

NUTECH COMPUTER TRAINING INSTITUTE

1682 E. GUDE DRIVE #102, ROCKVILLE, MD 20850

WEB: www.NUTECHTRAINING.COM TEL: 301-610-9300

MCTS SQL Server 2005 Database Administrator – Course Outlines

Exam 70–443: TS: Microsoft SQL Server 2005 – Designing a Database Server Infrastructure

1. Design for capacity requirements.

Analyze storage requirements.

Analyze network requirements.

Analyze CPU requirements.

Analyze the current configuration.

Analyze memory requirements.

Forecast and incorporate anticipated growth requirements into the capacity requirements.

2. Specify software versions and hardware configurations.

Choose a version and edition of the operating system

Choose a version of SQL Server 2005.

Choose a CPU type.

Choose memory options.

Choose a type of storage.

3. Design physical storage.

Design transaction log storage.

Design backup file storage.

Decide where to install the operating system.

Decide where to place SQL Server service executables.

Specify the number and placement of files to create for each database

4. Design instances.

Decide how many databases to create.

Decide on the placement of system databases for each instance.

Decide on the physical storage for the tempdb database for each instance

Decide on the number of instances

Decide on the naming of instances.

Decide how many physical servers are needed for instances.

Establish service requirements.

Specify instance configurations.

5. **Design a database consolidation strategy.**
 - Gather information to analyze the dispersed environment.
 - Identify potential consolidation problems.
 - Create a specification to consolidate SQL Server databases.
 - Design a database migration plan for the consolidated environment
 - Test existing applications against the consolidated environment.
6. **Designing Security for a Database Server Solution**
 - Analyze business requirements.
 - Gather business and regulatory requirements.
 - Decide how requirements will impact choices at various security levels.
 - Evaluate costs and benefits of security choices.
 - Decide on appropriate security recommendations.
 - Inform business decision makers about security recommendations and their impact
 - Incorporate feedback from business decision makers into a design.
7. **Integrate database security with enterprise-level authentication systems.**
 - Decide which authentication system to use.
 - Design Active Directory organizational units (OUs) to implement server-level security policies.
 - Ascertain the impact of authentication on a high-availability solution.
 - Establish the consumption of enterprise authentication.
 - Ascertain the impact of enterprise authentication on service up-time requirements.
8. **Develop Microsoft Windows server-level security policies.**
 - Develop a password policy.
 - Develop an encryption policy
 - Specify server accounts and server account rights.
 - Specify the interaction of the database server with antivirus software
 - Specify the set of running services and disable unused services
 - Specify the interaction of the database server with server-level firewalls
 - Specify a physically secure environment for the database server
9. **Modify the security design based on the impact of network security policies**
 - Analyze the risk of attacks to the server environment and specify mitigations
 - Design SQL Server service-level security
 - Specify logins
 - Select SQL Server roles for logins
 - Specify a SQL Server service authentication mode
 - Design a secure HTTP endpoint strategy
 - Design a secure job role strategy for the SQL Server Agent Service
 - Specify a policy for .NET assemblies
10. **Design database-level security.**
 - Specify database users.
 - Design schema containers for database objects
 - Specify database roles
 - Define encryption policies
 - Design DDL triggers
 - Design object-level security

- Design a permissions strategy
- Analyze existing permissions
- Design security for CLR objects in the database
- 11. Designing a Physical Database**
 - Modify an existing database design based on performance and business requirements
 - Ensure that a database is normalized.
 - Allow selected denormalization for performance purposes
 - Ensure that the database is documented and diagrammed
- 12. Design tables.**
 - Decide if partitioning is appropriate
 - Specify primary and foreign keys
 - Specify column data types and constraints
 - Decide whether to persist computed columns
 - Specify physical location of tables, including file groups and a partitioning scheme
- 13. Design file groups.**
 - Design file groups for performance
 - Design file groups for recoverability
 - Design file groups for partitioning
- 14. Design index usage.**
 - Design indexes for faster data access.
 - Design indexes to improve data modification
 - Specify physical placement of indexes
 - Design views.
 - Analyze business requirements
 - Specify row and column filtering
- 15. Create database conventions and standards.**
 - Define database object-naming conventions
 - Define consistent synonyms.
 - Define database coding standards.
 - Document database conventions and standards
 - Create database change-control procedures
 - Establish where to store database source code
 - Define procedures for moving from development to test.
 - Define procedures for promoting from test to production
 - Document the database change-control procedures
- 16. Designing a Database Solution for High Availability**
 - Select high-availability technologies based on business requirements
 - Analyze availability requirements.
 - Analyze environmental issues
 - Analyze potential problems related to processes and staff
 - Decide how quickly the database solution must failover
 - Choose automatic or manual failback.
 - Analyze costs versus benefits of various solutions
 - Combine high-availability technologies to improve availability

- 17. Develop a strategy for migration to a highly available environment**
 - Analyze the current environment
 - Ascertain migration options.
 - Choose a migration option.
 - Design a highly available database storage solution
 - Design the RAID solutions for your environment
 - Design a SAN solution.
- 18. Design a database-clustering solution.**
 - Design a Microsoft Cluster Service (MSCS) implementation
 - Design the cluster configuration of the SQL Server service
- 19. Design database mirroring.**
 - Design server roles for database mirroring
 - Design the initialization of database mirroring.
 - Design a test strategy for planned and unplanned role changes
- 20. Design a high-availability solution that is based on replication**
 - Specify an appropriate replication solution.
 - Choose servers for peer-to-peer replication
 - Establish a strategy for resolving data conflicts
 - Design a strategy to reconnect client applications
 - Design log shipping
 - Specify the primary server and secondary server
 - Switch server roles.
 - Design an application failover strategy
 - Design a strategy to reconnect client applications
- 21. Designing a Data Recovery Solution for a Database**
 - Specify data recovery technologies based on business requirements.
 - Analyze how much data the organization can afford to lose
 - Analyze alternative techniques to save redundant copies of critical business data
 - Analyze how long the database system or database can be unavailable
- 22. Design backup strategies.**
 - Specify the number and location of devices to be used for backup
 - Specify what data to back up
 - Specify the frequency of backup.
 - Specify the type of backup.
 - Create a disaster recovery plan.
 - Document the sequence of possible events.
 - Create a disaster decision tree that includes restore strategies
 - Establish recovery success criteria.
 - Validate restore strategies.