



POZNAN UNIVERSITY OF TECHNOLOGY

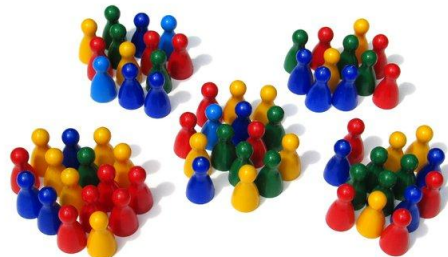
Data Warehouse Physical Design: Part III

Robert Wrembel
Poznan University of Technology
Institute of Computing Science
Robert.Wrembel@cs.put.poznan.pl
www.cs.put.poznan.pl/rwrembel



Lecture outline

➔ **Group By optimization**

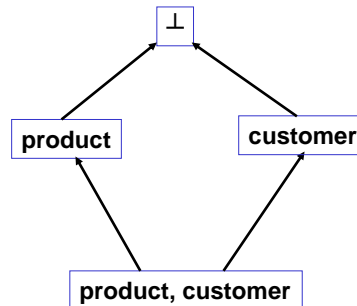




Small cuboid - 2D

⇒ GROUP BY CUBE (product, customer)

- GROUP BY product, customer
- GROUP BY product
- GROUP BY customer
- GROUP BY ()

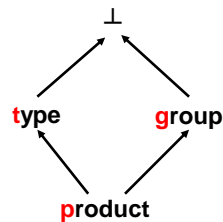


Dimension hierarchy

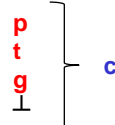
dimension Customer



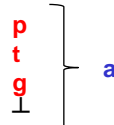
dimension Product



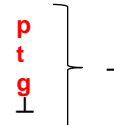
Product Customer



Product Customer

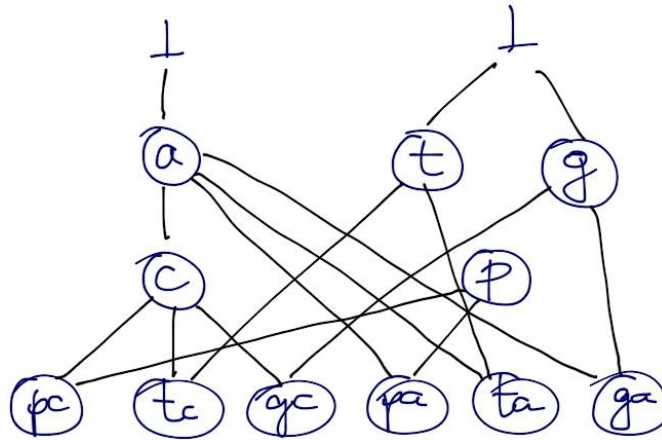


Product Customer





Dimension hierarchy

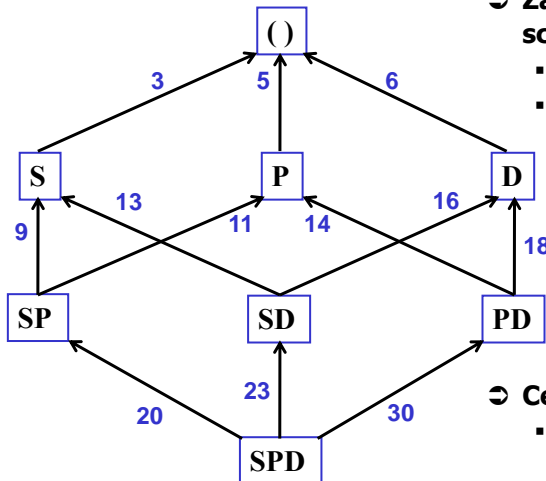


3D rollup

- ⇒ **GROUP BY ROLLUP (sklep, produkt, data)**
- **GROUP BY sklep, produkt, data**
 - **GROUP BY sklep, produkt**
 - **GROUP BY sklep**
 - **GROUP BY ()**



Optymalizacja GROUP BY



- ⇒ Założenia: dla każdego schematu grupowania znana
 - liczba unikalnych wartości
 - koszt wyliczenia na podstawie węzła podrzędnego

- ⇒ Cel optymalizacji
 - wyznaczyć wszystkie schematy grupowania w sposób minimalizujący sumę kosztów



Techniki optymalizacji

- ⇒ Najmniejszy węzeł rodzica (smallest parent)
 - obliczenie grupowania wyższego poziomu na podstawie najmniejszego (rozmiarowo) grupowania niższego poziomu
- ⇒ Buforowanie wyników w RAM (cache results)
 - wyniki grupowania są wykorzystane do następnego grupowania
 - group by SPD -> group by SP -> group by S
- ⇒ Ograniczenie odczytów z dysku (amortize scans)
 - w czasie jednego odczytu z dysku wyliczanych jest wiele schematów grupowania
 - np. jeśli wynik grupowania SPD jest składowany na dysku to wylicza się SP, SD, PD



Techniki optymalizacji

⇒ Współdzielenie wyników sortowania (share sorts)

- jeden schemat sortowania jest wykorzystany do wyliczenia wielu schematów grupowania

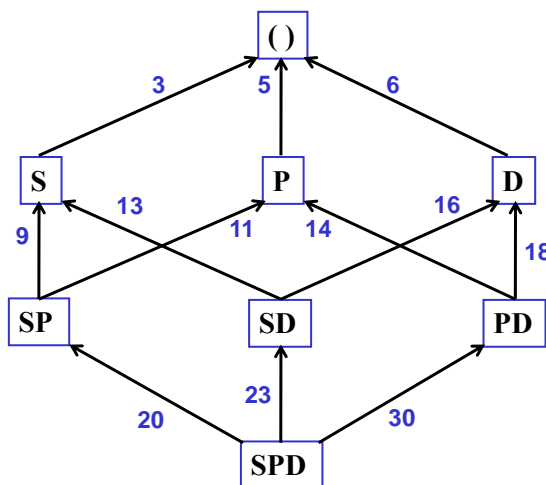
⇒ Podział dużej tabeli na części (partycje)

- zastosowanie haszowania do każdej części
- skalanie części (możliwe dla funkcji addytywnych)
- współdzielenie części dla wielu schematów grupowania

- ⇒ Agarwal S., Agrawal R., Deshpande M. P., Gupta A., Naughton F. J., Ramakrishnan R., Sarawagi S.: On the Computation of Multidimensional Aggregates. VLDB, 1996
- ⇒ Ross K. A., Srivastava D.: Fast Computation of Sparse Datacubes. VLDB, 1997
- ⇒ Beyer K., Ramakrishnan R.: Bottom-Up Computation of Sparse and Iceberg Cubes. SIGMOD, 1999
- ⇒ Wang W., Feng J., Lu H., Xu Yu J.: Condensed Cube: An Effective Approach to Reducing Data Cube Size. ICDE, 2002
- ⇒ Lakshmanan L., Pei J., Han J.: Quotient cube: How to summarize the semantics of a data cube. VLDB, 2002
- ⇒ Chen Z., Narasayya V.: Efficient Computation of Multiple Group By Queries. SIGMOD, 2005
- ⇒ Morfonios K., Ioannidis Y.: CURE for Cubes: Cubing Using a ROLAP Engine. VLDB, 2006



Algorytm PipeSort

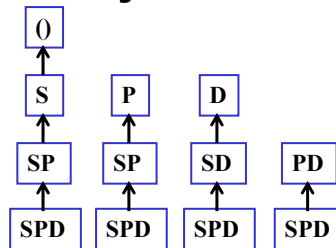


⇒ Przeszukiwanie grafu ↓

- ⇒ Dla każdego poziomu znajdowana jest krawędź o najmniejszym koszcie

⇒ Wykonywanie grupowania ↑

- ⇒ Buforowanie wyników grupowania z poziomu niższego





Algorytm PipeHash

- ⇒ Koncepcja jak PipeSort
- ⇒ Założenie: znane jest oszacowanie rozmiaru wierzchołka grafu (wyniku schematu grupowania)
- ⇒ Zamiast sortowania wykorzystywane haszowanie

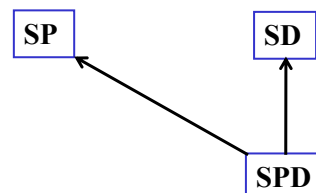


Algorytm Overlap

- ⇒ Koncepcja jak PipeSort
- ⇒ Założenie: znane jest oszacowanie rozmiaru wierzchołka grafu (wyniku schematu grupowania)
- ⇒ Przeglądanie grafu w szerz na każdym poziomie
- ⇒ Wynik grupowania jest dzielony na części

Żabka1	łaciate 2%	3.50
Żabka1	kajzerka	7.70

Żabka2	Polityka	9.00
Żabka2	łaciate 2%	3.50
Żabka2	kajzerka	7.70
Żabka2	Polityka	9.00





Algorytmy Memory-Cube i Partitioned-Cube

- Memory-Cube zakłada, że dane zmieszczą się w RAM i sortowanie będzie możliwe w RAM
- Partitioned-Cube zakłada, że dane nie zmieszczą się w RAM
 - podział danych na grupy ze względu na atrybuty grupujące, np. sklep, produkt, data
 - dla każdej unikalnej wartości atrybutu wiodącego, np. sklep in {Żabka1, Żabka2, ...} w agregacie na poziomie k
 - oblicz agregaty na poziomach k+i za pomocą Memory-Cube



Algorytmy optymalizacji

Algorytm	Przeglądanie przestrzeni grupowań	Kolejność wykonywania grupowań	Metoda (sortowanie/haszowanie)	Obsługiwane operatory	Hierarchie
PipeSort	↓	↕	Sortowanie	ROLLUP CUBE	TAK
PipeHash	↓	↕	Haszowanie	ROLLUP CUBE	TAK
Overlap	↓	↔	Sortowanie	ROLLUP CUBE	
Partitioned-Cube	↓	↕	Sortowanie	CUBE	
Memory-Cube	↓	↕	Sortowanie	CUBE	
BUC	↑	↕	Sortowanie	CUBE	
BU-BST	↑	↕	Sortowanie	CUBE	
QC-Tables	↑		Sortowanie	CUBE	NIE
Chen - Narasayya	↑	↕		GROUPING SETS	
CURE	↑	↕	Sortowanie Haszowanie	ROLLUP CUBE	TAK