

## Activity 06: Monitor & Manage

### Audience Poll 1

Q: You want to use a UI to review the assigned workload group and importance for a running SQL query. Which option should you use (pick only one)?

- A) Log Analytics
- B) Monitoring Hub SQL requests
- C) Query the `sys.dm_pdw_exec_requests` DMV

### Audience Poll 2

Q: What information is NOT provided when querying the `sys.dm_pdw_exec_requests` DMV (pick only one)?

- A) Count of sessions by user
- B) Queued, active or complete queries
- C) Query command text

### Audience Poll 3

Q: which of these can you not monitor directly from the Monitor hub (pick only one)?

- A) Pipeline Runs
- B) SQL request
- C) Notebook executions

### Audience Poll 4

Q: How do users release Spark resources used by their notebook (pick only one)?

- A) Stop any cells running in the notebook
- B) Publish the notebook
- C) Close the notebook in Studio

### Audience Poll 5

Q: Which of the following requests will have the highest priority in executing (pick only one)?

- A) A user with high importance
- B) A role with high importance
- C) A role with a resource class of `largerc`

### Audience Poll 6

Q: The customer is claiming that they are running multiple requests in a single session, but getting different classification results, is this possible (pick only one)?

- A) Yes
- B) No

### Audience Poll 7

Q: The customer is observing that critical requests that involve a high degree of locking scheduled first are being pre-empted by other query requests. How can the fix this (pick only one)?

- A) It cannot be controlled
- B) Assign the query requests to a user with normal importance

C) Assign the locking requests to a user with above\_normal or high importance

## Discussion Points

Topic	Discussion Comment
Monitoring - SQL	You can monitor active SQL requests using the SQL requests area of the Monitor Hub. This includes details like the pool, submitter, duration, queued duration, workload group assigned, importance and the request content. Query <code>sys.dm_pdw_exec_sessions</code> to list open and closed sessions, and retrieve the count of sessions by user. Query <code>sys.dm_pdw_exec_requests</code> to retrieve query details like listing all queued, active or complete queries, finding the longest running queries and viewing the query command text. Query
Monitoring SQL - Using DMVs	<code>sys.dm_pdw_nodes_os_performance_counters</code> for performance counters including memory and CPU utilization. Query <code>sys.dm_pdw_waits</code> to see which resources a request is waiting for and <code>sys.dm_pdw_resource_waits</code> to see wait information for a given query like the number of concurrency slots used and resource class assigned. Use <code>sys.dm_pdw_wait_stats</code> for historic trends analysis of waits.
Monitoring - Pipeline Runs	You can monitor pipeline runs using the Monitor Hub and selecting Pipeline runs. Here you can filter pipeline runs and drill in to view the activity runs associated with the pipeline run and monitor the running of in-progress pipelines.
Monitoring - Spark applications	You can monitor the execution Spark applications representing the execution of notebooks and jobs within the Monitor Hub, selecting Spark applications. Selecting a Spark application to view its progress and to launch the Spark UI to examine a running Spark job and stage details, or the Spark history server to examine a completed application.
Workload Classification - Load and query classification and subclassification	Loads (insert, update, delete), Query (Select). Sub classes of loads such as Data Pipeline Loads and Transformations. Sub classes of queries like ad-hoc queries, dashboard queries, cube refreshes
Workload Classification - Importance	Different importance levels assigned to workload classifications, five point scale low, below_normal, normal, above_normal, high. Requests not assigned explicit importance default to normal. Concurrent requests having same importance level are scheduled just like any other requests would be without workload classification.
Workload Classification - Unclassified statements	DBCC commands, BEGIN, COMMIT and ROLLBACK TRANSACTION are not classified
Workload Classification - Approaches	Use <code>sp_addrolemember</code> to map login to resource class OR use CREATE WORKLOAD CLASSIFIER to assign both importance and resource class to requests
Workload Classification - Evaluation	Classification is evaluated on a per request basis. Multiple requests in a single session can be classified differently.
Workload Classification - Precedence	Database user classification takes precedence over role membership classification. When multiple classes might be applicable to a user, user is given highest resource class assignment.
Workload Classification - System classifiers	The pre-defined databases roles that implement resource classes (e.g. smallrc, mediumrc, staticrc10, staticrc80) map by default to the normal importance level
Workload Classification - Mixing resource class assignments with classifiers	It is not a best practice to mix resource class role mappings with workload classifiers as this may create confusion and yield seemingly unexpected results. You should drop the existing resource class mappings and instead use just workload classifiers.
Workload Importance - Impact on Locking	By default Azure Synapse Analytics optimizes for throughput, so even though a higher locking need request might be scheduled first, other requests with lower locking needs may bypass it. Workload importance can be used to ensure the order, such that requests with high locking needs occur first by assigning those requests higher importance than other requests. Higher importance requests will always run before requests with lower importance.
Workload Importance - Impact on concurrent requests with different resource classes	Under the same importance, Azure Synapse Analytics optimizes for throughput. When mixed size requests (such as smallrc or mediumrc) are queued, Synapse will choose the earliest arriving request that fits within the available resources. If workload importance is applied, the highest importance request is scheduled next.
Workload Importance - Monitoring assignment of	Use <code>sys.dm_pdw_exec_requests</code> dynamic management view and examine the importance column.

importance to  
requests

Concurrency

To ensure each query has enough resources to execute efficiently, Azure Synapse Analytics tracks resource utilization by assigning concurrency slots to each query. The system puts queries into a queue based on importance and concurrency slots. Queries wait in the queue until enough concurrency slots are available. Importance and concurrency slots determine CPU prioritization.