# Azure SQL Database Performance Tuning

<https://www.faceofit.com/azure-sql-database-performance-tuning/>

**Azure SQL Database Performance Tuning** is an important topic even if the database does not need a dedicated DBA to manage and maintain. There are several automatic optimization options provided by Microsoft which you can leverage to know more about your Database performance. Read along to know more:

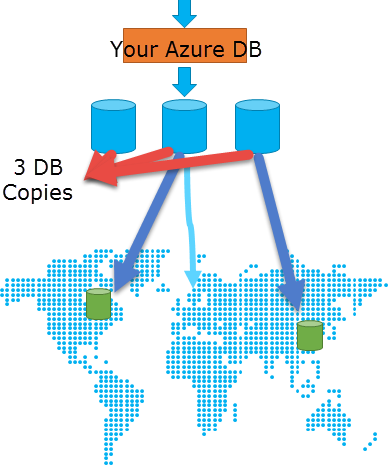
## Azure SQL Database Performance Tuning

Before discussing **Azure SQL Server Performance Tuning**, you need to understand the Azure SQL Architecture and how it is deployed. One might decide to go for Azure SQL Database because of the following reasons:

#### Save Time and Cost

* If you want to avoid purchasing and installing hardware yourself. No Hardware and software installations need to be done. Install and deploy SQL Databases on the fly.
* Azure SQL Database is a “**Database as a Service**” offering and includes in-built automatic management features such as automatic HA (High Availability) and built-in backup management.
* Built-In Performance insights and index recommendation.

**You need to manage Index Fragmentation by yourself. Use the Azure runbook script to automate this job. See**[**more here**](https://gallery.technet.microsoft.com/scriptcenter/Indexes-tables-in-an-Azure-73a2a8ea)**.**

[](https://www.faceofit.com/wp-content/uploads/2016/04/AzureSql1.png)

#### **Microsoft currently offers you the following:**

* Up to 99.99% SLA
* By default, Databases are replicated on three servers within the same location.
* Databases can be Geo-replicated to other data centers. Azure implements Always On Availability Groups under the hood to achieve this.
* Azure manages the underlying infrastructure and you do not need to manage these resources.

***Azure SQL is a multi-tenant deployment and resources are shared across customers.***

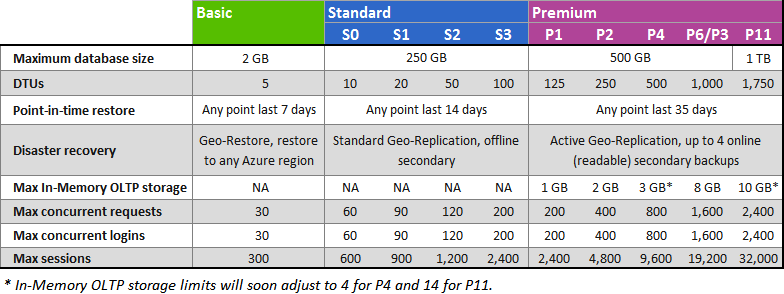
Since you do not have access to the Server hosting the Databases, you have limited options. So you will need to work with the options that you have.

**For example**:

* Use Connection Pooling
* Close Unused connections
* Ensure good network connectivity and less network latency.
* Ensure that you tune your queries and avoid Table Scans. Create appropriate Index’s to force an “Index Seek”
* Filter your results appropriately.

The performance of individual Azure SQL Databases is based on the service tier that you go for. This gives you guaranteed predictable performance. The throughput guaranteed for your database depends upon their respective service tiers, i.e. **Basic**, **Standard** or **Premium**. This guaranteed performance is measured in DTU’s for Azure SQL Database. More DTU’s gives you more **transactions per second**.

***Use***[***http://dtucalculator.azurewebsites.net***](http://dtucalculator.azurewebsites.net/)***to estimate the number of DTU’s that you need.***

*[](https://www.faceofit.com/wp-content/uploads/2016/04/AzureSql5.png)*

*Source: https://azure.microsoft.com/en-in/documentation/articles/sql-database-service-tiers/*

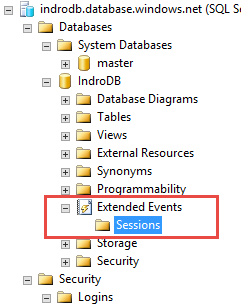
DTU is beyond the scope of this post, please visit here for [**official documentation**](https://azure.microsoft.com/en-in/documentation/articles/sql-database-service-tiers/).

[***See my earlier post on Performance Tuning for SQL Server on Azure Virtual Machines***](https://www.faceofit.com/performance-tuning-best-practices-for-sql-on-azure-vm/)

### **Tools to Monitor SQL Azure Performance:**

#### **Extended Events**

This feature was introduced with SQL Server 2008. On Azure SQL this was made available in 2015 and only available on V12 Databases. This allows you to get more internal information of your SQL Server. Much of the Extended Events for On-Premise SQL Server is similar to Azure SQL Database. See the [official Documentation here](https://azure.microsoft.com/en-in/documentation/articles/sql-database-xevent-db-diff-from-svr/).

[](https://www.faceofit.com/wp-content/uploads/2016/04/AzureSql2.png)

#### **2. Query Store**

This is also available on V12 Databases. The Query Store gives you performance insights on historical telemetry information collected by the Azure SQL service. Using this service you can:

* + Review the list of top resource-consuming queries.
  + Ability to select an individual query to view its details.

##### [Azure SQL Database Performance Tuning](https://www.faceofit.com/wp-content/uploads/2016/04/AzureSql3.png)

#### **3. Dynamic Management Views or DMV’s**

This allows you to see “point in time” information for your SQL Server queries. You need to set the Database context to your “Master Database” and then query the system tables for more database insights.

Here are some examples:

**Max concurrent requests**

SELECT COUNT(\*) AS [Concurrent\_Requests]

FROM sys.dm\_exec\_requests R

Max Concurrent requests for a specific DB

SELECT COUNT(\*) AS [Concurrent\_Requests]

FROM sys.dm\_exec\_requests R

INNER JOIN sys.databases D ON D.database\_id = R.database\_id

AND D.name = 'YourDBName'

Max Sessions

SELECT COUNT(\*) AS [Sessions]

FROM sys.dm\_exec\_connections C

INNER JOIN sys.dm\_exec\_sessions S ON (S.session\_id = C.session\_id)

INNER JOIN sys.databases D ON (D.database\_id = S.database\_id)

WHERE D.name = 'MyDatabase'

Missing Index's

SELECT CONVERT (varchar, getdate(), 126) AS runtime,

mig.index\_group\_handle,

mid.index\_handle,

CONVERT (decimal (28,1),

migs.avg\_total\_user\_cost \*

migs.avg\_user\_impact \*

(migs.user\_seeks + migs.user\_scans))

AS improvement\_measure,

'CREATE INDEX missing\_index\_' +

CONVERT (varchar, mig.index\_group\_handle) +

'\_' +

CONVERT (varchar, mid.index\_handle) +

' ON ' +

mid.statement +

' (' + ISNULL (mid.equality\_columns,'') +

CASE WHEN mid.equality\_columns IS NOT NULL

AND mid.inequality\_columns IS NOT NULL

THEN ','

ELSE ''

END + ISNULL (mid.inequality\_columns, '') +

')' +

ISNULL (' INCLUDE (' + mid.included\_columns + ')',

'') AS create\_index\_statement,

migs.\*,

mid.database\_id, mid.[object\_id]

FROM sys.dm\_db\_missing\_index\_groups mig

INNER JOIN sys.dm\_db\_missing\_index\_group\_stats migs

ON migs.group\_handle = mig.index\_group\_handle

INNER JOIN sys.dm\_db\_missing\_index\_details mid

ON mig.index\_handle = mid.index\_handle

WHERE CONVERT (decimal (28,1),

migs.avg\_total\_user\_cost \*

migs.avg\_user\_impact \*

(migs.user\_seeks + migs.user\_scans)) > 10

ORDER BY migs.avg\_total\_user\_cost \*

migs.avg\_user\_impact \*

(migs.user\_seeks + migs.user\_scans) DESC

Monitor I/O CPU or Memory Performance using sys.dm\_db\_resource\_stats

SELECT

AVG(avg\_cpu\_percent) AS 'Average CPU Utilization In Percent',

MAX(avg\_cpu\_percent) AS 'Maximum CPU Utilization In Percent',

AVG(avg\_data\_io\_percent) AS 'Average Data IO In Percent',

MAX(avg\_data\_io\_percent) AS 'Maximum Data IO In Percent',

AVG(avg\_log\_write\_percent) AS 'Average Log Write Utilization In Percent',

MAX(avg\_log\_write\_percent) AS 'Maximum Log Write Utilization In Percent',

AVG(avg\_memory\_usage\_percent) AS 'Average Memory Usage In Percent',

MAX(avg\_memory\_usage\_percent) AS 'Maximum Memory Usage In Percent'

FROM sys.dm\_db\_resource\_stats;

##### ***This telemetry information is recorded every 15 seconds and are maintained for one hour***

##### [Azure SQL Database Performance Tuning](https://www.faceofit.com/wp-content/uploads/2016/04/AzureSql6.png)

##### **4. Index Recommendation**

##### [Azure SQL Database Performance Tuning](https://www.faceofit.com/wp-content/uploads/2016/04/AzureSql4.png)

The Index recommendations that are offered in Azure SQL are recommended based on the existing telemetry collected by Azure SQL Database service. These computations run externally to Azure SQL DB  
and do not use any of the Azure SQL DB resources or create additional transactions. Implementing the index’s based on the recommendation, however, can have some impact like any other databases.

**Here are the advantages and its impact on SQL Performance:**

* Unlike On-Premises, SQL Azure is a shared multi-tenant service which if not tuned properly can have a greater impact in comparison to on-premises SQL deployments. Creating Index’s is crucial for this and one should follow Index recommendation thoroughly.
* You can collect the baseline performance data for comparisons before and after creating the indexes.
* The best feature of this is, in case this creates a negative impact, the index creation is automatically rolled back.

#### Resources Available Online:

* Extended Events in Azure SQL: <https://azure.microsoft.com/en-in/documentation/articles/sql-database-xevent-db-diff-from-svr/>
* Query Store:<https://azure.microsoft.com/en-us/documentation/articles/sql-database-query-performance/>
* Azure SQL Database performance guidance for single databases: <https://azure.microsoft.com/en-in/documentation/articles/sql-database-performance-guidance/>
* SQL Database Index Advisor <https://azure.microsoft.com/en-in/documentation/articles/sql-database-index-advisor/>

We hope that this post was helpful to get you started with **Azure SQL Database Performance Tuning**. If you have any suggestions or comments, please feel free to use the comment section below.