

## 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 SQL Server 2014 tasks. These changes are effective as of April 24, 2014.

### 1. Design a Database Structure (20-30%)

Tasks currently measured	Tasks Added/Changed post <i>April 2014</i>
<p><b>Design for business requirements</b> 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.)</p>	<p><b>Revised sub-task:</b></p> <ul style="list-style-type: none"> <li>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</li> </ul>
<p><b>Design physical database and object placement</b> Identify bad database architectural decisions; filestream and filetable; logical vs. physical design; file groups</p>	<p><b>Revised task – new full definition:</b></p> <ul style="list-style-type: none"> <li>design a physical database, including file placement, FILESTREAM, FILETABLE, file groups, and RAID</li> <li>configure system database settings</li> </ul>
<p><b>Design a table and index partitioning strategy</b> Develop optimal strategy for indexing; data distribution; archiving</p>	<p><b>Added sub-tasks:</b></p> <ul style="list-style-type: none"> <li>design columnstore indexes</li> <li>design XML indexes</li> </ul>
<p><b>Design a migration, consolidation, and upgrade strategy</b> Upgrade with minimal downtime; database deployments; multiple databases in same solution; contained databases</p>	<p><b>Revised task – new full definition:</b></p> <ul style="list-style-type: none"> <li>upgrade with minimal downtime</li> <li>design a cross-cluster migration</li> <li>plan a database deployment, including Windows PowerShell, Server Core, and contained databases</li> <li>migrate to SQL Azure</li> <li>migrate query plans</li> <li>design a migration strategy using Distributed Replay Controller</li> <li>design a SQL Server virtualization strategy</li> </ul>
<p><b>Design SQL Server instances</b> Spec out hardware for new instances; design an instance; design SQL to use only certain CPUs (affinity masks, etc.); design clustered instances</p>	<p><b>Added sub-tasks:</b></p> <ul style="list-style-type: none"> <li>design installation strategies, including sysprep, slipstream, and SMB file server</li> <li>define cross db ownership chaining</li> </ul>

including Microsoft Distributed Transaction Control (MSDTC); memory allocation	
<b>Design backup and recovery</b> Database snapshots; recovery models; transaction log backups; when to use differentials; file backup; striped backups	<b>Revised task – new full definition:</b> <ul style="list-style-type: none"><li>• design a backup strategy based on business needs, including differential, file, log, striped, and Windows Azure Blob Storage Service</li><li>• design a database snapshot strategy</li><li>• design appropriate recovery models</li><li>• design a system database backup strategy</li><li>• recover Tail-Log backups</li></ul>

## 2. Design databases and database objects (30-35%)

Tasks Currently Measured	Tasks Added/Changed post <i>April 2014</i>
<p><b>Design a database model</b></p> <p>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</p>	<p><b>Revised task – new full definition:</b></p> <ul style="list-style-type: none"> <li>• design a logical schema</li> <li>• design a data access and data layer architecture</li> <li>• design a database schema</li> <li>• design a security architecture</li> <li>• design a cross-server instance database model, including linked servers, security, providers, distributed transactions, distributed partitioned views, and Service Broker</li> </ul>
<p><b>Design tables</b></p> <p>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</p>	<p><b>Revised task – new full definition:</b></p> <ul style="list-style-type: none"> <li>• 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</li> <li>• design views and table valued functions</li> <li>• design a compression strategy, including row and page</li> <li>• select an appropriate data type</li> <li>• design computed columns</li> </ul>
<p><b>Design for concurrency</b></p> <p>Develop a strategy to minimize concurrency; handle concurrency to minimize locking and eliminate as much blocking as possible, and to avoid deadlocks; manage the transactions to limit the time to hold lock and have fast transactions (maximize concurrency); define</p>	<p><b>Revised task – new full definition:</b></p> <ul style="list-style-type: none"> <li>• develop a strategy to maximize concurrency</li> <li>• define a locking and concurrency strategy</li> <li>• design a transaction isolation strategy, including server database and session</li> <li>• design triggers for concurrency</li> </ul>

locking and concurrency strategy; impact of read committed snapshot/snapshot isolation; understand what it solves and what it costs	
<p><b>Design T-SQL stored procedures</b></p> <p>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</p>	<p><b>Revised task – new full definition:</b></p> <ul style="list-style-type: none"> <li>• create stored procedures</li> <li>• design a data access strategy using stored procedures</li> <li>• design appropriate stored procedure parameters, including input, output, and Table Valued</li> <li>• design error handling</li> <li>• design an In-Memory OLTP strategy for stored procedures</li> </ul>
<p><b>Design a management automation strategy</b></p> <p>Create a data archiving solution; create jobs to ensure good server health as DBCC Checkdb, statistics updates; improve database maintenance (DB index, backup, etc.) with custom script that executes some task only on when some values are overpassed (defragment/rebuild index); design automation and auditing (jobs, alerts, operators, SSIS, CDC, auditing, DDL triggers); automate (setup, maintenance, monitoring) across multiple databases and multiple instances; data flow and batch processing; testing load on database plus different stages</p>	<p><b>Added sub-tasks:</b></p> <ul style="list-style-type: none"> <li>• design automation and auditing, including Windows PowerShell</li> <li>• deploy to different environments, including development, staging, and production</li> </ul>
<p><b>Design for implicit and explicit transactions</b></p> <p>Manage transactions; use transactions in code; ensure data integrity by using transactions; trycatch; commit; throw</p>	<p><b>Added sub-task:</b></p> <ul style="list-style-type: none"> <li>• design for implicit and explicit transactions</li> </ul>

### 3. Design database security (15-20%)

Tasks Currently Measured	Tasks Added/Changed post <i>April 2014</i>
<p><b>Design an application strategy to support security</b></p> <p>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</p>	<p><b>Added sub-task:</b></p> <ul style="list-style-type: none"> <li>• design security, including EXECUTE AS and credentials</li> </ul>
<p><b>Design database, schema, and object security permissions</b></p> <p>Design a database schema that meets security requirements; schema ownership; ownership chaining; cross database chaining</p>	<p><b>Added sub-task:</b></p> <ul style="list-style-type: none"> <li>• design queries and stored procedures that use multiple schemas</li> </ul>
<p><b>Design instance-level security configurations</b></p> <p>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</p>	<p><b>Added sub-tasks:</b></p> <ul style="list-style-type: none"> <li>• choose an authentication type, including certificates</li> <li>• implement data encryption, including database master key and configuration</li> <li>• define a secure service account</li> </ul>

#### 4. Design a troubleshooting and optimization solution (24%)

Tasks Currently Measured	Tasks Added/Changed post <i>April 2014</i>
<p><b>Design a maintenance strategy for database servers</b></p> <p>Online rebuilds vs. offline rebuilds; maintenance plans; rebuild indexes; defrag indexes; check DB; statistics; grow the database; manage backups and history; retention policy</p>	<p><b>Revised task – new full definition:</b></p> <ul style="list-style-type: none"> <li>• design maintenance plans</li> <li>• design index maintenance, including rebuild, defragmentation, statistics, online rebuilds, offline rebuilds, and thresholds</li> <li>• maintain physical and logical consistency (DBCC)</li> <li>• manage database files, including LDF, MDF, In-Memory OLTP, and garbage collection;</li> <li>• define a retention policy</li> </ul>
<p><b>Troubleshoot and resolve concurrency issues</b></p> <p>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</p>	<p><b>Added sub-tasks:</b></p> <ul style="list-style-type: none"> <li>• use Extended Events</li> <li>• implement query hints to increase concurrency</li> </ul>
<p><b>Design and implement a high availability solution</b></p> <p>Understand the traditional failover clustering solution; configure failover clustering; design readable mirrors; create a highly available configuration with low RTO; design and ensure uptime to relevant TOS/RLAs (includes monitoring, patching, etc.); design and implement a replication architecture; implement a mirroring solution using HADRON</p>	<p><b>Removed sub-task:</b></p> <ul style="list-style-type: none"> <li>• implement a mirroring solution using HADRON</li> </ul> <p><b>Added sub-tasks:</b></p> <ul style="list-style-type: none"> <li>• design and implement a database mirroring architecture</li> <li>• implement a mirroring solution, including AlwaysOn and Availability Groups</li> <li>• design geographical fault-tolerance using Windows Azure SQL Database</li> </ul>
<p><b>Design a solution to monitor performance and concurrency</b></p> <p>Identify performance monitor counters to monitor; monitor for performance and bottlenecks, including Wait Stats; design a top consumer queries monitoring and review strategy; monitor for missing statistics and create them when needed</p>	<p>No changes</p>
<p><b>Design a monitoring solution at the instance level</b></p> <p>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</p>	<p><b>Added sub-tasks:</b></p> <ul style="list-style-type: none"> <li>• design auditing strategies, including Event traces, SQL Audit, event-based maintenance</li> <li>• create jobs to monitor server health</li> <li>• audit using Windows Logs</li> </ul>