

## Automatic Indexing - Oracle Database 19c

Por Francisco Riccio

### Introducción

Uno de los mayores desafíos de los Administradores de Base de Datos es mantener una base de datos con el mejor desempeño posible, manteniendo las sentencias SQL debidamente optimizadas. Las sentencias SQL se han venido optimizando muchas veces de manera reactiva ante incidencias reportadas, pero hoy la versión de Oracle Database 19c, permite proactivamente mejorar cada sentencia SQL a través de la creación automática de índices sin alguna intervención del Administrador de Base de Datos. Esta nueva funcionalidad aplica para todos los tipos de base de datos especialmente OLTP.

La creación automática de índices trabaja de la siguiente manera en intervalos de 15 minutos:

Fase	Actividades Realizadas
<b>Identificación</b>	Identificación de índices candidatos basado en las sentencias SQL ejecutadas, las cuales han sido almacenadas en un historial. Esta captura incluye planes de ejecución, estadísticas, bind variables relacionados a las sentencias.
<b>Creación</b>	Los índices (solo metadata sin registros) son creados con la opción de invisible con la finalidad que no puedan ser utilizados.
<b>Verificación</b>	A través del optimizador, los índices son materializados y validados contra las sentencias SQL con la finalidad de obtener métricas necesarias.
<b>Decisión</b>	Si los índices logran obtener un beneficio en desempeño, cambian de invisible a visible, en caso contrario, estos son cambiados a unusable y registrados en una "lista negra" de manera que nunca más vuelvan ser considerados en el futuro.
<b>Verificación en Línea</b>	Solo una de las sesiones que ejecutan la sentencia SQL será permitida a utilizar el nuevo índice con la finalidad de validar los potenciales resultados esperados.
<b>Monitoreo</b>	Los índices creados son monitoreados durante todo el ciclo de vida de la base de datos con la finalidad de ser eliminados en el tiempo si dejan de ser usados.

Los índices creados automáticamente soportan:

- Uno o varios campos.
- Basado en funciones.
- Compresión (Low)

## Implementación

A continuación se presentará la implementación y validación de esta nueva funcionalidad.

### A. Preparación de la Prueba

Para validar la funcionalidad se creó una tabla TEST con 10 millones de filas y ejecutando en concurrencia una sentencia SQL cuyo plan de ejecución realiza un FULL SCAN a la tabla.

SQL> desc test;		
Name	Null?	Type
-----	-----	-----
CAMPO1		NUMBER
CAMPO2		VARCHAR2 (30)
<u>Carga de 10 millones de registros:</u>		
begin		
for i in 1..10000000 loop		
insert into test values (i,'T'    to_char(i));		
end loop;		
commit;		
end;		
/		
<u>Sentencia SQL:</u>		
declare		
v varchar(30);		
begin		
for i in 1..10000000 loop		
select campo2 into v from test where campo1=i;		
end loop;		
end;		
/		

### B. Configuración de Automatic Indexing

Habilitar la Funcionalidad:

```
execute DBMS_AUTO_INDEX.CONFIGURE('AUTO_INDEX_MODE','IMPLEMENT|REPORT ONLY|OFF');
```

<pre>[oracle@srvbd ~]\$ sqlplus friccio/oracle@PDB_PRD  SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 21 22:20:48 2019 Version 19.2.0.0.0  Copyright (c) 1982, 2018, Oracle. All rights reserved.  Last Successful login time: Thu Mar 21 2019 22:20:32 -05:00  Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production Version 19.2.0.0.0  SQL&gt; execute DBMS_AUTO_INDEX.CONFIGURE('AUTO_INDEX_MODE','IMPLEMENT');  PL/SQL procedure successfully completed.</pre>
---

El valor del parámetro REPORT ONLY ejecuta todas las fases mencionadas pero no permite que los índices creados sean utilizados por las sentencias SQL.

Configurar el Tablespace utilizado para crear los Índices Automáticos:

*execute DBMS\_AUTO\_INDEX.CONFIGURE('AUTO\_INDEX\_DEFAULT\_TABLESPACE','<NOMBRE DEL TABLESPACE>');*

```
SQL> create tablespace TBS_IDX_AUTO datafile '/u02/oradata/PRD/PDB_PRD/tbs_idx_auto_01.dbf' size 100M;
Tablespace created.
SQL> execute DBMS_AUTO_INDEX.CONFIGURE('AUTO_INDEX_DEFAULT_TABLESPACE','TBS_IDX_AUTO');
PL/SQL procedure successfully completed.
```

Existen otros parámetros que podemos configurar como parte del paquete DBMS\_AUTO\_INDEX:

Parámetro	Objetivo
AUTO_INDEX_SCHEMA,<NOMBRE ESQUEMA>,FALSE TRUE	TRUE: Se considera la creación automática de índices en el esquema.
AUTO_INDEX_SPACE_BUDGET,%	Especifica el porcentaje de uso de espacio que puede ser utilizado para la creación de Índices Automáticos. Por defecto es: 50%.
AUTO_INDEX_RETENTION_FOR_AUTO,n días	Mantiene por un período de n días los índices en estado UNUSABLE. Por defecto es: 373 días
AUTO_INDEX_RETENTION_FOR_MANUAL,n días	Mantiene por un período de n días los índices que no han sido utilizados y fueron creados por el DBA manualmente. Por defecto Oracle no elimina los índices creados por el DBA que no tengan utilización.
AUTO_INDEX_REPORT_RETENTION,n días	Permite almacenar por una cantidad de días los análisis y beneficios que tuvieron los índices analizados durante la fase de validación. Por defecto, se almacena por 31 días.

Si se desea revisar toda la configuración, se puede consultar la siguiente vista:

*DBA\_AUTO\_INDEX\_CONFIG*

```
SQL> column PARAMETER_NAME format a50
SQL> column PARAMETER_VALUE format a50
SQL> select PARAMETER_NAME,PARAMETER_VALUE from DBA_AUTO_INDEX_CONFIG;
```

PARAMETER_NAME	PARAMETER_VALUE
AUTO_INDEX_DEFAULT_TABLESPACE	TBS_IDX AUTO
AUTO_INDEX_MODE	IMPLEMENT
AUTO_INDEX_REPORT_RETENTION	31
AUTO_INDEX_RETENTION_FOR_AUTO	373
AUTO_INDEX_RETENTION_FOR_MANUAL	
AUTO_INDEX_SCHEMA	
AUTO_INDEX_SPACE_BUDGET	50

7 rows selected.

### C. Revisión de los Índices Sugeridos

Cada 15 minutos se realiza la revisión de sentencias SQL que puedan tomar algún beneficio desempeño mediante índices, dichas tareas de revisión pueden ser consultadas a través de la siguiente vista: *DBA\_AUTO\_INDEX\_EXECUTIONS*

```
SQL> select EXECUTION_NAME,EXECUTION_START,EXECUTION_END,STATUS
from DBA_AUTO_INDEX_EXECUTIONS;
```

EXECUTION_NAME	EXECUTION_START	EXECUTION_END	STATUS
SYS_AI_2019-03-21/22:42:18	21-MAR-19	21-MAR-19	COMPLETED
SYS_AI_2019-03-21/22:57:23	21-MAR-19	21-MAR-19	COMPLETED
SYS_AI_2019-03-21/23:12:30	21-MAR-19	21-MAR-19	COMPLETED
SYS_AI_2019-03-21/23:27:35	21-MAR-19	21-MAR-19	COMPLETED
SYS_AI_2019-03-21/23:42:41	21-MAR-19		EXECUTING

Para revisar el detalle del análisis de una tarea específica, se consulta la vista *DBA\_AUTO\_INDEX\_STATISTICS*. En este caso se revisará la última del reporte previo: *SYS\_AI\_2019-03-21/23:42:41*

```
SQL> select * from DBA_AUTO_INDEX_STATISTICS
where EXECUTION_NAME='SYS_AI_2019-03-21/23:42:41';
```

EXECUTION_NAME	STAT_NAME	VALUE
SYS_AI_2019-03-21/23:42:41	Index candidates	0
SYS_AI_2019-03-21/23:42:41	Indexes created (visible)	1
SYS_AI_2019-03-21/23:42:41	Indexes created (invisible)	0
SYS_AI_2019-03-21/23:42:41	Indexes dropped	0
SYS_AI_2019-03-21/23:42:41	Space used in bytes	201326592
SYS_AI_2019-03-21/23:42:41	Space reclaimed in bytes	0
SYS_AI_2019-03-21/23:42:41	SQL statements verified	1
SYS_AI_2019-03-21/23:42:41	SQL statements improved	1
SYS_AI_2019-03-21/23:42:41	SQL statements managed by SPM	0
SYS_AI_2019-03-21/23:42:41	SQL plan baselines created	0
SYS_AI_2019-03-21/23:42:41	Improvement percentage	99.7

11 rows selected.

Se puede revisar las acciones ejecutadas por la base datos sobre los índices creados automáticamente a través de la vista: *DBA\_AUTO\_INDEX\_IND\_ACTIONS*

```
SQL> select INDEX_OWNER, INDEX_NAME, TABLE_OWNER, TABLE_NAME, COMMAND
from DBA_AUTO_INDEX_IND_ACTIONS
order by START_TIME;
```

INDEX_OWNER	INDEX_NAME	TABLE_OWNER	TABLE_NAME	COMMAND
FRICCIO	SYS_AI_ayv3vppgrvmw8	FRICCIO	TEST	CREATE INDEX
FRICCIO	SYS_AI_ayv3vppgrvmw8	FRICCIO	TEST	REBUILD INDEX
FRICCIO	SYS_AI_ayv3vppgrvmw8	FRICCIO	TEST	REBUILD INDEX
FRICCIO	SYS_AI_ayv3vppgrvmw8	FRICCIO	TEST	REBUILD INDEX
FRICCIO	SYS_AI_ayv3vppgrvmw8	FRICCIO	TEST	ALTER INDEX VISIBLE

Las verificaciones realizadas por la base de datos sobre un índice creado automáticamente se pueden visualizar en la siguiente vista: `DBA_AUTO_INDEX_VERIFICATIONS`

```
SQL> select SQL_ID, ORIGINAL_PLAN_HASH_VALUE, AUTO_INDEX_PLAN_HASH_VALUE from DBA_AUTO_INDEX_VERIFICATIONS;
```

SQL_ID	ORIGINAL_PLAN_HASH_VALUE	AUTO_INDEX_PLAN_HASH_VALUE
cxg3jf34hgts	1357081020	480040

```
SQL> select SQL_ID, ORIGINAL_BUFFER_GETS, AUTO_INDEX_BUFFER_GETS, STATUS from DBA_AUTO_INDEX_VERIFICATIONS;
```

SQL_ID	ORIGINAL_BUFFER_GETS	AUTO_INDEX_BUFFER_GETS	STATUS
cxg3jf34hgts	27432.1425	83	IMPROVED

En este ejemplo se puede apreciar que el nuevo índice reduce los buffer gets de: 27,432.1425 a 83.

Acorde al escenario presentado, se puede apreciar que se ha creado un índice en el tablespace `TBS_IDX_AUTO`:

```
SQL> column SEGMENT_NAME format a45
SQL> select segment_name, segment_type
from dba_segments
where tablespace_name = 'TBS_IDX_AUTO';
```

SEGMENT_NAME	SEGMENT_TYPE
SYS_AI_ayv3vppgrvmw8	INDEX

```
SQL> select auto, count(*) from dba_indexes group by auto;
```

AUT	COUNT (*)
YES	1
NO	2673

#### D. Obtención de Reportes de Automatic Indexing

Los reportes pueden ser obtenidos a través de los siguientes métodos:

- `DBMS_AUTO_INDEX.REPORT_ACTIVITY`
- `DBMS_AUTO_INDEX.REPORT_LAST_ACTIVITY`

El formato entregado puede ser: `TEXT`, `XML` y `HTML`.

Ejemplo:

Para este caso se obtendrá el reporte de los índices creados automáticamente en el intervalo de hace 3 días a la fecha actual menos 30 minutos.

```

SQL> set long 300000
SQL> select DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','ALL','ALL')
from dual;

DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','A
-----
GENERAL INFORMATION
-----
Activity start          : 19-MAR-2019 00:26:11
Activity end           : 21-MAR-2019 23:56:11
Executions completed   : 5
Executions interrupted : 0
Executions with fatal error : 0
-----

SUMMARY (AUTO INDEXES)
-----
DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','A
-----
Index candidates          : 1
Indexes created (visible / invisible) : 1 (1 / 0)
Space used (visible / invisible) : 201.33 MB (201.33 MB / 0 B)
Indexes dropped           : 0
SQL statements verified   : 1
SQL statements improved (improvement factor) : 1 (330.5x)
SQL plan baselines created : 0
Overall improvement factor : 330.5x
-----

SUMMARY (MANUAL INDEXES)
-----
DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','A
-----
Unused indexes          : 0
Space used              : 0 B
Unusable indexes       : 0
-----

INDEX DETAILS
-----
1. The following indexes were created:
*: invisible
-----

DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','A
-----
| Owner | Table | Index | Key | Type | Properties |
-----
| FRICCIO | TEST | SYS_AI_ayv3vppgrvmw8 | CAMPO1 | B-TREE | NONE |
-----

```

```

VERIFICATION DETAILS
-----
1. The performance of the following statements improved:
-----

DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','A
-----
Parsing Schema Name : FRICCIO

SQL ID : cxg3jf34hgfts

SQL Text : SELECT CAMPO2 FROM TEST WHERE CAMPO1=:B1

Improvement Factor : 330.5x

PLANS SECTION
-----

DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','A
-----

- Original
-----
Plan Hash Value : 1357081020

| Id | Operation | Name | Rows | Bytes | Cost | Time |
| 0 | SELECT STATEMENT | | | | 7513 | |
| 1 | TABLE ACCESS FULL | TEST | 302 | 9060 | 7513 | 00:00:01 |
-----

DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','A
-----

Notes
-----
- Dynamic sampling used for this statement ( level = 2 )

- With Auto Indexes
-----
Plan Hash Value : 480040

DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','A
-----

| Id | Operation | Name | Rows | Bytes | Cost | Time |
| 0 | SELECT STATEMENT | | 1 | | 4 | 00:00:01 |
| 1 | TABLE ACCESS BY INDEX ROWID BATCHED | TEST | 1 | | 4 | 00:00:01 |
| * 2 | INDEX RANGE SCAN | SYS_AI_ayv3vppgrvmw8 | 1 | | 4 | 00:00:01 |
-----

DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','A
-----

| 3 | 00:00:01 |
-----

Predicate Information (identified by operation id):
-----
* 2 - access("CAMPO1"=:B1)

Notes
-----

DBMS_AUTO_INDEX.REPORT_ACTIVITY(SYSTIMESTAMP-3,SYSTIMESTAMP-1/24/60*30,'TEXT','A
-----
- Dynamic sampling used for this statement ( level = 11 )

```

Se puede apreciar que la consulta con el SQLID: cxg3jf34hgfts, tenía un costo de 7,513 y con la implementación automática del índice tiene un costo actual de: 4, donde esto generó un beneficio de 330.5x.

Asimismo consultando el plan de ejecución de la sentencia SQL tenemos el mismo resultado:

```
SQL> set linesize 400
SQL> set autotrace on
SQL> select campo2 from test where campo1=250000;
```

CAMPO2

-----  
T250000

Execution Plan

-----  
Plan hash value: 480040

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	30	4 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID BATCHED	TEST	1	30	4 (0)	00:00:01
* 2	INDEX RANGE SCAN	SYS_AI_ayv3vppgrrvmw8	1		3 (0)	00:00:01

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

-----  
2 - access("CAMPO1"=250000)

Note

-----  
- dynamic statistics used: dynamic sampling (level=2)

Statistics

-----  
0 recursive calls  
0 db block gets  
5 consistent gets  
0 physical reads  
0 redo size  
560 bytes sent via SQL\*Net to client  
405 bytes received via SQL\*Net from client  
2 SQL\*Net roundtrips to/from client  
0 sorts (memory)  
0 sorts (disk)  
1 rows processed

## Conclusión

Esta nueva funcionalidad que nos provee Oracle Database 19c permite liberar de horas de trabajo al DBA de una manera proactiva para mantener la base de datos con un tiempo de respuesta eficiente, manteniendo todos los niveles de control que permitan no afectar el rendimiento actual.

## Publicado por:

**Francisco Riccio**, actualmente se desempeña como Arquitecto de Soluciones en Oracle Perú y es instructor de cursos oficiales de certificación Oracle. Es un Oracle Certified Professional en productos de Oracle Application, Base de Datos, Cloud & Virtualización.