

Exam 70-461: Querying Microsoft SQL Server 2012/2014 – Skills Measured

Audience Profile

This exam is intended for SQL Server database administrators, system engineers, and developers with two or more years of experience, who are seeking to validate their skills and knowledge in writing queries.

Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is not definitive or exhaustive.

NOTE: In most cases, exams do NOT cover preview features, and some features will only be added to an exam when they are GA (General Availability).

Create database objects (20–25%)

Create and alter tables using T-SQL syntax (simple statements)

- create tables without using the built in tools; ALTER; DROP; ALTER COLUMN; CREATE

Create and alter views (simple statements)

- create indexed views; create views without using the built in tools; CREATE, ALTER, DROP

Design views

- ensure code non regression by keeping consistent signature for procedure, views and function (interfaces); security implications

Create and modify constraints (simple statements)

- create constraints on tables; define constraints; unique constraints; default constraints; primary and foreign key constraints

Create and alter DML triggers

- inserted and deleted tables; nested triggers; types of triggers; update functions; handle multiple rows in a session; performance implications of triggers

Work with data (25–30%)

Query data by using SELECT statements

- use the ranking function to select top(X) rows for multiple categories in a single query; write and perform queries efficiently using the new (SQL 2005/8->) code items such as synonyms, and joins (except, intersect); implement logic which uses dynamic SQL and system metadata; write efficient, technically complex SQL queries, including all types of joins versus the use of derived tables; determine what code may or may not execute based on the tables provided; given a table with constraints, determine which statement set would load a table; use and understand different data access technologies; case versus isnull versus coalesce

Implement sub-queries

- identify problematic elements in query plans; pivot and unpivot; apply operator; cte statement; with statement

Implement data types

- use appropriate data; understand the uses and limitations of each data type; impact of GUID (newid, newsequentialid) on database performance, when to use what data type for columns

Implement aggregate queries

- new analytic functions; grouping sets; spatial aggregates; apply ranking functions

Query and manage XML data

- understand XML datatypes and their schemas and interop w/, limitations and restrictions; implement XML schemas and handling of XML data; XML data: how to handle it in SQL Server and when and when not to use it, including XML namespaces; import and export XML; XML indexing

Modify data (20–25%)

Create and alter stored procedures (simple statements)

- write a stored procedure to meet a given set of requirements; branching logic; create stored procedures and other programmatic objects; techniques for developing stored procedures; different types of storeproc result; create stored procedure for data access layer; program stored procedures, triggers, functions with T-SQL

Modify data by using INSERT, UPDATE, and DELETE statements

- given a set of code with defaults, constraints, and triggers, determine the output of a set of DDL; know which SQL statements are best to solve common requirements; use output statement

Combine datasets

- difference between UNION and UNION all; case versus isnull versus coalesce; modify data by using MERGE statements

Work with functions

- understand deterministic, non-deterministic functions; scalar and table values; apply built-in scalar functions; create and alter user-defined functions (UDFs)

Troubleshoot and optimize (25–30%)

Optimize queries

- understand statistics; read query plans; plan guides; DMVs; hints; statistics IO; dynamic vs. parameterized queries; describe the different join types (HASH, MERGE, LOOP) and describe the scenarios they would be used in

Manage transactions

- mark a transaction; understand begin tran, commit, and rollback; implicit vs explicit transactions; isolation levels; scope and type of locks; truncount

Evaluate the use of row-based operations vs. set-based operations

- when to use cursors; impact of scalar UDFs; combine multiple DML operations

Implement error handling

- implement try/catch/throw; use set based rather than row based logic; transaction management