

# Ventajas del uso de particiones de tablas en el Dedicated SQL Pool

---

## ¿Qué son las particiones de tablas?

Las particiones de tablas permiten dividir los datos en grupos más pequeños de datos. En la mayoría de los casos, se crean particiones de tablas en una columna de fecha. La creación de particiones es compatible con todos los tipos de tabla del Dedicated SQL Pool, entre los que se incluyen el Clustered columnstore index, el Clustered index y el Heap. La creación de particiones también es compatible con todos los tipos de distribución, incluidos la distribución por Hash o la distribución Round Robin.

La creación de particiones puede beneficiar el mantenimiento de datos y el rendimiento de las consultas. El hecho de que beneficie a ambos aspectos, o solo a uno de ellos, dependerá de la forma en que se carguen los datos y de la misma columna se puede usar para ambos propósitos, ya que la creación de particiones solo se puede hacer en una columna.

## Ventajas para las cargas

La principal ventaja de la creación de particiones en el Dedicated SQL Pool es que se mejora la eficiencia y el rendimiento de la carga de datos mediante el uso de la eliminación, el switching y el merging de particiones. En la mayoría de los casos, la creación de particiones de los datos se realiza en una columna de fecha que está estrechamente relacionada con el orden con el que se cargan los datos en el SQL Pool. Una de las mayores ventajas de utilizar particiones para mantener los datos es que se evita el registro de transacciones. Aunque la mera inserción, actualización o eliminación de datos puede ser el enfoque más sencillo, con un poco de esfuerzo, el uso de la creación de particiones durante el proceso de carga puede mejorar considerablemente el rendimiento.

El switching de particiones se puede utilizar para quitar o reemplazar rápidamente cualquier sección de una tabla. Por ejemplo, una tabla de datos de ventas podría contener datos solo de los últimos 36 meses. Al final de cada mes, el mes más antiguo de datos de ventas se elimina de la tabla. Estos datos pueden eliminarse mediante el uso de una instrucción **delete** para eliminar los datos del mes más antiguo.

Sin embargo, la eliminación de una gran cantidad de datos fila a fila con una instrucción delete puede tardar mucho tiempo, así como crear el riesgo de transacciones grandes que podría tardar mucho tiempo en revertirse si algo va mal. **Un enfoque mejor es eliminar (dropear) la partición de datos más antigua.** Aunque la eliminación de las filas individuales podría tardar horas, la de toda una partición puede tardar segundos.

## Ventajas para las consultas

La creación de particiones también puede utilizarse para mejorar el rendimiento de las consultas. Una consulta que aplica un filtro a los datos con particiones puede limitar el análisis a solo las particiones idóneas. Este método de filtrado puede evitar un recorrido de tabla completa y examinar solo un subconjunto más pequeño de datos. Con la introducción de los Clustered columnstore indexes, las ventajas de rendimiento de la eliminación del predicado son menores, pero en algunos casos puede haber beneficios para las consultas.

Por ejemplo, si la tabla de datos de ventas se particiona en 36 meses mediante el campo de fecha de ventas, las consultas que se filtren por esa fecha de ventas pueden omitir la búsqueda en las particiones que no coincidan con el filtro.

## Tamaño de la partición

Aunque la creación de particiones se puede usar para mejorar el rendimiento en algunos escenarios, la creación de una tabla con demasiadas particiones puede afectar negativamente al rendimiento en algunas circunstancias. Estas cuestiones se dan especialmente en las tablas con Clustered columnstore index.

Para que la creación de particiones sea útil, es importante saber cuándo usarla y el número de particiones que se deben crear. No hay ninguna regla inamovible con respecto a cuántas particiones son demasiadas, depende de los datos y del número de particiones en que se realiza la carga de manera simultánea. Normalmente, un esquema de partición correcta tiene decenas o centenas de particiones, no miles.



Al crear particiones en tablas con Clustered columnstore index, es importante considerar el número de filas que pertenece a cada partición. Para que tanto la compresión como el rendimiento de las tablas con Clustered columnstore index sean óptimos, se necesita un mínimo de un millón de filas por partición y distribución. Antes de que se creen particiones, el Dedicated SQL Pool ya divide cada tabla en 60 bases de datos distribuidas.



Todas las particiones que se agreguen a una tabla se sumarán a las distribuciones creadas en segundo plano. Usando este ejemplo, si la tabla de datos de ventas contenía 36 particiones mensuales, y dado que un Dedicated SQL Pool tiene 60 distribuciones, la tabla de datos de ventas debería contener 60 millones de filas por mes, o 2100 millones de filas cuando se rellenen todos los meses. Si una tabla contiene menos filas que el número mínimo recomendado de filas por partición, considere utilizar menos particiones para hacer que aumente el número de filas por partición.

Para obtener más información, consulte el artículo [Indexación](#), que incluye las consultas que pueden evaluar la calidad de los índices de almacén de columnas en clúster.

[Creación de particiones de tablas en el grupo de SQL dedicado - Azure Synapse Analytics | Microsoft Docs](#)

## Example

60 Distributions  365 Partitions  21900 Data Buckets

21900 Data Buckets  Ideal Segment Size (1M Rows)  21 900 000 000 Rows



Lower Granularity (week, month) can perform better depending on how much data you have.