Running Head: INDEXING AND ROLLBACK

Oracle: Indexing and Rollback Segments

Jered McClure

Walden University

Oracle: Indexing and Rollback Segments

1. Discuss the drawbacks of indexing and the dangers of indexing too many columns in a database.

 The primary concern with having too many indexes is in storage requirements. Each index essentially doubles the required storage space for each indexed column. This in turn can cause lookups across multiple indexes to slow down as Oracle blocks become full index leaves (BTree) become wide spread. However, in the instance that storage is not an issue (that is, space can be allocated dynamically and expanded infinitely); too many indexes can still cause issues.

 Database tables which require many changes; updates, inserts, and deletes, can occur performance drops based on the number of indexes required to be modified. In other words, every additional index on a table requires any changes to the indexed columns to be made twice, once on the primary table in question, and once in the index. Additionally, if a column is indexed in multiple formats (e.g. ascending, descending, paired with one or multiple other columns) each of those formats must also be modified on any change to their data. In a transactional system, this would be detrimental to timely performance, negating any normalization which has occurred.

2. What would cause the rollback segments to run out of space? How can that situation be avoided?

 The rollback segment method of managing undo space is the manual method of managing undo data in an Oracle database. Oracle does not recommend the use of manual segments as they are more complex and tend to require constant management from a DBA. As such it is better to use an undo TABLESPACE. An undo TABLESPACE operates under the same conditions as a normal TABLESPACE, but stores information specifically related to rollback.

In essence, the undo TABLESPACE can be set to MAXSIZE UNLIMITED. This will allow the TABLESPACE to grow as required, thereby avoiding any manual space adjustments a rollback segment would normally demand. However, in the instance a large DML statement is made which requires rollback information to be stored for long periods of time, this TABLESPACE can grow to unmanageable size. This is especially important to consider in active transactional environments where multiple connections can be making DML statements at the same time on the same table source.

In these situations, it is best to run massive DML statements after hours, or to break up the statement so that it runs a partial set of changes over a prolonged period of time. While the latter may be feasible, it is still better from a performance standpoint to run these queries after hours when few or no connections are likely to be making changes. This will increase DML performance and ensure that one statement is not walking on the digital toes of another.

Reference

Oracle. (2006). *10 Managing the Undo Tablespace*. Retrieved June 22, 2013, from Oracle Database Administrator's Guide 10g Release 2 (10.2): http://docs.oracle.com/cd/B19306\_01/server.102/b14231/undo.htm

Poweel, G., & McCullough-Dieter, C. (2010). *Oracle 10G: database Administrator: Implementation & Administration.* Boston: Cengage Learning.