

Microsoft[®] SQL Server 2005 Analysis Services

**OLAP и многомерный
анализ данных**



**Наиболее
полное
руководство**

В ПОДЛИННИКЕ[®]

Microsoft® SQL Server 2005 Analysis Services. OLAP и многомерный анализ данных

Александр Бергер
Ирина Горбач
Эдуард Меломед
Владислав Щербинин
Виталий Степаненко

Под общей редакцией
Александра Бергера,
Ирины Горбач

Санкт-Петербург
«БХВ-Петербург»

2007

УДК 681.3.06
ББК 32.973.26
Б48

Бергер, А. Б.

Б48 Microsoft® SQL Server 2005 Analysis Services. OLAP и многомерный анализ данных / Бергер А. Б., Горбач И. В., Меломед Э. Л., Шербинин В. А., Степаненко В. П. / Под общ. ред. А. Б. Бергера, И. В. Горбач. — СПб.: БХВ-Петербург, 2007. — 928 с.: ил. — (В подлиннике)
ISBN 978-5-94157-158-1

Книга, написанная разработчиками Microsoft SQL Server 2005 Analysis Services, дает читателю полное представление об его функционировании и устройстве. В ней рассмотрены основы многомерного анализа данных и дано глубокое представление о многомерных моделях данных и устройстве OLAP-сервера. Описаны основные концепции языка доступа к многомерным данным MDX и его расширенные возможности, а также архитектура сервера, методы обработки данных и алгоритмы доступа к данным. Приведены внутренние и внешние протоколы обмена данными, включая протокол XML/A. Рассмотрены алгоритмы управления ресурсами Analysis Services, в том числе алгоритмы управления памятью. Описан процесс создания эффективных клиентских приложений с использованием Analysis Services, механизмы интеграции многомерных и реляционных баз данных. Уделено внимание безопасности, а также администрированию Microsoft SQL Server 2005 Analysis Services.

Для разработчиков и бизнес-аналитиков

УДК 681.3.06
ББК 32.973.26

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Елена Кашлакова</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.01.07.

Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 74,82.

Тираж 2000 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 978-5-94157-158-1

© Бергер А. Б., Горбач И. В., Меломед Э. Л.,
Шербинин В. А., Степаненко В. П., 2007
© Оформление, издательство "БХВ-Петербург", 2007

Оглавление

Об авторах	1
Благодарности	3
Посвящение	5
Предисловие.....	7
Введение	11
Дополнительная информация	14
Часть I. ВВЕДЕНИЕ В ANALYSIS SERVICES	15
Глава 1. Что нового в Analysis Services 2005	17
Усовершенствование многомерного моделирования.....	17
Углубленная аналитика в Analysis Services 2005	19
Изменения в архитектуре "клиент-сервер"	20
Увеличение масштабируемости системы	20
Инструменты разработки и управления.....	21
Управление Analysis Services	21
База данных, используемая в книге в качестве примера	22
Данные о покупателях.....	22
Данные о магазинах.....	23
Данные о товарах и складах.....	23
Данные о времени.....	23
Данные о счетах	23
Данные о валютах	24
Данные о сотрудниках.....	24
Куб Warehouse and Sales	24
Куб HR.....	24
Куб Budget.....	25
Куб Sales and Employees	25
Глава 2. Многомерная модель данных	26
Концептуальная модель данных.....	27
Физическая модель данных	28

Прикладная модель данных.....	28
Многомерное пространство.....	28
Описание многомерного пространства	29
Резюме.....	37
Глава 3. UDM — универсальная многомерная модель данных.....	39
Резюме.....	42
Глава 4. Обзор архитектуры "клиент-сервер"	43
Двухуровневая архитектура.....	44
Одноуровневая архитектура.....	45
Трехуровневая архитектура.....	46
Четырехуровневая архитектура.....	47
Распределенные системы	48
Распределенное хранение	48
Тонкий и толстый клиенты	49
Резюме.....	50
ЧАСТЬ II. СОЗДАНИЕ МНОГОМЕРНЫХ МОДЕЛЕЙ	53
Глава 5. Концептуальная модель данных.....	55
Язык описания данных	55
Объекты в языке описания модели	56
Резюме.....	60
Глава 6. Измерения в концептуальной модели	62
Атрибуты измерения.....	63
Свойства и значения атрибутов.....	65
Связи между атрибутами.....	66
Зависимые атрибуты	66
Ключи элементов атрибута	70
Имена элементов атрибута.....	73
Связи между атрибутами.....	75
Дискретизация атрибутов.....	78
Родительские атрибуты	79
Иерархии измерений.....	80
Типы иерархий	81
Естественные иерархии и их уровни	81
Иерархии атрибутов.....	86
Иерархии родители-дети.....	88
Резюме.....	90

Глава 7. Кубы и многомерный анализ	92
Измерения куба.....	94
Атрибуты измерения куба	99
Иерархии измерений куба	100
Ролевые измерения.....	101
Куб измерения	102
Перспективы	103
Резюме.....	106
Глава 8. Меры и многомерный анализ данных	108
Меры в многомерном кубе	108
Сумма (SUM)	112
Минимальное и максимальное значения (MAX, MIN)	112
Количество (COUNT)	112
Количество уникальных значений (DISTINCT COUNT).....	113
Группы мер.....	114
Измерения группы мер	117
Гранулярность факта	117
Опосредованные измерения	124
Выражения мер	133
Подключенные группы мер.....	136
Резюме.....	136
Глава 9. Построение многомерной модели в BI Dev Studio	138
Создание источника данных.....	140
Создание и редактирование источника данных	140
Проектирование представления источника данных	141
Создание и редактирование представления источника данных	141
Проектирование измерения.....	144
Создание и редактирование измерения.....	144
Проектирование куба	149
Создание куба.....	149
Редактирование куба.....	151
Создание перспективы куба	155
Определение переводов для куба	156
Настройка и развертывание проекта для просмотра куба	157
Настройка проекта.....	157
Развертывание проекта.....	160
Просмотр куба.....	160
Резюме.....	161

Часть III. АНАЛИЗ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ MDX	163
Глава 10. Основы MDX	165
Оператор SELECT	166
Предложение SELECT	166
Определение координат в многомерном пространстве	167
Элементы по умолчанию и предложение WHERE	171
Контекст выполнения запроса	175
Алгебра множеств и основные операции с множествами	177
Объединение (Union).....	177
Пересечение (Intersect).....	178
Исключение (Except).....	179
Перекрестное соединение (CrossJoin)	179
Извлечение (Extract).....	180
Функции MDX.....	181
Функции для навигации в иерархиях	182
Функция для фильтрации множеств.....	184
Функции для упорядочивания данных.....	186
Ссылка на объекты в MDX и использование уникальных имен	188
Ссылка на объекты в MDX по имени	188
Ссылка на объекты в MDX по полному имени	188
Ссылка на объекты в MDX по уникальным именам.....	189
Резюме.....	190
Глава 11. Расширенные возможности MDX	191
Использование свойств элементов и ячеек в запросах MDX.....	191
Свойства элементов	191
Свойства ячеек	193
Неопределенные элементы.....	194
Неопределенные элементы, кортежи и пустые множества.....	194
Режим отсутствия элемента	195
Неопределенные значения и пустые ячейки	200
Преобразование типов между объектами MDX	205
Жесткие зависимости	207
Множества в предложении WHERE.....	209
Подзапросы и подкубы	212
Резюме.....	222
Глава 12. Вычисления MDX, основанные на кубе	224
Сценарии MDX.....	226
Вычисляемые элементы	227
Создание вычисляемых элементов.....	228
Свойство NON_EMPTY_BEHAVIOR.....	232

Присваивание	234
Оператор присваивания	235
Определение свойств ячеек	238
Оператор задания области действия	239
Функции Root и Leaves	242
Вычисляемые ячейки	245
Именованные множества	246
Порядок выполнения вычислений куба	250
Приоритет у наивысшей стадии	252
Разрешение рекурсии	254
Резюме	257
Глава 13. MDX-вычисления, заданные в измерении	258
Унарные операторы	258
Дополнительные формулы элементов	262
Полуаддитивные меры	265
Функция агрегирования ВуAccount	267
Порядок выполнения вычислений измерения	271
Правило приоритета ближайшего	272
Резюме	275
Глава 14. Расширение MDX с использованием хранимых процедур	277
Создание хранимых процедур	278
Создание CLR-сборок	279
Создание COM-сборок	285
Вызов хранимых процедур из MDX	286
Модель безопасности	288
Модель безопасности, основанная на ролях	289
Модель безопасности по правам доступа кода	289
Модель безопасности, основанная на идентификации пользователя	290
Объектная модель сервера	292
Операции над объектами метаданных	295
Операции над объектами MDX	297
Использование системных библиотек	299
Резюме	300
Глава 15. KPI, действия и DrillThrough	302
Ключевые индикаторы производительности	302
Определение KPI	303
Просмотр и запуск KPI	309
Действия	311
Определение действий	312
Получение информации о действиях	317

DrillThrough	323
Оператор DRILLTHROUGH.....	323
Определение столбцов DRILLTHROUGH	326
Резюме.....	329
Глава 16. Запись данных в Analysis Services	330
Оператор обратной записи	331
Обновляемые и необновляемые ячейки.....	336
Время жизни обратной записи.....	337
Разрешение обратной записи	339
Преобразование секции обратной записи в обычную секцию.....	341
Другие способы выполнения обратной записи.....	341
Резюме.....	342
ЧАСТЬ IV. ВЗАИМОДЕЙСТВИЕ МНОГОМЕРНОЙ И РЕЛЯЦИОННОЙ МОДЕЛЕЙ.....	343
Глава 17. Загрузка данных из реляционной базы данных	345
Загрузка данных	345
Источники данных	347
Свойства источника данных	348
Безопасность источника данных	349
Время ожидания выполнения запроса	352
Пуллинг соединений	352
Резюме.....	353
Глава 18. Представление источника данных и привязки объектов	354
Представление источника данных	354
Именованные запросы и именованные вычисления.....	357
Привязка объектов.....	358
Привязки столбцов.....	359
Привязка таблиц	360
Привязка к запросу.....	361
Резюме.....	362
Глава 19. Многомерные модели и схемы реляционных баз данных.....	363
Схемы реляционных хранилищ данных.....	363
Проблема синхронизации данных многомерной и реляционной модели	366
Изменения в фактах	366
Изменения в измерениях	367
Преимущества и недостатки различных схем реляционного хранилища данных.....	368

Создание реляционных схем из многомерной модели.....	369
Использование мастеров для создания реляционных схем	370
Использование шаблонов для создания реляционных схем.....	372
Резюме.....	373

ЧАСТЬ V. ЗАГРУЗКА ДАННЫХ В ANALYSIS SERVICES..... 375

Глава 20. Физическая модель данных 377

Внутренние компоненты хранения данных.....	378
Накопитель данных	378
Файловый накопитель.....	380
Битовый накопитель.....	380
Строковый накопитель.....	380
Накопитель сжатых данных.....	382
Хеш-индекс накопителя.....	383
Структура данных измерения.....	384
Структуры данных атрибута.....	384
Структуры данных иерархий	394
Физическая модель куба	400
Определение секций.....	400
Физическая модель секции.....	403
Удаленные секции	411
Обзор структур данных куба.....	411
Резюме.....	413

Глава 21. Загрузка измерений и кубов 415

Обработка измерения.....	415
Обработка атрибутов.....	415
Обработка иерархий.....	423
Создание таблиц декодирования.....	423
Построение индексов	424
Схема обработки измерения.....	425
Типы обработки измерения.....	425
Обработка ROLAP-измерений.....	428
Обработка измерений родители-дети	429
Обработка куба.....	430
Обработка данных.....	432
Создание агрегаций и построение индексов	434
Типы обработки куба.....	437
Мониторинг загрузки данных и настройка обработки ошибок	443
Свойства спецификации обработки ошибок	444
Действия при возникновении ошибок обработки	447
Резюме.....	448

Глава 22. Использование SQL Server Integration Services	450
Использование SSIS для обработки данных.....	452
Загрузка данных в Analysis Services с помощью SSIS	453
Загрузка данных в измерение	454
Загрузка данных в секции.....	456
Резюме.....	457
Глава 23. Дизайн агрегаций	458
Агрегации и набор агрегаций	458
Создание агрегаций	461
Измерения в стиле статических отчетов	462
Гибкие и жесткие агрегации.....	464
Агрегации и дизайн агрегаций	465
Алгоритм дизайна агрегаций	468
Оптимизация дизайна агрегаций	470
Конфигурация журнала запросов.....	470
Мониторинг использования агрегаций	472
Резюме.....	474
Глава 24. Управление актуальностью данных	475
Упреждающее кеширование	476
Установка времени и упреждающее кеширование	479
Частота обновлений.....	479
Длительная обработка кеша MOLAP.....	480
Сценарии упреждающего кеширования.....	481
MOLAP.....	483
MOLAP с расписанием	483
Автоматический MOLAP.....	483
MOLAP средней задержки обновления данных.....	484
MOLAP низкой задержки обновления данных	484
HOLAP реального времени.....	484
ROLAP реального времени	485
Уведомления об изменениях и порядок обработки объектов.....	485
Формирование расписания обработки	485
Типы уведомлений об изменениях	487
Получение уведомлений с помощью трассировки SQL Server.....	488
Получение уведомлений, инициированных пользователем.....	489
Получение уведомлений с помощью SQL-запросов.....	489
Рекомендации по использованию упреждающего кеширования	491
Мониторинг активности упреждающего кеширования.....	492
Резюме.....	493

Глава 25. Создание масштабируемых приложений	495
Подходы к увеличению масштабируемости.....	495
Вертикальное масштабирование	496
Горизонтальное масштабирование.....	497
OLAP-ферма.....	497
Хранилище данных.....	499
Балансировка сетевой нагрузки	501
Связанные измерения и группы мер.....	501
Обновления источника связанного объекта	502
Связанные измерения.....	503
Связанные группы мер.....	507
Удаленные секции	510
Обработка удаленных секций.....	514
Использование BI DEV Studio для создания связанных измерений	514
Использование BI Dev Studio для создания виртуального куба.....	516
Резюме.....	517
Часть VI. Архитектура ANALYSIS SERVER	519
Глава 26. Архитектура выполнения команд	521
Выполнение команд	521
Управление сессиями	525
Управление состоянием сервера	527
Выполнение команд, которые изменяют объекты Analysis Services.....	528
Создание объектов	529
Изменение объектов.....	530
Удаление объектов.....	531
Обработка объектов.....	532
Команды управления транзакциями.....	534
Синхронизация доступа к данным.....	536
Блокировка фиксирования транзакции.....	539
Прерывание выполнения команды.....	541
Пакет команд	543
Резюме.....	551
Глава 27. Управление памятью.....	552
Экономическая модель управления памятью.....	553
Управляющий памятью	554
Держатели памяти.....	554
Очистка памяти.....	556
Управление памятью системы кеша	558
Управление памятью файловых накопителей.....	559
Управление памятью пользовательских сессий.....	560

Другие держатели памяти	560
Распределители памяти	561
Диспетчер памяти	563
Модель обработки атрибутов и секций	565
Модель создания агрегаций	568
Модель построения индексов	569
Резюме	570

Глава 28. Архитектура выполнения запросов — расчет выражений MDX..... 572

Стадии выполнения запроса MDX	572
Разбор запроса MDX	575
Создание дерева виртуальных множеств	577
Оптимизация многомерного пространства удалением пустых кортежей	579
Создание областей действия вычислений	580
Глобальная область действия и кеш глобальной области действия	583
Область действия сессии и кеш областей действия сессии	584
Время существования глобальной области действия и области действия сессии	584
Вычисление значений ячеек	587
Создание планов выполнения вычисления	588
Оптимизация плана выполнения вычисления	589
Запуск и реализация плана выполнения вычисления	590
Кеши данных	591
Кеши измерений и групп мер	591
Кеши вычислений	594
Резюме	595

Глава 29. Архитектура выполнения запросов — чтение данных 597

Стадии выполнения запроса	598
Выполнение запросов к различным типам групп мер	600
Выполнение запросов к обычным группам мер	600
Выполнение запросов к ROLAP-секциям	604
Выполнение запросов к группам мер с мерами DISTINCT_COUNT	604
Выполнение запросов к удаленным секциям и связанным группам мер	608
Выполнение запросов к группам мер с опосредованными измерениями	609
Резюме	612

Часть VII. Доступ к данным, хранящимся в ANALYSIS SERVICES..... 613

Глава 30. Архитектура "клиент-сервер" и доступ к данным..... 615

Использование TCP/IP для доступа к данным	615
Использование Binary XML и сжатия для доступа к данным	617

Использование HTTP для доступа к данным.....	618
Доступ к данным в локальном кубе	620
Резюме.....	622
Глава 31. Интерфейсы доступа к данным.....	623
Использование XML/A для создания приложений	623
Использование библиотек Analysis Services для создания приложений.....	624
Создание аналитических приложений на неуправляемом коде	625
Создание аналитических приложений на управляемом коде	626
Администрирование многомерной модели с использованием DSO и AMO	628
Резюме.....	629
Глава 32. XML для анализа	630
Управление состоянием	630
Методы XML/A.....	634
Метод Discover	635
Метод Execute.....	640
Результат в формате MDDataset	642
Обработка ошибок и предупреждений.....	647
Ошибка анализа запроса	648
Ошибка формирования результата	650
Ошибка данных.....	651
Предупреждения	652
Резюме.....	653
Глава 33. ADOMD.NET.....	654
Создание аналитических приложений	654
Соединения в ADOMD.NET	656
Работа с объектами метаданных.....	666
Операции над коллекциями	667
Кеширование метаданных.....	672
Работа с коллекцией элементов	675
Работа с метаданными, которые не представлены в виде объектов	683
Объект AdomdCommand	690
Свойства ADOMDCommand.....	690
Методы ADOMDCommand.....	692
Использование объекта CellSet	698
Поддержка симметрии объектов	708
Работа с данными в табличном формате	712
Объект AdomdDataReader	715
Visual Studio для работы с многомерными данными	719
Выбор подхода для работы с данными.....	721
Использование параметров в запросах MDX.....	722

Асинхронное выполнение и прерывание команд	726
Обработка ошибок	732
AdomdErrorResponseException	733
Исключение AdomdUnknownResponseException	736
Исключение AdomdConnectionException	737
Исключение AdomdCacheExpiredException.....	737
Резюме.....	738
Глава 34. Объектная модель управления	740
Объектная модель АМО	740
Типы АМО-объектов	742
Зависимые объекты и ссылки на объекты	751
Соединение с сервером	761
Прерывание выполнения длительных операций.....	764
Загрузка объектов АМО	770
Работа с АМО в отключенном режиме	771
Использование объекта Scripter	773
Использование трассировок	776
Обработка ошибок	788
OperationException	788
Исключение ResponseFormatException.....	789
Исключение ConnectionException.....	790
Исключение OutOfSyncException.....	790
Резюме.....	791
Часть VIII. МОДЕЛЬ БЕЗОПАСНОСТИ ДАННЫХ	793
Глава 35. Модель прав доступа	795
Безопасность доступа к серверу. Аутентификация	796
Безопасность соединения TCP/IP.....	797
Безопасность соединения HTTP	797
Встроенная аутентификация Windows	798
Безопасность доступа к источникам данных.....	801
Выбор учетной записи службы Analysis Services.....	801
Настройка учетной записи для доступа к внешним данным.....	803
Изменение учетной записи службы Analysis Services.....	804
Безопасность работы именованных экземпляров	805
Безопасность работы на отказоустойчивом кластере	805
Резюме.....	806
Глава 36. Права доступа к объектам сервера	807
Роль администратора сервера	808
Роли и права доступа к данным.....	810
Безопасность управления многомерной моделью.....	813

Доступ к базе данных	814
Доступ к источникам данных	816
Доступ к измерениям	816
Доступ к кубам	816
Управление настройками безопасности	817
Резюме	818
Глава 37. Права доступа к данным измерения	819
Безопасность измерений	822
Разрешенные и запрещенные элементы	824
Визуальные итоги	829
Определения безопасности измерений	830
Тестирование безопасности измерений	832
Динамическая безопасность	833
Архитектура безопасности измерений	836
Порядок применения прав доступа	837
Резюме	839
Глава 38. Права доступа к ячейкам куба	840
Определение безопасности ячеек	840
Тестирование безопасности ячеек	844
Условная безопасность ячеек	846
Динамическая безопасность	847
Резюме	850
ЧАСТЬ IX. УПРАВЛЕНИЕ ANALYSIS SERVICES	851
Глава 39. Использование трассировки для контроля функционирования сервера	853
Архитектура трассировки	854
Создание объекта трассировки	856
SQL Server Profiler	858
Выбор параметров трассировки	858
Запуск трассировки	860
Трассировка обработки измерений и секций	863
Мониторинг прогресса выполнения обработки измерения	863
Мониторинг прогресса выполнения обработки секции	865
Трассировка обработки запросов	866
Черный ящик	869
Архитектура трассировки "черного ящика"	870
Конфигурирование работы "черного ящика"	871
Контроль состояния сервера	872
Резюме	874

Глава 40. Операции резервного копирования и восстановления	875
Резервное копирование данных	875
Планирование выполнения операций резервного копирования.....	876
Использование графического интерфейса для резервного копирования	878
Использование сценария DDL для резервного копирования	879
Выполнение резервного копирования внешней информации	880
Сценарии резервного копирования	882
Автоматизация операций резервного копирования	883
Восстановление резервных копий	884
Использование графического интерфейса для восстановления данных	885
Использование сценария DDL для восстановления резервных копий	886
Указание местоположений секций	887
Просмотр файловой системы сервера	888
Резюме.....	890
Глава 41. Стратегии развертывания проектов.....	891
Создание сценария развертывания	891
Синхронизация баз данных	892
Мастер синхронизации.....	894
Команда синхронизации баз данных.....	894
Синхронизация удаленных секций	897
Синхронизация и отказоустойчивые кластеры	900
Резюме.....	900
Предметный указатель	901

Об авторах



Александр Бергер, руководитель разработки OLAP-сервера для всех трех версий Analysis Services. Был одним из двух разработчиков, которые в 1994 г. начали разработку OLAP в компании Panorama (Израиль). После того как в 1996 г. компания Microsoft приобрела технологию OLAP, Александр переехал в Редмонд (США), где приступил к разработке первой версии OLAP-сервера. Имеет более 30 заявок и патентов в области OLAP и многомерных баз данных. Является одним из создателей спецификации OLE DB for OLAP и языка MDX.



Ирина Горбач, ведущий разработчик в группе Microsoft OLAP Server. Начиная с 1997 г., занимается разработкой различных компонентов системы Analysis Services. Принимала участие в разработке подсистемы расчетов OLAP-сервера (Formula Engine). Отвечала за клиентскую подсистему OLE DB for OLAP, ADOMD.NET, а также за интеграцию сервера с .NET Framework. Принимала участие в разработке спецификации XML for Analysis.



Эдуард Меломед, руководитель группы разработчиков OLAP-сервера. Входил в состав первоначальной группы разработчиков Analysis Services, переехавшей в Редмонд после приобретения Microsoft технологии Panorama Software Systems, Inc., которая и легла в основу технологии OLAP-сервера. Отвечает за инфраструктуру OLAP-сервера и играет важную роль в ее проектировании и разработке.



Владислав Щербинин, Microsoft Certified Professional, профессиональный разработчик и консультант по технологиям Microsoft: SQL Server, Analysis Services. Переводчик технической литературы. Член клуба The Russian SQL Server Club на сайте www.sql.ru.



Виталий Степаненко, Microsoft SQL Server MVP (Most Valuable Professional), Microsoft Certified Professional, руководитель группы разработки баз данных, Москва. Работает с Microsoft SQL Server и Analysis Services с 2000 г. Переводчик технической литературы. Член клуба The Russian SQL Server Club и ведущий новостной рассылки сайта www.sql.ru.

Благодарности

Эта книга является плодом трехлетнего труда авторов, и на протяжении всего этого времени многие оказывали помощь и поддержку в ее написании. Мы благодарим всех тех, кто способствовал выпуску книги, с начала ее написания и до того мгновения, когда ее возьмет в свои руки читатель.

В первую очередь мы хотим сказать большое спасибо нашему коллеге и другу Моше Пасуманскому. Его постоянная поддержка и глубокие знания внутреннего устройства Analysis Services оказали нам неоценимую помощь. Без Мошиных советов и его сайта www.mosha.com/msolap наша книга не была бы такой полной.

Мы хотим выразить особую благодарность редакторскому коллективу издательства "БХВ-Петербург", в частности, Игорю Шишигину — за его терпение и поддержку в течение всего проекта, и Елене Кашлаковой — за приведение нашего текста к канонам русского языка.

Эта книга не была бы возможна без всесторонней помощи наших коллег — разработчиков Analysis Services, особенно Олега Львовича, Михаила Энтина, Михаила Вовчика, Мариуса Думитру (Marius Dumitru), Акшая Мирчандари (Akshai Mirchandani), Адриана Думитраску (Adrian Dumitrascu), Балы Атур (Bala Atur) и Джеффри Ванг (Jeffrey Wang).

Массу полезных технических замечаний о содержании книги и большую помощь в проверке правильности и доступности текста для русскоязычного читателя оказал Владимир Штепа.

Мы также хотим выразить признательность Александру Гладченко и нашим коллегам из российского отделения Microsoft: Владимиру Павлову, Александру Ложечкину и Андрею Синкину, за значительную помощь в работе над технической терминологией многомерных баз данных.

Немалую роль в том, что эта книга увидела свет, сыграл форум sql.ru, мы благодарны его создателям и участникам, общение с которыми вдохновило нас на написание этой книги.

Посвящается моей семье и моим друзьям в России, Израиле и Америке.

Александр Бергер

*Моим самым любимым людям — мужу, детям,
родителям, бабушке и дедушке.
Без их поддержки я бы никогда не написала эту книгу.*

Ирина Горбач

*Моей дорогой и любимой жене Юлии за поддержку и понимание.
Моим родителям Раисе и Льву и моей сестре Миле,
чья любовь и внимание помогли мне стать тем, кто я есть.*

Эдуард Меломед

Моим дедушке и бабушке.

Владислав Щербинин

Моей матери и моему брату.

Виталий Степаненко

Предисловие

Приятно, что меня попросили написать предисловие к этой новой книге, которую я бы особенно выделил по двум причинам.

- Обычно разработчики, потратившие пять лет на разработку продукта, стремятся начать работу над новой версией, как только текущая версия готова к выпуску. Несомненно, задолго до того, как новая версия попадет к клиенту, разработчики уже работают над следующей версией. Так что для разработчиков потратить значительное время на написание такой подробной книги о продукте, как эта, очень нетипично.
- За мои годы работы аналитиком в "The OLAP Report" и намного ранее — руководителем проектов я редко встречал разработчиков, готовых предоставить такую глубокую и точную информацию о том, как на самом деле работает их продукт. Даже с соглашением о неразглашении немногие производители программного обеспечения готовы по собственной воле предоставить такой уровень внутренней информации.

Но почему это должно интересовать тех, кто не является разработчиками OLAP-серверов? Почему простой пользователь или разработчик приложений должны заботиться о том, что же на самом деле происходит под капотом? Обычному водителю ведь не надо знать точные подробности того, как работает двигатель машины.

Но существует множество причин, почему это так важно. Analysis Services на сегодняшний день несомненно наиболее широко используемый OLAP-сервер, а это само по себе означает, что большинство пользователей Analysis Services — новички в OLAP. Исследования "The OLAP Surveys" постоянно показывают, что основным аргументом при выборе системы является цена и тот факт, что система поставляется с SQL Server, а не такие факторы, как производительность, масштабируемость, простота использования или функциональность.

Это не означает, что в Analysis Services отсутствуют эти возможности, это просто означает, что обычные покупатели Analysis Services в меньшей степени заботятся о них, чем покупатели других продуктов. Но когда придет время создавать приложения, им несомненно придется принимать во внимание эти факторы, и эта книга поможет им успешно это осуществить. То, что Analysis Services рассматривается как продукт с низкой стоимостью, поставляющийся с SQL Server, не означает, что это простое небольшое дополне-

ние: особенно в версии 2005, это современный, обширный и сложный продукт. Далеко не очевидно, как он работает, и мало одних только предположений, чтобы полностью использовать его возможности.

Многие из новых пользователей Analysis Services, работавшие ранее с реляционными базами данных, могут предполагать, что базы данных OLAP аналогичны реляционным. Они заблуждаются, если только не брать во внимание кажущееся сходство между MDX и SQL. Надо действительно думать многомерно и понимать, как работают кубы Analysis Services.

И даже пользователи, имеющие опыт работы с другими серверами OLAP, обнаружат, что они отличаются друг от друга намного больше, чем реляционные базы данных. Если вы начнете использовать Analysis Services без понимания этих отличий и не зная то, как на самом деле работает Analysis Services, вы создадите себе проблемы в будущем. Даже если вам и удастся сразу получить правильные результаты, в будущем это может сказаться на производительности и сопровождаемости вашего приложения.

Исследования "The OLAP Surveys" постоянно показывают, что если что-то действительно и значит в OLAP, то это скорость выполнения запросов. Низкая производительность является самой большой причиной недовольства пользователей различных OLAP-систем, и Analysis Services не исключение. Низкая производительность запросов также была самым большим фактором, сдерживающим широкое внедрение.

Множество людей надеются, что увеличение производительности аппаратного обеспечения поможет им избавиться от проблем: если приложение работает слишком медленно, просто понадемся на следующее поколение более быстрого железа, чтобы решить проблему. Но результаты опросов "The OLAP Surveys" показывают, что такой подход не будет работать: количество жалоб на производительность запросов увеличивается с каждым годом, независимо от того, увеличивалась ли действительная производительность или нет. В нашу эру, когда все рассчитывают получать результат поиска в Web среди миллиардов документов, книг и сообщений менее чем за секунду, никто больше не хочет ждать пять или десять секунд простой отчет из небольшой внутренней базы данных. OLAP-приложению недостаточно просто быть быстрее электронной таблицы или реляционного приложения, на замену которым оно пришло — это приложение должно быть таким же быстрым, как и другие системы, которыми мы все пользуемся каждый день.

Хорошая новость состоит в том, что быстрое выполнение запросов возможно, если вы будете полностью использовать все возможности OLAP-сервера: по исследованиям "The OLAP Surveys", 57 процентов пользователей Analysis Services 2005 сообщили, что обычно их запросы выполняются менее, чем за пять секунд. Это было традиционным целевым критерием для времени выполнения запросов, но я считаю, что в новой эре мгновенного поиска в Web

это значение должно быть уменьшено до одной секунды. Это сложная цель, и для ее достижения разработчикам приложений потребуется на самом деле знать, что они делают, а также потратить время на оптимизацию их систем.

Как раз об этом эта книга. Авторы этой книги, которые работали над Analysis Services с самых первых дней, задолго до того, как эта система была названа Analysis Services — подробно описали, что на самом деле происходит внутри Analysis Services 2005, вплоть до структур записей данных. Попутно было описано множество параметров, приведена полезная информация о том, как эти параметры влияют на использование памяти и других компьютерных ресурсов.

Эта книга не предназначена для обучения новых пользователей работе с Analysis Services 2005, она для технически подготовленных специалистов, которые хотят использовать все возможности Analysis Services — за счет понимания, как оно на самом деле работает, так как эта книга написана теми, кто действительно знает это, в отличие от "внешних" авторов, которым часто приходится строить догадки. Если вы только начинаете использовать Analysis Services, вероятно, вам лучше начать с книги или курсов "как сделать?", а не с книги "что происходит внутри?", подобной этой.

Nigel Pendse (Найджел Пендз)

Редактор "The OLAP Report"

Автор "The OLAP Survey"

Введение

Разработка, которая привела к созданию Analysis Services 2005, началась в израильской фирме "Панорама" с небольшого проекта для британского издательства, которое было заинтересовано в анализе бизнеса на основе данных, накопленных в реляционной базе данных. Работа над проектом показала, что для реализации такой задачи требуется специальный подход к данным и специальные методы их обработки, которые могут оказаться полезными для большого числа задач.

Так, в конце 1994 г. два программиста приступили к проекту создания приложения для анализа данных, позволяющего предоставить руководителям предприятий средство для принятия более обоснованных решений. Приложение было выпущено в 1995 г. и успешно установлено на ряде предприятий. К концу года "Панорама" имела несколько десятков пользователей и могла позволить себе разработку следующей версии, которая должна была стать клиент-серверным приложением и состоять из сервера, хранящего данные в специальном формате, и графического клиента, позволяющего анализировать их в интерактивном режиме.

Летом 1996 г. разработку заметил один из ведущих OLAP-аналитиков, и ей заинтересовалась компания Microsoft, которая приобрела технологию к концу 1996 г. В начале 1997 г. немного разросшаяся группа разработчиков (Александр Бергер, Амир и Ариель Нетц, Моша Пасуманский и Эдвард Меломед) переехала из Тель-Авива в Редмонд и приступила к подготовке выпуска первой версии OLAP Server фирмы "Микрософт". Команда была расширена несколькими ветеранами Microsoft и новыми разработчиками (так, например, Ирина Горбач присоединилась к команде сразу после начала работы над проектом в Редмонде).

Однако стратегия, которая работала для небольшой компании, самостоятельно устанавливавшей свой продукт у каждого заказчика, не годилась для большой транснациональной корпорации. Чтобы сделать приложение доступным для установки на крупных предприятиях, потребовалось формализовать и стандартизовать протоколы обмена данными и отказаться от клиентской части с графическим интерфейсом в пользу поддержки продуктов других компаний. В начале 1997 г. группа из нескольких разработчиков (включающая Александра Бергера) собралась в гостинице на одном из островов недалеко от Сиглэ для генерации идей, которые привели к разработке интерфейсов Microsoft SQL Server Analysis Services. После долгих обсуж-

дений родилась идея создания стандарта для доступа к многомерным данным, и были заложены основы OLE DB for OLAP — расширения к стандарту обмена данных между приложением и реляционной базой данных. Важной частью стандарта стал язык MDX (MultiDimensional eXpressions) формирования запросов к многомерным данным, хранимым на OLAP Server. Основой MDX стал текстовый запрос к серверу, очень напоминающий команду Select языка SQL, только в нашем случае запрос является многомерным, запрашивающим данные из многомерного куба. Сегодня язык продолжает развиваться и меняться, но основные концепции, заложенные на острове, показали свою жизнеспособность и стали стандартом де-факто в промышленности.

Изначально предполагалось, что первая версия сервера выйдет в 1997 г. вместе с обновлением для SQL Server 6.5. Но доведение языка до состояния готовности к промышленному использованию, реализация нового типа провайдера OLE DB и обеспечения должной надежности и производительности сервера потребовало намного больше времени, чем изначально казалось разработчикам. Первая версия многомерной базы Microsoft OLAP Services была выпущена в конце 1998 г. вместе с новой версией SQL Server 7.0 (с кодовым названием "Сфинкс"). Кроме интеграции с SQL Server 7.0, OLAP Server был проинтегрирован с Excel Pivot Tables, который стал первым OLAP-клиентом для нового сервера.

Продукт оказался неожиданно хорошим для первой версии и очень скоро начал занимать существенные позиции на рынке многомерных аналитических баз данных. Появился целый ряд компаний, разработавший приложения на его базе, стандарт обмена данных, за исключением основных конкурентов, был благосклонно принят промышленностью. Несмотря на то, что набор функций первой версии был довольно ограничен, оказалось, что он покрывает достаточно большую область задач, а простота и элегантность позволили пользователям быстро освоить новый продукт. Так, под девизом "Многомерность в массы!" новая многомерная база данных фирмы Microsoft открыла рынок многомерных приложений для компаний любого уровня и сделала технологию, ранее доступную только большим корпорациям, пригодной для малого бизнеса. Как результат, рынок многомерного анализа значительно расширился, и на нем стали появляться новые незнакомые имена.

После успешного выпуска первой версии перед разработчиками встали новые задачи: начался этап подготовки к выпуску следующей версии SQL Server под кодовым названием Shiloh (Шайло). Теперь надо было понять, какую функциональность следует добавить в систему в первую очередь, чтобы получить продукт, который может стать лидирующей многомерной базой данных.

Задачу удалось сформулировать довольно легко, надо добавить такие функции, чтобы сделать OLAP-сервер платформой для решения задачи обработки данных в Хранилищах Данных (Data Warehouse). Для этого следовало ввести новые типы измерений, увеличить объемы обрабатываемых данных, расширить модель вычислений. Задача оказалась довольно сложной, особенно учитывая то, что состав разработчиков не претерпел больших изменений. Тем не менее, разработка была завершена к концу 1999 г., и на рынок вышел продукт под названием Analysis Services 2000 (AS2K). За пять лет его продажи выросли в несколько раз, и к 2004 г. AS2K занял лидирующую позицию на рынке многомерных баз данных, а в 2005 г. фирма Microsoft заняла 27% рынка OLAP-систем и оторвалась от своих конкурентов.

К 2000 г. многомерные базы данных на OLAP-серверах стали неотъемлемой частью ИТ-инфраструктуры многих компаний, и к ним стали предъявляться такие же требования по надежности и управляемости, как и к реляционным базам данных. Перед проектированием третьей версии произошел ряд изменений в структуре подразделения, занимавшегося разработкой OLAP-технологии. Группа была расширена, и была поставлена задача обеспечить дальнейшее развитие платформы и доведение ее до состояния, удовлетворяющего требованиям промышленных систем (enterprise systems). Так как изменение устаревшего кода AS2K было признано нецелесообразным, было решено начать разработку с нуля и заложить основы надежного и управляемого сервера в его базовую инфраструктуру. Тому, что из этого получилось, и посвящена эта книга.

Первая часть книги посвящена многомерной модели данных в новой версии OLAP-сервера. Мы попытались формализовать представление данных в многомерной модели и ввести необходимую терминологию и понятия. Кроме того, мы посвятили эту часть и аспектам технологии многомерной базы данных и общей архитектуре клиент-серверного приложения, воплощением которой и является Analysis Services 2005.

Следующий раздел книги посвящен описанию модели многомерных данных. Практическое понимание модели данных и языка ее описания открывает доступ ко всем функциональным возможностям Analysis Services 2005, иногда даже недоступным через графические интерфейсы.

Третья часть книги освещает основы языка MDX и дает детальный обзор его возможностей и методов его использования для доступа к многомерным данным. В этой части раскрываются методы расчетов данных в многомерной модели.

Следующие главы посвящены практическим вопросам хранения данных на сервере и методам оптимизации их хранения и доступа. Кроме того, рассмотрена архитектура сервера, практические аспекты управления и контроля его работы.

Большая часть книги посвящена методам доступа к данным клиентскими приложениями. Эти главы посвящены протоколу XML/A, объектным моделям доступа к данным ADOMD.NET и управления данными АМО.

Важный раздел книги посвящен вопросам безопасности данных и определения прав доступа к данным.

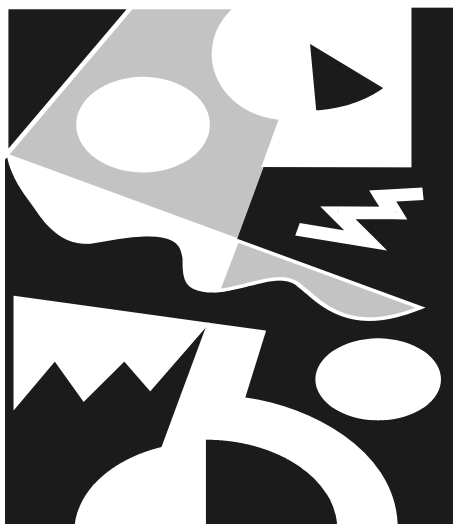
В последнем разделе раскрыты практические аспекты администрирования и мониторинга работы сервера.

При работе над этой книгой большое внимание было уделено разработке терминологии многомерных баз данных на русском языке. Это оказалось непростой задачей и вылилось во множество интересных дискуссий. Мы попытались сформировать основу русскоязычной терминологии многомерных баз данных, опираясь на термины, используемые в русифицированной версии SQL Server 2005, расширив и дополнив их.

Мы надеемся, что книга позволит вам освоить технологии многомерных баз данных и эффективно использовать Analysis Services 2005 на практике.

Дополнительная информация

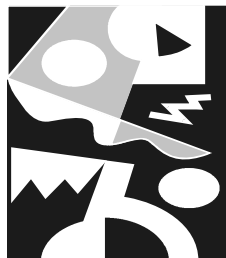
Для размещения дополнительной информации, а также для исправления возможных неточностей и ошибок в тексте книги мы разработали сайт www.e-tservice.com, посвященный этой книге. На нем вы сможете найти самую последнюю версию базы данных FoodMart 2005, используемую в книге, а также полный код примеров, приведенных в книге.



ЧАСТЬ I

ВВЕДЕНИЕ В ANALYSIS SERVICES

Глава 1



Что нового в Analysis Services 2005

Microsoft SQL Server Analysis Services является базовой платформой для развития систем бизнес-анализа (Business Intelligence). Analysis Services в версии Microsoft SQL Server 2005 предлагает совершенно новый подход к моделированию, администрированию и выполнению запросов к данным. В этой книге мы сфокусируемся на аспектах оперативной аналитической обработки данных (Online Analytical Processing, OLAP) в Analysis Services. Хотя трудно перечислить все усовершенствования, появившиеся в Analysis Services 2005, мы постараемся рассказать о наиболее важных.

В последнем разделе этой главы мы представим пример базы данных, Foodmart 2005, который мы будем использовать для демонстрации функциональности Analysis Services на страницах всей книги.

Усовершенствование многомерного моделирования

Analysis Services 2005 вводит понятие Унифицированной Многомерной Модели (Unified Dimensional Model, UDM), которая позволяет различным типам клиентских приложений получать доступ к данным как из реляционных, так и из многомерных баз данных без использования отдельных моделей для каждого типа баз данных.

Основой UDM является архитектура измерений на основе атрибутов. Архитектура измерений на основе атрибутов дает вам возможность группировать свойства (атрибуты), определяющие функционирование бизнеса, в одно измерение и отделить эти свойства от правил навигации по измерению — иерархий. В то время как измерения в Analysis Services 2000 имеют иерархи-

ческую структуру, измерения в Analysis Services 2005 основаны на атрибутах и дают возможность использовать множество иерархий.

В Analysis Services главным объектом, используемым в многомерном анализе, является куб (cube). Analysis Services 2005 поддерживает множество фактов (fact tables) в одном кубе. Меры из таблицы фактов группируются в группу мер (measure group); куб может иметь несколько групп мер. Куб с множеством групп мер имеет ту же функциональность, что и виртуальный куб в Analysis Services 2000. Группа мер аналогична кубу в предыдущих версиях.

Analysis Services 2005 поддерживает новые типы связей между измерениями и группами мер. Кроме обычных (regular) измерений, поддерживаются *опосредованные* (indirect) измерения, такие как *ссылочные* (referenced) измерения и измерения *многие-ко-многим*. Analysis Services 2005 представляет *ролевые* (role-playing) измерения, устраняя таким образом необходимость в дублировании измерений.

Унифицированная многомерная модель обеспечивает хранение различной информации, необходимой для поддержки организации. Для определения подпространства куба используются *перспективы* (perspectives). Перспективы позволяют различным группам пользователей видеть и анализировать различные части данных, хранящихся в кубе.

Унифицированная многомерная модель предоставляет возможность использовать множество источников данных (data sources) для создания многомерной модели. Analysis Services 2005 предоставляет уровень абстракции над реляционным источником данных — *представление источника данных* (data source view). Представление источника данных позволяет определять таблицы из реляционной базы для использования в многомерной модели. Кроме того, объект представления источника данных позволяет создавать вычисляемые столбцы и *представления* (view) над реляционными таблицами.

Analysis Services 2005 значительно усовершенствовал поддержку множества языков в многомерной модели. Вы можете переводить на другие языки все видимые элементы модели, такие как заголовки (captions), описания (descriptions) и др. В новой версии в значительной мере повышено удобство работы с системой в международной корпоративной среде: это дает вам возможность хранить данные на разных языках в одной и той же базе данных и использовать различные механизмы отображения в зависимости от языка.

Унифицированная многомерная модель расширяет возможности контроля над обновлением данных (data latency) с помощью механизма *упреждающего кеширования* (proactive caching). Упреждающее кеширование в Analysis Services позволяет создавать системы, которые автоматически обновляют данные на основе уведомлений об изменениях в соответствующих источниках данных.

Углубленная аналитика в Analysis Services 2005

Для доступа к данным, хранящимся в системах оперативной аналитической обработки данных (OLAP-системах), Analysis Services поддерживает язык запросов MDX (Multidimensional Expressions). Хотя в последней версии синтаксис MDX не сильно изменился, описываемый этим языком механизм расчетов был значительно модифицирован.

Механизм расчетов базируется на архитектуре измерений на основе атрибутов. Также введены изменения, упрощающие MDX и расширяющие область его применения.

Analysis Services 2005 расширяет аналитические возможности системы, предоставляя ряд новых функций агрегирования — полуаддитивных мер (semi-additive measures), а также дает возможность дизайнерам и разработчикам MDX создавать свои собственные сложные формулы вычислений.

В Analysis Services 2005 значительно упрощено определение и хранение вычислений в кубе. Теперь все вычисления в кубе задаются в одном и том же месте — сценарии MDX (MDX script). Единое место хранения для большинства вычислений упрощает разработку, улучшает визуальное восприятие и упрощает поддержку зависимостей между вычислениями.

Для упрощения разработки MDX вычислений Analysis Services 2005 предоставляет пошаговый отладчик. Также предоставляется *мастер систем бизнес-анализа* (Business Intelligence Wizard) в SQL Server Business Intelligence Development Studio (BI Dev Studio), который помогает создавать вычисления, связанные со временем, добавлять поддержку валют в кубы и добавлять множество других передовых аналитических возможностей.

Analysis Services 2005 предоставляет возможность создания бизнес-метрик, называемых Ключевыми Индикаторами Производительности (Key Performance Indicators, KPIs), которые позволяют отображать в простом понятном формате состояние организации и направление изменения этого состояния.

В Analysis Services 2005 система управления многомерной базой данных интегрирована с Общезыковой Средой Выполнения (Common Language Runtime, CLR), что позволяет создавать хранимые процедуры, используя различные языки программирования, поддерживаемые CLR. Объектная модель, предоставляемая Analysis Services 2005, позволяет программистам, пишущим процедурный код, использовать язык MDX.

Изменения в архитектуре "клиент-сервер"

В новой версии Analysis Services перешел на тонкую клиентскую архитектуру. Механизм вычислений Analysis Services 2005 полностью работает на сервере, поэтому все запросы выполняются и кешируются на сервере. Тонкий клиент не выполняет практически никаких операций кроме отправки запросов на сервер, получения и распаковки ответов.

Для коммуникации с клиентами Analysis Services 2005 использует протокол XML for Analysis (XML/A). Поэтому каждый экземпляр Analysis Services является Web-службой (Web-сервисом).

Для построения многомерной модели и управления Analysis Services представляется основанный на XML язык определения данных (Data Definition Language).

Analysis Services 2005 поставляется с тремя библиотеками для создания клиентских приложений: провайдер OLE DB for OLAP, ADO.NET (ActiveX Data Objects Multi-Dimensional) и объекты управления анализом (Analysis Management Objects, AMO).

Увеличение масштабируемости системы

Архитектура Analysis Services 2005 позволяет создавать хорошо масштабируемые приложения. Размер измерения больше не ограничен объемом памяти. Новая версия Analysis Services увеличивает масштабируемость многомерной модели, устраняя ограничения на количество многомерных объектов, таких как атрибуты, иерархии, измерения, кубы и группы мер. В этой версии также значительно увеличивается количество элементов, которое может быть загружено в измерение, а также снимается ограничение на количество дочерних элементов. Вы можете использовать Analysis Services 2005 в качестве платформы для масштабируемых решений, используя функциональность *удаленных секций* (remote partitions) и *связанных объектов* (linked objects).

Analysis Services 2005 обеспечивает эффективную поддержку многопроцессорных систем и позволяет обрабатывать множество объектов параллельно. Он также поддерживает 64-битную архитектуру, позволяющую использовать системы с очень большим объемом физической памяти. В новой версии

сервера реализована усовершенствованная система управления памятью, увеличивающая эффективность использования ресурсов.

Архитектура тонкого клиента расширяет масштабируемость систем с архитектурой промежуточных уровней (middle-tier), избавившись от неприятностей, связанных с использованием на промежуточном уровне толстого клиента.

Инструменты разработки и управления

SQL Server 2005 предоставляет новый набор инструментов для разработки приложений и управления серверами: SQL Server Business Intelligence Development Studio (BI Dev Studio) и Microsoft SQL Server Management Studio (SSMS).

Вы можете использовать BI Dev Studio для разработки полноценных систем бизнес-анализа (Business Intelligence) на основе Analysis Services, Reporting Services и Integration Services. BI Dev Studio интегрирован в оболочку Microsoft Visual Studio 2005, поддерживающую дополнительные типы проектов для Analysis Services. BI Dev Studio легко интегрируется с инструментами поддержки версий (source control systems). Это позволяет группе разработчиков баз данных совместно проектировать многомерные базы данных.

SQL Server Management Studio разработан для администраторов баз данных и позволяет управлять многомерными объектами, созданными разработчиками баз данных. SSMS позволяет администрировать Analysis Services, SQL Server, Reporting Services и Integration Services в единой консоли, которая объединяет функциональность управления, редактирования запросов и настройки производительности. Вы можете использовать SSMS как для написания запросов MDX или SQL, так и для отправки запросов XML/A в Analysis Services 2005.

Управление Analysis Services

Основанный на XML Язык Определения Данных (Data Definition Language, DDL) значительно облегчает создание сценариев и выполнение различных типов операций, включая моделирование данных, загрузку данных и управление данными. Сценарии XML (XML scripts) позволяют автоматизировать все типы операций управления системой.

Analysis Services 2005 интегрирован с приложением SQL Server Profiler и позволяет администраторам отслеживать состояние сервера и происходящих на нем процессов. Он также позволяет осуществлять контроль и мониторинг обработки и загрузки данных, анализировать выполнение запросов MDX, отслеживать пользовательские соединения и сессии. Команды, записанные в SQL Server Profiler, могут быть позже повторены на сервере, чтобы облегчить диагностику различных проблем.

Счетчики Производительности (Performance Counters) используются для сбора информации о действиях на сервере и о внутреннем состоянии сервера во время диагностики производительности системы. Вы также можете использовать специальные XML/A-запросы для определения состояния сервера.

Все серьезные проблемы, обнаруженные в процессе работы, могут быть отправлены в Microsoft с помощью технологии Dr. Watson. В случае критических ошибок в системе Analysis Services собирает дампы памяти, которые помогут группе поддержки продукта диагностировать и исправить проблему.

База данных, используемая в книге в качестве примера

В книгу включены примеры объектов многомерной базы данных и кода, созданные на основе работы одной организации, международной сети продуктовых магазинов — Food Mart. Используется FoodMart 2005 — вариация той самой базы данных, которая поставлялась с SQL Server 2000.

Food Mart продает продукты и другие товары в магазинах, расположенных в Соединенных Штатах, Канаде и Мексике. У компании есть несколько складов, из которых она доставляет товары в свои магазины. ИТ-отдел компании Food Mart собирает данные об операциях компании в базе данных SQL Server и использует Analysis Services для анализа данных.

Данные о покупателях

Информация о покупателях, их имена и адреса хранятся в таблице Customer в реляционной базе данных. В многомерной базе эта информация находится в измерении Customer. Измерение Customer с иерархией Customers позволяет анализировать данные по географическим категориям. Иерархии атрибутов Education, Gender, Marital Status, Occupation и Yearly Income предоставляют дополнительную информацию о покупателях.

Данные о магазинах

Информация об отдельных магазинах Food Mart собирается в таблице Stores и соответствующем измерении Store. Эта информация включает в себя расположение магазина, его название, Ф.И.О. менеджера, размер в квадратных футах и тип магазина, например, супермаркет или небольшой бакалейный магазин. Измерение Store имеет пользовательскую иерархию Stores, которую вы можете использовать для анализа данных по географическим категориям, и иерархии атрибутов, такие как Store Sqft, Store Type, Store Manager.

Данные о товарах и складах

Товары Food Mart сначала доставляются на один из складов, из которого они доставляются в магазины и продаются покупателям. Информация о товарах собирается в две таблицы: product_class и product. Эти таблицы формируют основу для измерения Product, которое имеет пользовательскую иерархию Products и две иерархии атрибутов — SRP и SKU.

Данные о складах хранятся в таблице warehouse, которая является основой для измерения Warehouse, имеющего одну пользовательскую иерархию Warehouses.

Данные о времени

Многомерная база данных FoodMart 2005 позволяет анализировать функционирование организации Food Mart по времени. Вся информация, относящаяся ко времени и датам, хранится в таблице time_by_date и соответствующем измерении Time. Измерение Time имеет две пользовательские иерархии: Time, которую вы можете использовать для просмотра данных по месяцам, и Time By Week, которую вы можете использовать для просмотра данных по неделям.

Данные о счетах

Вы можете производить анализ финансового состояния предприятия FoodMart 2005 на основе счетов, таких как Assets (Активы), Liabilities (Обязательства) и др. Вся информация, относящаяся к счетам, хранится в таблице account и соответствующем измерении Account. Измерение Account имеет одну иерархию типа родители-дети (parent-child) Accounts и иерархию атрибутов Account Type, предоставляющую информацию о типах счетов.

Данные о валютах

Мультинациональная корпорация, такая как предприятие Food Mart, должна иметь возможность производить операции в разных валютах. Для поддержки различных валют в хранилище данных FoodMart 2005 имеются таблица *currency* и соответствующее измерение *Currency*.

Данные о сотрудниках

Информация обо всех сотрудниках организации Food Mart хранится в трех таблицах: *employee*, *position* и *department*. Эти таблицы являются основой для измерений *Employee* и *Department*.

Для анализа производительности работы предприятия Food Mart используются четыре куба: *Warehouse and Sales*, *HR*, *Budget* и *Sales and Employees*.

Куб Warehouse and Sales

Куб *Warehouse and Sales* состоит из четырех групп мер: *Warehouse*, *Sales*, *Rates* и *Warehouse Inventory*. Группы мер *Warehouse* и *Sales* имеют по две секции (*partitions*) каждая; группы мер *Rates* и *Warehouse Inventory* имеют по одной секции.

Куб *Warehouse and Sales* может быть использован для получения информации о товарах, проданных покупателям в различных магазинах. Кроме того, можно анализировать общее количество проданных товаров и доход, полученный от продаж. А также информацию о следующем:

- какое количество товара было поставлено магазину;
- какое количество товара было заказано;
- каковы затраты определенного склада за последний квартал;
- какое количество товара осталось на складах в конце года.

Куб HR

Куб *HR* основан на таблице *salary by day*. Он включает такие измерения, как *Department*, *Time* и *Employee*.

Куб *HR* может использоваться для ответа на такие вопросы, как:

- какие уровни образования у менеджеров;
- каков средний уровень образования у сотрудников по разным странам.

Куб Budget

Куб Budget основан на таблице expense fact. Он включает такие измерения, как Currency, Account и Promotion. Этот куб включает связанную группу мер Sales, которая привязана к кубу Warehouse and Sales. Этот куб позволяет сопоставить бюджет и текущие доходы.

Куб Sales and Employees

Куб Sales and Employees основан на таблицах sales fact-1997 и sales fact-1998. Этот куб добавлен в базу данных FoodMart 2005, чтобы показать процесс задания прав доступа к данным о продажах для отдельных сотрудников организации Food Mart, используя динамическую безопасность данных.

Глава 2



Многомерная модель данных

Не так давно проекты баз данных различного типа были популярны как в академических кругах, так и в промышленности. Реляционные базы данных и системы их управления оказались абсолютными победителями и заняли доминирующие позиции в современной компьютерной индустрии.

Обеспечив простой и надежный метод накопления и обработки данных, реляционные базы данных выросли в многомиллиардную индустрию и обеспечивают работу огромного числа предприятий, а для многих предприятий являются приложениями, обеспечивающими функции, критические для деятельности.

Основной целью такой системы является надежность и точность, обеспечение постоянного, надежного и атомарного доступа к информации (Online Transaction Processing, OLTP).

Объемы данных, хранимые и обрабатываемые за один день OLTP-системой, могут достигать нескольких гигабайт в день; через некоторое время суммарный объем данных может достичь десятков и даже сотен терабайт. Такие большие объемы данных сложно хранить, но это ценный источник информации для понимания тенденций и способов функционирования предприятия или организации.

За последние несколько лет анализ большого объема данных для принятия стратегических решений и совершенствования процессов повседневного принятия решений вышел на первый план. Возникла потребность в системах, призванных обеспечить надежный и быстрый доступ к большому объему накопленной информации для ее анализа (Online Analytic Processing, OLAP). И если структуры данных и методы их организации в реляционных базах данных были оптимальны для накопления и обработки информации, они оказались далеки от оптимальных для анализа накопленной информации.

OLAP-системы спроектированы специально для выполнения анализа, их функции оптимизированы для чтения данных. За счет такого акцента на чтении OLAP превосходит по скорости своих двоюродных братьев — OLTP.

Одним из основных факторов в появлении OLAP стал новый тип структур данных — многомерные базы данных. Об этих системах организации и анализа данных мы и будем говорить в этой книге.

Analysis Services 2005 — это более полная, более гибкая реализация многомерных баз данных по сравнению с предыдущими версиями. Теперь мы можем работать с моделями практически любой сложности и значительного объема данных. Кроме того, усовершенствования в новой версии делают возможным использовать Analysis Services в качестве первичного источника данных вместо необходимости хранить копию данных в реляционной базе данных.

Дизайн и развитие многомерных баз данных, особенно Microsoft SQL Server Analysis Services 2005, системы, спроектированной и разработанной авторами этой книги, — вдохновлен успехом реляционных баз данных. Это, с одной стороны, наложило определенный отпечаток на терминологию и архитектуру системы, с другой — значительно упрощает понимание для читателей, знакомых с реляционными базами данных. Чтобы понимать все аспекты Analysis Services как системы для работы с многомерными данными, надо понимать многомерную модель данных, как эта модель определяет данные и процессы, и как система взаимодействует с другими системами хранения данных, в основном, с реляционными моделями данных. В этой главе вы найдете краткий обзор того, как данные загружаются в Analysis Services, в какой форме они хранятся и представляются пользователям.

Многомерная модель данных определяет представление данных на трех уровнях:

- концептуальной модели;
- физической модели;
- прикладной модели данных.

Концептуальная модель данных

Концептуальная модель описывает представление данных в системе и методы описания данных. Эта модель описывает данные в терминах конкретного набора данных и правила функционирования данных в условиях конкретного предприятия. В терминах этой модели пользователь описывает данные своего предприятия, их структуру, организацию, правила доступа, методы расчетов и преобразований.

В этом смысле концептуальная модель служит мостом между бизнес-моделью и многомерной моделью данных. Архитектор решений — основной пользователь концептуальной модели. Для определения концептуальной модели мы используем Язык Описания Данных (Data Definition Language, DDL) и Сценарий MDX (MDX Script).

Физическая модель данных

Как и для реляционных баз данных, физическая модель определяет то, как данные хранятся на физических носителях:

- где они хранятся, на каком диске или, возможно, в сети; в каких типах файлов хранятся данные;
- как они хранятся, в сжатом виде или нет, как данные индексируются;
- как осуществляется доступ к данным, как данные кешируются; где они могут кешироваться; как данные помещаются в память; как перемещать их из памяти и т. д.

Администратор баз данных является основным пользователем физической модели данных.

Прикладная модель данных

Прикладная модель определяет формат данных, в котором они передаются аналитическим приложениям. Основным пользователем прикладной модели данных является клиентское приложение, которое представляет модель пользователю. Прикладная модель создается с помощью Языка Многомерных Выражений (Multidimensional Expressions, MDX), который служит как для представления запросов к многомерной базе данных, так и для описания модели формирования данных внутри нее при помощи MDX-сценариев (MDX Scripts).

Многомерное пространство

При работе с реляционными базами данных используется двумерное пространство — таблица с записями (строками) и полями (столбцами). При работе с многомерными базами данных используется термин *куб* для описания многомерного пространства, хотя это не куб в геометрическом смысле слова. Геометрический куб имеет только три измерения. Многомерное про-

пространство данных может иметь любое количество измерений, и эти измерения не обязаны быть одинакового (или даже похожего) размера.

Одним из самых важных отличий между геометрическим пространством и многомерным пространством данных является то, что геометрический отрезок состоит из бесконечного числа точек, а многомерное пространство дискретно и содержит дискретное количество значений на каждом измерении.

Описание многомерного пространства

Далее приведены определения терминов, которые используются для описания многомерного пространства:

- Измерение (Dimension) — описывает элемент данных, по которому производится анализ. Например, довольно распространенным элементом анализа является время;
- Элемент (Member) — соответствует одной точке на измерении. Например, в измерении Время (Time) понедельник будет элементом измерения;
- Значение Элемента (Member Value) — это уникальная характеристика элемента. Например, в измерении Время (Time) это может быть некоторая дата;
- Атрибут (Attribute) — это полная коллекция элементов одного типа. Например, все дни недели будут атрибутом измерения Время (Time);
- Размер (Size), или Кардинальность (Cardinality) измерения — это количество элементов, которое оно содержит. Например, измерение Время (Time), состоящее из дней недели, будет иметь размер 7.

Иллюстрацию этих терминов мы начнем с описания трехмерного пространства. На рис. 2.1 показаны три измерения:

- время в месяцах (1);
- товары, описываемые их названиями (2);
- покупатели, описываемые их именами (3).

Эти три измерения используются для определения пространства продаж товара некоторым покупателям в течение периода времени, измеряемого в месяцах.

На рис. 2.1 приведена только одна продажа, показанная точкой в пространстве данных. Если каждую продажу товара представить точкой в многомерном пространстве, то все эти точки образуют фактическое пространство данных, или фактические данные.

Само собой разумеется, что количество фактических продаж намного меньше возможного числа продаж (если бы каждый товар продавался всем поку-

пателям в каждый месяц года). Конечно, это мечта любого предпринимателя, но в действительности этого не происходит.

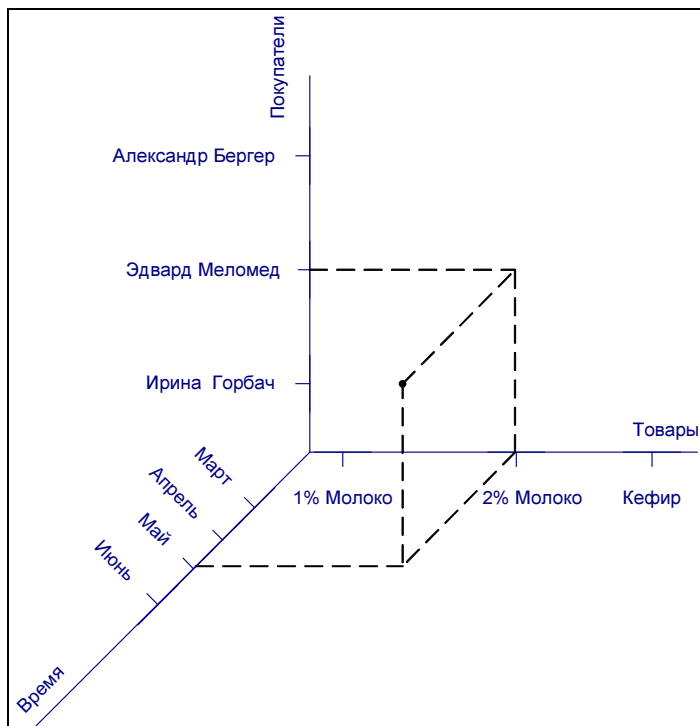


Рис. 2.1. Трехмерное пространство данных описывает продажи товаров покупателям в течение некоторого периода времени

Количество точек, соответствующее всем возможным продажам, образует теоретическое пространство данных. Размер теоретического пространства математически определяется перемножением размеров всех измерений. В случае существования большого числа измерений теоретическое пространство может оказаться очень большим, но, независимо от размера, оно остается ограниченным, так как каждое измерение является дискретным и ограниченным своим набором элементов.

В следующем списке определены еще несколько общих терминов, которые используются при описании многомерного пространства:

- Кортеж (Tuple) — это координата в многомерном пространстве;
- Срез (Slice) — это секция многомерного пространства, которая может быть определена кортежем.

Каждая точка геометрического пространства определяется набором координат, в трехмерном пространстве это X , Y и Z . Точно так же, как геометрическое пространство, многомерное пространство определяется набором координат. Этот набор называется *кортежем*. Кортеж играет важную роль в определении многомерных данных и при выполнении манипуляций над ними.

Например, точка пространства, показанная на рис. 2.1, определяется кортежем ([2% Молоко], [Эдвард Меломед], [Май]). Если элемент в одном или более измерений в кортеже заменяется звездочкой (*), которая играет роль символа подстановки и означает все элементы этого измерения, получится подпространство (фактически нормальное подпространство). Этот вид нормального подпространства называется *срезом*.

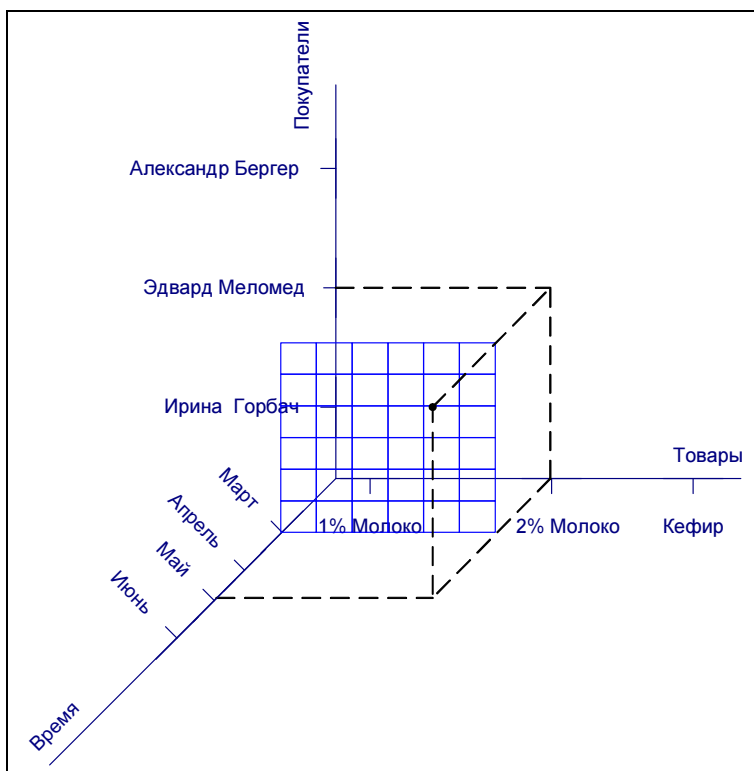


Рис. 2.2. Срез продаж марта определяется кортежем ([Март])

Примером среза для продаж всех товаров в мае всем покупателям является (*, *, [Май]). Обычно подстановочные символы в определениях кортежей не пишутся, а указываются только элементы, определяющие срез (т. е. те, ко-

торые представлены значениями), например: ([Май]). На рис. 2.2 показан срез, содержащий продажи, произведенные в марте.

Атрибуты измерений

Но как определить пространство продаж по кварталам, а не по месяцам? Если у вас существует единственный атрибут (месяцы) для вашего измерения Время, вы должны вручную сгруппировать месяцы в кварталы. Если вы рассматриваете несколько лет, ручная группировка становится неудобной.

В этом случае необходим некоторый способ визуализации месяцев, кварталов и лет (и любые другие виды деления времени, возможно, дней) в связи друг с другом — подобно шкале на линейке, которая позволяет отображать деление размера на метры, сантиметры и миллиметры.

В сущности, необходимы дополнительные атрибуты — кварталы, годы и т. д. Теперь можно использовать месяцы в качестве Ключевого Атрибута (Key Attribute) и связать другие атрибуты (зависимые атрибуты, related attributes) с месяцами — 3 месяца в квартале, 12 в году.

Итак, вернемся к примеру. Необходимо показать отдельные месяцы в каждом квартале и в каждом году. Чтобы сделать это, необходимо добавить два зависимых атрибута в измерение Время (квартал и год) и создать связь (зависимость, relationship) между этими атрибутами и ключевым атрибутом. Теперь можно создать шкалу, как показано на рис. 2.3, для измерения год-квартал-месяц.

Теперь у нас есть иерархическая структура для нашей шкалы — *иерархия измерений* (dimension hierarchy). В нашем примере иерархия измерений содержит три уровня иерархии (hierarchy levels) — годы, кварталы и месяцы. Каждый уровень соответствует атрибуту. Если вы посмотрите на рис. 2.4, вы увидите шкалу с ее иерархической структурой внутри многомерного пространства.



Рис. 2.3. Зависимые атрибуты (год, квартал) откалиброваны относительно ключевого атрибута (месяц)

Измерение может иметь более одной иерархии. Каждая иерархия, тем не менее, должна использовать один и тот же ключевой атрибут. Например, для подсчета времени в днях необходимо добавить другой атрибут — дни. Ключевым атрибутом в этом случае следует сделать день, а не месяц.

Теперь, когда ключевым атрибутом стал день, вы можете иметь две иерархии измерения: год-квартал-месяц-день и год-неделя-день.

ПРИМЕЧАНИЕ

Мы добавили атрибут день, потому что месяц не делится точно на недели. И неделю невозможно включить в иерархию год-квартал-месяц.

Ячейки

После добавления шкалы в измерение многомерного пространства вы можете видеть (на рис. 2.4), что на измерении появились новые позиции, соответствующие элементам зависимых атрибутов (квартал, год).

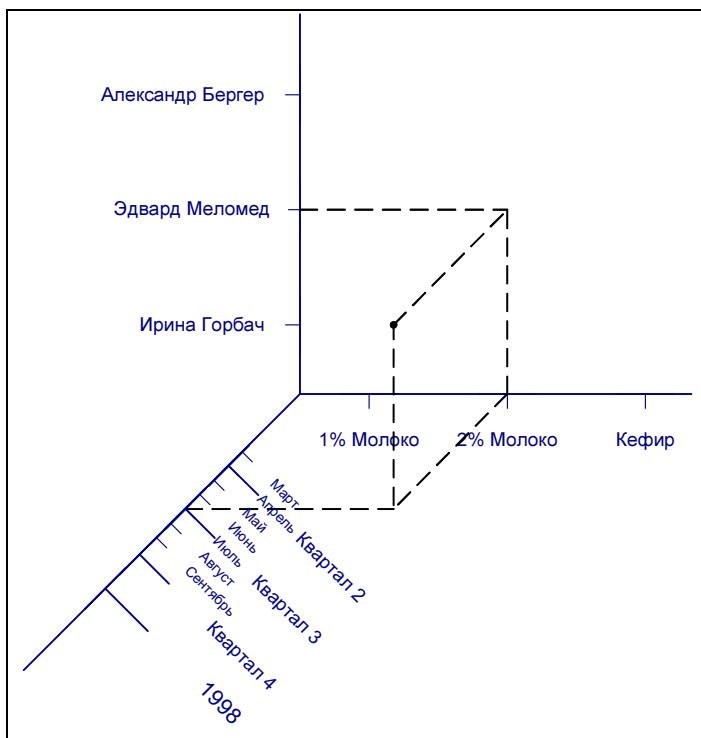


Рис. 2.4. Зависимые атрибуты создают новые точки в многомерном пространстве