

Índices

[Indices](#)

[Índice simple y compuesto.](#)

[UNIQUE INDEX](#)

[PRIMARY INDEX](#)

[KEY INDEX](#)

[Estructura de indices](#)

Indices

Definición

Un índice es una estructura de datos definida sobre una columna de tabla (o varias) y que permite localizar de forma rápida las filas de la tabla en base a su contenido en la columna indexada además de permitir recuperar las filas de la tabla ordenadas por esa misma columna.

Funciona de forma parecida al índice de un libro donde tenemos el título del capítulo y la página donde empieza dicho capítulo, en un índice definido sobre una determinada columna tenemos el contenido de la columna y la posición de la fila que contiene dicho valor dentro de la tabla.

La definición de los índices de la base de datos es tarea del administrador de la base de datos, los administradores más experimentados pueden diseñar un buen conjunto de índices, pero esta tarea es muy compleja, consume mucho tiempo y está sujeta a errores, incluso con cargas de trabajo y bases de datos con un grado de complejidad no excesivo.

En general, MySQL emplea los índices para las siguientes acciones:

- Encontrar las filas que cumplen la condición WHERE de la consulta cuyas columnas están indexadas.
- Para recuperar las filas de otras tablas cuando se emplean operaciones de tipo JOIN. Para ello, es importante que los índices sean del mismo tipo y tamaño ya que aumentará la eficiencia de la búsqueda. Por ejemplo: una operación de tipo JOIN sobre dos columnas que tengan un índice del tipo INT(10).
- Disminuir el tiempo de ejecución de las consultas con ordenación (ORDER BY) o agrupamiento (GROUP BY) si todas las columnas presentes en los criterios forman parte de un índice.
- Si la consulta emplea una condición simple cuya columna de la condición está indexada, las filas serán recuperadas directamente a partir del índice, sin pasar a consultar la tabla.

Tipos de índices

Índice simple y compuesto.

Un índice simple está definido sobre una sola columna de la tabla mientras que un índice compuesto está formado por varias columnas de la misma tabla (tabla sobre la cual está definido el índice).

Cuando se define un índice sobre una columna, los registros que se recuperen utilizando el índice aparecerán ordenados por el campo indexado. Si se define un índice compuesto por las columnas col1 y col2, las filas que se recuperen utilizando dicho índice aparecerán ordenadas por los valores de col1 y todas las filas que tengan el mismo valor de col1 se

ordenarán a su vez por los valores contenidos en col2, función igual que la cláusula ORDER BY vista en el tema de consultas simples.

Por ejemplo si definimos un índice compuesto basado en las columnas (provincia, localidad), las filas que se recuperen utilizando este índice aparecerán ordenadas por provincia y dentro de la misma provincia por localidad.

UNIQUE INDEX

Este tipo de índice se refiere a un índice en el que todas las columnas deben tener un valor único. Esto implica que no admite valores duplicados para la columna (o columnas) que componen el índice. Aplica la restricción de que los datos de la columna (o columnas) deben tener un valor único.

Pueden existir n UNIQUE INDEX Por cada tabla

PRIMARY INDEX

Este tipo de índice se refiere a un índice en el que todas las columnas deben tener un valor único (al igual que en el caso del índice UNIQUE) pero con la limitación de que sólo puede existir un índice PRIMARY en cada una de las tablas. Aplica la restricción de que los datos de la columna (o columnas) deben tener un valor único.

En MySQL, los datos de la tabla que está asociado el índice PRIMARY, se guardan según el orden especificado en este objeto.

KEY INDEX

Este tipo de índice se refiere a un índice normal, no único. Esto implica que admite valores duplicados para la columna (o columnas) que componen el índice. No aplica ninguna restricción especial a los datos de la columna (o columnas) que componen el índice sino que se emplea simplemente para mejorar el tiempo de ejecución de las consultas.

FULLTEXT

Estos índices se emplean para realizar búsquedas sobre texto (CHAR, VARCHAR y TEXT). Estos índices se componen por todas las palabras que están contenidas en la columna (o columnas) que contienen el índice. No aplica ninguna restricción especial a los datos de la columna (o columnas) que componen el índice sino que se emplea simplemente para mejorar el tiempo de ejecución de las consultas. Este tipo de índices sólo están soportados por InnoDB y MyISAM en MySQL 5.7.

SPATIAL

Estos índices se emplean para realizar búsquedas sobre datos que componen formas geométricas representadas en el espacio. Este tipo de índices sólo están soportados por InnoDB y MyISAM en MySQL 5.7.

Estructura de índices

Una vez hemos visto los tipos de índices, vamos a ver los distintos tipos de estructuras que se pueden crear para almacenar los índices junto con las características de cada uno de ellas:

- **B-TREE:** este tipo de índice se usa para comparaciones del tipo =, >, <, >=, <=, BETWEEN y LIKE (siempre y cuando se utilice sobre constantes que no empiecen por %). Para realizar búsquedas empleando este tipo de índice, se emplea cualquier columna (o conjunto de columnas) que formen el prefijo del índice. Por ejemplo: si un índice está formado por las columnas [A, B, C], se podrán realizar búsquedas sobre: [A], [A, B] o [A, B, C].

Un índice B-TREE puede ser usado también para los LIKE si el argumento del LIKE es una constante que no comience con un comodín. Por ej :

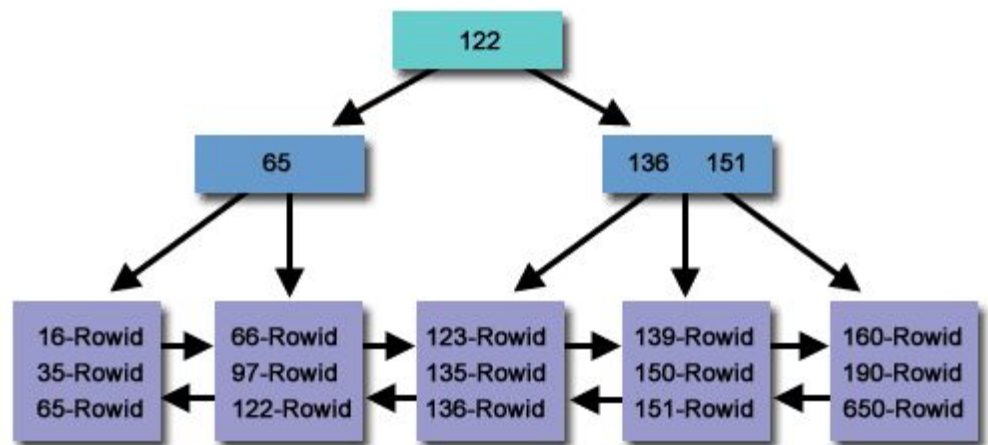
```
SELECT * FROM <tabla> where key_col like "Pedro%"
```

Todo índice que no tenga todos los niveles AND de una query no va a ser utilizado.

IS NULL va a ser utilizado si el nombre de columna está indexado.

La idea general de un B-TREE es que los valores se almacenan en orden y a su vez cada nodo hoja del árbol esté a la misma distancia del nodo root.

Los árboles B, mejoran la velocidad de acceso a datos debido a que MySQL no debe escanear toda la tabla, pero siempre arranca del nodo root. Los datos en cada nodo contienen punteros a nodos hijos y definen los límites mayores y menores a los nodos hojas. En cada uno de los nodos hoja hay un puntero a los datos en la tabla correspondiente.



- **HASH:** este tipo de índice sólo se usa para comparaciones del tipo = o <=>. Para este tipo de operaciones son muy rápidos en comparación a otro tipo de estructura. Para realizar búsquedas empleando este tipo de índice, se emplearán todas las columnas que componen el índice.

Un índice puede ser almacenado en cualquier tipo de estructura pero, en función del uso que se le vaya a dar, puede interesar crear el índice en un tipo determinado de estructura o en otro. Por norma general, un índice siempre se creará con la estructura de B-TREE, ya que es la estructura más empleada por la mayoría de operaciones.

Sintaxis para la creación de índices

```
CREATE [UNIQUE | FULLTEXT | SPATIAL] INDEX <NOMBRE_INDICE> ON
<NOMBRE_TABLA> (<LISTA_COLUMNAS>) [USING] <TIPO_INDICE>
```

Donde:

- **NOMBRE_INDICE** : es el nombre del índice.
- **NOMBRE_TABLA**: es el nombre de la tabla donde se va a crear el índice.
- **LISTA_COLUMNS**: nombre de la columna (o columnas) que formarán el índice.
- **TIPO_INDICE**: es el tipo del índice. Se emplea con USING [BTREE | HASH].

Creación de un índice UNIQUE :

```
CREATE UNIQUE INDEX <NOMBRE_INDICE> on <TABLA>(<CAMPO1>  
<ASC|DESC>,<CAMPO2> <ASC|DESC> ... <CAMPON> ASC|DESC);
```

En el siguiente ejemplo se crea un índice sobre la tabla PERSONAS ordenando el mismo por los campos apellido en forma descendente y nombre en forma ascendente.

```
CREATE UNIQUE INDEX  IDX_PERSONAS ON personas (apellido desc,  
nombre asc) USING BTREE;
```

Para incluir campos como nodos hojas del índice no agrupado se debe incluir la palabra clave INCLUDES.

Para eliminar un índice la sintaxis es la siguiente :

```
DROP INDEX <NOMBRE_INDICE> ON <TABLA>;
```