# Lab 9- using the optimizer and explain plan



Create the BIG and BIG2 tables and indexes using the script big1.sql and big2.sql

Scripts. Given a non-unique index BIG\_BIGNO on the BIGNO column of BIG, and a non-unique index BIG\_BNAME on the BNAME column of the BIG table, use EXPLAIN PLAN to determine how ORACLE would process the following utilizing ALL\_ROWS and FIRST\_ROWS.

EXPLAIN PLAN FOR

1. ORACLE> Select \* from big where bigno > 1;

SELECT \* FROM TABLE(dbms\_xplan.display

HINT: use AUTOTRACE to help provide the information.

ORACLE> delete from plan\_table;

2. ORACLE> explain plan for

Select \* from BIG where bname like ‘BET%’;

SQL> select \* from table (dbms\_xplan.display)

3. Using the Optimization type of FIRST\_ROWS then ALL\_ROWS do:

ORACLE> delete from plan\_table;

ORACLE> explain plan for Select MAX(BIGNO) from big;

SELECT \* FROM TABLE(dbms\_xplan.display

4. Using the Optimization type of ALL\_ROWS then FIRST\_ROWS do:

ORACLE> delete from plan\_table;

ORACLE> explain plan for

Select \* from big where bigno IN (500, 99500);

SELECT \* FROM TABLE(dbms\_xplan.display

# Lab 9- continued

5. Using the Optimization type of ALL\_ROWS and FIRST\_ROWS\_1000 do:

ORACLE> delete from plan\_table;

ORACLE> SELECT \* FROM BIG WHERE BIGNO = 1000 AND

BNAME = ‘ONE THOUSAND’;

Given non-unique indexes on all columns of the BIG2 table plus a

concatenated non-unique index on the BNAME and BIGNO columns

of the BIG2 table, use EXPLAIN PLAN to determine how ORACLE10G would

execute the following queries:

6. Using the Optimization type of ALL\_ROWS and FIRST\_ROWS do:

ORACLE> delete from plan\_table;

ORACLE> Select bigno from big2 where bname = ‘ONE THOUSAND’;

7. Using the Optimization type of ALL\_ROWS do:

ORACLE> delete from plan\_table;

ORACLE> select bigno, bname from big2

where lower(BNAME) = ‘one thousand’;

8. Using the Optimization type of ALL\_ROWS do:

ORACLE> delete from plan\_table;

ORACLE> select bigno, bname

from big2

where bname = ‘ ONE THOUSAND’ and bigno = 1;

9. Using the Optimization type of ALL\_ROWS and FIRST\_ROWS do:

ORACLE> delete from plan\_table;

ORACLE> select big.bigno, big2.bname

from big, big2

where big.bigno = ABS(big2.bigno + 0);

( THE DRIVING TABLE IS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

10. Using the Optimization type of ALL\_ROWS do:

ORACLE> delete from plan\_table;

ORACLE> select big.bigno, big2.bnam

from big, big2

where ABS(big.bigno) + 0 = big2.bigno;

(THE DRIVING TABLE IS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

11. The following exercise will have you alter you session setting the optimizer

to the type of ALL\_ROWS and than perform a series of EXPLAIN PLANs

for the same SQL statement except for the value being searched. The main

purpose of this is to demonstrate when it is more efficient for ORACLE to

perform a full table scan versus using an Index on the BIGNO column.

For each of these queries identify the

Cost Table Access %Rows Returned

1. ORACLE> Use the ALL\_ROWS Hint;

ORACLE> delete from plan\_table;

2. ORACLE> SELECT \* FROM BIG

WHERE BIGNO > 2999999;

3. ORACLE> SELECT \* FROM BIG

WHERE BIGNO > 2995000;

ORACLE> delete from plan\_table;

4. ORACLE> SELECT \* FROM BIG

WHERE BIGNO > 2980000;

ORACLE> delete from plan\_table;

5. ORACLE> SELECT \* FROM BIG

WHERE BIGNO > 2971000;

ORACLE> delete from plan\_table;

6. ORACLE> SELECT \* FROM BIG

WHERE BIGNO > 2960161;

ORACLE> delete from plan\_table;

7. ORACLE> SELECT \* FROM BIG

WHERE BIGNO > ?????;

(Where ????? identifies the point at which ORACLE

uses a FULL TABLE SCAN)