Rozwiązania zadań z egzaminu z baz danych

Krzysztof Ciebiera Janusz Dutkowski Zbigniew Jurkiewicz
Paweł Kucharczyk Krzysztof Stencel

6 lutego 2008

Zadanie 1 (JD)

Tabela ma trzy kolumny A, B i C. Znaleźć taki zbiór zależności funkcyjnych w tej tabeli, żeby jego domknięcie było jak najliczniejsze. Podać jakąkolwiek reprezentację tego zbioru i moc jego domknięcia. Uzasadnić.

Rozwiązanie Największe domknięcie to w tym wypadku domknięcie pełne. Rozważmy zbiór zależności:

$$\begin{aligned} A &\to B \\ B &\to C \\ C &\to A \end{aligned}$$

Wtedy jakikolwiek niepusty zbiór kolumn postawimy po lewej stronie zależności, po prawej możemy postawić dowolny niepusty ich podzbiór. Niepustych podzbiorów zbioru 3-elementowego jest 7, więc zależności (włącznie z wtórnymi, oczywistymi itd.) 49.

Zadanie 2 (PK)

SZBD ma bufory pamięci operacyjnej na 100 bloków dyskowych. Tabela `Zamowienia (numer NUMBER(10), towar VARCHAR(20), ilosc NUMBER(10))` zajmuje 10000 bloków dyskowych. W jednym bloku mieszczą się informacje o 50 zamówieniach. Wykonano nastepujące zapytanie:

```
SELECT towar
  FROM Zamowienia
  GROUP BY towar
  HAVING SUM(ilosc) >= ALL (SELECT sum(ilosc) FROM Zamowienia GROUP BY towar);
```

Podaj jak najefektywniejszy plan wykonania tego zapytania i oszacuj koszt tego planu (podaj liczbę operacji odczytu i zapisu bloku dyskowego) przy założeniu, że nie korzystamy z żadnych indeksów i że na poczatku wszystkie bufory są puste a dane są tylko na dysku.

Rozwiązanie Rozwiązanie jest oparte o sortowanie przez scalanie (*merge-sort*) tabeli wg kolumny towar, z tym, że wprowadzone są pewne modyfikacje w celu zmniejszenia kosztu całości. Oto algorytym:

1. Wczytywanie po 100 bloków, sortowanie ich wg. towaru i zapisywanie. Otrzymujemy 100 posortowanych fragmentów.
2. Scalanie 99 fragmentów otrzymanych z 1. (setny blok bufora jest potrzebny, aby to scalanie wykonywać).

3. Scalanie wyniku 2. i ostatniego fragmentu z 1. Ale uwaga - to już ostatni krok sortowania, więc można obliczać sum(ilosc) i pamiętać (jedna liczba - zmienna w programie) aktualnie największą wartość sum(ilosc). Do realizacji zapytania wystarczy dla każdej grupy zapisywać pary <towar, sum(ilosc)> (niestety w pesymistycznym wypadku to nie pomniejsza kosztu!).

4. Przechodzimy przez wynik 3. wypisując te towary, dla których sum(ilosc) równa się największej wartości zapamiętanej w 3.

Koszty:

Krok	Odczyty	Zapisy
1.	10000	10000
2.	9900	9900
3.	10000	10000
4.	10000	0
Razem	39900	29900

Możliwe jest jeszcze kilka sztuczek/udoskonaleń:

- W 1. można nie zapisywać jednego bloku, który zaraz będzie użyty do scalania. Wtedy oczywiście również w 2. jeden mniej odczyt.
- W 2. można nie zapisywać ostatniego scalonego bloku - przy założeniu, że 3. będzie wykonywane o odwrotnej kolejności. Wtedy oczywiście również w 3. jeden mniej odczyt.
- W 3. można nie zapisywać ostatnich 98 buforów (dwa są potrzebne do odczytu przy scalaniu), to redukuje też o 98 liczbę odczytów w 4.

Kilka uwag:

- Nie można zakładać nic o liczbie różnych towarów w tabeli - w szczególności jak w tabeli jest 500 tyś. rekordów tak może być 500 tyś. różnych towarów.
- A każdy z takich towarów może mieć zapisaną taką samą wartość w kolumnie ilosc ...; nie da się więc połączyć kroków 3. i 4.
- Nie mamy informacji o sposobie składowania na dysku danych typu NUMBER i VARCHAR, więc ocena, o ile zmniejszy się rozmiar wiersza gdy pominiemy kolumnę nr nie jest możliwa (w szczególności nikt nie gwarantuje, czy w kolumnie nr nie ma samych NULL, które w dodatku są reprezentowane bez zajmowania miejsca na dysku).
- Rozwiązanie oparte o hash, a nie sortowanie też jest możliwe, w podobnym koszcie, ale algorytm - m.in. z wyżej wymienionych powodów - jest dużo bardziej skomplikowany, trzeba też sporo założyć na temat funkcji mieszającej.

Zadanie 3 (KC)

Dany jest następujacy schemat tabeli reprezentującej odcinki dwukierunkowych linii kolejowych (w tabeli odcinki nie powtarzają się):

```
CREATE TABLE odcinki (
  stacjaA       VARCHAR(15) NOT NULL,
  stacjaB       VARCHAR(15) NOT NULL,
  dlugosc       INTEGER
);
```

Napisz w algebrze relacji zapytanie o to, czy z Warszawy można dojechać do Gdyni w co najwyżej dwóch przesiadkach.