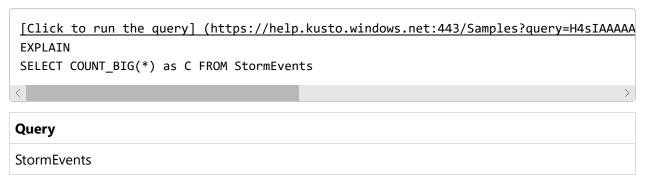
SQL to Kusto query translation

Kusto supports subset of SQL language. See the list of SQL known issues (../api/tds/sqlknownissues.html) for the full list of unsupported features.

Primary language to interact with Kusto is KQL (Kusto Query Language), and in order to make transition and learning experience easier, you can use Kusto service to translate SQL queries to KQL. This can be achieved by sending SQL query to Kusto services prefixing it with 'EXPLAIN' verb.

For example:



SQL to Kusto cheat sheet

The table below shows sample queries in SQL and thier KQL equivalients.

Category	SQL Query	Kusto Query
Select data from table	SELECT * FROM dependencies	dependencies
	SELECT name, resultCode FROM dependencies	<pre>dependencies project name, resultCode</pre>
	SELECT TOP 100 * FROM dependencies	dependencies take 100
Null evaluation	SELECT * FROM dependencies WHERE resultCode IS NOT NULL	<pre>dependencies where isnotnull(resultCode)</pre>
Comparison operatots (date)	<pre>SELECT * FROM dependencies WHERE timestamp > getdate()-1</pre>	<pre>dependencies where timestamp > ago(1d)</pre>
	SELECT * FROM dependencies WHERE timestamp BETWEEN AND	<pre>dependencies where timestamp > datetime(2016-10-01) and timestamp <= datetime(2016-11-01)</pre>

Category	SQL Query	Kusto Query
Comparison operators (string)	SELECT * FROM dependencies WHERE type = "Azure blob"	<pre>dependencies where type == "Azure blob"</pre>
	substring SELECT * FROM dependencies WHERE type like "%blob%"	<pre>// substring dependencies where type contains "blob"</pre>
	wildcard SELECT * FROM dependencies WHERE type like "Azure%"	<pre>// wildcard dependencies where type startswith "Azure" // or dependencies where type matches regex "^Azure.*"</pre>
Comparison (boolean)	SELECT * FROM dependencies WHERE !(success)	<pre>dependencies where success == "False"</pre>
Distinct	SELECT DISTINCT name, type FROM dependencies	dependencies summarize by name, type
Grouping, Aggregation	SELECT name, AVG(duration) FROM dependencies GROUP BY name	<pre>dependencies summarize avg(duration) by name</pre>
Column aliases, Extending	SELECT operationName as Name, AVG(duration) as AvgD FROM dependencies GROUP BY name	<pre>dependencies summarize AvgD = avg(duration) by operationName project Name = operationName, AvgD</pre>
Ordering	SELECT name, timestamp FROM dependencies ORDER BY timestamp ASC	<pre>dependencies project name, timestamp order by timestamp asc nulls last</pre>
Top n by measure	SELECT TOP 100 name, COUNT(*) as Count FROM dependencies GROUP BY name ORDER BY Count DESC	<pre>dependencies summarize Count = count() by name top 100 by Count desc</pre>
Union	SELECT * FROM dependencies UNION SELECT * FROM exceptions	union dependencies, exceptions

Category	SQL Query	Kusto Query
	SELECT * FROM dependencies WHERE timestamp > UNION SELECT * FROM exceptions WHERE timestamp >	<pre>dependencies where timestamp > ago(1d) union (exceptions where timestamp > ago(1d))</pre>
Join	<pre>SELECT * FROM dependencies LEFT OUTER JOIN exception ON dependencies.operation_Id = exceptions.operation_Id</pre>	<pre>dependencies join kind = leftouter (exceptions) on \$left.operation_Id == \$right.operation_Id</pre>
Nested queries	SELECT * FROM dependencies WHERE resultCode == (SELECT TOP 1 resultCode FROM dependencies WHERE resultId = 7 ORDER BY timestamp DESC)	<pre>dependencies where resultCode == toscalar(dependencies where resultId == 7 top 1 by timestamp desc project resultCode)</pre>
Having	SELECT COUNT(*) FROM dependencies GROUP BY name HAVING COUNT(*) > 3	<pre>dependencies summarize Count = count() by name where Count > 3</pre>