**Transact-SQL features supported in Azure Synapse SQL**

Azure Synapse SQL is a big data analytic service that enables you to query and analyze your data using the T-SQL language. You can use standard ANSI-compliant dialect of SQL language used on SQL Server and Azure SQL Database for data analysis.

Transact-SQL language is used in serverless SQL pool and dedicated model can reference different objects and has some differences in the set of supported features. In this page, you can find high-level Transact-SQL language differences between consumption models of Synapse SQL.

**Database objects**

Consumption models in Synapse SQL enable you to use different database objects. The comparison of supported object types is shown in the following table:

|  | **Dedicated** | **Serverless** |
| --- | --- | --- |
| **Tables** | Yes | No, the in-database tables are not supported. Serverless SQL pool can query only external tables that reference data stored in Azure Data Lake storage or Dataverse. |
| **Views** | Yes. Views can use query language elements that are available in dedicated model. | Yes, you can create views over external tables, the queries with the OPENROWSET function, and other views. Views can use query language elements that are available in serverless model. |
| **Schemas** | Yes | Yes, schemas are supported. Use schemas to isolate different tenants and place their tables per schemas. |
| **Temporary tables** | Yes | Temporary tables might be used just to store some information from the system views, literals, or other temporary tables. UPDATE/DELETE on temp table is also supported. You can join temporary tables with the system views. You cannot select data from an external table to insert it into temporary table or join a temporary table with an external table - these operations will fail because external data and temporary tables cannot be mixed in the same query. |
| **User defined procedures** | Yes | Yes, stored procedures can be placed in any user databases (not master database). Procedures can just read external data and use query language elements that are available in serverless pool. |
| **User defined functions** | Yes | Yes, only inline table-valued functions are supported. Scalar user-defined functions are not supported. |
| **Triggers** | No | No, serverless SQL pools do not allow changing data, so the triggers cannot react on data changes. |
| **External tables** | Yes. See supported data formats. | Yes, external tables are available and can be used to read data from Azure Data Lake storage or Dataverse. See the supported data formats. |
| **Caching queries** | Yes, multiple forms (SSD-based caching, in-memory, resultset caching). In addition, Materialized View are supported. | No, only the file statistics are cached. |
| **Result set caching** | Yes | No, the query results are not cached. Only the file statistics are cached. |
| **Materialized views** | Yes | No, the Materialized views are not supported in the serverless SQL pools. |
| **Table variables** | No, use temporary tables | No, table variables are not supported. |
| **Table distribution** | Yes | No, table distributions are not supported. |
| **Table indexes** | Yes | No, indexes are not supported. |
| **Table partitioning** | Yes. | External tables do not support partitioning. You can partition files using Hive-partition folder structure and create partitioned tables in Spark. The Spark partitioning will be synchronized with the serverless pool. If you are not using Spark, you can partition your files in folder structure and can create partitioned views on folder partition structure, but the external tables cannot be created on partitioned folders. |
| **Statistics** | Yes | Yes, statistics are created on external files. |
| **Workload management, resource classes, and concurrency control** | Yes, see workload management, resource classes, and concurrency control. | No, you cannot manage the resources that are assigned to the queries. The serverless SQL pool automatically manages the resources. |
| **Cost control** | Yes, using scale-up and scale-down actions. | Yes, you can limit daily, weekly, or monthly usage of serverless pool using the Azure portal or T-SQL procedure. |

**Query language**

Query languages used in Synapse SQL can have different supported features depending on consumption model. The following table outlines the most important query language differences in Transact-SQL dialects:

|  | **Dedicated** | **Serverless** |
| --- | --- | --- |
| **SELECT statement** | Yes. SELECT statement is supported, but some Transact-SQL query clauses, such as FOR XML/FOR JSON, MATCH, OFFSET/FETCH are not supported. | Yes, SELECT statement is supported, but some Transact-SQL query clauses like FOR XML, MATCH, PREDICT, GROUPNG SETS, and the query hints are not supported. |
| **INSERT statement** | Yes | No. Upload new data to Data lake using Spark or other tools. Use Cosmos DB with the analytical storage for highly transactional workloads. You can use CETAS to create an external table and insert data. |
| **UPDATE statement** | Yes | No, update Parquet/CSV data using Spark and the changes will be automatically available in serverless pool. Use Cosmos DB with the analytical storage for highly transactional workloads. |
| **DELETE statement** | Yes | No, delete Parquet/CSV data using Spark and the changes will be automatically available in serverless pool. Use Cosmos DB with the analytical storage for highly transactional workloads. |
| **MERGE statement** | Yes (preview) | No, merge Parquet/CSV data using Spark and the changes will be automatically available in serverless pool. |
| **CTAS statement** | Yes | No, CREATE TABLE AS SELECT statement is not supported in the serverless SQL pool. |
| **CETAS statement** | Yes, you can perform initial load into an external table using CETAS. | Yes, you can perform initial load into an external table using CETAS. CETAS supports Parquet and CSV output formats. |
| **Transactions** | Yes | Yes, transactions are applicable only on the meta-data objects. |
| **Labels** | Yes | No, labels are not supported in serverless SQL pools. |
| **Data load** | Yes. Preferred utility is COPY statement, but the system supports both BULK load (BCP) and CETAS for data loading. | No, you cannot load data into the serverless SQL pool because data is stored on external storage. You can initially load data into an external table using CETAS statement. |
| **Data export** | Yes. Using CETAS. | Yes. You can export data from external storage (Azure data lake, Dataverse, Cosmos DB) into Azure data lake using CETAS. |
| **Types** | Yes, all Transact-SQL types except cursor, hierarchyid, ntext, text, and image, rowversion, Spatial Types, sql\_variant, and xml | Yes, all Transact-SQL types are supported, except cursor, hierarchyid, ntext, text, and image, rowversion, Spatial Types, sql\_variant, xml, and Table type. See how to map Parquet column types to SQL types here. |
| **Cross-database queries** | No | Yes, the cross-database queries and the 3-part-name references are supported including USE statement. The queries can reference the serverless SQL databases or the Lake databases in the same workspace. Cross-workspace queries are not supported. |
| **Built-in/system functions (analysis)** | Yes, all Transact-SQL Analytic, Conversion, Date and Time, Logical, Mathematical functions, except CHOOSE and PARSE | Yes, all Transact-SQL Analytic, Conversion, Date and Time, Logical, and Mathematical functions are supported. |
| **Built-in/system functions (string)** | Yes. All Transact-SQL String, JSON, and Collation functions, except STRING\_ESCAPE and TRANSLATE | Yes. All Transact-SQL String, JSON, and Collation functions are supported. |
| **Built-in/system functions (Cryptographic)** | Some | HASHBYTES is the only supported cryptographic function in serverless SQL pools. |
| **Built-in/system table-value functions** | Yes, Transact-SQL Rowset functions, except OPENXML, OPENDATASOURCE, OPENQUERY, and OPENROWSET | Yes, all Transact-SQL Rowset functions are supported, except OPENXML, OPENDATASOURCE, and OPENQUERY. |
| **Built-in/system aggregates** | Transact-SQL built-in aggregates except, except CHECKSUM\_AGG and GROUPING\_ID | Yes, all Transact-SQL built-in aggregates are supported. |
| **Operators** | Yes, all Transact-SQL operators except !> and !< | Yes, all Transact-SQL operators are supported. |
| **Control of flow** | Yes. All Transact-SQL Control-of-flow statement except CONTINUE, GOTO, RETURN, USE, and WAITFOR | Yes. All Transact-SQL Control-of-flow statements are supported. SELECT query in WHILE (...) condition is not supported. |
| **DDL statements (CREATE, ALTER, DROP)** | Yes. All Transact-SQL DDL statement applicable to the supported object types | Yes, all Transact-SQL DDL statement applicable to the supported object types are supported. |

**Security**

Synapse SQL pools enable you to use built-in security features to secure your data and control access. The following table compares high-level differences between Synapse SQL consumption models.

|  | **Dedicated** | **Serverless** |
| --- | --- | --- |
| **Logins** | N/A (only contained users are supported in databases) | Yes, server-level Azure AD and SQL logins are supported. |
| **Users** | N/A (only contained users are supported in databases) | Yes, database users are supported. |
| **Contained users** | Yes. **Note:** only one Azure AD user can be unrestricted admin | No, the contained users are not supported. |
| **SQL username/password authentication** | Yes | Yes, users can access serverless SQL pool using their usernames and passwords. |
| **Azure Active Directory (Azure AD) authentication** | Yes, Azure AD users | Yes, Azure AD logins and users can access serverless SQL pools using their Azure AD identities. |
| **Storage Azure Active Directory (Azure AD) passthrough authentication** | Yes | Yes, Azure AD passthrough authentication is applicable to Azure AD logins. The identity of the Azure AD user is passed to the storage if a credential is not specified. Azure AD passthrough authentication is not available for the SQL users. |
| **Storage shared access signature (SAS) token authentication** | No | Yes, using DATABASE SCOPED CREDENTIAL with shared access signature token in EXTERNAL DATA SOURCE or instance-level CREDENTIAL with shared access signature. |
| **Storage Access Key authentication** | Yes, using DATABASE SCOPED CREDENTIAL in EXTERNAL DATA SOURCE | No, use SAS token instead of storage access key. |
| **Storage Managed Identity authentication** | Yes, using Managed Service Identity Credential | Yes, The query can access the storage using the workspace Managed Identity credential. |
| **Storage Application identity/Service principal (SPN) authentication** | Yes | Yes, you can create a credential with a service principal application ID that will be used to authenticate on the storage. |
| **Server roles** | No | Yes, sysadmin, public, and other server-roles are supported. |
| **SERVER SCOPED CREDENTIAL** | No | Yes, the server scoped credentials are used by the OPENROWSET function that do not uses explicit data source. |
| **Permissions - Server-level** | No | Yes, for example, CONNECT ANY DATABASE and SELECT ALL USER SECURABLES enable a user to read data from any databases. |
| **Database roles** | Yes | Yes, you can use db\_owner, db\_datareader and db\_ddladmin roles. |
| **DATABASE SCOPED CREDENTIAL** | Yes, used in external data sources. | Yes, database scoped credentials can be used in external data sources to define storage authentication method. |
| **Permissions - Database-level** | Yes | Yes, you can grant, deny, or revoke permissions on the database objects. |
| **Permissions - Schema-level** | Yes, including ability to GRANT, DENY, and REVOKE permissions to users/logins on the schema | Yes, you can specify schema-level permissions including ability to GRANT, DENY, and REVOKE permissions to users/logins on the schema. |
| **Permissions - Object-level** | Yes, including ability to GRANT, DENY, and REVOKE permissions to users | Yes, you can GRANT, DENY, and REVOKE permissions to users/logins on the system objects that are supported. |
| **Permissions - Column-level security** | Yes | Yes, column-level security is supported in serverless SQL pools. Column level encryption is also generally available, see Encrypt a column of data. |
| **Row-level security** | Yes | No, there is no built-in support for the row-level security. Use custom views as a workaround. |
| **Data masking** | Yes | No, built-in data masking is not supported in the serverless SQL pools. Use wrapper SQL views that explicitly mask some columns as a workaround. |
| **Built-in/system security & identity functions** | Some Transact-SQL security functions and operators: CURRENT\_USER, HAS\_DBACCESS, IS\_MEMBER, IS\_ROLEMEMBER, SESSION\_USER, SUSER\_NAME, SUSER\_SNAME, SYSTEM\_USER, USER, USER\_NAME, EXECUTE AS, OPEN/CLOSE MASTER KEY | Some Transact-SQL security functions and operators are supported: CURRENT\_USER, HAS\_DBACCESS, HAS\_PERMS\_BY\_NAME, IS\_MEMBER, IS\_ROLEMEMBER, IS\_SRVROLEMEMBER, SESSION\_USER, SESSION\_CONTEXT, SUSER\_NAME, SUSER\_SNAME, SYSTEM\_USER, USER, USER\_NAME, EXECUTE AS, and REVERT. Security functions cannot be used to query external data (store the result in variable that can be used in the query). |
| **Transparent Data Encryption (TDE)** | Yes | No, Transparent Data Encryption is not supported. |
| **Data Discovery & Classification** | Yes | No, Data Discovery & Classification is not supported. |
| **Vulnerability Assessment** | Yes | No, Vulnerability Assessment is not available. |
| **Advanced Threat Protection** | Yes | No, Advanced Threat Protection is not supported. |
| **Auditing** | Yes | Yes, auditing is supported in serverless SQL pools. |
| **Firewall rules** | Yes | Yes, the firewall rules can be set on the serverless SQL endpoint. |
| **Private endpoint** | Yes | Yes, the private endpoint can be set on the serverless SQL pool. |

Dedicated SQL pool and serverless SQL pool use standard Transact-SQL language to query data. For detailed differences, look at the Transact-SQL language reference.

**Tools**

You can use various tools to connect to Synapse SQL to query data.

|  | **Dedicated** | **Serverless** |
| --- | --- | --- |
| **Synapse Studio** | Yes, SQL scripts | Yes, SQL scripts can be used in Synapse Studio. Use SSMS or ADS instead of Synapse Studio if you are returning a large amount of data as a result. |
| **Power BI** | Yes | Yes, you can use Power BI to create reports on serverless SQL pool. Import mode is recommended for reporting. |
| **Azure Analysis Service** | Yes | Yes, you can load data in Azure Analysis Service using the serverless SQL pool. |
| **Azure Data Studio (ADS)** | Yes | Yes, you can use Azure Data Studio (version 1.18.0 or higher) to query a serverless SQL pool. SQL scripts and SQL notebooks are supported. |
| **SQL Server Management Studio (SSMS)** | Yes | Yes, you can use SQL Server Management Studio (version 18.5 or higher) to query a serverless SQL pool. SSMS shows only the objects that are available in the serverless SQL pools. |

**Note**

You can use SSMS to connect to serverless SQL pool and query. It is partially supported starting from version 18.5, you can use it to connect and query only.

Most of the applications use standard Transact-SQL language can query both dedicated and serverless consumption models of Synapse SQL.

**Data access**

Data that is analyzed can be stored on various storage types. The following table lists all available storage options:

|  | **Dedicated** | **Serverless** |
| --- | --- | --- |
| **Internal storage** | Yes | No, data is placed in Azure Data Lake or Cosmos DB analytical storage. |
| **Azure Data Lake v2** | Yes | Yes, you can use external tables and the OPENROWSET function to read data from ADLS. Learn here how to setup access control. |
| **Azure Blob Storage** | Yes | Yes, you can use external tables and the OPENROWSET function to read data from Azure Blob Storage. Learn here how to setup access control. |
| **Azure SQL/SQL Server (remote)** | No | No, serverless SQL pool cannot reference Azure SQL database. You can reference serverless SQL pools from Azure SQL using elastic queries or linked servers. |
| **Dataverse** | No, you can load CosmosDB data into a dedicated pool using Synapse Link in serverless SQL pool (via ADLS) or Spark. | Yes, you can read Dataverse tables using Synapse link. |
| **Azure Cosmos DB transactional storage** | No | No, you cannot access Cosmos DB containers to update data or read data from the Cosmos DB transactional storage. Use Spark pools to update the Cosmos DB transactional storage. |
| **Azure Cosmos DB analytical storage** | No, you can load CosmosDB data into a dedicated pool using Synapse Link in serverless SQL pool (via ADLS), ADF, Spark or some other load tool. | Yes, you can query Cosmos DB analytical storage using Synapse Link. |
| **Apache Spark tables (in workspace)** | No | Yes, serverless pool can read PARQUET and CSV tables using metadata synchronization. |
| **Apache Spark tables (remote)** | No | No, serverless pool can access only the PARQUET and CSV tables that are created in Apache Spark pools in the same Synapse workspace. However, you can manually create an external table that reference external Spark table location. |
| **Databricks tables (remote)** | No | No, serverless pool can access only the PARQUET and CSV tables that are created in Apache Spark pools in the same Synapse workspace. However, you can manually create an external table that reference Databricks table location. |

**Data formats**

Data that is analyzed can be stored in various storage formats. The following table lists all available data formats that can be analyzed:

|  | **Dedicated** | **Serverless** |
| --- | --- | --- |
| **Delimited** | Yes | Yes, you can query delimited files. |
| **CSV** | Yes (multi-character delimiters not supported) | Yes, you can query CSV files. For better performance use PARSER\_VERSION 2.0 that provides faster parsing. If you are appending rows to your CSV files, make sure that you query the files as appendable. |
| **Parquet** | Yes | Yes, you can query Parquet files, including the files with nested types. |
| **Hive ORC** | Yes | No, serverless SQL pools cannot read Hive ORC format. |
| **Hive RC** | Yes | No, serverless SQL pools cannot read Hive RC format. |
| **JSON** | Yes | Yes, you can query JSON files using delimited text format and the T-SQL JSON functions. |
| **Avro** | No | No, serverless SQL pools cannot read Avro format. |
| **Delta Lake** | No | Yes, you can query delta lake files, including the files with nested types. |
| **Common Data Model (CDM)** | No | No, serverless SQL pool cannot read data stored using Common Data Model. |