**Describe automatic tuning**

Automatic tuning is a monitoring and analysis feature that continuously learns about your workload and identifies potential issues and improvements.

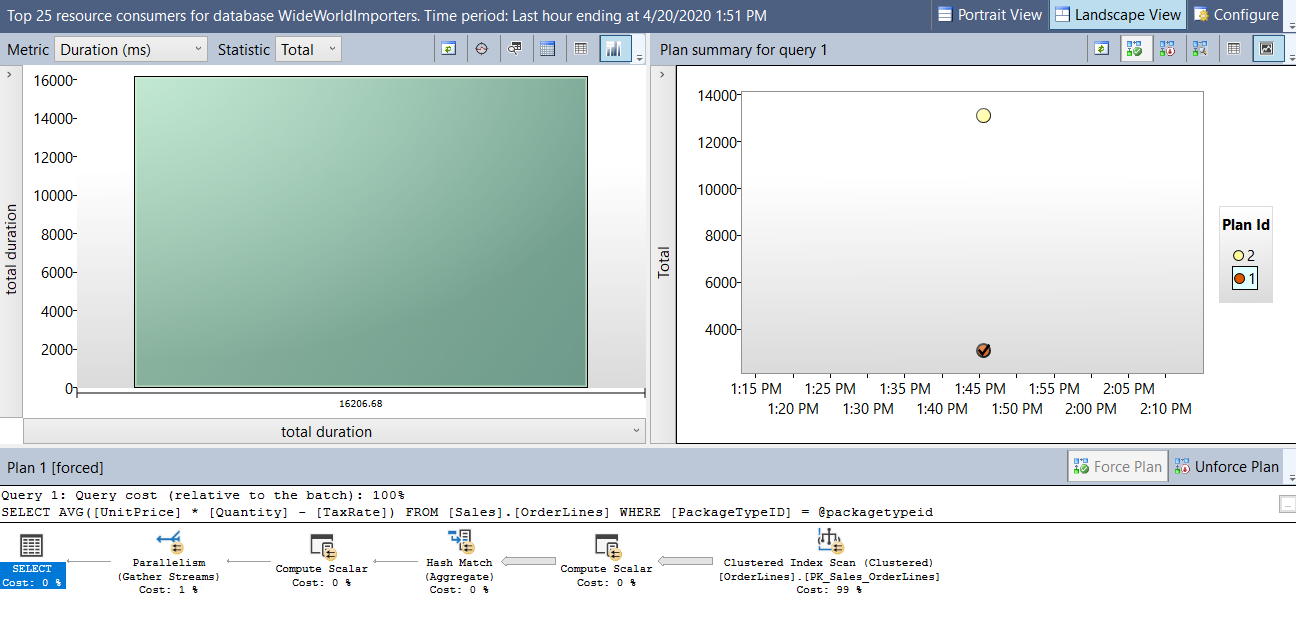
The automatic tuning recommendations are based on the data collected from Query Store. Execution plans evolve over time due to schema changes, index modifications, or changes to the data that cause updates to the statistics. This evolution can cause queries to perform poorly as the execution plan no longer meets the demands of the given query.

Furthermore, automatic tuning allows for the gathering and applying machine learning services against performance metrics to provide suggested improvements or even allow for self-correction.

Whether on-premises or in the cloud, automatic tuning allows you to identify issues caused by query execution plan regression. Additionally, in Azure SQL Database you can improve query performance by index tuning. Azure SQL Database automatic tuning can identify indexes that should be added or even removed from the database to enhance query performance.

**Automatic plan correction**

With the help of the Query Store data, the database engine can determine when query execution plans have regressed in performance. While you can manually identify a regressed plan through the user interface, the Query Store also provides the option to notify you automatically.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/configure-databases-for-optimal-performance/media/module-44-optimize-resources-final-25.png#lightbox)

In the example above, you can see a check mark on **Plan ID 1**, which means that the plan has been forced. After the feature is enabled, the database engine will automatically force any recommended query execution plan, when:

* The previous plan had a higher error rate than the recommended plan
* The estimated CPU gain was greater than 10 seconds
* The force plan has performed better than the previous one

The plan will revert back to the last known good plan after 15 executions of the query.

When plan forcing occurs automatically, the database engine will apply the last known good plan and will also continue to monitor query execution plan performance. If the forced plan doesn't perform better than the previous plan, it will be then unforced and force a new plan to be compiled. If the forced plan continues to outperform the previously bad plan, it will remain forced until such time as a recompile occurs.

You can enable automatic plan correction via a T-SQL query, as shown below. The Query Store must be enabled and must be in Read-Write mode for the command to succeed. If either of those two criteria aren't met, the ALTER statement will fail.

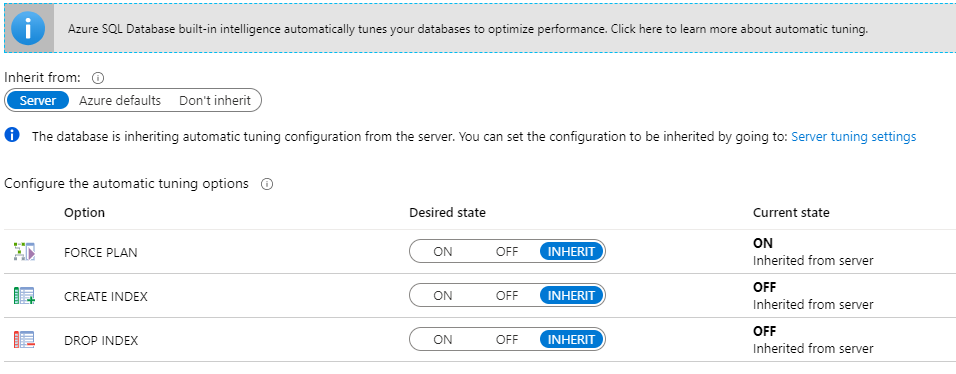
SQLCopy

ALTER DATABASE [WideWorldImporters] SET AUTOMATIC\_TUNING (FORCE\_LAST\_GOOD\_PLAN = ON);

You can examine the automatic tuning recommendations through a dynamic management view (DMV), sys.dm\_db\_tuning\_recommendations, which is available in SQL Server 2017 or higher and is also available in Azure SQL Database solutions. This DMV provides information such as reasons as to why the recommendation was provided, the type of recommendation, and the state of the recommendation. To confirm that automatic tuning is enabled for a database, check the view sys.database\_automatic\_tuning\_options.

**Automatic index management**

Azure SQL Database can perform automatic index tuning. Over time, the database will learn about existing workloads and provide recommendations on adding or removing indexes in order to provide better performance. Like forcing improved query plans, the database can be configured to allow for automatic index creation or removal depending on existing index performance, as shown below:



When enabled, the **Performance Recommendations** page will identify indexes that can be created or dropped depending on query performance. Remember this feature isn't available for on-premises databases and only available for Azure SQL Database.

Alternatively, use the following query to see the automatic tuning features enabled in your database:

SQLCopy

SELECT name,

desired\_state\_desc,

actual\_state\_desc,

reason\_desc

FROM sys.database\_automatic\_tuning\_options

Creating new indexes can consume resources, and the timing of the index creations is critical to ensure no negative effect is felt on your workloads.

Azure SQL Database will monitor the resources required to implement new indexes to avoid causing performance degradation. The tuning action is postponed until the available resources are available, for example if resources are required for existing workloads and not available for creating an index.

Monitoring ensures any action taken won't harm performance. If an index is dropped and query performance noticeably degrades, the recently dropped index will be automatically recreated.