

# Dedicated SQL Pool y Serverless SQL Pool

---

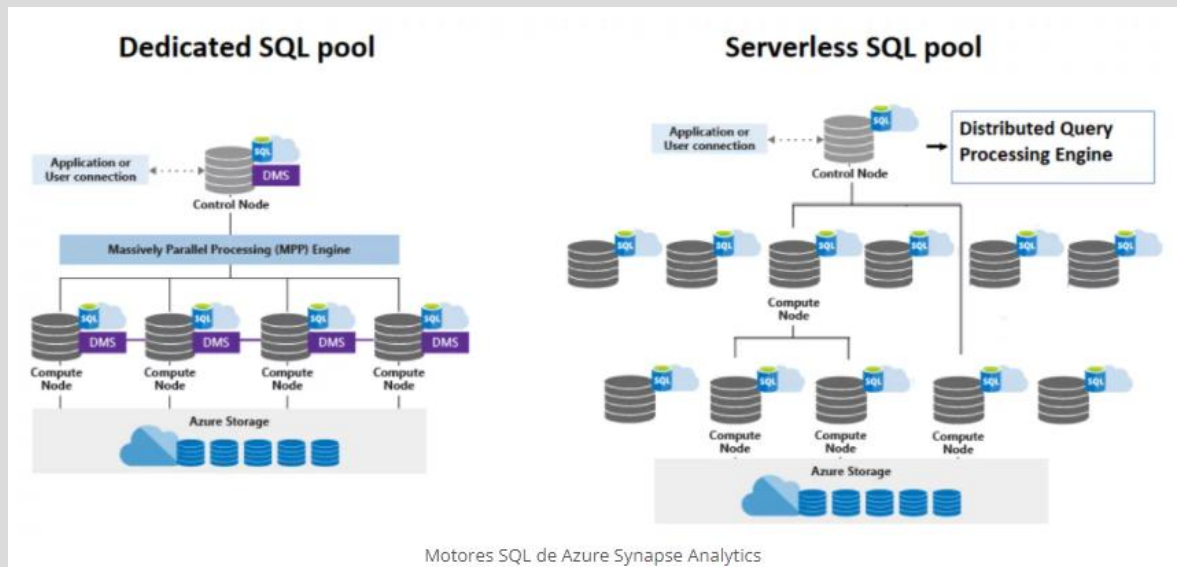
## Data Warehouse con Azure Synapse

Como hemos visto, Synapse tiene dos variantes del SQL Warehouse. Ambas están caracterizadas por separar el cómputo del almacenamiento. Existe un nodo de control que utiliza un motor SQL distribuido para optimizar nuestras consultas para ejecutar en paralelo. Ambas opciones nos proporcionan un **conector JDBC/ODBC** para conectarnos a través de cualquier cliente SQL compatible, por ejemplo, Power BI.

### SQL Dedicado

- El SQL dedicado usa el DWU como unidad de coste y de poder de cómputo. Se calcula a partir de la CPU, memoria e IO de los hosts. Synapse nos permite elegir clústers desde DW100c hasta DW30000c.
- Los ficheros de datos ingestados se deben **distribuir en particiones** para optimizar el rendimiento de las consultas. En concreto, al crear una tabla debemos especificar si queremos distribuir las particiones con una estrategia Hash, round robin o replicadas.
- Dedicated SQL Pool es básicamente un **motor de consulta distribuido** que puede utilizar para realizar análisis de big data de alto rendimiento utilizando el conocido lenguaje T-SQL. A diferencia del Serverless SQL Pool, el Dedicated SQL Pool también ofrece una solución de almacenamiento de datos que almacena los datos en una estructura de tabla con columnas (en tablas), es decir, podemos persistir nuestra data.
- La diferencia clave entre Serverless SQL Pool y el Dedicated SQL Pool es que el Dedicated SQL Pool viene con un almacenamiento interno.
- Así que **es básicamente un Data Warehouse tradicional con un motor de procesamiento masivamente paralelo**. Algunos de ustedes ya pueden estar familiarizados con su antiguo nombre como su SQL Data Warehouse antes de que se incorporara a Synapse como Dedicated SQL Pool.

- Si no estas usando el Dedicated SQL Pool puedes pausarlo cuando lo necesites, para no incurrir en costos innecesarios. En el caso de que ya no lo necesites, puedes eliminarlo y luego volver a crearlo cuando lo necesites nuevamente.
- También tenemos la opción para crear EXTERNAL TABLES.



- Veamos ahora la arquitectura del Dedicated SQL Pool. Utiliza la arquitectura de Driver node y Worker node que generalmente se ve en la computación distribuida, sólo que el driver aquí se llama Control node y el worker se llama Computer node. Las aplicaciones o usuarios se conectan al controlador, que ejecuta el motor de codificación distribuida. Una vez que el Control node recibe la consulta, la optimiza y luego pasa las instrucciones al Compute node para que haga su trabajo en paralelo. Una vez que los Compute nodes completan las tareas, los resultados se devuelven al Control node. A continuación, devuelve el resultado a la aplicación o al usuario.
- El clúster dedicado también nos permite aumentar o disminuir el número de nodos sin mover los datos, simplemente se redistribuyen las asignaciones de las particiones en los nodos disponibles, que pueden llegar hasta 60. Podemos detener el clúster cuando no vayamos a usarlo y volver a arrancarlo bajo demanda, pagando solamente por el almacenamiento.

## SQL Serverless

- Utilice esta opción para realizar análisis ad hoc rápidos de los datos.
- Aquí se le cobra en función de lo que se utilice. El cargo se basa en la cantidad de datos que procesas.
- En el caso del clúster SQL Serverless, el motor SQL distribuido divide la consulta en tareas, que leen los datos del almacenamiento, realiza agregaciones y ordena los resultados de otras tareas. **Este clúster permite definir tablas externas para exponer el almacenamiento con una interfaz SQL.**
- **En el Serverless SQL pool, sólo puedes crear algo conocido como tablas externas. si quieres persistir los datos en una tabla real, tienes que utilizar el Dedicated SQL pool.**
- En el clúster SQL Serverless no es necesario controlar su ejecución o escalado, ya que lo hace automáticamente y de forma transparente. El rendimiento es menor que en el caso del dedicado, ya que **no implementa la estrategia de particionado de datos.**