MAS DSE 201 Homework: Sales Cube

Milestone I [due Jan 28 at midnight]

Consider a database that captures customers with a name and state (of residence); states with a name; products that have a list price, a name and belong to a category; categories have names and descriptions; sales of a product to a customer capturing quantity and price paid (may be discounted on a case-by-case basis).

Produce an SQL schema that captures the above information.

Milestone II [due Feb 18 midnight]

Write the following queries:

1. Show the total sales (quantity sold and dollar value) for each customer.
2. Show the total sales for each state.
3. Show the total sales for each product, for a given customer. Only products that were actually bought by the given customer are listed. Order by dollar value.
4. Show the total sales for each product and customer. Order by dollar value.
5. Show the total sales for each product category and state.
6. For each one of the top 20 product categories and top 20 customers, return a tuple (top product, top customer, quantity sold, dollar value)

Milestone III [due March 4 midnight]

Your next goal is to deliver top performance for the queries. Expect a significant amount of experimentation until you tune the performance.

In particular, create any indices that will be beneficially used by the queries. Do not create useless indices. Deploy your solution and measure the performance of the Milestone II queries on cold runs.

Submit your index choices (in the form of the script with the CREATE INDEX statements) and the measured query performance improvement.

Milestone IV [due March 18 midnight]

Your next goal will be to improve the performance of Query 6 by appropriate precomputations. It is understood that the maintenance of the precomputed table(s) will lead to some slowdown while

viewers interact. Your precomputation choices must be such that the precomputations that you introduce collectively save more time to this option than they cost to maintain. Precomputed tables will also benefit from creating certain indices on them. Build any beneficial indices. Calibrate your solution against cold runs of the activity script.

Submit the following.

- The precomputed tables you created: CREATE TABLE statement and query that initially loaded it.
- Your new Query 6, which makes the best use of the precomputed tables you chose.
- The indices you created (CREATE INDEX scripts)