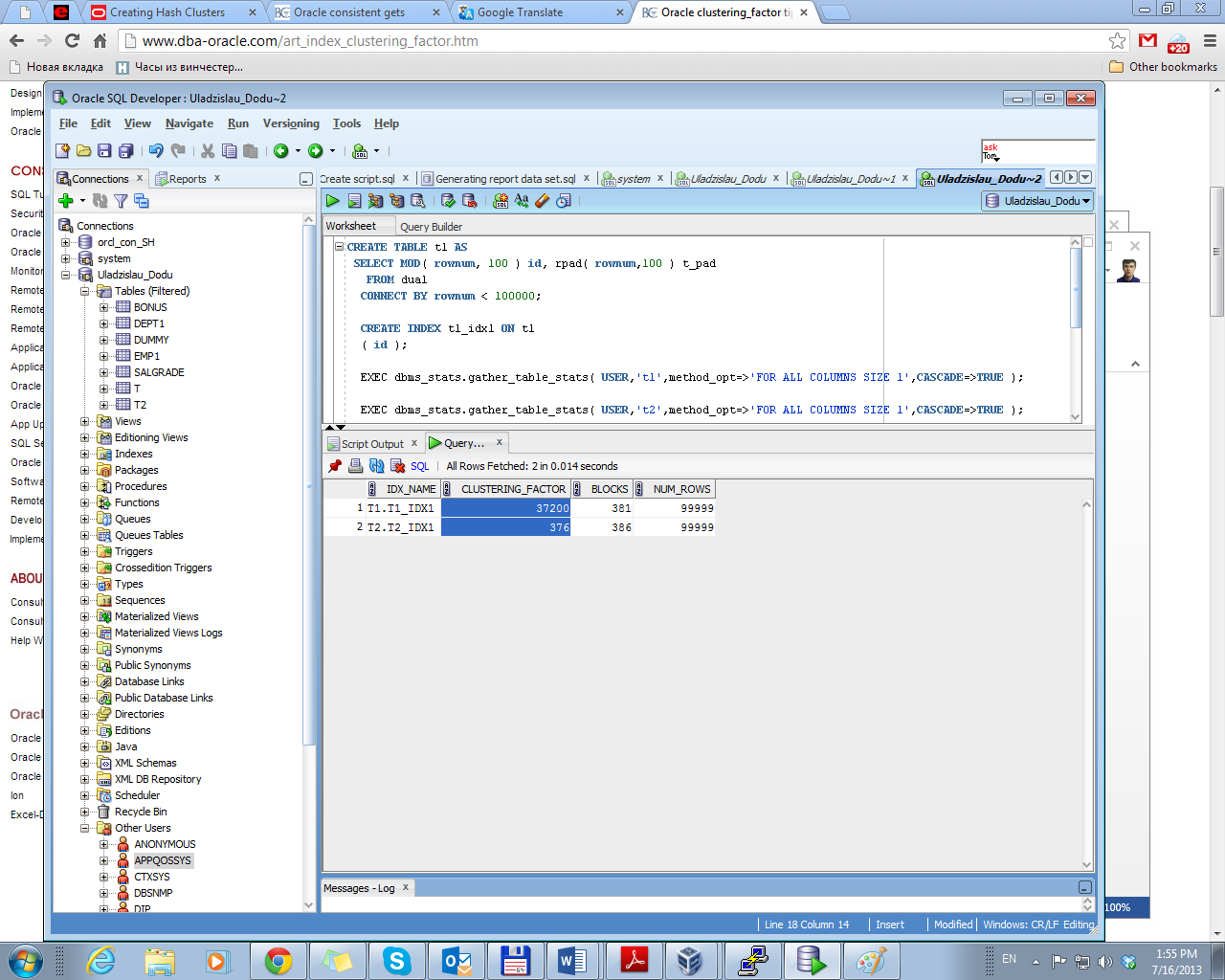
1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Count of Blocks | Count of Used Blocks | Count of Rows | Consistent gets | Description |
| 1 | 416 | 376 | 99999 | 461  (physical reads 376) | As we see 416 blocks are reserved and 376 are used. We need to retrieve data from HDD that why we see physical reeds here. |
| 2 | 416 | 0 | 0 | 381 | When we delete, amount of reserved blocks are same as #1. It is specific for delete statement. It just mark that some blocks are empty (delete some rows), but not reduce amount When we select data from t2 we are observe all blocks (416-381) |
| 3 | 416 | 1 | 1 | 381 | Single row we insert into block, but amount of all blocks are still the same |
| 4 | 6 | 0 | 0 | 5 | TRUNC command frees all data blocks. It leave only some blocks with system information(I think so) |

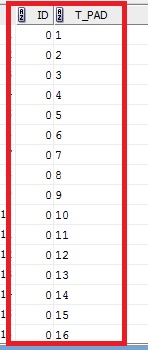
2.

2.1

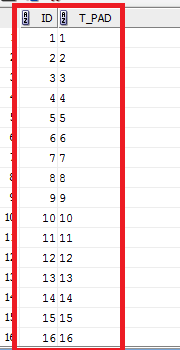


The clustering\_factormeasures how synchronized an index is with the data in a table.  A table with a high clustering factor is out-of-sequence with the rows and large index range scans will consume lots of I/O.  Conversely, an index with a low clustering\_factoris closely aligned with the table and related rows reside together of each data block, making indexes very desirable for optimal access.

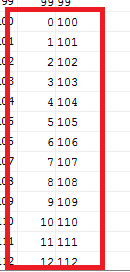
We have different values of clustering factor because we have different datasets for each table (trunc and mod functions). Data with the same id from t2 is distributed in the nearest blocks



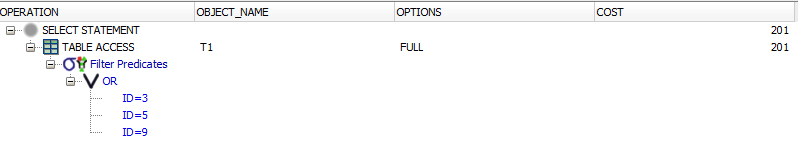
Data from t1 is randomly distributed in blocks

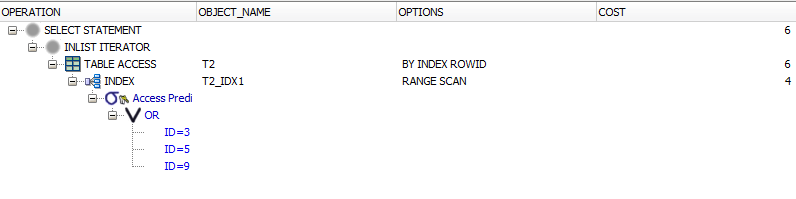


……………………………..



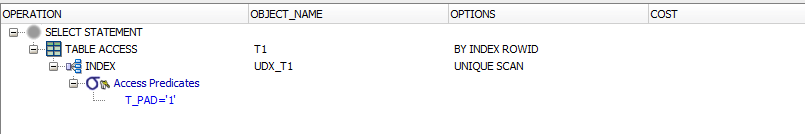
Best selective performance in executive has table (t2) which has cluster factor = amount of block, than (t1) cluster factor = amount of rows

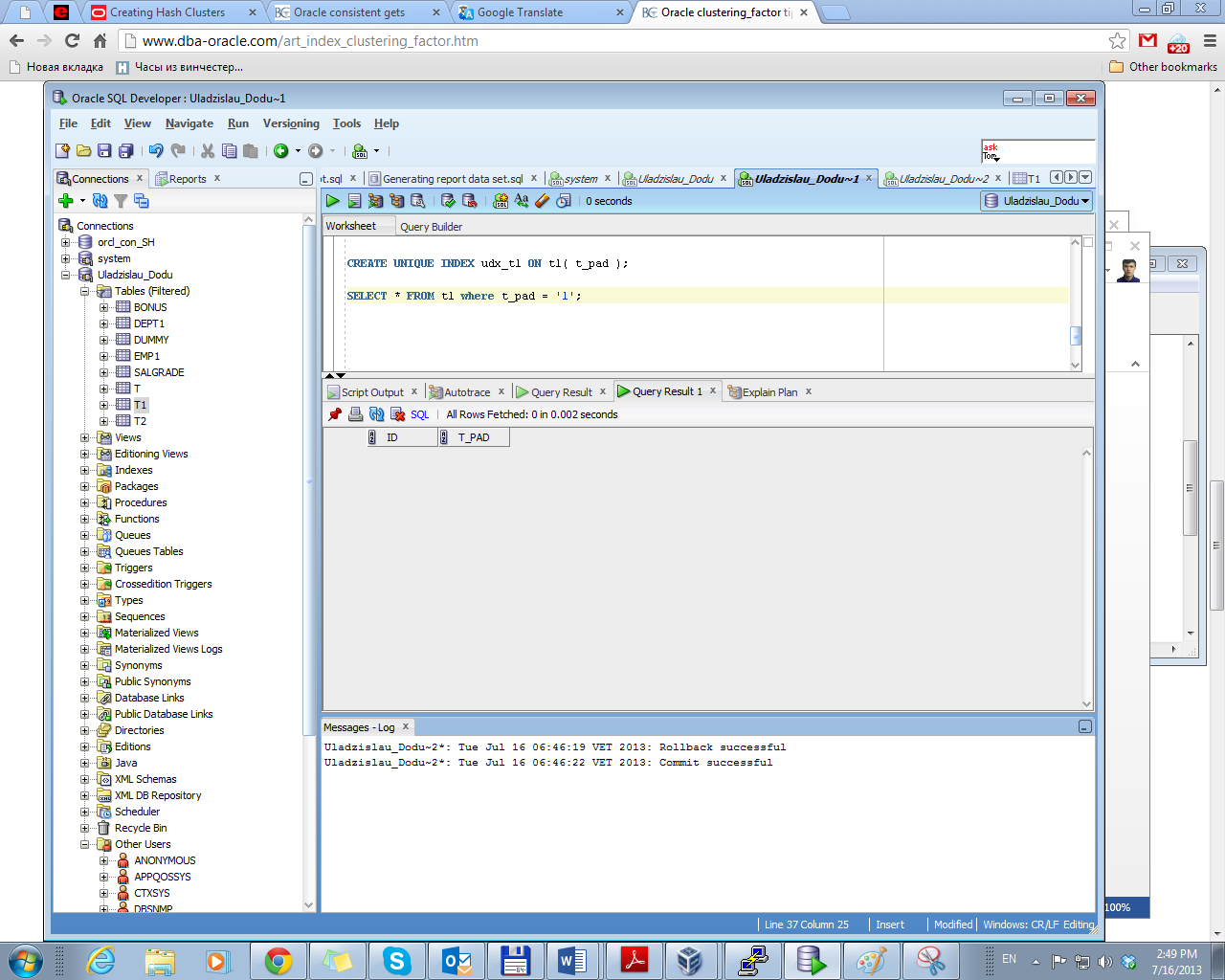




Because we can use range index scan (index distribution).

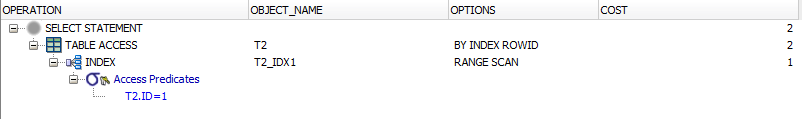
2.2

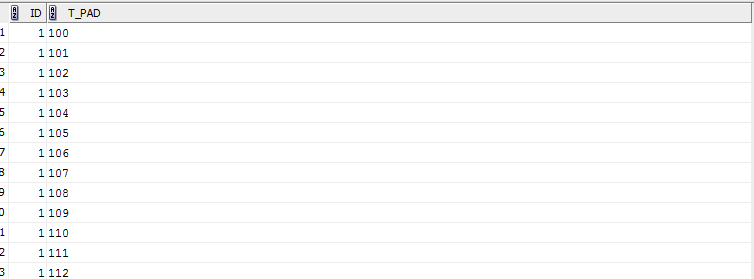




An index unique scan stops processing as soon as it finds the first record because no second record is possible.

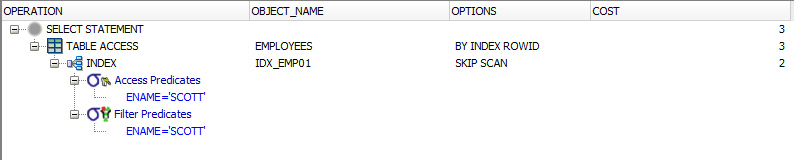
2.3.

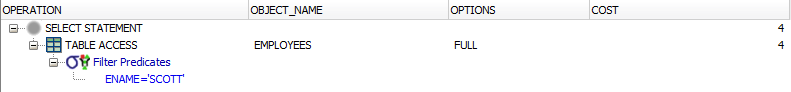




The index range scan proceeds horizontally through the linked list of leaf nodes until it locates a value greater than 1. It works because index’s leafs are linked.

2.4.





Skip scan are initiated by probing the index for distinct values of the prefix column. Each of these distinct values is then used as a starting point for a regular index search. The result is several separate searches of a single index that, when combined, eliminate the affect of the prefix column .

Columns in a composite index should appear in the order that makes the most sense for the queries that will retrieve data and need not be adjacent in the table.

Composite indexes can speed retrieval of data for SELECT statements in which the WHERE clause references all or the leading portion of the columns in the composite index. Therefore, the order of the columns used in the definition is important. In general, the most commonly accessed columns go first.