-- 1. Explique brevemente los datos que el comando arroja como resultado

EXPLAIN ANALYZE SELECT \* FROM consumoVoz

WHERE fecha = '2016-06-21';

Seq Scan on consumovoz (cost=0.00..36.94 rows=2 width=24) (actual time=0.055..0.570 rows=1 loops=1)

Filter: (fecha = '2016-06-21'::date)

Rows Removed by Filter: 1914

Total runtime: 0.655 ms

(4 rows)

"Cost" indica el tiempo en el que la consulta empieza a funcionar.

"Rows" indica el numero de filas que devolvera la busqueda.

"Width" indica el ancho estimado en bytes de las filas que devuelve l busqueda.

"actual time" es un promedio de cada iteracion que se multiplica por el numero de bucles para obtener el tiempo real.

"actual rows" las filas que devuelve la exploracion del indice.

"actual loops" cuantas veces se ejetuca la exploracion del indice.

-- 2. Cree un índice en consumoVoz para el atributo fecha.

CREATE INDEX ON consumoVoz(fecha);

-- 3. Ejecute de nuevo el explain del punto 1. ¿Cuál es la diferencia en el plan y tiempo de ejecución de la siguiente consulta?

Bitmap Heap Scan on consumovoz (cost=4.29..9.96 rows=2 width=24) (actual time=0.061..0.062 rows=1 loops=1)

Recheck Cond: (fecha = '2016-06-21'::date)

-> Bitmap Index Scan on consumovoz\_fecha\_idx (cost=0.00..4.29 rows=2 width=0) (actual time=0.053..0.053 rows=1 loops=1)

Index Cond: (fecha = '2016-06-21'::date)

Total runtime: 0.106 ms

(5 rows)

Podemos ver que el tiempo de ejecucion de esa consulta es mucho menor debido al tipo de busqueda que implementa gracias al btree que se genero apartir de asignarle un index a la tabla, haciendo que las consultas por ese index sean mucho mas rapidas.

-- 4.Cree un índice que disminuya el tiempo de ejecución una consulta que retorne la dirección y ciudad del cliente “Glendon” “Freda” (nombre - apellido) ¿El plan de ejecución usa el índice? ¿Disminuyó el tiempo de ejecución? ¿En cuanto?

CREATE INDEX ON Cliente(Nombre,Apellido);

EXPLAIN ANALYZE SELECT

direccion, nombreCiudad

FROM

Cliente NATURAL JOIN Ciudad

WHERE

Cliente.nombre ='Glendon' AND Cliente.Apellido = 'Freda';

Disminuye 0.172 ms gracias al indice.

0.262 sin indice

0.090 con indice en Nombre y apellido

-- 5. Cree un índice que disminuya el tiempo de ejecución de una consulta que retorne el total de segundos de los consumos de voz del período 100, ordenados por IDPlanTel. ¿El plan de ejecución usa el índice? ¿Disminuyó el tiempo de ejecución? ¿En cuanto?

CREATE INDEX ON Periodo(IDPeriodo);

EXPLAIN ANALYZE SELECT

IDPlantel,SUM(segundos)

FROM

ConsumoVoz NATURAL JOIN Periodo

WHERE

IDPeriodo = '100'

GROUP BY IDPlantel;

Sin index : 0.581 ms

Con index : 0.103 ms

-- 6. Cree una vista que contenga el id y nombre de los clientes cuya ciudad de residencia es ‘CALI’.

CREATE VIEW ClienteCali AS

SELECT

Cliente.ID,nombre

FROM

Cliente NATURAL JOIN Ciudad

WHERE

Ciudad.nombre = 'Cali';

-- 7. Use la vista que creó en el punto 6 para seleccionar la cantidad de teléfonos activos que tienen los clientes que residen en Cali.

SELECT

COUNT(Telefono.numeroLinea)

FROM

ClienteCali INNER JOIN Telefono ON(ID=IDCliente)

WHERE

estado = 'A';