



SQL SERVER TABLE STRUCTURE AND INDEX

Course ID : SQL-PG-TBIX



Scan for Details
on Our [Website](#)



Duration: 3 Days

(18 Hours) 09:00 AM – 04:00 PM



Price: Call (Inhouse training only)

* (excluding VAT 7%)

* Eligible for 200% tax deduction



Training Schedule

www.9experttraining.com

Category: Data

This course introduces table structures and various types of indexes, guiding participants in selecting the most appropriate options for different scenarios. It covers the execution mechanisms in Microsoft SQL Server and teaches participants how to analyze index performance within compiled plans to achieve optimal efficiency.

Objectives

1. Participants will understand different types of table structures.
2. Participants will understand various index structures.
3. Participants will be able to analyze the performance of compiled execution plans effectively.

Target Audience

1. Microsoft SQL Server administrators
2. Consultants
3. Directors and management personnel
4. Interested individuals

Prerequisites

Some experience in administering Microsoft SQL Server

System Requirements

1. Operating System: Windows 11 or 10
2. CPU : 1.6 GHz or faster
3. RAM : 8 GB or higher
4. Remote Desktop Connection
5. Stable Internet access

TRAINING TOPICS

DAY 1 Morning Session

9:00 AM – 12:00 PM

1. Guidelines for Table Design

- Table design principles
- Data normalization
- Understanding table relationships
- Primary keys and foreign keys
- The relationship between normalization and concurrency control
- Considering surrogate keys as primary key candidates
- Assigning data types to table columns

- Understanding data types and their variations
- Setting nullability
- Using and creating schemas
- Object naming conventions
- Creating and modifying table structures
- Introduction to temporary tables and their usage
- Computed columns

DAY 1 Afternoon Session

1:00 PM – 4:00 PM

2. Advanced Table Design

- Table partitioning
- Partition functions and partition schemes
- Creating partitioned tables and partitioned indexes
- Partitioning strategies
- Data compression: rationale, row-level, page-level, and Unicode compression
- Considerations for data compression
- Introduction to temporal tables and their creation
- Enabling system-versioning on existing table structures
- Key considerations for implementing temporal tables

3. Enforcing Data Integrity with Constraints

- Types of data integrity
- Options for enforcing data integrity
- Domain integrity and data type assignment
- Enforcing DEFAULT and CHECK constraints
- Enforcing entity and referential integrity
- Enforcing PRIMARY KEY, UNIQUE, and IDENTITY constraints
- Using sequences
- Enforcing FOREIGN KEY constraints with cascading referential actions
- Key considerations when implementing constraints

4. Understanding Index Operations

- Index creation concepts
- Data access methods in SQL Server
- Index structures
- Selectivity, density, and index depth
- Index fragmentation
- Indexes for different data types
- Indexes on computed columns
- Table structures: heap tables and clustered indexes
- Data operations on heap and clustered index structures
- The relationship between primary keys and clustering keys

- Non-clustered index structures and operations
- Single-column and multi-column index design
- Sorting order considerations
- Understanding and maintaining statistics

5. Advanced Index Usage

- Index usage strategies
- Covering indexes and the INCLUDE clause
- Filtered indexes
- Index management
- Fill factor and PAD_INDEX settings
- Index summaries and query hints
- Execution plans: actual vs. estimated, key components, and methods for capturing plans
- Live query statistics
- Introduction to columnstore indexes

6. Understanding Columnstore Indexes

- Non-clustered columnstore indexes
- Clustered columnstore index table structures
- Creating columnstore indexes (non-clustered and clustered)
- Creating clustered columnstore tables with primary keys and foreign keys

7. Using In-Memory Tables

- Memory-optimized tables and their use cases
- Creating filegroups for memory-optimized data
- Creating and querying memory-optimized tables
- Indexing options for memory-optimized tables
- Converting existing tables using Memory Optimization Advisor
- Introduction to natively compiled stored procedures and their use cases
- Enabling execution statistics for performance monitoring



 Download [PDF](#)