

V. RESTRICCIONES E INDICES

CREANDO ÍNDICES

A continuación, deberá añadir datos a las tablas, sin embargo, si nunca recupera los datos de la tabla, usted está simplemente desperdiciando espacio de almacenamiento. SQL Server no necesita tener índices en una tabla para recuperar los datos. Sin embargo, la mayoría de las organizaciones almacenan grandes cantidades de datos en una tabla y tienen la necesidad de poder recuperar datos de forma instantánea. Para permitir una rápida recuperación de datos y al mismo tiempo que se garantice que el rendimiento no disminuirá a medida que los usuarios agreguen registros a una tabla, es necesario agregar los índices a sus tablas.

Los índices no son un nuevo concepto de una base de datos estrictamente. Utilizamos los índices día a día, en la parte de atrás de algún libro, usted encontrará un índice en forma impresa. Si quería leer sobre algún tema en específico, se puede encontrar la información de dos formas distintas. Usted puede abrir el libro, desde la página 1, y escanear cada página hasta que encontró la información que necesitaba. O bien, podrá ver el índice en la contraportada del libro, localizar el tema que necesitaba, y luego ir directamente a la página correspondiente. Puede encontrar la información en ambos sentidos, pero usando el índice es mucho más eficiente. En este tema, se explorará la manera que SQL Server construye y utiliza los índices para garantizar la rápida recuperación de datos y el rendimiento de la estabilidad. Usted puede entonces aprender a construir en cluster, nonclustered, que abarcan los índices en sus tablas para alcanzar el equilibrio óptimo entre la velocidad y el mantenimiento necesario.

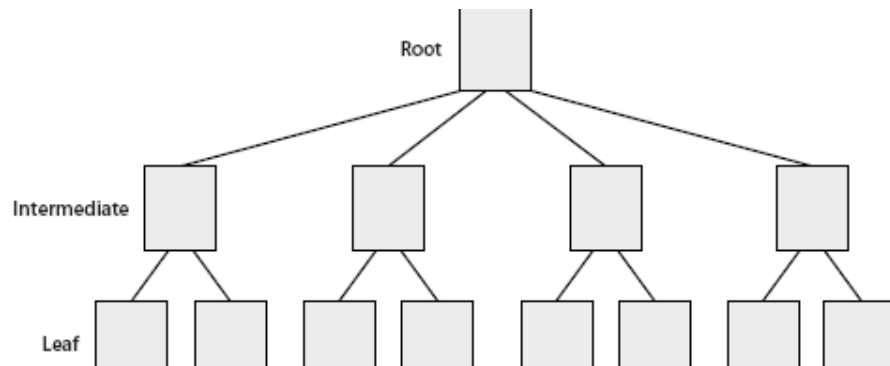
Comprendiendo la estructura de los índices

Un índice sólo es útil si puede ayudar a encontrar rápidamente los datos, independientemente del volumen de datos almacenado. Tomando un libro como ejemplo el índice contiene sólo una pequeña muestra de las palabras en el libro, por lo que proporciona una forma compacta para buscar información. Si el índice se organiza sobre la base en el que una palabra aparece en las páginas, usted tendría que leer muchos registros y las páginas para encontrar su información. En lugar de ello, los índices están organizados en orden alfabético, lo que significa que usted puede ir a un lugar específico en el índice para encontrar lo que necesita. La forma en que un índice es organizado en SQL Server es muy similar. En esta lección, podrá ver cómo SQL Server utiliza la

estructura de árbol para construir índices que proporciona rápida recuperación de datos, incluso con tablas muy grandes.

Explorando B-Trees

La estructura que SQL Server utiliza para construir y mantener los índices se llama un árbol balanceado, o B-tree. La ilustración en la Figura muestra un ejemplo de un B-tree.

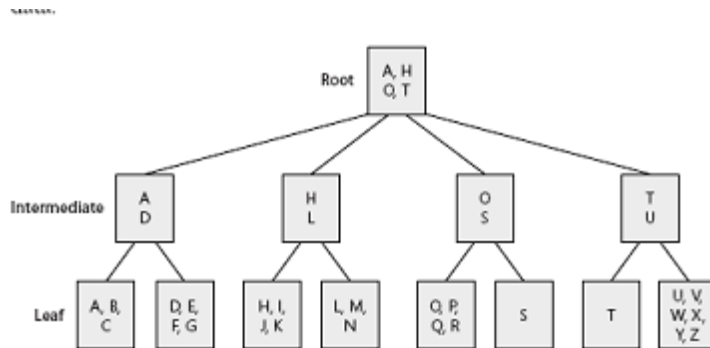


Un B-tree consiste en un nodo raíz que contiene una página de datos, cero o más niveles intermedios que contiene páginas adicionales.

La primera página contiene registros ordenados, en el orden que corresponden a los datos que están indexados. El número de índice de filas en una página está determinado por el espacio de almacenamiento requerido por las columnas definidas en el índice. Por ejemplo, un índice definido sobre una columna de 4-byte tendrá cinco veces más valores por página, si se define un índice de una columna char (60), requiere 60 bytes de almacenamiento por página.

SQL Server crea los niveles intermedios mediante la adopción de la primera entrada en cada página y almacena los registros en una página con un puntero a la página de donde se tomó la entrada. La página raíz se construye de la misma manera.

Mediante la construcción de un índice de esta manera, SQL Server puede buscar las tablas que tienen billones de filas de datos igual de rápido que si la tabla fuera de unos pocos cientos de registros de datos. La figura siguiente amplía más la idea de cómo son construidas las páginas en los diferentes niveles.



Si se busca el término “SQL Server”, la consulta sería de exploración de la página raíz. Encontrará el valor O, así como el valor T. Debido a que S viene antes de T, la consulta sabe que tiene que mirar en la página de O para encontrar los datos que necesita.

LA CREACIÓN DE ÍNDICES EN CLUSTER

El primer tipo de índice que se debería crear en una tabla es un índice agrupado. Como norma general, cada tabla debe tener un índice agrupado. Y cada tabla sólo puede tener un índice agrupado. En este tema se verá cómo crear un índice agrupado mediante el uso de la sentencia CREATE INDEX que es un comando Transact-SQL, incluyendo las opciones que puede especificar para el comando. Usted también aprenderá a desactivar y luego volver a agrupar este tipo de índice.

Implementando índices agrupados (CLUSTER)

Las columnas que usted defina para un índice agrupado se denominarán la agrupación clave. Un cluster causa que SQL Server pueda ordenar los datos en la tabla de acuerdo a la clave de la agrupación. Debido a que en una tabla solo se pueden ordenar en una única dirección, puede crear un único índice agrupado sobre una tabla.

La sintaxis general para la creación de un índice agrupado es:

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX index_name
```

```
ON <object> ( column [ ASC | DESC ] [ ,...n ] )
```

```
[ INCLUDE ( column_name [ ,...n ] ) ]
```

```
[ WITH ( <relational_index_option> [ ,...n ] ) ]
```

```

[ ON { partition_scheme_name ( column_name)

      | filegroup_name

      | default

      }

][ ; ]

```

Las opciones CLUSTERED y NONCLUSTERED designan el tipo de índice que se va a crear. Vamos a cubrir la opción NONCLUSTERED en el tema siguiente.

Después de que especifique que desea crear un índice agrupado, es necesario especificar un nombre para el índice. Cada índice debe tener un nombre que se ajuste a las normas de identificadores de objeto.

A continuación, se utiliza la cláusula ON para especificar el objeto sobre el cual debe crear el índice. Después de especificar la tabla o vista, debe especificar en el paréntesis en las columnas que desea crear el índice. Las palabras clave ASC y DESC especifican si el orden debe ser ascendente o descendente.

Deshabilitando un índice

Se puede deshabilitar utilizando la sentencia de SQL "ALTER INDEX", así como se muestra en la siguiente sintaxis:

```

ALTER INDEX { index_name | ALL }

      ON <object>

      DISABLE [ ; ]

```

Al desactivar un índice, el índice de definición permanece en el sistema de catálogo, pero SQL Server ya no lo utiliza. SQL Server no mantiene el índice como en los datos en la tabla de cambios, y el índice no puede ser utilizado para satisfacer las consultas. Y si desactiva un índice agrupado, todo el cuadro se convierte en inaccesible.

Habilitando un índice

Para habilitar un índice, debe eliminarlo y volver a regenerar y poblar la estructura de árbol. Puede hacerlo utilizando el siguiente comando ALTER INDEX, que se utiliza en la cláusula REBUILD:

```
ALTER INDEX { index_name | ALL }  
  
    ON <object>  
  
    REBUILD [ ; ]
```

Creando índices no agrupados (NONCLUSTERED)

Debido a que un índice nonclustered no impone un orden de clasificación en una tabla, puede crear hasta 249 índices nonclustered en una única tabla. Índices Nonclustered, al igual que los índices agrupados, creara una estructura de B-tree. Sin embargo, a diferencia de un índice agrupado, en un índice Nonclustered, el nivel de la hoja índice contiene un puntero a los datos en lugar de los datos reales.

Si la tabla tiene un índice agrupado, el puntero apunta a la clave de agrupación. Si la tabla no tiene un índice agrupado, el puntero es un identificador relativo (RID), que es una referencia a la ubicación física de los datos dentro de una página de datos.

Cuando la consulta llega a la hoja de nivel, usa el puntero para encontrar la clave de la agrupación. Le pregunta entonces al índice agrupado para llegar a la fila de datos. Si un índice agrupado no existe en la tabla, el puntero devuelve un RID, que causa que SQL Server utilice para escanear un mapa de asignación interna para localizar la página de referencia del RID a fin de que puedan regresar los datos solicitados.

Puede utilizar el mismo comando CREATE INDEX... para crear un índice nonclustered como usted lo hizo para crear un índice agrupado, salvo que debe especificar la palabra clave NONCLUSTERED.