

MCSA: SQL Server 2012/2014 – 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).

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

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

Exam 70-462: Administering Microsoft SQL Server 2012/2014 Databases

Install and configure (20–25%)

Plan installation

- evaluate installation requirements; design the installation of SQL Server and its components (drives, service accounts, etc.); plan scale-up vs. scale-out basics; plan for capacity, including if/when to shrink, grow, autogrow, and monitor growth; manage the

technologies that influence SQL architecture (for example, service broker, full text, scale out, etc.); design the storage for new databases (drives, filegroups, partitioning); design database infrastructure; configure a SQL Server standby database for reporting purposes; Windows-level security and service level security; Core mode installation; benchmark a server before using it in a production environment (SQLIO, Tests on SQL Instance); choose the right hardware

Install SQL Server and related services

- test connectivity; enable and disable features; install SQL Server database engine and SSIS (not SSRS and SSAS); configure an OS disk

Implement a migration strategy

- restore vs detach/attach; migrate security; migrate from a previous version; migrate to new hardware; migrate systems and data from other sources

Configure additional SQL Server components

- set up and configure all SQL Server components (Engine, AS, RS and SharePoint integration) in a complex and highly secure environment; configure full-text indexing; SSIS security; filestream; filetable

Manage SQL Server Agent

- create, maintain, and monitor jobs; administer jobs and alerts; automate (setup, maintenance, monitoring) across multiple databases and multiple instances; send to "Manage SQL Server Agent jobs"

Maintain instances and databases (15–20%)

Manage and configure databases

- design multiple file groups; database configuration and standardization: autoclose, autoshrink, recovery models; manage file space, including adding new filegroups and moving objects from one filegroup to another; implement and configure contained databases; data compression; configure TDE; partitioning; manage log file growth; DBCC

Configure SQL Server instances

- configure and standardize a database: autoclose, autoshrink, recovery models; install default and named instances; configure SQL to use only certain CPUs (affinity masks, etc.); configure server level settings; configure many databases/instance, many instances/server, virtualization; configure clustered instances including MSDTC; memory

allocation; database mail; configure SQL Server engine: memory, filloffactor, sp_configure, default options

Implement a SQL Server clustered instance

- install a cluster; manage multiple instances on a cluster; set up subnet clustering; recover from a failed cluster node

Manage SQL Server instances

- install an instance; manage interaction of instances; SQL patch management; install additional instances; manage resource utilization by using Resource Governor; cycle error logs

Optimize and troubleshoot (15–20%)

Identify and resolve concurrency problems

- examine deadlocking issues using the SQL server logs using trace flags; design reporting database infrastructure (replicated databases); monitor via DMV or other MS product; diagnose blocking, live locking and deadlocking; diagnose waits; performance detection with built in DMVs; know what affects performance; locate and if necessary kill processes that are blocking or claiming all resources

Collect and analyze troubleshooting data

- monitor using Profiler; collect performance data by using System Monitor; collect trace data by using SQL Server Profiler; identify transactional replication problems; identify and troubleshoot data access problems; gather performance metrics; identify potential problems before they cause service interruptions; identify performance problems; use XEvents and DMVs; create alerts on critical server condition; monitor data and server access by creating audit and other controls; identify IO vs. memory vs. CPU bottlenecks; use the Data Collector tool

Audit SQL Server instances

- implement a security strategy for auditing and controlling the instance; configure an audit; configure server audits; track who modified an object; monitor elevated privileges as well as unsolicited attempts to connect; policy-based management

Manage data (20–25%)

Configure and maintain a back-up strategy

- manage different backup models, including point-in-time recovery; protect customer data even if backup media is lost; perform backup/restore based on proper strategies including backup redundancy; recover from a corrupted drive; manage a multi-TB database; implement and test a database implementation and a backup strategy (multiple files for user database and tempdb, spreading database files, backup/restore); back up a SQL Server environment; back up system databases

Restore databases

- restore a database secured with TDE; recover data from a damaged DB (several errors in DBCC checkdb); restore to a point in time; file group restore; page level restore

Implement and maintain indexes

- inspect physical characteristics of indexes and perform index maintenance; identify fragmented indexes; identify unused indexes; implement indexes; defrag/rebuild indexes; set up a maintenance strategy for indexes and statistics; optimize indexes (full, filter index); statistics (full, filter) force or fix queue; when to rebuild vs. reorg and index; full text indexes; column store indexes

Import and export data

- transfer data; bulk copy; bulk insert

Implement security (15–20%)

Manage logins and server roles

- configure server security; secure the SQL Server using Windows Account / SQL Server accounts, server roles; create log in accounts; manage access to the server, SQL Server instance, and databases; create and maintain user-defined server roles; manage certificate logins

Manage database security

- configure database security; database level, permissions; protect objects from being modified; auditing; encryption

Manage users and database roles

- create access to server / database with least privilege; manage security roles for users and administrators; create database user accounts; contained login

Troubleshoot security

- manage certificates and keys; endpoints

Implement high availability (5–10%)

Implement AlwaysOn

- implement AlwaysOn availability groups; implement AlwaysOn failover clustering

Implement replication

- troubleshoot replication problems; identify appropriate replication strategy

Exam 70-463: Implementing a Data Warehouse with Microsoft SQL Server 2012/2014

Design and implement a data warehouse (10–15%)

Design and implement dimensions

- design shared/conformed dimensions; determine if you need support for slowly changing dimensions; determine attributes; design hierarchies; determine whether you need star or snowflake schema; determine the granularity of relationship with fact tables; determine the need for auditing or lineage; determine keys (business transactional or your own data warehouse/surrogate keys); implement dimensions; implement data lineage of a dimension table

Design and implement fact tables

- design a data warehouse that supports many to many relationships; appropriately index a fact table; using columnstore indexes; partitioning; additive measures; semi additive measures; non additive measures; implement fact tables; determining the loading method for the fact tables; implement data lineage of a fact table; design summary aggregation tables

Extract and transform data (20–25%)

Define connection managers

- plan the configuration of connection managers; package level or project level connection manager; define a connection string; parameterization of connection strings

Design data flow

- define data sources and destinations; distinguish blocking and non-blocking transformations; use different methods to pull out changed data from data sources; determine appropriate data flow components; determine the need for supporting Slowly Changing Dimensions (SCD); determine whether to use SQL Joins or SSIS lookup or merge join transformations; batch processing versus row by row processing; determine the appropriate transform to use for a specific task; determine the need and method for identity mapping and deduplicating; fuzzy lookup, fuzzy grouping and Data Quality Services (DQS) transformation; determine the need for custom data sources, destinations, and transforms; determine what to do with erroneous rows; determine auditing needs; trusted/authoritative data sources, including warehouse metadata; extracting only changed rows

Implement data flow

- debug data flow; use the appropriate data flow components; SQL / SSIS data transformation; create SSIS packages that support slowly changing dimensions; use the lookup task in SSIS; map identities using SSIS fuzzy lookup (advanced); specify a data source and destination; use data flows; different categories of transformations; read, transform and load data; understand which transforms to use to accomplish a specific business task; data correction transformation; performance tune an SSIS dataflow; optimize Integration Services packages for speed of execution; maintain data integrity, including good data flow

Manage SSIS package execution

- schedule package execution by using SQL Server Agent; execute packages by using DTEXEC; execute packages by using SQL Server Management Studio; implement package execution; plan and design package execution strategy; use PowerShell to execute script; monitor the execution using Management Studio; use DTEXECUI; ETL restartability

Implement script tasks in SSIS

- determine if it is appropriate to use a script task; extending the capability of a control flow; perform a custom action as needed (not on every row) during a control flow

Load data (25–30%)

Design control flow

- determine control flow; determine containers and tasks needed; determine precedence constraints; design an SSIS package strategy with rollback, staging and transaction control; decide between one package or multiple packages; determine event handlers; determine variables; determine parameters on package and project level; determine connection managers and whether they are package or project level; determine the need

for custom tasks; determine how much information you need to log from a package; determine the need for checkpoints; determine security needs

Implement package logic by using SSIS variables and parameters

- user variables; variable scope, data type; implement parameterization of properties using variables; using variables in precedence constraints; referring to SSIS system variables; design dynamic SSIS packages; package configurations (file or SQL tables); expressions; package and project parameters; project level connection managers; variables; implement dynamic package behavior; configure packages in SSIS for different environments, package configurations (xmlconfiguration file, SQLServer table, registry entry; parent package variables, environment variable); parameters (package and project level); project connection managers; property expressions (use expressions for connection managers)

Implement control flow

- checkpoints; debug control flow; implement the appropriate control flow task to solve a problem; data profiling; use sequence containers and loop containers; manage transactions in SSIS packages; managing parallelism; using precedence constraint to control task execution sequence; creating package templates; using the execute package task

Implement data load options

- implement a full and incremental data load strategy; plan for an incremental update of the relational Data Mart; plan for loads into indexed tables; configure appropriate bulk load options; select an appropriate load technique (SSIS Destination versus T-SQL) and load partitioned tables

Implement script components in SSIS

- create an SSIS package that handles SCD Type 2 changes without using the SCD component; work with script component in SSIS; deciding when it is appropriate to use a script component versus a built in; source, transformation, destination component; use cases: web service source and destination, getting the error message

Configure and deploy SSIS solutions (20–25%)

Troubleshoot data integration issues

- performance issues; connectivity issues; execution of a task or transformation failed; logic issues; demonstrate awareness of the new SSIS logging infrastructure; troubleshoot a failed package execution to determine the root cause of failure; troubleshoot SSIS

package failure from an invalid datatype; implement break points; data viewers; profile data with different tools; batch cleanup

Install and maintain SSIS components

- software installation (IS, management tools); development box and server; install specifics for remote package execution; planning for installation (32- versus 64-bit); upgrade; provisioning the accounts; creating the catalog

Implement auditing, logging, and event handling

- audit package execution by using system variables; propagate events; use log providers; log an SSIS execution; create alerting and notification mechanisms; use Event Handlers in SSIS to track ETL events and errors; implement custom logging

Deploy SSIS solutions

- create and configure an SSIS catalog; deploy SSIS packages by using the deployment utility; deploy SSIS packages to SQL or file system locations; validate deployed packages; deploy packages on multiple servers; how to install custom components and tasks; deploy SSIS packages by using DTUTIL

Configure SSIS security settings

- SSIS catalog database roles; package protection levels; secure Integration Services packages that are deployed at the file system; secure Integration Services parameters, configuration

Build data quality solutions (15–20%)

Install and maintain data quality services

- installation prerequisites; .msi package; adding users to the DQ roles; identity analysis, including data governance

Implement master data management solutions

- install Master Data Services (MDS); implement MDS; create models, entities, hierarchies, collections, attributes; define security roles; import/export; subscriptions

Create a data quality project to clean data

- profile Online Transaction Processing (OLTP) and other source systems; data quality knowledge base management; create data quality project; use data quality client;

improve data quality; identity mapping and deduplicating; handle history and data quality; manage data quality/cleansing