Z
b	0	10
x	2	11
x	1	12
z	3	14
c	5	12

Lösung ✔

RelaX zu dieser Datenbank: http://dbis-uibk.github.io/relax/calc.htm?data=gist:98e1f0d0710f119c3fcb1113371ae885

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

Lösung ✔

A	B	C	X	Z
0	8	c	b	10
1	19	b	x	11
2	14	z	x	12
3	15	e	z	14
5	7	f	c	12

RelaX-Abfrage:

```
R join S
```

b) *0.5 Punkte* Berechnen Sie $R \bowtie_{R.B=S.Z} S$.

Lösung ✔

R.A	B	C	X	S.A	Z
2	14	z	z	3	14

RelaX-Abfrage:

```
R join R.B=S.Z S
```

c) *0.5 Punkte* Berechnen Sie $(\sigma_{A>0} S) \bowtie (\sigma_{B<10} R)$.

Lösung ✔

X	A	Z	B	C
c	5	12	7	f

RelaX-Abfrage:
```
(sigma A > 0 (S)) join (sigma B < 10 (R))
```

d) *0.5 Punkte* Berechnen Sie $(\sigma_{A<2}S) \bowtie (\pi_{B,C}(\sigma_{B<10}R))$.

Lösung ✔

X	A	Z	B	C
b	0	10	8	c
b	0	10	7	f
b	0	10	9	x
x	1	12	8	c
x	1	12	7	f
x	1	12	9	x

RelaX-Abfrage:
```
(sigma A < 2 S) join (pi B,C (sigma B < 10 R))
```

e) *1 Punkt* Berechnen Sie $\sigma_{A<2}((\pi_{A,B}R) \cup (\pi_{A,B\leftarrow Z}S))$.

Lösung ✔

A	B
0	8
1	19
0	10
1	12

RelaX-Abfrage:
```
sigma A < 2 ((pi A,B (R)) union (pi A,B <- Z (S)))
```

f) *1 Punkt* Berechnen Sie $(\rho_P(\sigma_{A<4}(R)) \rtimes (\pi_{X,A}S)) \ltimes S$.

Lösung ✔

P=

A	B	C
0	8	c
1	19	b
2	14	z
3	15	e

Ergebnis=

P.B	P.C	S.X	S.A	S.Z
8	c	b	0	10
14	z	x	2	11
19	b	x	1	12
15	e	z	3	14
null	null	c	5	12

RelaX-Abfrage:
```
(rho P (sigma A < 4 (R)) left outer join (pi X,A (S))) right join S
```

Aufgabe 2 (Relationale Algebra) [6 Punkte]

Gegeben sei das folgende Relationenschema:

```
Customer (CustomerId, FirstName, LastName, Address, Email)
InvoiceParts (InvoicePartId, InvoiceId, TrackId, UnitPrice, Quantity)
Invoice (InvoiceId, CustomerId, InvoiceDate, Total)
Genre (GenreId, Name)
Playlist (PlaylistId, Name)
PlaylistContent (PlaylistId, TrackId)
Artist (ArtistId, Name)
Album (AlbumId, Title, ArtistId)
Track (TrackId, Name, AlbumId, GenreId, Miliseconds, Bytes, UnitPrice)
```

Erstellen Sie auf Basis dieses Relationenschemas die folgenden Anfragen in relationaler Algebra. Sie können dazu RelaX (einen Rechner für relationale Algebra) 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 und den enthaltenen Daten erreichbar: https://dbis-uibk.github.io/relax/calc.htm?data=gist:e8628d74e467b945a564d27d4d74b83e. Geben Sie für die folgenden Aufgaben **sowohl die Abfrage als auch das Ergebnis** ab. Bei sehr großen Ergebnismengen geben Sie bitte die ersten zehn Zeilen und die Gesamtanzahl der Tupel in der Ergebnismenge an.

Hinweis ⚠

Sie können im RelaX-Tool Zwischenergebnisse einer Variable zuweisen und später auf diese zugreifen:

```
Result1 = Artist join (Artist.ArtistId = Album.AlbumId) Album
pi Name, Title Result1
```

a) *1 Punkt* Geben Sie bitte die Id, die CustomerId, das InvoiceDate und die Gesamtsumme (Total) aller Rechnungen aus, die eine Gesamtsumme von über 10 Euro aufweisen.

Lösung ✔

```
pi InvoiceId, CustomerId, InvoiceDate, Total (sigma Total > 10 (Invoice))
```

InvoiceId	CustomerId	InvoiceDate	Total
82	28	2009-12-18	13.86
138	37	2010-08-23	13.86
180	29	2011-02-25	13.86
187	8	2011-03-28	13.86
243	17	2011-12-01	13.86
285	9	2012-06-04	13.86
292	47	2012-07-05	13.86
348	56	2013-03-10	13.86
390	48	2013-09-12	13.86
397	27	2013-10-13	13.86

10 Tupel gesamt[a]

[a] Beachten Sie bitte, dass bei den Musterlösungen die Spaltennamen der angeführten Lösungstabellen aus Gründen der Übersichtlichkeit abgekürzt sind (so wird z.B. aus Invoice.InvoiceId lediglich InvoiceId).

b) *1 Punkt* Geben Sie bitte alle Rechnungen aus, die mit November 2009 datiert sind. Dabei sollten die Id und das Datum der Rechnung, die Gesamtsumme und der Nachname des Kunden ausgegeben werden.

Hinweis ⚠

Datumseinträge werden im Format YYYY-MM-DD gespeichert und angegeben und können mit <, >, etc. verglichen werden. (z.B. `birthday > '1934-01-01'`)

Lösung ✔

```
pi InvoiceId, InvoiceDate, Total, LastName
((sigma InvoiceDate>'2009-10-31' and InvoiceDate<'2009-12-01' (Invoice))
join Invoice.CustomerId = Customer.CustomerId Customer)
```

InvoiceId	InvoiceDate	Total	LastName
70	2009-11-07	1.98	Cunningham
71	2009-11-07	1.98	Barnett
72	2009-11-08	3.96	Francis
74	2009-11-12	8.91	Lefebre

4 Tupel gesamt

c) *1 Punkt* Finden Sie alle Tracks des Genres `Rock`, die auch tatsächlich gekauft wurden. Geben Sie dazu den Namen des Tracks und dessen Id aus.

Lösung ✔

```
pi Track.Name, Track.TrackId
(sigma Genre.Name = 'Rock'
((Track join Track.TrackId = InvoiceParts.TrackId InvoiceParts)
  join Track.GenreId = Genre.GenreId Genre))
```

Track.Name	Track.TrackId
In Your Honor	989
No Way Back	990
The Last Song	994
Free Me	995
Still	999
What If I Do?	1000
Friend Of A Friend	1003
...	

62 Tupel gesamt

d) *1 Punkt* Finden Sie für die vorhandenen Playlists die enthaltenen Tracks. Geben Sie dazu bitte den Namen des Tracks und der Playlist aus.

Lösung ✔

```
pi Playlist.Name, Track.Name
(Playlist
  join Playlist.PlaylistId = PlaylistContent.PlaylistId PlaylistContent
  join Track.TrackId = PlaylistContent.TrackId Track
)
```

Playlist.Name	Track.Name
Music	In Your Honor
Music	No Way Back
Music	Best Of You
Music	DOA
Music	Hell
Music	The Last Song
Music	Free Me
Music	Resolve
Music	The Deepest Blues Are Black
...	...

372 Tupel gesamt

e) *1 Punkt* Geben Sie für alle Kunden, deren Nachname mit 'A' oder 'B' beginnt, die von ihnen gekauften Lieder (Track Name, Artist Name, Album Name) aus und führen Sie auch den Vor- und Nachnamen des Kunden an.

Lösung ✔

```
pi LastName, FirstName, Track.Name -> Trackname,
Artist.Name -> Artistname, Album.Title -> Albumtitle
(sigma LastName < 'C' Customer
join Customer.CustomerId = Invoice.CustomerId Invoice
join Invoice.InvoiceId = InvoiceParts.InvoiceId InvoiceParts
join InvoiceParts.TrackId = Track.TrackId Track
join Track.AlbumId = Album.AlbumId Album
join Album.ArtistId = Artist.ArtistId Artist)
```

LastName	FirstName	Trackname	Artistname	Albumtitle
Barnett	Julia	So Central Rain	R.E.M.	The Best Of R.E.M.: The IRS Years
Barnett	Julia	Pretty Persuasion	R.E.M.	The Best Of R.E.M.: The IRS Years
Barnett	Julia	Gimmie Shelters	Rolling Stones	No Security
Barnett	Julia	Thief In The Night	Rolling Stones	No Security
Barnett	Julia	Out Of Tears	Rolling Stones	Voodoo Lounge
Barnett	Julia	In Your Honor	Foo Fighters	In Your Honor [Disc1]
Barnett	Julia	Free Me	Foo Fighters	In Your Honor [Disc1]
Brown	Robert	Californication	Red Hot Chili Peppers	Californication
Brown	Robert	Road Trippin'	Red Hot Chili Peppers	Californication
Bernard	Camille	Don't Go Back	R.E.M.	The Best Of R.E.M.: The IRS Years
Bernard	Camille	I Believe	R.E.M.	The Best Of R.E.M.: The IRS Years

11 Tupel gesamt

f) *1 Punkt* Finden Sie alle Tracks, die in zwei verschiedenen Playlists vorkommen. In einem ersten Schritt reicht es, lediglich die TrackId dieser Tracks auszugeben.

Lösung ✔

```
pi PlaylistContent1.TrackId
(sigma PlaylistContent1.PlaylistId != PlaylistContent2.PlaylistId
((rho PlaylistContent1 PlaylistContent)
join PlaylistContent1.TrackId = PlaylistContent2.TrackId
  (rho PlaylistContent2 PlaylistContent)))
```

TrackId
989
990
991
992
993
994

147 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 23:59 (Mitternacht) ab.