



**РОСТИСЛАВ
МИХЕЕВ**

MS SQL Server 2005

для администраторов

СПЕЦИАЛЬНЫЙ КУРС

УСТАНОВКА И НАСТРОЙКА SQL SERVER 2005

**НАСТРОЙКА БЕЗОПАСНОСТИ, ШИФРОВАНИЕ ДАННЫХ
СРЕДСТВАМИ SQL SERVER 2005**

**РЕЗЕРВНОЕ КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ,
СРЕДСТВА ОБЕСПЕЧЕНИЯ ОТКАЗОУСТОЙЧИВОСТИ**

**АВТОМАТИЗАЦИЯ АДМИНИСТРАТИВНЫХ ОПЕРАЦИЙ,
ОБЪЕКТНЫЕ МОДЕЛИ SMO, SQL-DMO И WMI**

**ПЕРЕНОС И ПРЕОБРАЗОВАНИЕ ДАННЫХ
СРЕДСТВАМИ SQL SERVER INTEGRATION SERVICES**

**МОНИТОРИНГ И ОПТИМИЗАЦИЯ
ПРОИЗВОДИТЕЛЬНОСТИ SQL SERVER 2005**

Ростислав Михеев

MS SQL Server 2005

для администраторов

СПЕЦИАЛЬНЫЙ КУРС

Санкт-Петербург

«БХВ-Петербург»

2006

УДК 681.3.06
ББК 32.973.26-018.2
М69

Михеев Р. Н.

М69 MS SQL Server 2005 для администраторов. — СПб.: БХВ-Петербург, 2006. — 544 с.: ил.

ISBN 5-94157-796-6

Рассмотрены вопросы администрирования СУБД SQL Server 2005. Впервые на русском языке подробно рассматривается информация по использованию среды SSIS для передачи и преобразования данных, применение объектных моделей SMO, SQL-DMO и WMI для автоматизации администрирования, работа со встроенными средствами шифрования данных SQL Server 2005. В каждой главе приведена информация об отличиях новой версии SQL Server от предыдущей, а для начинающих администраторов предусмотрены задания для самостоятельной работы с подробными ответами. Книгу можно использовать и в качестве учебного пособия при проведении обучения по администрированию SQL Server 2005.

Для программистов и администраторов баз данных

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

| | |
|-------------------------|----------------------------|
| Главный редактор | <i>Екатерина Кондукова</i> |
| Зам. главного редактора | <i>Игорь Шишигин</i> |
| Зав. редакцией | <i>Григорий Добин</i> |
| Редактор | <i>Нина Седых</i> |
| Компьютерная верстка | <i>Ольга Сергиенко</i> |
| Корректор | <i>Зинаида Дмитриева</i> |
| Дизайн серии | <i>Инны Тачиной</i> |
| Оформление обложки | <i>Елены Беляевой</i> |
| Зав. производством | <i>Николай Тверских</i> |

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 07.07.06.

Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 43,86.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-796-6

© Михеев Р. Н., 2006
© Оформление, издательство "БХВ-Петербург", 2006

Оглавление

| | |
|---|-----------|
| Предисловие..... | 1 |
| Глава 1. Обзор новых возможностей SQL Server 2005 | 3 |
| 1.1. Новые возможности SQL Server 2005 для администраторов | 3 |
| 1.1.1. Изменения в наборе административных средств..... | 3 |
| 1.1.2. Новые возможности для оптимизации производительности | 4 |
| 1.1.3. Новые возможности системы безопасности..... | 7 |
| 1.1.4. Новые возможности для обеспечения отказоустойчивости | 8 |
| 1.1.5. Новые возможности системы репликации | 10 |
| 1.1.6. Новые объектные модели для автоматизации администрирования | 12 |
| 1.2. Новые возможности SQL Server 2005 для программистов..... | 12 |
| 1.3. Новые возможности SQL Server Integration Services (DTSX)..... | 20 |
| Глава 2. Планирование и установка SQL Server 2005 | 23 |
| 2.1. Планирование установки SQL Server 2005 | 23 |
| 2.1.1. Оценка архитектуры приложения на основе SQL Server 2005..... | 23 |
| 2.1.2. Выбор оборудования..... | 26 |
| 2.1.3. Выбор операционной системы и ее компонентов | 31 |
| 2.1.4. Выбор редакции SQL Server 2005..... | 32 |
| 2.2. Установка SQL Server 2005 | 34 |
| 2.2.1. Несколько слов об установке | 34 |
| 2.2.2. Начало установки. Выбор набора компонентов | 35 |
| 2.2.3. Работа с именованными экземплярами..... | 38 |
| 2.2.4. Выбор учетной записи для служб SQL Server..... | 40 |
| 2.2.5. Выбор режима аутентификации SQL Server 2005 | 43 |
| 2.2.6. Выбор кодировки и порядка сортировки..... | 46 |
| 2.2.7. Остальные параметры установки..... | 48 |
| 2.3. Автоматизированная и удаленная установка | 49 |
| 2.4. Обновление предыдущих версий SQL Server и миграция с Microsoft Access..... | 50 |
| 2.5. Проверка установки и выполнение послеустановочных задач | 53 |
| 2.5.1. Проверка результатов установки | 53 |
| 2.5.2. Настройка сетевых библиотек для доступа клиентов | 54 |
| 2.5.3. Другие послеустановочные задачи..... | 57 |
| Задание для самостоятельной работы 2.1. Установка именованного экземпляра SQL Server 2005 | 58 |

| | |
|--|-----------|
| Глава 3. Средства администрирования SQL Server 2005 | 61 |
| 3.1. SQL Server Management Studio | 61 |
| 3.1.1. Что такое SQL Server Management Studio | 61 |
| 3.1.2. Окно <i>Registered Servers</i> | 62 |
| 3.1.3. Окно <i>Object Explorer</i> | 64 |
| 3.1.4. Окно <i>Document</i> | 66 |
| 3.1.5. Окно <i>Solution Explorer</i> | 66 |
| 3.1.6. Другие окна SQL Server Management Studio | 68 |
| 3.1.7. Приемы работы со скриптами | 70 |
| 3.2. Business Intelligence Development Studio | 74 |
| 3.3. SQL Server Configuration Manager | 75 |
| 3.3.1. Что такое SQL Server Configuration Manager | 75 |
| 3.3.2. Службы SQL Server 2005 | 76 |
| 3.3.3. Настройка серверных сетевых библиотек средствами SQL Server Configuration Manager | 78 |
| 3.3.4. Настройка клиентских сетевых библиотек средствами SQL Server Configuration Manager. SQL Native Client | 79 |
| 3.4. SQLCmd | 81 |
| 3.4.1. Применение SQLCmd | 81 |
| 3.4.2. Специальный режим подключения Dedicated Administrator Connection | 83 |
| 3.5. SQL Server Surface Area Configuration | 84 |
| 3.6. SQL Server Profiler | 86 |
| 3.7. Database Engine Tuning Advisor | 88 |
| 3.8. Другие графические утилиты SQL Server 2005 | 90 |
| 3.9. Другие консольные утилиты SQL Server 2005 | 92 |
| Задание для самостоятельной работы 3.1. Работа со скриптами в SQL Server Management Studio и SQLCmd | 94 |
| Задание для самостоятельной работы 3.2. Работа с серверными сетевыми библиотеками и псевдонимами | 95 |
| Глава 4. Создание баз данных и настройка параметров | 97 |
| 4.1. Служебные и учебные базы данных SQL Server 2005 | 97 |
| 4.2. Создание пользовательских баз данных | 99 |
| 4.2.1. Создание базы данных из SQL Server Management Studio | 99 |
| 4.2.2. Создание базы данных при помощи команды <i>CREATE DATABASE</i> | 100 |
| 4.2.3. Альтернативные способы создания базы данных | 102 |
| 4.3. Файлы баз данных и журналов транзакций | 104 |
| 4.4. Применение файловых групп | 108 |
| 4.5. Режим восстановления базы данных | 110 |
| 4.6. Режимы работы базы данных | 112 |
| 4.7. Другие параметры базы данных | 114 |
| 4.8. Расширенные свойства баз данных | 119 |
| 4.9. Выполнение некоторых служебных операций с базами данных | 120 |
| 4.9.1. Увеличение размера базы данных | 121 |
| 4.9.2. Уменьшение размера базы данных | 121 |

| | |
|---|------------|
| 4.9.3. Перенос файлов базы данных | 123 |
| 4.9.4. Переименование базы данных | 124 |
| 4.9.5. Изменение владельца базы данных | 124 |
| 4.9.6. Удаление базы данных | 124 |
| 4.9.7. Проверка целостности базы данных | 125 |
| Задание для самостоятельной работы 4.1. Перенос баз данных с SQL Server 2000 на SQL Server 2005 | 125 |
| Глава 5. Безопасность SQL Server 2005 | 129 |
| 5.1. Терминология и основы системы безопасности SQL Server 2005 | 130 |
| 5.2. Логины SQL Server 2005 | 131 |
| 5.2.1. Выбор типа логина | 131 |
| 5.2.2. Создание логина и настройка его параметров | 134 |
| 5.2.3. Режимы аутентификации. Аудит попыток входа | 138 |
| 5.2.4. Логины, создаваемые по умолчанию | 140 |
| 5.2.5. Серверные роли. Разрешения на уровне сервера | 141 |
| 5.3. Пользователи баз данных | 144 |
| 5.3.1. Что такое пользователь базы данных | 144 |
| 5.3.2. Пользователь и схема | 145 |
| 5.3.3. Создание, изменение и удаление пользователей базы данных | 146 |
| 5.3.4. Встроенные пользователи базы данных | 147 |
| 5.3.5. Роли баз данных | 148 |
| 5.3.6. Предоставление прав на объекты в базе данных | 150 |
| 5.4. Роли приложений | 153 |
| 5.5. Изменение контекста выполнения. Выражение <i>EXECUTE AS</i> | 155 |
| 5.6. Применение сертификатов и шифрование данных в SQL Server 2005 | 158 |
| 5.6.1. Основы применения сертификатов и шифрования данных | 158 |
| 5.6.2. Защита сетевого трафика SQL Server 2005 | 160 |
| 5.6.3. Принципы шифрования информации в базах данных SQL Server 2005 | 168 |
| 5.6.4. Создание сертификатов и ключей и применение криптографических функций | 170 |
| Задание для самостоятельной работы 5.1. Назначение прав на объекты SQL Server 2005 и изменение контекста выполнения | 173 |
| Задание для самостоятельной работы 5.2. Шифрование информации в таблицах баз данных | 176 |
| Глава 6. Резервное копирование и восстановление баз данных SQL Server 2005 | 179 |
| 6.1. Основы резервного копирования SQL Server 2005 | 179 |
| 6.2. Планирование резервного копирования | 180 |
| 6.2.1. Лента или жесткий диск? | 180 |
| 6.2.2. Логические устройства или явно указанный путь? | 182 |
| 6.2.3. Типы резервного копирования | 183 |
| 6.2.4. Расписание резервного копирования | 186 |

| | |
|---|-----|
| 6.3. Проведение резервного копирования | 187 |
| 6.3.1. Средства для выполнения резервного копирования..... | 187 |
| 6.3.2. Параметры резервного копирования..... | 188 |
| 6.3.3. Получение информации о резервном копировании и создание отчетов | 194 |
| 6.4. Основы восстановления баз данных | 195 |
| 6.5. Подготовка к восстановлению | 197 |
| 6.6. Проведение восстановления..... | 198 |
| 6.7. Специальные ситуации восстановления..... | 203 |
| 6.7.1. Восстановление базы данных в оперативном режиме | 203 |
| 6.7.2. Восстановление отдельных страниц базы данных | 204 |
| 6.7.3. Восстановление системных баз данных..... | 205 |
| Задание для самостоятельной работы 6.1. Резервное копирование и восстановление базы данных | 207 |

Глава 7. Средства обеспечения отказоустойчивости SQL Server 2005..... 209

| | |
|---|-----|
| 7.1. Работа SQL Server 2005 в кластере | 209 |
| 7.1.1. Преимущества кластеров..... | 209 |
| 7.1.2. Терминология и варианты конфигурации кластера | 210 |
| 7.1.3. Установка SQL Server 2005 в кластер | 212 |
| 7.2. Автоматическая доставка журналов | 213 |
| 7.2.1. Что такое "автоматическая доставка журналов" | 213 |
| 7.2.2. Терминология доставки журналов | 214 |
| 7.2.3. Настройка доставки журналов | 215 |
| 7.2.4. Мониторинг доставки журналов..... | 221 |
| 7.2.5. Действия в случае сбоя основного сервера | 224 |
| 7.2.6. Как отменить доставку журналов | 226 |
| 7.3. Зеркальное отображение баз данных..... | 227 |
| 7.3.1. Что такое "зеркальное отображение баз данных" | 227 |
| 7.3.2. Терминология зеркального отображения баз данных..... | 229 |
| 7.3.3. Настройка зеркального отображения..... | 229 |
| 7.3.4. Мониторинг зеркального отображения | 232 |
| 7.3.5. Смена ролей серверов | 233 |
| 7.3.6. Приостановка и отмена зеркального отображения..... | 235 |
| Задание для самостоятельной работы 7.1. Настройка доставки журналов | 236 |

Глава 8. Автоматизация администрирования SQL Server 2005..... 241

| | |
|--|-----|
| 8.1. Автоматизация административных операций средствами SQL Server Agent | 242 |
| 8.1.1. Что такое SQL Server Agent | 242 |
| 8.1.2. Параметры настройки SQL Server Agent | 243 |
| 8.1.3. Работа с заданиями SQL Server Agent..... | 248 |
| 8.1.4. Безопасность при выполнении заданий | 257 |
| 8.1.5. Просмотр истории выполнения заданий..... | 260 |
| 8.1.6. Мультисерверные задания..... | 261 |
| 8.1.7. Работа с предупреждениями..... | 264 |
| 8.1.8. Работа с операторами..... | 270 |

| | |
|--|-----|
| 8.2. Настройка электронной почты в SQL Server 2005 | 271 |
| 8.2.1. Обзор возможностей SQL Server 2005 для работы с электронной почтой.... | 271 |
| 8.2.2. Работа с Database Mail..... | 273 |
| 8.2.3. Работа с SQLMail..... | 278 |
| 8.2.4. Альтернативные способы работы с электронной почтой SQL Server и SQL Server Agent..... | 281 |
| 8.3. Планы обслуживания баз данных | 284 |
| Задание для самостоятельной работы 8.1. Применение заданий, предупреждений и операторов | 288 |

Глава 9. Выполнение административных операций при помощи объектных моделей SMO, SQL-DMO и WMI 293

| | |
|--|-----|
| 9.1. Применение скриптов для выполнения административных операций | 293 |
| 9.2. Объектная модель SMO | 295 |
| 9.2.1. Обзор объектной модели SMO | 295 |
| 9.2.2. Общие приемы работы с объектами SMO | 298 |
| 9.2.3. Объект <i>SMO.Server</i> | 300 |
| 9.2.4. Объект <i>SMO.Database</i> | 305 |
| 9.3. Объектная модель SQL-DMO..... | 306 |
| 9.3.1. Обзор объектной модели SQL-DMO..... | 306 |
| 9.3.2. Объект <i>SQLDMO.Application</i> | 308 |
| 9.3.3. Объект <i>SQLDMO.SQLServer2</i> | 308 |
| 9.3.4. Объект <i>SQLDMO.Database2</i> | 311 |
| 9.4. WMI и SQL Server 2005 | 312 |
| 9.4.1. Что такое WMI..... | 312 |
| 9.4.2. WMI-поставщики для работы с SQL Server | 315 |
| 9.4.3. Программные средства для работы с WMI | 316 |
| 9.4.4. Подключение к службе WMI | 319 |
| 9.4.5. Язык WQL: подключение к объектам WMI | 320 |
| 9.4.6. Работа с событиями в WMI | 323 |
| 9.4.7. Объекты WMI Provider for Configuration Management | 326 |
| 9.4.8. Работа с WMI Provider for Server Events | 328 |
| Задание для самостоятельной работы 9.1. Применение объектной модели SMO | 330 |
| Задание для самостоятельной работы 9.2. Применение объектной модели SQL-DMO..... | 331 |
| Задание для самостоятельной работы 9.3. Работа с WMI Provider for Configuration Management..... | 332 |

Глава 10. Применение SQL Server Integration Services 335

| | |
|--|-----|
| 10.1. Зачем нужны SQL Server Integration Services..... | 335 |
| 10.2. Средства для работы с SSIS | 336 |
| 10.3. Применение мастера импорта и экспорта данных | 339 |
| 10.4. Пример работы с SSIS Designer | 343 |
| 10.5. Менеджеры подключений | 349 |

| | |
|---|-----|
| 10.6. Работа с Data Flow Task | 352 |
| 10.6.1. Что такое Data Flow Task | 352 |
| 10.6.2. Элементы Data Flow Task | 353 |
| 10.6.3. Источники и назначения Data Flow Task | 354 |
| 10.6.4. Преобразования Data Flow Task | 356 |
| 10.6.5. Пути и логика выполнения Data Flow Task | 364 |
| 10.7. Script Task и ActiveX Script Task | 366 |
| 10.8. Bulk Insert Task | 368 |
| 10.9. Execute SQL Task | 369 |
| 10.10. XML Task | 370 |
| 10.11. Message Queue Task | 371 |
| 10.12. Execute Package Task и Execute DTS 2000 Package Task | 374 |
| 10.13. Transfer Database Task | 375 |
| 10.14. Другие задачи копирования объектов SQL Server | 376 |
| 10.15. File System Task и FTP Task | 377 |
| 10.16. Send Mail Task | 377 |
| 10.17. Execute Process Task | 378 |
| 10.18. Web Service Task | 378 |
| 10.19. WMI Data Reader Task и WMI Event Watcher Task | 380 |
| 10.20. Задачи Analysis Services | 383 |
| 10.21. Контейнеры SSIS | 383 |
| 10.22. Задачи планов обслуживания | 386 |
| 10.23. Ограничения предшественников | 387 |
| 10.24. Протоколирование выполнения пакетов | 389 |
| 10.25. Работа с конфигурациями | 391 |
| 10.26. Хранение пакетов | 393 |
| 10.27. Безопасность пакетов SSIS | 395 |
| 10.28. Запуск пакетов SSIS на выполнение | 398 |
| Задание для самостоятельной работы 10.1. Создание пакетов SSIS для переноса данных | 401 |

Глава 11. Мониторинг и оптимизация производительности SQL Server 2005 407

| | |
|---|-----|
| 11.1. Введение в мониторинг работы и оптимизацию производительности SQL Server 2005 | 407 |
| 11.2. Мониторинг активности пользователей | 408 |
| 11.2.1. Применение Activity Monitor | 408 |
| 11.2.2. Другие средства мониторинга активности пользователей | 409 |
| 11.2.3. Работа с профилировщиком | 410 |
| Задание для самостоятельной работы 11.1. Сбор информации о запросах, выполняемых приложением | 418 |
| 11.2.4. Применение триггеров DDL | 419 |
| 11.2.5. Другие средства мониторинга активности пользователей и уведомления о событиях | 421 |
| 11.3. Журналы SQL Server 2005 | 422 |

| | |
|--|------------|
| 11.4. Мониторинг производительности SQL Server 2005 | 423 |
| 11.4.1. Основы мониторинга производительности | 423 |
| 11.4.2. Терминология и общие принципы мониторинга производительности | 424 |
| 11.4.3. Средства для мониторинга и анализа производительности | 426 |
| 11.4.4. Нагрузочное тестирование | 427 |
| 11.4.5. Приемы работы с Системным монитором | 430 |
| 11.4.6. Основы работы с объектами и счетчиками | 433 |
| 11.4.7. Счетчики для анализа загрузки процессора | 434 |
| 11.4.8. Счетчики для анализа загрузки оперативной памяти | 437 |
| 11.4.9. Счетчики для анализа производительности дисковой подсистемы | 439 |
| 11.4.10. Счетчики для анализа производительности сетевой подсистемы | 441 |
| 11.4.11. Объекты Системного монитора для мониторинга работы SQL Server 2005 | 444 |
| Задание для самостоятельной работы 11.2. Приемы работы с Системным монитором | 449 |
| 11.5. Оптимизация работы SQL Server | 452 |
| 11.5.1. Общие вопросы оптимизации работы SQL Server | 452 |
| 11.5.2. Оптимизация операционной системы для работы с SQL Server 2005 | 453 |
| 11.5.3. Общая оптимизация работы SQL Server 2005 с помощью Best Practices Analyzer | 457 |
| 11.5.4. Оптимизация подключений к SQL Server 2005 | 459 |
| 11.5.5. Оптимизация системы индексов | 462 |
| Задание для самостоятельной работы 11.3. Оптимизация системы индексов | 466 |
| 11.5.6. Дефрагментация индексов и таблиц | 467 |
| Задание для самостоятельной работы 11.4. Дефрагментация таблиц и индексов | 474 |
| 11.5.7. Работа с блокировками | 475 |
| Задание для самостоятельной работы 11.5. Управление уровнем блокировок | 479 |
| 11.5.8. Оптимизация запросов | 480 |
| Глава 12. Репликация в SQL Server 2005..... | 487 |
| 12.1. Зачем нужна репликация | 487 |
| 12.2. Новые возможности репликации SQL Server 2005 | 489 |
| 12.3. Терминология системы репликации | 490 |
| 12.4. Типы репликации | 492 |
| 12.5. Подготовка к настройке репликации | 494 |
| 12.6. Настройка репликации | 496 |
| 12.7. Средства администрирования и мониторинга репликации | 501 |
| 12.7.1. Средства администрирования репликации | 501 |
| 12.7.2. Применение Replication Monitor | 504 |
| 12.7.3. Другие средства мониторинга репликации | 507 |
| Задание для самостоятельной работы 12.1. Настройка одноранговой репликации ... | 508 |
| Предметный указатель | 513 |

Предисловие

7 ноября 2005 г. корпорация Microsoft представила новую версию своей флагманской системы управления базами данных — SQL Server 2005. С момента выхода предыдущей версии SQL Server прошло пять лет. Изменений в SQL Server 2005 по сравнению с SQL Server 2000 — множество. Появились совершенно новые возможности (например, подсистема Service Broker и встроенное шифрование), а некоторые подсистемы были практически созданы заново (DTS/SSIS). Даже специалистам с большим опытом работы с SQL Server 2000 многое приходится осваивать заново.

Автор этой книги — сертифицированный преподаватель Microsoft, который читает курсы по SQL Server уже в течение 8 лет (начиная с версии 6.5). Параллельно с преподаванием автору часто приходилось решать проблемы, связанные с SQL Server, на самых разных предприятиях. Много раз про свои найденные решения рассказывали слушатели. Таким образом, постепенно накапливался опыт решения различных проблем, множился список не всегда очевидных, но эффективных приемов.

С другой стороны, автору стал очевиден недостаток существующих источников информации для администраторов баз данных по SQL Server 2005.

Главный источник информации — это, конечно, встроенная документация по SQL Server 2005 (Books Online). Но на момент написания этой книги русской версии Books Online еще не было, что для многих специалистов, особенно в регионах, является большой проблемой.

Другой русскоязычной литературы по SQL Server 2005 на момент написания этой книги также практически не встречалось.

Еще один источник информации по SQL Server 2005 — это официальные курсы Microsoft (MOC — Microsoft Official Curriculum). Но методических материалов к ним на русском языке нет. Кроме того, у курсов Microsoft есть существенный недостаток — их маркетинговая направленность. В методических материалах по курсам активно продвигается информация о преимуществах и возможностях, и гораздо меньше говорится о проблемах SQL Server и его недостатках. Обычно администраторам приходится самостоятельно искать информацию о проблемах, столкнувшись с ними на практике.

Именно поэтому автор и решил написать эту книгу.

Автору хотелось бы сразу отметить некоторые моменты:

- эта книга посвящена только вопросам, связанным с администрированием SQL Server. Вопросы, касающиеся разработки баз данных и приложений SQL Server 2005, будут рассмотрены в другой книге, которая готовится к выходу в издательстве "БХВ-Петербург". Рабочее название книги — "Microsoft SQL Server 2005 для разработчиков";
- автор попытался сделать акцент на том, в какой именно ситуации, при возникновении какой проблемы нужно использовать то или иное средство. При этом некоторые технические подробности, не очень важные для понимания сути проблемы, опускались. Например, не приводится полное описание всех параметров, принимаемых хранимой процедурой. Зная, какую именно хранимую процедуру нужно использовать, описание параметров несложно найти в справке;
- автор постарался поместить в книгу ту информацию, которая, по опыту проведения курсов, очень востребована администраторами, но недостаточно рассматривается в официальных учебных курсах. Например, большое внимание уделяется вопросам, связанным с мониторингом и оптимизацией производительности баз данных SQL Server 2005, оптимизацией системы индексов и блокировок, автоматизацией выполнения административных операций средствами SMO и SQL-DMO, переносом и преобразованием данных средствами DTSX/SSIS, защитой информации в базах данных SQL Server 2005 и т. п.

Эту книгу можно использовать как учебное пособие для самостоятельного обучения или для проведения обучения специалистов. Для каждой главы предусмотрены задания для самостоятельной работы с подробными решениями. Эти решения можно также использовать как пошаговую инструкцию при выполнении различных операций.

Автор ждет вопросы от читателей этой книги и будет рад, если ему удастся помочь. Информация о каких-то неочевидных приемах работы с SQL Server или об ошибках в книге также будет принята с благодарностью. С автором можно связаться по электронной почте **rostislavm@askit.ru**.

Автор книги работает в учебном центре **Академии специальных курсов по информационным технологиям** (г. Санкт-Петербург). Учебный центр проводит обучение как по SQL Server 2005, так и по многим другим программным продуктам Microsoft и других производителей. Обучение производится как в учебных классах в Санкт-Петербурге, так и на территории предприятий по всей России. Информацию о некоторых учебных курсах можно найти в конце этой книги или на сайте **www.askit.ru**. Ваши вопросы по обучению направляйте по электронной почте **info@askit.ru** или по телефону **8 (812) 922 47 60**.

ГЛАВА 1



Обзор новых возможностей SQL Server 2005

В SQL Server 2005 появилось множество новых возможностей по сравнению с предыдущей версией этого программного продукта — SQL Server 2000. Задача этой главы — дать обзор новых возможностей SQL Server 2005 для специалистов, которые работали с более ранними версиями SQL Server. Подробнее большинство этих возможностей будет рассмотрено в следующих главах.

1.1. Новые возможности SQL Server 2005 для администраторов

1.1.1. Изменения в наборе административных средств

Первое, что бросается в глаза при знакомстве с SQL Server 2005, — это кардинальное изменение набора средств администрирования. В SQL Server 2005 вы не найдете ни Enterprise Manager, ни Query Analyzer, ни Service Manager, ни многих других привычных программ. Вместо них появились другие, совершенно новые средства. Конечно, многие возможности унаследованы из предыдущих версий, но добавлено и много нового. Приведем перечень главных средств администрирования SQL Server 2005.

- ❑ **SQL Server Management Studio** — главный инструмент администратора. Эта консоль администрирования, которая основана на Visual Studio, совместила в себе возможности Enterprise Manager, Query Analyzer, Analysis Manager, средств администрирования Reporting Services и Notification Services. Про SQL Server Management Studio будет рассказано в *разд. 3.1*.
- ❑ **Business Intelligence Development Studio** — еще одна консоль, основанная на Visual Studio. Она позволяет работать с проектами Analysis Services,

Integration Services (так называется новая версия Data Transformation Services, DTS), Reporting Services и Report Model. Проекты Integration Services и работа с ними средствами Business Intelligence Development Studio будут рассматриваться в *гл. 10*.

- ❑ **SQLCmd** — это утилита командной строки, призванная заменить `isql` и `osql` из предыдущих версий SQL Server. В ней появилось множество новых возможностей (например, Dedicated Administrator Connection — специальный выделенный административный режим подключения). Работа с этой утилитой будет рассматриваться в *разд. 3.4*.
- ❑ **SQL Server Configuration Management** — эта программа объединила и расширила возможности Service Manager, Server Network Utility и Client Network Utility из предыдущих версий SQL Server. Ее применение для настройки служб SQL Server, серверных и клиентских сетевых библиотек будет рассматриваться в *разд. 3.3*.
- ❑ **SQL Server Surface Area Configuration** — программное средство, в котором централизовано управление доступными возможностями SQL Server 2005 (например, из него можно включать и отключать возможность работы со сборками .NET из кода Transact-SQL, объектами для работы с электронной почтой, хранимыми процедурами автоматизации и т. п.). Про работу с этим программным средством будет рассказано в *разд. 3.5*.

1.1.2. Новые возможности для оптимизации производительности

Одной из основных целей, которые преследовала компания Microsoft при разработке SQL Server 2005, было повышение масштабируемости по сравнению с предыдущими версиями SQL Server. Надо сказать, что эта цель была во многом достигнута. Помимо общей оптимизации работы ядра базы данных, в SQL Server 2005 было реализовано множество новых средств, которые можно использовать для оптимизации производительности:

- ❑ **Возможность секционирования (*partitioning*) для таблиц и индексов.** Это очень важное нововведение позволяет производить более гибкое управление производительностью работы сервера, а также повышает удобство администрирования. Например, архивные данные в таблице (которые практически не изменяются, а обращения к ним производятся редко) можно поместить в одном разделе, а текущие данные, с которыми работают пользователи, — в другом. Второй раздел с текущими данными можно поместить на самый быстрый RAID-массив. Также можно производить различные операции по обслуживанию (например, дефрагментацию индексов) только для этого раздела.

- ❑ **Уровень изоляции моментальных снимков (*snapshot isolation*).** Этот новый план изоляции транзакций призван решить проблему с блокировками, которая доставляла специалистам немало проблем при работе с предыдущими версиями SQL Server. При использовании этого режима изоляции транзакций запросы, которые обращаются к данным только на чтение, не накладывают блокировки на записи в таблице.
- ❑ **Поддержка большого числа экземпляров SQL Server на одном компьютере.** Теперь на одном компьютере можно установить до 50 экземпляров SQL Server 2005 Enterprise Edition. Для других редакций SQL Server 2005 оставлено то же ограничение, что и для SQL Server 2000, — максимум 16 экземпляров SQL Server на компьютере. Про работу с именованными экземплярами будет рассказано в *разд. 2.2.3*.
- ❑ **Немедленная инициализация файлов (*instant file initialization*).** Это средство позволяет не заполнять пустое пространство файлов баз данных при их создании или увеличении двоичными нулями (как это было в предыдущих версиях SQL Server). В результате выполнение таких операций заметно ускоряется. Отметим только, что эта возможность доступна лишь при работе SQL Server 2005 на Windows Server 2003 или Windows XP (при работе под Windows 2000 она недоступна). Подробно про немедленную инициализацию файлов будет рассказываться в *разд. 4.3*.
- ❑ **Возможность отключения индексов.** Эта возможность может пригодиться, например, для диагностики проблем. Кроме того, перестроение отключенного некластеризованного индекса требует намного меньше места на диске. Отключение индекса производится при помощи команды `ALTER INDEX ... DISABLE` (см. *разд. 11.5.6*).
- ❑ **Ограничение количества процессоров для операций создания, изменения и удаления индексов (параметр `maxdop`).** Операции с индексами для больших таблиц могут оказаться очень ресурсоемкими. В результате работа пользователей может быть затруднена. Чтобы ограничить системные ресурсы, которые отводятся для выполнения операций с индексами, в SQL Server 2005 предусмотрен новый параметр `MAXDOP` (MAXimum Degree of Parallelism — максимальная степень распараллеливания). При помощи этого параметра можно определить максимальное количество процессоров, которые будут использоваться для выполнения операций с индексами.
- ❑ **Отложенное удаление и перестроение больших объектов.** Эта новая возможность автоматически используется SQL Server 2005 при удалении и перестроении таблиц индексов, которые занимают много места в базе данных (больше 128 экстендов, т. е. более 8 Мбайт). При этом в базе данных происходит только логическое удаление (т. е. для пользователя таблица

или индекс будут выглядеть как удаленные в обычном режиме). Физические же операции с экстендами, которые могут потребовать значительного времени, производятся в асинхронном режиме после завершения транзакции.

- ❑ **Динамические представления (*dynamic views*).** Новый тип представлений SQL Server 2005 позволяет получать информацию о различных аспектах работы SQL Server 2005. Например, информацию о текущих подключениях пользователей можно просмотреть при помощи динамического представления `sys.dm_exec_sessions`, а информацию о блокировках — `sys.dm_tran_locks`. Динамические представления будут рассматриваться в *разд. 11.2.2*.
- ❑ **Триггеры DDL (*DDL triggers*).** Эта новая возможность может использоваться для мониторинга выполнения команд DDL (Data Definition Language — язык изменения данных), т. е. команд, которые создают, изменяют или удаляют объекты в базах данных. Такие триггеры могут использоваться для аудита, дополнительных проверок и т. п. Подробнее про триггеры DDL будет рассказано в *разд. 11.2.4*.
- ❑ **Уведомления о событиях (*event notifications*).** При помощи этого средства можно отслеживать выполнение команд DDL и события трассировки (те же события SQL Server, которые видны в профилировщике). После настройки уведомлений о событиях уведомления передаются в очередь программного модуля Service Broker в виде файлов XML.
- ❑ **Асинхронное обновление статистики (параметр `AUTO_UPDATE_STATISTICS_ASYNC`).** Этот параметр предназначен для того, чтобы сделать время выполнения запроса более прогнозируемым. По умолчанию (значение параметра `FALSE`) запрос, выполняемый на SQL Server, может инициировать обновление статистики, если выяснится, что статистика, необходимая для выбора правильного плана выполнения этого запроса, устарела. Однако расчет новой статистики займет определенное время, в результате чего общее время выполнения запроса может сильно возрасти. Если мы установим для параметра базы данных `AUTO_UPDATE_STATISTICS_ASYNC` значение `TRUE`, то запрос также может инициировать обновление статистики. Но в этом случае статистика будет обновляться в асинхронном режиме, параллельно с выполнением запроса (при этом допускается, что для запроса будет выбран неоптимальный план).
- ❑ **Горячее добавление оперативной памяти (*hot-add memory*).** Если ваше оборудование поддерживает добавление оперативной памяти "на лету", то SQL Server 2005 сможет использовать такую добавленную память без необходимости перезапуска сервера. Такое решение может оказаться очень удобным для серверов, которые работают круглосуточно и остановка которых может привести к проблемам для пользователей.

- ❑ **Динамическое управление памятью при использовании AWE (Address Windowing Extensions — оконное расширение адресации).** Это средство позволяет 32-разрядным компьютерам работать с оперативной памятью размером более 4 Гбайт. В предыдущих версиях размер памяти, которую мог использовать SQL Server при применении AWE, был статическим. В SQL Server 2005 его можно изменять динамически. Про работу с AWE будет рассказано в *разд. 11.4.8*.

1.1.3. Новые возможности системы безопасности

В систему безопасности SQL Server 2005 были внесены очень большие изменения. Она стала более зрелой и функциональной. Наиболее важные новые возможности представлены далее.

- ❑ **Большое количество новых разрешений.** Эти разрешения предусмотрены для самого сервера, для баз данных и для многочисленных объектов в базах данных. Возможности предоставления разрешений были значительно расширены по сравнению с предыдущими версиями SQL Server. Про разрешения SQL Server подробно будет говориться в *гл. 5*.
- ❑ **Разделение владельцев и схемы (*separation of users and schemas*).** Эта новая и очень удобная возможность позволяет отделить именование и группировку объектов (которая производится при помощи схемы) от владения объектами. Кроме того, схему очень удобно использовать и в других ситуациях. Например, пользователю можно предоставить разрешения на схему (при этом он получит права на все объекты этой схемы), назначить ему схему по умолчанию и т. п. Про работу со схемами будет рассказываться в *разд. 5.3.2*.
- ❑ **Встроенные средства шифрования данных.** В SQL Server 2005 теперь можно шифровать данные в таблицах, используя встроенные возможности самого сервера. При этом предусмотрено множество способов шифрования. Для этой цели можно использовать сертификаты, асимметричные и симметричные ключи и просто пароли. Подробно про применение встроенных средств для шифрования данных будет рассказано в *разд. 5.6*.
- ❑ **Изменение контекста выполнения кода Transact-SQL.** Эта возможность позволяет менять контекст прямо в процессе выполнения этого кода. Для этого используется выражение `EXECUTE AS` (см. *разд. 5.5*).
- ❑ **Расширенные возможности работы с логинами SQL Server.** Например, теперь для логинов можно применять *парольные политики*, заставляя пользователей *менять пароль при следующем входе в систему* и т. п. Про работу с логинами речь пойдет в *разд. 5.2*.

1.1.4. Новые возможности для обеспечения отказоустойчивости

Одна из главных задач, которая была поставлена перед разработчиками SQL Server 2005, — это радикальное повышение отказоустойчивости SQL Server 2005 и снижение времени простоя сервера. Для решения этих проблем в SQL Server 2005 появились следующие возможности:

- **Возможность создания, изменения и удаления индексов в оперативном режиме (*online*).** Пользователи могут одновременно с выполнением этих операций работать с соответствующей таблицей. Эта возможность может оказаться очень удобной для серверов, которые должны работать в круглосуточном режиме. Она доступна только для SQL Server 2005 Enterprise Edition. Для того чтобы ей воспользоваться, предусмотрен новый параметр `ONLINE` для команд `CREATE INDEX`, `ALTER INDEX`, `DROP INDEX` и `ALTER TABLE`.
- **Поддержка большего числа узлов при работе в кластере.** Теперь количество узлов в кластере SQL Server 2005 ограничивается только возможностями операционной системы (это утверждение верно только для редакций Enterprise Edition и Developer Edition, поскольку SQL Server 2005 Standard Edition поддерживает кластер максимум из двух узлов). В предыдущих версиях SQL Server можно было использовать кластеры максимум из четырех узлов для 32-разрядных систем и максимум из восьми узлов для 64-разрядных систем. Про работу с кластерами будет рассказано в *разд. 7.1*.
- **Выделенное административное подключение (*Dedicated Administrator Connection*).** При запуске SQL Server 2005 обязательно резервируются ресурсы на одно подключение. Если на SQL Server 2005 возникли какие-то проблемы (например, некорректно написанный скрипт забрал себе все ресурсы сервера), вы можете использовать зарезервированные ресурсы для подключения в этом специальном режиме. Затем вы можете, к примеру, закрыть соединение, которое запустило этот скрипт. Выделенное административное подключение обладает приоритетом перед всеми остальными. Кроме того, подключившись в этом режиме, вы получаете дополнительные права на выполнение действий, которые при обычном подключении запрещены (например, на внесение изменений напрямую в таблицы базы данных `master`). Подробно про работу в режиме выделенного административного подключения будет рассказываться в *разд. 3.4.2*.
- **Зеркальное отображение баз данных (*database mirroring*).** Зеркальное отображение — очень важная новая возможность, которая позволяет поддерживать точную копию текущей базы данных в оперативном режиме

(online) на другом сервере, а также производить автоматическое переключение пользователей между серверами в том случае, если первый сервер вышел из строя. Возможности зеркального отображения баз данных очень похожи на возможности кластера, однако для его настройки вам не нужно никакого специального оборудования, а сами серверы, которые принимают в нем участие, могут находиться очень далеко друг от друга. К сожалению, достичь необходимого уровня зрелости этой технологии пока не удалось, поэтому Microsoft не рекомендует использовать зеркальное отображение для рабочих серверов. Кроме того, серверам, на которых настроено зеркальное отображение, отказано в поддержке со стороны соответствующих подразделений Microsoft. Подробно про зеркальное отображение баз данных будет рассказано в разд. 7.3.

- ❑ **Моментальные снимки баз данных (database snapshots).** С пользовательской точки зрения моментальные снимки представляют собой слепки базы данных по состоянию на определенный момент времени. Их можно использовать для того, чтобы вернуться к состоянию базы данных на определенный момент времени, например, на момент до начала выполнения рискованной операции (применение патчей, присланных разработчиками приложения), или для создания отчетов (на начало месяца, квартала и т. п.). Однако моментальные снимки кардинально отличаются от резервных копий баз данных, которые можно использовать для тех же целей. Изначально моментальный снимок базы данных — это просто набор пустых страниц (поэтому он создается очень быстро). При внесении любого изменения в базу данных старый вариант страницы, в которую вносится изменение, передается в распоряжение снимка, а изменения вносятся уже в новый вариант этой страницы. Для этого используются достаточно сложные возможности файловой системы NTFS. Снимки баз данных доступны только в редакции Enterprise Edition.
- ❑ **Контрольные суммы (checksums) для проверки целостности страниц базы данных.** В предыдущих версиях SQL Server для проверки целостности страниц базы данных использовались только контрольные биты. Такая технология называлась обнаружением поврежденных страниц (*torn page detection*). Она доступна и в SQL Server 2005, но кроме нее можно также использовать контрольные суммы для страниц базы данных (по умолчанию для создаваемых баз данных настроено именно использование контрольных сумм). Применение контрольных сумм позволяет более надежно обнаруживать любые повреждения в файлах данных. Подробно про использование контрольных сумм и альтернативных возможностей для проверки целостности страниц будет рассказываться в разд. 4.7 (параметр `PAGE_VERIFY`).

- ❑ **Зеркалирование носителей при резервном копировании (*mirrored backup media*).** Теперь при проведении резервного копирования можно создавать одновременно несколько зеркальных копий создаваемых файлов. Эта возможность призвана повысить надежность резервного копирования. Подробно про нее будет рассказано в *разд. 6.3.2*.
- ❑ **Открытие базы данных для доступа пользователей в фазе отката при восстановлении работоспособности экземпляра (*instance recovery*).** Теперь пользователи могут работать с базой данных еще до завершения отката всех незавершенных транзакций при восстановлении базы данных после сбоя. Эта возможность позволяет сократить время, необходимое для восстановления системы. Она доступна только в SQL Server 2005 Enterprise Edition. Подробнее с ней можно ознакомиться в *разд. 6.4*.
- ❑ **Игнорирование ошибок во время операций восстановления.** Если при восстановлении (или резервном копировании) базы данных обнаружилась какая-то ошибка, в SQL Server 2005 можно попытаться ее проигнорировать и попробовать продолжить восстановление или резервное копирование. Для этой цели предназначен новый параметр команд BACKUP и RESTORE — CONTINUE_AFTER_ERROR. Подробнее про него и про другие параметры будет рассказываться в *разд. 6.6*.
- ❑ **Восстановление открытой базы данных (*online database restore*).** Эта возможность призвана сократить время простоя при сбоях. Однако для восстановления в таком режиме предусмотрено множество ограничений, например, в любом случае файл или файловая группа, для которой проводится это восстановление, должна быть переведена в автономный режим (*offline*), нельзя производить восстановление для первого файла базы данных и т. п. Такой режим восстановления будет рассматриваться в *разд. 6.7.1*.
- ❑ **Режим EMERGENCY для базы данных.** Этот режим предназначен для проведения диагностики при подозрении на наличие каких-то проблем. База данных в этом режиме доступна только на чтение, запись в журнал транзакций не производится, обращаться к ней могут только администраторы сервера. Подробно про этот режим будет рассказано в *разд. 4.6*.

1.1.5. Новые возможности системы репликации

В системе репликации SQL Server 2005 сохранены все возможности предыдущих версий. В добавление к ним появилось большое количество новых средств, которые позволяют расширить функциональность этой системы.

- ❑ **Возможность выбора учетных записей для агентов репликации.** Если в предыдущих версиях SQL Server репликация происходила практически

только от имени учетной записи, под которой работал SQL Server Agent, то в SQL Server 2005 у вас появились богатые возможности настройки безопасности для разных агентов репликации.

- ❑ **Новые возможности работы со столбцами идентификатора (*identity columns*).** Появилась возможность реплицировать столбец *identity* именно как столбец идентификатора, а не как обычный столбец с базовым типом данных (например, *int*). Кроме того, появилась возможность распределять диапазоны значений идентификатора между серверами, которые принимают участие в репликации слиянием.
- ❑ **Трассировочные маркеры (*tracer tokens*).** Такие маркеры представляют собой небольшие объемы служебных данных, которые предназначены специально для диагностики репликации. После отправки такого диагностического пакета можно проследить его путь стандартными средствами мониторинга репликации.
- ❑ **Инициализация данных на подписчике вручную при помощи резервных копий.** Эта возможность очень удобна при настройке репликации больших таблиц. Она позволяет не забивать каналы репликации большим объемом данных.
- ❑ **Возможность изменять структуру опубликованных таблиц (добавлять и удалять столбцы).** Эти изменения будут переданы подписчику стандартными средствами репликации.
- ❑ **Возможность использовать одноранговую (*peer-to-peer*) репликацию.** При использовании этого типа репликации изменения можно вносить на любом сервере, который принимает в ней участие, эти изменения отразятся на всех остальных серверах.
- ❑ **Расширение поддержки репликации с базами данных других производителей.** Появилась возможность напрямую подписываться на публикацию в базе данных Oracle. Кроме того, можно использовать репликацию моментальных снимков и транзакционную репликацию в случаях, когда в роли издателя выступает SQL Server 2005, а в роли подписчика — Oracle или IBM DB2.
- ❑ **Новый набор программных объектов RMO (Replication Management Objects — объекты управления репликацией).** Эти программные объекты используются для автоматизации управления репликацией.
- ❑ **Разрешение конфликтов при репликации слиянием при помощи собственных программных модулей.** Такие модули можно создавать на любом .NET-совместимом языке.

Подсистема репликации SQL Server 2005 и ее новые возможности будут рассматриваться в гл. 12.

1.1.6. Новые объектные модели для автоматизации администрирования

Одной из самых удачных и удобных для администраторов новых возможностей SQL Server 7.0 и SQL Server 2000 стала объектная модель SQL-DMO (Distributed Management Objects), при помощи которой можно автоматизировать любые операции по администрированию SQL Server (при помощи программного кода на языках VBScript или JavaScript или на любом другом COM-совместимом языке программирования — Visual Basic, VBA, C++, Java, Delphi и т. п.). Поддержка объектной модели SQL-DMO сохранена и в SQL Server 2005, но, к сожалению, она не развивается. При помощи SQL-DMO нельзя обращаться к новым объектам SQL Server 2005 — только к тем, которые были и в SQL Server 2000. Причина проста: объектная модель SQL-DMO не требует поддержки .NET, а именно .NET-совместимые объектные модели Microsoft старается развивать в первую очередь. Зато в SQL Server 2005 появились новые объектные модели для автоматизации администрирования.

- ❑ **Объектная модель SMO (SQL Management Objects).** Эта объектная модель применяется для выполнения административных операций на SQL Server 2005 и призвана заменить объектную модель SQL-DMO, которая использовалась в предыдущих версиях. Отличием модели SMO является то, что она реализована средствами .NET. Объектная модель SMO будет рассматриваться в *разд. 9.2*.
- ❑ **Новые поставщики (провайдеры) WMI (Windows Management Instrumentarium).** Они обеспечивают возможности обращения к SQL Server 2005 средствами стандартного интерфейса WMI (см. *разд. 9.4*). WMI Provider for Configuration Management обеспечивает средствами для работы со службами SQL Server и серверными сетевыми библиотеками (см. *разд. 9.4.7*), а WMI Provider for Server Events предоставляет возможности для работы с уведомлениями о событиях SQL Server (*event notifications*) (см. *разд. 9.4.8*).

1.2. Новые возможности SQL Server 2005 для программистов

В этом разделе кратко перечисляются те новые возможности, которые SQL Server 2005 предоставляет для программистов. Эти возможности подробно рассмотрены в другой книге автора, которая называется "SQL Server 2005 для программистов" и готовится к выходу в издательстве "БХВ-Петербург".

- ❑ **CLR (Common Runtime Language) Integration — интеграция среды выполнения .NET.** Теперь можно обращаться к классам сборки .NET

прямо из кода Transact-SQL. Это, с одной стороны, ответ Oracle, где можно обращаться к классам Java, а с другой стороны, расширение возможностей, реализованных в предыдущих версиях SQL Server при помощи хранимых процедур автоматизации (`SP_OACreate`, `SP_OAMethod` и т. п.), которые использовались для обращения к классам COM. Обращаться к сборкам .NET очень удобно во многих ситуациях. Например, когда вы хотите использовать уже готовый программный код (для выполнения операций в файловой системе, для работы с электронной почтой и т. д.), когда вам нужно реализовать выполнение некоторых ресурсоемких операций (например, шифрование) во внешних программных модулях, когда вы хотите использовать программные конструкции обычных языков программирования и т. п. Отметим только, что для всех баз данных SQL Server 2005 CLR Integration по умолчанию отключена (как и возможность применения хранимых процедур автоматизации `SP_OACreate`, `SP_OAMethod` и др.). Проще всего для включения этих возможностей использовать консоль SQL Server Surface Area Configuration (см. разд. 3.5).

- ❑ **Возможность создавать свои собственные агрегатные функции.** В предыдущих версиях SQL Server можно было использовать только встроенные агрегатные функции (`SUM()`, `MAX()`, `MIN()`, `AVG()` и т. п.). В SQL Server 2005 можно создавать свои собственные агрегатные функции со своей программной логикой. Эти функции должны быть написаны на .NET-совместимых языках в виде классов с обязательным набором свойств и методов. Затем скомпилированную сборку нужно зарегистрировать в базе данных при помощи команды `CREATE ASSEMBLY`, а затем еще раз зарегистрировать ее в базе данных уже как пользовательскую агрегатную функцию при помощи команды `CREATE AGGREGATE`.
- ❑ **Возможность создавать свои пользовательские типы данных .NET.** В предыдущих версиях SQL Server также можно было создавать пользовательские типы данных (на основе встроенных), но здесь имеется в виду совершенно другое. В SQL Server предыдущих версий пользовательские типы данных использовались в основном как дополнительное средство проверки вводимых пользователем значений (классический пример — пользовательский тип данных для почтового индекса). В SQL Server 2005 появилась возможность создавать пользовательские типы данных в виде классов .NET со своим набором свойств и методов. Затем эти типы можно использовать для столбцов таблиц, или для переменных в коде Transact-SQL, или для передачи параметров хранимым процедурам и функциям и т. п.

Для того чтобы можно было применять пользовательские типы данных, нужно вначале создать сборку .NET с соответствующим набором свойств и методов, затем зарегистрировать ее в базе данных при помощи команды

CREATE ASSEMBLY, а затем создать пользовательский тип данных в базе данных при помощи команды CREATE TYPE;

- **Создание программных объектов (хранимых процедур, триггеров, пользовательских функций) с применением кода .NET.** С использованием средств .NET можно создавать не только агрегатные функции и пользовательские типы данных. Программный код .NET можно использовать и в хранимых процедурах, триггерах и пользовательских функциях. Реализация этой возможности похожа на работу с пользовательскими типами данных: вначале создаем сборку .NET с требуемым программным кодом и стандартным набором свойств и методов, затем регистрируем эту сборку в базе данных при помощи команды CREATE ASSEMBLY, а затем используем ее при создании программного объекта. Например, при создании хранимой процедуры с использованием кода .NET нам потребуется команда CREATE PROCEDURE ... AS EXTERNAL NAME.
- **Возможность обращения к SQL Server 2005 как к Web-службе (Native XML Web Services).** При помощи этого средства клиенты могут обращаться на SQL Server напрямую по протоколу HTTP и передавать запросы в XML-совместимом формате SOAP. Результаты запроса возвращаются также в формате SOAP/XML. При этом SQL Server выглядит как стандартная Web-служба — с доступным стандартным определением на языке WSDL (Web Services Definition Language — язык определений Web-служб), со стандартными средствами аутентификации и шифрования данных и т. п.

Преимущества обращения к SQL Server как к Web-службе очевидны:

- упрощается интеграция с приложениями и средствами третьих фирм. Web-службы — это стандарт, который поддерживается практически всеми крупными производителями средств разработки;
- протокол HTTP, который используется для обращения к Web-службам, очень удобен для работы с низкоскоростными и ненадежными соединениями, а также для выполнения асинхронных запросов. В результате резко упрощается реализация удаленных и мобильных клиентов;
- упрощается решение вопросов, связанных с безопасностью. Например, чтобы разрешить доступ к SQL Server 2005 на брандмауэрах, при использовании Web-службы потребуется открыть минимум портов. Для целей аутентификации и шифрования данных также можно использовать стандартные средства, применяемые для защиты Web-серверов.

При настройке SQL Server 2005 как Web-службы необходимо определить точки подключения по HTTP (*HTTP Endpoints*). Такие точки подключения создаются в базе данных при помощи команды CREATE ENDPOINT. Отметим

только, что для работы с ними потребуется специальный драйвер HTTP.SYS, который поставляется только с Windows Server 2003 и Windows XP SP2. Поэтому такая возможность будет недоступна для SQL Server 2005, установленного на Windows 2000.

- **Новый тип данных XML.** Он может использоваться для хранения документов XML и фрагментов документов XML (т. е. документов, для которых не предусмотрен корневой элемент) в столбцах таблиц. Кроме того, этот тип данных можно использовать для переменных Transact-SQL, для параметров хранимых процедур и функций и т. п. Его можно использовать для хранения данных в формате XML размером до 2 Гбайт.

Отметим, что для этого типа данных в SQL Server 2005 предусмотрен специальный набор методов, который позволяет, например, выполнять для документа XML запросы на языке XQuery, вносить в него изменения при помощи команд на специальном языке XML DML (XML Data Modification Language — язык изменения данных XML) и т. п.

- **Параметры для массовой загрузки данных в виде документов XML.** В предыдущих версиях SQL Server можно было определять некоторые параметры массовой загрузки данных при помощи файлов форматирования. В этих файлах можно было указать соответствия между столбцами в текстовом файле и в таблице SQL Server, выбрать только нужные столбцы и т. п. В SQL Server 2005 для указания этих параметров можно использовать файлы XML (которые можно создать вручную или сгенерировать при помощи утилиты `bcp`). Про массовую загрузку данных средствами SSIS (SQL Server Integration Services) будет рассказано в *разд. 10.8*.

- **Возможность применения конструкции `try ... catch` для обработки исключений в коде Transact-SQL.** Теперь ошибки, которые возникают в ходе выполнения команд Transact-SQL, можно перехватывать и обрабатывать при помощи тех же синтаксических конструкций, что и в .NET-совместимых языках, например, в C#. За счет этого значительно повысилось как удобство, так и надежность обработки исключений.

- **Служебные представления каталога баз данных.** В SQL Server 2005 появилось множество новых служебных представлений, предназначенных для получения информации об объектах баз данных. Частично они повторяют служебные таблицы из предыдущих версий SQL Server (сама система служебных таблиц изменилась очень сильно), например, `sys.sysobjects`, но многие из них впервые появились в SQL Server 2005.

- **Новые функции ранжирования (`RANK()`, `DENSE_RANK()`, `NTILE()` и `ROW_NUMBER()`).** Эти функции позволяют определить "ранг" для каждой записи в возвращаемом наборе по отношению к какому-нибудь значению, например, к сумме закупок для заказчика.

- ❑ **Выражение OUTPUT для команд INSERT, UPDATE и DELETE.** С его помощью можно вернуть набор записей, для которых были произведены операции вставки, изменения или удаления. Этот возвращаемый набор удобно использовать для протоколирования, дополнительных проверок, выдачи подтверждающих сообщений в приложениях и т. п.
- ❑ **Модификатор max для типов данных varchar, nvarchar и varbinary.** При использовании модификатора max эти типы данных могут использоваться для хранения информации размером до 2 Гбайт. Фактически, varchar(max) заменяет тип данных text, nvarchar(max) — ntext, а varbinary(max) — image. Традиционные типы данных text, ntext и image также оставлены, но только для обеспечения обратной совместимости. Преимуществом такого подхода является то, что с большими текстовыми и двоичными значениями теперь можно работать, как с обычными (получать их при помощи курсоров, передавать текстовые значения строковым функциям и т. п.)
- ❑ **Общие табличные выражения (Common Table Expression).** Это специальный набор записей, который определяется при выполнении запроса и может быть использован в этом запросе (например, для соединения — join). Работа с общими табличными выражениями очень похожа на работу с вложенными запросами, однако общие табличные выражения гибче. Например, к данным в общем табличном выражении можно обращаться несколько раз в рамках одного запроса.
- ❑ **Оператор APPLY.** Этот оператор позволяет вызвать функцию, возвращающую табличный набор записей, для каждой записи из первой (внешней таблицы) и передать ей значение из столбца этой таблицы. Оператор APPLY можно использовать в двух вариантах:
 - CROSS APPLY проверяет, возвращает ли эта табличная функция непустой набор значений, и если да, то записывает в возвращаемый набор результатов запись из внешней таблицы;
 - OUTER APPLY возвращает все записи из внешней таблицы (а не только те, для которых табличная функция вернула непустое значение). Но если табличная функция возвращает NULL, то в тот столбец, в который должно подставляться ее значение, также будет записываться NULL.

Например, при помощи оператора APPLY можно пройти по всем записям таблицы MyTable, для каждой записи передать значение из столбца Column1 этой таблицы функции MyFunction и получить в результате только те записи из MyTable, для которых эта функция возвращает непустое значение:

```
SELECT * FROM MyTable CROSS APPLY MyFunction(MyTable.Column1);
```

- ❑ **Операторы pivot и unpivot.** Эти операторы позволяют менять местами столбцы и строки в возвращаемых результатах, что может быть очень

удобно при создании отчетов (особенно отчетов в виде перекрестных таблиц — *cross-tabs*). В принципе то же самое можно было сделать и в предыдущих версиях SQL Server, но для этого потребовалось бы создать программный код на языке Transact-SQL. В SQL Server 2005 за счет возможности применения этих операторов выполнение такой операции значительно упростилось.

- ❑ **Уведомление об изменениях в базе данных (Query notification).** Это еще одна новая и очень удобная возможность SQL Server 2005, которая позволяет отслеживать изменения в таблицах базы данных и реагировать на эти изменения. Обычный пример использования такого уведомления выглядит так: предположим, что наше приложение должно показывать какую-то информацию из базы данных, например, список всех заказчиков с суммой продаж по каждому. Запрос, который генерирует эту информацию, является достаточно ресурсоемким, и мы заинтересованы в том, чтобы выполнять его как можно реже. Поэтому приложение, выполняя этот запрос, одновременно передает на SQL Server 2005 запрос на уведомление. Затем оно использует кэшированные данные выполненного запроса. Как только данные, которые использовались в запросе, изменятся на SQL Server, приложению сразу придет уведомление об изменении. Оно очистит кэш и выполнит запрос заново.

Для работы с уведомлениями необходимо настроить службу Service Broker для базы данных. Применения Notification Services (специального программного компонента SQL Server 2005) или уведомлений о событиях (*event notifications*) не требуется.

- ❑ **Выражение `BULK` для функции `OPENROWSET()`.** С его помощью функция `OPENROWSET()`, которая традиционно использовалась для обращения к удаленным источникам данных, теперь может применяться и для массовой загрузки данных (*bulk load*) на SQL Server. При этом можно загружать как обычный табличный набор записей, так и большие двоичные и текстовые данные. Для этой цели в SQL Server 2005 предусмотрен специальный тип поставщика массовой вставки — BULK Provider.
- ❑ **Возможность оператором `TOP` принимать выражения и использоваться в командах `INSERT`, `UPDATE` и `DELETE`.** Раньше для этого оператора можно было применять только числовые значения. Теперь ему можно передавать выражения, которые будут вычисляться в момент выполнения (например, переменные). Теперь оператор `TOP` можно использовать и в командах `INSERT`, `UPDATE` и `DELETE`, например, чтобы вставить во временную таблицу записи о десяти самых крупных заказчиках.
- ❑ **Оператор `AT` для команды `EXECUTE`.** Этот оператор позволяет выполнить хранимую процедуру на подключенном (*linked*) сервере.

- ❑ **Выражение `TABLESAMPLE`.** Это выражение позволяет вернуть только определенное количество строк в наборе записей. Оно может использоваться, например, для отладки запросов, которые в обычном режиме возвращают большое количество данных. Можно указать процент от общего числа записей, которые нужно вернуть, или явно указать количество записей. В отличие от оператора `TOP`, возвращаются случайно выбранные значения.
- ❑ **Параметры `SET NULL` и `SET DEFAULT` для каскадного обновления данных.** Теперь в выражении `REFERENCES` при создании таблицы можно указать, кроме параметра `CASCADE`, еще и эти два значения. Параметр `SET NULL` означает, что при каскадном обновлении данных для столбцов в таблицах с внешними ключами будет устанавливаться значение `NULL`, а параметр `SET DEFAULT` — значение по умолчанию.
- ❑ **Снятие ограничения на максимальный размер записи в таблице в 8060 байт.** Если превышение размера в 8060 байт возникло из-за данных в столбцах `nvarchar`, `varchar`, `varbinary` и `sql_variant`, то ошибки при вставке или изменении данных не возникает. Механизм реализации этой возможности выглядит так же, как и при работе со столбцами `text`, `ntext` и `image`: на странице, принадлежащей таблице, помещается указатель, а сами данные перемещаются на другую страницу. Такое решение может снизить производительность, но, с другой стороны, в некоторых ситуациях может оказаться очень удобным.
- ❑ **Возможность изменения плана выполнения запроса без внесения изменений в текст запроса (руководства по запросам — *plan guides*).** Эта новая возможность SQL Server 2005 очень пригодится администраторам и специалистам, которые работают с уже готовыми приложениями. Для создания таких руководств на SQL Server 2005 предусмотрена хранящая процедура `sp_create_plan_guide`. Подробнее про нее и про работу с планами выполнения запросов будет рассказываться в *разд. 11.5.7*.
- ❑ **Форсированная параметризация (*forced parameterization*).** При использовании этой возможности любое явно определенное значение, которое передается команде `SELECT`, `INSERT`, `UPDATE` или `DELETE`, будет трактоваться как параметр. В некоторых ситуациях при применении такой форсированной параметризации можно получить выигрыш в производительности за счет снижения повторных компиляций планов выполнения команд Transact-SQL.
- ❑ **Параметр `PERSISTED` для вычисляемых столбцов.** Этот параметр позволяет рассчитывать значения для таких производных столбцов не в момент выполнения запроса, а при занесении данных в таблицу. Если расчеты в вычисляемом столбце могут потребовать значительных ресурсов, то такое решение может оказаться очень удобным. В предыдущих версиях SQL

Server заранее рассчитывать значения для вычисляемых столбцов можно было только при помощи индексированных представлений.

- ❑ **Множественные активные наборы результатов MARS (Multiple Active Result Set).** Эта новая возможность, которая обеспечивается средствами SQL Native Client, в некоторых ситуациях позволяет очень сильно повысить производительность работы клиентских приложений. Ее смысл очень прост: приложение может посылать новый запрос, не дожидаясь возврата результатов предыдущего. Фактически запросы теперь можно производить в асинхронном режиме.
- ❑ **Управление блокировками для страниц индексов.** В SQL Server 2005 появилась возможность определять, как именно будут налагаться блокировки на элементы индекса: на уровне записей (`ALLOW_ROW_LOCKS`) или на уровне страниц (`ALLOW_PAGE_LOCKS`). Эти параметры определяются при создании или изменении индекса при помощи команд `CREATE INDEX` и `ALTER INDEX` соответственно. Более дробный режим блокировок (на уровне записей) обычно больше подходит для систем OLTP, а блокировки на уровне страниц — для хранилищ данных.
- ❑ **Индексы для столбцов с типом данных XML.** Сами столбцы с типом данных XML также являются новой возможностью SQL Server. Индексы для таких столбцов позволяют серьезно ускорить доступ к данным. Индексы, которые бывают двух типов — первичные и вторичные, позволяют проиндексировать имена элементов XML, пути к ним в документе XML, значения атрибутов и т. п.
- ❑ **Новые хинты оптимизатора (*optimizer hints*).** Они позволяют более точно определить план и особенности выполнения запросов. В SQL Server 2005 предусмотрено четыре новых хинта:
 - **RECOMPILE** — план выполнения такого запроса не будет использоваться повторно для аналогичных запросов. Обычно такой хинт применяется для запросов, параметры которых изменяются очень сильно, и кэшированный план может оказаться неоптимальным;
 - **OPTIMIZE FOR** — позволяет оптимизировать план выполнения под конкретное значение передаваемого параметра для запроса;
 - **USE PLAN** — предписывает запросу использовать явно определенный план (передав его явно в виде строкового значения в формате XML);
 - **PARAMETRIZATION** — позволяет переопределить для запроса режим параметризации, если тот режим, который установлен на уровне всей базы данных, для данного запроса неоптимален.

- ❑ **Библиотека SQL Server Native Client.** Этот новый программный интерфейс практически представляет собой надстройку над набором драйверов OLE DB, которая позволяет использовать новые возможности SQL Server 2005, например, множественные активные наборы результатов (Multiple Active Result Set, MARS), поддержку типа данных XML и т. п.

1.3. Новые возможности SQL Server Integration Services (DTSX)

Что появилось нового в новой версии DTS, которая в SQL Server 2005 называется SQL Server Integration Services (SSIS, используется также название DTSX), описать достаточно сложно, поскольку новой является вся подсистема. Изменился формат пакетов (теперь пакеты — это файлы в XML-совместимом формате), изменилась среда разработки пакетов (разработка пакетов производится из среды Business Intelligence Development Studio, т. е. фактически из Visual Studio.NET 2005), изменилась среда выполнения пакетов и т. п. Поэтому здесь мы отметим только некоторые новые наиболее важные возможности SSIS. Подробнее работа с SSIS будет рассматриваться в *гл. 10*.

- ❑ **Разработку пакетов SSIS теперь можно производить без подключения к SQL Server.** Фактически теперь пакеты SSIS — это специальный тип проекта Visual Studio, и разработку пакетов вполне можно производить в автономном режиме, без подключения к источникам данных.
- ❑ **Появились новые, очень мощные средства для целей отладки и протоколирования работы пакетов.** Теперь для этой цели можно использовать как стандартные средства Visual Studio, так и специализированные средства SSIS (Data Viewers — просмотрщики данных, Log Providers — набор драйверов для ведения протоколов выполнения пакетов и т. п.).
- ❑ **Появился новый набор элементов для управления логикой работы пакетов (Control Flow Tasks).** При помощи этих элементов очень удобно определять программную логику выполнения пакетов. Например, при помощи контейнера **Foreach Loop** можно выполнять определенные операции с каждым членом какой-то коллекции (например, с набором файлов в каталоге), при помощи контейнера **For Loop** — выполнять операции в цикле, пока выбранное вами условие не вернет **FALSE**, и т. п.
- ❑ **В SSIS предусмотрено большое количество новых задач.** В их число входят: Data Flow Task, Script Task, XML Task, File System Task, Web Service Task, WMI Data Reader Task, WMI Event Watcher Task и т. п. В результате функциональность пакетов SSIS значительно возросла.