|  |  |
| --- | --- |
|  |  |

70-465:

Designing Database Solutions for Microsoft SQL Server

The following tables show where changes to Exam 70-465 have been made to include updates that relate to database development and management-related tasks. These changes are effective as of February 18, 2016.

1. **Design a database structure (no change: 20–30%)**

|  |  |
| --- | --- |
| **Tasks currently measured** | **Tasks to be added/removed/changed in February 2016** |
| **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 (versus by remodeling) by using SQL Server features (materialization via indexed views and more) | Revised subtasks:  translate business needs to data structures; 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**  Design a physical database, including file placement, FILESTREAM, FILETABLE, file groups, and RAID; configure system database settings | No change |
| **Design a table and index partitioning strategy**  Develop the optimal strategy for indexing, archive using partitions and tables, design columnstore indexes, design XML indexes | No change |
| **Design a migration, consolidation, and upgrade strategy**  Upgrade with minimal downtime; design a cross-cluster migration; plan a database deployment, including Windows PowerShell, Server Core, and contained databases; migrate to SQL Database; migrate query plans; design a migration strategy using Distributed Replay Controller; design a SQL Server virtualization strategy | Removed subtask:  migrate to SQL Database |
| **Design SQL Server instances**  Identify hardware for new instances; design CPU affinity; design clustered instances using Microsoft Distributed Transaction Control (MSDTC); define instance memory allocation; design installation strategies, including sysprep, slipstream, and SMB file server; define cross db ownership chaining | No change |
| **Design backup and recovery**  Design a backup strategy based on business needs, including differential, file, log, striped, and Microsoft Azure Blob Storage Service; design a database snapshot strategy; design appropriate recovery models; design a system database backup strategy; recover Tail-Log backups | Removed subtask:  Microsoft Azure Blob Storage Service |

1. **Design databases and database objects (no change: 30-35%)**

|  |  |
| --- | --- |
| **Tasks currently measured** | **Tasks to be added/removed/changed in February 2016** |
| **Design a database model**  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 | No change |
| **Design tables**  Design tables appropriately, including physical tables, temp tables, temp table variables, common table expressions, commonstore 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 | Revised subtask:  columnstore indexes |
| **Design for concurrency**  Develop a strategy to maximize concurrency; define a locking and concurrency strategy; design a transaction isolation strategy, including server database and session; design triggers for concurrency | No change |
| **Design T-SQL stored procedures**  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 | No change |
| **Design a management automation strategy**  Create a data archiving solution; design automation and auditing, including jobs, alerts, operators, SSIS, CDC, auditing, DDL triggers, and Windows PowerShell; automate across multiple databases and instances; design data batch processing: design a database load test; deploy to different environments, including development, staging, and production | No change |
| **Design for transactions**  Manage transactions, including time, savepoint, and mark; design for implicit and explicit transactions; ensure data integrity by using transactions; design error handling for transactions, including TRY, CATCH, and THROW | No change |

1. **Design database security (no change: 15-20%)**

|  |  |
| --- | --- |
| **Tasks currently measured** | **Tasks to be added/removed/changed in February 2016** |
| **Design an application strategy to support security**  Design security, including security roles, signed stored procedures, encryption, contained logins, EXECUTE AS, and credentials; implement schemas and schema security; design security maintenance, including SQL logins, integrated authentication, permissions, and mirroring | No change |
| **Design database, schema, and object security permissions**  Design a database schema that meets security requirements, schema ownership, ownership chaining, cross database chaining | No change |
| **Design instance-level security configurations**  Implement separation of duties using different login roles; choose an authentication type, including logon triggers, regulatory requirements, and certificates; implement data encryption, including database master key and configuration; implement Data Description Language (DDL) triggers; define a secure service account | No change |

1. **Design a troubleshooting and optimization solution (no change: 20-25%)**

|  |  |
| --- | --- |
| **Tasks currently measured** | **Tasks to be added/removed/changed in February 2016** |
| **Design a maintenance strategy for database servers**  Design maintenance plans; design index maintenance, including rebuild, defragmentation, statistics, online rebuilds, offline rebuilds, and thresholds; maintain physical and logical consistency (DBCC); manage database files, including LDF, MDF, In-Memory OLTP, and garbage collection; define a retention policy | No change |
| **Troubleshoot and resolve concurrency issues**  Examine deadlocking issues using SQL Server logs and trace flags; design a reporting database infrastructure, including replicated databases; monitor concurrency, including Dynamic Management Views (DMV); diagnose blocking, including live locking and deadlocking; diagnose waits; use Extended Events; implement query hints to increase concurrency | No change |
| **Design and implement a high availability solution**  Configure failover clustering, including multi-subnet; design readable mirrors; create a highly available configuration with low recovery time; design and ensure uptime requirements, including monitoring and patching; design and implement a database mirroring architecture; design and implement a replication architecture; implement a mirroring solution, including AlwaysOn and Availability Groups; design geographical fault-tolerance using Microsoft Azure SQL Database | Removed subtasks:  design and implement a database mirroring architecture; design geographical fault-tolerance using Microsoft Azure SQL Database  Revised subtask:  Implement AlwaysOn Availability Groups and AlwaysOn failover clusters |
| **Design a solution to monitor performance and concurrency**  Identify performance monitor counters; monitor for performance and bottlenecks, including Wait Stats; design a query monitoring and review strategy; monitor for missing statistics | No change |
| **Design a monitoring solution at the instance level**  Design auditing strategies, including Extended Events, Event traces, SQL Audit, Profiler-scheduled or event-based maintenance, Performance Monitor, and DMV usage; set up file and table growth monitoring; collect performance indicators and counters; create jobs to monitor server health; audit using Windows Logs | No change |