grupoesfera | INNOVACION Y DESARROLLO

Introducción al lenguaje de consulta estructurado (SQL)

Structured Query Language

Vistas

- Una vista es una consulta guardada sobre los datos
- No son parte del esquema físico de la base de datos
- Los datos se actualizan con cada consulta

Vistas

- Ventajas
  - Limitan la exposición de los datos
  - Permiten simplificar la vista de consultas complejas
  - Pueden usarse como tablas de resultados usando funciones de agregación
  - Ocupan muy poco espacio (sólo se guarda la consulta y no sus resultados)

Vistas

- Sintaxis
  - `CREATE VIEW JefePorSucursal AS SELECT e.nombre, e.apellido, s.nombre FROM Empleado e, Sucursal s WHERE e.idJefe IS NULL AND e.idSucursal = s.id`

Vistas (Ejercicios)

- E62: Crear 3 vistas de clientes, una para cada categoría
- E63: Crear una vista que muestre para cada empleado, su jefe y la sucursal donde trabaja
- E64: Crear una vista que muestre el total consumido por cada cliente
- E65: Crear una vista que muestre todos los datos de las facturas junto con el total de productos incluidos en cada una de ellas y el nombre y apellido del cliente.

Índices

- Son estructuras de datos que permiten acelerar la búsqueda de registros en las tablas
- Pero retrasan las escrituras
- Y ocupan espacio

Índices

- La búsqueda secuencial (full table scan) es muy ineficiente
  - En una tabla con 1000 registros donde buscar cada registro toma un segundo, una búsqueda afortunada toma 1 segundo... pero una búsqueda desafortunada toma 1000 segundos.
- En promedio, las búsquedas sin índice deben revisar la mitad de la tabla todas las veces
- Las búsquedas secuenciales son O(N) o de “tiempo lineal”

Índices

- Los índices permiten búsquedas de O(log(N)) o hasta O(1)
  - Para el ejemplo anterior, O(log(N)) haría que el peor tiempo fuera de 3 segundos
  - O(1) haría que fuera de un segundo

Índices

- Sintaxis
  - `CREATE INDEX indiceApellido ON Cliente(apellido)`
- Sintaxis para índices con unicidad
  - `CREATE UNIQUE INDEX indiceApellido ON Cliente(apellido)`
    - Previamente a la creación del índice, si la tabla ya tiene valores en apellido entonces estos deben ser únicos
    - Posteriormente a la creación del índice, no se pueden insertar valores que ya existan en la columna apellido
- Al usar PRIMARY KEY en la definición de un campo se crea un índice (único) implícito sobre esa columna

Índices

Ejemplos de consultas en donde el motor usa el índice:

- Cuando se busca por un valor constante
  - `SELECT nombre FROM Cliente WHERE nombre = 'Lavalle';`
- Cuando se busca usando JOINS contra una columna indexada
  - `SELECT * FROM Factura INNER JOIN Cliente ON Factura.idCliente = Cliente.id`
- Cuando se busca usando Wildcards (que no estén al inicio del parámetro de búsqueda)
  - `SELECT nombre FROM Cliente WHERE nombre = 'Lav%';`
- Ordenando los valores de una columna que tiene índice
  - `SELECT nombre FROM Cliente ORDER BY nombre;`

Índices Múltiples

- Ejemplo:
  - `CREATE INDEX indiceNombreYApellido ON Cliente(nombre,apellido);`
- Ejemplos definiendo dirección de orden del índice
  - `CREATE INDEX indiceNombreYApellido ON Cliente(nombre ASC, apellido DESC);`

Índices (Ejercicios)

- Buscar, para todas las consultas usadas desde el ejercicio E01 hasta el E45, la necesidad de crear índices, crearlos si es necesario
  - En MySQL se puede usar el comando 'EXPLAIN' para obtener información del optimizador sobre el plan de ejecución de la sentencia.
  - En SQL Server se puede usar 'SET SHOWPLAN_ALL (Transact-SQL)'