

Optimizing queries using materialized views: a practical, scalable solution

Goldstein and Larson, ACM SIGMOD, 2001

Questions

Philipp

1. Was ist das view-matching Problem, welches in der Arbeit Beachtung findet? Was sagt es aus?
2. In der Arbeit werden die drei verschiedenen Tests besprochen, um zu zeigen, dass alle Reihen die von der Query benötigt werden in der View enthalten sind. Welche waren die drei Tests? Warum vereinfacht die Abwesenheit von Disjunktionen bei einem der Tests den Algorithmus bzw. warum könnte es den Algorithmus komplexer machen? (Betrifft konkret den Range subsumption Test und die Aussage, dass es ohne Disjunktionen einfacher ist. Leider geht nicht hervor, was das Problem hierbei war.)
3. In der Arbeit von Goldstein und Larson wird das Fast Filtering von Views mittels "filter trees" und "lattice index" vorgestellt. Für die späteren Experimente und deren Auswertung haben sie einen Baum verwendet, welcher 8 Levels/Ebenen hat und sich wie folgt aus diesen Partitioning Conditions zusammensetzt:
From top to bottom, the levels are: hubs, source tables, output expressions, output columns, residual constraints, and range constraints. For aggregation views, there are two additional levels: grouping expressions and grouping columns."
Dieser Filter Tree wirkt sich natürlich in seiner Zusammensetzung auf die Ergebnisse aus, ist dies der am ehesten optimale Baum oder gibt es womögliche bessere Anordnungen? Oder folgt der Aufbau logisch aus den verschiedenen Partitioning Conditions? Wieso kommen dann hubs vor source tables? Warum immer expressions vor columns?

Vasker

1. Zeigen Sie dass $W_q \Rightarrow W_v$ gilt, falls das Ergebniss von SELECT Anweisung $SELECT * FROM T_1, T_2, \dots, T_n WHERE W_q$ erzeugt eine Untermenge des $SELECT * FROM T_1, T_2, \dots, T_n WHERE W_v$ Anweisung.
2. Welche Bedingungen muss ein View erfüllen, um Aggregat Query von View zu berechnen?
3. Was ist ein filter tree?

Daniel

1. How does the decision algorithm (to determine if all required rows exist in a view) work in principle and what are the three tests applied?
2. Why is it not sufficient to just keep view information in the memory to speed up view matching? Describe briefly the data structure used to solve this problem.
3. What experiments have been done and what do the results tell us about the performance and the scalability of the approach introduced by this paper?