

# SQL ESTÁNDAR BÁSICO

ÍNDICES

Ing. Arturo Rangel Álvarez

Marzo, 2021

# Qué es un índice?

- ▶ Un índice es una especie de apuntador a una fila de una determinada tabla de la base de datos. No es más que una referencia que asocia el valor de una determinada columna (o conjunto de columnas) con las filas que contienen el valor (o valores).
- ▶ Los índices mejoran el tiempo de recuperación de los datos en las consultas realizadas a la base de datos.
- ▶ La creación de índices implica un aumento en el tiempo de ejecución sobre aquellas consultas de inserción, actualización y eliminación (ya que tendrán que actualizar el índice).
- ▶ Los índices necesitan un espacio para almacenarse, por lo que también tienen un coste adicional en forma de espacio en disco.

# Utilidad de los índices

- ▶ Encontrar las filas que cumplen la condición WHERE de la consulta cuyas columnas estén 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.
- ▶ Disminuir el tiempo de ejecución de las consultas con ordenación (ORDER BY) o agrupamiento (GROUP BY).
- ▶ 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.
- ▶ Si no existieran los índices, las consultas a la base de datos empezaría buscando por la primera fila de la tabla hasta la última. Esto implica que, cuanto más filas tenga la tabla, más tiempo tardará en realizar la consulta.

# Tipos de índices

- ▶ NDEX (NON-UNIQUE): 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.
- ▶ UNIQUE: 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.
- ▶ PRIMARY: 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.
- ▶ FULLTEXT: estos índices se emplean para realizar búsquedas sobre texto.
- ▶ SPATIAL: estos índices se emplean para realizar búsquedas sobre datos que componen formas geométricas representadas en el espacio.

# En qué estructuras se almacenan los índices?

- ▶ 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 utilizará 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].
- ▶ 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 utilizarán todas las columnas que componen el índice.

# Sintaxis INDEX

- ▶ «CREATE [UNIQUE | FULLTEXT | SPATIAL] INDEX index\_name ON table\_name (index\_col\_name...) index\_type;»
- ▶ Donde:
  - ▶ index\_name: es el nombre del índice.
  - ▶ table\_name: es el nombre de la tabla donde se va a crear el índice.
  - ▶ index\_col\_name: nombre de la columna (o columnas) que formarán el índice.
  - ▶ index\_type: es el tipo del índice. Se emplea con USING [BTREE | HASH].
- ▶ CREATE UNIQUE INDEX mi\_indice\_unico ON mi\_tabla (mi\_columna) USING HASH;

# Crear tabla con índices

- ▶ `CREATE TABLE usuario`
- ▶ `( clave INTEGER PRIMARY KEY,`
- ▶ `nombre VARCHAR(12),`
- ▶ `diasvigencia INT,`
- ▶ `fecha date`
- ▶ `);`
- ▶ `CREATE UNIQUE INDEX indx_nombre ON usuario (nombre) USING HASH;`
- ▶ `CREATE INDEX indx_fecha ON usuario (fecha);`