Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals Review

Neutral Description :

The paper talks about data cube and different operations(aggregate functions) over it including cube operator. The cube operator being the powerset of all the possible group bys on the data , the paper explains the representation of different possible group by in 1-3 dimensions. It finally generalizes this concept to a cube operator which contains 2^N different cubing operations and in each cubing operation, the columns with which group by is not considered (say *i* out of N values [these are the values over which the aggregates are computed]) get the cell value ALL. The paper further mentions Roll-up which is the decreasing order of the number of columns with which the group bys has been taken, i.e as the number of columns with which the group by occurs decreases, the number of aggregate values also decrease. Whereas drill down is the corresponding opposite operation.

The paper moves on to display the sql queries for the cubing operation which would have 2^N different queries UNIONed together, but rather uses the keyword 'CUBE' and the names of different columns with which the cube is to be done.

Cubing is further divided programmatically as 3 different pseudo code operations i.e.

- Start : allocates handle in the main memory for the cubing operation.
- Next : gets the handle returned by start , gets the exact value to be added for the desired aggregation and does the minimal aggregation.
- End : again returns the handle with the generated cubed values.

For every row of data there are 2^N Next operations.

In a quest to optimize many steps in the above proposed, the paper mentions to :

- keep the ALL values while CUBEing as NULL values.

- Minimize data movement and compute aggregates at the lowest system level.
- Organize aggregations in memory by using hashing.
- Strings can be converted to hashed integers if possible.
- If number of aggregates is too large then sorting or hybrid hashing can be used to organize data for aggregating.
- The CUBEing time can be decreased exponentially as the independent group bys can easily be recognised and executed parallely.

Where the authors come from ?
Two different organizations (Microsoft and IBM) from which 8 researchers have gathered to come up with this paper in order to apply/compute CUBE operator over the data stored in the data warehouse. These are also called the OLAP operations covered over the data in those companies.

What Authors have done ?
Authors have given not only an overview but a full imagination of a data cube and the CUBing operator. They have considered different difficult scenarios – how to compute, store in an efficient manner the CUBE over different schema of the data warehouse. Apart from the five main aggregate operations the ones which are not handled are also mentions like the *rank, n_tile, cumulative and percent.*

Why they think CUBING is important ?
As the warehouse contains the history it has HUGE amount of data and there are often myriad of group by queries running over different possible views of its schema. Hence the CUBE operator will find all the possible group bys and store as just different pointers whenever asked by different executing queries.

Contributions made by them that they missed to mention ?
CUBING can have great applications when applied to not only text data which the authors have mentioned e.g. image and sound data etc. And the very data which is generated by CUBING can be used as **ensembles** to aid massive events and its predictions like weather forecasting.

Final Critique : The paper covers a very important concept and is written and very well explained manner. It has a novelty approach to bring a very vital operation of a data warehouse. There are no gaps or technical issues and an explainatory diagram has been given where ever needed. Using Multi arrays this computation can be done in a very efficient manner which brought in the concept of MOLAP.