**Legal Notice:** This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of Datamola.

U1M4.LW.Access and Join Methods Part 1

#### Kiryl Bucha

Contents

[1. Table access full scan 3](#_Toc43824003)

[1.1. Task 1: Full Scans and the High-water Mark and Block reading 3](#_Toc43824004)

[2. Index Scan types 5](#_Toc43824005)

[2.1. Task 2: Index Clustering factor parameter 5](#_Toc43824006)

[2.2. Task 3: Index Unique Scan 6](#_Toc43824007)

[2.3. Task 4: Index Range Scan 6](#_Toc43824008)

[2.4. Task 5: Index Skip Scan 7](#_Toc43824009)

# 1. Table access full scan

## 1.1. Task 1: Full Scans and the High-water Mark and Block reading

Step 1:

# CREATE TABLE t2 AS

SELECT TRUNC( rownum / 100 ) id, rpad( rownum,100 ) t\_pad

FROM dual

CONNECT BY rownum < 100000;

Step 2:

# CREATE INDEX t2\_idx1 ON t2

( id );

**Step 3:**

Block count:

*#* *select blocks from user\_segments where segment\_name = 'T2';*

Used Block Count:

# select count(distinct (dbms\_rowid.rowid\_block\_number(rowid))) block\_ct from t2 ;

Explain Plan:

*# SET autotrace ON;*

*# SELECT COUNT( \* )*

*FROM t2 ;*

Statistics

----------------------------------------------------------

0 recursive calls

0 db block gets

1541 consistent gets

0 physical reads

0 redo size

528 bytes sent via SQL\*Net to client

519 bytes received via SQL\*Net from client

2 SQL\*Net roundtrips to/from client

0 sorts (memory)

0 sorts (disk)

rows processed

**NOTE:** If you received next error: Check PLUSTRACE role is enabled. Please make next steps:

1. Run next script connected as sysdba:

# @ $ORACLE\_HOME/sqlplus/admin/plustrce.sql;

1. Grant role PLUSTRACE to $UserName$

# grant plustrace to buchak;

Step 4: Delete All Rows from table

# DELETE FROM t2;

**Step 5:** Repeat Step 3 and collect results.

Step 6: Insert 1 row

# INSERT INTO t2

( ID, T\_PAD )

VALUES

( 1,'1' );

COMMIT;

**Step 7:** Repeat Step 3 and collect results.

Step 8: Truncate Table

# TRUNCATE TABLE t2;

**Step 9:** Repeat Step 3 and collect results.

**Task Results:**

Expected:

Summary table with all result and text description of analyses this results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Count of Blocks | Count of Used Blocks | Count of Rows | Consistent gets | Description |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 2. Index Scan types

## 2.1. Task 2: Index Clustering factor parameter

Step 1: Create table t2 as on task 1 step 1-2

Step 2: Create table t1 as listed below

# CREATE TABLE t1 AS

SELECT MOD( rownum, 100 ) id, rpad( rownum,100 ) t\_pad

FROM dual

CONNECT BY rownum < 100000;

Step 3:

# CREATE INDEX t1\_idx1 ON t1

( id );

Step 4: Calculate statistic for both tables:

# EXEC dbms\_stats.gather\_table\_stats( USER,'t1',method\_opt=>'FOR ALL COLUMNS SIZE 1',CASCADE=>TRUE );

# EXEC dbms\_stats.gather\_table\_stats( USER,'t2',method\_opt=>'FOR ALL COLUMNS SIZE 1',CASCADE=>TRUE );

**Step 5:** Select Clustering Factor

# SELECT t.table\_name||'.'||i.index\_name idx\_name,

i.clustering\_factor,

t.blocks,

t.num\_rows

FROM user\_indexes i, user\_tables t

WHERE i.table\_name = t.table\_name

AND t.table\_name IN( 'T1','T2' );

**Task Results:**

Expected:

* Screenshot of the step 5;
* Description of the parameter clustering factor;
* Explanation: why for indexes *t1\_idx1* and *t2\_idx1* we have different values ;
* Which Index has best selective performance in execution Select clause filtered by IN ( , list of values, );

## 2.2. Task 3: Index Unique Scan

Step 1:

# CREATE UNIQUE INDEX udx\_t1 ON t1( t\_pad );

***Step 2***

# SELECT t1.\* FROM t1 where t1.t\_pad = '1';

**Task Results:**

Expected:

* Screenshot of the step 2;
* Description of process: How oracle read block on step 2;

## 2.3. Task 4: Index Range Scan

**Step 1:**

# SELECT t2.\* FROM t2 where t2.id = '1';

**Task Results:**

Expected:

* Screenshot of the step 1;
* Description of process: How oracle read block on step 1;

## 2.4. Task 5: Index Skip Scan

Step 1:

# CREATE TABLE employees AS

SELECT \*

FROM scott.emp;

Step 2:

# CREATE INDEX idx\_emp01 ON employees

( empno, ename, job );

**Step 3:**  Get trace and statistic of explain plan

# SELECT /\*+INDEX\_SS(emp idx\_emp01)\*/ emp.\* FROM employees emp where ename = 'SCOTT';

# SELECT /\*+FULL\*/ emp.\* FROM employees emp WHERE ename = 'SCOTT';

**Task Results:**

Expected:

* 2 Screenshots of the step 3;
* Description of process: How oracle analyses index that was created on step 2;
* Summary table with all result and text description of analyses this results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Count of Blocks | Count of Used Blocks | Count of Rows | Consistent gets | Description |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |