

SQL Advanced

Course Overview – 1-day course

A crucial skill for anyone working with data is the ability to access and analyse that data. Structured Query Language (SQL) enables users to access data from databases, combine data from multiple, related tables and to aggregate, sort, filter or limit the data retrieved as needed.

In this course, students will use more sophisticated techniques for querying databases and reshaping and manipulating result sets. Participants will learn to reshape data by pivoting and unpivoting. They will write more complex queries using subqueries, temporary tables and common table expressions. They will use recursive techniques to query hierarchical data and create date dimension tables. Students will build on their understanding of scalar user-defined functions and local variables to create table-valued functions and use table variables.

Students will also learn to simplify querying and increase their efficiency by using views and stored procedures.

In this course, students work with an Azure SQL database and write queries in Microsoft's Transact-SQL (T-SQL). Much of what is covered will be useful for users working with other database products.

Detailed Content

<p>Introduction SQL and T-SQL</p>	<p>Semi joins</p>	<p>Executing stored procedures with parameters</p>
<p>Reshaping data Use PIVOT to turn rows into columns Use UNPIVOT to rotate columns into rows</p>	<p>Variables Local variables Table variables Temporary tables vs CTEs vs table variables</p>	<p>Query Performance Improving query performance Basic principles View execution plans Interpret the execution plan Temporary tables, views, CTEs and table variables Functions Reusing code</p>
<p>Writing More Complex Queries Subqueries Local and global temporary tables Create, modifying and delete temporary tables Common table expressions Nested CTEs Recursive CTEs Query hierarchical data Create date dimension tables</p>	<p>User-Defined Functions Scalar user-defined functions Table-valued functions Multi-statement table-valued functions Schemas Use APPLY to join output from table-valued functions to a table</p>	
<p>More Complex Joins Cross joins Self joins Anti joins</p>	<p>Control of Flow IF...ELSE BEGIN...END</p>	
	<p>Simplifying Code Reuse Working with views Table-valued functions Working with stored procedures</p>	