

Vistas e Índices

Sitio: [FRANCISCO DE GOYA](#)
Curso: Bases de Datos Pendientes
Libro: Vistas e Índices

Imprimido por: Lucía Hernández Montero
Día: lunes, 24 de octubre de 2022, 09:44

Tabla de contenidos

1. Índices

1.1. Sintaxis

1.2. Creación y borrado de índices

2. Vistas

2.1. Creación y borrado

1. Índices

Los índices en una BD funcionan como un índice en un libro físico: ayuda a encontrar antes un dato.

Estructuralmente es una lista ordenada por cierto campo donde cada elemento indica una fila de la tabla. No es necesario que el usuario de la base de datos sea consciente de que se usa, ya que si es posible, lo hace el SGBD.

Automáticamente toda tabla con clave primaria tiene un índice con dicha columna.

Ventajas

- Aumenta la velocidad de ejecución de consultas. Especialmente en tablas de millones o miles de millones de filas.

Desventajas

- Es necesario que se diseñe correctamente para que sea útil. No cualquier índice redundará en una mayor eficiencia.
- Puede reducir el rendimiento de sentencias insert, update y delete. Además de modificar el dato en la tabla se ha de actualizar también en el índice
- Ocupa espacio en el disco duro adicional al propio de la tabla.

Uso

El SGBD se encarga automáticamente de usarlo, de forma transparente al usuario, bajo ciertas condiciones:

- El campo indexado está en el where de un select.
- Dos campos indexados de dos tablas diferentes, igualados en un join.
- Es usado para ordenar el resultado con un order by.
- Una consulta que solo use el índice, usará este y no la tabla.

1.1. Sintaxis

Los índices se pueden crear mediante una sentencia específica o mediante un alter table y un create table. El sintaxis de creación de un índice es el siguiente:

```
create [opcion1] index <name> on <tabla> [opcion2]
```

En el caso de opción 1 existen tres posibilidades sobre el funcionamiento del índice respecto a sus columnas:

- index: índice sin restricciones.
- unique: es el conocido en la UT5. No permite que haya repeticiones, pero sí admite nulos (en caso de que la columna lo permita).
- primary key: también es el conocido en la UT5, pero no puede crearse mediante esta sentencia, sino solo por alter o create table.
- fulltext: solo admite columnas de tipo varchar, char y text.
- spatial: solo admite columnas de tipo point y geometry.

En el caso de la opción 2, hay dos opciones que hacen referencia al funcionamiento interno del índice:

- btree: los índices btree usan árboles binarios (opción por defecto en innodb).
 - El tiempo de búsqueda es logarítmico (1000 filas = 10 pasos; 10000000 filas = 20 pasos, etc).
 - Permite comparadores <, >, <=, >=, =, !=, between y like.
- hash: usan una función que devuelve la posición en la tabla directamente.
 - El tiempo de búsqueda siempre es 1.
 - Solo permite comparadores = y !=.

Para el borrado de un índice el sintaxis es:

```
drop index <name> on <tabla>
```

1.2. Creación y borrado de índices

Por ejemplo:

```
create index indice1 on facturas(cliente,producto) using BTREE;
```

Adicionalmente se puede crear durante la creación de una tabla, o en un alter table:

```
create table A (a int primary key, b int, c int, index (b,c));
```

Y

```
create table B (a int primary key, b int, c int);
```

```
alter table B add index indice2 (b,c);
```

El borrado será mediante:

```
drop index indice2 on B;
```

2. Vistas

Las vistas son tablas virtuales que se almacenan en cada base de datos, como una tabla más. Sin embargo, las vistas no almacenan datos, no se puede insertar en ellas, sino que sus datos son realmente los datos de otras tablas.

Por ello se basan en una consulta que utiliza las tablas de la base de datos en la que se encuentra.

2.1. Creación y borrado

Las vistas se crean de la siguiente forma:

```
create view <nombre_vista> as <select>;
```

Este select puede ser cualquier select que pueda ser ejecutado. Para ver sus "datos" se hará un select donde la vista es usada como una tabla:

```
select * from <nombre_vista>;
```

Para borrar la visita solo es necesario:

```
drop view <nombre_vista>;
```