

Ejercicios sobre Índices

Base de datos: Jardinería

1. Consulte cuáles son los índices que hay en la tabla `producto` utilizando las dos instrucciones SQL que nos permiten obtener esta información de la tabla.

show index from producto;

describe producto;

2. Haga uso de `EXPLAIN` para obtener información sobre cómo se están realizando las consultas y diga 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 producto  
WHERE codigo_producto = 'OR-114';
```

En este caso se realiza solo una comprobación, ya que se usa el índice de la primary key para la realización de la consulta.

```
SELECT *  
FROM producto  
WHERE nombre = 'Evonimus Pulchellus';
```

En este caso se realizan 276 comprobaciones porque no se usa ningún índice para realizar la consulta, por tanto se tienen que revisar todos los registros de la tabla.

Por supuesto, tras ver los resultados, la primera consulta es la que mejor rendimiento y más optimizada está.

3. Suponga que estamos trabajando con la base de datos `jardineria` y 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 AVG(total)  
FROM pago  
WHERE YEAR(fecha_pago) = 2008;
```

```
SELECT AVG(total)
```

```
FROM pago
WHERE fecha_pago >= '2010-01-01' AND fecha_pago <= '2010-12-31';
```

En este caso, ya que no se dispone de índice en el campo `fecha_pago` de la tabla `pago`, las dos serían igual de eficientes puesto que las fechas no estarían ordenadas y habría que buscar en todos los registros por igual. Si le añadiésemos un índice al campo `fecha_pago` para mejorar las consultas, igualmente la primera consulta con el `YEAR`, invalida el índice y no lo usa, mientras la segunda consulta sí que usará dicho índice.

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

```
SELECT *
FROM cliente INNER JOIN pedido
ON cliente.codigo_cliente = pedido.codigo_cliente
WHERE cliente.nombre_cliente LIKE 'A%';
```

create index idx_nombre_cliente on cliente(nombre_cliente);

5. ¿Por qué no es posible optimizar el tiempo de ejecución de las siguientes consultas, incluso haciendo uso de índices normales? Prueba a optimizar dichas consultas con el índice adecuado.

Porque los índices lo que hacen es ordenar el campo donde lo creas, ya sea numéricamente o alfabéticamente (empezando por el comienzo), por tanto, si consultamos patrones que además están en medio o final de la palabra con un índice normal, aún ordenando el campo, seguiremos teniendo que verificar todos los registros para encontrar el patrón, pues no existe ningún orden.

```
SELECT *
FROM cliente INNER JOIN pedido
ON cliente.codigo_cliente = pedido.codigo_cliente
WHERE cliente.nombre_cliente LIKE '%A%';
```

```
SELECT *
FROM cliente INNER JOIN pedido
ON cliente.codigo_cliente = pedido.codigo_cliente
WHERE cliente.nombre_cliente LIKE '%A';
```

Estas consultas no se podrían optimizar con ningún índice, puesto que con consultas con patrones de 3 o menos letras o búsquedas que den más del 50% de registros como resultado, `mysql` desecha los resultados y no muestra nada.