



Microsoft Office

866-963-4440

---

## **2782– Designing Microsoft SQL Server 2005 Database**

**2 Day**

This two-day instructor-led course provides students with the knowledge and skills to design databases for Microsoft SQL Server 2005 using business requirements to guide their decisions (beyond structured third normal form [3NF] modeling techniques). Students will also learn to incorporate security requirements throughout their design.

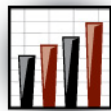
### **WHO SHOULD ATTEND**

This course is intended for current professional database developers who have three or more years of on-the-job experience developing SQL Server database solutions in an enterprise environment.

### **PREREQUISITES**

Before attending this course, students must:

- Have experience reading user requirements and business-need documents. For example, development project vision/mission statements or business analysis reports.
- Have experience reading and drawing business process flow charts.
- Have experience reading and drawing entity relationship (ER) diagrams.
- Understand Transact-SQL syntax and programming logic.
- Be able to design a database to 3NF and know the tradeoffs when backing out of the fully normalized design (denormalization) and designing for performance and business requirements in addition to being familiar with design models, such as Star and Snowflake schemas.
- Have basic monitoring and troubleshooting skills.
- Have basic knowledge of the operating system and platform. That is, how the operating system integrates with the database, what the platform or operating system can do, and how interaction between the operating system and the database works.
- Have basic knowledge of application architecture. That is, how applications can be designed in three layers, what applications can do, how interaction between the application and the database works, and how the interaction between the database and the platform or operating system works.
- Know how to use a data modeling tool.
- Be familiar with SQL Server 2005 features, tools, and technologies.
- Have a Microsoft Certified Technology Specialist: Microsoft SQL Server 2005 credential, or equivalent experience.



Microsoft Office

866-963-4440

---

In addition, it is recommended, but not required, that students have completed:

- Course 2778: Writing Queries Using Microsoft SQL Server 2005 Transact-SQL.
- Course 2779: Implementing a Microsoft SQL Server 2005 Database.
- Course 2780: Maintaining a Microsoft SQL Server 2005 Database.

## **AT COURSE COMPLETION**

After completing this course, students will be able to:

- Approach database design from a systematic perspective, gather database requirements, and formulate a conceptual design.
- Analyze and evaluate a logical database design.
- Apply best practices for creating a physical database design.
- Apply best practices when designing for database scalability.
- Design a database access strategy.
- Use best practices to model database dependencies.

## **LESSON TOPICS**

### **Module 1: Approaching Database Design Systematically**

This module explains how to acquire the skills to approach database design with a systematic perspective. A systematic approach involves formulating your database design process, following guidelines on how to gather and document database requirements, and following best practices when formulating a conceptual design.

#### **Lessons**

- Overview of Database Design
- Gathering Database Requirements
- Creating a Conceptual Database Design

#### **Lab : Beginning the Database Design Process**

- Gathering Database Requirements
- Creating a Conceptual Design



Microsoft Office

866-963-4440

---

After completing this module, students will be able to:

- Apply a systematic approach to database design.
- Devise an appropriate strategy for gathering database requirements for a specified project.
- Formulate requirements into a conceptual model that serves as a basis for defining entities, attributes, and relationships.

## **Module 2: Modeling a Database at the Logical Level**

This module explains the best practices followed when you build a new logical database model. You will also learn the guidelines for normalization when designing an OLTP model and when designing a data warehouse database. Finally, you will learn to evaluate the existing logical model of a database.

### **Lessons**

- Building a Logical Database Model
- Designing for OLTP Activity
- Designing for Data Warehousing
- Evaluating Logical Models

### **Lab : Modeling a Database at the Logical Level**

- Determine Entities, Attributes, Relationships, Keys, and Constraints
- Normalization and Schema Assignment

After completing this module, students will be able to:

- Apply best practices to the task of building a new logical database model.
- Apply guidelines for normalization when designing an OLTP model.
- Apply guidelines for designing a data warehouse database.
- Evaluate an existing logical model of a database.

## **Module 3: Modeling a Database at the Physical Level**

This module explains the guidelines to be followed when designing physical database objects and constraints. The module also covers the best practices for designing database security and for designing database and server options. Finally, this module covers the best practices for evaluating the physical model.

### **Lessons**

- Designing Physical Database Objects



Microsoft Office

866-963-4440

---

- Designing Constraints
- Designing for Database Security
- Designing Server and Database Options
- Evaluating the Physical Model

#### **Lab : Modeling a Database at the Physical Level**

- Specify Database Object Naming Standards
- Define Tables and Columns and Choose Data Types

After completing this module, students will be able to:

- Apply guidelines for designing physical database objects.
- Apply best practices when designing constraints.
- Include security best practices in the design of databases.
- Apply best practices when designing database and server options.
- Apply best practices when evaluating the physical model.

#### **Module 4: Designing for Database Performance**

This module explains the best practices to be followed for designing indexes. The module also covers the guidelines for planning table optimization, and choosing additional optimization techniques.

##### **Lessons**

- Designing Indexes
- Planning for Table Optimization
- Planning for Database Optimization

#### **Lab : Designing for Database Scalability**

- Apply Optimization Techniques

After completing this module, students will be able to:

- Apply best practices for designing indexes.
- Apply guidelines when planning for table optimization.
- Apply guidelines in choosing additional optimization techniques.



Microsoft Office

866-963-4440

---

## **Module 5: Designing a Database Access Strategy**

This module explains the best practices to be followed when designing for secure data access. The module also covers the guidelines for designing user-defined functions. Finally, this module explains the best practices for designing stored procedures.

### **Lessons**

- Designing for Secure Data Access
- Designing User-Defined Functions
- Designing Stored Procedures

### **Lab : Designing a Database Access Strategy**

- Design data retrieval objects
- Design security for data retrieval objects

After completing this module, students will be able to:

- Apply best practices when designing for secure data access.
- Apply guidelines for designing user-defined functions.
- Apply best practices for designing stored procedures.

## **Module 6: Modeling Database Dependencies**

This module explains guidelines for modeling local database dependencies. This module also covers the guidelines for modeling remote database dependencies.

### **Lessons**

- Modeling Local Database Dependencies
- Modeling Remote Database Dependencies

### **Lab : Modeling Database Dependencies**

- Design Cross-Database Access
- Design Linked Servers

After completing this module, students will be able to:

- Apply guidelines for modeling local database dependencies.
- Apply guidelines for modeling remote database dependencies.