

# Creación y Mantenimiento de Índices

## Contenido

Introducción	1
Creación de índices	2
Opciones de creación de índices	11
Mantenimiento de índices	16
Introducción a las estadísticas	25
Consultas en la tabla sysindexes	36
Configuración de índices mediante el Asistente para optimización de índices	38
Consideraciones acerca del rendimiento	41

## Notas para el instructor

Este módulo ofrece a los alumnos una introducción a la creación y mantenimiento de los índices con las opciones de CREATE INDEX. Describe cómo los procedimientos de mantenimiento cambian físicamente los índices. El módulo explica las herramientas de mantenimiento y describe el uso de estadísticas en Microsoft® SQL Server™ 2000. También describe formas de comprobar el uso de índices y trata el modo de saber que funcionan correctamente. El módulo concluye con una explicación de cuándo utilizar el Asistente para optimización de índices.

Después de completar este módulo, los alumnos serán capaces de:

- Crear índices y vistas indizadas con características de unicidad o composición.
- Utilizar las opciones de CREATE INDEX
- Describir cómo mantener índices a lo largo del tiempo.
- Describir cómo el optimizador de consultas crea, almacena, mantiene y utiliza estadísticas para optimizar consultas.
- Consultar la tabla **sysindexes**
- Describir cómo funciona el Asistente para optimización de índices y cuándo se debe utilizar.
- Describir los aspectos relacionados con el rendimiento que afectan a la creación y mantenimiento de índices.

# Introducción

**Objetivo del tema**

Proporcionar una introducción a los temas y objetivos del módulo.

**Explicación previa**

En este módulo aprenderá a crear y mantener índices.

- Creación de índices
- Opciones de creación de índices
- Mantenimiento de índices
- Introducción a las estadísticas
- Consultas en la tabla sysindexes
- Configuración de índices mediante el Asistente para optimización de índices
- Consideraciones acerca del rendimiento

Al programar una base de datos, deseará crear índices útiles que le permitan obtener un rápido acceso a los datos. Con Microsoft® Windows® 2000, puede crear y mantener índices y estadísticas. Cuando se utiliza el Asistente para optimización de índices, Microsoft SQL Server™ 2000 crea índices, analiza las consultas y determina los índices que se deben crear.

Después de completar este módulo, el alumno será capaz de:

- Crear índices y vistas indizadas con características de unicidad o composición.
- Utilizar las opciones de CREATE INDEX
- Describir cómo mantener índices a lo largo del tiempo.
- Describir cómo el optimizador de consultas crea, almacena, mantiene y utiliza estadísticas para optimizar consultas.
- Consultar la tabla **sysindexes**
- Describir cómo funciona el Asistente para optimización de índices y cuándo se debe utilizar.
- Describir los aspectos relacionados con el rendimiento que afectan a la creación y mantenimiento de índices.

## ◆ Creación de índices

**Objetivo del tema**

Presentar los temas acerca de la creación de índices.

**Explicación previa**

Ahora que se ha familiarizado con las distintas arquitecturas de índices, veamos cómo crear índices y obtener información acerca de los índices existentes.

- Creación y eliminación de índices
- Creación de índices únicos
- Creación de índices compuestos
- Creación de índices en columnas calculadas
- Obtención de información acerca de los índices existentes

---

Ahora que se ha familiarizado con las distintas arquitecturas de índices, explicaremos cómo crear y eliminar índices, y cómo obtener información acerca de los índices existentes.

## Creación y eliminación de índices

### Objetivo del tema

Presentar las instrucciones CREATE INDEX y DROP INDEX.

### Explicación previa

Los índices se crean con la instrucción CREATE INDEX y se pueden quitar con la instrucción DROP INDEX.

#### ■ Uso de la instrucción CREATE INDEX

- Los índices se crean automáticamente en tablas con las restricciones PRIMARY KEY o UNIQUE
- Los índices pueden crearse en vistas si cumplen ciertos requisitos

```
USE Northwind
CREATE CLUSTERED INDEX CL_lastname
ON employees(lastname)
```

#### ■ Uso de la instrucción DROP INDEX

```
USE Northwind
DROP INDEX employees.CL_lastname
```

Los índices se crean con la instrucción CREATE INDEX y se pueden quitar con la instrucción DROP INDEX.

**Nota** Para ejecutar cualquiera de estas instrucciones en una base de datos, es necesario ser el propietario de la tabla.

### Uso de la instrucción CREATE INDEX

La instrucción CREATE INDEX permite crear índices. También puede usar el Asistente para creación de índices del Administrador corporativo de SQL Server. Al crear un índice basado en una o varias columnas de una tabla, tenga en cuenta los siguientes hechos e instrucciones:

- SQL Server crea índices automáticamente cuando se crea una restricción PRIMARY KEY o UNIQUE en una tabla. La definición de una restricción PRIMARY KEY o UNIQUE es preferible a la creación de índices ordinarios.
- Para ejecutar la instrucción CREATE INDEX es necesario ser el propietario de la tabla.
- Es posible crear índices en las vistas.
- SQL Server almacena la información de los índices en la tabla del sistema **sysindexes**.
- Antes de crear un índice basado en una columna, determine qué índices hay ya basados en ella.

- Utilice índices de tamaño reducido; para ello, defínalos con columnas de pequeño tamaño. Normalmente, los índices pequeños son más eficientes que los que tienen valores de clave mayores.
- Seleccione las columnas por su unicidad, de forma que cada valor de clave identifique un número pequeño de filas.
- Al crear un índice agrupado se vuelven a generar todos los índices no agrupados existentes.

**Sintaxis**

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ]
INDEX nombreÍndice ON { tabla | vista } ( columna [ ASC | DESC ] [ ,...n ] )
[WITH
[PAD_INDEX ]
[[,] FILLFACTOR = factorDeRelleno ]
[[,] IGNORE_DUP_KEY ]
[[,] DROP_EXISTING ]
[[,] STATISTICS_NORECOMPUTE ]
[[,] SORT_IN_TEMPDB ]
]
[ON grupoDeArchivos ]
```

**Ejemplo 1**

En este ejemplo se crea un índice agrupado con la columna **Lastname** de la tabla **Employees**.

```
CREATE CLUSTERED INDEX CL_lastname
ON employees(lastname)
```

**Uso de la instrucción DROP INDEX**

La instrucción DROP INDEX permite quitar un índice de una tabla. Al quitar un índice, tenga en cuenta los siguientes hechos:

- Al ejecutar la instrucción DROP INDEX, SQL Server reclama el espacio de disco que ocupaba el índice.
- No es posible utilizar la instrucción DROP INDEX con índices creados con la restricción PRIMARY KEY o UNIQUE. Para quitar estos índices, primero es necesario quitar la restricción.
- Al quitar una tabla se quitan también todos sus índices.
- Al quitar un índice agrupado, todos los índices no agrupados de la tabla se vuelven a generar automáticamente.
- Para quitar un índice es necesario encontrarse en la base de datos en la que reside ese índice.
- La instrucción DROP INDEX no se puede utilizar en las tablas de sistema.

**Sintaxis**

```
DROP INDEX 'tabla.índice | vista.índice' [ , ...n ]
```

**Ejemplo 2**

En este ejemplo se quita el índice **cl\_lastname** de la tabla **Employees**.

```
USE Northwind
DROP INDEX employees.CL_lastname
```

## Creación de índices únicos

### Objetivo del tema

Explicar cómo los índices únicos aseguran que las columnas indizadas son únicas.

### Explicación previa

En un índice único no se permite que dos filas tengan el mismo valor de clave.

```
USE Northwind
CREATE UNIQUE NONCLUSTERED INDEX U_CustID
ON customers(CustomerID)
```

Customers			
CustomerID	CompanyName	ContactName	...
QUICK	QUICK-Stop	Horst Kloss	
BONAP	Bon app'	Laurence Lebihan	
RANCH	Rancho grande	Sergio Gutiérrez	

**Los valores de clave duplicados no se permiten cuando se agrega a la tabla una fila nueva**

RANCH	Santé Gourmet	Jonas Bergulfsen	...
-------	---------------	------------------	-----

Un índice *único* asegura que todos los datos de una columna indizada sean únicos y que no haya valores duplicados.

Los índices únicos aseguran que los datos de las columnas indizadas son únicos. Si la tabla tiene una restricción **PRIMARY KEY** o **UNIQUE**, SQL Server crea automáticamente un índice único al ejecutar la instrucción **CREATE TABLE** o **ALTER TABLE**.

### Cómo asegurar que los datos de las columnas indizadas son únicos

Puede crear un índice único, ya sea agrupado o no agrupado, cuando los propios datos sean inherentemente únicos.

Sin embargo, si es necesario exigir la unicidad, es preferible crear restricciones **PRIMARY KEY** o **UNIQUE** que crear un índice único. Al crear un índice único, tenga en cuenta los siguientes hechos e instrucciones:

- SQL Server crea automáticamente índices únicos basados en las columnas de la tabla definidas con las restricciones **PRIMARY KEY** o **UNIQUE**.
- Si una tabla contiene datos, SQL Server comprueba si hay valores duplicados al crear el índice.
- SQL Server comprueba si hay valores duplicados cada vez que se utiliza la instrucción **INSERT** o **UPDATE**. Si hay valores de clave duplicados, SQL Server cancela la instrucción y genera un mensaje de error con el primer duplicado.
- Asegúrese de que cada fila tenga un valor único; no puede haber dos filas con el mismo número de identificación si se crea un índice único basado en esa columna. Esta norma asegura que cada entidad se pueda identificar de forma única.
- Sólo debe crear índices únicos en las columnas en las que pueda exigirse la integridad. Por ejemplo, no es conveniente crear un índice único en la columna **LastName** de la tabla **Employees**, ya que puede haber varios miembros con los mismos apellidos.

**Ejemplo 1**

En este ejemplo se crea un índice único no agrupado llamado **U\_CustID** en la tabla **Customers**. El índice se basa en la columna **CustomerID**. El valor de la columna **CustomerID** debe ser único para cada fila de la tabla.

```
USE Northwind
CREATE UNIQUE NONCLUSTERED INDEX U_CustID
ON customers(CustomerID)
```

**Búsqueda de todos los valores duplicados de una columna**

Si en el momento de crear un índice único ya hay valores de clave duplicados, la instrucción **CREATE INDEX** produce un error. SQL Server devuelve un mensaje de error con el primer valor duplicado, pero puede haber otros. Puede usar la secuencia de comandos siguiente en cualquier tabla para buscar todos los valores duplicados de una columna. Reemplace el texto en cursiva por información específica de su consulta.

```
SELECT index_col, COUNT (index_col)
FROM tablename
GROUP BY index_col
HAVING COUNT(index_col)>1 ORDER BY index_col
```

**Ejemplo 2**

Este ejemplo determina si hay identificaciones de cliente duplicadas en la columna **CustomerID** de la tabla **Customers**. En caso de que las haya, SQL Server devuelve la identificación de cliente y el número de entradas duplicadas en el conjunto de resultados.

```
SELECT CustomerID, COUNT(CustomerID) AS '# of Duplicates'
FROM Northwind.dbo.Customers
GROUP BY CustomerID
HAVING COUNT(CustomerID)>1
ORDER BY CustomerID
```

**Resultado**

<b>CustomerID</b>	<b># of Duplicates</b>
-------------------	------------------------

(0 filas afectadas)



## Creación de índices compuestos

### Objetivo del tema

Explicar el funcionamiento de los índices compuestos.

### Explicación previa

Los índices compuestos especifican más de una columna como valor de clave.

```
USE Northwind
CREATE UNIQUE NONCLUSTERED INDEX U_OrdID_ProdID
ON [Order Details] (OrderID, ProductID)
```

Order Details				
OrderID	ProductID	UnitPrice	Quantity	Discount
10248	11	14.000	12	0.0
10248	42	9.800	10	0.0
10248	72	34.800	5	0.0

Diagram illustrating a composite index on the **Order Details** table. Two pink arrows point from the **OrderID** and **ProductID** columns to a box labeled **Clave compuesta** (Composite Key). The columns are labeled **Columna 1** and **Columna 2** respectively.

### Sugerencia

Señale que la columna **OrderID** es la primera por ser más selectiva que la columna **ProductID**.

Los índices *compuestos* especifican más de una columna como valor de clave. Puede crear índices compuestos:

- Cuando resulte mejor buscar claves formadas por dos o más columnas.
- Si las consultas sólo hacen referencia a las columnas del índice.

Por ejemplo, una guía telefónica es un buen ejemplo de utilidad de un índice compuesto. La guía está ordenada por apellidos. En cada apellido, la ordenación es por nombres, ya que a menudo hay entradas con los mismos apellidos.

Al crear un índice compuesto, tenga en cuenta los siguientes hechos e instrucciones:

- Es posible combinar hasta 16 columnas en un mismo índice compuesto. La suma de las longitudes de las columnas que forman el índice no puede superar 900 bytes.
- Todas las columnas de un índice compuesto deben proceder de la misma tabla, excepto cuando el índice se crea en una vista.
- Defina en primer lugar la columna con mayor unicidad. La primera columna definida en la instrucción **CREATE INDEX** se conoce como columna de *orden superior*.
- La cláusula **WHERE** de una consulta debe hacer referencia a la primera columna del índice compuesto para que el optimizador de consultas lo utilice.
- Un índice basado en (**columna1**, **columna2**) no es igual a uno basado en (**columna2**, **columna1**): cada uno tiene un orden de columnas distinto. La columna que contenga datos más selectivos o que devuelva el menor porcentaje de filas determina, a menudo, el orden de las columnas.

- Los índices compuestos son útiles para las tablas con claves de varias columnas.
- Puede utilizar índices compuestos para mejorar el rendimiento de las consultas y reducir el número de índices creados en una tabla.

---

**Nota** Normalmente, no es útil tener varios índices basados en las mismas columnas.

---

### Ejemplo

En este ejemplo se crea un índice compuesto no agrupado basado en la tabla **Order Details**. Las columnas **OrderID** y **ProductID** son los valores de clave compuesta. Observe que la columna **OrderID** aparece en primer lugar por ser más selectiva que la columna **ProductID**.

```
USE Northwind
CREATE UNIQUE NONCLUSTERED INDEX U_OrdID_ProdID
ON [Order Details] (OrderID, ProductID)
```

## Creación de índices en columnas calculadas

### Objetivo del tema

Explicar los requisitos para crear un índice en una columna calculada.

### Explicación previa

Para poder crear un índice en una columna calculada, es necesario cumplir ciertos requisitos.

- Es posible crear índices en columnas calculadas cuando se dan las siguientes circunstancias:
  - La expresión de la columna calculada es determinista y precisa
  - La opción de conexión ANSI\_NULL está activada
  - La columna calculada no puede evaluar los tipos de datos **text**, **ntext** o **image**
  - Las opciones SET requeridas se activan cuando crea el índice y cuando las instrucciones INSERT, UPDATE o DELETE cambian el valor del índice
  - Se desactiva la opción NUMERIC\_ROUNDABORT
- Es posible que el optimizador de consultas ignore un índice en una columna calculada

### Para su información

Señale que los Libros en pantalla de SQL Server contienen información adicional acerca de las funciones deterministas y no deterministas.

Es posible crear índices en columnas calculadas cuando se dan las siguientes circunstancias:

- La expresión de la columna calculada es *determinista*. Las expresiones deterministas siempre devuelven el mismo resultado.
- La opción de conexión ANSI\_NULL está activada cuando se ejecuta la instrucción CREATE TABLE. La función OBJECTPROPERTY informa de que la opción está activada a través de la propiedad **IsAnsiNullsOn**.
- La expresión de columna calculada que se define para la columna calculada no puede evaluar los tipos de datos **text**, **ntext** o **image**.
- La conexión en la que se crea el índice y todas las conexiones que intentan instrucciones INSERT, UPDATE o DELETE que van a cambiar valores del índice tienen seis opciones SET activadas y una desactivada. Las opciones siguientes deben estar activadas:
  - ANSI\_NULLS
  - ANSI\_PADDING
  - ANSI\_WARNINGS
  - ARITHABORT
  - CONCAT\_NULL\_YIELDS\_NULL
  - QUOTED\_IDENTIFIER
- Además de estas opciones activadas, la opción NUMERIC\_ROUNDABORT debe estar desactivada.

### Para su información

Señale que los Libros en pantalla de SQL Server contienen información adicional acerca de las opciones SET que afectan a los resultados.

---

**Nota** El optimizador de consultas omite un índice basado en una columna calculada en cualquier instrucción SELECT que se ejecute con una conexión que no tiene esa misma configuración de opciones.

---

## Obtención de información acerca de los índices existentes

### Objetivo del tema

Explicar cómo obtener información acerca de los índices existentes.

### Explicación previa

Antes de crear, modificar o quitar un índice puede ser necesario obtener información acerca de los índices existentes.

- Uso del procedimiento almacenado del sistema **sp\_helpindex**

```
USE Northwind
EXEC sp_helpindex Customers
```

- Uso del procedimiento almacenado de sistema **sp\_help *nombreTabla***

Antes de crear, modificar o quitar un índice puede ser necesario obtener información acerca de los índices existentes.

### Uso del procedimiento almacenado del sistema **sp\_helpindex**

Puede utilizar el Administrador corporativo de SQL Server o ejecutar el procedimiento almacenado de sistema **sp\_helpindex** para obtener información acerca de un índice, como su nombre, su tipo o sus opciones en una tabla determinada.

### Ejemplo

En este ejemplo se muestran los índices de la tabla **Customers**.

```
USE Northwind
EXEC sp_helpindex Customers
```

### Resultado

index_name	index_description	index_keys
PK_Customers	clustered, unique, Primary Key located on PRIMARY	CustomerID
PostalCode	nonclustered located on PRIMARY	PostalCode
City	nonclustered located on PRIMARY	City

(1 filas afectadas)

### Sugerencia

Muestre el conjunto de resultados de **sp\_help *nombreTabla***; para ello, ejecute lo siguiente:

```
USE Northwind
EXEC sp_help Customers
```

Señale la información adicional de la tabla.

### Uso del procedimiento almacenado de sistema **sp\_help *nombreTabla***

También puede ejecutar el procedimiento almacenado de sistema **sp\_help *nombreTabla*** para obtener información acerca de los índices y otros datos de la tabla.

## ◆ Opciones de creación de índices

**Objetivo del tema**

Proporcionar una breve introducción a algunas de las opciones de creación de índices.

**Explicación previa**

SQL Server ofrece opciones de creación de índices que pueden acelerar este proceso y, también, mejorar su rendimiento a lo largo del tiempo.

- Uso de la opción FILLFACTOR
- Uso de la opción PAD\_INDEX

---

SQL Server ofrece opciones de creación de índices que pueden acelerar este proceso y, también, mejorar su rendimiento a lo largo del tiempo.

## Uso de la opción FILLFACTOR

### Objetivo del tema

Determinar cuándo y cómo utilizar la opción FILLFACTOR para aumentar el rendimiento.

### Explicación previa

Puede utilizar la opción FILLFACTOR para optimizar el rendimiento de las instrucciones INSERT y UPDATE en las tablas que contienen índices agrupados o no agrupados.

- Especifica cuánto debe llenarse la página
- Afecta a las páginas del nivel de hoja

#### Páginas de datos llenas

Con	...	470401
Funk	...	470402
White	...	470403
Rudd	...	470501
White	...	470502
Barr	...	470503

Akhtar	...	470601
Funk	...	470602
Smith	...	470603
Martin	...	470604
Smith	...	470701
Ota	...	470702

Martin	...	470801
Phua	...	470802
Jones	...	470803
Smith	...	470804
Ganio	...	470901
Jones	...	470902

#### Fillfactor 50 = 50% de páginas de hoja llenas

Con	...	470401
Funk	...	470402
White	...	470403

Rudd	...	470501
White	...	470502
Barr	...	470503

Akhtar	...	470601
Funk	...	470602
Smith	...	470603

Martin	...	470604
Smith	...	470701
Ota	...	470702

Martin	...	470801
Phua	...	470802
Jones	...	470803

Smith	...	470804
Ganio	...	470901
White	...	470902

Puede utilizar la opción FILLFACTOR para optimizar el rendimiento de las instrucciones INSERT y UPDATE en las tablas que contienen índices agrupados o no agrupados.

Cuando se llena una página de índice, SQL Server debe dedicar tiempo a dividirla para dejar sitio a las nuevas filas. Con la opción FILLFACTOR se asigna un porcentaje de espacio libre en las páginas de índice del nivel de hoja para reducir la división de las páginas.

**Nota** La opción FILLFACTOR sólo se aplica cuando se crea o se vuelve a generar el índice. SQL Server no mantiene dinámicamente el porcentaje de espacio asignado especificado en las páginas de índice.

El valor de FILLFACTOR que se debe especificar en una tabla depende de la frecuencia con que se modifiquen los datos (instrucciones INSERT y UPDATE) y del entorno de la organización. En general, debe hacer lo siguiente:

- Utilice un valor de FILLFACTOR pequeño en los entornos de proceso de transacciones en línea (OLTP).
- Utilice un valor de FILLFACTOR grande en los entornos de Analysis Services de SQL Server.

La tabla siguiente muestra los valores de la opción FILLFACTOR y los entornos en que se utilizan habitualmente.

Porcentaje de FILLFACTOR	Páginas del nivel de hoja	Páginas de niveles distintos del de hoja	Actividad en los valores de clave	Entorno de trabajo habitual
0 (predeterm.)	Rellenar completamente	Dejar sitio para una entrada de índice	Sin modificación o modificaciones leves	Analysis Services
1–99	Rellenar hasta el porcentaje especificado	Dejar sitio para una entrada de índice	Modificaciones de moderadas a frecuentes	Mixto u OLTP
100	Rellenar completamente	Dejar sitio para una entrada de índice	Sin modificación o modificaciones leves	Analysis Services

Al utilizar la opción FILLFACTOR, tenga en cuenta los siguientes hechos e instrucciones:

- Los valores de FILLFACTOR están comprendidos entre el 1 y el 100 por cien.
- El valor de FILLFACTOR predeterminado es 0. Este valor llena las páginas de índice del nivel de hoja al 100 por cien y deja sitio para el tamaño máximo de una entrada de índice en las páginas de índice no situadas en el nivel de hoja. No puede especificar explícitamente un valor de FILLFACTOR = 0.
- El valor de FILLFACTOR predeterminado puede modificarse en el servidor mediante el procedimiento almacenado de sistema **sp\_configure**.
- La tabla del sistema **sysindexes** almacena el valor de FILLFACTOR aplicado por última vez y otra información del índice.
- El valor de FILLFACTOR se especifica en porcentajes. El porcentaje determina cuánto deben llenarse las páginas del nivel de hoja. Por ejemplo, un valor de FILLFACTOR de 65 llena un 65 por ciento de cada página del nivel de hoja y deja libre un 35 por ciento del espacio de página para nuevas filas. El tamaño de la fila influye en el número de filas que caben en la página para el porcentaje de FILLFACTOR especificado.
- Utilice la opción FILLFACTOR en las tablas en las que se insertan muchas filas o cuando los valores de clave de un índice agrupado se modifican con frecuencia.

## Uso de la opción PAD\_INDEX

### Objetivo del tema

Explicar el uso de la opción PAD\_INDEX.

### Explicación previa

Puede utilizar la opción PAD\_INDEX con la opción FILLFACTOR para optimizar el rendimiento al crear o volver a generar un índice.

- La opción PAD\_INDEX se aplica a las páginas de índice no situadas en el nivel de hoja
- Si no se especifica PAD\_INDEX, el valor predeterminado deja espacio para una entrada de fila en las páginas no situadas en el nivel de hoja
- El número de filas en las páginas no situadas en el nivel de hoja nunca es menor que dos
- PAD\_INDEX utiliza el valor de FILLFACTOR

```
USE Northwind
CREATE INDEX OrderID_ind
ON Orders(OrderID)
WITH PAD_INDEX, FILLFACTOR=70
```

### Sugerencia

Señale que los alumnos utilizarán la opción PAD\_INDEX pocas veces. Se obtiene mayor rendimiento cuando se utiliza FILLFACTOR adecuadamente.

La opción PAD\_INDEX especifica el porcentaje con que se deben llenar las páginas de índice no situadas en el nivel de hoja. La opción PAD\_INDEX sólo se puede usar cuando se especifica también FILLFACTOR, ya que el valor de porcentaje de PAD\_INDEX está determinado por el especificado en FILLFACTOR.

La tabla siguiente muestra el efecto de los valores de la opción FILLFACTOR al utilizar la opción PAD\_INDEX, así como los entornos en los que suelen utilizarse los valores de PAD\_INDEX.

Porcentaje de FILLFACTOR	Páginas del nivel de hoja	Páginas de niveles distintos del de hoja	Actividad en los valores de clave	Entorno de trabajo habitual
1–99	Rellenar hasta el porcentaje especificado	Rellenar hasta el porcentaje especificado	Modificaciones de moderadas a frecuentes	OLTP

Al utilizar la opción PAD\_INDEX, tenga en cuenta lo siguiente:

- SQL Server aplica el porcentaje especificado por la opción FILLFACTOR tanto a las páginas del nivel de hoja como a las de otros niveles.
- De forma predeterminada, SQL Server deja siempre espacio suficiente para acomodar al menos una fila del tamaño máximo del índice en cada página que no está en el nivel de hoja, sin importar lo elevado que sea el valor de FILLFACTOR.
- El número de elementos de una página de índice que no está en el nivel de hoja nunca es inferior a dos, sin importar lo bajo que sea el valor de FILLFACTOR.
- PAD\_INDEX utiliza el valor de FILLFACTOR.



**Ejemplo**

En este ejemplo se crea el índice **OrderID\_ind** en la columna **OrdersID** de la tabla **Orders**. Al especificar la opción **PAD\_INDEX** con la opción **FILLFACTOR**, SQL Server crea páginas de nivel de hoja y de niveles distintos que están llenas en un 70 por ciento. Sin embargo, si no se utilizara la opción **PAD\_INDEX**, las páginas del nivel de hoja se llenarían en un 70 por ciento y las de otros niveles se llenarían casi completamente.

```
USE Northwind
CREATE INDEX OrderID_ind
  ON Orders(OrderID)
  WITH PAD_INDEX, FILLFACTOR=70
```

## ◆ Mantenimiento de índices

**Objetivo del tema**

Proporcionar una breve introducción a varias herramientas de mantenimiento.

**Explicación previa**

Una vez creados los índices, debe mantenerlos para asegurar que su rendimiento sea óptimo.

- Fragmentación de los datos
- Instrucción DBCC SHOWCONTIG
- Instrucción DBCC INDEXDEFRAG
- Opción DROP\_EXISTING

Una vez creados los índices, debe mantenerlos para asegurar que su rendimiento sea óptimo. Con el tiempo, los datos se fragmentan. La fragmentación de los datos debe administrarse en función del entorno de la organización.

SQL Server proporciona el Asistente para optimización de índices, que hace un seguimiento automático del uso de los índices, y ayuda a mantener y crear índices con un rendimiento óptimo.

También puede utilizar diversas opciones y herramientas como ayuda para volver a generar los índices y comprobar su optimización.

## Fragmentación de los datos

### Objetivo del tema

Explicar cómo se produce la fragmentación de datos y los métodos para administrarla.

### Explicación previa

Con el tiempo, a medida que se modifican las tablas se produce la fragmentación. Puede aplicar distintos métodos para administrar la fragmentación en función de su entorno de trabajo.

#### ■ Cómo se produce la fragmentación

- SQL Server reorganiza las páginas de índice cuando se modifican los datos
- La reorganización hace que se dividan las páginas de índice

#### ■ Métodos para administrar la fragmentación

- Elimine y vuelva a crear un índice, y especifique un valor para FILLFACTOR
- Vuelva a generar un índice y especifique un valor para FILLFACTOR

#### ■ Entorno de trabajo

- La fragmentación de datos puede ser beneficiosa en el entorno OLTP
- La fragmentación de datos puede ser desfavorable para el entorno Analysis Server

En función de su entorno de trabajo, la fragmentación puede afectar al rendimiento favorable o desfavorablemente.

### Cómo se produce la fragmentación

La fragmentación se produce al modificar los datos. Por ejemplo, cuando se agregan o se eliminan filas de una tabla, o cuando cambian los valores de las columnas indizadas, SQL Server ajusta las páginas de índice para adaptarlas a los cambios y mantener el almacenamiento de los datos indizados. Este ajuste de las páginas de índice se conoce como *división de páginas*. El proceso de división aumenta el tamaño de la tabla y el tiempo necesario para procesar las consultas.

### Métodos para administrar la fragmentación

En SQL Server hay dos métodos para administrar la fragmentación. El primero consiste en quitar y volver a crear un índice agrupado, y especificar un valor con la opción FILLFACTOR. El segundo consiste en volver a generar un índice y especificar un valor de FILLFACTOR.

### Entorno de trabajo

El grado de fragmentación aceptable en una base de datos depende del entorno de trabajo:

- En un entorno OLTP, la fragmentación puede ser beneficiosa, ya que las escrituras son frecuentes. Un sistema OLTP típico tiene un gran número de usuarios simultáneos que agregan y modifican datos continuamente.
- La fragmentación puede ser perjudicial en un entorno de Analysis Services, ya que implica lecturas frecuentes.

## Instrucción DBCC SHOWCONTIG

### Objetivo del tema

Explicar el uso de la instrucción DBCC SHOWCONTIG.

### Explicación previa

Si le preocupa la fragmentación, puede utilizar la instrucción DBCC SHOWCONTIG para averiguar si una tabla o sus índices están fragmentados.

#### ■ Qué determina la instrucción DBCC SHOWCONTIG

- Si una tabla o un índice están muy fragmentados
- Si las páginas de datos o de índices están llenas

#### ■ Cuándo ejecutar la instrucción DBCC SHOWCONTIG

- Si las tablas han sido muy modificadas
- Si las tablas contienen datos importados
- Si las tablas parecen provocar un bajo rendimiento en las consultas

### Sugerencia

Muestre el uso de la instrucción DBCC SHOWCONTIG con una tabla de la base de datos **ClassNorthwind** y describa a grandes rasgos el resultado.

La instrucción DBCC SHOWCONTIG muestra información acerca de la fragmentación de los datos e índices de una tabla específica.

### Qué determina la instrucción DBCC SHOWCONTIG

Al ejecutar la instrucción DBCC SHOWCONTIG, SQL Server recorre las páginas de índices del nivel de hoja para determinar si una tabla o un índice específicos están muy fragmentados. También determina si las páginas de datos o de índices están llenas.

### Cuándo ejecutar la instrucción DBCC SHOWCONTIG

Puede ejecutar la instrucción DBCC SHOWCONTIG en tablas con muchas modificaciones, que contengan datos importados o que parezcan provocar un bajo rendimiento en las consultas. Al ejecutar la instrucción DBCC SHOWCONTIG, tenga en cuenta los siguientes hechos e instrucciones:

- SQL Server requiere una referencia a un identificador de tabla o de índice al ejecutar la instrucción DBCC SHOWCONTIG. Puede consultar la tabla **sysindexes** para obtener el identificador de tabla o de índice.
- Determine la frecuencia con que debe ejecutar la instrucción DBCC SHOWCONTIG. Mida la actividad diaria, semanal o mensual de las tablas.

### Sugerencia

Demuestre una de las dos maneras de obtener el identificador de una tabla. Una de ellas consiste en utilizar la función **OBJECT\_ID**. La otra supone hacer una consulta de la tabla del sistema **sysindexes** de la base de datos **ClassNorthwind**.

En la tabla siguiente se describen las estadísticas que devuelve la instrucción DBCC SHOWCONTIG.

Estadística	Descripción
Páginas recorridas	Número de páginas en la tabla o el índice.
Extensiones recorridas	Número de extensiones en la tabla o el índice.
Cambios de extensión	Número de veces que la instrucción DBCC ha salido de una extensión mientras recorría sus páginas.
Promedio de páginas por extensión	Número de páginas por extensión en la cadena de páginas.
Densidad de recorrido [Cuenta óptima: Cuenta real]	La densidad de recorrido es 100 (es un porcentaje) si todo es contiguo; si es inferior a 100, hay fragmentación. La cuenta óptima es el número ideal de cambios de extensión que habría si todos los elementos fueran contiguos. La cuenta real es el número real de cambios de extensión.
Fragmentación del recorrido lógico	El porcentaje de páginas no ordenadas devueltas del recorrido de las páginas de hoja de un índice. Este número no es relevante en montones e índices de texto. Una página no ordenada es aquella para la que la siguiente página indicada en un Mapa de asignación de índices (IAM) es diferente a la señalada por el puntero de página siguiente en la página de hoja.
Fragmentación de recorrido de extensión	El porcentaje de extensiones no ordenadas devueltas del recorrido de las páginas de hoja de un índice. Este número no es relevante en montones. Una extensión no ordenada es aquella en la que la extensión que contiene la página actual de un índice no corresponde a la extensión física siguiente (es decir, la extensión que va tras la que contiene la página anterior de un índice).
Promedio de bytes libres por página	Número promedio de bytes libres en las páginas recorridas. Cuanto mayor sea este número, menos llenas estarán las páginas; son preferibles los números bajos. Sin embargo, tenga presente que este número también se ve afectado por el tamaño de fila. Un tamaño de fila grande puede dar como resultado un número mayor.
Promedio de densidad de página (completa)	Valor que muestra hasta qué punto está llena una página. En él se considera el tamaño de fila, por lo que es una indicación más precisa de lo llena que está la página. Los porcentajes altos son preferibles a los bajos.

## Sintaxis

### DBCC SHOWCONTIG

```
[({nombreTabla | idTabla | nombreVista | idVista }
[, nombreÍndice | idÍndice ] )]
[ WITH
{ ALL_INDEXES | FAST
[, ALL_INDEXES ] | TABLERESULTS
[, { ALL_INDEXES } ]
[, { FAST | ALL_LEVELS } ]
}
]
```

**Ejemplo**

En este ejemplo se ejecuta una instrucción que tiene acceso a la tabla **Customers**.

```
USE Northwind
DBCC SHOWCONTIG (Customers, PK_Customers)
```

**Resultado**

```
DBCC SHOWCONTIG recorriendo la tabla 'Customers'...
Tabla: 'Customers' (2073058421); Id. de índice: 1, Id. de base
de datos: 6
Realizado recorrido de nivel TABLE.
Páginas recorridas:                3
Extensiones recorridas:            2
Cambios de extensión:              1
Promedio de páginas por extensión: 1.5
Densidad de recorrido [Cuenta      50.00% [1:2]
óptima:Cuenta real]:
Fragmentación del recorrido lógico: 0.00%
Fragmentación de recorrido de      50.00%
extensión:
Promedio de bytes libres por página: 246.7
Promedio de densidad de página      96.95%
(completa):
Ejecución de DBCC completada. Si hay mensajes de error,
consulte al administrador del sistema.
```

**Sugerencia**

Vea si estos resultados son adecuados en un entorno determinado.

## Instrucción DBCC INDEXDEFRAG

**Objetivo del tema**

Explicar el uso de la instrucción DBCC INDEXDEFRAG.

**Explicación previa**

La instrucción DBCC INDEXDEFRAG puede desfragmentar el nivel de hoja de índices agrupados y no agrupados basados en tablas y vistas.

**■ Uso de la instrucción DBCC INDEXDEFRAG**

- Desfragmenta el nivel de hoja de un índice
- Organiza las páginas de nivel de hoja de forma que el orden físico coincida con el orden lógico de izquierda a derecha
- Mejora el rendimiento del recorrido de índices

**■ Desfragmentación y regeneración de índices**

A medida que los datos de una tabla cambian, algunas veces los índices se *fragmentan*. La instrucción DBCC INDEXDEFRAG puede *desfragmentar* el nivel de hoja de índices agrupados y no agrupados basados en tablas y vistas. Al desfragmentar las páginas se organizan de forma que su orden físico coincida con el orden lógico de izquierda a derecha de los nodos de hoja. Esta reorganización mejora el rendimiento del recorrido de índices.

### Uso de la instrucción DBCC INDEXDEFRAG

Cuando se utiliza la instrucción DBCC INDEXDEFRAG, ésta:

- Compacta las páginas de un índice, teniendo en cuenta el valor de FILLFACTOR especificado cuando se creó el índice. Toda página vacía creada como resultado de esta compactación se quitará.
- Desfragmenta un archivo a la vez cuando un índice abarca más de un archivo. Las páginas no migran entre archivos.
- Informa al usuario del porcentaje previsto completado. Dichos informes se producen cada cinco minutos. La instrucción DBCC INDEXDEFRAG puede terminarse en cualquier punto del proceso y todo el trabajo completado se conserva.
- Es una operación en línea. No establece bloqueos durante un tiempo prolongado y no bloquea las consultas o actualizaciones en ejecución. La desfragmentación se registra siempre al completo, sin importar la configuración del modelo de recuperación de la base de datos.

## Desfragmentación y regeneración de índices

El tiempo necesario para desfragmentar está relacionado con la cantidad de fragmentación. Un índice muy fragmentado podría requerir más tiempo para desfragmentarse que para regenerarse. Se tarda menos en desfragmentar un índice relativamente desfragmentado que en regenerar uno nuevo.

---

**Nota** El uso de la instrucción DBCC INDEXDEFRAG no mejora el rendimiento cuando los índices están físicamente desfragmentados en el disco. Para desfragmentar físicamente un índice, hay que regenerarlo.

---

### Sintaxis

DBCC INDEXDEFRAG

```
( { nombreBaseDeDatos | IdBaseDeDatos | 0 }  
  , { nombreTabla | IdTabla | 'nombreVista' | IdVista }  
  , { nombreÍndice | IdÍndice }  
 ) [ WITH NO_INFOMSGS ]
```

### Ejemplo

En este ejemplo se ejecuta la instrucción DBCC INDEXDEFRAG en el índice **mem\_no\_CL** de la tabla **Member** de la base de datos **credit**.

```
DBCC INDEXDEFRAG(credit, member, mem_no_CL)
```

### Resultado

Pages scanned	Pages moved	Pages removed
150	28	9

(1 filas afectadas)



## Opción DROP\_EXISTING

**Objetivo del tema**

Explicar el uso de la opción DROP\_EXISTING para cambiar la definición de un índice o regenerar índices.

**Explicación previa**

Puede utilizar la opción DROP\_EXISTING para cambiar una definición de un índice o para regenerar índices.

- **Regeneración de un índice**
  - Reorganiza las páginas de hoja
  - Quita la fragmentación
  - Vuelve a calcular las estadísticas de índices
- **Cambio de las características de un índice**
  - Tipo
  - Columnas del índice
  - Opciones

```
CREATE UNIQUE NONCLUSTERED INDEX U_OrdID_ProdID  
ON [Order Details] (OrderID, ProductID)  
WITH DROP_EXISTING, FILLFACTOR=65
```

Puede utilizar la opción DROP\_EXISTING para cambiar las características de un índice o regenerarlo sin necesidad de quitarlo y volverlo a crear. La ventaja de utilizar la opción DROP\_EXISTING consiste en que permite modificar los índices creados con las restricciones PRIMARY KEY y UNIQUE.

### Regeneración de un índice

Ejecute la instrucción CREATE INDEX con la opción DROP\_EXISTING para regenerar un índice agrupado o no agrupado:

- Reorganice las páginas del nivel de hoja mediante la compresión o expansión de las filas.
- Quite la fragmentación.
- Vuelva a calcular las estadísticas del índice.

**Para su información**  
No enseñe la instrucción DBCC DBREINDEX. DROP\_EXISTING proporciona más funcionalidad. La única ventaja de DBCC DBREINDEX es que permite regenerar varios índices simultáneamente.

## Cambio de las características de un índice

Al utilizar la opción DROP\_EXISTING puede cambiar las siguientes características de un índice:

- Tipo
  - Puede convertir un índice no agrupado en un índice agrupado.
  - No es posible convertir un índice agrupado en no agrupado.
- Columnas del índice
  - Puede cambiar la definición del índice para especificar columnas distintas.
  - Puede especificar columnas adicionales o quitar columnas de los índices compuestos.
  - Puede cambiar las columnas del índice para que sean únicas o no únicas.
- Opciones
  - Puede cambiar el valor de porcentaje de FILLFACTOR o PAD\_INDEX.

Al utilizar la opción DROP\_EXISTING, tenga en cuenta los siguientes hechos e instrucciones:

- En el caso de un índice agrupado, SQL Server requiere tener 1,2 veces el espacio que ocupa la tabla para volver a organizar físicamente los datos.
- La opción DROP\_EXISTING acelera el proceso de generación de índices agrupados y no agrupados, ya que elimina el proceso de ordenación.
- Utilice la opción FILLFACTOR con la opción DROP\_EXISTING si desea que las páginas del nivel de hoja se llenen hasta un porcentaje determinado. Esto puede ser útil cuando sea necesario asignar espacio para nuevos datos o cuando haya que compactar el índice.
- No es posible volver a generar los índices de las tablas de sistema.
- La opción DROP\_EXISTING en un índice agrupado ayuda a evitar la tarea innecesaria de eliminar y volver a crear índices no agrupados si el índice agrupado se regenera en la misma columna.
- Los índices no agrupados se regeneran una sola vez y sólo si las claves son distintas.

### Ejemplo

En este ejemplo se vuelve a generar el índice existente, **U\_OrdID\_ProdID**, de la tabla **Order Details**. El índice se define como agrupado, compuesto y con la opción de que se llene cada página hasta un 65 por ciento. Esta instrucción producirá un error si ya existe un índice agrupado en la tabla **Order Details**.

```
CREATE UNIQUE NONCLUSTERED INDEX U_OrdID_ProdID
ON [Order Details] (OrderID, ProductID)
WITH DROP_EXISTING, FILLFACTOR=65
```

## ◆ Introducción a las estadísticas

**Objetivo del tema**

Presentar los temas de esta sección.

**Explicación previa**

Las estadísticas se pueden crear sobre índices y columnas. Debido a que el optimizador de consultas utiliza estadísticas para optimizar consultas, es necesario que conozca la forma en que se recopilan, almacenan, crean, actualizan y ven.

- Cómo se recopilan las estadísticas
- Cómo se almacenan las estadísticas
- Creación de estadísticas
- Actualización de estadísticas
- Vista de las estadísticas

---

Las estadísticas se pueden crear sobre índices y columnas. Debido a que el optimizador de consultas utiliza estadísticas para optimizar consultas, es necesario que conozca la forma en que se recopilan, almacenan, crean, actualizan y ven.

## Cómo se recopilan las estadísticas

**Objetivo del tema**

Explicar cómo SQL Server recopila estadísticas.

**Explicación previa**

Las estadísticas son un muestreo de los valores de columna.

- **Lee los valores de columna o una muestra de ellos**
  - Produce una lista de valores distribuida y ordenada uniformemente
- **Realiza un recorrido completo o un muestreo de las filas**
  - Determina dinámicamente el porcentaje de filas que se van a muestrear en función del número de filas que hay en la tabla
- **Selecciona muestras**
  - De la tabla o del índice no agrupado más pequeño creado con las columnas
  - Todas las filas de la página de datos se utilizan para actualizar la información estadística

Las *estadísticas* son un muestreo de los valores de columna.

### Lee los valores de columna o una muestra de ellos

SQL Server recopila estadísticas mediante la lectura de todos los valores de columna o de una muestra de ellos para producir una lista de valores distribuida y ordenada uniformemente conocida como *pasos de distribución*. Para generar los pasos de distribución SQL Server realiza un recorrido completo o un recorrido de muestreo y, después, selecciona las muestras.

### Realiza un recorrido completo o un muestreo de las filas

SQL Server determina dinámicamente el porcentaje de filas que se van a muestrear en función del número de filas que hay en la tabla. Para recopilar estadísticas el optimizador de consultas realiza un recorrido completo o un muestreo de las filas.

- La opción **SAMPLE** es la predeterminada para actualizar y crear estadísticas.
- La opción **FULLSCAN** se utiliza cuando:
  - Se crean los índices.
  - La opción **FULLSCAN** aparece especificada en la instrucción **CREATE STATISTICS**.
  - Se ejecuta la instrucción **UPDATE STATISTICS**.

## Selecciona muestras

La muestra se selecciona al azar en las páginas de datos de la tabla o del índice no agrupado más pequeño creado con las columnas que necesitan las estadísticas. Una vez que una página se ha leído en el disco, todas las filas de la página de datos se utilizan para actualizar la información estadística.

Cuando el optimizador de consultas recopila muestras:

- El tamaño de la tabla determina el método elegido.
- Se muestrea un número mínimo de valores para crear estadísticas útiles.
- Si el número de filas especificado es demasiado pequeño como para que resulte útil, el optimizador de consultas corrige automáticamente la muestra en función del número de filas existentes en la tabla.
- En el caso de los índices compuestos, las estadísticas sólo se mantienen en la primera columna definida.

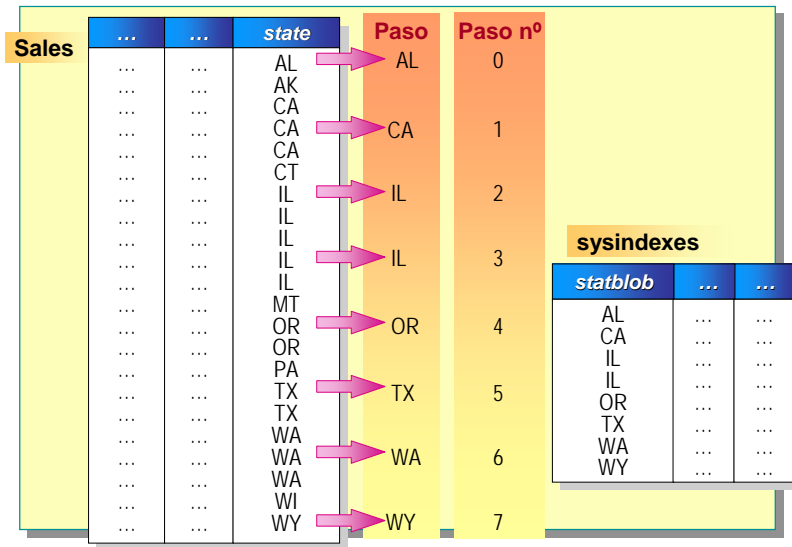
## Cómo se almacenan las estadísticas

### Objetivo del tema

Explicar cómo se almacenan las estadísticas en SQL Server.

### Explicación previa

Las estadísticas se almacenan en la columna **statblob** de la tabla del sistema **sysindexes**.



### Sugerencia

Al crear un índice con la tabla **Sales**, SQL Server crea estadísticas automáticamente.

Este ejemplo muestra cómo SQL Server crea las estadísticas.

Las estadísticas se almacenan en la columna **statblob** de la tabla del sistema **sysindexes**.

## Pasos de distribución

Cada valor almacenado en la columna **statblob** se denomina paso de distribución. Los pasos de distribución hacen referencia al espacio entre muestras de datos o al número de filas que hay entre ellas antes de que se tome y almacene la siguiente muestra. Los valores de clave primero y último del índice siempre se incluyen en las estadísticas. Puede haber hasta 300 valores, de los que el punto final es el valor tricentésimo.

## Contenido de la columna statblob

Además de almacenar pasos de distribución, la columna **statblob** almacena también:

- La fecha y la hora en que las estadísticas se actualizaron por última vez.
- El número de filas existentes en la tabla.
- El número de filas muestreadas para crear el histograma y determinar la densidad.
- El número de pasos de distribución.
- La longitud promedio de la clave.

- La densidad de las columnas individuales y de todas las columnas combinadas.
- El número de filas correspondientes a un paso del histograma.
- El número de filas que tienen un valor igual al límite superior del paso del histograma.
- El número de valores distintos en un paso del histograma.

---

**Nota** La columna **statblob** se define como un tipo de datos **image**.

---

## Creación de estadísticas

### Objetivo del tema

Explicar cuándo se crean las estadísticas y cuándo es conveniente crearlas.

### Explicación previa

Puede crear estadísticas automática o manualmente. Sin embargo, debe permitir que SQL Server cree estadísticas de forma automática.

#### ■ Creación automática de estadísticas

- Para columnas indizadas que contienen datos
- Para columnas no indizadas que se utilizan en un predicado de combinación o en una cláusula WHERE

#### ■ Creación manual de estadísticas

- Columnas que no están indizadas
- Todas las columnas de un índice compuesto, excepto la primera

---

Puede crear estadísticas automática o manualmente. Sin embargo, debe permitir que SQL Server cree estadísticas de forma automática.

### Creación automática de estadísticas

Cuando el valor predeterminado de la opción de base de datos **crear estadísticas automáticamente** está establecido en ON, SQL Server crea estadísticas automáticamente:

- Para columnas indizadas que contienen datos.
- Para columnas no indizadas que se utilizan en un predicado de combinación o en una cláusula WHERE.

El optimizador de consultas activa la creación automática de estadísticas al optimizar una consulta. Esto puede resultar un inconveniente si el optimizador de consultas determina que las estadísticas están perdidas. El plan de ejecución incluirá la acción de creación de estadísticas, lo que requerirá tiempo adicional para procesar la consulta.

---

**Nota** Cuando se ejecuta una consulta y se examina el plan de ejecución, el optimizador de consultas puede sugerir acciones correctoras, como crear o actualizar las estadísticas, o crear un índice. En ese punto, puede crear o actualizar inmediatamente estadísticas e índices.

---



## Creación manual de estadísticas

Puede ejecutar la instrucción `CREATE STATISTICS` para crear un histograma y los grupos de densidad asociados de columnas específicas. Puede crear estadísticas en:

- Columnas que no están indizadas:
- Todas las columnas de un índice compuesto, excepto la primera.
- Columnas calculadas sólo cuando se den las condiciones que permitan crear un índice en esas columnas.
- Columnas de tipos de datos **image**, **text** y **ntext** que no están definidas.

La creación manual de estadísticas es útil si tiene una columna en la que un índice no sería beneficioso, pero las estadísticas en esa columna pueden ser útiles para crear planes de ejecución más óptimos. Tener estadísticas en esas columnas elimina la carga de trabajo que supone un índice, al tiempo que permite al optimizador de consultas utilizar la columna al optimizar consultas.

---

**Nota** Para crear manualmente estadísticas en una tabla es necesario ser su propietario.

---

### Sintaxis parcial

```
CREATE STATISTICS nombreEstadísticas ON {tabla| vista} (columna [...n])
```

## Actualización de estadísticas

### Objetivo del tema

Describir cómo SQL Server determina la frecuencia de actualización de las estadísticas y cuándo actualizarlas.

### Explicación previa

Con el tiempo, las estadísticas se quedan desfasadas, lo que puede afectar al rendimiento del optimizador de consultas.

- Frecuencia de actualización de las estadísticas
- Actualización automática de estadísticas
- Actualización manual de estadísticas
  - Si crea un índice antes de introducir datos en una tabla
  - Si la tabla se trunca
  - Si agrega muchas filas a una tabla que contiene un mínimo de datos o ningún dato, y tiene pensado consultar inmediatamente esa tabla

---

Con el tiempo, las estadísticas se quedan desfasadas, lo que puede afectar al rendimiento del optimizador de consultas.

## Frecuencia de actualización de las estadísticas

SQL Server actualiza la información estadística cuando ésta ya no es actual. El volumen de datos de la columna relativo a la cantidad de datos que cambian determina la frecuencia de actualización. Por ejemplo:

- Las estadísticas de una tabla que contiene 10.000 filas puede requerir actualización cuando 1.000 valores de índice han cambiado, ya que estos valores representan un porcentaje significativo de la tabla.
- Las estadísticas de una tabla que contiene 10 millones de entradas de índice puede no requerir actualización cuando 1.000 valores de índice han cambiado, ya que estos valores representan un porcentaje pequeño de la tabla.

SQL Server siempre toma muestras de un número mínimo de filas. Las tablas con un tamaño inferior a 8 megabytes (MB) siempre se recorren totalmente para recopilar estadísticas.

---

**Nota** Cuando las estadísticas están desfasadas o no están disponibles, SQL Server emite un mensaje de advertencia. Éste aparece cuando se examina el plan de ejecución. Puede utilizar el Analizador de SQL para supervisar la clase de suceso **Missing Column Statistics**. Esta clase de suceso indica cuándo faltan estadísticas.

---

## Actualización automática de estadísticas

Debe permitir que SQL Server actualice estadísticas automáticamente. Cuando la opción de base de datos **crear estadísticas automáticamente** está establecida en ON (valor predeterminado), SQL Server actualiza automáticamente las estadísticas al quedarse desfasadas.

Por ejemplo, si una tabla se actualiza bastante desde la última vez que se crearon o actualizaron las estadísticas, SQL Server actualiza automáticamente las estadísticas para optimizar las consultas que utilizan la tabla.

El optimizador de consultas activa la actualización automática de estadísticas al optimizar una consulta. Esto puede resultar un inconveniente si el optimizador de consultas determina que las estadísticas están desfasadas. El plan de ejecución incluirá la acción de actualización de estadísticas, lo que requerirá tiempo adicional para procesar la consulta.

## Actualización manual de estadísticas

Puede ejecutar la instrucción UPDATE STATISTICS para actualizar la información acerca de la distribución de los valores de clave para una o varias estadísticas de una tabla especificada. Es posible que le convenga actualizar manualmente las estadísticas de una tabla o una columna en las siguientes situaciones:

- Si crea un índice antes de introducir datos en una tabla.
- Si la tabla se trunca.
- Si agrega muchas filas a una tabla que contiene un mínimo de datos o ningún dato, y tiene pensado consultar inmediatamente esa tabla.

### Sintaxis parcial

```
UPDATE STATISTICS tabla | vista [índice | (nombreEstadísticas[,...n])]
```

---

**Nota** Para ver una lista de nombres y descripciones de índices, ejecute el procedimiento almacenado del sistema **sp\_helpindex** con la tabla de nombres.

---

## Vista de las estadísticas

### Objetivo del tema

Explicar el modo de ver la información de las estadísticas.

### Explicación previa

Puede ver la información estadística en la página de distribución de un índice o una columna con la instrucción DBCC SHOW\_STATISTICS.

- La instrucción DBCC SHOW\_STATISTICS devuelve la información estadística en la página de distribución de un índice o una columna
- La información estadística incluye:
  - La hora en que las estadísticas se actualizaron por última vez
  - El número de filas muestreadas para producir el histograma
  - Información de densidad
  - Longitud promedio de la clave
  - Información del paso del histograma

Puede ver la información estadística en la página de distribución de un índice o una columna con la instrucción DBCC SHOW\_STATISTICS.

La tabla siguiente describe la información que devuelve la instrucción DBCC SHOW\_STATISTICS.

Nombre de columna	Descripción
<b>Updated</b>	Fecha y hora en que las estadísticas se actualizaron por última vez.
<b>Rows</b>	Número de filas existentes en la tabla.
<b>Rows sampled</b>	Número de filas muestreadas para la información estadística.
<b>Steps</b>	Número de pasos de distribución.
<b>Density</b>	Selectividad del prefijo de la primera columna del índice (no es frecuente).
<b>Average key length</b>	Longitud promedio del prefijo de la primera columna del índice.
<b>All density</b>	Selectividad de un conjunto de prefijos de columnas de índice (frecuente).
<b>Average length</b>	Longitud promedio de un conjunto de prefijos de columnas de índice.
<b>Columns</b>	Nombre de los prefijos de columnas de índice para los que se muestran las columnas <b>All density</b> y <b>Average length</b> .
<b>RANGE_HI_KEY</b>	Valor del límite superior de un paso del histograma.
<b>RANGE_ROWS</b>	Número de filas de la muestra comprendidas en un paso del histograma, excluido el límite superior.

*(continuación)*

Nombre de columna	Descripción
<b>EQ_ROWS</b>	Número de filas de la muestra que tienen un valor igual al límite superior del paso del histograma.
<b>DISTINCT_RANGE_ROWS</b>	Número de valores distintos en un paso del histograma, excluido el límite superior.
<b>AVG_RANGE_ROWS</b>	Número promedio de valores duplicados en un paso del histograma, excluido el límite superior ( $\text{RANGE\_ROWS} / \text{DISTINCT\_RANGE\_ROWS}$ para $\text{DISTINCT\_RANGE\_ROWS} > 0$ ).

**Sintaxis****DBCC SHOW\_STATISTICS** (*tabla*, *destino*)

Ver las estadísticas resulta de utilidad, normalmente, cuando se desea ajustar el rendimiento para consultas específicas. En la mayor parte de las aplicaciones, no es necesario ver las estadísticas.

## Consultas en la tabla sysindexes

### Objetivo del tema

Explicar cómo consultar la tabla **sysindexes** para obtener información acerca de tablas e índices.

### Explicación previa

Puede consultar la tabla **sysindexes** para obtener información acerca de tablas e índices, además de las estadísticas de cada índice.

- Almacena la información de la tabla y del índice
  - Tipo de índice (**indid**)
  - Espacio utilizado (**dpages**, **reserved** y **used**)
  - Fillfactor (**OrigFillFactor**)
- Almacena las estadísticas de cada índice

Puede consultar la tabla **sysindexes** para obtener información acerca de tablas e índices, además de las estadísticas de cada índice. La tabla siguiente es una lista parcial de la información que puede ver y que procede de los datos almacenados en la tabla **sysindexes**.

Columna	Descripción	Valores
<b>indid</b> (tipo de índice)	Id. del índice (tipo de índice)	Los valores posibles son: <ul style="list-style-type: none"> <li>● 0 para tablas no agrupadas</li> <li>● 1 para índices agrupados</li> <li>● &gt;1 para índices no agrupados</li> <li>● 255 para tablas que tienen datos de texto o imagen</li> </ul>
<b>dpages</b> (espacio utilizado)	Número de páginas de índice de nivel de hoja	Para <b>indid</b> = 0 o <b>indid</b> = 1, <b>dpages</b> es el número de páginas utilizadas. Para <b>indid</b> =255, <b>dpages</b> está establecido en 0. De otro modo, <b>dpages</b> es el número de páginas utilizadas del índice no agrupado.
<b>reserved</b> (espacio utilizado)	Número de páginas reservadas para un índice	Para <b>indid</b> = 0 o <b>indid</b> = 1, <b>reserved</b> es el número de páginas asignadas a todos los índices y datos de la tabla. Para <b>indid</b> = 255, <b>reserved</b> es el número de páginas asignadas a datos de texto o imagen. De otro modo, <b>reserved</b> es el número de páginas asignadas al índice.
<b>used</b> (espacio utilizado)	Cantidad de espacio utilizado por un índice	Para <b>indid</b> = 0 o <b>indid</b> = 1, <b>used</b> es el número total de páginas utilizadas en todos los índices y datos de la tabla. Para <b>indid</b> = 255, <b>used</b> es el número de páginas utilizadas en datos de texto o imagen. De otro modo, <b>used</b> es el número de páginas utilizadas en el índice.

*(continuación)*

Columna	Descripción	Valores
<b>OrigFillFactor</b> (fillfactor)	Valor de FILLFACTOR original utilizado cuando se creó el índice	Este valor no se mantiene; sin embargo, puede resultar útil si necesita volver a crear un índice y no recuerda el valor de FILLFACTOR que se utilizó.
<b>minlen</b>	Tamaño mínimo de una fila	Valor entero.
<b>xmaxlen</b>	Tamaño máximo de una fila	Valor entero.
<b>maxirow</b>	Tamaño máximo de una fila del índice no situada en el nivel de hoja	Valor entero.
<b>keys</b>	Descripción de las columnas de clave	Sólo se aplica si la entrada es un índice.
<b>statversion</b>	Número de veces que se han actualizado las estadísticas	Valor entero.
<b>statblob</b>	Objeto largo binario de estadísticas (BLOB)	Almacena la información estadística.

**Ejemplo**

En este ejemplo se ejecuta una instrucción que tiene acceso al identificador de índice y a otra información de la tabla del sistema **sysindexes**. Especifique el nombre de un índice agrupado (*nombreÍndice*) en la cláusula WHERE para obtener su identificador de índice.

```
SELECT id, indid, reserved, used, origfillfactor, name
FROM Northwind.dbo.sysindexes
WHERE name = 'PK_customers'
```

**Resultado**

id	indid	reserved	used	origfillfactor	name
2073058421	1	15	15	0	PK_Customers

(1 filas afectadas)

# Configuración de índices mediante el Asistente para optimización de índices

**Objetivo del tema**

Presentar el Asistente para optimización de índices.

**Explicación previa**

Ya sea un usuario sin experiencia o uno avanzado, el Asistente para optimización de índices puede ayudarle a crear índices adecuados en una base de datos nueva o a comprobar los índices existentes en su base de datos actual.

- **Utilice el Asistente para optimización de índices para:**
  - Recomendar o comprobar la configuración de índices óptima
  - Facilitar el análisis del costo
  - Recomendar formas de ajustar la base de datos
  - Especificar criterios cuando evalúa una carga de trabajo
- **No utilice el Asistente para optimización de índices en:**
  - Tablas a las que se hace referencia mediante consultas cruzadas de base de datos que no existen
  - Tablas del sistema, restricciones PRIMARY KEY, índices únicos

Ya sea un usuario sin experiencia o uno avanzado, el Asistente para optimización de índices puede ayudarle a crear índices adecuados en una base de datos nueva o a comprobar los índices existentes en su base de datos actual. El Asistente para optimización de índices examina la carga de consultas con el fin de determinar qué índices son útiles, mientras que la característica de plan de ejecución muestra los índices que se utilizan en las consultas.

## Determinar cuándo utilizar el Asistente para optimización de índices

Los usuarios sin experiencia pueden utilizar el asistente para crear rápidamente una configuración de índices óptima. Los usuarios avanzados pueden utilizarlo para establecer una configuración de índices de referencia y, después, personalizarla o comprobar la configuración de sus índices.

El Asistente para optimización de índices puede:

- Recomendar o comprobar la configuración de índices óptima para una base de datos dada una carga de trabajo o un archivo de traza aplicado, mediante el análisis de costos del optimizador de consultas.
- Facilitar el análisis del costo de los cambios propuestos, como son:
  - Uso de los índices actuales y de los recomendados.
  - Mejora del rendimiento de las consultas para las 100 consultas más costosas y participación de la tabla en una carga de trabajo.



- Recomendar formas de ajustar la base de datos para un conjunto pequeño de consultas problemáticas.
- Especificar criterios que tener en cuenta cuando el Asistente para optimización de índices evalúa una carga de trabajo, como el número máximo de consultas que se debe ajustar, el espacio máximo que pueden ocupar los índices recomendados y el número máximo de columnas por índice.

## Determinar cómo utilizar el Asistente

Al utilizar el Asistente para optimización de índices, tenga en cuenta los siguientes hechos e instrucciones:

- El usuario que invoca el Asistente para optimización de índices debe ser miembro de la función fija de servidor **sysadmin** puesto que las consultas de la carga de trabajo se analizan en el contexto de seguridad del usuario.
- No se recomienda utilizar el Asistente para optimización de índices en los siguientes casos:
  - Tablas a las que se hace referencia mediante consultas cruzadas de base de datos que no existen en la base de datos seleccionada actualmente.
  - Tablas del sistema.
  - Restricciones PRIMARY KEY e índices únicos.
  - El asistente puede eliminar o reemplazar un índice agrupado que no sea único o que se haya creado con una restricción PRIMARY KEY.
- No se recomienda eliminar ningún índice cuando la opción **Mantener todos los índices existentes** está seleccionada.

El asistente recomienda utilizar sólo índices nuevos, siempre que sea conveniente. Desactivar esta opción puede producir una mejora general superior en el rendimiento de la carga de trabajo.
- Se recomienda dejar seleccionada la opción **Todas las vistas indizadas**.
- Las sugerencias pueden impedir que el Asistente para optimización de índices elija un plan de ejecución mejor. Considere quitar las sugerencias de índice de las consultas antes de analizar la carga de trabajo.

- Si desea reducir el tiempo de ejecución del Asistente para optimización de índices, debe:
  - Asegurarse de que no está seleccionada la opción **Realizar un análisis minucioso** del cuadro de diálogo **Seleccionar servidor y base de datos**. Seleccionar esta opción hace que el Asistente para optimización de índices realice un análisis pormenorizado de las consultas, lo que da lugar al aumento del tiempo de ejecución. También puede producir una mejora general superior en el rendimiento de la carga de trabajo ajustada.
  - Ajustar sólo un subconjunto de las tablas de la base de datos.
  - Reducir el tamaño del archivo de carga de trabajo.

---

**Nota** Si utiliza el Asistente para optimización de índices para analizar una secuencia de comandos de Transact-SQL que no tiene una extensión de nombre de archivo .sql, como Mi\_secuencia.txt, y abre el archivo con la opción **Formato de archivo** establecida en **Auto**, el asistente genera el mensaje de error **Formato de archivo no válido**. En lugar de eso, establezca la opción **Formato de archivo** a **ANSI SQL** o **UNICODE SQL**.

---

## Consideraciones acerca del rendimiento

**Objetivo del tema**

Comentar consideraciones acerca del rendimiento al planear y crear índices.

**Explicación previa**

Al crear o utilizar índices, haga lo siguiente para reducir su efecto en el rendimiento.

- Cree índices basados en claves externas
- Cree los índices agrupados antes que los no agrupados
- Considere la creación de índices compuestos
- Cree varios índices para una tabla que se lea con frecuencia
- Utilice el Asistente para optimización de índices

---

Al crear o utilizar índices, haga lo siguiente para reducir su efecto en el rendimiento.

- Cree índices basados en claves externas, ya que, normalmente, las consultas harán referencia a ellas.
- Cree los índices agrupados antes que los no agrupados, ya que los primeros cambian el orden físico de las filas de la tabla.
- Cree índices compuestos. El rendimiento de las consultas mejora con los índices compuestos, especialmente cuando los usuarios buscan información de varias maneras.
- Cree varios índices para cada tabla, especialmente para las que se lean con frecuencia. El rendimiento de las consultas mejora cuando la tabla tiene a la vez un índice agrupado y varios índices no agrupados.
- Utilice el Asistente para optimización de índices con el fin de hacer un seguimiento automático del uso de los índices y como ayuda para el mantenimiento y creación de índices con el mejor rendimiento.