Datenbanksysteme WS 2017/18
Prof. Dr.-Ing. Sebastian Michel
Manuel Hoffmann, Marc Dörr
TU Kaiserslautern, FB Informatik – Lehrgebiet Informationssysteme
http://dbis.informatik.uni-kl.de
Übungsblatt 8: Ausgabe 19.12.2017, Abgabe 08.01.2018 16:00 MEZ

 [1]

Wir wünschen frohe Weihnachten und einen guten Rutsch.

Aufgabe 1: Top-k Algorithmen (1 P.)

a) Wenden Sie für die gegebenen drei Indexlisten den TA-Algorithmus für $k = 2$ mit Addition als Aggregationsfunktion an. Geben Sie alle Indexlistenzugriffe an sowie die aktuellen top-k Dokumente nach jedem dieser Schritte. Wie viele sequenzielle und wie viele wahlfreie Zugriffe sind insgesamt angefallen?

L_1
d_1 0.9
d_3 0.5
d_7 0.4
d_2 0.3
d_4 0.2
d_5 0.2
d_6 0.1

L_2
d_3 0.8
d_4 0.8
d_7 0.7
d_1 0.3
d_6 0.2
d_5 0.2
d_2 0.2

L_3
d_1 0.7
d_6 0.6
d_4 0.5
d_7 0.5
d_2 0.3
d_3 0.1
d_5 0.1

b) Gegeben folgender Top-k-Algorithmus:

1. Wähle zufällig eine der Listen aus
2. Hole die obersten k Einträge, $s :=$ Score des k-ten Eintrags
3. Hole aus allen anderen Listen alle Einträge mit Score $\geq s$
4. Aggregiere alle geholten Scores und gib die Top-k zurück

Ist dieser Algorithmus korrekt? Begründen Sie ihre Antwort.

c) Betrachten Sie die folgenden Aggregationsfunktionen:

i) *max*

ii) *spread* = *max* - *min*, wobei *max* und *min* sich auf die Scores des selben Dokuments in unterschiedlichen Listen beziehen.

Geben Sie für jede dieser Aggregationsfunktionen einen "optimalen" Algorithmus an zur Berechnung der top-k Ergebnisse. Passen Sie dazu den TA-Algorithmus an die Besonderheiten der Aggregationsfunktionen an. Wenden Sie die Algorithmen auf die Daten aus Aufgabenteil a) an und vergleichen Sie die Performance der Algorithmen zum originalen TA-Algorithmus.

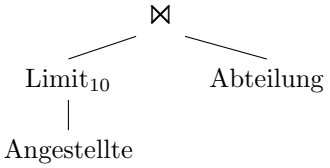
[1] Quelle: openclipart.org

Aufgabe 2: LIMIT-Operator (1 P.)

a) Die SQL-Anweisung `LIMIT` beschränkt die Ausgabe auf eine vorgegebene Anzahl von Tupeln, üblicherweise/sinnvollerweise nach einer Sortierung (ASC oder DSC) eines oder mehrerer Attribute. Das Datenbanksystem sollte natürlich versuchen diesen Operator möglichst früh im Operatorbaum auszuführen, damit Kosten minimiert werden. Aber natürlich darf durch die "optimierte" Platzierung des Operators das Ergebnis der Anfrage nicht verändert werden. Betrachten Sie die drei folgenden SQL-Anfragen mit jeweils angegebenen Operatorbaum. Beschreiben Sie wieso der Operatorbaum korrekt bzw. inkorrekt bzgl. der entsprechenden Anfrage ist. Geben Sie, optional, ein möglichst allgemeines Kriterium für Korrektheit an.

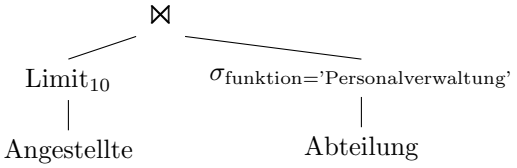
i)

```
SELECT *
FROM Angestellte an, Abteilung ab
WHERE an.arbeitet_in = ab.id
ORDER BY an.salary DESC
LIMIT 10
```



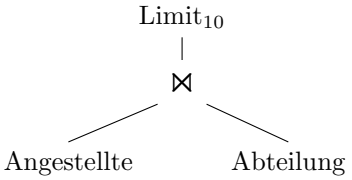
ii)

```
SELECT *
FROM Angestellte an, Abteilung ab
WHERE an.arbeitet_in = ab.id
  AND ab.funktion = 'Personalverwaltung'
ORDER BY an.gehalt DESC
LIMIT 10
```

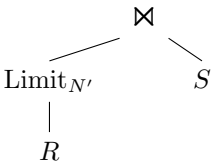


iii)

```
SELECT *
FROM Angestellte an, Abteilung ab
WHERE an.arbeitet_in = ab.id
ORDER BY ab.budget
LIMIT 10
```



b) Eine aggressive Strategie platziert Limit-Operatoren so früh wie möglich. Natürlich muss das Attribut nachdem sortiert werden soll dort bekannt sein. Gegeben eine einfache Anfrage $Limit_N(R \bowtie S)$ über zwei Relationen $R(A, B)$ und $S(A, C)$, die N Tupel zurückliefern soll. Der Limit-Operator beziehe sich auf die höchsten N Werte von $R.B$. Der Limit-Operator wurde wie folgt dargestellt heruntergeschoben:



Wie kann im Allgemeinen die Kardinalität N' der zurückgelieferten Tupel in des nach unten geschobenen Limit-Operatoren in Abhängigkeit von N sinnvoll gesetzt werden kann? Dabei dürfen die vom Datenbanksystem bekannten Kardinalitätsschätzungen verwendet werden. Was kann dabei schief gehen?

Aufgabe 3: Skyline Anfragen (1 P.)

a) Gegeben ist die folgende Tabelle mit Handytarifen verschiedener Provider:

ID	preis	freiminuten	freisms	netz	handy
1	9.95	100	50	Vodafone	Android
2	4.95	100	100	E-Plus	Android
3	9.95	100	100	Telekom	Android
4	19.95	200	100	O2	iPhone
5	14.95	100	50	Vodafone	Android
6	14.95	50	50	E-Plus	Android
7	18.95	200	100	Telekom	Android
8	18.95	200	100	Telekom	iPhone
9	12.95	100	50	Telekom	Android
10	12.95	0	200	Vodafone	Android

Gesucht ist die Skyline über die folgenden fünf Dimensionen:

- *preis:* billiger ist besser,
- *freiminuten*, *freisms:* mehr ist besser,
- *netz:* Telekom ist besser als Vodafone, Vodafone besser als O2; O2 und E-Plus sind gleich gut,
- *handy:* nicht vergleichbar.

Auf der Vorlesungs-Webseite finden Sie die SQL-Statements zur Erzeugung der Tabelle *handytarif* sowie der Funktion *schlechter(netz, netz)*, die `true` liefert, wenn das erste Netz schlechter als das zweite ist.

- Erstellen Sie eine SQL-Anfrage zur Bestimmung der Skyline (ohne den `SKYLINE`-Operator zu benutzen) über alle Handytarife mit Preis $\geq$ 10 Euro.

b) Betrachten wir den Fall, dass bzgl. Dominanz "kleinere Werte" besser sind als größere Werte und ferner Werte aus $\mathbb{R}^+$. Zeigen Sie, dass der Nächste Nachbar (NN) zum Ursprung (0,0) auch immer in der Skyline ist.

Aufgabe 4: Transaktionen (1 P.)

a) Für eine Datenbank gebe es die Konsistenzbedingung $0 \leq A \leq B$. Beschreiben Sie jeweils für die folgenden Transaktionen, ob die Konsistenzbedingung erfüllt ist oder nicht. Begründen Sie Ihre Antworten.

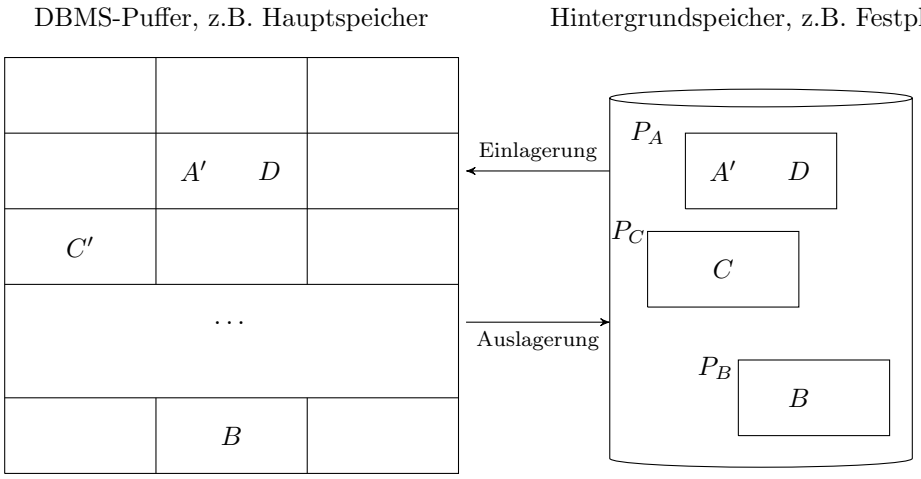
T_1 : `B:= A+B; A:= A+B;`

T_2 : `A:= B+1; B:= A+1;`

T_3 : `A:= A+B; B:= A+B;`

b) Geben Sie für jede Transaktion aus Aufgabenteil a) die Lese- und Schreibaktionen an und stellen Sie den Effekt dieser Aktionen auf Hauptspeicher und Festplatte dar. Gehen Sie dabei initial von $A = 5$ und $B = 10$ aus. Beschreiben Sie ferner ob es möglich ist, mit passender Ordnung der OUTPUT-Aktionen, dass Konsistenz auch in dem Fall eines Absturzes während die Transaktionen laufen gewahrt wird.

c) Gegeben zwei Transaktionen T_1 und T_2 und folgende Situation der Seiten $\mathsf{P_A}$, $\mathsf{P_B}$ und $\mathsf{P_C}$ sowie der Datensätze A, B, C und D, wobei $'$ geänderte Datensätze markieren.



In folgender Tabelle sind Operationen von T_1 und T_2 zu verschiedenen Zeitpunkten gegeben.

Zeit	T_1	T_2
0	READ(A,a)	READ(C,c)
10	a:=a+10	c:=c*2
20		READ(B,b)
30	WRITE(A,a)	
40	READ(D,d)	b:=b+c/4
50	d:=17*d+42	
60	OUTPUT(A)	WRITE(C,c)
70	WRITE(D,d)	OUTPUT(C)
80	COMMIT	
90		COMMIT

Diskutieren Sie, ob die obige Abbildung den Ablauf bzgl. Einlagerung und Auslagerung von Seiten gemäß der Tabelle widerspiegelt, ob dies nur zu gewissen Zeitpunkten passt (wenn ja, zu welchen und wieso) oder ob es nicht bzw. nie passt.

Wenn das System zu einem dieser Zeitpunkte abstürzt, was müsste beim Wiederanlauf getan werden, um ACID zu garantieren? Welche Teile von ACID sind hier betroffen? Wie ändert sich die Situation zum Zeitpunkt 91?