Elastic Query Limitations

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:27 AM PST

Contents

Elastic Query Limitations

Issue

This is an article regarding known limitations and best practices related to Elastic Query. For a more comprehensive article about general troubleshooting, see Wiki article Elastic Query troubleshooting.

Best practices

Review the following items to avoid issues with connectivity, security, and performance.

- **Required:** Ensure that the Elastic Query local database has access to the remote database. In the remote Azure SQL Database server's firewall configuration, you must enable the option "Allow Azure services and resources to access this server", and you must set the option "Public network access" to "Selected networks" (not to "Disabled"). If either "Allow Azure services" or "Public network access" is disabled, the external table cannot connect to the remote database. See the "Connectivity issues" section below for further details about this requirement.
- **Required:** Ensure that the credential provided in the external data source definition can successfully log into the remote database and has sufficient permissions to access the remote table.
- **Security consideration:** Azure Active Directory (AAD) principals are not supported in Elastic Query, only SQL Authentication will work. Managed Identity is in the scope of AAD principals and is also not supported.
- **Security consideration:** Users with access to the external table automatically gain access to the underlying remote tables under the credential given in the external data source definition. Carefully manage access to the external table in order to avoid undesired elevation of privileges through the credential of the external data source. Regular SQL permissions can be used to GRANT or REVOKE access to an external table just as it were a regular table.
- **Performance:** Elastic Query works best for queries where most of the computation can be done on the remote databases. You typically get the best query performance with selective filter predicates that can be evaluated on the remote databases, or joins that can be performed completely on the remote database. Other query patterns may need to load large amounts of data from the remote database and may perform poorly.
- **Performance:** Avoid including `NVARCHAR(max)` columns in the external table. Columns of data type `NVARCHAR(max)` will disable the advanced batching techniques that are used in the Elastic Query

implementation. If an external table includes `NVARCHAR(max)` columns, it will significantly affect the query performance if a large amount of data needs to be transferred from the remote to the local database.

- **Performance:** The performance of Elastic Queries over large databases will be slow in the lower service tiers, like Standard or the smaller General Purpose tiers. Use at least the Premium tier, a higher-level General Purpose size, or Business Critical to receive reliable performance for large databases.
- **Performance:** Elastic Query is not the preferred solution for ETL (Extract - Transform - Load) operations when there is a large amount of data movement from and especially to a remote database. If your workloads include heavy ETL processing, consider moving to Azure SQL Managed Instance instead of using Elastic Query on Azure SQL Database.

Limitations as listed in the public documentation

The list below is copied from the article Elastic Query preview limitations. There is some topic overlap to the "Best Practices" section above though with different wording.

> **Preview limitations**
>
> - Running your first elastic query can take up to a few minutes on smaller resources and Standard and General Purpose service tier. This time is necessary to load the elastic query functionality; loading performance improves with higher service tiers and compute sizes.
> - Scripting of external data sources or external tables from SSMS or SSDT is not yet supported. Import/Export for SQL Database does not yet support external data sources and external tables. If you need to use Import/Export, drop these objects before exporting and then re-create them after importing.
> - Elastic query currently only supports read-only access to external tables. You can, however, use full Transact-SQL functionality on the database where the external table is defined. This can be useful to, e.g., persist temporary results using, for example, SELECT <column_list> INTO <local_table>, or to define stored procedures on the elastic query database that refer to external tables.
> - Except for nvarchar(max), LOB types (including spatial types) are not supported in external table definitions. As a workaround, you can create a view on the remote database that casts the LOB type into nvarchar(max), define your external table over the view instead of the base table and then cast it back into the original LOB type in your queries.
> - Columns of nvarchar(max) data type in result set disable advanced batching technics used in Elastic Query implementation and may affect performance of query for an order of magnitude, or even two orders of magnitude in non-canonical use cases where large amount of non-aggregated data is being transferred as a result of query.
> - Column statistics over external tables are currently not supported. Table statistics are supported, but need to be created manually.
> - Cursors are not supported for external tables in Azure SQL Database.
> - Elastic query works with Azure SQL Database only. You cannot use it for querying a SQL Server instance.
> - Private links are currently not supported with elastic query for those databases that are targets of external data sources.

Internal Doc Reference

- Elastic Query troubleshooting

Public Doc Reference

- Azure SQL Database elastic query overview
- Elastic query preview limitations
- Resolve Elastic Query issues in Azure SQL Database
- Get started with cross-database queries (vertical partitioning)
- Cross-database Query in Azure SQL Database
- Query across cloud databases with different schema (vertical partitioning)
- Reporting across scaled-out cloud databases (horizontal partitioning)
- Lesson Learned #191: Performance comparison using Inner vs Left Join in Linked Server/Elastic Query
- Outbound firewall rules for Azure SQL Database

How good have you found this content?