

Índices en MySQL

José Manuel Sánchez Álvarez

Índices

MySQL usa índices para encontrar rápidamente filas con valores de columna específicos. Sin un índice, MySQL debe escanear toda la tabla para localizar las filas relevantes. La tabla más grande, más lenta busca.

Introducción al índice

Un índice es una estructura de datos como B-Tree que mejora la velocidad de recuperación de datos en una tabla a costa de escrituras y almacenamiento adicionales para mantenerla.

El optimizador de consultas puede usar índices para localizar datos rápidamente sin tener que escanear cada fila de una tabla para una consulta determinada.

Cuando crea una tabla con una clave primaria o una clave única, MySQL crea automáticamente un índice especial llamado PRIMARY. Este índice se llama índice agrupado.

El índice PRIMARY (primario) es especial porque el índice en sí se almacena junto con los datos en la misma tabla. El índice agrupado impone el orden de las filas en la tabla.

Otros índices distintos al índice PRIMARY se denominan índices secundarios o índices no agrupados.

Sentencia MySQL CREATE INDEX

Normalmente, se crean índices para una tabla en el momento de la creación. Por ejemplo, la siguiente instrucción crea una nueva tabla con un índice que consta de dos columnas c2 y c3.

```
CREATE TABLE t (
  c1 INT PRIMARY KEY,
  c2 INT NOT NULL,
  c3 INT NOT NULL,
  c4 VARCHAR(10),
  INDEX (c2, c3)
);
```

Para agregar un índice para una columna o un conjunto de columnas, use la instrucción CREATE INDEX de la siguiente manera:

```
CREATE INDEX index_name ON table_name (column_list)
```

Para crear un índice para una columna o una lista de columnas, especifique el nombre del índice, la tabla a la que pertenece el índice y la lista de columnas.

Por ejemplo, para agregar un nuevo índice para la columna c4, utilice la siguiente instrucción:

```
CREATE INDEX idx_c4 EN t (c4);
```

De forma predeterminada, MySQL crea índices mediante árboles B (B-Tree) si no se especifica el tipo. A continuación se muestra el tipo de índice permitido según el motor de almacenamiento de la tabla:

Motor de almacenamiento	Tipo de índices permitidos
InnoDB	B-Tree
MyISAM	B-Tree
MEMORY/HEAP	Hash, B-Tree

Ejemplo de MySQL CREATE INDEX

La siguiente declaración busca empleados cuyo cargo es Representante de ventas

```
SELECT
  employeeNumber,
  lastName,
  firstName
FROM
  employees
```

```
WHERE
    jobTitle = 'Sales Rep';
```

La salida son 17 filas. Para ver lo que hace internamente MySQL podemos utilizar el comando EXPLAIN en el SELECT anterior:

#id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered
1	SIMPLE	employees	null	ALL	null	null	null	null	23	10.00

Vemos que se han recorrido 23 filas para producir el resultado. Ahora creamos el índice siguiente:

```
CREATE INDEX jobTitle ON employees(jobTitle);
```

Al hacer el mismo EXPLAIN sobre el SELECT anterior, ahora sólo se escanean 17 filas, las mismas que resultados:

#id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered
1	SIMPLE	employees	NULL	ref	jobTitle	jobT				

Para ver los índices que tiene una tabla:

```
SHOW INDEXES FROM employees;
```

Sintaxis de la instrucción MySQL DROP INDEX

Para eliminar un índice existente de una tabla, use la instrucción DROP INDEX de la siguiente manera:

```
DROP INDEX index_name ON table_name
[algorithm_option | lock_option];
```

■ Algoritmo

`algorithm_option` permite especificar un algoritmo específico utilizado para la eliminación del índice. A continuación se muestra la sintaxis de la cláusula `algorithm_option`:

```
ALGORITHM [=] {DEFAULT|INPLACE|COPY}
```

Para la eliminación del índice, se admiten los siguientes algoritmos:

COPIAR: la tabla se copia en la nueva tabla fila por fila, el DROP INDEX se realiza en la copia de la tabla original. Las instrucciones de manipulación de datos concurrentes como INSERT y UPDATE no están permitidas.

INPLACE: la tabla se reconstruye donde está, en lugar de copiarse a una nueva. MySQL emite un bloqueo exclusivo de metadatos en la tabla durante las fases de preparación y ejecución de la operación de eliminación de índice. Este algoritmo permite comandos de manipulación de datos concurrentes.

Tenga en cuenta que la cláusula ALGORITHM es opcional. Si lo omite, MySQL usa INPLACE. En caso de que INPLACE no sea compatible, MySQL usa COPY.

Usar DEFAULT tiene el mismo efecto que omitir la cláusula ALGORITHM.

- Lock La opción `lock_option` controla el nivel de lecturas y escrituras concurrentes en la tabla mientras se elimina el índice.

A continuación se muestra la sintaxis de `lock_option`:

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

Se admiten los siguientes modos de bloqueo:

DEFAULT: permite tener el nivel máximo de concurrencia para un algoritmo dado. Primero, permite lecturas y escrituras concurrentes si es compatible. Si no, permite lecturas concurrentes si es compatible. Si no, impone el acceso exclusivo.

NONE: si el motor soporta NONE, puede tener lecturas y escrituras concurrentes. De lo contrario, MySQL emite un error.

SHARED: si se admite SHARED, puede haber lecturas concurrentes, pero no escrituras. MySQL emite un error si las lecturas concurrentes no son compatibles.

EXCLUSIVE: fuerza el acceso exclusivo.