My Notes on this Section

Exam Perspective

SQL Server optimization has been removed from the syllabus on 31st July 2020, but you can still see questions based on this topic.

There are many questions in this section that will check once again your Azure SQL Data warehouse distribution knowledge, if you understand the difference between Round-robin, hash, and replicated.

You will see one question on TTL And maybe 1 or 2 questions on the access tier, both topics are very easy, you should not miss these questions.

My Notes

Time to live TTL, Azure Cosmos DB provides the ability to delete items automatically from a container after a certain time period.

By default, you can set time to live at the container level and override the value on a per-item basis.

After you set the TTL at a container or at an item level, Azure Cosmos DB will automatically remove these items after the time period, since the time they were last modified.

Time to live value is configured in seconds.

Examples

This section shows some examples with different time to live values assigned to container and items:

Example 1

TTL on container is set to null (DefaultTimeToLive = null)

TTL on item	Result
ttl = null	TTL is disabled. The item will never expire (default).
ttl = -1	TTL is disabled. The item will never expire.
ttl = 2000	TTL is disabled. The item will never expire.

Example 2

TTL on container is set to -1 (DefaultTimeToLive = -1)

TTL on item	Result
ttl = null	TTL is enabled. The item will never expire (default).
ttl = -1	TTL is enabled. The item will never expire.
ttl = 2000	TTL is enabled. The item will expire after 2000 seconds.

Example 3

TTL on container is set to 1000 (DefaultTimeToLive = 1000)

TTL on item	Result
ttl = null	TTL is enabled. The item will expire after 1000 seconds (default).
ttl = -1	TTL is enabled. The item will never expire.
ttl = 2000	TTL is enabled. The item will expire after 2000 seconds.

Time to Live (TTL) in Azure Cosmos DB
Configure time to live in Azure Cosmos DB

Stream analytics best practices

You should start with six SUs for queries that do not use PARTITION BY. This is considered a best practice.

You should allocate more SUs than you need. This is another best practice.

You should keep the SU metric below 80 percent. This allows Streaming Analytics to account for usage spikes
Understand and adjust Streaming Units

ExpressRoute.
This creates a dedicated link between your on-premises datacenter and Azure. This improves performance when copying data to Azure.

Tuning Azure Data Lake Storage Gen2 for performance

AD Connect allows you to synchronize user accounts between on-premises AD and Azure AD.

Access Tiers
The **Archive tier** is optimized for storing data that is rarely accessed and that is kept for at least 180 days.

The **Cool tier** is optimized for storing data that is accessed infrequently and that is kept for at least 30 days.

The **Hot tier** is optimized for storing data that is accessed frequently.
Azure Blob storage: hot, cool, and archive access tiers

Azure SQL Server
A columnstore index stores column values and increases the aggregate queries that use theses indexes.
Memory-optimized tables store all data and schema in memory, increasing the performance for queries.
Partitioned view can be used to split large tables across multiple smaller tables.
Non-clustered index is generally used to increase filter performance and to lookup rows with specific values.
A heap is a table without a clustered index with table data stored without any specific order. Every query in a heap should perform a table scan. A full table scan has better performance than a table with a clustered index for small tables.

Azure SQL Data Warehouse

Fasted loading process is Polybase.

You can write and run PolyBase T-SQL commands. This is a fully parallel operation and is the fastest option.

You can also use a Copy Activity in Azure Data Factory with the copy method set to PolyBase. This option creates and executes the Polybase commands automatically. This also offers the fastest performance.

The following options are not parallel operations and are thus slower:

- BCP
- SQL BulkCopy API
- SSIS
- Azure Data Factory using a Copy Activity and the bulk insert option

References

Data loading strategies for data warehousing
Tutorial: Load the New York Taxicab dataset