**Azure Synapse Dedicated SQL Pools: Workload Management**

**(Source:**[**Doc**](https://sql-stijn.com/2021/12/15/azure-synapse-dedicated-sql-pools-workload-management/)**)**

Workload management is one of the tools you can use to ensure a smooth-running workload by leveraging different groups with classifiers.

**workload group?**

In Dedicated SQL pools you can define 8 custom workload groups in which you can handle resource allocation.

Configurable options for Workload Groups:

1. *Min. Resources%*
2. *Min. & Max Resources % per Request*
3. *Cap Resources %*
4. *Concurrency Range*
5. *Effective Values*
6. *Classifiers*
7. *Request Importance*
8. *Query Execution Timeout*

**Min Resources %**

This setting will allow you to reserve a certain percentage of your total capacity for your workload group.

Side effect : If you are not using the workload group capacity you will lose that capacity.

**Min. & Max Resource per Request**

This setting will allow you to set the min & max percentage of your capacity for each request in the workload group. By default, the Max Percentage per request will be set the same as the Min Percentage per request.

When we alter the Max Percentage per request from the default to a higher value than the Min Percentage per Request, each request will take resources between the min and max percentage per request.

**Cap Resources %**

With the Cap Resources %, you can set the maximum number of resources (DWU) a resource group can possibly consume. You can set multiple resource groups to consume 100% of the resources, if the resources are taken then queuing with priority will apply or the first in first out principle.

**Concurrency Range**

The concurrency range is not really a setting as such; however, it is calculated based on the settings you have given to your workload group. It will calculate the min and max number of concurrent queries you can potentially run with this workload group.

**Effective Values**

The effective values show us the actual values which are currently being used by the workload group management. These can differ from the actual values that we entered as settings for our workload group.

**Classifiers**

if you do not have a classifier no workload groups will be used. To configure users to be “classified” into a workload group, you need to configure classifiers. One workload group can have multiple classifiers and a classifier can be set at user/group/role level.

A classifier has the following settings which can be configured

**-Name**

This is the name of your classifier

**-Member**

This can be a single user (AD or SQL), an AD group or a role. All users configured as members can potentially use the workload group they have been configured in as a classifier

**-Request Importance**

This can be set to low, below normal, normal, above normal and high. When all concurrency slots of the workload group are used, by default all queries are queued and will work in a First in First out principle. With Request importance a classifier with a higher priority will jump the queue and get their queries executed before queued queries with lower importance.

**-Label**

When using the Option(LABEL) in your queries equal to this configured value, these queries can be classified by their label as well. The label option is an optional parameter

**-Session Context**

When using the sys.sp\_set\_session\_context, you can classify queries to this workload group as well if the set sessions context value is equal to the configured value. The Session Context option is an optional parameter

**-Start time and End Time**

When setting a start time and an end time, your user will be classified against this workload group only between the specified timeframes. This option is also an optional parameter

If multiple of these classifiers exist for the same user a weighting exercise is performed. The following table represents how the weighting happens:

Classifier Parameter Weight

USER 64

ROLE 32

WLM\_LABEL 16

WLM\_CONTEXT 8

START\_TIME/END\_TIME 4

**Request Importance**

Request importance is a setting which comes into play when all requested resources are not available on the server or when the cap limit of the resource group has been reached.

The request importance will bypass the first in first out mechanism and queries will be able to jump the queue based on their importance. Queries with the same importance level will go back to the First in First Out principle on their importance level.

**Query Execution Timeout in Sec**

To ensure that in a workload you do not get queries running for hours/days we can set the query to timeout after a certain time has passed.