COISAS QUE VOCÊ PRECISA SABER SOBRE ESTATÍSTICAS DO ORACLE

DBACORP®

Inteligência aplicada a TI

Dennis Ruiz
dennis.ruiz@dbacorp.com.br

DESCRIÇÃO

Neste artigo irei falar sobre Estatísticas do Oracle de uma forma clara e exemplificada, citando as formas utilizadas como RBO e CBO com o título “Coisas que você precisa saber sobre estatísticas do Oracle”.

Para iniciar o artigo, acho prudente entendermos como o Oracle trabalha com o otimizador. Até a versão 9i, eram utilizados RBO e CBO.

O QUE É CBO - COST-BASED OPTIMIZATION (OTIMIZAÇÃO BASEADA EM CUSTO)?

A otimização baseada no custo do Oracle foi projetada para determinar a maneira mais eficiente de realizar uma instrução SQL, mas não pode chegar a fazer isso sem uma boa atualização das informações de estatísticas sobre os dados que estão sendo acessados. O otimizador pode usar uma abordagem baseada em regras (RULE) para trabalhar sem informação de estatística, mas esta abordagem é menos inteligente do que a abordagem baseada nos custos. Com a abordagem baseada em regras, o otimizador escolhe um plano de execução com base num conjunto de regras sobre que tipos de operações normalmente irá executar mais rápido do que outros. Com a abordagem baseadas nos custos (CBO), o fatore do otimizador serão baseados em informações de estatísticas sobre o conteúdo dos objetos de esquema particulares (tabelas, clusters ou índices) que estão sendo acessados.

O que é RBO - Rule Based Optimizer (Otimização Baseada em Regra)

O RBO está obsoleto desde Oracle 10g. A funcionalidade ainda está presente, mas nenhuma nova funcionalidade foi incluída nele e não é mais suportada pela Oracle. Ele só está presente para fornecer compatibilidade com versões anteriores durante a migração para o otimizador de consulta (Cost Optimizer Base).

Sobre as opções de configurações do otimizador

- Os valores para o parâmetro OPTIMIZER_MODE, CHOOSE e RULE ainda existem, mas não são mais suportados.
- O valor padrão para o parâmetro OPTIMIZER_MODE a partir da versão 10g é ALL_ROWS.
- Os hints CHOOSE e RULE ainda existem, mas não são mais suportados.
- Códigos exigindo uso do RBO devem ser migrados para usar o otimizador de consulta CBO. Caso tenha um problema de performance utilizando RBO, abrir um chamado na Oracle não será possível por não ser mais suportado.

CBO e histogramas

A proporção estimada de linhas a ser retornado por uma fonte de linhas, mais comumente conhecida como a SELETIVIDADE, desempenha um papel importante na otimização da consulta pelo CBO. Ele tem um valor entre 0 e 1. A grosso modo, CBO prefere fazer verificação de índice, se uma pequena proporção de linhas satisfazer o predicado e uma varredura completa na tabela (FULL TABLE SCAN), se grande parte dos dados da tabela precisa ser obtida.

Banco utilizado no exemplo:

```
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
PL/SQL Release 11.2.0.4.0 - Production
CORE     11.2.0.4.0      Production
TNS for Linux: Version 11.2.0.4.0 - Production
NLSRTL Version 11.2.0.4.0 - Production
```

Por exemplo, a consulta dada abaixo provavelmente irá optar por usar verificação de índice, se existir uma pequena fração de linhas onde “deptno” é igual a 10:

```
select * from emp where deptno=10;
```

Para estimar a seletividade (ou em outras palavras, utilizar um plano de execução ideal), CBO assume várias entradas na forma de estatísticas, os parâmetros de configuração etc. Do ponto de vista da coluna de uma tabela, CBO coleta as seguintes estatísticas:

- Número de valores distintos em uma coluna;
- Valor baixo e alto de uma coluna;
- Número de nulos em uma distribuição de dados da coluna;
- OU informações do histograma (opcional).

Na ausência de histogramas, o otimizador calcula as estatísticas com base nas três primeiras opções da informação, ou seja, o número de valores distintos na coluna, os valores baixos e altos para uma coluna, número de nulos em uma coluna e número de registros na tabela subjacente.

Com esta informação, o otimizador assume a distribuição de dados uniformes entre os valores baixos e altos de uma coluna, ou em outras palavras, a ocorrência de cada valor distinto de uma coluna serem iguais.

Para um melhor entendimento, vamos exemplificar por meio de uma tabela com 10.000 linhas e duas colunas. Coluna ALL_DISTINCT contém linhas distintas que variam de 1-10.000. A coluna (SKEW) tem valores 1-10 durante as primeiras 10 linhas e valor de 10.000 para restante dos registros, ou seja, 9.990 linhas.

```
DENNISDB SQL >create table HISTOGRAM as select rownum ALL_DISTINCT, 10000 SKEW from dual connect
by level <= 10000;

DENNISDB SQL >update HISTOGRAM set skew= ALL_DISTINCT where rownum<=10;

10 rows updated.


DENNISDB SQL > select skew, count(*) from HISTOGRAM group by skew order by skew;

      SKEW   COUNT(*)
---------- ----------
         1          1
         2          1
         3          1
         4          1
         5          1
         6          1
         7          1
         8          1
         9          1
        10          1
     10000       9990

11 rows selected.
```

Agora coletamos as estatísticas do otimizador. “Note” que desta vez não estamos criando histogramas devido o valor SIZE ser igual a 1.

```
DENNISDB SQL >exec dbms_stats.gather_table_stats(user,'HISTOGRAM', method_opt=>'for all columns
size 1');

PL/SQL procedure successfully completed.

DENNISDB SQL > select column_name,num_distinct,density from user_tab_col_statistics where table_
name='HISTOGRAM';

COLUMN_NAME                    NUM_DISTINCT    DENSITY
------------------------------ ------------ ----------
ALL_DISTINCT                          10000      .0001
SKEW                                     11 .090909091
```

Estatísticas de densidade para uma coluna representam a sua seletividade que é calculada como 1 / # NUM_DISTINCT, se não estiver presente o histograma para uma coluna. Em caso de histogramas, cálculo da densidade depende do tipo de histograma e versão do Oracle. O valor de densidade é entre 0 e 1. Otimizador usa esta estatística para estimar o número de linhas a serem retornados (também chamado de cardinalidade) por uma consulta usando esta coluna no predicado.

Filosoficamente falando, “a cardinalidade é um número que expressa o comportamento (número de ocorrências) de determinada entidade associada a uma ocorrência da entidade em questão através do relacionamento”. No caso, entidade é o campo-chave a ser usado no relacionamento que será estabelecido.

A cardinalidade efetiva (effective cardinality) é a quantidade de registros que realmente serão selecionados da fonte de dados. Esse valor depende das condições definidas na consulta, já que as condições filtrarão os valores que devem ser retornados. Ela é calculada pelo produto da cardinalidade básica e a seletividade das condições informadas na consulta. Se não houver nenhuma condição na consulta, então a cardinalidade efetiva é igual à cardinalidade básica.

A cardinalidade de junção (join cardinality) é o número de linhas produzido quando é feita a junção de duas fontes de dados. Uma junção é o produto cartesiano de duas fontes de dados, aplicando as condições para efetuar a junção entre as fontes. Dessa forma, a cardinalidade de junção é o produto da cardinalidade das duas fontes de dados multiplicado pela seletividade das condições de junção.

A cardinalidade de distinção (distinct cardinality) é o número de valores diferenciados em uma coluna da fonte de dados. Por exemplo, a tabela “HISTOGRAM” possui 10.000 registros, mas apenas 11 valores diferentes para o campo “SKEW”, então a cardinalidade de distinção desta tabela é 11.

A cardinalidade de agrupamento (group cardinality) é a quantidade de registros existentes em uma fonte de dados após se aplicar o operador GROUP BY. Este valor depende da cardinalidade de distinção de cada coluna informada no operador GROUP BY. Trata-se de um valor entre a maior cardinalidade de distinção entre os campos do agrupamento e o menor valor entre o produto da cardinalidade de distinção dos campos e a quantidade de registros na fonte de dados.

Então **cardinalidade** = seletividade * Número de linhas.
Vamos ver os valores de cardinalidade para diferentes valores da coluna SKEW no predicado:

```
DENNISDB SQL >explain plan for select * from histogram where skew=1;

Explained.

DENNISDB SQL >select * from table(dbms_xplan.display);

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------
Plan hash value: 941738150

-------------------------------------------------------------------------------
| Id  | Operation         | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |           |   909 |  6363 |    17  (53)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| HISTOGRAM |   909 |  6363 |    17  (53)| 00:00:01 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("SKEW"=1)

13 rows selected.
```

```
DENNISDB SQL > explain plan for select * from histogram where skew=10000;

Explained.

DENNISDB SQL >select * from table(dbms_xplan.display);

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------
Plan hash value: 941738150

-------------------------------------------------------------------------------
| Id  | Operation         | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |           |   909 |  6363 |    17  (53)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| HISTOGRAM |   909 |  6363 |    17  (53)| 00:00:01 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("SKEW"=10000)

13 rows selected.
```

O Oracle está assumindo a distribuição de dados uniformes nos valores da coluna SKEW e estimando a cardinalidade = densidade * 10000 = 909,09 linhas. No entanto, sabemos que temos apenas uma linha com SKEW = 1 e 9990 linhas com SKEW = 10000. Essa suposição é obrigada a resultar em plano de execução ideal. Por exemplo, se temos um índice na coluna SKEW, o Oracle vai usá-lo para SKEW = 10000 considerando o número de linhas a serem retornados ser igual a 909, ou apenas 9,09%.
Agora criei e coletei estatísticas para o índice:

```
DENNISDB SQL >create index skew_idx on histogram(skew);

Index created.


DENNISDB SQL > exec dbms_stats.gather_index_stats(user,'SKEW_IDX');

PL/SQL procedure successfully completed.
```

Analisando o plano agora com o filtro 10000:

```
DENNISDB SQL >explain plan for select * from histogram where skew=10000;

Explained.

DENNISDB SQL >select * from table(dbms_xplan.display);

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------
Plan hash value: 2822933374

-----------------------------------------------------------------------------------------
| Id  | Operation                    | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |           |   909 |  6363 |     5  (20)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID| HISTOGRAM |   909 |  6363 |     5  (20)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN           | SKEW_IDX  |   909 |       |     3  (34)| 00:00:01 |
-----------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("SKEW"=10000)

14 rows selected.
```

Desta forma, entenderemos que sem dar entradas adicionais, CBO assume uma distribuição uniforme de dados entre valores baixos e altos de uma coluna e escolhe um "plano ideal".

Oracle Histogramas

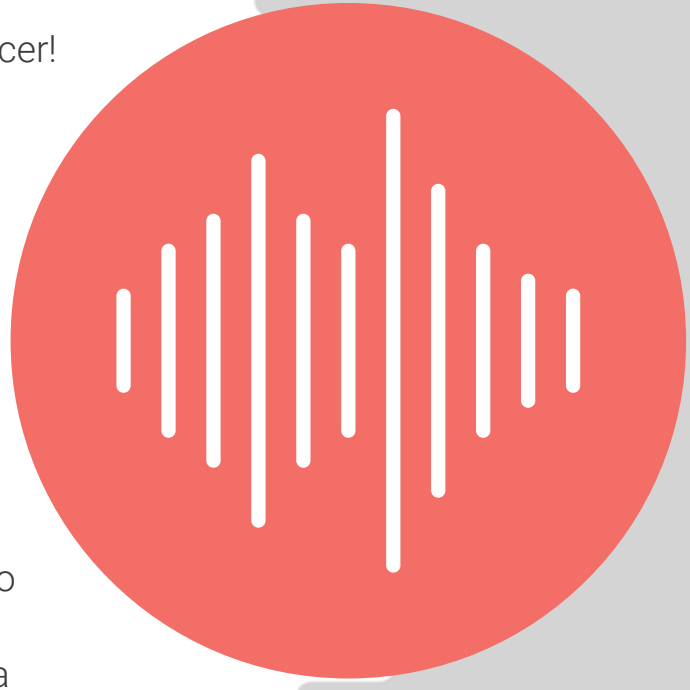
Uma vez que o histograma é criado para uma coluna, ele conta ao CBO a frequência de um valor de coluna. Portanto, no nosso caso, seria dizer ao otimizador que temos 1 (frequência) da coluna SKEW = 1 e 9990 da coluna SKEW = 10000. Assim, ele vai permitir ao otimizador escolher melhor os planos de execução.

No Oracle, até a versão 11g tínhamos dois tipos de histogramas. Na versão 12c temos a combinação e o aperfeiçoamento destes histogramas que farei um breve comentário mais à frente. Trata-se dos histogramas Top Frequency Histograms e Hybrid Histograms (presentes no 12c).

Em primeiro lugar, é onde o Oracle escolhe para armazenar cada valor distinto, juntamente com o número de linhas ou a frequência para esse valor. Na terminologia Oracle, chamamos isso de histogramas de largura equilibrada (width-balanced) ou histogramas de frequência (frequency).
Isso é eficiente e possível para as colunas que têm o pequeno número de valores distintos. No entanto, para colunas com grande número de valores distintos não é viável armazenar todos e cada valor, juntamente com a sua frequência. Claro que com recursos ilimitados (espaço para armazenamento e poder computacional durante o tempo de análise), poderíamos armazenar a frequência para cada valor distinto em cada situação e fornecer informações definitivas para o otimizador, mas que não é o caso na vida real, e nesses casos, a Oracle usa um método diferente de armazenar dados chamados histogramas de altura equilibrada (height-balanced).

Do ponto de vista do usuário, há uma cláusula única para criar um histograma para uma coluna, e o Oracle decide automaticamente qual o tipo de histograma criar com base no número de valores distintos. No entanto, as informações armazenadas de histograma são interpretadas de forma diferente, dependendo do tipo de histograma.

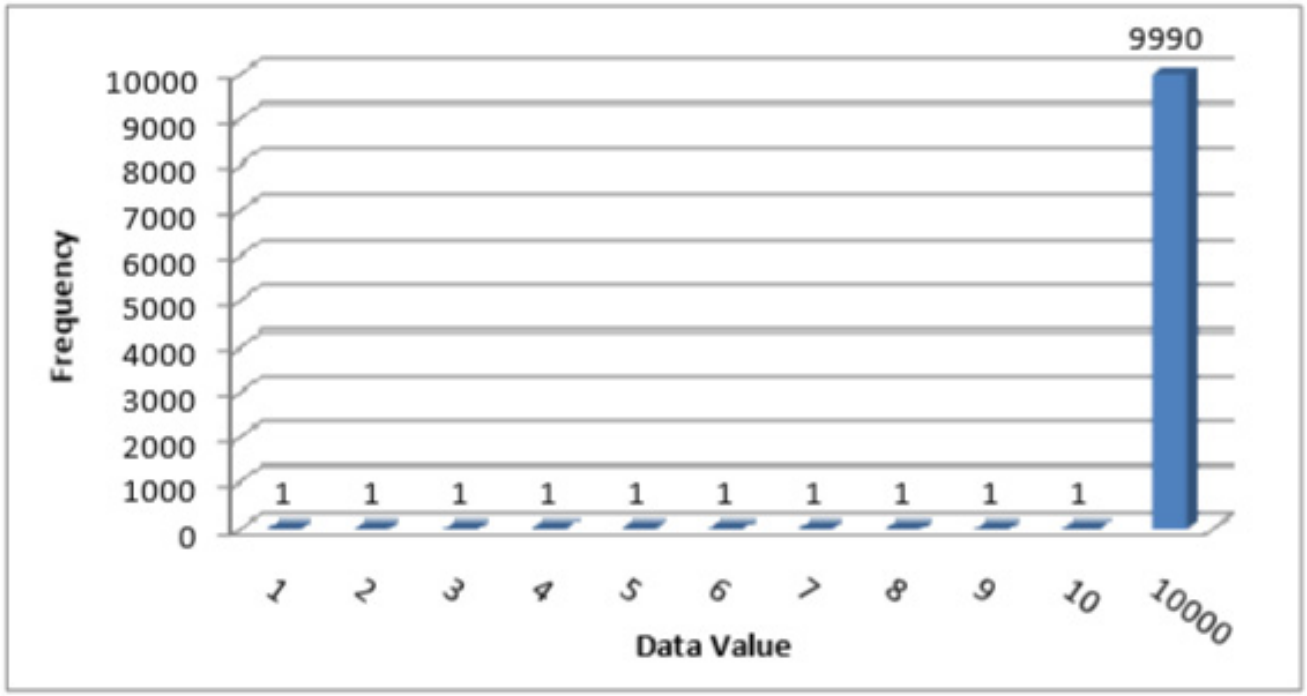
Se você está vivo até aqui, ótimo! A coisa daqui pra frente irá piorar. Bora enlouquecer!



Width-balanced (Largura equilibrada) OU Frequency Histograms (histogramas de Frequência)

Este tipo de valores da coluna do histograma é dividido em categorias de tamanho igual ou em buckets (baldes) na terminologia Oracle.
Este tipo de histograma é similar aos gráficos de barras, mas no caso da Oracle, cada segmento contém apenas um único valor que corresponde a um valor distinto.
A figura a seguir é uma representação gráfica de valores de dados na coluna de inclinação. Alguns pontos são claros ao olhar para o gráfico:

- Existem 11 buckets (baldes) representados no eixo dos X, cada um para um único valor distinto.
- Eixo Y está mostrando suas frequências correspondentes. Frequência é de 1 para dados de valores 1-10 e 9990 para o valor de 10000.
- Pode-se dizer facilmente a frequência de um valor de dados em particular ao olhar para essa de informação (formatos gráficos ou outros).



Agora vamos criar um histograma de frequência para a coluna SKEW e ver como os dados são armazenados nas visualizações de dados do dicionário.
Veremos as diferentes opções disponíveis para a criação de histograma em uma seção posterior, por enquanto vamos entender que METHOD_OPT 'for column column_name size n' cria histograma para a coluna "column_name" com "n" número de baldes.

Coletando estatísticas para a coluna SKEW com 11 buckets (baldes):

```
DENNISDB SQL > exec dbms_stats.gather_table_stats(user,'HISTOGRAM',method_opt=>'for columns SKEW
size 11');

PL/SQL procedure successfully completed.
```

Verificando o resultado olhando a VIEW user_tab_histograms:

```
DENNISDB SQL >col column_name for A20
DENNISDB SQL >select column_name, endpoint_number, endpoint_value from user_tab_histograms where
table_name='HISTOGRAM' and column_name='SKEW';

COLUMN_NAME          ENDPOINT_NUMBER ENDPOINT_VALUE
-------------------- --------------- --------------
SKEW                               1              1
SKEW                               2              2
SKEW                               3              3
SKEW                               4              4
SKEW                               5              5
SKEW                               6              6
SKEW                               7              7
SKEW                               8              8
SKEW                               9              9
SKEW                              10             10
SKEW                           10000          10000

11 rows selected.
```

A primeira instrução cria histograma para o nome da coluna de inclinação com 11 buckets (baldes), pois sabemos que existem 11 valores distintos para a coluna SKEW.

A segunda instrução mostra dados de histograma que são armazenados no dicionário do Oracle. A forma como a informação armazenada no histograma é interpretada de forma diferente, dependendo se o número de buckets (baldes) solicitados é menor que o número de valores distintos ou é o mesmo, ou em outras palavras, depende do tipo de histograma. Abaixo, refere-se à explicação apenas histogramas de frequência.

O ENDPOINT_VALUE mostra o valor real e a coluna ENDPOINT_NUMBER mostra o número cumulativo de linhas, ou em outras palavras, de frequência cumulativa. Para calcular a frequência de um valor de uma coluna particular, precisamos subtrair o valor acumulado anterior a partir do correspondente ENDPOINT_NUMBER.

Por exemplo, para ENDPOINT_VALUE 5, temos ENDPOINT_NUMBER 5 e o ENDPOINT_NUMBER anterior é 4, portanto, o número de linhas com um valor 5 é 1.
De modo semelhante, por ENDPOINT_VALUE 10000, a sua ENDPOINT_NUMBER é 10000 e o ENDPOINT_NUMBER do segmento anterior é de 10, de modo que o número de linhas com valor 10.000 seria 10000-10 = 9990.

Seguindo o SQL abaixo, pode ser usado para traduzir a informação histograma armazenadas no dicionário de dados em um relatório amigavelmente de dados subjacente:

```
DENNISDB SQL >SELECT endpoint_value  AS column_value,
  3                 endpoint_number AS cummulative_frequency,
  4                 endpoint_number - LAG(endpoint_number, 1, 0) OVER (ORDER BY endpoint_number)
AS frequency
  5             FROM user_tab_histograms
  6            WHERE table_name = 'HISTOGRAM'
  7              AND column_name = 'SKEW';

COLUMN_VALUE CUMMULATIVE_FREQUENCY  FREQUENCY
------------ --------------------- ----------
           1                     1          1
           2                     2          1
           3                     3          1
           4                     4          1
           5                     5          1
           6                     6          1
           7                     7          1
           8                     8          1
           9                     9          1
          10                    10          1
       10000                 10000       9990

11 rows selected.
```

Armazenamento de frequência total OU cumulativa, ao invés de frequência individual é particularmente útil em range scans (varreduras por intervalo), onde a cardinalidade de um predicado como WHERE SKEW <= 10 está prontamente disponível.

Agora, uma vez que criamos histograma para a coluna SKEW, vamos ver se faz a diferença.

```
DENNISDB SQL >SELECT column_name,
  3                 density,
  4                 histogram
  5            FROM user_tab_col_statistics
  6           WHERE table_name = 'HISTOGRAM'
  7             AND column_name = 'SKEW';

COLUMN_NAME             DENSITY HISTOGRAM
-------------------- ---------- ---------------
SKEW                      .00005 FREQUENCY

1 row selected.
```

Com o valor da coluna SKEW = 10000

```
DENNISDB SQL >explain plan for select * from histogram where skew=10000;

Explained.


DENNISDB SQL >select * from table(dbms_xplan.display);

PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------------------
Plan hash value: 941738150

-------------------------------------------------------------------------------
| Id  | Operation         | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |           |  9990 | 69930 |    17  (53)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| HISTOGRAM |  9990 | 69930 |    17  (53)| 00:00:01 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("SKEW"=10000)

13 rows selected.
```

Com o valor da coluna SKEW = 1

```
DENNISDB SQL > explain plan for select * from histogram where skew=1;

Explained.


DENNISDB SQL >select * from table(dbms_xplan.display);

PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------------------
Plan hash value: 2822933374

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |           |     1 |     7 |     2   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID| HISTOGRAM |     1 |     7 |     2   (0)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN           | SKEW_IDX  |     1 |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("SKEW"=1)

14 rows selected.
```

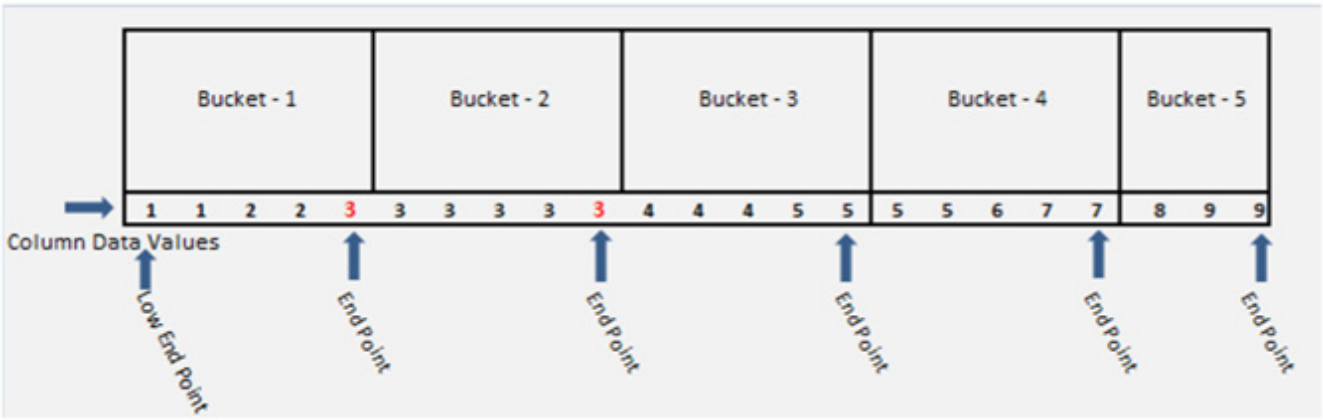
Então agora o otimizador está calculando com precisão a cardinalidade de 9990 linhas e devido a esse cálculo agora o otimizador escolheu varredura completa da tabela para o predicado SKEW = 10000. Além disso, note que agora o valor de densidade é alterada para 0,00005 que é 1 / (2 * num_rows) ou 0,5 / num_rows.

Height-balanced Histograms

Neste caso dos histogramas de frequência, o Oracle aloca um bucket (balde) para cada valor distinto. No entanto, o número máximo possível de Buckets é 254 (até a versão 11g e 2048 na versão 12c), então se você tem tabelas com um grande número de valores distintos (superior a 254), você teria que ir para Height-balanced Histograms (histogramas de altura equilibrada).

Em histogramas de altura equilibrada, uma vez que temos muitos valores distintos maiores do que os buckets, então o Oracle classifica primeiro os dados da coluna, e em seguida, o conjunto de dados completo é dividido em número de buckets (baldes) e esses contêm o mesmo número de valores (que é por isso que eles são chamados histogramas de altura equilibrada), exceto o último bucket que pode ter menos valores do que os demais.

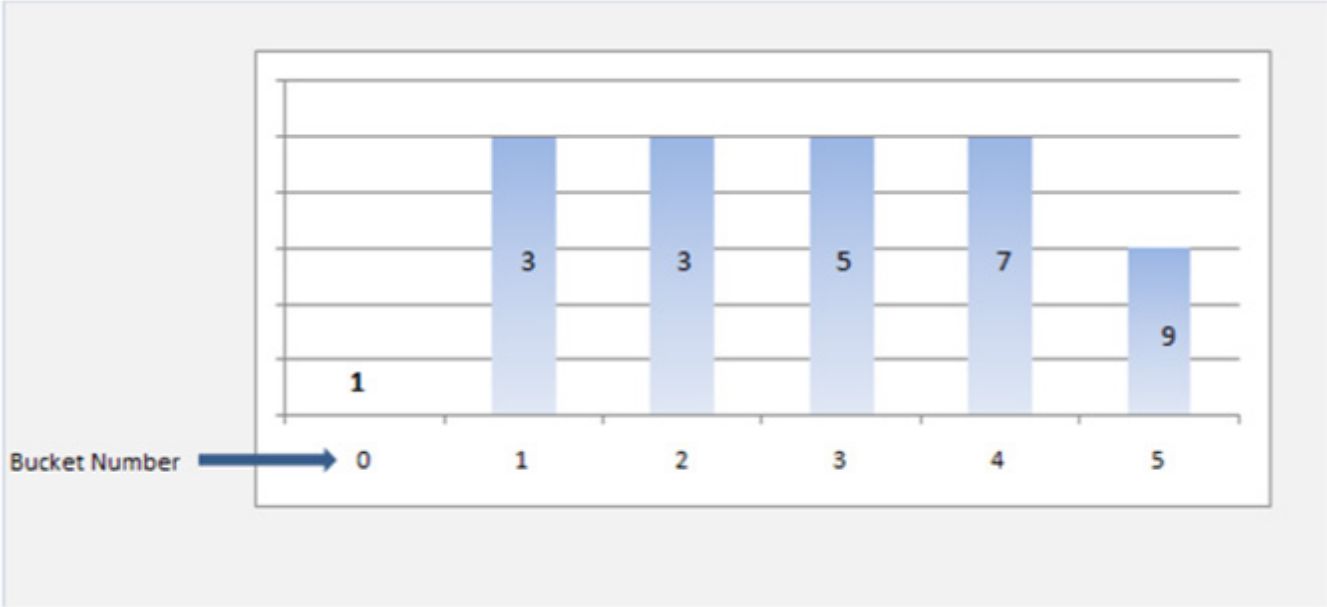
Não existe um comando separado para criar Height-balanced Histograms (histogramas altura equilibrada). Quando o número de buckets solicitados for menor que o número de valores distintos em uma coluna, o Oracle cria histogramas altura equilibrada e o significado de ENDPOINT_VALUE e ENDPOINT_NUMBER é bem diferente. Para entender como interpretar as informações do histograma, vamos dar um outro exemplo de dados da coluna que tem 23 valores e há 9 valores distintos na coluna. Vamos supor que tenha coletado histograma com 5 buckets (baldes). Abaixo encontra-se uma representação de como os dados serão armazenados no histograma.



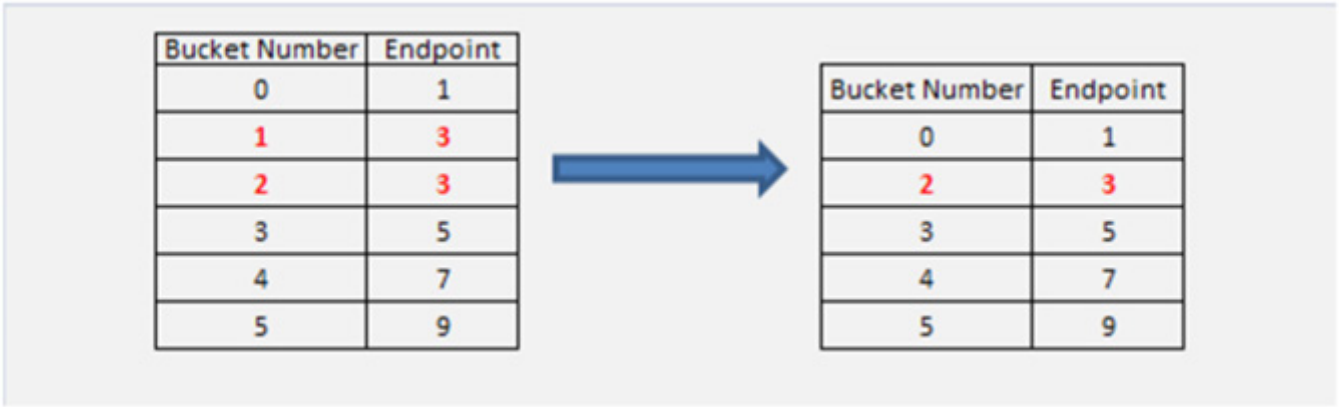
Nós podemos fazer a seguinte conclusão baseado na imagem anterior:

- Número de baldes é menor do que o número de valores distintos na coluna.
- Uma vez que temos solicitados 5 buckets, de modo que o conjunto de dados total será dividido em bucket de tamanho igual, exceto o último balde, que neste caso tem apenas 3 valores.
- Os pontos finais de cada bucket e o primeiro ponto do primeiro segmento são marcados, já que são de interesse especial.
- Valor de dados '3' são marcados na cor vermelha; é especial na medida em que é ponto final em múltiplos recipientes.

A figura seguinte é uma forma alternativa de mostrar o histograma.



Com 5 buckets (baldes) e 23 valores significa que há 5 valores em cada segmento, exceto que o último bucket tem 3 valores. Na verdade este é o caminho a qual o Oracle armazena informações histograma altura equilibrada em dados nas views de dicionário, com uma pequena alteração.
Desde que balde 1 e 2 têm tanto 3 como um ponto final, Oracle não armazena um bucket, de modo a economizar espaço. Então ambos os buckets serão fundidos (combinados) e uma única entrada será armazenada.



Bucket Number	Endpoint
0	1
1	3
2	3
3	5
4	7
5	9

Bucket Number	Endpoint
0	1
2	3
3	5
4	7
5	9

Vamos criar histograma na coluna SKEW, desta vez com o número de baldes menor do que o número real de valores distintos que é de 11.

```
DENNISDB SQL > exec dbms_stats.gather_table_stats(user,'HISTOGRAM',method_opt=>'for columns skew
size 5');

PL/SQL procedure successfully completed.



DENNISDB SQL >SELECT endpoint_value  AS column_value,
  2                 endpoint_number AS cummulative_frequency,
  3                 endpoint_number - LAG(endpoint_number, 1, 0) OVER (ORDER BY endpoint_number)
AS frequency
  4             FROM user_tab_histograms
  5            WHERE table_name = 'HISTOGRAM'
  6              AND column_name = 'SKEW';

COLUMN_VALUE CUMMULATIVE_FREQUENCY  FREQUENCY
------------ --------------------- ----------
           1                     0          0
       10000                     5          5
```

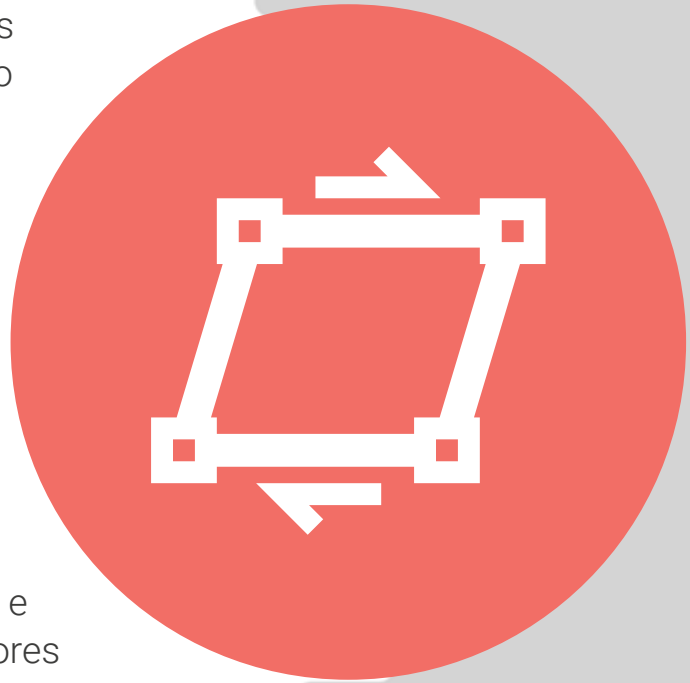
Neste, os Buckets 1-5, todos têm 10000 como um ponto final, para os buckets de 1-4 não são armazenados de forma a economizar espaço.
Seguinte consulta SQL pode ser usada para exibir os números de baldes e respectivos números de ponto final.

```
DENNISDB SQL >SELECT bucket_number
  2                 ,max(skew) AS endpoint_value
  3             FROM ( SELECT skew
  4                           ,ntile(5) OVER (
  5                            ORDER BY skew
  6                                          ) AS bucket_number
  7                      FROM histogram
  8                  )
  9            GROUP BY bucket_number
 10            ORDER BY bucket_number;

BUCKET_NUMBER ENDPOINT_VALUE
------------- --------------
            1          10000
            2          10000
            3          10000
            4          10000
            5          10000

5 rows selected.
```

Aqui, nTILE (5) é uma função analítica, ela divide um conjunto em 5 buckets (baldes) dos dados ordenados. Assim, em resumo, em histogramas de altura equilibrada, os dados são divididos em diferentes buckets, onde cada segmento contém o mesmo número de valores. O valor mais alto em cada bucket é gravado em conjunto (ENDPOINT_VALUE) com o valor mais baixo no primeiro bucket (bucket 0). Além disso, ENDPOINT_NUMBER representa o número de bucket. Uma vez que os dados são gravados no buckets, reconhecemos 2 tipos de valores de dados - valores não-populares e valores populares.



Valores populares são aquelas que ocorrem várias vezes no ENDPOINT.
Por exemplo, no nosso exemplo anterior, 3 é um valor popular e a coluna SKEW 10000 é um valor popular.
Valores não-populares são aqueles que não ocorrem várias vezes no ENDPOINT.
Como você pode estar pensando, valores populares e não-populares não são fixos e dependem do tamanho do bucket. Alterar o tamanho do bucket irá resultar em valores populares diferentes.

Resumindo, os dois tipos de histogramas do 11g:

- Valores distintos inferiores ou iguais ao número de buckets: quando você tem um número menor de valores distintos do que o número de buckets, a coluna ENDPOINT_VALUE contém os valores distintos si próprios, enquanto a coluna ENDPOINT_NUMBER contém o número acumulado de linhas com o valor menor do que da coluna (histogramas de frequência).
- Mais número de valores distintos do que o número de buckets: quando tiver mais número de valores distintos do que o número de buckets, a coluna ENDPOINT_NUMBER contém a ID do bucket e do ENDPOINT_VALUE contêm o maior valor em cada bucket. Bucket 0 é especial na medida em que mantém o baixo valor para essa coluna (histogramas de altura equilibrada).

Top Frequency Histograms (12c)

É utilizado se:

- Se o NDV (Number Distinct Values) for maior que a quantidade de Buckets indicados na coleta;
- For utilizado AUTO_SAMPLE_SIZE na execução das coletas das estatísticas.
- O percentual de linhas ocupadas pelos Top Values for igual ou maior que P, sendo que P = (1-(1/Buckets))*100.

Hybrid Histograms (12c)

É utilizado se:

- Se o número de Buckets indicados na coleta for menor que o NDV;
- For utilizado AUTO_SAMPLE_SIZE na execução das coletas das estatisticas;
- Os critétios para Top Frequency Histograms não se aplicam.

Criando histogramas

A procedure GATHER_TABLE_STATS da package DBMS_STATS é usada para coletar as estatísticas de tabelas e colunas e, opcionalmente, podemos instruir para criar histograma em determinadas colunas usando o parâmetro METHOD_OPT:
O parâmetro 'METHOD_OPT' da procedure aceita seguintes valores:

- FOR ALL [Indexed / Hidden] Columns [Size option]
- FOR COLUMNS column_name [Size option] column_name [Size option] ...

A palavra-chave SIZE especifica o número máximo de buckets (baldes) para o histograma e pode ser setado com os seguintes valores:

SIZE {integer | REPEAT | AUTO | SKEWONLY}

- integer: Número de buckets de histograma. Deve estar no intervalo 1-254 (para Oracle 11g).
- Repeat: Coleta histogramas APENAS nas colunas que já têm histogramas.
- Auto: o Oracle determina às COLUNAS coletar os histogramas com base na distribuição de dados e a carga de trabalho (Workload) das colunas.
- Skewonly: o Oracle determina às COLUNAS coletar os histogramas com base na distribuição dos dados das colunas.

A opção AUTO também considera a carga de trabalho (workload) das colunas. O que significa que ele verifica as consultas SQL com o nome das colunas na qual tem o predicado.
O padrão para METHOD_OPT é alterada para "FOR ALL COLUMNS SIZE AUTO" em 10g, o que no 9i costumava ter "FOR ALL COLUMNS SIZE 1". Em outras palavras, o Oracle agora decide automaticamente para nós quais colunas precisam de histogramas e número de buckets também. Isto parece situação ideal, mas esta tem muitas ressalvas que não estão no foco deste artigo.

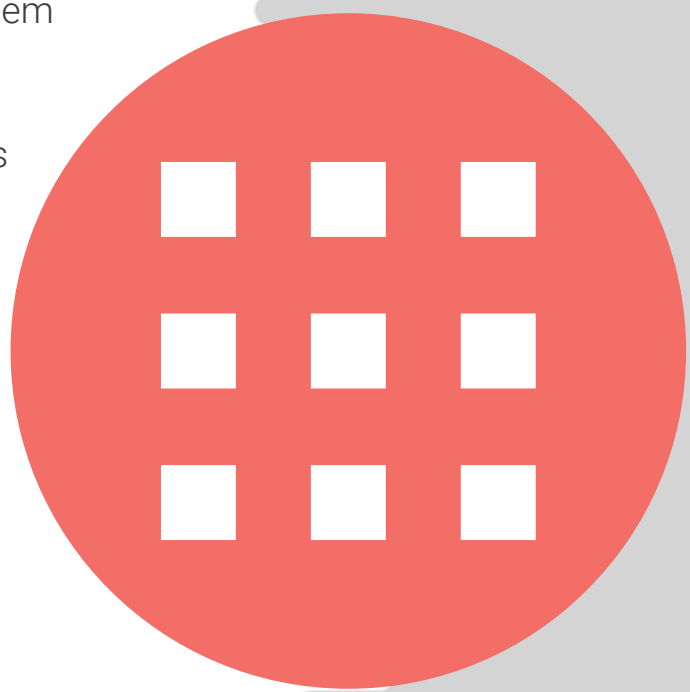
Visualização dos histogramas

Estamos buscando informações do histograma o tempo todo, agora é hora de ver em detalhes as várias opções disponíveis para obter informações de histograma.

Podemos encontrar informações sobre histogramas existentes no banco de dados através da view do dicionário de dados DBA_TAB_HISTOGRAMS. Esta VIEW lista histogramas em colunas de todas as tabelas. O valor real pode ser armazenado em ENDPOINT_ACTUAL_VALUE se a coluna não é um número (isto é, um VARCHAR2) e os primeiros seis bytes de alguns valores são os mesmos.

Número de buckets no valor do histograma e densidade de cada coluna pode ser encontrada na visão de dicionário de dados DBA_TAB_COLUMNS e DBA_TAB_COL_STATISTICS.

Existem VIEWs correspondentes disponíveis para a partição e sub-partições das colunas, por exemplo DBA_SUBPART_HISTOGRAMS etc.



CONCLUSÃO

Vimos até agora conceitos básicos de histogramas do Oracle e como interpretar os dados do histograma armazenados no dicionário de dados para os tipos de histogramas. Como mencionado anteriormente, o uso básico do histograma é proporcionar melhor estimativa da proporção da tabela (cardinalidade) que preenche os critérios de filtragem.
Compreender os cálculos de cardinalidade é o próximo passo na jornada de histogramas do Oracle. Da mesma forma, variáveis de ligação (Bind Variable) e histogramas partilham uma relação muito especial - uma contradição em seu propósito.
Além disso, até agora vimos apenas histogramas em valores numéricos. Temos ainda histogramas em outros tipos de dados como colunas de data e caracteres.
Descer ao nível de criar coletas de estatísticas / histogramas para cada tabela/coluna seria o melhor dos mundos, mas nem sempre é a realidade para nós, consultores, que temos apenas 1 ou 2 dias de visitas para entender toda mecânica do ambiente do seu cliente. Mas a boa notícia para muitos ambientes é que o Oracle com a coleta de histogramas no automático acerta nessa sestatísticas.
Há bastante material sobre esse assunto, mas acredito que este artigo vai dar um start no entendimento dos processos internos nesse mundo do banco de dados chamado Oracle.

LINKS PARA SABER MAIS

Fonte:

https://docs.oracle.com/cd/B10500_01/server.920/a96533/rbo.htm

DBACORP®

Inteligência aplicada a TI

Há 18 anos, a DBACorp atua na área de TI e garante resultados reais para os seus investimentos. Afinal, 95% dos nossos clientes aprovam e recomendam os serviços que oferecemos.

Temos soluções sob medida para os mais diversos projetos em banco de dados, infraestrutura e cloud computing.

Oferecemos serviços como suporte 24x7x365, alocação de profissionais, venda de licenças, monitoração proativa, alta disponibilidade e sustentação.

Tudo isso com equipes certificadas de DBAs e de infraestrutura, prontas para atender todos os segmentos e tamanhos de empresas.

E para garantir ainda mais a qualidade, a DBACorp é parceira dos principais players do mercado, como Oracle, Microsoft, Amazon Web Services e WMware.

DBACorp. Inteligência aplicada a TI e ao seu negócio.

Dennis Ruiz
dennis.ruiz@dbacorp.com.br

DBACORP®

Inteligência aplicada a TI

ESCRITÓRIOS

São Paulo - SP
R. Samuel Morse, 74, Conj. 21
Brooklin Novo - CEP 04576-060
+55 11 2348-4699

Rio de Janeiro - RJ
Av. Rio Branco, 1 - 12º andar
Centro - CEP 20090-003
+55 21 2588-8150

REDES SOCIAIS