

# AGILE DATABASE DEVELOPMENT USING VISUAL STUDIO 2010

ADD2010 | 3 Days



## INTRODUCTION

This three-day, instructor-led course provides students with the knowledge and skills to properly manage the SQL Server 2005/2008 database development lifecycle in an Agile environment. The student will learn how to manage changes to the database schema, ensuring quality through T-SQL unit testing, and how to automate the building and deploying of SQL Server databases. Intended for developers and administrators already comfortable with the concepts of designing, programming, testing and deploying SQL Server databases, this course focuses on using Visual Studio 2010 Premium/Ultimate edition to maximize productivity while minimizing mistakes.

## AUDIENCE

This course is intended for SQL Server 2005/2008 database developers and administrators. Application developers who are involved with writing and testing T-SQL code and working with test data will also gain value from this course.

## AT COURSE COMPLETION

After attending this course, students will be able to:

- Understand the database development capabilities of Visual Studio 2010
- Understand the areas of integration with Team Foundation Server
- Understand how the tools support an Agile database development lifecycle
- Work offline so changes don't impact actual servers
- Create and configure database projects
- Create and configure server projects
- Import existing schemas and scripts
- Place database projects under Team Foundation Version Control
- Compare schemas for differences from two SQL Server instances
- Compare data for differences from two SQL Server instances
- Generate scripts to reconcile differences in schemas and data
- Understand and create effective T-SQL unit tests
- Understand and use database refactoring correctly
- Automatically generate test data for various scenarios
- Integrate test data generation with T-SQL unit testing
- Automate the building and deploying of database changes
- Use MSBuild and Team Build to manage build and deployment
- Extend Visual Studio by creating your own database refactoring method
- Extend Visual Studio by creating your own static code analysis rule
- Extend Visual Studio by creating your own test data generator
- Extend Visual Studio by creating your own unit test condition

## PREREQUISITES

Before attending this course, the student should have experience developing and administering SQL Server 2005/2008 databases. It would also be beneficial if the student has worked on a team-based software development project and is familiar with their organization's development lifecycle and practices.

## **COURSE OUTLINE**

### **Module 1: Agile Database Development**

This module reminds us of the reality of developing and maintaining SQL Server databases in an Agile environment. Topics include understanding Agile database development and techniques, especially those that are supported in Visual Studio 2010.

#### Lessons

- Agile database development
- Agile database techniques
- Visual Studio 2010 Premium Edition
- Case study
- Resources

#### Lab Exercises

- Meet the AdventureWorks team
- Review the environment
- Deploy the database

### **Module 2: Database Projects**

This module introduces database projects and how they support an offline, disconnected mode of development and testing, which is important in an Agile environment. Topics include a discussion of the integration with Visual Studio 2010 and support for importing schema and scripts, server projects, permissions and how to handle dependencies (3- and 4-part names).

#### Lessons

- Architecture of a database project
- Creating and managing database projects
- Visual Studio 2010 integration
- Importing database schema
- Importing SQL scripts
- Permissions

#### Lab Exercises

- Explore Visual Studio integration
- Create a new database project
- Modify the database project
- Import a database schema
- Import a SQL script

## **Module 3: Change Management**

This module discusses how database projects integrate with Team Foundation Server's change management capabilities. Topics include an introduction to Team Foundation Server, working with Team Projects, version control, work item integration, as well as the schema and data compare and change management features.

### Lessons

- Introduction to Team Foundation Server
- Working with Team Projects
- Process templates and work items
- Using Team Foundation Version Control
- Placing database projects under version control
- Detecting and reconciling database schema differences
- Detecting and reconciling table data differences

### Lab Exercises

- Create a Team Project and work items
- Configure Team Foundation Version Control
- Place a database project under version control
- Explore and Visual Studio 2010 version control features
- Compare schemas for differences from two SQL Server instances
- Compare data for differences from two SQL Server tables
- Generate scripts to reconcile differences in schemas and data
- Explore the MSSCCI provider from SQL Server Management Studio (optional)

## **Module 4: Unit Testing**

This module introduces the need for writing and running T-SQL unit tests within the Visual Studio 2010. Topics include creating test projects, writing good unit tests and the integration of T-SQL unit tests with your existing .NET unit tests. Static code analysis is also covered in this module.

### Lessons

- Reasons to unit test your database code
- How to create database unit tests manually
- How to create database unit tests automatically
- Assertions vs. conditions
- Executing unit tests
- Managing and publishing test results
- Static code analysis

### Lab Exercises

- Create and execute database unit tests
- Use pre-test and post-test actions
- Use an ordered list to run tests
- Export and import test results
- Create unit tests automatically from the Schema View window
- Run static code analysis

## **Module 5: Refactoring**

This module introduces the concept of refactoring databases and the various refactoring methods found in Visual Studio 2010.

### Lessons

- Introduction to refactoring
- Refactoring support in Visual Studio 2010
- Database refactoring methods
- Rename refactoring methods
- Move refactoring method
- Fully-qualify refactoring method
- Expand-wildcard refactoring method
- Resources

### Lab Exercises

- Approaches to renaming database objects
- Use the rename refactor method
- Preview changes
- Undo refactoring using global undo
- Review the refactoring log

## **Module 6: Data Generation**

This module discusses the issues surrounding test data, from security risks to the loss of functionality, when working with less than perfect data. Topics include the data generation architecture and how data generation plans can be used to generate meaningful, deterministic data for any number of reasons.

### Lessons

- Issues with test data and some potential solutions
- Data generation plans
- Data generators
- Regular expression generator
- Data bound generator
- Creating and running a data generation plan
- Automatically generating data prior to running unit tests

### Lab Exercises

- Create a data generation plan
- Use the Regular Expression data generator
- Use the Data Bound generator
- Configure generation for related table and set ratios
- Generate data automatically prior to running unit tests

## **Module 7: Build and Deploy**

This module wraps up the lifecycle with the discussion of building and deploying any schema changes to a target SQL Server instance. Topics include the build and deployment architecture, integration with MSBuild, deployment options and scripts, and how to automate the entire process using Team Foundation Build. The discussion of how to manage and deploy-to multiple SQL Server environments and handle changes to static/lookup data are also included in this module.

### Lessons

- The need for automated build and deployment
- Building database projects
- Deploying schema changes
- The VSDBCMD.exe command line utility
- Integration with MSBuild
- Integration with Team Foundation Build

### Lab Exercises

- Build the database project
- Deploy database schema changes
- Use custom deployment scripts
- Automate using MSBuild and Team Foundation Build

## **Module 8: Extensibility**

This module describes the various extensibility points found within the Visual Studio 2010 as they pertain to SQL Server database development. Topics include customizing the various project and item templates, building custom refactoring methods, building custom static code analysis rules, building custom test data generators, and building custom unit test conditions.

### Lessons

- Visual Studio 2010 Areas of extensibility
- Visual Studio database project and item templates
- Custom database refactoring methods
- Custom static code analysis rules
- Custom test data generators
- Custom unit test conditions
- Custom check-in policies

### Lab Exercises

- Customize database object templates
- Create a custom data generator
- Create a custom unit test condition
- Create a custom code analysis rule (optional)
- Create a custom refactoring method (optional)
- Create and debug a custom check-in policy (optional)

## **Course Designer**

This course was designed by Richard Hundhausen of Accentient, Inc. Richard is a Visual Studio ALM MVP and Microsoft Regional Director, as well as an experienced developer and trainer.

For more information, contact Mark Dunn at 770 653-6364 or email [mark@dunntesting.com](mailto:mark@dunntesting.com).