

Iteración 3

Juan Cañizarez, Daniel Del Castillo

ISIS 2304- Sistemas Transaccionales

Universidad de los Andes, Bogotá, Colombia

{je.canizarez, d.delcastillo}@uniandes.edu.co

Fecha de presentación: 4 de diciembre de 2019

Introducción

En el presente informe se mostrarán los aspectos más importantes del caso EpsAndes. Primero, se presentarán los índices seleccionados en la base de datos con el fin de mejorar la eficiencia de las consultas. Seguidamente, se mostrarán las consultas con sus planes de ejecución.

Índices

Los índices que crea Oracle por defecto son índices primarios sobre cada una de las tablas. A su vez, Oracle también crea índices sobre las columnas las cuales sus valores son únicos ya que la selectividad de estas columnas es muy baja. Estos índices mejoran a la eficiencia de las queries ya que se pueden usar en el plan de ejecución para mejorar con la efectividad de las consultas.

INDEX_NAME	INDEX_TYPE
1 AFILIADO_PK	NORMAL
2 CITAS_PK	NORMAL
3 CMP_PK	NORMAL
4 CMP_UNIQUE	NORMAL
5 CONSULTAEM_PK	NORMAL
6 EPS_PK	NORMAL
7 FECHA_FECHA_UNIQUE	NORMAL
8 FECHA_PK	NORMAL
9 IPS_NOMBRE_UNIQUE	NORMAL
10 IPS_PK	NORMAL
11 MEDICO_DOCUMENTO_UNIQUE	NORMAL
12 MEDICO_NUMREGISTROMED_UNIQUE	NORMAL
13 MEDICO_PK	NORMAL
14 ORDENES_IDAFILIADO_UNIQUE	NORMAL
15 ORDENES_IDMEDICO_UNIQUE	NORMAL
16 ORDENES_PK	NORMAL
17 PRIORIDAD_PK	NORMAL
18 PRIORIDAD_PRIORIDAD_UNIQUE	NORMAL
19 PS_PK_UNIQUE	NORMAL

20	RECEPCIONISTA_PK	NORMAL
21	RECETA_PK	NORMAL
22	ROL_PK	NORMAL
23	ROL_ROL_UNIQUE	NORMAL
24	SA_PK	NORMAL
25	SERVICIOSCAMPANA_PK	NORMAL
26	SERVICIO_PK	NORMAL
27	SM_IDMEDICO_UNIQUE	NORMAL
28	SM_IDSERVICIO_UNIQUE	NORMAL
29	SM_PK	NORMAL
30	TD_PK	NORMAL
31	TD_TIPODOCUMENTO_UNIQUE	NORMAL
32	TIPOSERVICIO_PK	NORMAL
33	TIPOSERVICIO_TIPO_UNIQUE	NORMAL
34	TRABAJAN_IDEPS_UNIQUE	NORMAL
35	TRABAJAN_IDIPS_UNIQUE	NORMAL
36	TRABAJAN_PK	NORMAL
37	USUARIO_DOCUMENTO_UNIQUE	NORMAL
38	USUARIO_LOGIN_UNIQUE	NORMAL
-		
39	USUARIO_PK	NORMAL

Figura 1. Índices creados por Oracle.

Justificación de creación de índices

Índice sobre fechaAsistida de ServiciosAfiliado.

Para nuestra base de datos, vale la pena crear un índice sobre la columna fecha asistida ya que la selectividad de esta está debajo del 25%, además, se usa en la gran mayoría de requerimientos de consultas de nuestro proyecto. La mejor estructura para almacenar este índice sería un árbol B+ ya que es ideal para consultas en rango y generalmente, las consultas van a incluir un rango de fechas.

Índice sobre TipoServicio en ServiciosAfiliado.

Es importante tener un índice sobre tipo de servicio debido a su cardinalidad y el número de veces que este es usado en las consultas de las tres iteraciones. La estructura para almacenar este índice es un Hash ya que es mejor en tiempo para consultas de valores específicos y generalmente siempre se va a consultar un valor específico en una consulta.

Índice compuesto fechaAsistida, TipoServicio y IdAfiliado en ServiciosAfiliado

Es importante tener un índice sobre estos valores de la tabla ya que la selectividad conjunta de estos tres atributos es baja y en adición, se mejora el tiempo de efectividad para el requerimiento de consulta 5 el cual incluye estas tres columnas y es una consulta bastante común en un contexto como el presentado en el proyecto.

Índice sobre TipoServicio en Citas.

Tener un índice sobre citas mejora la efectividad tanto para registrar campañas como para deshabilitar servicios. También, una de las razones para crear un índice sobre este atributo es su selectividad la cual es 14%. La mejor estructura para almacenar un índice es un hash ya que se requieren de valores específicos y no rangos.

Índice sobre IPS en citas.

Tener un índice sobre las IPS ya que la selectividad de la columna es baja y para los nuevos requerimientos de consulta presentados en la iteración 3 además de que los presentados en la primera iteración es un atributo que se tiene en cuenta para las consultas 1,9,10 y 11. También una de las razones para crear un índice sobre este atributo es su baja selectividad. La mejor estructura para almacenar este índice es un Hash ya que no es posible que se presenten consultas en rangos si no consultas exactas sobre dicho atributo.

Requerimientos de consulta

RC12:

```
WITH oth_no_esp as (select idafiliado
from citas
inner join SERVICIO sds on CITAS.IDSERVICIO = SDS.ID
and sds.TIPOSERVICIO != 7),

ESPEC AS (
select idafiliado, 1 siempre_espec,count(distinct sds.TIPOSERVICIO)cuenta
from citas
inner join SERVICIO sds on CITAS.IDSERVICIO = SDS.ID
and sds.TIPOSERVICIO = 7
where citas.idafiliado not in (select idafiliado from oth_no_esp)
group by idafiliado
),

cit_mes as( -- Cuenta el número de meses distintos en los que ha hecho una cita
```

```

select idafiliado, count(distinct
to_char(TO_DATE(FECHAASISTIDA,'YYYY/mm/DD'),'mm/yy')) as count_citas
from SERVICIOSAFILIADO
group by idafiliado
order by count_citas desc),

```

```

min_fe as( -- Cuenta el número de meses distintos en los que ha hecho una cita
select idafiliado, min(TO_DATE(fechaASISTIDA,'yyyy/mm/dd')) as first_cita
from SERVICIOSAFILIADO
group by idafiliado
),

```

```

month_fd as ( -- Cuenta cuántos meses han transcurrido desde su primera cita
SELECT idafiliado, MONTHS_BETWEEN
(TO_DATE('12-2019','MM-YYYY'),
TO_DATE(to_char(first_cita,'mm/yyyy') ,'MM-YYYY')
) meses_primera_cita, first_cita
FROM min_fe ),

```

```

final_freq as (
select cit_mes.idafiliado, first_cita, count_citas, meses_primera_cita
from month_fd
inner join cit_mes
on cit_mes.idafiliado = month_fd.idafiliado
where count_citas = meses_primera_cita),

```

```

citas2 as(SELECT COUNT(*) as citasUsuario, idafiliado FROM SERVICIOSAFILIADO
GROUP BY idafiliado order by count(*) desc),

```

hospital as (SELECT COUNT(*) as cuentaHospitalizaciones, r.idafiliado FROM
ORDENES r WHERE r.TIPOSERVICIO= 7
GROUP BY r.idafiliado),

c2 as(SELECT c.idafiliado, 1 siempreHosp, c.citasUsuario as citas,
h.cuentaHospitalizaciones as hospitalizaciones FROM citas2 c,hospital h WHERE
c.idafiliado=h.idafiliado AND c.citasUSUARIO =h.cuentaHospitalizaciones)

select usuario.nombre, afiliado.DOCUMENTO, case when siempre_espec = 1 then
'Siempre especializado'

when count_citas is not null then 'Pide cita todos los meses' when siempreHosp=1 then
'Siempre hospitalizado'

end razon

, case when count_citas is not null then count_citas else 0 end CitasSolicitadas ,case when
cuenta is not null then cuenta else 0 end ServiciosDistintosSolicitados, case when
hospitalizaciones is not null then hospitalizaciones else 0 end hospitalizaciones

from usuario left outer join afiliado on usuario.documento=afiliado.documento

left outer join final_freq

on final_freq.idafiliado = afiliado.documento

left outer join espec

on espec.idafiliado = afiliado.documento left outer join c2 on c2.idafiliado=espec.idafiliado

where siempre_espec = 1 or count_citas is not null or siempreHosp=1;

Plan de ejecución

SQL 0,362 segundos				
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1 9
FILTER Filter Predicates NOT EXISTS (SELECT 0 FROM SERVICIO SERVICIO,CITAS CITAS WHERE LNNVL(CITAS.IDAFILIADO<>:B1) AND CITAS.IDSERVICIO=SERVICIO.ID AND SERVICIO.TIPOSERVICIO<>7)				
TABLE ACCESS NESTED LOOPS	CITAS	FULL		38 3
TABLE ACCESS Filter Predicates LNNVL(CITAS.IDAFILIADO<>:B1)	CITAS	SEMI		1 3
TABLE ACCESS Filter Predicates SERVICIO.TIPOSERVICIO<>7	SERVICIO	FULL		1 2
TABLE ACCESS INDEX Access Predicates CITAS.IDSERVICIO=SERVICIO.ID	SERVICIO_PK	BY INDEX ROWID		8 1
		UNIQUE SCAN		1 0

