

15.11.2022

## Übungsblatt 4

### 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)			Tag(PostID, Tag)	
PostID	Headline	Author	PostID	Tag
1	My favourite recipes	Mary Potter	1	cooking
2	Setting up a Linux VM	Jane Doe	1	recipe
3	First Steps in Python	Bob Smith	1	diy
4	Travels 2019	John Doe	3	coding
5	Introduction to SEO	Anne Johnson	3	hacking
6	Knitting a scarf	William Gold	4	usa
7	Getting ready for my first marathon	Alicia Silverstone	4	roadtrip
			4	travels
			6	wool
			6	diy

Berechnen Sie das Ergebnis folgender Abfragen:

- a)  $\sigma_{\text{Headline}=\text{"Travels 2019"}} \text{Post}$
- b)  $\sigma_{\text{Post.PostID} < 4} (\text{Post} \bowtie_{\text{Post.PostID}=\text{Tag.PostID}} \text{Tag})$
- b) ☐ ★ Für die Ausführung der Operationen  $\cup$ ,  $-$ ,  $\cap$  müssen die Schemata der beiden Relationen ident sein - wieso gilt dies nicht für z.B. Joins?
- c) ☐ ★ Was ist der Unterschied zwischen einem NATURAL JOIN und einem EQUIJOIN?
- d) ☐ ★ Berechnen und vergleichen Sie das Ergebnis folgender Abfragen für die gegebenen Relationen  $R$  und  $S$ .
- $R \bowtie S$
  - $R \bowtie_{R.id=S.id} S$

R(id,name,length)		
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)			
id	from_date	to_date	length
1	2019-11-01	2019-11-03	3
4	2019-12-23	2020-01-06	15

e)  Gegeben sei die folgende Relation Drivers<sup>1</sup>:

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

Sie können die Ergebnisse der folgenden Abfragen händisch oder mit dem RelaX-Tool berechnen:

- Berechnen Sie das Ergebnis von  $\gamma_{\text{team}; \text{MAX}(\text{points}) \rightarrow \text{top}, \text{COUNT}(\text{driverID}) \rightarrow \text{numDrivers}}(\text{Drivers})$ .
- Was wird durch Abfrage  $\gamma; \text{AVG}(\text{points}) \rightarrow \text{avg}(\text{Drivers})$  berechnet?

## Hausaufgabenteil (Zuhause zu lösen; Abgabe nötig)

### Aufgabe 1 (Relationale Algebra 1)

[6 Punkte]

Gegeben sei das folgende Relationenschema:

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

Erstellen Sie auf Basis dieses Relationenschemas die folgenden Abfragen 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:

<sup>1</sup>Drivers <https://dbis-uibk.github.io/relax/calc/gist/bb4cec3e91daa1d18539ae56e9855774>

<https://dbis-uibk.github.io/relax/calc/gist/e8628d74e467b945a564d27d4d74b83e>.

Geben Sie für die folgenden Aufgaben **sowohl die Abfrage als auch das Ergebnis und die Anzahl an Tupel** ab. Bei großen Ergebnismengen geben Sie anstatt des gesamten Ergebnisses nur die **ersten zehn Zeilen** an.

Verwenden Sie für die Operatoren die ausgeschriebene Form, nicht die Symbole (pi statt  $\pi$ , join statt  $\bowtie$ , ...).

#### 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) 0.5 Punkte Geben Sie die Id, die CustomerId, das InvoiceDate und die Gesamtsumme (Total) aller Rechnungen aus, die eine Gesamtsumme von kleiner als 5 Euro aufweisen.

#### Abgabe



1a\_query.txt  
 1a\_result.txt

- b) 0.5 Punkte Geben Sie alle Rechnungen aus, die mit November 2009 datiert sind. Dabei sollten die Id und das Datum der Rechnung, die Gesamtsumme und den Nachnamen der 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')

#### Abgabe



1b\_query.txt  
 1b\_result.txt



- c) 0.5 Punkte Finden Sie alle Tracks des Genres Rock, die auch tatsächlich gekauft wurden. Geben Sie dazu den Namen des Tracks und dessen Id aus.

#### Abgabe





1c\_query.txt  
 1c\_result.txt




- d) **0.5 Punkte** Finden Sie für die vorhandenen Playlists die enthaltenen Tracks. Geben Sie dazu den Namen des Tracks und auch den Namen der Playlist aus.

Abgabe	↑
 1d_query.txt	
 1d_result.txt	


- 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.

Abgabe	↑
 1e_query.txt	
 1e_result.txt	


- f) **1 Punkt** Geben Sie die ID, den Titel und den UnitPrice aller Lieder aus, die nicht in Playlist 5 enthalten sind. Finden Sie hierfür zwei Lösungswege: einmal indem Sie den Mengendifferenz-Operator verwenden und einmal über einen Outer Join.

Abgabe	↑
 1f_query_1.txt	
 1f_query_2.txt	
 1f_result.txt	

- g) **1 Punkt** Geben Sie die ID und den Namen aller Künstler aus,
- deren durchschnittliche Songlänge über 4 Minuten und 10 Sekunden liegt *und*
  - deren durchschnittliche Dateigröße der Songs unter 8,5 MB liegt.

Abgabe	↑
 1g_query.txt	
 1g_result.txt	

- h) **1 Punkt** Finden Sie jene Käufer, die nach dem 1. Januar 2010 mindestens drei Lieder eines Albums gekauft haben und geben Sie den Nachnamen und die ID des Käufers, die ID und den Namen des Albums und die Anzahl der gekauften Tracks aus. Benennen Sie die Spalte Title des Albums in Name um.

Abgabe	↑
 1h_query.txt	
 1h_result.txt	

## Aufgabe 2 (Query-Abarbeitung und Optimierung)

[4 Punkte]

Betrachten wir wieder das relationale Modell aus Aufgabe 1:

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

Das Ziel dieser Aufgabe ist es, eine Abfrage auf drei verschiedene Weisen zu implementieren und deren Abarbeitung zu analysieren. Da die naive Implementierung dieser Abfrage sehr Ressourcenintensiv ist, können Sie ihre Abfragen auf einer leeren Datenbank testen. Die leere Datenbank ist unter folgendem Link verfügbar:

<https://dbis-uibk.github.io/relax/calc/gist/1250111065ab1e60e019928fd51dd72b>.

Die mit Daten befüllte Datenbank ist unter folgendem Link verfügbar:

<https://dbis-uibk.github.io/relax/calc/gist/e8628d74e467b945a564d27d4d74b83e>

Für die folgenden Aufgaben soll folgende Anfrage in relationaler Algebra beantwortet werden:

*Finden Sie Vorname, Nachname, Songname und Kaufdatum aller Kunden, die nach dem 01.01.2010 einen Rock-Song gekauft haben.*

- a) 0.5 Punkte Damit man die Effizienz einer Abfrage abschätzen kann, muss man die Größe der enthaltenen Relationen kennen. Finden Sie mittels Relax-Abfragen heraus, wie viele Tupel sich in den folgenden Relationen in der mit Daten befüllten Datenbank befinden und ergänzen Sie die fehlenden Werte. Geben Sie die ausgefüllte Tabelle als Textdatei ab.

Relation	Anzahl Tupel
Artist	
Album	11
Track	
Rock Tracks	
Playlist	
PlaylistContent	
Genre	9
Invoice	32
Invoice after 2010-01-01	
InvoiceParts	
Customer	

### Abgabe



2a.txt

- b) 1 Punkt Formulieren Sie die Abfrage, indem Sie nur **eine** Projektion, **eine** Selektion und beliebig viele Kreuzprodukte (und ggf. Umbenennungen falls nötig/gewünscht) verwenden. Ihre Abfrage soll also dem folgenden Schema entsprechen:

$$\pi_{...}(\sigma_{...}(A \times \dots \times Z)).$$

Führen Sie Ihre Abfrage auf der leeren Datenbank aus und zeichnen Sie den Operatorbaum (bzw. verwenden Sie den von Relax zur Verfügung gestellten) auf. Berechnen Sie außerdem für jeden Knoten im Operatorbaum, wieviele Tupel die Abfrage zu den entsprechenden Zeitpunkten enthält.

#### Abgabe



2b\_query.txt

2b\_tree.pdf

- c) **1 Punkt** Formulieren Sie die Abfrage nun so um, dass die Selektionen, die den jeweiligen Joins entsprechen, auch direkt nach dem Kreuzprodukt ausgeführt werden. Verwenden Sie also hier noch keine expliziten Joins, sondern simulieren Sie diese nach der Regel

$$A \bowtie_c B = \sigma_c(A \times B)$$

#### Abgabe



2c\_query.txt

2c\_tree.pdf

- d) **1.5 Punkte** Wie Sie aus der vorigen Aufgabe wahrscheinlich gesehen haben, lohnt es sich so früh wie möglich die Ergebnismenge durch Selektion zu verringern. Optimieren Sie Ihre Abfrage nun weiter, indem Sie

- 1) anstatt Kreuzprodukt und Selektion einen Join-Operator verwenden
- 2) die Selektionen bzgl. Datum und Genre nach unten schieben
- 3) die Join-Reihenfolge optimieren

Geben Sie für jeden dieser Punkte an, warum diese die Abfrage optimieren, und zeichnen Sie schlussendlich noch einmal einen Operatorbaum mit den optimierten Tupelzahlen pro Knoten.

#### Abgabe



2d\_query.txt

2d\_tree.pdf

2d\_explanation.txt

Wenn Sie diese Aufgabe gelöst haben, probieren Sie die naive Abfrage aus 2b) und die optimierte Abfrage auf der mit Daten befüllten Datenbank auszuführen. Merken Sie den Unterschied?

**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.