Querying and Tuning Azure SQL Data Warehouse

Warner Chaves
SQL MCM / MS DATA PLATFORM MVP
@warchav sqlturbo.com

What's in This Module?

Concurrency and transaction model

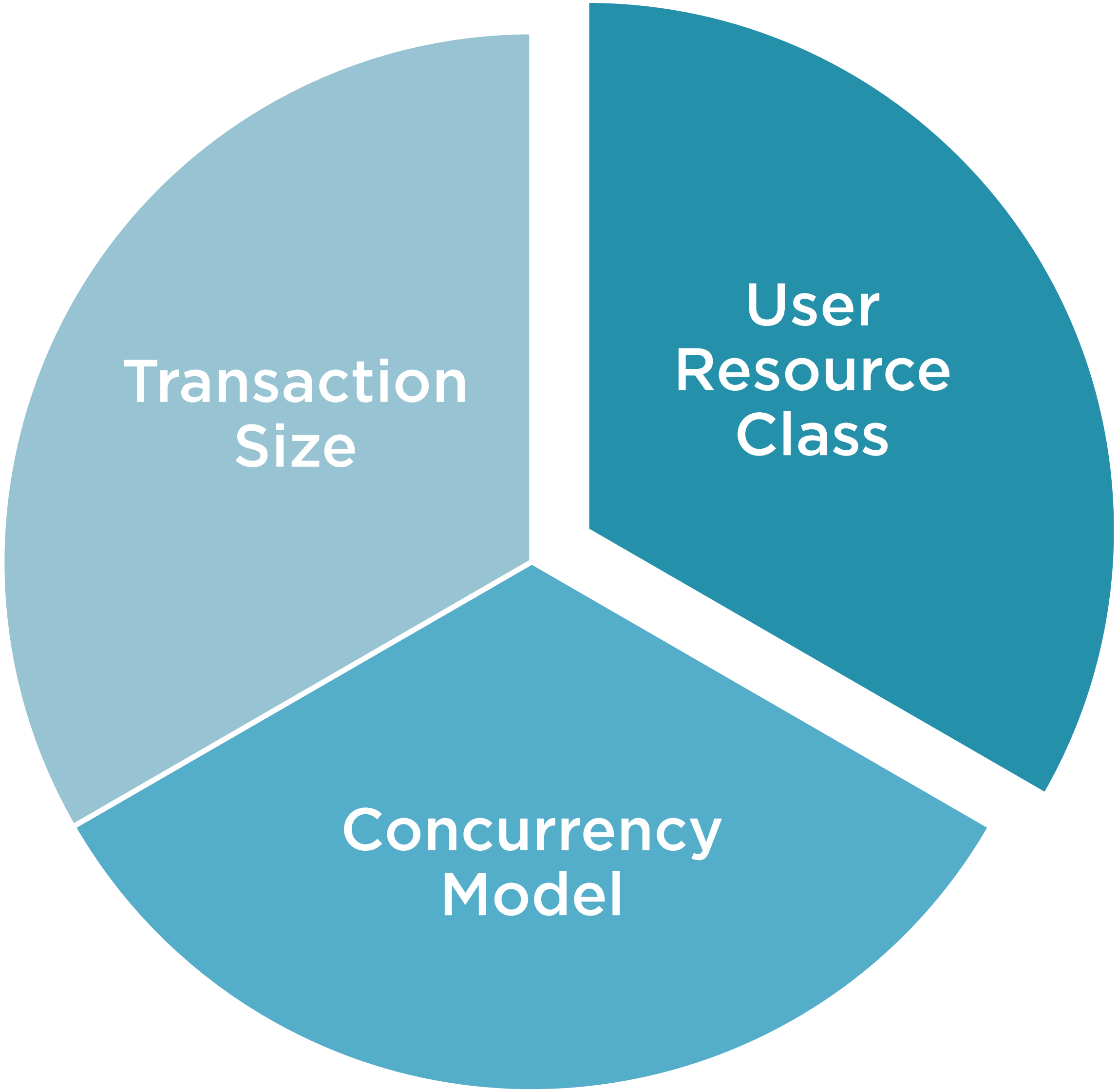
Maintenance operations

Tracking and scaling query performance

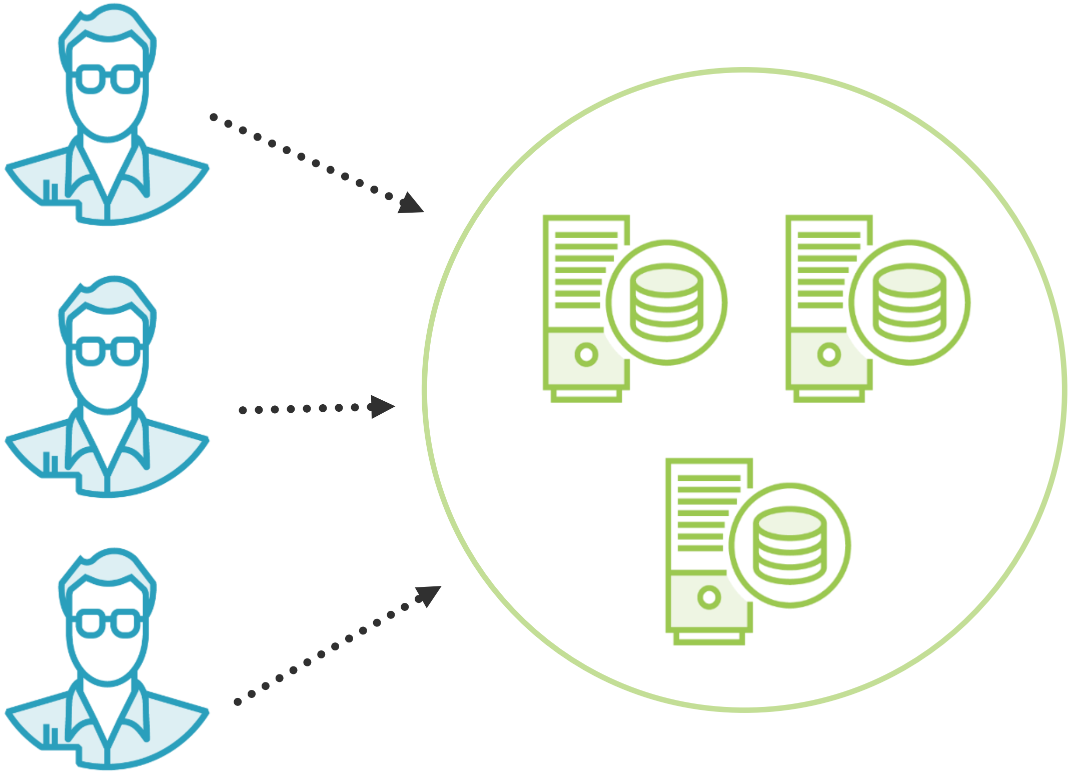
Workload Management

Users belong to a resource class and submit queries to the service.

Azure needs to execute the queries, provide predictable performance and control the resources of the system.


User Resource Class
Transaction Size
Concurrency Model

Concurrency Model



Controls how many queries can execute at any given moment.

Two Maximum Limits

1024 Connections

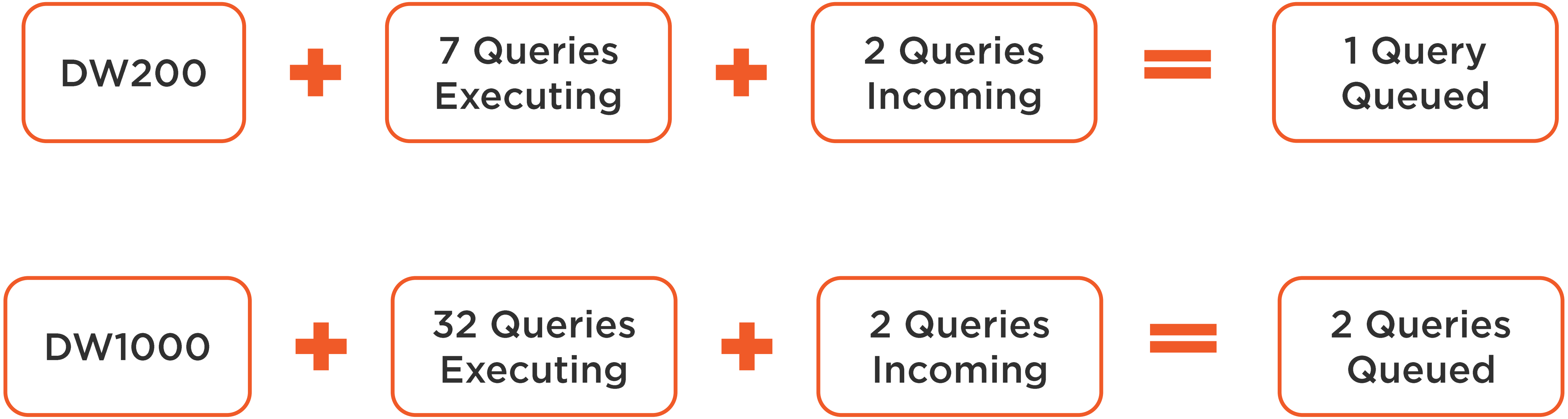
32 Concurrent Queries

Concurrency Queries and Concurrency Slots

Number of:	DWU									
	100	200	300	400	500	600	1,000	1,200	1,500	2,000
Slots:	4	8	12	16	20	24	40	48	60	80

If there are more than 32 concurrent queries OR you exceed the number of concurrency slots then the query will be queued until BOTH thresholds can be satisfied.

For now, let’s assume each query is consuming 1 concurrency slot:



DW200 + 7 Queries Executing + 2 Queries Incoming = 1 Query Queued

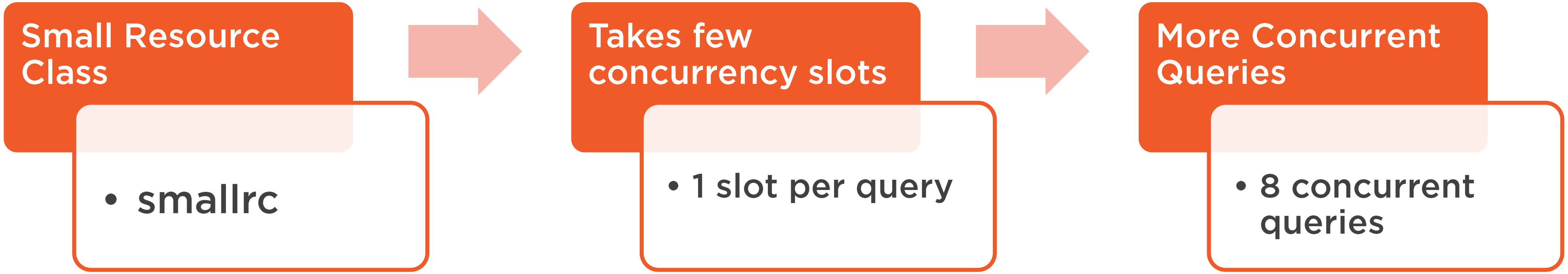
DW1000 + 32 Queries Executing + 2 Queries Incoming = 2 Queries Queued

Slots per Class

Class	Smallrc	Mediumrc	Largerc	Xlargerc
DWUS	100-2000	100-2000	100-2000	100-2000
Slots	1	1 - 16	2 - 32	4 - 64

Resource Class and Concurrency Slots

On a DWU200 system:



Small Resource Class

- smallrc

Takes few concurrency slots

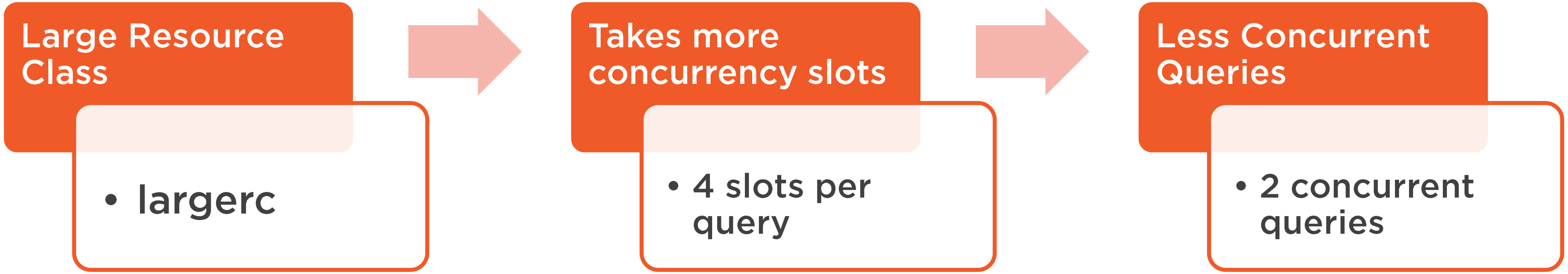
- 1 slot per query

More Concurrent Queries

- 8 concurrent queries

With a Larger Resource Class

On a DWU200 system:



Large Resource Class

- largerc

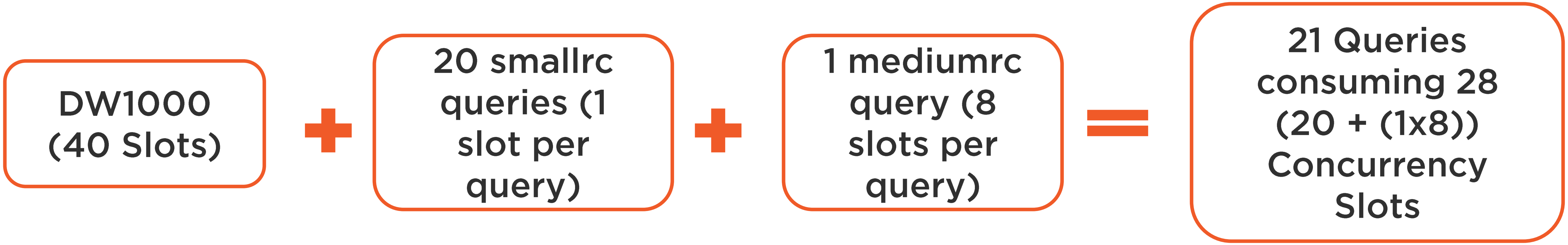
Takes more concurrency slots

- 4 slots per query

Less Concurrent Queries

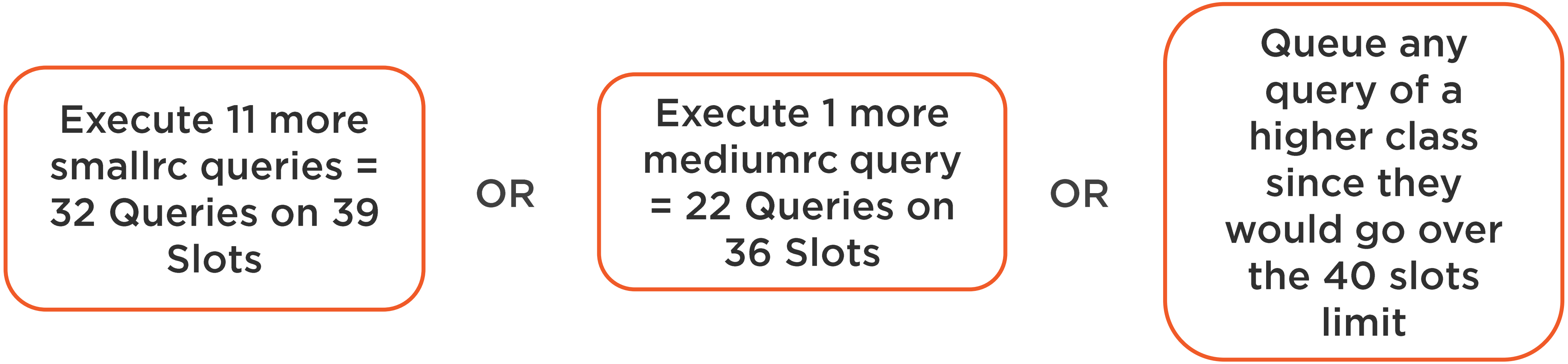
- 2 concurrent queries

Let’s see a more complex workload example:



DW1000 (40 Slots) + 20 smallrc queries (1 slot per query) + 1 mediumrc query (8 slots per query) = 21 Queries consuming 28 (20 + (1x8)) Concurrency Slots

In this state, the following could happen:



Execute 11 more smallrc queries = 32 Queries on 39 Slots

OR

Execute 1 more mediumrc query = 22 Queries on 36 Slots

OR

Queue any query of a higher class since they would go over the 40 slots limit

SELECT queries against system views, stats and other management commands do not use concurrency slots.

Transaction Sizes

Controls how much data can be part of a transaction in the Data Warehouse.

Transaction Limits

Number of:	DWU									
	100	200	300	400	500	600	1,000	1,200	1,500	2,000
GB per distribution:	1	1.5	2.25	3	3.75	4.5	7.5	9	11.25	15

A DW200 transaction doing equal work per distribution could consume 60x1.5GB = 90GB of space

A heavily skewed distribution can make a transaction hit the limit before reaching the entire system-wide transaction limit.

Demo

Monitoring concurrency and queueing levels in the system

Let’s talk about maintenance necessary in Azure SQL Data Warehouse

Types of Stats

Single Column

Multi-Column

Index Stats

Maintaining Statistics

The service does not create or maintain stats automatically, it is up to you to do it.

Stats Operations

Creating new stats

- Sampled single-column stats is a good start.
- Multi-column stats for joins involving multiple columns.
- Focus on columns used in JOINs, GROUP BY, HAVING and WHERE clauses.
- Increase the sample size if necessary.

Updating existing stats

- If new dates or dimension categories are added to the warehouse.
- If new data loads have completed.
- If an UPDATE changes the distribution of data.
- If a DELETE changes the distribution of data.

Index Defrag

Heap

Does not have a Defrag option

B-Tree Index

Useful for removing low levels of fragmentation

Columnstore

Proactively compresses CLOSED rowgroups

Index Rebuild

Heap

Can be rebuilt to remove forward pointers

B-Tree Index

Will remove high levels of fragmentation

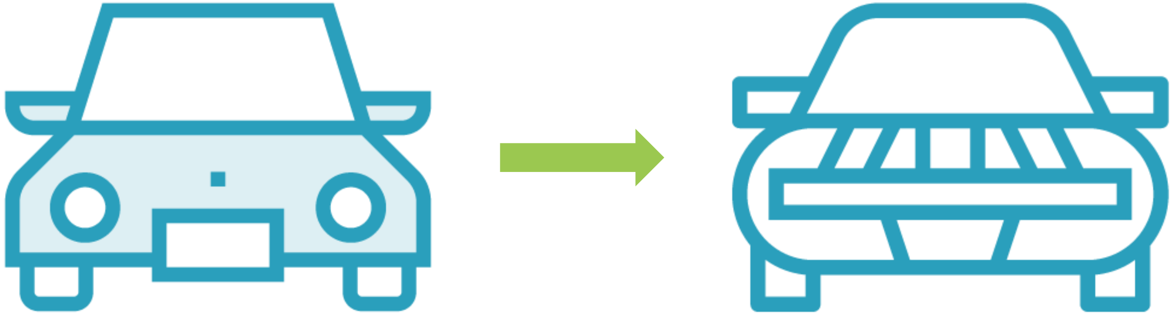
Columnstore

Can increase the density of the segments

Rebuilding an index is an OFFLINE operation in Azure SQL Data Warehouse.

On a large table with heavy fragmentation, it is often faster to recreate the table with CTAS and switch it with the older one than to REBUILD the Index.

Scaling Performance



1. Increase the User Resource Class.
2. Increase the Data Warehouse Units.

Increase User Resource Class

```
EXEC sp_addrolemember 'largerc', 'loaduser'
```

Higher Resource Class – more memory and CPU

More concurrency slots – less concurrent queries

The highest role assigned takes precedence

Increase DWUs

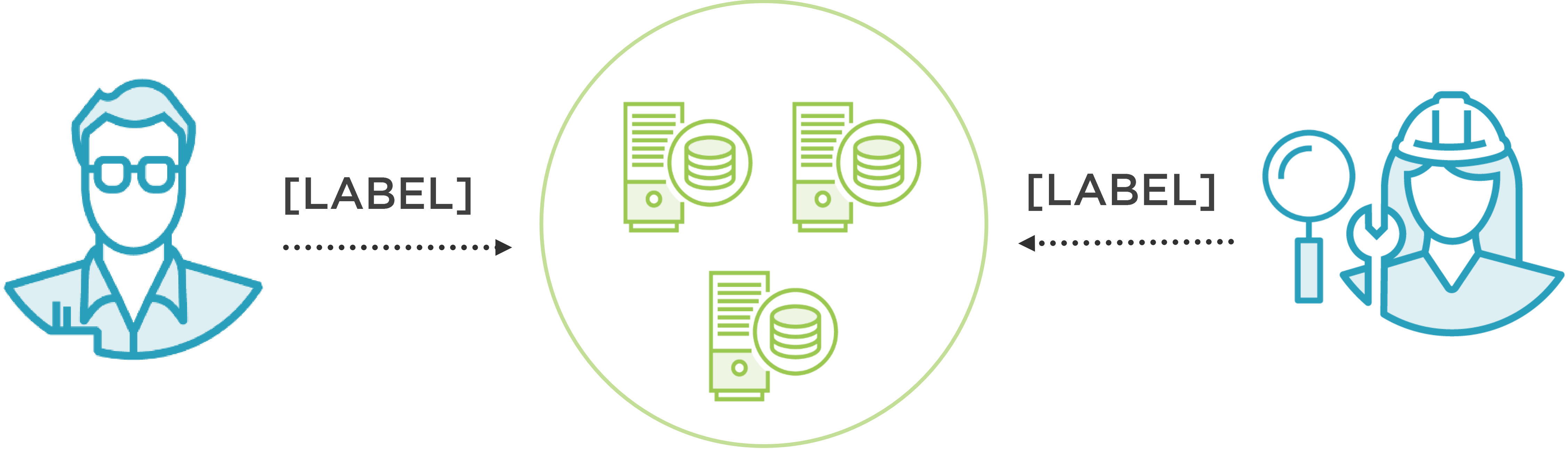
```
ALTER DATABASE ADVENTUREWORKSDW
MODIFY (SERVICE_OBJECTIVE = 'DW1000');
```

It is an OFFLINE operation

Make sure there are no loads or transactions in progress

Can also be done through the Azure Portal

Tracking Queries with Labels



Query Label Example

User Query

```
SELECT sum(Quantity)

FROM FactTransactionHistory

OPTION (LABEL =
'QuantitySum');
```

Admin Query

```
SELECT  *

FROM
sys.dm_pdw_exec_requests r

WHERE   r.[label] =
'QuantitySum';
```

Demo

Labeling a query and tracking its execution

Demo

Scaling up the system for performance

Summary

Azure SQL DW Concurrency Model is very different from SQL Server.

The service imposes a limit on Transaction size as well.

There are several maintenance operations that are still our responsibility.

DWUs can be scaled elastically up and down depending on workload.

Query labels can help troubleshoot issues faster.

Get out there and give Azure SQL Data Warehouse a try!