



http://www.db2mag.com/db_area/archives/2003/q2/idug.shtml

Better BI Performance

By David Beulke

[Quarter 2, 2003](#)

 [Printer-Friendly Version](#)  [Email this Story](#)  [Bookmark to del.icio.us](#)  [Digg It!](#)

DB2 UDB version 8.1 for Linux, Unix, and Windows provides many built-in business intelligence (BI) features, including materialized query tables (often called materialized views and formerly called automatic summary tables), `CUBE` , `ROLLUP` , grouping capabilities, `RANK` , `DENSE_RANK` , `ROW_NUMBER` , and various aggregation functions.

Built-in DB2 features are much more efficient than any online analytic processing tool or application routine, and they help improve network throughput by reducing the number of rows returned. I've found that materialized views are one of the most useful DB2 features for improving performance in BI systems.

If you want to address data warehouse performance, you must understand data warehouse application flow and SQL activity. Start by gathering specific SQL statements, their `WHERE` clause criteria, and variable values. Then relate that information to the number of times the statements are executed and to summary I/O performance. Capture this information for the problem time period through a trace or software tool.

Focus on executed SQL that uses `GROUP BY` , `ORDER BY` , or various exotic looking SQL `SELECT` clauses. Analyze these SQL statements to determine whether common `WHERE` clause criteria exists. If it does, analyze the criteria to find out if a new materialized view would boost performance.

A performance analysis I did for one client showed that many SQL queries were using a time range and time period `WHERE` clause against the fact table. Analyzing these queries revealed processes that were summarizing information based on time period and accounting period durations. My team created different materialized views with different time period summarizations (based on weekly, monthly, and quarterly accounting periods and other specialized criteria). These materialized views were then integrated into the overall data warehouse environment with regular processing, `RUNSTATS` , and other maintenance activities.

System performance improved because these materialized views rolled up the information once for everyone to reference instead of every time a user asked for it. Users got information in seconds instead of hours.

But creating the views isn't the end of the task. You also need to advertise the materialized views within the user community through demonstrations, emails, and management meetings.

Make sure the new materialized views are integrated into all tools that access the detailed fact table. Front-end tools should be populated with accurate descriptions and statistics of the many materialized view summarization levels available

Materialized views greatly reduce the number of I/Os necessary to get answers to common questions. If you only have a detailed data warehouse fact table, getting sales totals by month requires scanning the whole table and grouping by month every time the information is requested. With materialized views, the scan is done once and captured for everyone to use.

You can learn about these features from fellow users at IDUG (International DB2 Users Group) conferences. Check www.idug.org for conference details.

David Beulke is an independent consultant, president-elect of IDUG, and an IBM Gold Consultant. You can reach him at dbeulke@compuserve.com.

Comments? Questions?

Give us your feedback or ask a question of the author.

Please enter your e-mail address below:
