

Iteración 4

Juan S. Ramirez Beltran

Andrés S. Triana Arjona

Universidad de los Andes, Bogotá, Colombia

j.ramirez@uniandes.edu.co

a.triana@uniandes.edu.co

Fecha de presentación: Mayo 30 de 2021

Tabla de contenido

1	Impacto de la introducción de los nuevos requerimientos	1
	Requerimiento funcional de consulta 10 y 11:.....	1
	Requerimiento funcional de consulta 12:.....	1
	Requerimiento funcional de consulta 13:.....	1
2	Diseño físico	2
2.1	Requerimiento funcional de consulta 10:	2
2.2	Requerimiento funcional de consulta 11:	2
3	Índices creados para cada requerimiento de consulta.....	3
3.1	Requerimiento funcional de consulta 10:	3
3.2	Requerimiento funcional de consulta 11:	5
3.3	Requerimiento funcional de consulta 12:	6
3.4	Casos en los que se usaron los índices creados de manera propia.....	7

1 Impacto de la introducción de los nuevos requerimientos

En esta iteración era necesario hacer 4 requerimientos de consulta nuevos. Aquí los enlistamos para mostrar los cambios:

Requerimiento funcional de consulta 10 y 11: En este requerimiento no se necesitó ninguna información ni modificación al original, esto se debe a que ya estábamos preparados y cada persona guarda su propio estado de vacunación, lo único que se necesita es pedir ese estado y revisar que sea vacunado o no vacunado. Se tuvo que crear una sentencia dinámica, que iba más allá de los diseños originales y utiliza Strings para construirse

Requerimiento funcional de consulta 12: En este requerimiento no se necesitó ninguna información ni modificación al original, pero se requería en cierta manera un código que nunca se utilizó en SQL que era el que tenía que ver con todo lo que es de creación de las semanas del año para poder distribuir las fechas de las citas que ya se tenían

Requerimiento funcional de consulta 13: En este requerimiento si se necesitaban muchas modificaciones a las clases. En primer lugar, no teníamos nada que registrara que registrara información de los lotes, por ningún lado teníamos las fechas en las que llegaban los

lotes. También teníamos problemas con la asistencia a las citas, pues no teníamos nada para modelar si una cita fue cancelada o no, lo cual requeriría de un parámetro nuevo en cita.

2 Diseño físico

2.1 Requerimiento funcional de consulta 10:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				106
SORT				106
NESTED LOOPS				106
NESTED LOOPS				506
TABLE ACCESS	CITA	FULL		506
Filter Predicates				
AND				
CTA.FECHA >= TO_DATE(' 2021-06-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
CTA.FECHA <= TO_DATE(' 2021-12-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
INDEX	CEDULA	UNIQUE SCAN	1	
Access Predicates				
CD.CEDULA = CTA.CIUDADANO				
TABLE ACCESS	CIUDADANO	BY INDEX ROWID	1	
Filter Predicates				
AND				
CD.REGION = 'Andina'				
CD.ESTADO_VACUNACION = 'Vacunado'				
CD.OFICINA_REGIONAL_ASIGNADA = 1				

Figura 1. Req 10 imagen índices

Como se puede evidenciar, en caso del requerimiento 10, solo se utiliza un índice por default, este índice utilizado es el de la cedula, el cual es una Pk de ciudadano, que es precisamente la tabla en la cual se hace la sentencia. El índice que se utiliza de cédula es un índice de tipo primario y es de tipo único.

2.2 Requerimiento funcional de consulta 11:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1
SORT		ORDER BY		1
NESTED LOOPS				506
NESTED LOOPS				506
TABLE ACCESS	CITA	FULL		506
Filter Predicates				
AND				
CTA.FECHA >= TO_DATE(' 2021-06-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
CTA.FECHA <= TO_DATE(' 2021-12-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				
INDEX	CEDULA	UNIQUE SCAN	1	
Access Predicates				
CD.CEDULA = CTA.CIUDADANO				
TABLE ACCESS	CIUDADANO	BY INDEX ROWID	1	
Filter Predicates				
AND				
CD.REGION = 'Andina'				
CD.ESTADO_VACUNACION = 'No Vacunado'				
CD.OFICINA_REGIONAL_ASIGNADA = 1				

Figura 2. Req 11 imagen índices

En el caso del requerimiento 11, era extremadamente similar al requerimiento 10, todo sigue igual y se usa el mismo índice de cedula, solo que ahora el estado de vacunación que se busca es “No vacunado”.

3 Índices creados para cada requerimiento de consulta

3.1 Requerimiento funcional de consulta 10:

Sentencia SQL:

Caso en que lo pida el administrador del plan

```
--Metodo req 10 adminsitrador plan de vacunacion

SELECT  cd.cedula,    cd.nombre_completo,    cd.estado_vacunacion,
cd.region,          cd.desea_ser_vacunado,    cd.plan_de_vacunacion,
cd.punto_vacunacion, cd.oficina_regional_asignada

FROM  ciudadano  cd  inner  join  cita  cta  on  cd.cedula  =
cta.ciudadano

INNER  JOIN  list_condiciones_ciudadano  cond  on  cd.cedula  =
cond.ciudadano

WHERE  fecha  BETWEEN  TO_DATE('2021/06/08', 'yyyy/mm/dd')  and
TO_DATE('2021/12/08', 'yyyy/mm/dd')  and  estado_vacunacion  =
'Vacunado'  and  region  ='Andina'

ORDER BY punto_vacunacion;
```

Caso en que lo pida el administrador de EPS

```
--Metodo req 10 adminsitrador de eps

select  cd.cedula,    cd.nombre_completo,    cd.estado_vacunacion,
cd.region,          cd.desea_ser_vacunado,    cd.plan_de_vacunacion,
cd.punto_vacunacion, cd.oficina_regional_asignada

from  ciudadano  cd  inner  join  cita  cta  on  cd.cedula  =
cta.ciudadano

where  region  ='Andina'  and  estado_vacunacion  =  'Vacunado'  and
oficina_regional_asignada  =1  and  fecha  between
TO_DATE('2021/06/08', 'yyyy/mm/dd')  and  TO_DATE('2021/12/08',
'yyyy/mm/dd')

order by punto_vacunacion;
```

Distribución de datos con respecto a parámetros:

Caso 1

Primero hicimos la prueba solo con índices originales, los parámetros utilizados fueron los siguientes

	CEDULA	NOMBRE_COMPLETO	ESTADO_VACUNACION	REGION	DESEA_SER_VACUNADO	PLAN_DE_VACUNACION	PUNTO_VACUNACION	OFICINA
1	1001175754	PABLO DANIEL PEREIRA	Vacunado	Andina	0	1	1	
2	1001174854	NELSON EDUARDO BOZZOLASCO	Vacunado	Andina	0	1	134	
3	1001171704	DENISE HAIFA COSTA	Vacunado	Andina	0	1	199	
4	1001177630	GONZALO COSTA	Vacunado	Andina	0	1	199	

Figura 3. Req 10 Caso 1

Parámetros:

```
where region = 'Andina' and estado_vacunacion = 'Vacunado' and
oficina_regional_asignada = 1 and fecha between
TO_DATE('2021/06/08', 'yyyy/mm/dd') and TO_DATE('2021/12/08',
'yyyy/mm/dd')
```

```
order by punto_vacunacion;
```

Caso 2:

En el segundo caso, utilizamos parámetros más amplios que en el anterior y estos fueron los resultados

	CEDULA	NOMBRE_COMPLETO	ESTADO_VACUNACION	REGION	DESEA_SER_VACUNADO	PLAN_DE_VACUNACION	PUNTO_VACUNACION	OFICINA
1	1001175754	PABLO DANIEL PEREIRA	Vacunado	Andina	0	1	1	
2	1001175663	DENISE HAIFA CASTROMAN	Vacunado	Andina	1	1	11	
3	1001174854	NELSON EDUARDO BOZZOLASCO	Vacunado	Andina	0	1	134	
4	1001177630	GONZALO COSTA	Vacunado	Andina	0	1	199	
5	1001171704	DENISE HAIFA COSTA	Vacunado	Andina	0	1	199	
6	1001172594	JUAN MARTIN SILVERA	Vacunado	Andina	0	1	199	
7	1001169928	JOSE ANTONIO DE LA IGLESIA	Vacunado	Andina	1	1	209	
8	1001169929	EFRAIN ANDRES ANDRADE	Vacunado	Andina	0	1	209	
9	1001174146	JOSE LUIS DE BELLIS	Vacunado	Andina	1	1	209	
10	1001174836	DANIEL COELLO	Vacunado	Andina	0	1	296	

Figura 4. Req 10 Caso 2

Parámetros:

```
where region = 'Andina' and estado_vacunacion = 'Vacunado' and
oficina_regional_asignada = 1 and fecha between
TO_DATE('2021/01/01', 'yyyy/mm/dd') and TO_DATE('2021/12/31',
'yyyy/mm/dd')
```

```
order by punto_vacunacion;
```

3.2 Requerimiento funcional de consulta 11:

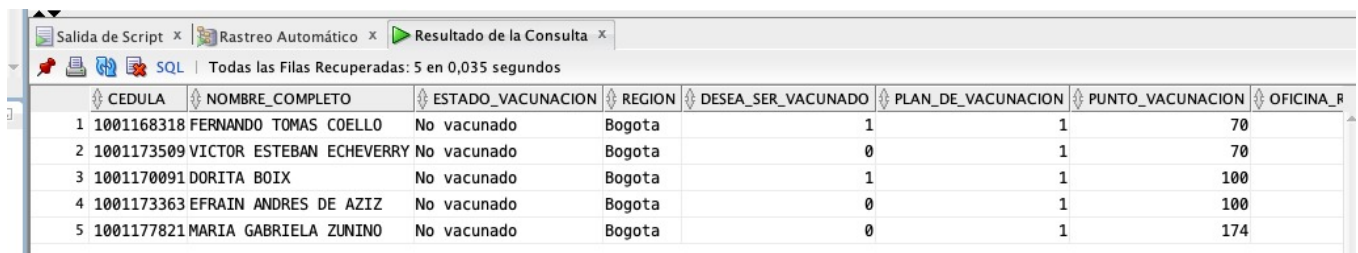
Sentencia SQL:

Caso en que lo pida el administrador del plan

```
select *  
  
from ciudadano  
  
where region ='Andina' and estado_vacunacion = 'No Vacunado'  
  
order by punto_vacunacion;
```

Caso 1

Caso solo con índices originales en el que se usan parámetros más amplios



Salida de Script x Rastreo Automático x Resultado de la Consulta x

Todas las Filas Recuperadas: 5 en 0,035 segundos

	CEDULA	NOMBRE_COMPLETO	ESTADO_VACUNACION	REGION	DESEA_SER_VACUNADO	PLAN_DE_VACUNACION	PUNTO_VACUNACION	OFICINA_R
1	1001168318	FERNANDO TOMAS COELLO	No vacunado	Bogota	1	1	70	
2	1001173509	VICTOR ESTEBAN ECHEVERRY	No vacunado	Bogota	0	1	70	
3	1001170091	DORITA BOIX	No vacunado	Bogota	1	1	100	
4	1001173363	EFRAIN ANDRES DE AZIZ	No vacunado	Bogota	0	1	100	
5	1001177821	MARIA GABRIELA ZUNINO	No vacunado	Bogota	0	1	174	

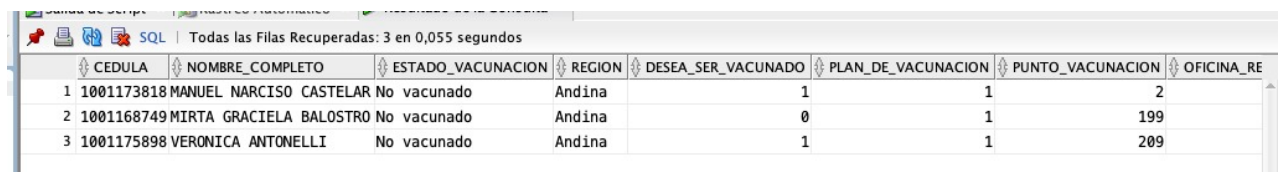
Figura 5. Req 11 Caso 1

Parámetros:

```
where region ='Bogota' and estado_vacunacion = 'No vacunado' and  
fecha between TO_DATE('2021/01/01', 'yyyy/mm/dd') and  
TO_DATE('2021/12/31', 'yyyy/mm/dd')  
  
order by punto_vacunacion;
```

Caso 2

Caso solo con índices originales en el que se usan parámetros menos amplios.



Salida de Script x Rastreo Automático x Resultado de la Consulta x

Todas las Filas Recuperadas: 3 en 0,055 segundos

	CEDULA	NOMBRE_COMPLETO	ESTADO_VACUNACION	REGION	DESEA_SER_VACUNADO	PLAN_DE_VACUNACION	PUNTO_VACUNACION	OFICINA_RE
1	1001173818	MANUEL NARCISO CASTELAR	No vacunado	Andina	1	1	2	
2	1001168749	MIRTA GRACIELA BALOSTRO	No vacunado	Andina	0	1	199	
3	1001175898	VERONICA ANTONELLI	No vacunado	Andina	1	1	209	

Figura 6. Req 11 Caso 2

Parámetros:

```

where region = 'Andina' and estado_vacunacion = 'No vacunado' and
fecha between TO_DATE('2021/10/10', 'yyyy/mm/dd') and
TO_DATE('2021/12/31', 'yyyy/mm/dd')

order by punto_vacunacion;

```

3.3 Requerimiento funcional de consulta 12:

Sentencia SQL:

```
--Metodo req 12 administrador oficina regional
```

```

SELECT p.localizacion, punto_vacunacion, to_char(fecha -
7/24, 'IW') semana, count(id_cita) contador

FROM cita c inner join punto_vacunacion p on c.punto_vacunacion
= p.id_punto_vacunacion

where p.oficina_regional_eps = 1

GROUP BY p.localizacion, punto_vacunacion, to_char(fecha -
7/24, 'IW')

order by semana, contador DESC;

```

```
--Metodo req 12 administrador plan de vacunacion
```

```

SELECT p.localizacion, punto_vacunacion, to_char(fecha -
7/24, 'IW') semana, count(id_cita) contador

FROM cita c inner join punto_vacunacion p on c.punto_vacunacion
= p.id_punto_vacunacion

GROUP BY p.localizacion, punto_vacunacion, to_char(fecha -
7/24, 'IW')

order by semana, contador DESC;

```

Caso 1:

Caso normal en el que solo se usan los índices originales.

SQL Se han recuperado 950 filas en 0,919 segundos			
LOCALIZACION	PUNTO_VACUNACION	SEMANA	CONTADOR
1 Diagonal 7 # 96-99	239 01		1
2 Transversal 27 # 85-80	179 01		1
3 Diagonal 29 # 15-32	265 01		1
4 Carrera 10 # 68-6	198 01		1
5 Diagonal 123 #14	3 01		1
6 Diagonal 28 # 41-42	70 01		1
7 Diagonal 80 # 51-42	221 01		1
8 Diagonal 70 # 5-82	162 01		1
9 Diagonal 54 # 1-10	78 01		1
10 Diagonal 50 # 11-2	19 01		1
11 Transversal 66 # 32-1	229 01		1
12 Transversal 43 # 73-29	112 01		1
13 Carrera 14 # 13-79	92 01		1
14 Diagonal 57 # 83-55	209 01		1
15 Diagonal 66 # 58-80	141 01		1

Figura 7. Req 12 Caso 1

Parámetros:

```
GROUP BY p.localizacion,punto_vacunacion, to_char(fecha -
7/24,'IW')
```

```
order by semana, contador DESC;
```

3.4 Casos en los que se usaron los índices creados de manera propia

Los índices creados fueron los siguientes:

```
CREATE INDEX indice_region_ciudadano
```

```
ON Ciudadano(region);
```

```
CREATE INDEX indice_punto_vacunacion_ciudadano
```

```
ON Ciudadano(punto_vacunacion);
```

```
CREATE INDEX indice_ciudadano_cita
```

```
ON Cita(ciudadano);
```

```
CREATE INDEX indice_oficina_regional_eps_punto_vacunacion
ON punto_vacunacion(oficina_regional_eps);

commit;
```

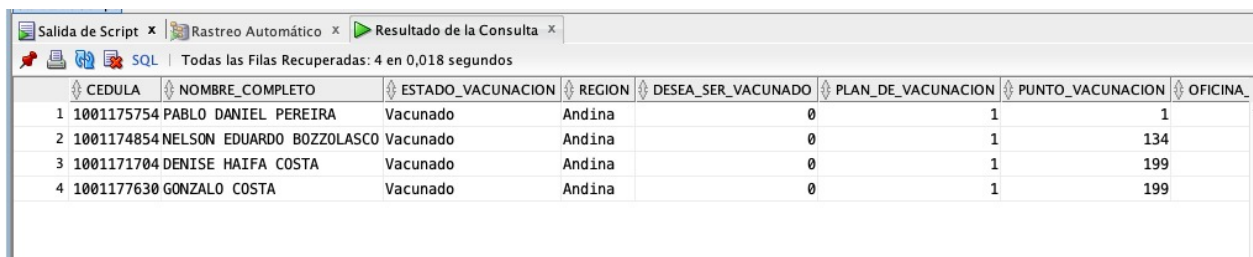
Los criterios bajo los cuales se eligieron todos los índices fueron:

- Son índices usados en WHERE con una búsqueda exacta. Esto significa que nunca son usados en rangos o contains.
- La selectividad no supera el 20% en ninguno de los casos. Pues la selectividad más alta con la que nos podemos encontrar es con 1/#OFICINAS_REGIONALES_EPS que para nuestro caso sería 1/35. Equivalente alrededor de un 2,85%.
- Para atributos como "ciudadano" en la tabla Cita no existe repetición. Esto facilita la implementación del índice y aumenta su factor de aceleración.
- Estas columnas NO son actualizables. El único atributo que se puede actualizar a través de la interfaz es la oficina regional en la tabla punto_vacunacion. Sin embargo, esta operación no es ejecutada de manera frecuente ni en grandes cantidades.
- Estos atributos nunca se llaman dentro de funciones si no son cabeza o porción del índice
- NO pueden ser nulos, por lo que condiciones como IS NULL, IS NOT NULL, LIKE o NOT EXISTS no se usen en ninguna parte del programa.
- Finalmente, es muy importante recalcar que son columnas usadas frecuentemente en JOINS (FK) por lo que el índice es un factor bastante influyente en la efectividad del programa

3.4.1 Requerimiento 10 con índices propios

Caso 3

Ahora se utilizaron para la búsqueda los índices que se mencionaron anteriormente. El caso 3 se hace basado en los mismos criterios de búsqueda que el caso 1 del requerimiento 10.



	CEDULA	NOMBRE_COMPLETO	ESTADO_VACUNACION	REGION	DESEA_SER_VACUNADO	PLAN_DE_VACUNACION	PUNTO_VACUNACION	OFICINA_
1	1001175754	PABLO DANIEL PEREIRA	Vacunado	Andina	0	1	1	
2	1001174854	NELSON EDUARDO BOZZOLASCO	Vacunado	Andina	0	1	134	
3	1001171704	DENISE HAIFA COSTA	Vacunado	Andina	0	1	199	
4	1001177630	GONZALO COSTA	Vacunado	Andina	0	1	199	

Figura 8. Req 10 Caso 3

Como se puede evidenciar, hubo una mejora de 0,035 segundos a 0,018 segundos, una mejora de casi el 50% gracias a la utilización de los nuevos índices que fueron creados

3.4.2 Requerimiento 11 con índices propios

Caso 3

Ahora se utilizaron para la búsqueda los índices que se mencionaron anteriormente. El caso 3 se hace basado en los mismos criterios de búsqueda que el caso 2 del requerimiento 11.



	CEDULA	NOMBRE_COMPLETO	ESTADO_VACUNACION	REGION	DESEA_SER_VACUNADO	PLAN_DE_VACUNACION	PUNTO_VACUNACION	OFICINA_RE
1	1001173818	MANUEL NARCISO CASTELAR	No vacunado	Andina	1	1	2	
2	1001168749	MIRTA GRACIELA BALOSTRO	No vacunado	Andina	0	1	199	
3	1001175898	VERONICA ANTONELLI	No vacunado	Andina	1	1	209	

Figura 9. Req 11 Caso 3

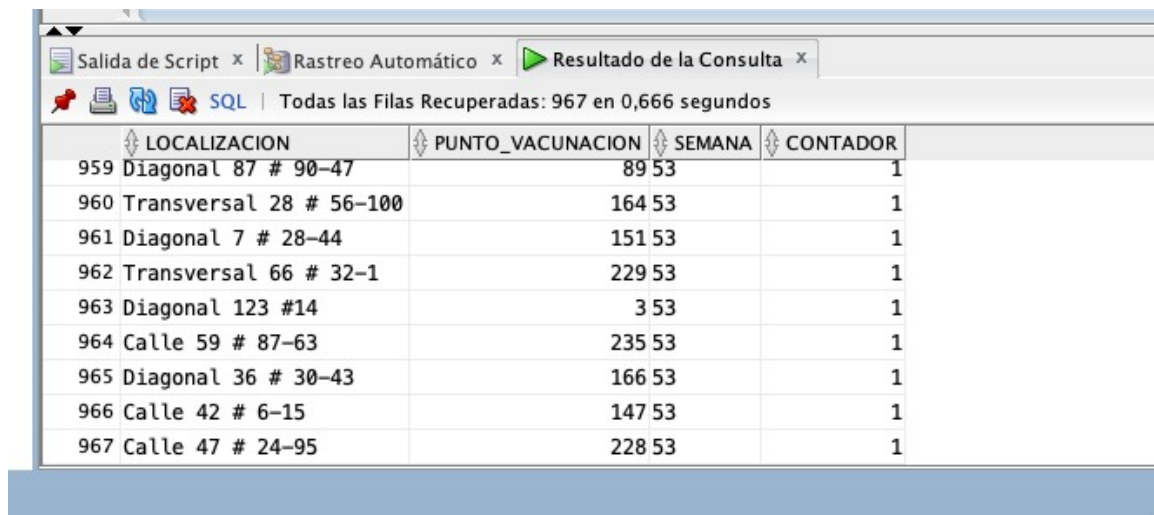
Como se puede evidenciar, se tuvo una muy buena mejora, pasando de 0,053 a 0,023 segundos, lo cual representaría una mejora de más del 50% al haber incluido los índices nuevos.

3.4.3 Requerimiento 12 con índices propios

Caso 2

Ahora se utilizaron para la búsqueda los índices que se mencionaron anteriormente.

Resultado:



	LOCALIZACION	PUNTO_VACUNACION	SEMANA	CONTADOR
959	Diagonal 87 # 90-47	89	53	1
960	Transversal 28 # 56-100	164	53	1
961	Diagonal 7 # 28-44	151	53	1
962	Transversal 66 # 32-1	229	53	1
963	Diagonal 123 #14	3	53	1
964	Calle 59 # 87-63	235	53	1
965	Diagonal 36 # 30-43	166	53	1
966	Calle 42 # 6-15	147	53	1
967	Calle 47 # 24-95	228	53	1

Figura 10. Req 12 Caso 2

Como se puede evidenciar, con respecto al caso 1, bajo la misma sentencia, con índices, se obtuvo una mejora en el tiempo de 0.253 segundos, que representa una mejora del 28.7% en el tiempo de búsqueda