ORACLE 11G DATABASE STATISTICS – EXPRESSION STATISTICS

Inderpal S. Johal, Data Softech Inc.

OVERVIEW

Prior to Oracle 11g, you can't create the virtual column in the Database tables unless you use Function based indexes. There was one workaround shown by Jonathan in one of his Blog (http://jonathanlewis.wordpress.com/2006/10/31/virtual-columns-revisited/) to create virtual Columns in Oracle9 and 10g, which will help, query optimizer to get the correct cardinalities for the queries.

Oracle 11g allows you to create such virtual column without creating any additional index and collect statistics on these virtual column to improve the functioning of Query optimizer. You can use user-defined functions without any restriction. Please Note that creating Function based Index are also allowed to achieve the same functionality but it will affect the system resource when any DML Update is performed against the table.

In Nutshell what can be done on Group of columns as explained in Multicolumn statistics can also be done with the expression

HOW EXPRESSION STATISTICS WORKS

QUERY RESULTS WITH NO STATISTICS

Execute the Query to see the number of Records in the Table

```
SQL> select count(*) from indy_test where lower(country)='india';
  COUNT(*)
----------
      2189
```

Make sure that there is no Statistics available on the table INDY_TEST

```
SQL> select column_name, num_distinct, histogram
  2  from user_tab_col_statistics
  3  where table_name = 'INDY_TEST'
  4  /
no rows selected
```

Create the Explain Plan to see the Query Optimizer Estimates

```
SQL> explain plan for
  2  select *
  3  from indy_test where lower(country) = 'india';
Explained.
```

Optimizer Estimates 1852 Rows and so we will now collect the Statistics in the Next Step

```
SQL> select plan_table_output
  2  from table(dbms_xplan.display('plan_table',null,'BASIC ROWS'));

PLAN_TABLE_OUTPUT
--------------------------------------------
```

```
| Id  | Operation          | Name       | Rows  |
-----------------------------------------------
|   0 | SELECT STATEMENT   |            |  1852 |
|   1 |  TABLE ACCESS FULL| INDY_TEST  |  1852 |
-----------------------------------------------
```

7 rows selected.

QUERY RESULTS WITH OPTIMIZER STATISTICS COLLECTED

Collect the Optimizer Statistics using DBMS_STATS package

```
SQL> exec dbms_stats.gather_table_stats(null,'indy_test',method_opt => 'for all columns
size skewonly') ;
```

PL/SQL procedure successfully completed.

Check if the Optimizer Statistics is collected

```
SQL> select column_name, num_distinct, histogram
  2  from user_tab_col_statistics
  3  where table_name = 'INDY_TEST'
  4  /
COLUMN_NAME                    NUM_DISTINCT HISTOGRAM
------------------------------ ------------ ---------------
USERID                               161262 HEIGHT BALANCED
USERNAME                             161262 HEIGHT BALANCED
USERSTATUSID                              4 FREQUENCY
COUNTRY                                  34 FREQUENCY
STATE                                    47 FREQUENCY
```

Create the Explain Plan to see the Query Optimizer Estimates

```
SQL> explain plan for
  2  select *
  3  from indy_test where lower(country) = 'india';
```

Explained.

Optimizer Estimates 1833 Rows which are even reduced and so let's create a virtual column with Statistics generated on this virtual column in next step

```
SQL> select plan_table_output
  2  from table(dbms_xplan.display('plan_table',null,'BASIC ROWS'));

PLAN_TABLE_OUTPUT
-----------------------------------------------
| Id  | Operation          | Name       | Rows  |
-----------------------------------------------
|   0 | SELECT STATEMENT   |            |  1833 |
|   1 |  TABLE ACCESS FULL| INDY_TEST  |  1833 |
-----------------------------------------------
```

7 rows selected.

QUERY RESULTS WITH EXPRESSION STATISTICS COLLECTED

Create the extended Stats on the Expression used in the query i.e. lower(country)

```
SQL> exec dbms_stats.gather_table_stats(null,'indy_test',method_opt => 'for all columns
size skewonly for columns (lower(country)) size skewonly');
```

PL/SQL procedure successfully completed.

Check if the Virtual Column with Statistics is available

```
SQL> select column_name, num_distinct, histogram
  2  from user_tab_col_statistics
  3  where table_name = 'INDY_TEST'
  4  /
COLUMN_NAME                    NUM_DISTINCT HISTOGRAM
------------------------------ ------------ ---------------
USERID                               161262 HEIGHT BALANCED
USERNAME                             161262 HEIGHT BALANCED
USERSTATUSID                              4 FREQUENCY
COUNTRY                                  34 FREQUENCY
STATE                                    47 FREQUENCY
SYS_STUEJ3KN8Q$YG7Z8LBEZ7A6P3G            34 FREQUENCY
6 rows selected.
```

Make sure that the Virtual Column shown above in RED is the one linked to expression lower(country)

```
SQL> select table_name,extension_name,extension from user_stat_extensions where
table_name='INDY_TEST';

TABLE_NAME      EXTENSION_NAME                 EXTENSION
--------------- ------------------------------ --------------------
INDY_TEST       SYS_STUEJ3KN8Q$YG7Z8LBEZ7A6P3G (LOWER("COUNTRY"))
```

Create the Explain Plan to see the Query Optimizer Estimates

```
SQL> explain plan for
  2  select *
  3  from indy_test where lower(country) = 'india';
Explained.
```

Optimizer Estimates 2035 Rows which are close to number of records (2189)

```
SQL> select plan_table_output
  2  from table(dbms_xplan.display('plan_table',null,'BASIC ROWS'));
PLAN_TABLE_OUTPUT
--------------------------------------------
| Id  | Operation          | Name       | Rows  |
--------------------------------------------
|   0 | SELECT STATEMENT   |            |  2035 |
|   1 |  TABLE ACCESS FULL| INDY_TEST |  2035 |
--------------------------------------------

7 rows selected.
```

Drop the Extended Virtual column

Find the detail about the expression or Virtual column which you want to delete

```
SQL> select table_name,extension_name,extension from user_stat_extensions where
table_name='INDY_TEST';
```

```
TABLE_NAME       EXTENSION_NAME                    EXTENSION
---------------- --------------------------------- --------------------
INDY_TEST        SYS_STUEJ3KN8Q$YG7Z8LBEZ7A6P3G (LOWER("COUNTRY"))
```

Delete the Virtual Column using drop_extended_stats procedure

```
SQL> exec dbms_stats.drop_extended_stats(null,'indy_test','(lower(country))');
```

PL/SQL procedure successfully completed.

Verify if the expression and its statistics is deleted

```
SQL> select column_name, num_distinct, histogram
  2  from user_tab_col_statistics
  3  where table_name = 'INDY_TEST'
  4  /

COLUMN_NAME                    NUM_DISTINCT HISTOGRAM
------------------------------ ------------ ---------------
USERID                               161262 HEIGHT BALANCED
USERNAME                             161262 HEIGHT BALANCED
USERSTATUSID                              4 FREQUENCY
COUNTRY                                  34 FREQUENCY
STATE                                    47 FREQUENCY
```

```
SQL> select table_name,extension_name,extension from user_stat_extensions where
table_name='INDY_TEST';
```

no rows selected