EJERCICIOS DE OPTIMIZACION DE CONSULTAS

1. Consultar los índices de la tabla *linped* utilizando las instrucciones SQL que nos permiten obtener esta información.

SHOW INDEX FROM linped

2. Haz uso de EXPLAIN para obtener información sobre cómo se están realizando las consultas y di cuál de las dos consultas realizará menos comparaciones para encontrar el producto que estamos buscando. ¿Cuántas comparaciones se realizan en cada caso? ¿Por qué?

```
Select *
From vendedor
Where numvend= ‘1’;
```

La primera consulta tiene un select_type SIMPLE, lo cual indica que está bien optimizada, además ha ido directamente a encontrar el resultado sin pasar por comparaciones.

```
Select *
From vendedor
Where nomvend= ‘Agapito lafuente del corral’
```

La segunda consulta, pese a ser de select_type SIMPLE también, ha tenido que comparar 21 filas.

El motivo se debe a que hay un índice para la primera consulta que se encarga de filtrar el resultado y optimizar la BDD.

3. Queremos saber optimizar las siguientes consultas. ¿Cuál de las dos sería más eficiente? Se recomienda hacer uso de EXPLAIN para obtener información sobre cómo se están realizando las consultas.

```
SELECT sum(cantrecibida*preciocompra)

FROM linped

WHERE YEAR(fecharecep) = 1992
```

SELECT sum(cantrecibida*preciocompra)

FROM linped

WHERE fecharecep >= '1992-01-01' and fecharecep <='1992-12-31'

La consulta más eficiente es la que no hace uso de la función YEAR(), dado que esta podría ser filtrada mediante un índice, siendo el caso contrario para la que incluye YEAR().

4. Optimizar la siguiente consulta creando índices cuando sea necesario. Se recomienda hacer uso de EXPLAIN para obtener información sobre cómo se están realizando las consultas.

SELECT *

FROM vendedor INNER JOIN preciosum

ON vendedor.numvend = preciosum.numvend

WHERE nomvend LIKE 'A%'

Primero, creo el índice pertinente:

CREATE FULLTEXT INDEX idx_nomvend ON vendedor (nomvend);

Luego hago la consulta adaptada al índice junto a EXPLAIN para verificar la optimización:

EXPLAIN SELECT *

FROM vendedor v

WHERE MATCH (v.nomvend) against ('a*' IN boolean mode)

5. Crea un índice FULLTEXT sobre las columnas numpieza y nompieza de la tabla pieza.

CREATE FULLTEXT INDEX idx_numnompieza ON pieza (numpieza, nompieza)

6. Una vez creado el índice del ejercicio anterior realiza la siguiente consulta haciendo uso de la función MATCH, para buscar todos las piezas que contienen la palabra IBM en el nombre o en la descripción.

EXPLAIN SELECT pi.numpieza

FROM pieza pi

WHERE MATCH (pi.numpieza, pi.nompieza) against ('ibm')

7. Crea un índice compuesto por las columnas nomvend y nombrecomer de la tabla vendedor

CREATE FULLTEXT INDEX idx_nomvend_nomcomer ON vendedor (nomvend, nombrecomer);

8. Una vez creado el índice del ejercicio anterior realice las siguientes consultas haciendo uso de EXPLAIN:

· Busca el vendedor Pedro de la empresa Soft. ¿Cuántas filas se han examinado hasta encontrar el resultado?

EXPLAIN SELECT v.*

FROM vendedor v

WHERE MATCH (v.nomvend, v.nombrecomer) against ('Pedro soft');

Solo se ha tenido que examinar una sola fila gracias al índice creado anteriormente.

· Busca el vendedor anterior utilizando solamente la empresa soft. ¿Cuántas filas se han examinado hasta encontrar el resultado?

EXPLAIN SELECT v.*
FROM vendedor v

WHERE MATCH (v.nombrecomer) against ('soft');

No deja ejecutar la consulta ya que el orden de los parámetros influye, en todo caso podría buscar individualmente v.nomvend que ocupa la primera posición en el orden del índice creado.

· Busca el vendedor anterior utilizando solamente el nombre Pedro.
¿Cuántas filas se han examinado hasta encontrar el resultado? ¿Qué ha ocurrido en este caso?

EXPLAIN SELECT v.*

FROM vendedor v

WHERE MATCH (v.nomvend) against ('pedro');

Aquí sí deja ejecutar la consulta por lo que se explicó en la consulta anterior, puesto que aquí al ocupar la primera posición si puede usarse individualmente.

9. Calcula cuál podría ser un buen valor para crear un índice sobre un prefijo de la columna nompieza de la tabla pieza. Crea el índice y ejecuta algunas consultas para probarlo.

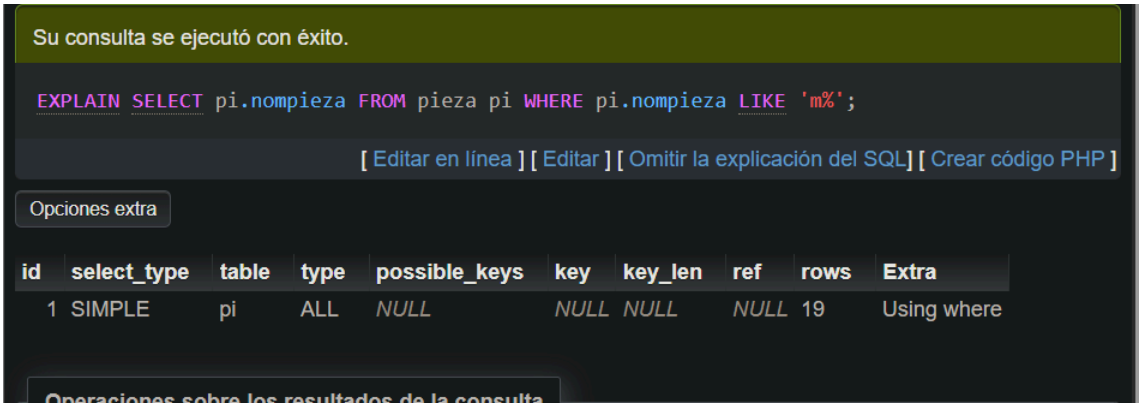
CREATE INDEX idx_nompieza_reducido ON pieza (nompieza(20));

EXPLAIN SELECT pi.nompieza

FROM pieza pi

WHERE pi.nompieza LIKE 'm%'

Antes de usarlo:



Después de aplicar el índice correctamente: