Suggested Practices

Chapter 1: Selecting and Designing SQL Server Services to Support Business Needs

Mapping Technologies and Goals

- **Practice** In SQL Server 2005, you can disseminate data in many different ways, including:
  - Backing up a database and restoring it to a new database.
  - Detaching a database, copying it, and attaching it to a different SQL Server instance.
  - Using log shipping.
  - Using database mirroring and database mirroring with database snapshot on the mirrored database.
  - Employing snapshot, transactional, or merge replication.
  - Using Integration Services.
  - Using INSERT . . . SELECT and SELECT INTO T-SQL statements, which can copy data to new tables in the current database, a different database on the same instance, or a database in a different instance.

  For each of the following scenarios, evaluate which of the preceding technologies are most appropriate:

  - You must maintain a read-only copy of part of the data with low latency.
  - You have to maintain high availability.
  - You need to prepare a disaster recovery plan.
  - You must maintain multiple read-write copies of a database or parts of a database.
  - You have to maintain a data warehouse with many transformations on the data.
  - You have to prepare an ad hoc copy of a small part of the data for quick analysis.

Mapping Technologies to Specific Needs

- **Practice** You can analyze data through reports by using Reporting Services, through UDM cubes by using Analysis Services, and through data mining algorithms by using Analysis Services. Which technology would you use when your business needs are as follows?
  - Your company's advanced analysts need hundreds of different aggregate views, in real time, of data that has been collected over five years.

- Your customer relationship management (CRM) application users need a deeper understanding of their customers. Specifically, they need to know whether any significant groups of customers have a lot of similarity between customers in the same group and a lot of dissimilarity between customers from different groups.
- You have users who have only basic technical knowledge and who need the current data of only a single customer per query. The users have access to your company intranet, but they cannot use rich analytical tools such as Excel.

Chapter 2: Designing a Logical Database

Designing a Logical Database

- **Practice 1** Create a diagram for the *AdventureWorks* sample database. (Refer to the "Before You Begin" section of Chapter 2 for more information.) Use any diagramming tool with which you are familiar. If you do not have a specialized diagramming tool, you can create ER diagrams quickly by using SQL Server Management Studio. Try to distinguish the tables that came to the model through normalization and the supertype and subtype tables.
- **Practice 2** Create a diagram for the *AdventureWorksDW* demo database. Compare it with the AdventureWorks diagram. Which one is simpler? Why? Can you imagine any additional data flow needed?

Chapter 3: Designing a Physical Database

Designing a Physical Database

- **Practice** Check the triggers and constraints in the *AdventureWorks* demo database.

Chapter 4: Designing a Database for Performance

Designing a Database for Performance

- **Practice 1** Analyze the indexes in the *AdventureWorks* database.
- **Practice 2** Check the views in the *AdventureWorks* database. Decide which of those views would benefit from indexing.

Chapter 5: Using Appropriate Database Technologies and Techniques for Your Application

Using Appropriate Database Technologies and Techniques for Your Application

- **Practice 1** Using the dbo.FactResellerSales table in the *AdventureWorksDW* demo database, try to partition the table and its indexes so that the indexes are aligned.
- **Practice 2** Think of the databases in your company or in a company you know. Examine where the data is already aggregated and whether you can find any better way to design the aggregations.

Chapter 6: Designing Objects That Retrieve Data

Designing Objects That Retrieve Data

- **Practice 1** Analyze the views, stored procedures, and functions in the *AdventureWorks* demo database.
- **Practice 2** Review the views in the *AdventureWorks* demo database and think about which of those views would benefit from indexing.
- **Practice 3** Looking at the tables in the *AdventureWorks* demo database, which tables would match a partitioning scheme?
- **Practice 4** Which stored procedures could be translated into UDFs for flexibility?

Chapter 7: Designing Objects That Extend Server Functionality

Designing Objects That Extend Server Functionality

- **Practice 1** Create stored procedures to maintain information in tables. Create two different stored procedures to insert customer data: one to insert individual customer data and another to insert customer data that is also organizations' data or business data. The stored procedures should update multiple tables and follow best practices learned in Chapter 7.
- **Practice 2** Review the SQL Server 2005 Books Online AdventureWorksSQLCLRLayer sample. You can install the sample code by running the installer from the following path: C:\Program Files\Microsoft SQL Server\90\Tools\Samples\SqlServerSamples.msi. By default, the sample is installed in C:\Program Files\Microsoft SQL Server\90\Samples\Engine\Programmability\CLR\AdventureWorks.

Chapter 8: Designing a Secure Application Solution

Designing Security

- **Practice 1** Create an SSIS package. Encrypt the complete package with a user key. Create a scheduled job that uses an SSIS task to execute the package. Run the job. Try to make the job work without exposing sensitive information.
- **Practice 2** Use SQL Server Profiler to trace all reads from a SQL Server database and an Analysis Services cube.

Chapter 9: Designing a Secure Database

Designing Security

- **Practice 1** Create two database users. Use the EXECUTE AS statement to change the execution context from your current context to the first user context. Without reverting the context, change it again to the second user you created. Revert the context for the first time. Revert it for the second time. Use the *user_name()* function to check the execution context in each step.
- **Practice 2** List all the ways you can hide data from end users and applications.

Chapter 10: Designing a Unit Test Plan for a Database

Creating a Test Project

- **Practice** Create a complete testing project, including test scripts, setup scripts, teardown scripts, a testing database, and testing data.

Analyzing Data and Creating Unit Tests

- **Practice** Analyze the views, stored procedures, functions, and constraints defined in the *AdventureWorks* sample database and create unit tests for some of them.

Performing Resource Use Tests

- **Practice** Review the performance of some of the AdventureWorks stored procedures by executing a resource use test.

Implementing Code Coverage

- **Practice** Implement a code coverage solution, using the material from these suggested practices.

Chapter 11: Creating a Database Benchmarking Strategy

Creating a Plan

- **Practice** Analyze the *AdventureWorks* sample database and set performance objectives for its execution on your computer. Validate your metrics by writing some unit tests (as discussed in Chapter 10, "Designing a Unit Test Plan for a Database").

Using a Baseline to Validate Performance Objectives

- **Practice** Generate a baseline for AdventureWorks and validate the measurements against the performances objectives you set in the preceding practice.

Identifying Ways to Improve Performance

- **Practice** Look at different ways to improve the performance of the *AdventureWorks* database on your computer.

Implementing a Plan

- **Practice** Implement a plan for responding to performance changes. What actions should you take if memory consumption increases? What actions should you take if connection contention occurs?

Chapter 12: Creating a Plan for Deploying a Database

Creating a Plan for Deploying a Database

- **Practice 1** Deploy the entire *AdventureWorksLT* database by using the Copy Database Wizard.
- **Practice 2** Use SSIS to accomplish the required tasks defined in the case scenario.

Chapter 13: Controlling Changes to Source Code

Controlling Changes to Source Code

- **Practice** Implement extended properties on all objects in the TestDB or TestDB2 database project created in this Chapter 13 to describe Object_Version. (TestDB is source-controlled with Visual Studio Team Foundation Server, and TestDB2 is source-controlled with Visual SourceSafe.) Include an extended property on the database objects themselves. Add these queries to their respective projects and source control.

Chapter 14: Designing for Data Distribution

Configure DatabaseMail

- **Practice** Configure DatabaseMail by using stored procedures. The Simple Database-Mail Configuration template can help.

Perform Tasks with SSMS

- **Practice** Using SSMS, perform the following tasks:
  - Create an operator.
  - Create an alert based on a performance condition.
  - Configure an alert to notify the operator as the response.
  - Force the performance condition and check that the operator is notified.

Create a Notification Services Application

- **Practice** Create a simple Notification Services application. To accomplish this, you can take advantage of the Notifications Services Tutorial at *http://msdn2.microsoft.com/en-us/library/ms170337.aspx*.

Use the Reporting Services Tools

- **Practice** Look at the Reporting Services Tools. You can take advantage of the Reporting Services Tools tutorial at *http://msdn2.microsoft.com/en-us/library/aa337424.aspx*.

Secure an HTTP Endpoint

- **Practice** Using the HTTP endpoint you created in the practice exercises in Lesson 3, "Specifying a Web Services Solution for Distributing Data," create some users and grant CONNECT permissions on the HTTP endpoint to them.

Chapter 15: Designing Applications That Support Reporting and Use Reporting Services

Designing Applications That Support Reporting and Use Reporting Services

- **Practice 1** Download the Financial Reporting Services Pack from *http://msdn2.microsoft.com/en-us/architecture/aa902621.aspx*. Install the application and study the provided reports, evaluating the techniques necessary to provide users with different SSRS experiences.

- **Practice 2** Practice the whole SSRS cycle: development, management, and report delivery. Use BIDS to develop reports, use SSMS and Report Manager to deploy and manage reports, and use Report Manager to experience report delivery.
- **Practice 3** Review the SQL Server 2005 Books Online SQL Server Reporting Services samples. You can install the sample code by running the installer from the following path: C:\Program Files\Microsoft SQL Server\90\Tools\Samples\SqlServerSamples.msi. By default, the sample is installed in C:\Program Files\Microsoft SQL Server\90\Samples\Reporting Services\Report Samples\AdventureWorks Sample Reports.

Chapter 16: Developing Applications for Notification Services

Using ICF and NMO

- **Practice** Configure a Notification Services instance by using both NMO and an ICF.

Using ADF and NMO

- **Practice** Configure an application by using both NMO and an ADF.

Reviewing Event Class and Event Provider Properties

- **Practice** Review the properties defined in an event class and in an event provider, which are discussed in Lesson 2, "Defining Notification Services Events and Event Providers." Plan the application data for one of your own projects.

Reviewing Subscription Elements

- **Practice** Review subscription elements that are common to typical subscriptions, such as subscription and event rules. These are discussed in Lesson 5, "Creating Subscriptions."

Planning Indexing for Performance

- **Practice** Analyze the Notification Services database tables in one of your projects. Consider what type of columns to index to improve performance.

Chapter 17: Developing Packages for Integration Services

Developing Packages for Integration Services

- **Practice 1** Use the SQL Server Import and Export Wizard to create SSIS packages. You can start the wizard from SSMS when you right-click any database and select Tasks and then Import Data.
- **Practice 2** Design and create some packages to populate the *AdventureWorks DW* database with data from the *AdventureWorks* relational database.
- **Practice 3** Review the SQL Server 2005 Books Online SQL Server Integration Services Samples. You can install the sample code by running the installer from the following path: C:\Program Files\Microsoft SQL Server\90\Tools\Samples\SqlServerSamples. msi. By default, the sample is installed in drive :\Program Files\Microsoft SQL Server \90\Samples\Integration Services\Package Samples.