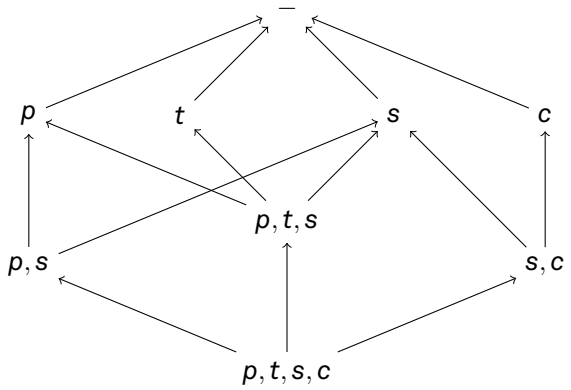
4. Übungsblatt

Ausgabe: 27. Mai 2020 · Besprechung: ab. 10. Juni 2020

Aufgabe 1

Ein Data Warehouse-Schema enthält die vier 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 Budget an Speicherplatz für **15 000 Tupeln** vorgesehen. Ermitteln Sie die optimale Menge von anzulegenden materialisierten Sichten. Verwenden Sie dazu das Maximum Benefit-Verfahren von Harinarayan [**?**], das auch in den Übungsprojekten vorgestellt wird. Notieren Sie dabei die einzelnen Schritte und protokollieren Sie Kosten bzw. Benefits für jeden Schritt.

Aufgabe 2

Die folgende Sternschema-Anfrage soll mit indexbasierten Strategien ausgewertet werden.

```
select sum(Sales.Revenue)
  from Sales, Territory, Date
 where Sales.TerritoryKey = Territory.Key
   and Sales.DateKey = Date.Key
   and Territory.Country = 'United States'
   and Date.CalendarYear between 2005 and 2008
```

Dazu zeigt Abbildung **??** einen Plan für die Auswertungsstrategie *„Index on value columns of dimension tables“* (Strategie 1, Vorlesungsfolie 117). Oben in der Abbildung wird der Anfrageplan dargestellt; unten sehen Sie vier Indizes die bei der Auswertung genutzt werden.

Im Allgemeinen läuft die Auswertung von Anfrageplänen folgendermaßen ab. An den Blättern werden Tupel aus den Tabellen der Datenbank gelesen und an die weiteren Operatoren des Plans bis zum Ergebnis an der Wurzel weiter gereicht. Im Detail ist Strategie 1 im Folgenden.

Auswertung des Anfrageplans

Zunächst werden die Dimensionstabellen *Territory* und *Date* unabhängig voneinander gelesen. Dabei werden die Indizes $\texttt{IX}_1$ und $\texttt{IX}_3$ in *Index Scans* (`IXSCAN`) genutzt um die *Record Identifier* (`rids`) jener Tupel zu bestimmen, die die Filterkriterien erfüllen. Danach lesen `FETCH`-Operationen die Tupel mit den entsprechenden `rids`.

Nun wird je Dimension ein *Index Nested Loops Join* (`INLJ`) mit der Faktentabelle berechnet. Die Indizes $\texttt{IX}_2$ und $\texttt{IX}_4$ liefern dabei die `rids` der passenden Faktentupel. Anschließend wird die Schnittmenge der beiden `rid`-Listen berechnet um die Faktentupel zu bestimmen, die zu beiden Dimensionen passen. Die Faktentupel werden mit einer `FETCH`-Operation gelesen und als Ergebnis aggregiert.

a) Verbesserung von Strategie 1

Auf Vorlesungsfolien 120 und 121 wird ein Trick erwähnt, der die Effizienz von Strategie 1 steigert. Geben Sie eine abgewandelte Form des Anfrageplans an der die Variante umsetzt. Spezifizieren Sie auch die verwendeten Indizes.

b) Strategie 2

Geben Sie einen Plan an, der die Anfrage mit Strategie 2 *„Index on primary key of dimension tables“* (Folie 118) umsetzt. Welche Indizes werden für den Plan benötigt?

c) Hub Join

Welche Indizes werden benötigt um die Anfrage mit einem Hub-Join zu verarbeiten?

d) Vergleich

Nennen Sie Vor- und Nachteile der jeweiligen Techniken.

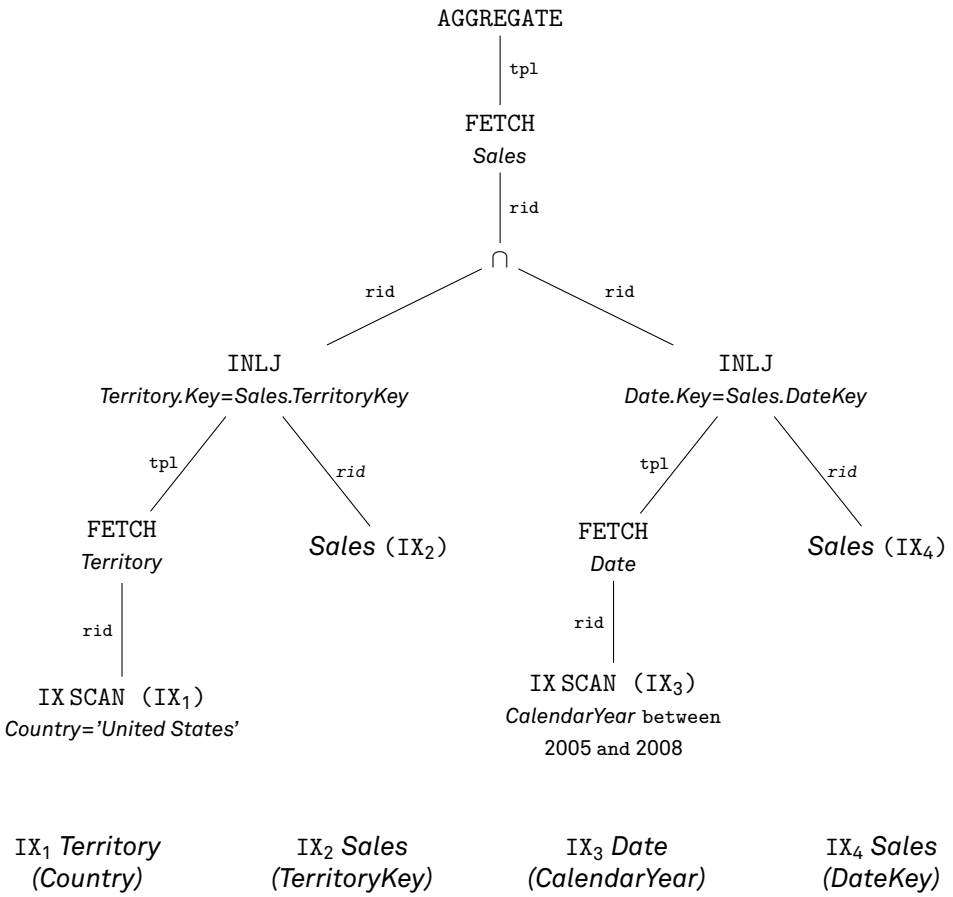


Abbildung 1: Anfrageplan für Index-Strategie 1 (Folie 117)

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.