

INSTITUTO TECNOLÓGICO DE JEREZ

Andrea Yerovi Acevedo López

TOPICOS AVANZADOS

Salvador Acevedo

Ingeniería en Sistemas Computacionales IV



03/Abril/2020

Jerez de García Salinas, Zac

MySQL

1. *¿Qué son los índices en una base de datos?*

Un índice es una estructura de datos definida sobre una columna de tabla (o varias) y que permite localizar de forma rápida las filas de la tabla en base a su contenido en la columna indexada además de permitir recuperar las filas de la tabla ordenadas por esa misma columna.

2. *¿Para qué se utilizan?*

Permite localizar de forma rápida las filas de la tabla en base a su contenido en la columna indexada además de permitir recuperar las filas de la tabla ordenadas por esa misma columna.

3. *¿Qué tipos existen?*

Índice simple y compuesto, Índice agrupado y no agrupado, Índice único

4. *¿Cuáles son las mejores recomendaciones para el manejo de índices?*

- Hay que evitar crear demasiados índices en tablas que se actualizan con mucha frecuencia y procurar definirlos con el menor número de columnas posible. Es conveniente utilizar un número mayor de índices para mejorar el rendimiento de consultas en tablas con pocas necesidades de actualización, pero con grandes volúmenes de datos. Un gran número de índices contribuye a mejorar el rendimiento de las consultas que no modifican datos, como las instrucciones SELECT, ya que él [[optimizador
- De consultas]] dispone de más índices entre los que elegir para determinar el método de acceso más rápido. Se recomienda utilizar una longitud corta en la clave de los índices agrupados. Los índices agrupados también mejoran si se crean en columnas únicas o que no admitan valores NULL.
- Un índice único en lugar de un índice no único con la misma combinación de columnas proporciona información adicional al optimizador de consultas y, por tanto, resulta más útil.

5. *Comandos para la administración de índices*

```
DROP INDEX UQ_Name ON DiffIndexTypesDemo, INSERT INTO  
DiffIndexTypesDemo VALUES ('John', 'Ramtha')  
GO 100
```

6. *¿Cómo es la Reorganización de índices?*

Un factor clave para conseguir una E/S de disco mínima para todas las consultas de bases de datos es asegurarse de que se creen y se mantengan buenos índices. Un paquete puede usar la tarea Reorganizar índice para reorganizar los índices de una base de datos individual o de varias bases de datos.

7. *¿Cómo se aplica la Reorganización de índices?*

```
USE MyDatabase;  
GO
```

```
ALTER INDEX ALL ON MyTable REORGANIZE;  
GO
```

8. ¿Qué es la Reconstrucción de índices?

Se debe examinar y determinar qué índices son susceptibles de ser reconstruidos. Cuando un índice está descompensado puede ser porque algunas partes de éste han sido accedidas con mayor frecuencia que otras.

9. ¿Cómo se aplicase la Reconstrucción de índices?

```
ALTER INDEX <index_name> REBUILD;
```

Para reconstruir una partición de un índice podríamos hacer los siguientes:

```
ALTER INDEX <index_name> REBUILD PARTITION <nb_partition>  
NOLOGGING;
```

Comando ALTER INDEX

Como hemos comentado esta sentencia se utiliza para cambiar o reconstruir un Índice existente en la base de datos. Para reconstruir un Índice bastaría con lazar la siguiente sentencia: ALTER INDEX REBUILD;

10. Sintaxis para la creación de índices

```
CREATE [UNIQUE|FULLTEXT|SPATIAL] INDEX index_name  
[USING index_type]  
ON tbl_name (index_col_name,...)
```

index_col_name:

col_name [(length)] [ASC | DESC]

Ejemplo:

```
CREATE TABLE lookup (id INT) ENGINE = MEMORY;  
CREATE INDEX id_index USING BTREE ON lookup (id);
```

ORACLE

11. ¿Qué son los índices en una base de datos?

Un índice es una estructura de datos definida sobre una columna de tabla (o varias) y que permite localizar de forma rápida las filas de la tabla en base a su contenido en la columna indexada además de permitir recuperar las filas de la tabla ordenadas por esa misma columna.

12. ¿Para qué se utilizan?

Permite localizar de forma rápida las filas de la tabla en base a su contenido en la columna indexada además de permitir recuperar las filas de la tabla ordenadas por esa misma columna.

13. ¿Qué tipos existen?

Índice simple y compuesto, Índice agrupado y no agrupado, Índice único

14. ¿Cuáles son las mejores recomendaciones para el manejo de índices?

- Hay que evitar crear demasiados índices en tablas que se actualizan con mucha frecuencia y procurar definirlos con el menor número de columnas posible. Es conveniente utilizar un número mayor de índices para mejorar el

rendimiento de consultas en tablas con pocas necesidades de actualización, pero con grandes volúmenes de datos. Un gran número de índices contribuye a mejorar el rendimiento de las consultas que no modifican datos, como las instrucciones SELECT, ya que él [[optimizador

- De consultas]] dispone de más índices entre los que elegir para determinar el método de acceso más rápido. Se recomienda utilizar una longitud corta en la clave de los índices agrupados. Los índices agrupados también mejoran si se crean en columnas únicas o que no admitan valores NULL.
- Un índice único en lugar de un índice no único con la misma combinación de columnas proporciona información adicional al optimizador de consultas y, por tanto, resulta más útil.

15. Comandos para la administración de índices

```
create [bitmap | unique] index nombre_indice on
nombre_tabla (nombre_columna [, nombre_columna2] ...) [reverse];
select index_name Nombre, index_type Tipo,
table_name Tabla, tablespace_name Tablespace,
secondary Secundario
from all_indexes
where table_name = 'FACTURACION'
```

16. ¿Cómo es la Reorganización de índices?

Un factor clave para conseguir una E/S de disco mínima para todas las consultas de bases de datos es asegurarse de que se creen y se mantengan buenos índices. Un paquete puede usar la tarea Reorganizar índice para reorganizar los índices de una base de datos individual o de varias bases de datos.

17. ¿Cómo se aplica la Reorganización de índices?

```
ALTER INDEX <index_name> REBUILD;
```

Para reconstruir una partición de un índice podríamos hacer los siguientes:

```
ALTER INDEX <index_name> REBUILD PARTITION <nb_partition>
NOLOGGING;
```

Comando ALTER INDEX

18. ¿Qué es la Reconstrucción de índices?

Para reconstruir un índice de una tabla utilizaremos la cláusula rebuild con el comando alter index. Un ejemplo de reconstrucción de un índice en Oracle podría ser:

```
ALTER INDEX nombre_indice REBUILD;
```

19. ¿Cómo se aplicase la Reconstrucción de índices?

```
ALTER INDEX <index_name> REBUILD;
```

Para reconstruir una partición de un índice podríamos hacer los siguientes:

```
ALTER INDEX <index_name> REBUILD PARTITION <nb_partition>
NOLOGGING;
```

Comando ALTER INDEX

Como hemos comentado esta sentencia se utiliza para cambiar o reconstruir un Índice existente en la base de datos. Para reconstruir un Índice bastaría con lazar la siguiente sentencia: ALTER INDEX REBUILD;

20. Sintaxis para la creación de índices

```
CREATE [UNIQUE] INDEX index_name  
ON table_name (column1, column2, ... column_n)  
[ COMPUTE STATISTICS ];
```

Ejemplo:

```
CREATE INDEX supplier_idx  
ON supplier (supplier_name);
```

ORACLE

REFERENCIAS

V. (2020). 4.3 Índices, reorganización y reconstrucción. Retrieved 3 April 2020, from <http://administraciondebasededatos123456789.blogspot.com/2017/06/43-indices-reorganizacion-y.html>

Reconstrucción de índices – Orasite.com. (2020). Retrieved 3 April 2020, from <https://orasite.com/tutoriales/administracion/reconstruccion-de-indices>

Índice de base datos - EcuRed. (2020). Retrieved 3 April 2020, from https://www.ecured.cu/%C3%8Dndice_de_base_datos