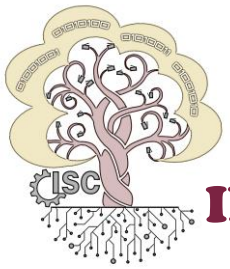


INSTITUTO TECNOLOGICO SUPERIOR DE JEREZ

Ingeniería en Sistemas Computacionales

6to Semestre

Alumno:

Daniel Alejandro de la Rosa Castañeda

NC:16070126

Materia:

Administración de Bases de Datos

Nombre del trabajo

Cuestionario

Profesor:

ISC Salvador Acevedo Sandoval

1. Asignación de espacio en disco para base de datos

La forma en la que se configura el espacio en disco duro es de la siguiente forma:

```
USE master;
GO
ALTER DATABASE AdventureWorks2012
MODIFY FILE
    (NAME = test1dat3,
    SIZE = 20MB);
GO
```

(Microsoft, 2017)

2. Asignación de espacio en disco para tablas

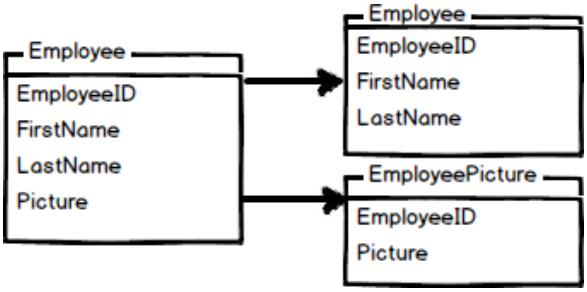
SQL Server no permite la asignación de espacio para las tablas, ya que lo hace de una forma predeterminada (Microsoft, 2017).

4. particionamiento

En SQL Server existen dos tipos de particionamiento: vertical y horizontal.

Particionamiento Vertical.

El particionamiento vertical de tablas es principalmente usado para incrementar el desempeño de SQL Server especialmente en casos cuando una consulta retorna todas las columnas de una tabla que contiene un número de columnas de texto muy amplio o BLOB. En este caso, para reducir los tiempos de acceso, las columnas BLOB pueden ser divididas a su propia tabla. Otro ejemplo es restringir el acceso a datos sensibles, por ejemplo, contraseñas, información salarial, etc. La partición vertical divide una tabla en dos o más tablas que contienen diferentes columnas:



Ejemplo de particionamiento Vertical.

Un ejemplo de particionamiento vertical puede ser una tabla grande con reportes para empleados que contiene información básica, como el nombre del reporte, el id, el número de reporte y una gran columna con la descripción del reporte. Asumiremos que un ~95% de los usuarios están buscando en la parte del nombre del reporte, el número, etc., y que sólo un ~5% de las peticiones están abriendo el campo de las descripciones de los reportes y viendo la descripción. Asumamos que todas esas búsquedas conducirán a los escaneos del índice agrupado y dado que los escaneos del índice leen todas las filas en la tabla el costo de la consulta es proporcional al número total de

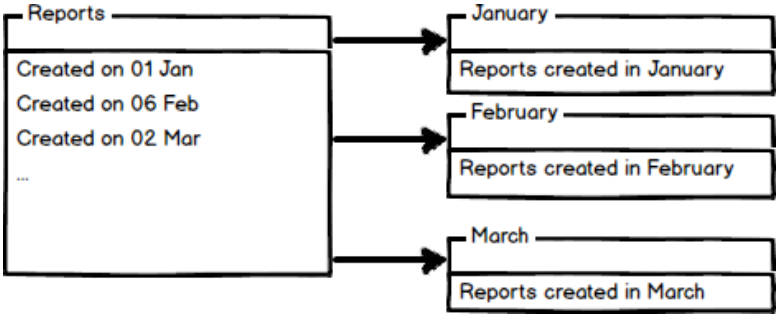
filas en la tabla y nuestro objetivo es minimizar el número de operaciones IO y reducir el costo de la búsqueda.

Particionamiento Horizontal

El particionamiento horizontal divide una tabla en múltiples tablas que contienen el mismo número de columnas, pero menos filas. Por ejemplo, si una tabla contiene un gran número de filas que representan reportes mensuales podría ser particionada horizontalmente en tablas por años, con cada tabla representando todos los reportes para un año específico. De esta manera las consultas que requieren datos para un año específico sólo referenciarán la tabla apropiada. Las tablas deberían ser particionadas en una manera que las consultas referencian tan pocas tablas como sea posible.

Las tablas son particionadas horizontalmente basadas en una columna que será usada para particionar y los rangos asociados a cada partición. La columna de particionamiento es usualmente una columna de fecha pero todos los tipos de datos que son válidos para usarse como columnas de índice pueden ser usados como columna de partición, excepto columnas timestamp. Los siguientes tipos de datos no pueden ser especificados: ntext, text, image, xml, varchar(max), nvarchar(max), o varbinary(max), el tipo definido por el usuario Microsoft .NET Framework common language runtime (CLR), columnas de tipo de datos de alias.

Hay dos enfoques diferentes que podríamos usar para lograr la partición de la tabla. El primero es crear una nueva tabla particionada y simplemente copiar los datos desde su tabla existente en la nueva tabla y renombrarla. El segundo enfoque es particionar una tabla existente reconstruyendo o creando un índice agrupado en la tabla.



Para crear una tabla particionada para almacenar reportes mensuales primero crearemos grupos de archivos adicionales. Un grupo de archivos es una unidad de almacenamiento lógica. Cada base de datos tiene un grupo de archivos que contiene el archivo de datos primario (.mdf). Un grupo de archivos adicional y definido por el usuario puede ser creado para contener archivos secundarios (.ndf). Nosotros crearemos 12 grupos de archivos por cada mes:

```
ALTER DATABASE PartitioningDB
ADD FILEGROUP January
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP February
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP March
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP April
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP May
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP June
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP July
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP Avgust
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP September
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP October
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP November
GO
ALTER DATABASE PartitioningDB
ADD FILEGROUP December
GO
```

	AvailableFilegroups
1	PRIMARY
2	January
3	February
4	March
5	April
6	May
7	June
8	July
9	Avgust
10	September
11	October
12	November
13	December

https://www.sqlshack.com/es/particionamiento-de-tablas-de-bases-de-datos-en-sql-server/

Bibliografía

Microsoft. (18 de Junio de 2017). Obtenido de Microsoft: https://docs.microsoft.com/es-es/sql/sql-server/maximum-capacity-specifications-for-sql-server?view=sql-server-2017

Microsoft. (13 de Marzo de 2017). Obtenido de Microsoft: https://docs.microsoft.com/es-es/sql/relational-databases/databases/increase-the-size-of-a-database?view=sql-server-2017