

Exam 70-762: Developing SQL Databases – Skills Measured

Audience Profile

This exam is intended for database professionals who build and implement databases across organizations and who ensure high levels of data availability. Their responsibilities include creating database files, data types, and tables; planning, creating, and optimizing indexes; ensuring data integrity; implementing views, stored procedures, and functions; and managing transactions and locks.

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).

Design and implement database objects (25–30%)

Design and implement a relational database schema

- design tables and schemas based on business requirements, improve the design of tables by using normalization, write table create statements, determine the most efficient data types to use

Design and implement indexes

- design new indexes based on provided tables, queries, or plans; distinguish between indexed columns and included columns; implement clustered index columns by using best practices; recommend new indexes based on query plans

Design and implement views

- design a view structure to select data based on user or business requirements, identify the steps necessary to design an updateable view, implement partitioned views, implement indexed views

Implement columnstore indexes

- determine use cases that support the use of columnstore indexes, identify proper usage of clustered and non-clustered columnstore indexes, design standard non-clustered indexes in conjunction with clustered columnstore indexes, implement columnstore index maintenance

Implement programmability objects (20–25%)

Ensure data integrity with constraints

- define table and foreign key constraints to enforce business rules, write Transact-SQL statements to add constraints to tables, identify results of Data Manipulation Language (DML) statements given existing tables and constraints, identify proper usage of PRIMARY KEY constraints

Create stored procedures

- design stored procedure components and structure based on business requirements, implement input and output parameters, implement table-valued parameters, implement return codes, streamline existing stored procedure logic, implement error handling and transaction control logic within stored procedures

Create triggers and user-defined functions

- design trigger logic based on business requirements; determine when to use Data Manipulation Language (DML) triggers, Data Definition Language (DDL) triggers, or logon triggers; recognize results based on execution of AFTER or INSTEAD OF triggers; design scalar-valued and table-valued user-defined functions based on business requirements; identify differences between deterministic and non-deterministic functions

Manage database concurrency (25–30%)

Implement transactions

- identify DML statement results based on transaction behavior, recognize differences between and identify usage of explicit and implicit transactions, implement savepoints within transactions, determine the role of transactions in high-concurrency databases

Manage isolation levels

- identify differences between Read Uncommitted, Read Committed, Repeatable Read, Serializable, and Snapshot isolation levels; define results of concurrent queries based on isolation level; identify the resource and performance impact of given isolation levels

Optimize concurrency and locking behavior

- troubleshoot locking issues, identify lock escalation behaviors, capture and analyze deadlock graphs, identify ways to remediate deadlocks

Implement memory-optimized tables and native stored procedures

- define use cases for memory-optimized tables versus traditional disk-based tables, optimize performance of in-memory tables by changing durability settings, determine best case usage scenarios for natively compiled stored procedures, enable collection of execution statistics for natively compiled stored procedures

Optimize database objects and SQL infrastructure (20–25%)

Optimize statistics and indexes

- determine the accuracy of statistics and the associated impact to query plans and performance, design statistics maintenance tasks, use dynamic management objects to review current index usage and identify missing indexes, consolidate overlapping indexes

Analyze and troubleshoot query plans

- capture query plans using extended events and traces, identify poorly performing query plan operators, create efficient query plans using Query Store, compare estimated and actual query plans and related metadata, configure Azure SQL Database Performance Insight

Manage performance for database instances

- manage database workload in SQL Server; design and implement Elastic Scale for Azure SQL Database; select an appropriate service tier or edition; optimize database file and tempdb configuration; optimize memory configuration; monitor and diagnose scheduling and wait statistics using dynamic management objects; troubleshoot and analyze storage, IO, and cache issues; monitor Azure SQL Database query plans

Monitor and trace SQL Server baseline performance metrics

- monitor operating system and SQL Server performance metrics; compare baseline metrics to observed metrics while troubleshooting performance issues; identify differences between performance monitoring and logging tools, such as perfmon and dynamic management objects; monitor Azure SQL Database performance; determine best practice use cases for extended events; distinguish between Extended Events targets; compare the impact of Extended Events and SQL Trace; define differences between Extended Events Packages, Targets, Actions, and Sessions