70-459:
Transition Your MCITP: Database Administrator 2008 or MCITP: Database Developer 2008 to MCSE: Data Platform

The following tables show where changes to exam 70-459 have been made to include updates that relate to SQL Server 2014 tasks. These changes are effective as of May 15, 2014.

1. Implement database objects

Tasks currently measured	Tasks Added/Changed post *May 2014*
Create and alter tables (complex statements) Develop an optimal strategy for using temporary objects (table variables and temporary tables); how not to rely on triggers solely as a means to manage a table; data version control and management; create tables without using the built-in tools; understand the difference between @Table and #table	**Added sub-tasks:** • create calculated columns • implement partitioned tables, schemas, and functions • implement column collation • implement in-memory OLTP
Design, implement, and troubleshoot security Grant, deny, revoke; unable to connect; execute as; certificates; loginless user; database roles and permissions; contained users; change permission chains	**Added sub-tasks:** • implement cross db ownership chaining • implement schema security • implement server roles • review effective permissions • troubleshoot and repair orphaned users
Create and modify constraints (complex statements) Create constraints on tables; define constraints; performance implications	**Added sub-tasks:** • implement cascading deletes • configure constraints for bulk inserts

2. Implement programming objects

Tasks Currently Measured	Tasks Added/Changed post *May 2014*
Design and implement stored procedures Create stored procedures and other programmatic objects; techniques for developing stored procedures; different types of stored procedure results; create stored procedure for data access layer; analyze and rewrite procedures and processes; program stored procedures, with T-SQL and CLR#; use table valued parameters; encryption	**Added sub-tasks:** • implement error handling, including TRY…CATCH • configure appropriate connection settings • design appropriate query paging, including OFFSET and FETCH
Design T-SQL table-valued and scalar functions Ensure code non regression by keeping consistent signature for procedure, views and function (interfaces); turn scripts that use cursors and loops into a SET based operation	**Revised task – new full definition:** • modify scripts that use cursors and loops into a SET-based operation • design deterministic and non-deterministic functions
Create and alter views (complex statements) Set up and configure partitioned tables and partitioned views; design a best practice for using views and stored procedures and remove the direct usage of tables	**Revised task – new full definition:** • set up and configure partitioned tables and partitioned views • create indexed views

3. Design database objects

Tasks Currently Measured	Tasks Added/Changed post *May 2014*
Design tables Data design patterns; develop normalized and de-normalized SQL tables; understand the difference between physical tables, temp tables, temp table variables, and common table expressions; design transactions; design views; describe advantages / disadvantages of using a GUID as a clustered index; understand performance implications of # versus @ temp tables and how to decide which to use, when, and why; use of set-based rather than row-based logic; encryption (other than TDE); table partitioning; filestream and filetable	**Added sub-task:** • design tables for In-Memory OLTP
Create and alter indexes Create indexes and data structures; create filtered indexes; create an indexing strategy; design and optimize indexes; design indexes and statistics; assess which indexes on a table are likely to be used given different search arguments (SARG); column store indexes; semantic indexes	**Added sub-task:** • create spatial indexes
Design data integrity Design table data integrity policy (checks, private key/foreign key, uniqueness, XML schema); select a primary key; data usage patterns	**Added sub-task:** • design a table data integrity policy, including nullability

4. Optimize and troubleshoot queries

Tasks Currently Measured	Tasks Added/Changed post *May 2014*
Optimize and tune queries Tune a badly performing query; identify long running queries; review and optimize code; analyze execution plans to optimize queries; tune a query that is poorly written; tune queries using execution plans and database tuning advisor (DTA); design advanced queries: pivots, utilizing common table expressions (CTE), design the database layout and optimize queries (for speed and/or data size); understand different data types; basic knowledge of query hints; tune query workloads, using realistic data sets not being production data sets; demonstrate use of recursive CTE; full text search; control execution plans	**Added sub-tasks:** • implement semantic search • implement plan guides
Troubleshoot and resolve performance problems Interpret performance monitor data; impact of recovery modal on database size, and recovery; how to clean up if .MDF and .LDF files get too large; identify and fix transactional replication problems; detect and resolve server hung, failure; identify and troubleshoot data access problems	**Added sub-tasks:** • integrate performance monitor data with SQL Traces • manage tempdb contention and auto growth • implement Resource Governor • monitor and resolve In-Memory OLTP issues, including merge and garbage collection
Collect performance and system information Use Data Management Views to determine performance issues; from system metadata; gather trace information by using the SQL Server Profiler; develop monitoring strategy for production database; run a Profiler trace and analyze the results; run Profiler for troubleshooting application; collect output from the Database Engine Tuning Advisor; extended events	**Revised task – new full definition:** • monitor performance using Dynamic Management Views • collect output from the Database Engine Tuning Advisor; • design Extended Events Sessions; • review and interpret Extended Event logs; • optimize Extended Event session settings; • use Activity Monitor to minimize server impact and determine IO bottlenecks • monitor In-Memory OLTP resources

5. Design a Database Structure

Tasks currently measured	Tasks Added/Changed post *May 2014*
Design for business requirements Business to data translations; identify which SQL Server components to use to support business requirements; design a normalization area; de-normalize technically (vs. by remodeling) by using SQL Server features (materialization via indexed views, etc.)	**Revised sub-task:** • de-normalize a database by using SQL Server features, including materialization using indexed views, distributed partitioned views, filtered and non-key column indexes, and snapshots
Design physical database and object placement Identify bad database architectural decisions; filestream and filetable; logical vs. physical design; file groups	**Revised task – new full definition:** • design a physical database, including file placement, FILESTREAM, FILETABLE, file groups, and RAID • configure system database settings
Design SQL Server instances Spec out hardware for new instances; design an instance; design SQL to use only certain CPUs (affinity masks, etc.); design clustered instances including Microsoft Distributed Transaction Control (MSDTC); memory allocation	**Added sub-tasks:** • design installation strategies, including sysprep, slipstream, and SMB file server • define cross db ownership chaining

6. Design databases and database objects

Tasks Currently Measured	Tasks Added/Changed post *May 2014*
Design a database model Design a logical schema; design a normalized database; design data access and data layer architecture; understand the relational model; design a normalized data model; design a database schema; create/maintain a schema upgrade and downgrade script which include the most optimal schema deployment and data migration; review common modeling practices: Entity-Attribute-Value (EAV), generalization/specialization, star-schema, etc.; optimize the design for normalization to the right level for the application looking forward to possible scenarios in the future; design security architecture; relational database design; design/modify database schemas; design appropriately normalized and data typed table schemas to meet business requirement; design a strategy to use linked servers, security, providers, distributed transactions; understand impact of collation, ANSI NULLS, QUOTED IDENTIFIER; interpret a database design to match a set of statements that describe the design	**Revised task – new full definition:** • design a logical schema • design a data access and data layer architecture • design a database schema • design a security architecture • design a cross-server instance database model, including linked servers, security, providers, distributed transactions, distributed partitioned views, and Service Broker
Design tables Data design patterns; develop normalized and de-normalized SQL tables; understand the difference between physical tables, temp tables, temp table variables and common table expressions; design transactions; design views; describe advantages/disadvantages of using a GUID as a clustered index; understand performance implications of # vs. @ temp tables and how to decide which to use, when and why; how to use table valued parameters to sps; use of set-based rather than row-based logic; filestream and filetable; semantic engine; sequences; row/page compression; data type selection	**Revised task – new full definition:** • design tables appropriately, including physical tables, temp tables, temp table variables, common table expressions, columnstore indexes, user defined table types, FILESTREAM, FILETABLE, and In-Memory OLTP • design views and table valued functions • design a compression strategy, including row and page • select an appropriate data type • design computed columns
Design T-SQL stored procedures Write a stored procedure to meet a given set of requirements; design a best practice for using views and stored procedures and remove the direct usage of tables	**Revised task – new full definition:** • create stored procedures • design a data access strategy using stored procedures • design appropriate stored procedure parameters, including input, output, and Table Valued • design error handling

	• design an In-Memory OLTP strategy for stored procedures

7. Design database security

Tasks Currently Measured	Tasks Added/Changed post *May 2014*
Design an application strategy to support security Design security; implement schemas and schema security; design maintenance (SQL logins vs. integrated authentication, permissions, mirroring issues, etc.); use appropriate mechanisms to enforce security roles, signed stored procedures, etc.; encryption; contained logins	**Added sub-task:** • design security, including EXECUTE AS and credentials
Design instance-level security configurations Implement separation of duties using different login roles; design/implement a data safety strategy that meets the requirements of the installation; choosing authentication type, logon triggers, regulatory requirements; transparent data encryption; Data Description Language (DDL) triggers	**Added sub-tasks:** • choose an authentication type, including certificates • implement data encryption, including database master key and configuration • define a secure service account

8. Design a troubleshooting and optimization solution

Tasks Currently Measured	Tasks Added/Changed post *May 2014*
Troubleshoot and resolve concurrency issues 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	**Added sub-tasks:** • use Extended Events • implement query hints to increase concurrency
Design a monitoring solution at the instance level Design auditing strategies including XE, Profiler, Perfmon, and DMV usage; set up file and table growth monitoring; collect performance indicators and counters; content management systems; policies	**Added sub-tasks:** • design auditing strategies, including Event traces, SQL Audit, event-based maintenance • create jobs to monitor server health • audit using Windows Logs