

Indexaciones de tablas en un Dedicated SQL Pool

Tipos de índice

El dedicated SQL pool ofrece varias opciones de indexación, entre las que se incluyen índices de almacén de columnas agrupados (**clustered columnstore indexes**), índices agrupados (**clustered indexes**) e índices no agrupados (**nonclustered indexes**) y una opción no-index también conocida como montón (**heap**).

Para crear una tabla con un índice, consulte la documentación de [CREATE TABLE \(dedicated SQL pool\)](#)

Clustered columnstore indexes

De forma predeterminada, el Dedicated SQL Pool crea un **Clustered columnstore index** cuando no se especifican opciones de índice en una tabla. Las tablas con **Clustered columnstore index** ofrecen el máximo nivel de compresión de datos, así como el mejor rendimiento general de las consultas. Por lo general, las tablas con “Clustered columnstore index” funcionan mejor que las tablas “Heap” o “Clustered index”, y **suelen ser la mejor opción para tablas grandes**. Por estos motivos, el **Clustered columnstore index** es el mejor lugar por el que empezar cuando no se sabe cómo indexar una tabla.

Para crear una tabla de almacén de columnas agrupado, simplemente especifique **CLUSTERED COLUMNSTORE INDEX** en la cláusula **WITH** u omita la cláusula **WITH**:

```
CREATE TABLE myTable
(
    id int NOT NULL,
    lastName varchar(20),
    zipCode varchar(6)
)
WITH ( CLUSTERED COLUMNSTORE INDEX );
```

Hay algunos escenarios en los que un almacén de columnas agrupado puede que no sea una buena opción:

- Las tablas con Clustered columnstore index no admiten varchar (max), nvarchar (max) y varbinary (max). Considere la posibilidad de usar un Clustered index o Heap en su lugar.
- Las tablas con Clustered columnstore index pueden que sean menos eficientes para datos transitorios. Considere la posibilidad de usar tablas Heap, e incluso tablas temporales.
- Tablas pequeñas con menos de 60 millones de filas. Considere la posibilidad de usar tablas Heap.

Heap tables

Cuando almacene datos temporalmente en el Dedicated SQL Pool, puede encontrarse con que el uso de tablas Heap agiliza el proceso global. Esto se debe a que las cargas en “montones” son más rápidas que la indexación de tablas y, en algunos casos, la posterior lectura se puede realizar desde la memoria caché. *If you are loading data only to stage it before running more transformations, loading the table to heap table is much faster than loading the data to a clustered columnstore table.* Además, la carga de datos en una tabla temporal es una operación mucho más rápida que la carga de una tabla en un almacenamiento permanente. Después de cargar los datos, puede crear índices en la tabla para obtener un mejor rendimiento de las consultas.

Las tablas con Clustered columnstore index empiezan a lograr una compresión óptima cuando hay más de 60 millones de filas. En el caso de tablas de búsqueda pequeñas, menos de 60 millones filas, considere la posibilidad de usar HEAP o el Clustered index para mejorar el rendimiento de las consultas.

Para crear una tabla Heap, solo es preciso especificar HEAP en la cláusula WITH:

```
CREATE TABLE myTable
(
    id int NOT NULL,
    lastName varchar(20),
    zipCode varchar(6)
)
WITH ( HEAP );
```

Clustered and nonclustered indexes

Los Clustered index pueden ser mejores que las tablas con Clustered columnstore index cuando se necesita recuperar rápidamente una sola fila. En el caso de las consultas en las que se deba efectuar una búsqueda en una sola o en muy pocas filas a gran velocidad, considere la posibilidad de incorporar un Clustered index o un Nonclustered index. La desventaja de utilizar un Clustered index es que solo se benefician las consultas que utilicen un filtro muy selectivo en la columna del Clustered index.

Para mejorar el filtro en las restantes columnas se puede agregar un Nonclustered index a las mismas. Sin embargo, cada índice que se agrega a una tabla agrega espacio y tiempo de procesamiento a las cargas.

Para crear una tabla de Clustered index, solo es preciso especificar CLUSTERED INDEX en la cláusula WITH:

```
CREATE TABLE myTable
(
  id int NOT NULL,
  lastName varchar(20),
  zipCode varchar(6)
)
WITH ( CLUSTERED INDEX (id) );
```

[Indexación de tablas - Azure Synapse Analytics | Microsoft Docs](#)

Tomar en cuenta las siguientes recomendaciones para utilizar un CLUSTERED INDEX:

- Puedes crear un Clustered index sólo en una columna específica de una tabla
- Si hay unas cuantas consultas que utilizan una columna muy específica, puedes considerar la creación de un Clustered index
- Esta opción se puede utilizar si no se puede crear un Clustered columnstore index basado en las restricciones de columna.

Tomar en cuenta las siguientes recomendaciones para utilizar un NONCLUSTERED INDEX:

- Si desea mejorar el filtrado en otras columnas, puede crear un Nonclustered index para las otras columnas