Exam 70-465: Designing Database Solutions for Microsoft SQL Server – Skills Measured

Audience Profile

This exam is intended for database professionals who design and build database solutions in the organization. They are responsible for the creation of plans and designs for database structure, storage, objects, and servers. They create the plan for the environment in which the database solution runs.

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 a database structure (25–30%)

Design for business requirements

- 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

Design a table and index partitioning strategy

- develop the optimal strategy for indexing, archive using partitions and tables, design columnstore indexes, design XML indexes

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 query plans; design a migration strategy using Distributed Replay Controller; design a SQL Server virtualization strategy

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

Design backup and recovery

- design a backup strategy based on business needs, including differential, file, log, and striped; design a database snapshot strategy; design appropriate recovery models; design a system database backup strategy; recover Tail-Log backups

Design databases and database objects (30–35%)

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

Design tables

- 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 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

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

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

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

Design database security (10–15%)

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

Design database, schema, and object security permissions

- design a database schema that meets security requirements, schema ownership, ownership chaining, cross database chaining

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

Design a troubleshooting and optimization solution (25–30%)

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

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

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 replication architecture; 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

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