Data Cube

Implementing Data Cubes Efficiently,
Venky Harinarayan, Anand Rajaraman
Jefrey D. Ullman, SIGMOD-96

Data Cube

- Is a relational aggregation operator
- Generalizes, group by's, cross tabs, and sub-totals
- Can be applied on any relational database schema
- More pertinent to Data Warehousing and OLAP

Data Cube from Fact Table

- Given n dimensions each dimension with l_i distinct values, the maximum size of the fact table is:

Data Cube

- How many cuboids are there in a data cube with *n* dimensions?
- 1 base cuboid, 1 apex cuboid, and others…

Size of Data Cube

- Consider a star schema with *n* dimension tables and a fact table with a measure attribute
- Let each dimension have l_i distinct values, then the size of the data cube is:

Data Cube – computation issues

- From a fact table – what is the best way to compute a data cube?
- Naïve method ---
- What is storage overhead? How many disk block accesses are needed?

Data Cube – better algorithms

Issues:

- Need to store the base cuboid – no benefit there.
- What about other cuboids? Can some efficiency be achieved?
- How do we develop a formalism for this?

Formalism

- Notion of a query result being used to generate another query result.
- What are the advantages of this idea?
- How can this idea be adapted for data cube computation?

Lattice structure

- A data cube forms a lattice structure.
- Therefore, lower level cubes can be used to compute higher level cubes.
- What is the cost advantage?
- The idea is to select a set of cuboids to materialize
- The algorithm is a greedy algorithm.

More on Lattice structure

- Let $<L, \leq>$ be a lattice of queries.
- To answer a query Q, we select an ancestor of Q, say Q_A that has been materialized.
- The cost of answering Q depends on the number of tuples in Q_A.

The problem is to select a set of materialized views to minimize the total cost of processing all the queries

Cost Model

- Given $<L, \leq>$, we need to calculate the cost of processing a query is calculated as follows:
- Let C(v) be the cost of view v, (no. of tuples in v)

Cost Model

- Let S be a set of all views materialized thus far:
- Benefit of view v relative to S, which we denote as B(v,S), is defined as follows:
- For each $w \leq v$, define B_w by
  - Let u be the view of least cost in S, such that $w \leq u$. There is at least one such u
  - If $C(v) < C(u)$, then $B_w = C(u) - C(v)$. Otherwise, $Bw = 0$
- $B(v,S) = \Sigma_{w \leq v} B_w$

HRU Greedy Algorithm

```
S = {top view}
for i = 1 to k do begin
  Select that view v not in S such
  that B(v,S) is maximized;
  S = S union {v}
end;

Resulting S is the greedy
  selection;
```


HRU Greedy Algorithm - analysis

HRU greedy algorithm always assures at least 63% of the optimal benefit.