

Oracle Performance III

Estadísticas e Índices

¡BIENVENIDOS!

Soy Enrique Gallardo

Soy Administrador de Bases de Datos Oracle

 <https://www.linkedin.com/in/enriquegallardo/>

Sobre mi

- Más de 12 años como DBA
- DBA líder en proyectos internacionales
- OCP en Oracle 11g y 12c
- OCE en RAC & Grid
- Especializado en Alta Disponibilidad
- Trabajando con Oracle desde la versión 8i hasta 20c



Objetivos

Durante este taller veremos dos elementos clave en el rendimiento de sentencias en entornos Oracle, como son los índices y las estadísticas, veremos los principales tipos de índices y revisaremos la configuración en la recolección y actualización de estadísticas de tablas e índices.

Formación recomendada

Se recomienda previamente haber realizado las siguientes formaciones disponibles en la plataforma:

- Arquitectura básica de una BBDD
- **Oracle Performance I - Introduccion al análisis de rendimiento**
- **Oracle Performance II - Herramientas principales de análisis**

De esta manera creemos que se puede sacar más provecho a la formación propuesta.

ÍNDICE

- Introducción a los índices
- Función de los índices
- Tipos de índices
- ¿Qué son las estadísticas?
- Recopilación de estadísticas (Paquete DBMS_STATS)

Introducción a los índices

El funcionamiento es bastante similar al índice de un libro, ya que normalmente se almacenan dos datos, el valor y un puntero (rowid) hacia la tabla.

Los índices requieren espacio y mantenimiento, ya que cada vez que se inserten datos, se producirán operaciones de balanceo sobre los índices.

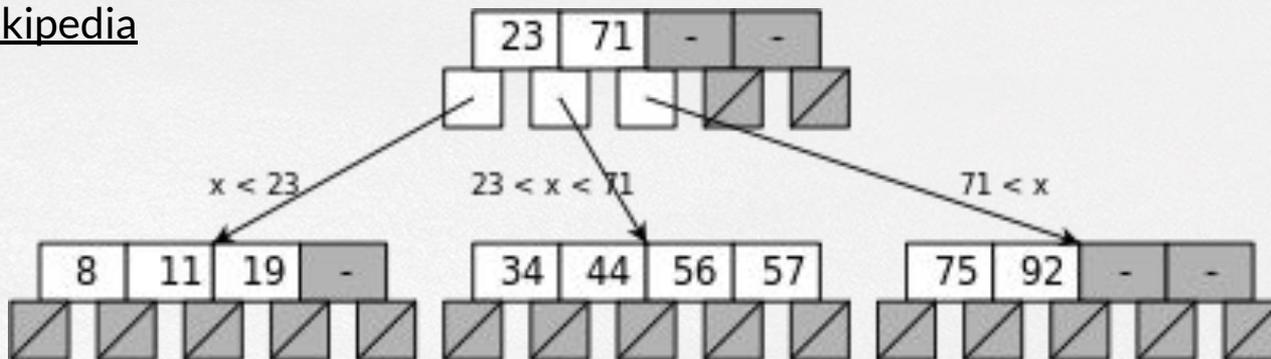
Es una práctica habitual tener tablespaces dedicados al almacenamiento de índices, para separar el crecimiento/mantenimiento de estos de los tablespaces de datos.

B-tree

Primero de todo resulta fundamental revisar qué es un árbol balanceado (B-tree), ya que es la estructura más comumente usada en los índices.

Como vemos, se trata de una estructura de separación por nodos, nodos ramas y nodos hojas que se mantienen balanceados para optimizar accesos.

Wikipedia



Búsqueda con y sin índices

No index
1
3
100
24
5
98
233
77
12
87
300
43
120
89
11
122
67
34
250
99

B-Tree	
<=12	1
	3
	5
	11
	12
24-77	24
	34
	43
	67
	77
87-100	87
	98
	89
	99
	100
120-300	120
	122
	233
	250
	300

Tipos de índices - Bitmap

Casi siempre vamos a tener índices B-tree, de hecho, cuando creamos índices, es el tipo de índice que se usará por defecto, pero es bueno saber que existen otros tipos de índices, por ejemplo los Bitmaps (Mapa de bits), que crean un mapa de bits con los valores y los rowid.

Estos índices pueden mejorar el rendimiento en algunos campos con un número bajo de valores posibles (Género, región, etc) pero no son recomendables para otros con gran número de valores posibles (apellido)

Id	Gen	H	M
1	H	1	0
2	M	0	1
3	M	0	1
4	M	0	1
5	H	1	0
6	M	0	1
7	H	1	0
8	M	0	1
9	H	1	0
10	H	1	0

Tipos de índices

Veremos en una práctica como crear índices de una manera sencilla desde SQLDeveloper.

¿Qué son las estadísticas?

Primero de todo explicaremos qué es el CBO (Cost-Based Optimizer) u optimizador basada en costes.

Anteriormente había otro modelo, el RBO (Rule-Base Optimizer) que establecía una serie de reglas para trazar los planes de ejecución, pero esta herramienta está anticuada y fuera de soporte

El CBO utiliza las estadísticas de tablas e índices para trazar el plan de ejecución más óptimo basado en el coste que tendría, estas estadísticas contienen metadatos (datos de datos) como puede ser la cardinalidad, número de registros, volumen, etc. Por lo que es recomendable tener unas estadísticas actualizadas y representativas de los datos que tenemos para ayudar al CBO a elegir el mejor plan de ejecución.

Información sobre las estadísticas

Para obtener información sobre las estadísticas de los diferentes componentes de la BBDD podemos acceder a las siguientes tablas:

- DBA_TABLES
- DBA_TAB_STATISTICS
- DBA_TAB_COLUMNS
- DBA_TAB_COL_STATISTICS
- DBA_PART_COL_STATISTICS
- DBA_INDEXES
- DBA_IND_STATISTICS

Recopilación de estadísticas (Paquete DBMS_STATS)

El paquete DBMS_STATS nos permite realizar diferentes operaciones con las estadísticas, vamos a realizar varias operaciones de recolección de estadísticas usando los diferentes modos.

El parámetro ESTIMATE_PERCENT puede ser de mucho utilidad en entornos donde la recolección de estadísticas se eternice debido al elevado volumen de datos, ya que estima unas estadísticas basándonos en un porcentaje de los datos, sin llegar al 100% de los mismos, de hecho podemos usar el valor AUTO_SAMPLE_SIZE para dejar al optimizador elegir el valor óptimo.

CASCADE nos permite incluir los índices en la recolección de estadísticas y con DEGREE podemos abrir varios hilos para su ejecución en paralelo

Automatización de recolección de estadísticas

Una práctica habitual es realizar actualizaciones de estadísticas a través de Jobs en procesos nocturnos o de baja actividad.

También es posible que tengamos un modelo mixto, donde parte de las estadísticas se recopilen automáticamente de manera general (usando `GATHER_SCHEMA_STATS` o `GATHER_DATABASE_STATS`) y generemos estadísticas más completas sobre determinados objetos (normalmente los de más volumen o con más modificaciones)

Automatización de recolección de estadísticas

En versiones anteriores a 10g, la recolección de estadísticas era una problemática bastante grande en cuanto al rendimiento.

Vamos a ver los Jobs programados que realizan automáticamente la recolección de estas estadísticas.

Enlaces útiles

Documentación oficial sobre rendimiento

<https://docs.oracle.com/en/database/oracle/oracle-database/19/performance.html>

Hands-on labs

<https://www.oracle.com/virtualization/technologies/vm/downloads/hol-oraclevm-downloads.html>

White Paper sobre rendimiento

<https://www.oracle.com/technetwork/oem/db-mgmt/wp-self-managing-database-3406478.pdf>

Uso de Bind Variables

Como hemos visto, la fase de parseo de una consulta puede ser de las más pesadas, incluso más que la ejecución en si, por ello, para permitir al sistema la re-ejecución de consultas similares es altamente recomendable el uso de Bind Variables para que estas consultas no tengan que ser parseadas muchas veces y sean tratadas como consultas similares, resumiendo, lo que se pretende evitar es un “hard parse” y realizar un “soft parse”, mucho mas óptimo.

Uso de Bind Variables

Veremos con un ejemplo práctico como usar variables y la diferencia que encontramos en cuanto al rendimiento en sentencias simples.