

OPTIMIZACIÓN DE LAS CONSULTAS EN SQL

Autor: Hugo Pelayo

Asignatura: Bases de datos

Fecha: 24 de febrero 2023

Índice

Introducción	3
Utilización de índices	4
Tipos de Índices.....	5
Usos. Modificación y eliminación de índices	6
Bibliografía.....	8

Introducción

El sistema gestor de bases de datos MySQL está diseñado de tal manera que permite funcionar con un rendimiento notable, aún así es fácil que nosotros como programadores realicemos consultas sobre la base de datos o cambios sobre la misma que reduzcan el rendimiento de nuestro gestor.

Una de las soluciones adoptadas en MySQL para la mejora de rendimiento en el procesamiento de datos de una base es la indexación. Esta herramienta es de las más conocidas para la mejora de rendimiento en bases de datos MySQL y, paradójicamente, la no utilización de la misma acostumbra a manifestar carencias de rendimiento.

Los índices se definen como un tipo de datos que son, en esencia, unas estructuras de datos que sirven para mejorar la velocidad de las operaciones sobre una base. Estas estructuras utilizan un identificador único por cada fila o registro de una tabla para facilitar el acceso a los datos. Funciona de una manera similar a las entradas de un diccionario guardando parejas de datos donde tendríamos una clave que es la palabra en cuestión y un valor que son el conjunto de entradas o acepciones de la palabra, de modo que si queremos encontrar la definición de una palabra en concreto la buscaríamos por su nombre o clave en el diccionario y no lo recorreríamos por completo hasta encontrar la palabra en cuestión, una mecánica similar se aplica con los índices sobre bases de datos de MySQL.

Para buscar un elemento indexado basta con buscar dicho índice en la base de datos y una vez encontrado, devolver el registro con dicha clave o índice.

Los índices pueden ser creados utilizando una o más columnas, proporcionando la base tanto para búsquedas rápidas al azar como para accesos eficientes. Los índices normalmente se implementan como árboles B, árboles B+, árboles B* (o B estrella) o una mezcla de estos tres. Entre las principales ventajas del uso de índices en la ordenación de bases de datos destacan:

- Ordenación de las tablas por varios criterios simultáneamente.

- Es menos costoso ordenar un archivo índice, porque incluye sólo referencias a la información y no la información en sí (mecánica similar a la de un puntero en programación).
- El coste de inserción y eliminación es menor
- Con los registros siempre ordenados se utilizarán algoritmos mucho más eficientes que el simple recorrido secuencial de las consultas.

Entre algunas desventajas sobre el uso de índices destacan:

- El consumo de espacio en disco
- Aun teniendo registros pequeños, el mantener en orden un índice disminuye la velocidad de las operaciones de escritura sobre la tabla.

Utilización de índices

MySQL utiliza los índices para encontrar los registros que contienen los valores específicos de las columnas indicadas de forma más eficiente. Sin la utilización de índices el gestor realiza las búsquedas de forma lineal, es decir, si queremos hacer una consulta, esta se realiza empezando por la primera fila, se recorren entonces todos los registros de una tabla o combinación de varias según sea el caso y se evalúan y devuelve los registros que cumplan las condiciones que hayamos indicado en nuestra *query*. En casos generales, MySQL emplea los índices para las siguientes acciones:

- Encontrar registros que cumplan condiciones de una cláusula **WHERE** en una consulta sobre una tabla cuyos registros están indexados.
- Recuperación de registros de tablas que se emplean en operaciones **JOIN**. Es importante que los índices sean compatibles, es decir, que tengan mismo tamaño y tipo.
- Disminución del tiempo de ejecución de las consultas resultantes del uso de las cláusulas **ORDER BY** o **GROUP BY** si todas las columnas presentes en los criterios forman parte del índice.

- Si la consulta utiliza una condición sencilla columna con una columna donde la condición está indexada, las filas serán recuperadas directamente a partir del índice, sin pasar a consultar la tabla.

Tipos de Índices

Existen cinco tipos principales de índices en MySQL: INDEX (NON-UNIQUE), UNIQUE, PRIMARY, FULLTEXT, SPATIAL. A continuación, se describen brevemente:

- Index: Hace referencia a un índice normal, no único. Este tipo de índice admite valores duplicados en una columna.
- Unique: Es un tipo de índice en que los valores de la columna en concreto deben ser únicos, no se admiten valores duplicados para la columna que compone el índice.
- Primary: Al igual que los índices de tipo Unique, los valores de la columna que componen este índice deben ser únicos entre ellos, con la diferencia de que sólo puede existir un índice por tabla.
- Fulltext: Estos índices se utilizan para las búsquedas sobre texto (columnas de tipo VARCHAR, CHAR Y TEXT). Estos índices se componen por todas las palabras que están contenidas en las columnas que contienen el Fulltext.
- Spatial: estos índices se emplean para realizar búsquedas sobre datos que componen formas geométricas representadas en el espacio.

Usos. Modificación y eliminación de índices

La sintaxis básica para la creación de índices es:

```
CREATE [UNIQUE | FULLTEXT | SPATIAL] INDEX index_name
    [index_type]
    ON tbl_name (key_part,...)
    [index_option]
    [algorithm_option | lock_option] ...

key_part: {col_name [(length)] | (expr)} [ASC | DESC]
```

Figura 1. Sintaxis creación índice. Fuente: mysql.com

También se pueden crear en el momento de creación de la tabla:

```
CREATE TABLE t1 (
    col1 VARCHAR(10),
    col2 VARCHAR(20),
    INDEX (col1, col2(10))
);
```

Figura 2. Sintaxis creación índice. Fuente: mysql.com

La sintaxis para la eliminación de los índices es:

```
DROP INDEX index_name ON tbl_name
    [algorithm_option | lock_option] ...

algorithm_option:
    ALGORITHM [=] {DEFAULT | INPLACE | COPY}

lock_option:
    LOCK [=] {DEFAULT | NONE | SHARED | EXCLUSIVE}
```

Figura 3. Sintaxis eliminación de índices. Fuente: mysql.com

Result Grid			
Filter Rows:		Export:	Wrap Cell Content:
Query_ID	Duration	Query	
41	0.00010475	SET profiling = 1	
42	0.00005125	SHOW WARNINGS	
43	0.00088475	DESCRIBE Empleados	
44	0.02566225	CREATE UNIQUE INDEX Idx_EmpNo ON Empleados (Emp_No)	
45	0.00024875	SELECT * FROM Empleados WHERE Emp_No = '7900' LIMIT 0, 50	
46	0.00014450	SELECT * FROM Empleados USE INDEX (Idx_EmpNo) WHERE Emp_No = '7900' LIMIT 0, 50	
47	0.00026175	SELECT * FROM Empleados WHERE Emp_No = '7900' LIMIT 0, 50	
48	0.00016600	SELECT * FROM Empleados USE INDEX (Idx_EmpNo) WHERE Emp_No = '7900' LIMIT 0, 50	
49	0.00026900	SELECT * FROM Empleados WHERE Emp_No = '7900' LIMIT 0, 50	
50	0.00018650	SELECT * FROM Empleados USE INDEX (Idx_EmpNo) WHERE Emp_No = '7900' LIMIT 0, 50	

Figura 4. Comparativa ejecución utilizando índices. Fuente: elaboración propia

Para ejemplos de usos prácticos se adjunta el fichero “Ejemplo usos.sql”.

Bibliografía

- [1] R. V. Amor, «Introducción a índices en MySQL,» autentia, 11 Septiembre 2015. [En línea]. Available: <https://www.adictosaltrabajo.com/2015/09/11/introduccion-a-indices-en-mysql/>. [Último acceso: 24 Febrero 2023].
- [2] Wikipedia, «Índice (bases de datos),» Wikipedia, 29 Noviembre 2022. [En línea]. Available: [https://es.wikipedia.org/wiki/%C3%8Dndice_\(base_de_datos\)](https://es.wikipedia.org/wiki/%C3%8Dndice_(base_de_datos)). [Último acceso: 24 Febrero 2023].
- [3] M. G. G. Ó. P. M. Luis Alberto Casillas Santillán, Bases de datos MySQL, UOC.
- [4] MySQL, «EXPLAIN Output Format,» [En línea]. Available: <https://dev.mysql.com/doc/refman/8.0/en/explain-output.html>. [Último acceso: 24 Febrero 2023].