

## **AGILE DATABASE TECHNIQUES USING VISUAL STUDIO TEAM SYSTEM 2008**

Course ADT08: Three days; Instructor-Led Course Syllabus



### **INTRODUCTION**

This three-day, instructor-led course provides students with the knowledge and skills to properly manage the SQL Server database development lifecycle in an agile environment. The student will learn how to manage changes to the database structure, 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 Team System 2008 Database Edition to maximize productivity while minimizing mistakes.

### **AUDIENCE**

This course is intended for SQL Server 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 capabilities of the VSTS Database Edition
- Understand the areas of integration with Team Foundation Server
- Understand how VSTS supports an agile database development life cycle
- Work offline so changes don't impact actual servers
- Create and configure database projects
- Import existing schemas and scripts
- Place database projects under Team Foundation Version Control
- Compare schemas from two SQL Server instances for differences
- Compare data from two SQL Server instances for differences
- 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 in 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 VSTS by building your own test data generators
- Extend VSTS by building your own unit test assertions

### **PREREQUISITES**

Before attending this course, the student should have experience developing and administering SQL Server 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 Software Development Life Cycle.

## **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 found in the Visual Studio Team System Database Edition.

#### Lessons

- Agile database development
- Agile database techniques
- Visual Studio 2008 Team System Database Edition
- Sample scenario
- Resources

#### Lab Exercises

- Meet the virtual team
- Review the environment
- Deploy the AdventureWorks 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 2008 and support for importing schema and scripts into these database projects.

#### Lessons

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

#### Lab Exercises

- Explore Visual Studio integration
- Create a new database project
- Modify the database project
- Import a database schema
- Import a SQL script
- Explore the design-time validation database (optional)

## **Module 3: Change Management**

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

### Lessons

- Introduction to Team Foundation Server
- Working with team projects
- Methodologies and work items
- Using Team Foundation Version Control
- Placing database projects under version control
- Comparing two database schemas for differences
- Comparing data from two databases for differences
- Generating change scripts to merge differences

### Lab Exercises

- Create a team project and work items
- Configure version control
- Place a database project under version control
- Explore Visual Studio 2008 version control integration
- Use version control features
- Compare schemas from two SQL Server instances for differences
- Compare data from two SQL Server tables for differences
- 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 unit testing and related support within the Database Edition. Topics include test projects, writing good unit tests, and the integration of T-SQL unit tests with your existing .NET unit tests.

### Lessons

- Unit testing
- Creating database unit tests
- SQL assertions and test conditions
- Executing unit tests and managing test results

### 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

## **Module 5: Refactoring**

This module introduces the concept of refactoring and its support in the Database Edition.

### Lessons

- Introduction to refactoring
- Refactoring support in Visual Studio 2008
- Code refactoring
- Database refactoring
- Rename refactoring
- Regression testing
- Resources

### Lab Exercises

- Rename schema objects using various methods
- 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 meaningless 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 potential solutions
- Data generation plans
- Data generators
- Regular expression generator
- Data bound generator
- Creating and running a data generation plan
- Integrating with unit testing

### Lab Exercises

- Review the existing schema and database unit tests
- Create a data generation plan
- Use a regular express generator
- Use a 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 life cycle with the discussion of building and deploying any schema changes to a target SQL Server instance. Topics include the build and deployment architecture, Team Foundation Build and MSBuild integration, deployment options and scripts, and how to automate the entire process.

### Lessons

- The need for automated build and deployment
- Building
- Deploying
- Integration with MSBuild
- Integration with Team Build

### Lab Exercises

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

## **Module 8: Extensibility**

This module describes the various extensibility points found within the Visual Studio Team System 2008 Database Edition, including templates, data generators, unit test conditions, and check-in policies.

### Lessons

- Customization vs. extensibility
- Areas of extensibility
- Visual Studio templates
- Custom test conditions
- Custom data generators
- Custom check-in policies

### Lab Exercises

- Edit database object templates
- Create a custom data generator
- Create a custom unit test assertion
- Create a custom check-in policy (optional)
- Debug a custom check-in policy (optional)

## **Course Designer**

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

For more information, visit [www.accentient.com](http://www.accentient.com)