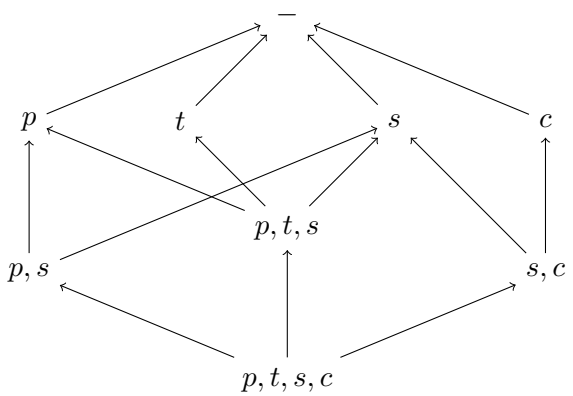
5. Übungsblatt

Besprechung: 20.06. Gruppe A+C – 21.06. Gruppe B+D

Aufgabe 1

Ein Data Warehouse-Schema enthält die drei Dimensionen p, t, s und c. Mögliche Gruppierungen und mögliche Ableitungen davon ergeben sich durch folgenden Verband (*lattice*). Die Tabelle gibt die Ergebnisgröße der möglichen Gruppierungen an.



Gruppierung	# rows
p	10
t	20
s	30
c	400
p, s	100
p, t, s	4 000
s, c	3 000
p, t, s, c	10 000

Zur Beschleunigung von Anfragen sollen materialisierte Sichten angelegt werden. Dazu ist ein (Speicherplatz-)Budget von **15 000 Tupeln** vorgesehen. Ermitteln Sie die optimale Menge von anzulegenden materialisierten Sichten. Verwenden Sie dazu das Maximum Benefit-Verfahren, das in Übungsprojekt 5 basierend auf dem Artikel [1] vorgestellt wird. Notieren Sie dabei die einzelnen Schritte und protokollieren Kosten bzw. Benefits für jeden Schritt.

Aufgabe 2

Eine wichtige Operation zum Abschätzen des Ergebnisses eines Joins ist das Testen auf Enthaltensein. Kann ausgeschlossen werden, dass der Joinschlüssel eines Tupels in der zu verknüpfenden Tabelle enthalten ist, braucht dieses Tupel für den Join nicht berücksichtigt werden. Der Bloom Filter ist eine probabilistische Datenstruktur, die das effiziente Testen auf Enthaltensein auf einer Menge $\mathcal{M}$ als Abschätzung ermöglicht. Der Filter besteht aus einem m dimensionalen Bitvektor and k Hashfunktionen, die Werte auf eine Zahl von 0 bis m-1 abbilden. Jeder Wert in $\mathcal{M}$

wird mit jeder der k Funktionen gehasht und für das Ergebnis an der entsprechenden Position eine 1 in den Bitvektor eingetragen.

Für eine weitere Menge $\mathcal{T}$ als Testmenge soll eine möglichst große Menge $\mathcal{N} \subseteq \mathcal{T}$ bestimmt werden von Werten in $\mathcal{T}$, die nicht in $\mathcal{M}$ enthalten sind, für die also $\mathcal{N} \cap \mathcal{M} = \emptyset$ gilt. Dazu wird jeder Wert t in $\mathcal{T}$ mit jeder der k Funktionen gehasht und geprüft, ob für das Ergebnis an der entsprechenden Position eine 0 im Bitvektor eingetragen ist. Ist dies für mindestens ein Ergebnis der Fall, wird t in $\mathcal{N}$ aufgenommen.
a) Veranschaulichen Sie sich zunächst die Funktionsweise eines Bloom Filters für zwei kleine Beispielmengen $\mathcal{M}$ und $\mathcal{T}$ von ganzen Zahlen und zwei einfache Hashfunktionen.
b) Skizzieren Sie dann die Verwendung eines Bloom Filters für folgende Anfrage auf einem Stern-Schema

```
select sum(f.SalesAmount) as Betrag,
  SalesTerritoryRegion as Verkaufsgebiet,
  dd.EnglishMonthName as Monat
from  FactResellerSales f  inner join DimSalesTerritory dt
            on f.SalesTerritoryKey=dt.SalesTerritoryKey
                  inner join DimDate dd on f.OrderDateKey=dd.DateKey
where dt.SalesTerritoryCountry = 'United States'
and dd.CalendarYear=2008
group by SalesTerritoryRegion, dd.EnglishMonthName
```

c) Für die Fehlerwahrscheinlichkeit p, dass ein $t \in \mathcal{T}$ mit $t \notin \mathcal{M}$ nicht in $\mathcal{N}$ aufgenommen wird, kann folgende Formel verwendet werden

$$p = (1 - (1 - \frac{1}{m})^{k \cdot n})^k$$

wobei m die Länge des Bitvektors, k die Zahl der Hashfunktionen und $n =\mid \mathcal{M} \mid$ ist. Erklären Sie, warum die Formel diese Fehlerwahrscheinlichkeit berechnet. Bestimmen Sie die Mindestlänge des Bitvektors für $k = 7$, $n = 5000$ unter der Bedingung $p \leq 0.01$.

Literatur

[1] Venky Harinarayan, Anand Rajaraman und Jeffrey D. Ullman: Implementing Data Cubes Efficiently. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data. ACM Press, 1996, Seiten 205-216.