JACKIE's Forum

Materialized Query Tables (MQTs)

Jackie Jansen

Many of you are probably asking, "Just what are Materialized Query Tables or MQTs and why should I care?" Well long term you will care and short term anyone with a data warehouse or data mart should care. Materialized Query Tables or MQTs are also known as automatic summary tables. In past columns I have discussed creating summary tables for users that initiate long running queries. The theory is, if you take the time to create a table that at least partially summarizes or aggregates the information the user is looking for, when they query this much smaller table their results will be correspondingly faster.

Let me give you a simple example. Suppose that you are a retailer and have users that frequently run queries asking for daily or weekly sales by store. Retail companies often have a very large number of sales transactions. These queries may take a long time to process. Assume you created a summary table nightly with one record for every combination of store and date for the past year that contained sales totals. If you had 100 stores you would have 36,500 records (365 * 100). Referencing this small summary table would result in the users' queries executing very quickly whether they were asking for daily or weekly sales.

Now if the users also wanted daily or weekly sales by product line, you would need to create another summary table with grouping by product line instead of store. The problem with this approach is that the users need to know which table to query. Do they need to reference the underlying detail table or can they use the product summary or the daily summary tables. MQTs solve this problem. You still need to create your summary table but now you tell DB2 that it is an MQT. While users can query this table directly the benefits actually come from the fact that they don't need to. When you execute a query the DB2 optimizer will now check to see if an MQT exists that will shorten the overall run time of the query. If one exists and will help, the optimizer will automatically use it. If an MQT that will help doesn't exist then the query optimizer will simply process the detail table normally. The users don't need to be told that any summary tables exist. They simply query the detail table and the system will find the appropriate MQT for them.

Notice that the MQT doesn't need to be an exact match. In our previous example, if we created an MQT for daily sales and requested a weekly aggregate DB2 would simply sum seven daily records to get each weekly total. This would be much faster than reading and summing thousands or more records for each week.

I started by saying that MQTs were of much more interest today to customers that have a data mart or a data warehouse. You might ask why since we often have long running queries against operational data. First remember that this is the iSeries initial release of MQTs. Currently they are not automatically synchronized with the base table they are created from. If during the day you add records to your base table, this data will not be reflected in your MQT. If you used MQTs in this environment your query might have different results depending on whether the optimizer chooses to use the base table or the MQT. This isn't really any different than when you created summary tables in the past but since the user is now always referencing the base detailed table it could be confusing.

If your company has a nightly job that moves some of your data into a separate database for query purposes, you can add MQT synchronization to this job. You can automatically refresh or recreate your MQTs that are built over this query/information database.

Both iSeries Navigator and the CREATE TABLE statement support the parameters necessary to configure a Materialized Query Table. For example:

```
CREATE TABLE DailyMQT AS
(SELECT   Branch,   salesdate,
SUM(Revenue) as Total_Revenue,
SUM(Units) as Total_Units,
FROM SalesDetails
GROUP BY Branch, salesdate)
DATA INITIALLY IMMEDIATE
REFRESH DEFERRED
MAINTAINED BY USER;
```

In the above example the user would direct their queries to the SalesDetails table. The database monitor and Visual Explain will show you which table SalesDetail, or DailyMQT the optimizer chose to use. There is a REFRESH TABLE command that can be used to refresh or synchronize the MQT with its underlying table.

For those of you that have not yet moved to V5R3 here is one more reason to move over. This support is delivered with the database group 5 PTFs SF99503 for V5R3.

For more information see the DB2 UDB for iSeries home page www.ibm.com/iseries/db2. Have a look at "What's New" under "Key Benefits". Here you can find additional details and access to a whitepaper by **Mike Cain** on the creation and use of MQTs. TUG

***Jackie Jansen** is a Senior Consulting IT Specialist. She currently works in the IBM Americas Advanced Technical Support Solutions Centre. Jackie is a frequent speaker at iSeries Technical Conferences and User Group meetings.* Contact her at jjansen@ca.ibm.com.