Bases de datos I

Módulo 4. SQL
Profesor: Lucas Catardo

Sesión 20. Buenas prácticas

Septiembre 23 de 2021

CREATE

- Utilizar VARCHAR en lugar de TEXT (hasta 8000 caracteres)
- Evaluar el uso de CHAR y VARCHAR. Utilizar CHAR cuando la longitud de los registros tiene poca variación.
- No usar columnas con tipos de datos FLOAT, REAL O DATETIME como FOREIGN KEY.
- Utilizar CONSTRAINT para mantener la integridad de los datos.
- Evitar claves primarias compuestas.

SELECT

- Evitar el uso de SELECT * FROM tabla. Especificar los campos requeridos.
- Especificar el alias de la tabla delante de cada campo definido en el SELECT.
- Evitar el uso de GROUP BY, DISTINCT y ORDER BY. Evaluar si estas tareas las puede hacer la aplicación que recibe los datos.

WHERE

- Evitar el uso de wildcards en LIKE como "%valor%", ya que el uso del comodín al inicio de la cadena obliga al DBMS a buscar en todos los registros.
- En subconsultas, utilizar EXISTS y NOT EXISTS en lugar de IN y NOT IN.
- Evitar utilizar funciones dentro de las condiciones del WHERE.

UNION

- Utilizar UNION ALL en lugar de UNION cuando se sabe que los registros no se repiten, para evitar la ejecución implícita de un DISTINCT.

CRUD

- Utilizar SET NOCOUNT ON en operaciones CRUD para evitar el conteo de las filas afectadas.

Orden de Procesamiento de una Query

¿Cómo se escribe?
SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY

¿Cómo se ejecuta?
FROM
WHERE
GROUP BY
HAVING
SELECT
ORDER BY

En detalle:
FROM
ON
JOIN
WHERE (El motor no interpreta los alias de columnas porque aún no existen).
GROUP BY (Ídem)
HAVING
SELECT (Se crean las columnas con alias)
DISTINCT
ORDER BY (Se pueden utilizar alias)
LIMIT (Es una opción de visualización, no de cálculo)

Índices

Es una estructura de datos que mejora la velocidad de las consultas, por medio de un identificador único de cada fila de una tabla, permitiendo un rápido acceso a los registros.

Ventajas
- Mejora el rendimiento de las consultas
- Mejora sustancialmente el rendimiento de las consultas que contienen agregaciones y combinaciones (GROUP BY, JOIN).

Desvantajas
- Las tablas utilizadas para almacenar los índices consumen espacio.
- Cuando se realizan operaciones de actualización, inserción o borrado, se requiere actualizar las tablas de los índices asociadas.

Recomendaciones:
- Evitar crear índices en tablas que se actualizan frecuentemente
- Usar claves cortas en lso índices agrupados. Los índices agrupados mejoran si se crean en columnas únicas o que no admiten valores NULL.

Tipos de índices

- Simple.
Definido sobre una sola columna.

CREATE INDEX "I_libros_autor"
ON "libros" (autor);

- Compuesto.
Formado por varias columnas de la misma tabla.

CREATE INDEX "I_libros_autoreditorial"
ON "libros" (autor, editorial);

- Agrupado (CLUSTERED)
Almacena los datos de las filas en orden. Solo se puede crear un único índice agrupado en una tabla de base de datos. Esto funciona de manera eficiente únicamente si los datos se ordenan en orden creciente o decreciente.

CREATE CLUSTERED INDEX "I_libros_autor"
ON "libros" (autor);

- No agrupado
Organiza los datos de forma aleatoria, pero el ínidice especifica internamente un orden lógico. El orden del índice no es el mismo que el ordenamiento físico de los datos. Los índices no agrupados funcionan bien con tablas donde los datos se modifican con frecuencia y el índice se crea en las columnas utilizadas en orden por las declaraciones WHERE y JOIN.

CREATE NONCLUSTERED INDEX "I_libros_autor"
ON "libros" (autor);

Sintaxis:

Crear un índice:
CREATE INDEX "nombre_indice"
ON "nombre_tabla" (nombre_columna);

Eliminar un índice:
ALTER TABLE "nombre_tabla"
DROP INDEX "nombre_indice";

Analizar y almacenar la distribución de claves para una tabla:
ANALYZE TABLE nombre_tabla;

Preguntas

- ¿Los textos largos y formateados de la base de datos se almacenan en la base de datos?
- ¿qué vemos en Bases de datos II?
- ¿qué función tienen los índices, cómo y cuándo se usan?
- ¿qué elementos adicionales hay que estudiar de bases de datos relacionales?
- ¿cómo se migra una base de datos de mysql, por ej, a postgreSQL?
- ¿qué diferencias significativas hay entre bases de datos relacionales?

Sincrónico

Subconsultas:

```
select * from video as v
where idVideo not in (select Video_idVideo from playlist_video where Video_idVideo = v.idVideo)
```

Es preferible usar not exists

```
select * from video as v
where not exists (select idVideo from playlist_video where Video_idVideo = v.idVideo)
```

Para eliminar índices con FK:

Refrescar los índices para que sean más performantes (se pueden haber fragmentado):
Reorganize
Rebuild
(Funciona en SQL Server)

CLUSTERED INDEX / NONCLUSTERED INDEX no se pueden generar en MySQL.

Sesión 21.

Sincrónico.

Todo lo que está en el SELECT debe estar en el GROUP BY.

Para filtrar datos nulos, la versión estándar es:

WHERE dato IS NULL;

La siguiente versión:
WHERE ISNULL(dato); puede no funcionar en algunos motores.