

TEMA 5  
OPTIMIZACIÓN DE CONSULTAS DISTRIBUIDAS – PARTE 3

---

**5.4. ESTADÍSTICAS DE LA BASE DE DATOS CENTRALIZADA.**

- Para la obtención de los costos que realiza el optimizador, se apoya fuertemente en las estadísticas recolectadas con respecto a los objetos que se involucran en una consulta.
- Por lo anterior, las estadísticas representan un papel crítico para la selección del plan de ejecución adecuado.

**5.4.1. Clasificación de las estadísticas**

- Estadísticas a nivel de tabla:
  - Número de registros
  - Número de bloques de datos.
  - Longitud promedio de un registro
- Estadísticas a nivel columna:
  - Numero de valores distintos de una columna (NDV)
  - Numero de valores nulos de una columna.
  - Valores mínimos y máximos
  - Distribución de los datos (Histograma).
  - Estadísticas extendidas.
- Estadísticas de los índices
  - Número de nodos intermedios de un índice B-tree
  - Número de hojas de un árbol B-Tree
  - Factor de clusterización de un índice.
- Estadísticas del sistema
  - Uso de lecturas y escrituras a disco.
  - Uso de procesador.

Todos estos valores pueden ser consultados en el diccionario de datos.

**5.4.2. Estadísticas a nivel tabla.**

En Oracle se emplea la función `DBMS_STATS.GATHER_TABLE_STATS` para recolectar este tipo de estadísticas y almacenarlas en `DBA_TAB_STATISTICS`.

**Ejemplo:**

```
col table_name format A20
select table_name,num_rows,blocks,empty_blocks,avg_row_len,last_analyzed
from dba_tab_statistics
where owner ='MEDICOS';
```

TABLE_NAME	NUM_ROWS	BLOCKS	EMPTY_BLOCKS	AVG_ROW_LEN	LAST_ANAL
ESPECIALIDAD	53	5	0	45	22-APR-17
DIAGNOSTICO	14423	370	0	52	22-APR-17
MEDICO	5000	874	0	1065	22-APR-17
PACIENTE	15000	1756	0	773	22-APR-17
CITA	20000	118	0	33	22-APR-17
MEDICAMENTO	2005	244	0	547	22-APR-17
RECETA	20000	73	0	19	22-APR-17

#### 5.4.3. Estadísticas a nivel columna.

##### Ejemplo:

```
col table_name format A20
col column_name format A25
col low_value format A20
col high_value format A20
set linesize 300
select c.table_name,c.column_name,c.num_distinct,
       TO_CHAR(UTL_RAW.CAST_TO_NUMBER(c.low_value)) low_value,
       TO_CHAR(UTL_RAW.CAST_TO_NUMBER(c.high_value)) high_value,
       c.num_nulls,c.avg_col_len,c.histogram,c.last_analyzed
from dba_tab_statistics t, dba_tab_col_statistics c
where c.table_name=t.table_name
and t.owner='MEDICOS' and c.table_name='PACIENTE'
and c.column_name in('PACIENTE_ID')
union
select c.table_name,c.column_name,c.num_distinct,
       UTL_RAW.CAST_TO_VARCHAR2(c.low_value) low_value,
       UTL_RAW.CAST_TO_VARCHAR2(c.high_value) high_value,
       c.num_nulls,c.avg_col_len,c.histogram,c.last_analyzed
from dba_tab_statistics t, dba_tab_col_statistics c
where c.table_name=t.table_name
and t.owner='MEDICOS' and c.table_name='PACIENTE'
and c.column_name
in('NOMBRE','AP_PATERNO','AP_MATERNO','GENERO','NUM_SEGURO'
   ,'CURP','EMAIL','OCUPACION','FECHA_FIN_VIGENCIA');
```

TABLE_NAME	COLUMN_NAME	NUM_DISTINCT	LOW_VALUE	HIGH_VALUE	NUM_NULLS	AVG_COL_LEN	HISTOGRAM	LAST_ANAL
PACIENTE	AP_MATERNO	250	Adams	Young	0	8	NONE	22-APR-17
PACIENTE	AP_PATERNO	250	Adams	Young	0	8	NONE	22-APR-17
PACIENTE	CURP	15000	ABC798701GYT0S7029	ZYX674334JUYHR4765	0	19	NONE	22-APR-17
PACIENTE	EMAIL	13344	aadams3m@wiley.com	wyoungna@fotki.com	1656	20	NONE	22-APR-17
PACIENTE	FECHA_FIN_VIGENCIA	4853	01-APR-01	31-OCT-16	4669	8	NONE	22-APR-17
PACIENTE	GENERO	2	Female	Male	0	6	NONE	22-APR-17
PACIENTE	NOMBRE	200	Aaron	Willie	0	7	NONE	22-APR-17
PACIENTE	NUM_SEGURO	15000	100-04-0885	999-99-9426	0	12	NONE	22-APR-17
PACIENTE	OCUPACION	195	Account Coordinator	Web Developer IV	1881	17	NONE	22-APR-17
PACIENTE	PACIENTE_ID	15000	1	15000	0	5	NONE	22-APR-17

#### 5.4.4. Estadísticas para índices.

El principal uso es para determinar el costo de realizar escaneos a un índice. Las estadísticas se almacenan en DBA\_IND\_STATISTICS. Algunos valores incluyen:

- BLEVEL: Número de bloques requeridos que se deben leer para llegar a los nodos hoja del índice.
- Distinct keys: Número de valores indexados diferentes.
- Número promedio de bloques que se requieren para almacenar cada uno de los valores indexados.

##### Ejemplo:

```

set linesize 200
col index_name format A30
select index_name, blevel, leaf_blocks, distinct_keys,
       avg_leaf_blocks_per_key, avg_data_blocks_per_key
from dba_ind_statistics
where owner = 'MEDICOS'
and index_name
in ('PACIENTE_PK', 'CITA_MEDICO_IDX', 'MEDICO_ESPECIALIDAD_IDX');

```

INDEX_NAME	BLEVEL	LEAF_BLOCKS	DISTINCT_KEYS	AVG_LEAF_BLOCKS_PER_KEY	AVG_DATA_BLOCKS_PER_KEY
PACIENTE_PK	1	27	15000	1	1
MEDICO_ESPECIALIDAD_IDX	1	10	53	1	89
CITA_MEDICO_IDX	1	42	4911	1	3

#### 5.4.5. Recolección de estadísticas.

- En Oracle, la funcionalidad para realizar la administración de recolección de estadísticas se encuentra dentro del paquete PL/SQL DBMS\_STATS.
- Por default la BD realiza una recolección **automática** de estadísticas para todos los objetos que no cuentan con ellas, o aquellas marcadas como “Stale” (Obsoletas).
- Existe otro tipo de estadísticas llamadas “Estadísticas dinámicas suplementarias”. Como su nombre lo indica, se emplean para mejorar las decisiones del optimizador e incluyen: conteo de los bloques que integran a una tabla o índice, estadísticas de operaciones JOIN, GROUP BY.
- Por default la BD realiza recolección de estadísticas bajo ciertos eventos, por ejemplo, carga masiva de datos.
- Los resultados de la ejecución de una consulta SQL también pueden ser recolectados para mejorar los valores estadísticos.
- Adicionalmente, la recolección de estadísticas puede ser manual y a diferentes niveles de generalidad: por objeto, por esquema, o para toda la BD. Se emplean los siguientes procedimientos almacenados:
  - gather\_index\_stats
  - gather\_table\_stats
  - gather\_schema\_stats
  - gather\_dictionary\_stats
  - gather\_database\_stats
- Cuando una tabla es monitoreada, si esta ha sido modificada en más de un 10% (operaciones DML), sus estadísticas se consideran como obsoletas (“Stale”).
- En general, la recolección automática de estadísticas es suficiente para tablas que no se modifican con gran frecuencia.
- Para determinar si una tabla tiene estadísticas obsoletas se puede aplicar el siguiente procedimiento:
  1. Antes de realizar la consulta, se debe persistir la información de monitoreo de los objetos que se encuentra en memoria a disco:

```
begin
  dbms_stats.flush_database_monitoring_info;
end;
/
```

2. Consultar el valor de la columna `stale_stats` para verificar si requiere o no. Esta columna tiene los siguientes valores:

- YES. Las estadísticas están obsoletas.
- NO. Las estadísticas no están obsoletas.
- NULL. Las estadísticas nunca han sido recolectadas.

#### Ejemplo:

```
SQL> connect sys/system@jrcbd_s1 as sysdba
Connected.
select table_name, stale_stats
from dba_tab_statistics
where owner = 'MEDICOS';
```

TABLE_NAME	STA
ESPECIALIDAD	NO
DIAGNOSTICO	NO
MEDICO	NO
PACIENTE	NO
CITA	NO
MEDICAMENTO	NO
RECETA	NO

- Recolección de estadísticas por tabla. Ejemplo:

```
begin
  dbms_stats.gather_table_stats (
    ownname => 'MEDICOS',
    tabname => 'PACIENTE',
    degree => 2
  );
end;
/
```

El parámetro “degree” indica el nivel de paralelismo a emplear para realizar la recolección.

- Recolección a nivel de esquema:

```
begin
  dbms_stats.gather_schema_stats (
    ownname => 'MEDICOS',
    degree => 2
  );
end;
/
```

#### **5.4.6. Histogramas.**

- Representa una medida estadística especial que se emplea para los casos en los que la distribución de los datos no es homogénea. Típicamente existen los siguientes tipos de histogramas:

- Frequency histograms and top frequency histograms
- Height-Balanced histograms (legacy)
- Hybrid histograms
- Por default el optimizador asume una distribución uniforme de los valores de una columna. Por ejemplo, para una columna `nombre`, el optimizador asume que existe el mismo número de empleados con el nombre JUAN, PACO, etc. Lo anterior es poco común que ocurra en la realidad.
- Los histogramas resuelven este inconveniente.

Para provocar la generación de histogramas, se debe producir la siguiente secuencia de pasos:

Suponer que se desea crear un histograma para la columna `nombre` de la tabla `MEDICO`.

1. Ejecutar la recolección de estadísticas para la tabla en cuestión. En esta ocasión se debe especificar el parámetro `METHOD_OPT` para indicar las columnas a las que se les creará el histograma:

```
begin
  dbms_stats.gather_table_stats (
    ownname => 'MEDICOS',
    tabname => 'MEDICO',
    degree   => 2,
    method_opt => 'for columns nombre size AUTO'
  );
end;
/
```

Adicionalmente se puede especificar el siguiente valor si se desea obtener histograma para todas las columnas

```
method_opt => 'for all columns size AUTO'
```

2. Ejecutar una consulta que involucre a las columnas anteriores como predicado:

```
select count(*) from medicos.medico where nombre like 'M%';
```

3. Ejecutar nuevamente la recolección de estadísticas (Paso 1).
4. Consultar la información del diccionario de datos para validar la existencia del histograma:

```
select c.table_name, c.column_name, c.histogram
from dba_tab_statistics t, dba_tab_col_statistics c
where c.table_name=t.table_name
and t.owner='MEDICOS' and c.table_name='MEDICO'
and c.column_name in('MEDICO_ID')
union
select c.table_name, c.column_name, c.histogram
from dba_tab_statistics t, dba_tab_col_statistics c
where c.table_name=t.table_name
and t.owner='MEDICOS' and c.table_name='MEDICO'
and c.column_name in('NOMBRE', 'AP_PATERNO',
  'AP_MATERNO', 'CEDULA', 'ESPECIALIDAD_ID');
```

TABLE_NAME	COLUMN_NAME	HISTOGRAM
MEDICO	AP_MATERNO	NONE
MEDICO	AP_PATERNO	NONE
MEDICO	CEDULA	NONE
MEDICO	ESPECIALIDAD_ID	NONE
MEDICO	MEDICO_ID	NONE
MEDICO	NOMBRE	FREQUENCY

- Observar el valor FREQUENCY para el campo nombre.

Para consultar el histograma generado, se puede realizar la siguiente consulta:

```
col table_name format A20
col column_name format A25
col endpoint_actual_value format A25
set linesize 200
select table_name, column_name, endpoint_number, endpoint_actual_value
from user_histograms where table_name='MEDICO'
and column_name='NOMBRE'
and rownum <=10
order by 1;
```

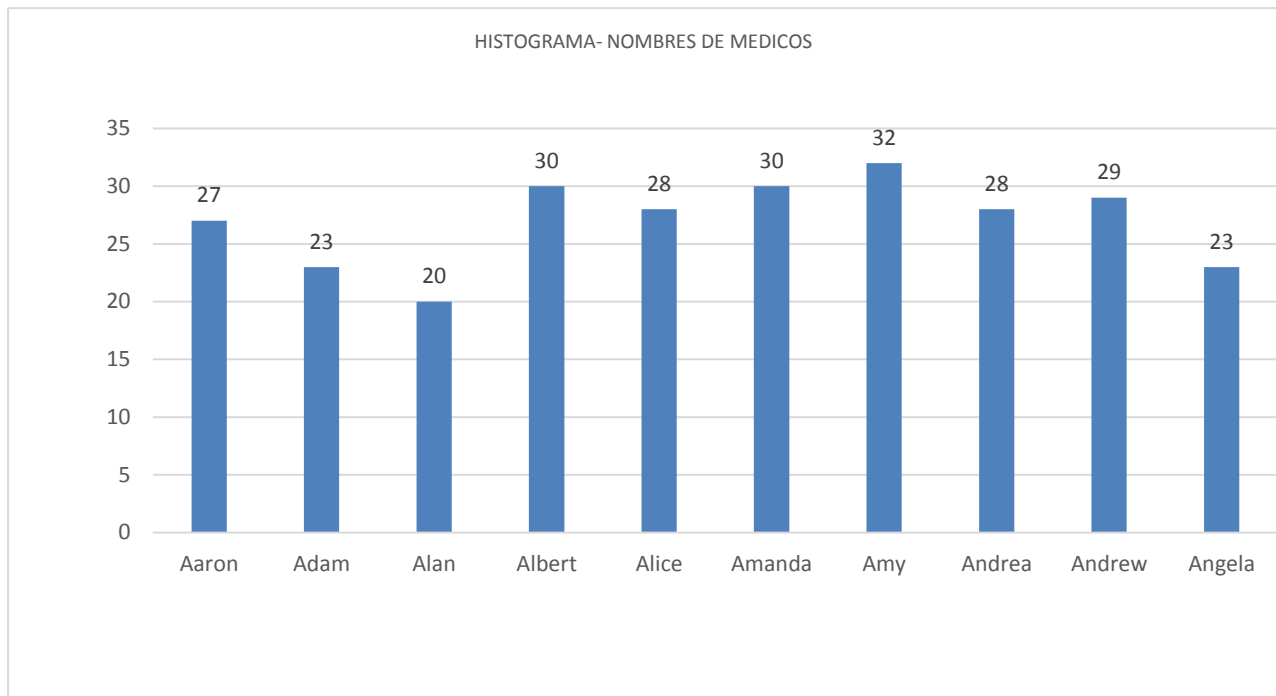
El resultado obtenido es el siguiente: Solo se muestran los 10 primeros registros.

TABLE_NAME	COLUMN_NAME	ENDPOINT_NUMBER	ENDPOINT_ACTUAL_VALUE
MEDICO	NOMBRE	27	Aaron
MEDICO	NOMBRE	50	Adam
MEDICO	NOMBRE	70	Alan
MEDICO	NOMBRE	100	Albert
MEDICO	NOMBRE	128	Alice
MEDICO	NOMBRE	158	Amanda
MEDICO	NOMBRE	190	Amy
MEDICO	NOMBRE	218	Andrea
MEDICO	NOMBRE	247	Andrew
MEDICO	NOMBRE	270	Angela

- Observar el resultado de la columna endpoint\_number. Su valor indica la suma acumulativa del número de ocurrencias de cada valor de la columna nombre. Es decir:
  - Existen 27 registros con nombre Aaron
  - Existen 50-27 = 23 registros con nombre Adam
  - Existen 70-50 = 20 registros con el nombre Alan, y así sucesivamente.

La siguiente consulta confirma los resultados:

SQL> select nombre, count(*)	NOMBRE	COUNT(*)
from medico	Aaron	27
where nombre like 'A%'	Adam	23
group by nombre order by 1	Alan	20
	Albert	30
	Alice	28
	Amanda	30
	Amy	32
	Andrea	28
	Andrew	29
	Angela	23

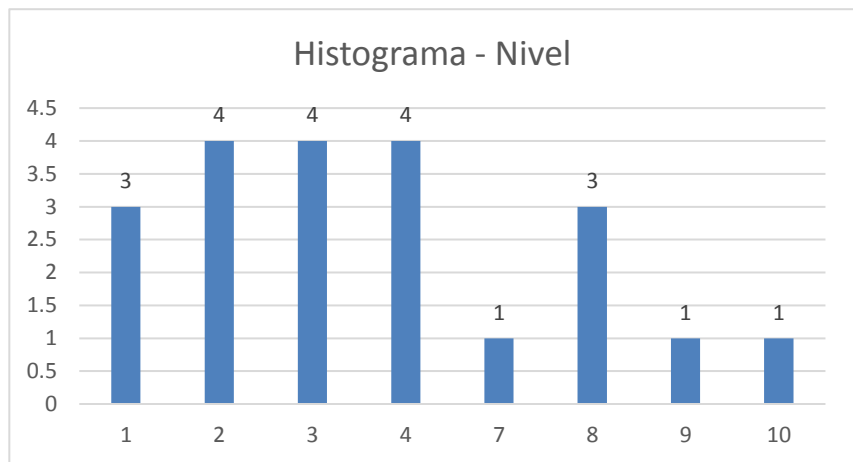
Ejemplo:

Considere la siguiente tabla de datos. Construir un histograma para la columna nivel.

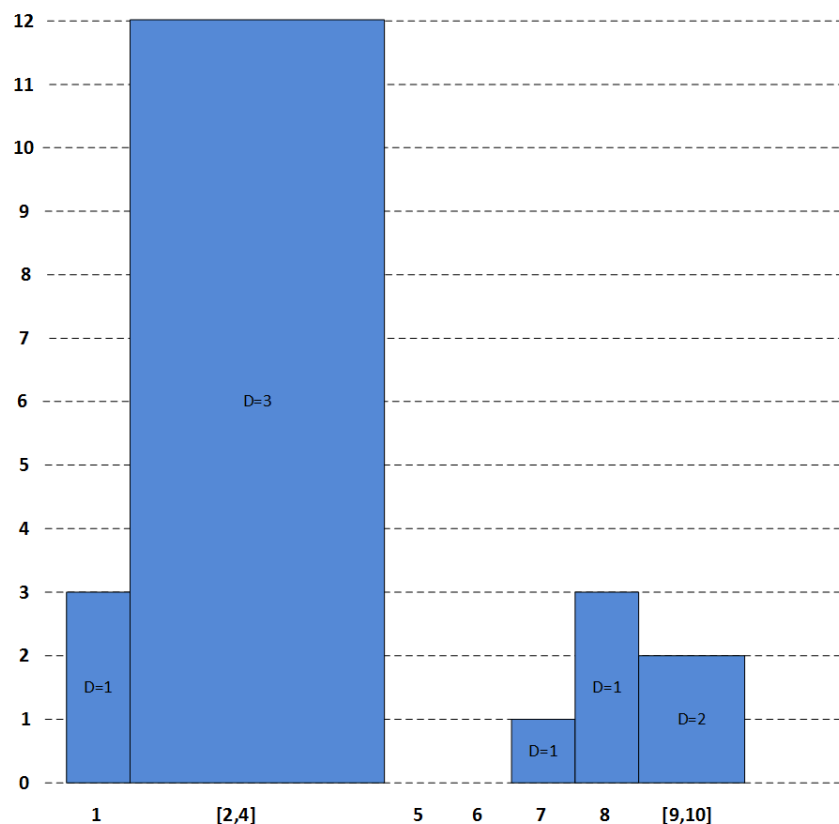
Puesto_id	Nivel
10	7
20	2
30	2
40	2
50	2
60	3
70	3
80	3
90	3
100	4
110	4
120	4
130	4
140	8
150	8
160	8
170	9
180	1
190	1
200	1
201	10

Los datos para construir el histograma son los siguientes:

Nivel	Num. registros
1	3
2	4
3	4
4	4
7	1
8	3
9	1
10	1



- Observar en el histograma anterior la existencia de barras adyacentes con el mismo número de registros (misma cardinalidad). Cuando esta condición ocurre, se forman rangos de valores:



D = Número de valores distintos en un mismo rango. La distribución en cada barra o rango de valores se considera **uniforme**.

Para estimar el número correcto de registros, la BD consulta estos rangos.

- Si no existiera el histograma, la BD determinará que existen 8 valores diferentes de la columna nivel.
- Al existir un total de 21 registros, se obtendría  $21/8 = 2.6 \sim 3$  registros esperados por cada valor del campo nivel.
- Lo anterior resulta incorrecto para ciertos casos, por ejemplo, existe 1 solo registro con nivel 7,9 y 10, y 0 registros para nivel 5 y 6.