

3. Übungsblatt

Ausgabe: 45. Kalenderwoche · Besprechung: 46/47. Kalenderwoche

Aufgabe 0

- Update Rules: Erklären Sie die unterschiedlichen Typen aus der Vorlesung und geben Sie Beispiele!
- Erklären Sie was ein materialisierter View ist!
- Geben Sie Beispiele in SQL für das Anlegen von materialisierten Views!
- Was versteht man unter einem Querylog?

Aufgabe 1

Ein wichtiger Aspekt beim Verwenden von materialisierten Sichten oder Views ist die Auswahl dieser aus der Menge aller möglichen Views. Dies ist besonders unter Berücksichtigung der Kosten für das Anlegen dieser Views wichtig. Das Paper im Anhang behandelt nun Wege und Strategien um die k wichtigsten Views für einen OLAP Cube zu bestimmen. Lesen Sie das Paper gründlich durch und lösen Sie folgende Aufgaben:

Beschreiben Sie den Lattice für mehrere hierarchische Dimensionen! Geben Sie Beispiele an! Beschreiben Sie wie man mehrere hierarchische Dimensionen in einem solchen Lattice zusammenfasst! Erklären Sie das lineare Kostenmodell und den Greedy-Algorithmus zur Auswahl von k Views zum Materialisieren (inklusive Pseudocode)! Nennen Sie Möglichkeiten um den Lattice mit Gewichten für die Views auszustatten! Diese Gewichte können dann im Greedy-Algorithmus berücksichtigt werden (Tipp: Sie haben ein Querylog.).

Aufgabe 2

Eine wichtige Operation das Abschätzen der Resultgröße für Joins ist der Testen auf Enthaltensein. Der Bloom Filter ist eine probabilistische Datenstruktur, die das effiziente Testen auf Enthaltensein auf einer Menge ermöglicht. Der Filter besteht aus einem m dimensionalen Bitvektor and k Hashfunktionen, die Werte auf ein Zahl von 0 bis $m-1$ abbilden. Die Werte können

Zahlen aber auch Strings sein. Das Einfügen in den Bloom Filter geschieht über k faches Hashen. Alle k Hashfunktionen werden auf den neu einzufügenden Wert angewendet. Die Hashwerte sind nun Positionen im Bitvektor die auf Eins gesetzt werden. Der Test auf Enthaltensein funktioniert so, dass der zu testende Wert auf den k Hashfunktionen angewendet wird. Sind jetzt alle Hashwerte Positionen im Bitvektor die auf Eins stehen, so sagen wir der Wert ist enthalten. Die Wahrscheinlichkeit eines falschen positiven Treffer ist $p = (1 - (1 - \frac{1}{m})^{k \cdot n})^k$ nachdem n Werte in den Bloom Filter eingefügt worden.

Gegeben Sei ein Bloom Filter mit 10 Hashfunktionen und einem 10.000 dimensionalen Bitvektor. Nach wievielen eingefügten Werten ist die Wahrscheinlichkeit eines falschen positiven Treffers größer als 0,5.

Implementieren Sie einen Bloom Filter in Java! Ihre Implementierung soll die zwei Parameter k und m übergeben bekommen. Schreiben Sie eine generische Methode zum Einfügen und zum Testen auf Enthaltensein. Ihre Einfügemethode soll eine Exception schmeißen, sobald die Wahrscheinlichkeit eines falschen positiven Treffers größer als 0.5 ist. Implementieren Sie spezielle Versionen dieser Methoden für Strings, Double und Integerwerte. Welche Hashfunktionen könnten Sie verwenden? Testen Sie Ihre Implementierung für zufällige Zahlen und ausgedachte Stringwerte. Geben Sie Ihre Implementierung bitte ausgedruckt und kommentiert in der Übung ab!