THE BUFFER POOL

DB2 TIPS, THOUGHTS & RUMORS

The DB2 grapevine has been hot and heavy these past few months. As most of you know, the new version of DB2 is generally available.

Before proceeding, however, review The Buffer Pool's last trivia question: In the EXPLAIN process, recall how a tablespace scan is indicated in the PLAN TABLE. The value of the ACCESS-TYPE column is set to the character R. Why? No R appears in the words tablespace scan. What does the R indicate? Answer: The R in the PLAN_TABLE column, ACCESSTYPE, stands for Relational. When DB2 was just a glimmer in IBM's eye, a tablespace scan was referred to as a relational scan; hence, the use of R to indicate a scan.

Next Version Or Release

You know the new DB2 from IBM is officially called Version 3. But what is expected after that? Rumor says that Version 4 will follow instead of the more customary Version 3.2. But nothing official is coming out of Santa Teresa, so you will have to wait and see. And what will be in that future DB2 release? Likely candidates are triggers, row-level locking, increased parallelism, outer join (well, you can hope) and the ability to update DB2/2 from DB2/MVS.

REORG That Catalog

Want to reorganize your DB2 catalog? Oh, what a task! Here is a juicy tidbit you may want to try. Rumor has it that you can use the utility program (provided by IBM to migrate from release to release of DB2) to reorganize the DB2 catalog. Now, be careful, this has not undergone any formal testing, but a reputable source at IBM/Santa Teresa indicated that it should work. Although slow, it is the only method currently available. For 2.3 and Version 3, the utility is called CATMAINT. Prior releases used a utility called DUMPCAT.

And, by the way, rumors persist that the next release of DB2 (the one after Version 3) will have a catalog reorganization feature; but until then, you will have to improvise.

Increased Rows Without Compression

Most of you already know that with Version 3 compression, pages can contain 255 rows instead of 127. But did you know that in Version 3 you can get 255 rows per tablespace page without using the new data compression features? It's true. After moving to DB2 Version 3, simply issue the following set of SQL statements:

```
ALTER TABLESPACE tsname
COMPRESS YES;
COMMIT;
ALTER TABLESPACE tsname
COMPRESS NO;
COMMIT;
```

When the tablespace is altered to indicate COMPRESS YES, DB2 will modify the DBD to indicate that 255 rows can be stored per tablespace page. But, when you alter the tablespace back to COMPRESS NO status, the DBD is not changed. Be careful, because if the database is ever dropped and re-created to perform maintenance, you will need to reissue the previously mentioned sequence of statements for all tablespaces that must have 255 rows per page.

Beware The Buffers

More Version 3 news: Instead of four buffer pools, DB2 Version 3 will offer 60. This may be old news to you, but a possible "gotcha" lurks within this new feature. Of the 60 buffer pools, 10 are for 32K tablespaces and the remaining 50 are for 4K tablespaces. Now, take a close look at the buffer pool names:

- 4K buffers — BP0 through BP49
- 32K buffers — BP32K through BP32K9.

Due to the buffer pool naming convention, you may end up allocating too many of the wrong size buffers if you are not careful. How? Consider BP32 vs. BP32K. BP32 is the 33rd 4K buffer pool, but BP32K is the first 32K buffer pool. Simply by adding or dropping the letter K, an improper allocation could be made.

Androgynous Software

Did you ever notice that people have a tendency to personify DB2? How many times have you heard (or maybe said), " ... he creates an access path based on catalog statistics ... " or " ... she uses the correct version based on the consistency token." What do *you* think — is DB2 male, female or androgynous? Or is it something else altogether? Send your comments to The Buffer Pool via Prodigy (WHNX44A) or CompuServe (70410,237).

Tuning Through Deferred Index Creation

The addition of the DEFER parameter to the CREATE INDEX DDL will provide several new opportunities for tuning index usage. The primary reason for the addition of this parameter is to enable DBAs to create an index and defer population until later, using the more efficient RECOVER INDEX utility. However, you may want to consider creating deferred indexes to perform access path tuning. When an index is created specifying the DEFER parameter, it is placed in recover pending status and the index data is not populated. However, the index will be considered by the optimizer during access path determination. So, you may want to use DEFER YES to create several different indexes, update the DB2 catalog to indicate production-level statistics and run EXPLAIN to see which indexes DB2 finds most useful. Though this is not the reason IBM provided deferred index support, it may prove to be a worthwhile, time-saving feature for tuning DB2 access.

Trivia Question

What is the minimum number of tables required to implement DB2 Referential Integrity (RI)?

DB2 Tips Or Stories?

If you have any interesting DB2 tips or stories, share them with your peers. Send your information to The Buffer Pool in care of *Relational Database Journal* to make sure it receives the attention it deserves. Or, contact the author on Prodigy (WHNX44A) or CompuServe (70410,237).