

Índices en MySQL 2

Índices

Ver qué índices tenemos

- En una tabla

```
SHOW INDEXES FROM table_name;
```

Especificando a qué base de datos pertenece la tabla

```
SHOW INDEXES FROM table_name  
IN database_name;
```

La siguiente consulta es igual a la anterior:

```
SHOW INDEXES FROM database_name.table_name;
```

INDEX y KEY son sinónimos, así como IN y FROM así que las 2 siguientes consultas son equivalentes:

```
SHOW INDEX IN table_name  
FROM database_name;
```

- Información devuelta por SHOW INDEXES

- table

El nombre de la tabla

- non_unique

1 si el índice puede contener duplicados, 0 si no puede.

- `keyname`

El nombre del índice. El índice de clave principal siempre tiene el nombre PRIMARY.

- `seqinindex`

El número de secuencia de la columna en el índice. El primer número de secuencia de columna comienza desde 1.

- `columnname`

El nombre de la columna

- `collation`

La colación (collation) representa cómo se ordena la columna en el índice. A significa ascendente, B significa descendente o NULL significa no ordenado.

- `cardinality`

La cardinalidad devuelve un número estimado de valores únicos en el índice.

Tenga en cuenta que cuanto mayor sea la cardinalidad, mayores serán las posibilidades de que el optimizador de consultas use el índice para las búsquedas.

- `subpart`

El prefijo de índice. Es nulo si toda la columna está indexada. De lo contrario, muestra el número de caracteres indexados en caso de que la columna esté parcialmente indexada.

- `packed`

indica cómo se empaqueta la llave; NUL si no se empaqueta.

- `null`

YES si la columna puede contener valores NULL y en blanco si no lo contiene.

- `indextype`

representa el método de índice utilizado, como BTREE , HASH , RTREE o FULLTEXT .

- comentario

La información sobre el índice no descrita en su propia columna.

- `indexcomment`

muestra el comentario para el índice especificado cuando crea el índice con el atributo COMMENT .

- visible

Si el índice es visible o invisible para el optimizador de consultas o no; YES si es así, NO si no.

- `expression`

Si el índice usa una expresión en lugar del valor de columna o prefijo de columna, la expresión indica la expresión para la parte clave y también la columna `columnname` es NULL.

■ Filtrar la información de los índices

Podemos añadir una cláusula where para filtrar lo que muestra `show indexes`.

```
SHOW INDEXES FROM table_name
WHERE condition;
```

Índices únicos UNIQUE

- Crear índices UNIQUE Añadiéndolos a una tabla

```
CREATE UNIQUE INDEX index_name
ON table_name(index_column_1,index_column_2,...);
```

O con la creación de la tabla

```
CREATE TABLE table_name(
...
    UNIQUE KEY(index_column_,index_column_2,...)
);
```

- MySQL UNIQUE Index & NULL

A diferencia de otros sistemas de bases de datos, MySQL considera los valores NULL como valores distintos. Por lo tanto, puede tener múltiples valores NULL en el índice UNIQUE.

Así es como se diseñó MySQL. No es un error a pesar de que se informó como un error.

Importante, la restricción UNIQUE no se aplica a los valores NULL, excepto para el motor de almacenamiento BDB.

Índices para los prefijos de una columna PREFIX INDEX

Cuando tenemos columnas de texto y queremos añadir un índice sobre ella hay que tener en cuenta, que si el texto es largo el índice va a usar mucho espacio y las inserciones van a ser más lentas. Una opción es crear el índice sólo sobre los primeros n caracteres con `column_name(length)`:

```
CREATE TABLE table_name(
    column_list,
    INDEX(column_name(length))
);
```

o

```
CREATE INDEX index_name
ON table_name(column_name(length));
```

- Podemos buscar la longitud óptima con:

```
SELECT
    COUNT(DISTINCT LEFT(productName, 20)) unique_rows
FROM
    products;
```

Donde 20 es la longitud de la cadena que vamos a usar para el distinct.
Cuanto más se acerque este SELECT a uno sin usar LEFT mejor.