TU München, Fakultät für Informatik
Lehrstuhl III: Datenbanksysteme
Prof. Dr. Thomas Neumann

Übung zur Vorlesung
Einsatz und Realisierung von Datenbanksystemen im SoSe15

Moritz Kaufmann (moritz.kaufmann@tum.de)
http://db.in.tum.de/teaching/ss15/impldb/

Blatt Nr. 10

Hausaufgabe 1

Sie eine SQL-Anfrage, die basierend auf dem Schema aus Abbildung 1 einen dreidimensionalen Quader berechnet, der es unserem Handelsunternehmen erlaubt, entlang der folgenden Dimensionen drill-down/roll-up Anfragen zu stellen:

- Produkttyp,
- Bezirk,
- Alter.

Das Handelsunternehmen ist dabei nur an Daten aus Deutschland interessiert, die in die Hochsommersaison fallen. Verwenden Sie den **cube**-Operator.

Verkäufe					
VerkDatum	Filiale	Produkt	Anzahl	Kunde	Verkäufer
25-Jul-00	Passau	1347	1	4711	825
…	…	…	…	…	…

Filialen			
Filialenkennung	Land	Bezirk	…
Passau	D	Bayern	…
…	…	…	…

Kunden			
KundenNr	Name	wiealt	…
4711	Kemper	43	…
…	…	…	…

Verkäufer					
VerkäuferNr	Name	Fachgebiet	Manager	wiealt	…
825	Handyman	Elektronik	119	23	…
…	…	…	…	…	…

Zeit								
Datum	Tag	Monat	Jahr	Quartal	KW	Wochentag	Saison	…
…	…	…	…	…	…	…	…	…
25-Jul-00	25	Juli	2000	3	30	Dienstag	Hochsommer	…
…	…	…	…	…	…	…	…	…
18-Dec-01	18	Dezember	2001	4	52	Dienstag	Weihnachten	…
…	…	…	…	…	…	…	…	…

Produkte					
ProduktNr	Produkttyp	Produktgruppe	Produkthauptgruppe	Hersteller	…
1347	Handy	Mobiltelekom	Telekom	Siemens	…
…	…	…	…	…	…

Abbildung 1: Schema des Handelsunternehmens.

Siehe Übungsbuch

Hausaufgabe 2 - Top-K Berechnung

Die in Abbildung 2 dargestellten Relationen Autos und Unterhalt dienen der Bewertung von Autos. Eine junge Studierende sucht ein Auto mit guter Balance zwischen Sportlichkeit und Kosten. Sie überlegt sich wie die drei Werte Preis, PS und monatlicher Unterhalt in einen Score umberechnet werden können und nutzt schließlich folgende Formel:

$Preis - (100 * PS) + 24 * Unterhalt$

Zeigen Sie die phasenweise Berechnung der Top-3 Ergebnisse jeweils mit dem Threshold- und dem NRA-Algorithmus. Prüfen sie vor der Berechnung ob Teile der Scoringformel schon innerhalb jeder Relation vorberechnet werden können.

Auto	Preis	PS
Seat Leon	25000€	200
Audi A1	17000€	96
Citroen DS 4	20679€	100
Mini One	16500€	75
Mercedes C-Klasse	35000€	160
Porsche Cayenne	80100€	420

Auto	Unterhalt p. Monat
Seat Leon	215€
Audi A1	220€
Citroen DS 4	225€
Mini One	262€
Mercedes C-Klasse	290€
Porsche Cayenne	430€

Abbildung 2: Autokauf und -Unterhaltskosten.

Der erste Teil der Formel $Preis - (100 * PS)$ kann schon für jede Reihe in der Relation Autokauf vorberechnet werden.

Auto	PreisPS
Seat Leon	5000
Audi A1	7400
Mini One	9000
Citroen DS 4	10679
Mercedes C-Klasse	19000
Porsche Cayenne	38100

Bei der Top-3 Berechnung wird dann mit dieser sortierten Relation und der sortierten Unterhaltskosten Relation gearbeitet.

Threshold Algorithmus

Zw. Ergebnis: Phase 1

Auto	Score
Threshold	10160
Seat Leon	10160

Zw. Ergebnis: Phase 2

Auto	Score
Seat Leon	10160
Threshold	12680
Audi A1	12680

Zw. Ergebnis: Phase 3

Auto	Score
Seat Leon	10160
Audi A1	12680
Threshold	14400
Mini One	15288
Citroen DS 4	16079

Zw. Ergebnis: Phase 4

Auto	Score
Seat Leon	10160
Audi A1	12680
Mini One	15288
Citroen DS 4	16079
Threshold	16967

NRA Algorithmus

NRA: Phase 1

Auto	Score
Seat Leon	10160

NRA: Phase 2

Auto	Score
Seat Leon	10160
Audi A1	12680

NRA: Phase 3

Auto	Score
Seat Leon	10160
Audi A1	12680
Citroen DS 4	14400[p]
Mini One	14400[p]

NRA: Phase 4

Auto	Score
Seat Leon	10160
Audi A1	12680
Mini One	15288
Citroen DS 4	16079

Hausaufgabe 3 - Skyline

Geben die Relation Klausur:

MatrNr	Vorbereitungszeit	Note
1	150	1.7
2	70	2.7
3	450	2.0
4	180	1.7
5	2500	1.3

a) Formulieren Sie die Anfrage, die die MatrNr in der Skyline für die Attribute Vorbereitungszeit und Note erzeugt (kleiner ist jeweils besser) in SQL mit Hilfe des Skyline Operators.

b) Formulieren Sie die Anfrage in SQL ohne Skyline Operator.

c) Bestimmen Sie das Ergebnis der Anfrage.

SQL mit Skyline:

select MatrNr from Klausur k skyline of k.Vorbereitungszeit min, k.Note min

SQL ohne Skyline:

```
select MatrNr from Klausur k
where not exists (
select * from klausur dom
where
dom.Vorbereitungszeit <= k.Vorbereitungszeit and
dom.Note <= k.Note and (
dom.Vorbereitungszeit < k.Vorbereitungszeit or
dom.Note < k.Note)
)
```

Ergebnis:

1) Ist in Skyline (Kann in Vorbereitungszeit nur von MatrNr 2 dominiert werden, dort ist aber Note schlechter)
2) Ist in Skyline (Minimum für Vorbereitungszeit)
3) Ist nicht in Skyline, dominiert von MatrNr 1
4) Ist nicht in Skyline, dominiert von MatrNr 1
5) Ist in Skyline (Minimum für Note)

Hausaufgabe 4 - Frequent Itemsets

Zeigen Sie die weiteren Phasen des À priori-Algorithmus für unser Beispiel in Abbildung 3 (hier ist lediglich bis inkl. 2. Phase dargestellt). Damit eine Menge von Produkten ein frequent itemset ist, muss sie in mindestens 3/5 aller Verkäufe enthalten sein, d.h. $minsupp = 3/5$.

VerkaufsTransaktionen	
TransID	Produkt
111	Drucker
111	Papier
111	PC
111	Toner
222	PC
222	Scanner
333	Drucker
333	Papier
333	Toner
444	Drucker
444	PC
555	Drucker
555	Papier
555	PC
555	Scanner
555	Toner

Zwischenergebnisse	
FI-Kandidat	Anzahl
{Drucker}	4
{Papier}	3
{PC}	4
{Scanner}	2
{Toner}	3
{Drucker, Papier}	3
{Drucker, PC}	3
{*Drucker, Scanner*}	
{Drucker, Toner}	3
{Papier, PC}	2
{*Papier, Scanner*}	
{Papier, Toner}	3
{*PC, Scanner*}	
{PC, Toner}	2
{*Scanner, Toner*}	

Abbildung 3: Ausgangssituation für den À priori-Algorithmus

Siehe Übungsbuch

Hausaufgabe 5 - Row vs Column - Store

Gegeben eine Tabelle *Produkte* mit folgendem Schema und 10000 Einträgen:

Id (8 Byte) | Name (32 Byte) | Preis (8 Byte) | Anzahl (8 Byte)

Wieviele Daten werden für folgende Queries in den CPU-Cache geladen? Unterscheiden sie jeweils zwischen Row und Column Store.

1. *select * from Produkte*
2. *select Anzahl from Produkte*

Daten können maximal mit Cacheline Granularität (64 Byte) in den Cache geladen werden. Das heißt, selbst wenn nur auf einen 64 bit Integer Wert zugegriffen wird, muss die komplette Cacheline geladen werden. Mit diesem Hintegrund ergeben sich folgende Ergebnisse:

1. *select * from Produkte*
   a) Row: $10000 * 56 = 560000$ Byte

b) Column: $10000 * 8 + 10000 * 32 + 10000 * 8 + 10000 * 8 = 560000$ Byte

2. *select Anzahl from Produkte*

   a) Row: $10000 * 56 = 560000$ Byte

   b) Column: $10000 * 8 = 80000$ Byte