

Оглавление

Об авторах.....	12
О редакторе.....	13
Предисловие от издательства.....	14
Предисловие	15
ЧАСТЬ I. ВВЕДЕНИЕ	19
Глава 1.1 DAX в бизнес-аналитике	20
Пятислойная модель бизнес-аналитики	21
Аналитика предприятия и аналитика конечных пользователей	22
Для чего используется DAX и где его найти	24
Excel.....	25
Power BI	25
SQL Server Analysis Services	25
Azure Analysis Services.....	26
Инструменты для разработки моделей и написания выражений DAX.....	26
Сделано при помощи DAX: визуальные элементы и интерактивные отчеты	27
Подходы к разработке решений.....	30
Использование Power BI для ускорения разработки аналитических решений.....	31
Мы не знаем точно, что нам нужно	32
Наши данные некорректны	33
Цикл цифровой трансформации.....	33
Заключение.....	35
Глава 1.2 Разработка модели	36
Колоночное хранение данных.....	36
Реляционные базы данных.....	36
Колоночные базы данных.....	37
Типы данных и кодирование.....	38
Связи	41
Данные в Excel	41
Реляционные базы данных.....	42
Реляционная модель Power BI	43
Свойства связей.....	44
Активные и неактивные связи.....	45
Направление фильтрации	46
Кратность.....	49

Оптимальная структура модели данных	50
Схемы «звезда» и «снежинка»	50
Проблемы, присущие схеме «звезда»	51
Концепции реляционных моделей, которых нужно избегать при моделировании в Power BI	52
Независимые измерения	52
Только одна таблица фактов	54
Хранилище данных как единый источник истины	55
Использование связей типа «многие ко многим»	56
Производительность модели и использование памяти	57
Заключение	59

Глава 1.3 Применение DAX..... 60

Вычисляемые столбцы	60
Вычисляемые таблицы	62
Меры	64
Фильтры безопасности	66
Запросы DAX	66
Таблицы дат	68
Создание таблицы дат	69
Практические советы по использованию языка DAX	71
Первым делом меры!	71
Создавайте явные меры	71
Используйте базовые меры в качестве строительных блоков	71
Прячьте от пользователей ненужные им элементы модели	72
Не смешивайте данные и меры – создайте для мер отдельную таблицу	72
Типы таблиц	74
Заключение	75

Глава 1.4 Контексты и фильтры 76

Введение в контексты DAX	77
Контекст строки	77
Контекст запроса	79
Контекст фильтра	81
Обнаружение фильтров	82
Сравнение контекста запроса и фильтра с контекстом строки	83
Фильтрация в DAX: использование функции CALCULATE	83
Шаг 1. Настройка контекста фильтра	85
Шаг 2. Удаление существующих фильтров	87
Шаг 3. Применение новых фильтров	87
Шаг 4. Вычисление выражения в новом контексте	88
Удаление фильтров при помощи функций группы ALL	90
Функции логики операций со временем	93
Изменение поведения связей	96

Табличные функции в DAX.....	98
Табличные агрегации.....	98
Использование виртуальных таблиц.....	100
Контекст в табличных функциях.....	102
Производительность при использовании табличных функций.....	105
Фильтрация при помощи табличных функций	106
Использование функции CALCULATETABLE	107
Фильтры и таблицы.....	108
Использование функции TREATAS.....	112
Переменные в DAX.....	113
Заключение.....	116

ЧАСТЬ II ПРИМЕНЕНИЕ DAX НА РЕАЛЬНЫХ ПРИМЕРАХ..... 117

Глава 2.1 Безопасность в DAX 118

Знакомство с безопасностью на уровне строк (RLS).....	118
Роли безопасности	119
Фильтры безопасности в DAX.....	121
Динамическая безопасность на уровне строк.....	122
Особенности моделирования данных с RLS.....	125
Проверка ролей безопасности.....	130
Тестирование отчетов с динамическим подключением	132
Модель для имперсонализации	133
Добавление таблицы rImpersonation в модель данных ...	134
Добавление тестовой роли безопасности.....	134
Собираем все вместе	136
Иерархическая безопасность с применением функций PATH....	137
Иерархические таблицы	137
Знакомство с функциями группы PATH	138
PATH	138
PATHCONTAINS.....	139
PATHLENGTH	139
PATHITEM.....	139
PATHITEMREVERSE	139
Использование функций PATH совместно с RLS.....	140
Продвинутая навигация по иерархии в RLS.....	140
Безопасность атрибутов.....	143
Применение безопасности атрибутов	144
Безопасность на уровне объектов и ее ограничения.....	144
Динамическая защита атрибутов: безопасность на уровне значений.....	145
Безопасность на уровне значений: моделирование данных	146
Безопасность на уровне значений: фильтры безопасности ..	149
Безопасность на уровне значений: сложные сценарии	150

Разработка моделей данных с безопасностью на уровне значений.....	153
Защита уровней агрегации.....	153
Меры не могут быть защищены, а таблицы фактов – могут.....	154
Ограничение гранулярности таблицы фактов.....	154
Защита уровней агрегации с помощью составных моделей данных.....	155
Сочетание защиты уровней агрегации с безопасностью на уровне значений.....	158
Защита уровня агрегации как атрибута.....	161
Заключение.....	165

Глава 2.2 Динамическое изменение визуализации 166

Рассматриваемый пример.....	167
Динамические меры.....	169
Базовые меры показателей.....	169
Создание вспомогательной таблицы.....	171
Создание динамической меры.....	172
Динамический выбор показателя и используемой даты.....	174
Динамические метки.....	179
Описание сценария.....	180
Создание вспомогательной таблицы.....	180
Создание меры DAX с использованием динамических меток.....	182
Комбинирование динамических меток с динамическими вычислениями.....	184
Заключение.....	186

Глава 2.3 Альтернативные календари 187

Недельный и григорианский календари.....	187
Что такое недельный календарь?.....	188
Номера недель.....	188
Периоды.....	190
Кварталы.....	190
Годы.....	191
Создание недельного календаря.....	191
Настройка дат.....	191
Определение даты начала года.....	193
Определение даты конца года.....	194
Создание дополнительных столбцов в таблице дат.....	196
Анализ с применением логики операций со временем для недельного календаря.....	199
Модель Power BI.....	199
Расчет нарастающих итогов с начала года.....	201
Расчет приростов продаж.....	203
Расчет скользящих средних продаж за отчетный год.....	207
Поддержание актуальности отчетов.....	211

Таблица Date Selection	211
Создание вариантов выбора в таблице	214
Использование вариантов выбора дат в мерах.....	217
Заключение.....	219

Глава 2.4 Использование механизма AutoExist 220

Модель данных Power BI	220
Как Power BI визуализирует данные из модели	222
Фильтры и контекст	222
Как использование мер способно изменить поведение визуального элемента.....	224
Знакомство с запросами DAX визуальных элементов.....	226
Что такое AutoExist и как он работает	229
Использование нескольких фильтров в визуальном элементе.....	229
Как AutoExist позволяет оптимизировать вычисление формул DAX	231
Пример: пропущенные рабочие дни	233
Условия задачи	234
Структура модели.....	234
Анализ сумм заказов.....	235
Расширение таблицы Calendar	238
Анализ рабочих дней	240
Куда делся мой рабочий день?	241
Как решить проблему пропущенных рабочих дней	242
Источник проблемы	243
Изменение структуры модели для обхода механизма AutoExist	244
Всегда учитывайте контекст.....	244
Исправляем ошибку с рабочими днями	248
Оптимизация производительности отчетов с AutoExist.....	250
Гранулярность в таблицах фактов.....	250
Фильтрация в нескольких таблицах фактов.....	251
Оптимизация структуры модели данных.....	254
Оптимизация визуальных элементов	255
Заклучение.....	257

Глава 2.5 Взаимодействие внутри компании 258

Моделирование процесса ведения продаж в компании QuantoVikes.....	258
Процесс ведения продаж	259
Модель данных Power BI	260
Взаимоотношения между филиалами одной компании.....	262
Представление с филиалами против сводного представления	263
Сопоставление внутренних продаж и покупок	264
Визуализация внутренних движений компании.....	268

Будущие продажи.....	273
Анализ одиночных продаж.....	273
Анализ долгосрочных продаж.....	276
Анализ старых заказов.....	278
Анализ текущих заказов.....	282
Оптимизация расчета для текущих заказов.....	288
Анализ излишков по счетам.....	290
Анализ своевременных заказов.....	294
Коррекция для запаздывающих заказов.....	296
Дальнейшая оптимизация.....	299
Тестирование сложных вычислений.....	301
Заключение.....	303

Глава 2.6 Исследуем будущее: прогнозирование и будущая стоимость..... 305

Финансовые расчеты.....	306
Приведенная стоимость и чистая приведенная стоимость....	307
Внутренняя норма доходности.....	308
Финансовые функции DAX.....	308
Бизнес-сценарий и модель данных.....	311
Подбор приемлемой ставки и индексов.....	314
Вычисление будущей стоимости.....	316
Первоначальные инвестиции и остаточная стоимость.....	316
Нерегулярные денежные потоки.....	317
Повторяющиеся денежные потоки.....	318
Положительные и отрицательные денежные потоки.....	321
Вычисление чистой приведенной стоимости.....	322
Вычисление внутренней нормы доходности.....	326
Расчет аренды, покрывающей издержки.....	328
Поиск суммы покрывающей издержки аренды с помощью аппроксимации.....	329
Оптимизация аппроксимационного подхода.....	333
Заклучение.....	338

Глава 2.7 Анализ состояния запасов..... 339

Моделирование данных, ориентированных на статусы.....	339
Гранулярность складских запасов.....	344
Базовые складские расчеты.....	346
Целевой уровень запасов.....	349
Прогнозирование складских запасов.....	355
Два типа прогнозирования.....	355
Прогнозирование продаж как инструмент предсказания изменений запасов.....	356
Экстраполяция как инструмент предсказания изменений запасов.....	360
Вычисление долгосрочных складских запасов.....	366

Использование целевого уровня запасов на основе прогноза	370
Использование линейной регрессии при экстраполяции товарных запасов.....	372
Заключение.....	376

Глава 2.8 Планирование кадровых ресурсов 377

Модель данных Power BI для планирования кадровых ресурсов	377
Расчет показателей продажи.....	380
Оптимизация расчета показателей продажи.....	383
Вычисление требуемых объемов ресурсов для реализованных проектов	385
Работа с итоговыми значениями	389
Оптимизация вычисления трудовых ресурсов	390
Оптимизация модели данных Power BI	393
Уровни агрегации.....	396
Заключение.....	397

Предметный указатель 399

Об авторах

Мишель Розема (Michiel Rozema) – один из ведущих специалистов в области Power BI из Нидерландов. Он обладает степенью магистра по математике и работает в сфере информационных технологий в роли управляющего и консультанта уже более 25 лет. В течение восьми лет Мишель возглавлял отдел аналитики в Microsoft Netherlands, запустив в стране платформу Power BI. Является автором двух книг на голландском языке, посвященных Power Pivot и Power BI. Также состоит в числе основателей официальной группы голландских пользователей Power BI и организаторов Power BI Summer School. Выступает на различных конференциях по Power BI. В 2019 году Мишель впервые стал обладателем статуса Microsoft MVP и вместе с соавтором книги Хенком Влотманом, также обладающим статусом MVP, основал компанию Quanto, специализирующуюся на консультациях в области Power BI.

Хенк Влотман (Henk Vlootman) – старший бизнес-консультант в области Power Platform, Power BI и Excel. Является обладателем статуса Microsoft MVP с 2013 года и сооснователем официальной группы голландских пользователей Power BI и Power BI Summer School. Выступает в роли спикера на многочисленных конференциях, посвященных Power BI, по всему миру. Хенк написал две книги по Excel и две по Power Pivot/Power BI. Карьеру начал в 1992 году, основав собственную компанию, после чего добился успеха в роли консультанта по Excel. В настоящее время вместе с соавтором книги Мишелем Роземой управляет компанией Quanto, специализирующейся на Power BI.

О редакторе

Грег Деклер (Greg Deckler) – обладатель статуса MVP в категории Data Platform и активный член сообщества информационных технологий в Колумбусе, штат Огайо. Является основателем пользовательской группы Columbus Azure ML and Power BI User Group (CAMLPUUG) и спикером многочисленных конференций в США. Будучи активным блогером и членом сообщества, много помогает новичкам в области Power BI. Имеет в своем активе более 180 загрузок в Power BI Quick Measures Gallery и более 5000 ответов на вопросы пользователей. Грег является заместителем председателя отдела облачных технологий в региональной компании Fusion Alliance. Его перу принадлежат три книги, посвященные Power BI: *Learn Power BI*, *DAX Cookbook* и *Power BI Cookbook, второе издание*. Также он разработал внешний инструмент для Power BI Desktop под названием *Microsoft Hates Greg's Quick Measures* и записывает видео по Power BI на своем канале в YouTube.

Я бы хотел поблагодарить моего сына, семью и все сообщество Power BI за их поддержку.

Предисловие

Прочитав эту книгу, вы сможете вывести свои аналитические навыки работы с платформой Power BI на новый качественный уровень. Вы узнаете об истинной мощи языка запросов DAX и научитесь строить на его основе эффективные решения в практической области.

Для кого эта книга

Если вы – аналитик со знанием DAX в Power BI или других аналитических инструментов Microsoft, эта книга поможет вам выйти на новый уровень владения языком DAX и начать использовать аналитические модели гораздо более эффективно.

Эта книга не для новичков – практический опыт работы с DAX вам все-таки понадобится.

Структура книги

Глава 1.1. DAX в бизнес-аналитике. В этой главе мы рассмотрим сферу бизнес-аналитики в целом и аналитические модели в современных решениях, которым отводится центральная роль в этой области. Модели Power BI идеально подходят для применения в этой среде, и не в последнюю очередь благодаря мощи языка DAX.

Глава 1.2. Разработка модели. Во второй главе книги мы поговорим о фундаментальных концепциях, лежащих в основе моделей Power BI. Вы узнаете, чем модели данных в Power BI принципиально отличаются от других систем управления данными и как выглядит оптимальная модель в этой платформе.

Глава 1.3. Применение DAX. В этой главе мы пройдемся по концепциям, реализуемым в Power BI с помощью языка DAX, в числе которых – вычисляемые столбцы и таблицы, меры, правила безопасности и запросы. Мы также дадим несколько полезных советов по работе с DAX.

Глава 1.4. Контексты и фильтры. Здесь мы поговорим о контексте строки, контексте запроса и контексте фильтра, а также обсудим, какую роль играют контексты при вычислении формул DAX. Кроме того, мы посмотрим, как можно преобразовывать контексты при помощи функции CALCULATE, добавляя и удаляя фильтры из существующих контекстов. В дополнение мы познакомимся с функциями логики операций со временем и табличными функциями DAX, узнаем о связи между таблицами и фильтрами, а также поговорим о переменных DAX.

Все эти главы будут посвящены базовым концепциям моделирования данных и языка DAX, чтобы подготовить вас к более глубокому погружению в тему.

Во второй части книги мы рассмотрим более продвинутые приемы работы с DAX на примерах из реальной практики, с которыми мы сталкивались и продолжаем сталкиваться в своей работе.

Глава 2.1. Безопасность в DAX. В этой главе мы поговорим о концепциях безопасности в моделях Power BI и роли DAX в этой области. Мы подробно обсудим механизм безопасности на уровне строк, роли безопасности, иерархии, атрибуты и уровни агрегации в процессе моделирования данных с применением DAX.

Глава 2.2. Динамическое изменение визуализации. Здесь мы рассмотрим использование таблиц-помощников и функции SWITCH для перехвата пользовательского ввода. Мы продемонстрируем богатые возможности динамического изменения привязки данных с помощью DAX с целью создания высокодинамичных визуальных элементов. В зависимости от ваших намерений таблица-помощник может содержать всего несколько строк с параметрами или большой список значений на основе других данных из модели Power BI.

Глава 2.3. Альтернативные календари. В этой главе мы поговорим о реализации логики операций со временем в случаях, когда используемый вами календарь отличается от стандартного григорианского, применение которого подразумевается по умолчанию в моделях Power BI. В конце главы мы предложим более гибкую альтернативу фильтру Power BI с относительными датами, которая будет поддерживать работу с нестандартными календарями.

Глава 2.4. Использование механизма AutoExist. Здесь мы поговорим о том, *какие именно* вычисления производятся при наполнении визуального элемента данными из модели Power BI. Понимание принципов работы механизма AutoExist позволит вам лучше разобраться в том, почему иногда данные в визуальных элементах не соответствуют вашим ожиданиям. Кроме того, вы узнаете, что именно этому полезному механизму мы обязаны быстротой формирования отчетов с визуальными элементами, в которых задействовано множество столбцов из разных таблиц.

Глава 2.5. Взаимодействие внутри компании. В этой главе обсуждаются такие распространенные бизнес-сценарии, как организация взаимодействия в рамках одной компании и консолидированные представления, а также выставление счетов по открытым продажам. Мы узнаем, как отслеживать состояние контекста, строить меры на DAX и проводить углубленный анализ.

Глава 2.6. Исследуем будущее: прогнозирование и будущая стоимость. В данной главе мы поговорим о финансовых показателях для анализа прогноза инвестиций. Мы обсудим такие распространенные метрики, как будущая стоимость, приведенная стоимость, чистая приведенная стоимость и внутренняя норма доходности, а также функции DAX для их расчета: XNPV и XIRR. Кроме того, мы познакомимся с параметрами What-if и научимся использовать их в сложных вычислениях.

Глава 2.7. Анализ состояния запасов. Здесь мы рассмотрим способы анализа складских запасов, хотя все указанные приемы могут быть с легкостью применены к анализу любых данных, связанных со статусами. Мы обсудим варианты моделирования таких данных, научимся рассчитывать состояние запасов на любой момент времени и сравнивать фактические значения с целевыми. Также мы заглянем в будущее и научимся строить прогнозы в DAX, в том числе при помощи линейной регрессии.

Глава 2.8. Планирование кадровых ресурсов. В заключительной главе книги мы поговорим об анализе потребности организации в человеческих ресурсах (выраженных в штатных единицах) при подготовке к проектам. С технической точки зрения мы на этом примере научимся работать с несколькими таблицами фактов, к которым нужно будет обращаться одновременно для получения требуемого результата. При этом нашей целью будет не только извлечение корректных цифр, но и оптимизация этого процесса.

Как извлечь максимум из книги

Как мы уже упоминали, для комфортного чтения этой книги вы должны обладать определенным практическим опытом работы с DAX и желанием развивать его и узнать что-то новое о применении этого языка запросов в сложных нестандартных сценариях. Таким образом, мы не будем подробно рассказывать о том, как работают функции DAX и какие аргументы они принимают, если речь не идет об исключительных случаях, поскольку всю эту вводную информацию можно легко найти в документации.

Все модели данных, обсуждаемые в этой книге, можно загрузить из хранилища GitHub по адресу <https://github.com/PacktPublishing/Extreme-DAX> и со страницы книги. Кроме того, в каждой главе мы будем давать ссылку на конкретный файл или файлы PBIX, с которыми будем работать. Мы рекомендуем вам параллельно с чтением разбирать указанные файлы самостоятельно.

Для открытия файлов вы можете использовать последнюю версию Power BI Desktop. Примеры всех визуализаций и запросов, показанные в этой книге, реализованы в сопроводительных файлах.

Сопроводительные файлы

Сопроводительные файлы к книге можно загрузить со страницы на сайте издательства и из хранилища GitHub по адресу <https://github.com/PacktPublishing/Extreme-DAX>.

Цветные изображения

По следующей ссылке вы можете скачать в виде PDF все рисунки и диаграммы, использованные в книге: https://static.packt-cdn.com/downloads/9781801078511_ColorImages.pdf.

Условные обозначения

На протяжении книги мы будем использовать следующие условные обозначения и шрифты.

Код в тексте: так в тексте книги мы будем выделять код на языке DAX. Пример: «Применение неактивных связей позволит вам избежать многократного включения выражения ALL('Calendar') в код DAX».

Блоки кода будут выделены следующим образом:

```
Total Sales =  
CALCULATE(  
    SUM(fProjectSales[Budget]),  
    USERELATIONSHIP(fProjectSales[StartDate], 'Calendar'[Date])  
)
```

Для привлечения внимания к важной части текста мы будем использовать курсив. Пример: «Обратите внимание, что связь между таблицами fProjectSales и Calendar мы сделали *неактивной*. Причина в том, что в большинстве вычислений мы не будем группировать проекты по дате начала, а вместо этого будем распределять результаты по целому временному интервалу, *начинающемуся* с этой даты».

Новые термины, важные слова и текст, который вы видите на экране, будут подсвечены жирным шрифтом, например: «Раздел **Workloads** содержит настройки, связанные с производительностью».



Предупреждения и важные примечания будут выводиться так.

ЧАСТЬ I

Введение

DAX в бизнес-аналитике

Нет сомнений в том, что в наше время одним из главных активов любой организации является информация. Мы прекрасно ощущаем это на себе, когда компании выстраиваются в очередь в погоне за нашими личными данными. И это происходит не от того, что мы им интересны как личности (хотя, разумеется, среди нас есть и выдающиеся люди), просто в совокупности данные о нас всех позволяют компаниям делать важные выводы и тем самым повышать эффективность своего бизнеса.

И это верно не только применительно к коммерческим структурам. Ответственные организации, клиники и университеты также активно ведут сбор информации в целях улучшения качества своей деятельности. Так или иначе, сегодня именно информация является двигателем прогресса и инноваций.

В то же время процесс преобразования исходных данных в информацию, а затем в аналитические сведения может быть весьма сложным и утомительным. Он включает в себя сбор данных из самых разных источников, исследование скрытых структур и зависимостей в них с учетом требуемого контекста. Именно поэтому задачи по анализу данных традиционно отдаются на откуп отделам информационных технологий, хотя это далеко не всегда бывает оптимально, поскольку речь зачастую идет о *бизнес-аналитике*, и знание специфики ведения бизнеса здесь крайне важно.

Эта книга посвящена одному из мощнейших средств аналитики на сегодняшний день – **DAX** (Data Analysis eXpressions – выражения для анализа данных). Мы предполагаем, что у вас уже есть определенный опыт работы с этим языком запросов и вы хотите улучшить свои навыки. А для этого очень важно понимать, для чего DAX подходит идеально, а для чего – нет. Кроме того, вы должны выработать привычку использовать DAX везде, где он может проявить себя наилучшим образом.

В этой главе мы рассмотрим основные концепции, которые помогут вам заложить эти основы. Темы, которые будут рассмотрены в этой главе:

- пятислойная модель бизнес-аналитики;
- аналитика предприятия и аналитика конечных пользователей;
- для чего используется DAX и где его найти;
- инструменты для разработки моделей и написания выражений DAX;
- сделано при помощи DAX: визуальные элементы и интерактивные отчеты;
- подходы к разработке решений;
- цикл цифровой трансформации.

Пятислойная модель бизнес-аналитики

Чтобы можно было обсуждать аналитику структурированно и всеобъемлюще, мы разработали простую схему, описывающую основные компоненты и ответственности в рамках аналитического решения. И имя для схемы, представленной на рис. 1.1.1, выбрано тоже простое – *пятислойная модель бизнес-аналитики*.



Рис. 1.1.1. Пятислойная модель бизнес-аналитики

Первый и нижний слой – **Подключение** – знаменует собой отправную точку для аналитики: если вам нужно проанализировать данные, для начала их необходимо как-то получить. Источниками при этом могут быть файлы Excel, текстовые файлы, полноценные базы данных или серверы в интернете.

Обычно исходные данные поставляются не в самом лучшем виде для анализа, особенно если они происходят из разных источников. И для исправления этой неприятности служит второй слой на нашей схеме – **Подготовка**. Этот важный процесс может включать в себя различные действия, такие как преобразование типов, трансформация данных, восстановление исторических сведений или их сопоставление на основе ключевых атрибутов.

Получение чистых и опрятных наборов данных на этапах подключения и подготовки может отнять очень много сил и времени. Создание *хранилища данных* (data warehouse), – а именно этим занимаются в отделе ИТ при подготовке данных, – зачастую выливается в разработку сопутствующих проектов, которые длятся годами. Но самое грустное, что к моменту готовности хранилища все понимают, что мир давно ушел вперед и оно утратило свою актуальность.

После приведения исходных данных в формат, готовый для обработки, приходит черед третьего слоя под названием **Анализ**. Именно здесь бьется сердце всего аналитического решения в целом. Построение аналитических моделей позволяет осуществлять срез и фильтрацию данных, рассчитывать агрегаты всех видов и добавлять вычисления для получения специфических выводов.

На слое **Визуализация** происходит создание отчетов и дашбордов на основе созданных ранее аналитических моделей. Мы назвали этот слой **Визуализация**, а не просто **Вывод**, поскольку для нас принципиально не просто предоставить пользователю результаты, а отобразить их таким образом, чтобы он мог сделать важные выводы и сконцентрировать внимание на ключевых показателях. Классические отчеты со страницами детализации не очень подходят для извлечения важных сведений, и пользователю нередко приходится полностью выгружать отчет в Excel и там агрегировать нужные ему показатели. Действительно эффективная визуализация данных позволяет пользователю точно выделить для себя наиболее важные составляющие и при необходимости погрузиться в детальные сведения прямо из отчета.

На верхнем слое – **Распространение** – выполняется осуществление доступа пользователей к отчетам и дашбордам по заранее установленным правилам.

Строите ли вы свое аналитическое решение в Excel, используете ли Power BI, разрабатываете собственную систему бизнес-аналитики уровня предприятия или вовсе не используете никакие системы автоматизации, вы так или иначе будете проходить сквозь все пять слоев. При должной организации процесса разработки аналитического решения все слои четко отделены друг от друга, и все преобразования данных выполняются в нужном слое. Это позволяет, в частности, избежать многократного дублирования логики. К тому же четкое разграничение между слоями поможет легко и безболезненно справиться с возникшими изменениями – к примеру, при смене источников данных.

Как вы увидите далее в этой главе, DAX живет именно в слое **Анализ** и тесно связан с соседними слоями **Подготовка** и **Визуализация**. Визуализации мы посвятим отдельный раздел, а пока нам необходимо разобраться, *кто и что делает в бизнес-аналитике*.

Аналитика предприятия и аналитика конечных пользователей

Компании в наше время становятся все более зависимы от данных. В частности, они нередко оценивают свою деятельность, обращаясь к дашбордам с *ключевыми показателями эффективности* (key performance indicators – KPI). Обычно эти дашборды строятся, исходя из очень высоких стандартов – в высших эшелонах организаций тщательно обсуждают долгосрочную стратегию и бизнес-процессы, способы измерения KPI и варианты их отслеживания. В результате утвержденные дашборды с выводом этих ключевых показателей могут долгое время не меняться, а их созданием и поддержкой занимается центральный отдел информационных технологий или специализированный центр бизнес-аналитики.

Также следствием зависимости компаний от данных является тот факт, что *каждое* принимаемое решение основывается исключительно на полученных цифрах. В результате появляется необходимость в более динамичной аналитике, способной отвечать на оперативные, не запланированные заранее запросы. Такой вид аналитики получил название *аналитика самообслуживания* (self-service BI), что предполагает возможность построения полноценных бизнес-решений без вмешательства отдела ИТ. Взглянув еще раз на нашу схему пя-

тислойной модели бизнес-аналитики, вы можете понять, что это практически невозможно. Мало у кого из пользователей есть необходимые знания, навыки и время, чтобы обслуживать все пять слоев. В идеале конечные пользователи и отдел информационных технологий должны дополнять друг друга, занимаясь приоритетными для них задачами.

Мы различаем два вида аналитики: *аналитику предприятия* (enterprise BI) и *аналитику конечных пользователей* (end-user BI). В первом случае вся аналитическая нагрузка по созданию и поддержке решений ложится на плечи отдела ИТ, а в качестве технической базы для обслуживания большого количества пользователей используются масштабируемые серверные системы и облачные платформы. При этом большое внимание уделяется качеству данных, а все сопутствующие процессы хорошо выстроены и отлажены.

Что касается аналитики конечных пользователей, то здесь, как ясно из названия, ключевую роль играют бизнес-пользователи. Для качественного выполнения своей повседневной работы им необходимо быстро получать те или иные данные из общей системы, и они это делают. Многие из них пытаются строить собственные аналитические решения в Excel, однако сложность структуры документов в Excel в конечном счете приводит к еще большим временным затратам на поддержку актуальности данных и масштабирование решения. К тому же Excel не предназначен для обработки сколько-нибудь больших данных. Зная это, компания Microsoft расширила возможности Excel за счет Power Query и Power Pivot как средств подготовки и анализа данных, что позволило превратить этот инструмент в подобие полноценной аналитической платформы для конечного пользователя. В результате совокупность этих инструментов вылилась в продукт, который мы сегодня называем **Power BI**.

Прелесть аналитической платформы Microsoft состоит в том, что в рамках нее инструменты аналитики предприятия и аналитики конечного пользователя могут работать совместно без каких-либо сложностей. И движущей силой этого процесса выступает технология, лежащая в основе Power BI. По сути, с ее появлением мы получили возможность сочетать ранее разрозненные ветви аналитики в одной архитектуре с разными уровнями возможностей самообслуживания, что показано на рис. 1.1.2.



Рис. 1.1.2. Интеграция аналитики предприятия и аналитики конечного пользователя

Наша пятисловная модель идеально подходит для демонстрации возможностей пользователей в рамках системы самообслуживания. Пользователи, нуждающиеся в собственной аналитике, могут интегрироваться в общую аналитическую систему на любом уровне. К примеру, они могут ограничиться созданием собственных визуализаций на базе существующих (корпоративных) аналитических моделей или строить свои аналитические модели на основе централизованно подготовленных наборов данных. Наконец, опытные бизнес-пользователи могут самостоятельно собирать и подготавливать данные, объединять их с корпоративными данными и создавать свои модели и отчеты. Более того, в своей системе они могут использовать объекты, созданные другими пользователями.

Очевидно, что при движении вниз сквозь описанные выше слои значительно повышается и уровень сложности выполняемых пользователем операций. И речь идет не только о компетенциях, которыми, безусловно, он должен обладать, но и об ответственности в виде следования корпоративным стандартам и инструкциям. Централизованные отделы ИТ и бизнес-аналитики должны всячески способствовать тому, чтобы избранные пользователи могли строить собственные аналитические решения. При правильной организации это ведет к созданию среды, в которой все пользователи оказываются в выигрыше от доступа к нужной им аналитической информации. Мы называем это *коллективной аналитикой* (collective analytics).

Power BI предлагает возможности по реализации коллективной аналитики. Одним из таких инструментов является DAX, который способен наделить пользователей возможностью не только создавать свои собственные аналитические решения, но и успешно встраиваться в разработку решений корпоративного уровня. О последней возможности мы поговорим в одном из следующих разделов, а пока нужно напомнить вам, что такое DAX и где его найти.

Для чего используется DAX и где его найти

В аналитических решениях, построенных на базе платформы Microsoft, DAX присутствует на слое **Анализ**. DAX живет внутри аналитических моделей в виде языка написания формул для определения вычислений и прочей логики. По сути, модели и DAX являются двумя сторонами одной и той же монеты: дизайн модели непосредственно влияет на сложность кода DAX, а ваши навыки в области DAX определяют вид и сложность модели (о ключевых концепциях моделирования данных мы будем говорить в главе 1.2).

Мощь языка DAX кроется в его потрясающих возможностях в плане агрегации данных. В состав DAX входит множество функций и конструкций, помогающих выстроить агрегацию идеальным образом для получения необходимой аналитики. В прошлом далеко не все типы агрегаций можно было рассчитывать напрямую – вместо этого они должны были быть реализованы на этапе подготовки данных. К примеру, накопительные итоги с начала года могут быть рассчитаны в DAX при помощи одной функции, тогда как в Excel или традиционных инструментах отчетности вам понадобится настроить немало метрик, показывающих, какие транзакции принадлежат к периоду накопления.

В результате итоговое решение, хоть и более сложно реализованное, все равно будет уступать в динамике формулам DAX, с помощью которых можно легко получить доступ и к историческим накопительным данным.

Как вы поняли, мы ведем к тому, что использование DAX позволяет значительно снизить усилия по подготовке данных по сравнению с традиционными инструментами. В то же время синтаксис DAX и некоторые его ключевые концепции в целом похожи на формулы в Excel, что облегчает его изучение специалистами из этой области. Но это отнюдь не значит, что достичь *высшей степени мастерства* в DAX просто. При изучении этого языка вы поймете, что сложность вопросов, на которые вы можете ответить при помощи DAX, будет постоянно расти, что повлечет за собой применение более изысканных конструкций и выражений DAX. В этой книге мы приведем немало примеров достаточно сложного применения языка DAX, что должно помочь вам разобраться в его тонкостях.

Все современные платформы для работы с данными от Microsoft в настоящее время поддерживают встраивание аналитических моделей, включая DAX. По-прежнему немного сбивает с толку, что в разных продуктах модели именуются по-разному. Давайте рассмотрим некоторые продукты от Microsoft и применение в них аналитических моделей и DAX.

Excel

Начиная с версии 2010 Microsoft *Excel* предоставляет пользователям возможности по моделированию данных с помощью надстройки *Power Pivot*. *Power Pivot*, также именуемая **Моделью данных** (Data Model) в Excel, представляет собой полноценную аналитическую модель на основе DAX.

Power BI

Самой продвинутой платформой для работы с данными в линейке компании Microsoft является, безусловно, *Power BI*. Впервые этот инструмент был представлен в качестве надстройки в *Office 365*, но в 2015 году был выделен в отдельный продукт. Аналитическая модель в Power BI называется *набором данных Power BI* (Power BI dataset) или просто набором данных, и именно здесь обитает DAX.

Наборы данных Power BI и другие объекты платформы функционируют в рамках *облачной службы Power BI* (Power BI cloud service), доступ к которой осуществляется с помощью сайта Power BI – powerbi.com. В числе других точек доступа к службе Power BI можно выделить мобильное приложение Power BI, платформу *Microsoft Teams* и даже пользовательские приложения, использующие для встраивания особую версию службы *Power BI Embedded*. Кроме того, есть еще *Power BI Report Server* – серверное программное обеспечение для клиентов, которые не могут или не хотят пользоваться облачными службами.

SQL Server Analysis Services

Серверная платформа для работы с данными *Microsoft SQL Server* содержит аналитический компонент под названием *Analysis Services (SSAS)*. Появившись на стыке веков под именем OLAP Services, SSAS на протяжении многих лет

представлял собой классический сервер *OLAP* (Online Analytical Processing – оперативная аналитическая обработка данных), ныне известный как **Многомерная модель** (Multidimensional). Начиная с версии SQL Server 2012 был представлен новый аналитический подход, получивший название **Табличная модель** (Tabular), в основе которого лежал язык DAX.

Вас может заинтересовать вопрос о различиях между двумя этими аналитическими концепциями, используемыми в SSAS. Подробности этой темы выходят за рамки данной книги, но главное отличие состоит в том, что многомерная модель основывается на классической реляционной концепции хранения данных, которая наилучшим образом подходит для выполнения агрегаций и сложных вычислений применительно к по-настоящему большим объемам данных. По сути, многомерная модель или куб (cube) выполняют все эти расчеты заранее, в момент обработки модели. В результате эта концепция стала значительно уступать в гибкости и динамике табличной модели данных, в которой благодаря мощи DAX все необходимые вычисления могут выполняться на лету, а разработчик отчетов не ограничен в работе только теми уровнями агрегации, которые были реализованы проектировщиком модели.

Azure Analysis Services

Azure Analysis Services (AAS) представляет собой полностью управляемую облачную службу, в основе которой лежит тот же табличный движок, что и в основе SSAS. Разница лишь в том, что служба AAS работает исключительно в облаке, что позволяет компании не заботиться об обслуживании баз данных и серверов. Также это решение обладает большой гибкостью, поскольку вы можете при необходимости в любой момент масштабировать свои ресурсы, чтобы они отвечали вашим требованиям.



Как мы уже сказали, аналитическая модель на основе DAX присутствует в разных продуктах под разными именами: Power Pivot, модель данных, набор данных или табличная модель. В результате мы столкнулись с определенными терминологическими сложностями при написании книги. Поскольку главным образом мы будем говорить о Power BI, мы решили использовать в книге термин *модель Power BI* (Power BI model) или просто модель. Термин *аналитическая модель* мы используем только при описании пятислойной структуры аналитики.

Инструменты для разработки моделей и написания выражений DAX

Вы можете сами сделать выбор в пользу того или иного инструмента для работы с DAX в зависимости от целевой платформы модели:

- для работы с моделью Power Pivot вы можете использовать Excel с одноименной надстройкой;
- для набора данных Power BI подойдет Power BI Desktop;



Стоит заметить, что фактически существует три версии Power BI Desktop. Одна из них может быть загружена с сайта Power BI по адресу powerbi.com. Вторая устанавливается из магазина Windows Store и автоматически обновляется, как и любое другое программное обеспечение из этого источника. А с учетом того, что обновления Power BI Desktop выпускаются почти ежемесячно, это немаловажно, хотя иногда бывает неприятно, когда обновления касаются областей, которые не вызывали вопросов. При желании вы можете установить на компьютер две версии одновременно. Третья версия Power BI Desktop, которая также может быть загружена с сайта Power BI, представляет собой специальную редакцию для работы с Power BI Report Server.

- для работы с моделью Tabular в SSAS или AAS вы можете использовать инструмент Visual Studio, предоставляющий богатые возможности для профессиональной разработки, включая интеграцию с системой контроля версий, возможность написания скриптов и совместимость;
- для наборов данных Power BI в емкости Power BI Premium вы можете выбирать между использованием Power BI Desktop и Visual Studio. Можно вести разработку посредством *конечных точек XMLA* (XMLA endpoint) – техники, реализованной в Power BI Premium и позволяющей набору данных Power BI извне выглядеть точно так же, как модель Tabular;
- также вы можете воспользоваться инструментами от сообщества, в частности Tabular Editor и DAX Studio. Они могут быть полностью интегрированы в Power BI Desktop.

При написании данной книги мы использовали «чистую» установку Power BI Desktop – бесплатную, такую же, какая уже наверняка установлена у вас. Каждый читатель книги может без труда загрузить Power BI Desktop и с его помощью проверить все примеры из файлов, находящихся в репозитории GitHub по адресу <https://github.com/PacktPublishing/Extreme-DAX> и на странице книги.

Сделано при помощи DAX: визуальные элементы и интерактивные отчеты

При обсуждении пятислойной модели бизнес-аналитики мы вскользь отметили важность визуальных отчетов. Но эффективное построение отчетов возможно только на основе моделей, использующих всю мощь DAX.

Визуализация данных является краеугольным камнем в процессе получения аналитических выводов из данных. Рассмотрите пример отчета, показанный на рис. 1.1.3.

SKU	Sales_Invoiced	Sales_Ordered	Costs_Invoiced	Costs_Ordered
Plain Z-block ZZ9-Y28437			14,392	7,000
Plain Z-block ZZ9-U28242		7,789	7,158	3,579
Plain Z-block ZZ9-T26881	397,505	20,000	26,963	13,482
Plain Z-block ZZ9-S64558	397,505	5,000	6,741	3,370
Plain Z-block ZZ9-R40552		0		
Plain Z-block ZZ8-Y18897	14,638	0	0	0
Plain Z-block ZZ8-U88875		0	26,167	12,727
Plain Z-block ZZ8-T61179		0	0	0
Plain Z-block ZZ8-S36266	102,555	21,848	33,041	16,802
Plain Z-block ZZ7-Y20451	60,877	11,470	48,411	23,504
Plain Z-block ZZ7-U37943	2,316,010	18,893	0	0
Plain Z-block ZZ7-T56946	373,786	53,481	141,109	70,711
Plain Z-block ZZ7-S23177	4,165,307	0	0	0
Plain Z-block ZZ7-R27880	603,548	107,124	379,770	188,075
Plain Z-block ZZ6-Y64561	70,576	17,396	69,854	34,748
Plain Z-block ZZ6-U54228	6,082,221	643,833	1,112,547	555,126
Plain Z-block ZZ6-T77239	3,167,617	302,802	729,264	370,290
Plain Z-block ZZ6-S11071	3,740,085	211,956	474,656	222,601
Plain Z-block ZZ6-R61826	225,502	0	0	0
Total	392,931,619	413,196,311	614,578,321	312,851,914

Рис. 1.1.3. Данные о продажах в табличном представлении

Можете ли вы быстро сказать, какие проблемы или возможности для развития есть у отчетной организации? Если да, наши поздравления – вы просто блестяще умеете работать с цифрами! Но большинство людей склонны анализировать более визуально оформленные отчеты. На рис. 1.1.4 приведен вариант графического представления тех же исходных данных.

При взгляде на эту диаграмму становится очевидно, что один из товаров продается гораздо лучше остальных. Это очень интересный и ценный вывод, позволяющий судить о том, что если компании удастся вывести продажи остальных товаров на тот же уровень, ее финансовое состояние значительно улучшится. Кроме того, данный вывод позволяет задуматься о том, что привлекательного есть в этом товаре, пользующемся таким огромным спросом. Может, его всегда приобретает один и тот же покупатель? Или с ним связана какая-то специфическая география продаж? И как обстоят дела с маржинальностью по этой товарной единице? Быть может, такая его популярность связана с тем, что мы торгуем им себе в убыток?

Таким образом, здесь мы имеем дело с обычными причинно-следственными связями, постоянно сопровождающими процесс принятия решений на основе визуальных отчетов. Аналитические выводы всегда порождают новые

Подходы к разработке решений

Модели данных Power BI и DAX позволяют использовать иной подход к разработке аналитических решений – с гораздо большим вовлечением бизнес-пользователей. Это способно существенно повысить ценность извлекаемых выводов для компании в целом.

Традиционный подход к созданию аналитических систем с централизацией ИТ-решений начинается с подключения к данным и их подготовки. На первый взгляд, здесь все логично – в конце концов, нам нужны исходные данные для получения аналитических выводов.

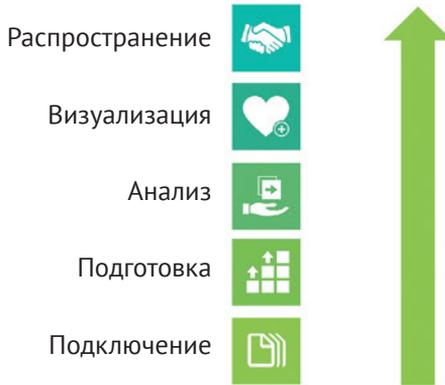


Рис. 1.1.6. Традиционный подход к разработке аналитических решений

Этот подход обычно материализуется в создании корпоративного хранилища данных. Идея наличия такого хранилища состоит, как ясно из названия, в хранении всех данных организации в одном месте и использовании их в качестве основы для отчетов.

Понятно, что это может быть непросто, поскольку компании могут пользоваться большим количеством источников с абсолютным разнообразием содержащихся в них данных. Традиционно хранилище реализуется с использованием реляционных систем управления базами данных, а это означает, что все исходные данные организации должны вписываться в схему базы данных в рамках единого хранилища.

Упомянутое разнообразие исходных данных приводит к повышению сложности общей схемы хранилища. Кроме того, при смене или добавлении источника новые данные также должны быть совместимы с существующей схемой – в противном случае придется менять хранилище для удовлетворения требований всех источников. Вследствие этого проекты, в основе которых лежат хранилища данных, часто славятся продолжительным временем реализации и высокой стоимостью. Многие специалисты построили свою карьеру на хранилищах данных, но не меньшее их количество сломали свои карьеры при внедрении подобных решений.

К тому же у традиционного подхода существует и более фундаментальный недостаток. Начиная с источников данных и постепенно поднимаясь по сту-

пенькам пятислойной модели, вы рискуете упустить важные и очень актуальные выводы, относящиеся к бизнесу. Несмотря на то что большинство проектов на основе хранилищ данных в процессе построения так или иначе учитывают существующие бизнес-требования, в реальности во многих случаях они в итоге отходят на второй план. Техническая сложность подобных проектов слишком высока, чтобы бизнес мог принимать в их разработке активное участие. В результате к окончанию работы над хранилищем данных (или к моменту его первого выпуска в рабочем окружении) оно зачастую уже не соответствует актуальным требованиям бизнеса.

В ожидании выхода корпоративных отчетов в организациях нередко применяется тактика «теневой информатизации», при которой пользователи вооружаются любыми доступными им средствами (обычно это Excel) и данными, до которых только могут дотянуться, чтобы строить собственные аналитические решения. И хотя эта тактика может помочь в плане получения быстрой актуальной аналитики, в долгосрочной перспективе она не способна решить проблемы организации с подходом к созданию аналитических решений. Качество данных (а значит, и достоверность сделанных на их основе выводов) в этом случае будет под вопросом, и риск принять неверное решение может оказаться слишком велик.

Использование Power BI для ускорения разработки аналитических решений

Взамен подхода, описанного выше, Power BI позволяет двигаться в обе стороны в рамках пятислойной модели, как показано на рис. 1.1.7.

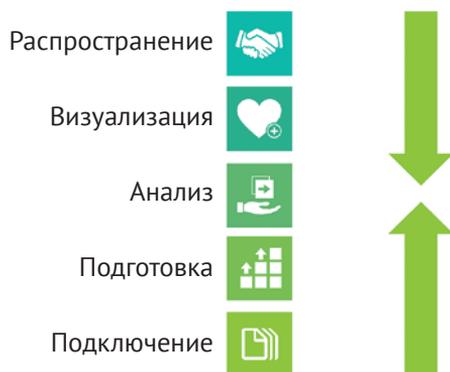


Рис. 1.1.7. Подход к разработке аналитических решений с помощью Power BI

Здесь мы используем Power BI не только в качестве целевой платформы для нашего решения, но и как инструмент для упрощения и рационализации нашего проекта. Применяя Power BI, вы можете воспользоваться всеми специфическими возможностями этого инструмента, включая быстрое создание отчетов и получение ценных выводов вне зависимости от местонахождения исходных данных. Модели данных Power BI и DAX идеально подходят для таких задач.

Этот подход базируется на двух основных принципах:

1. Мы не знаем точно, что нам нужно.
2. Наши данные некорректны.

Следствием этих принципов является невозможность сделать все правильно с первого раза. Вместо этого вам придется внедрять решение итеративно – методом последовательных ошибок и улучшений.

Мы не знаем точно, что нам нужно

Этот принцип означает отсутствие ожиданий от владельца бизнеса или от вас самих в этой роли, что вы предоставите корректные спецификации для отчетов. Если вы когда-нибудь сталкивались в работе с аналитическими системами, то вам должно быть это знакомо. И даже если вам кажется, что вы точно знаете, чего хотите от отчета, скорее всего, вы не учли всех деталей. А если учли, то наверняка в процессе разработки системы возникнет недопонимание в отношении реализации конкретного решения по причине отсутствия у разработчика необходимых знаний в области требуемого бизнес-контекста.

Как следствие нет большого смысла тратить слишком много времени на сбор требований, а также запись и подтверждение спецификаций. Вы можете просто принять к сведению, что первый созданный отчет все равно будет неправильным, так что лучше побыстрее создать его, чтобы как можно скорее ошибиться. Получается, что легче исправить все возможные ошибки и недочеты на практике, чем тратить время на абстрактное формулирование всех необходимых требований.

Описываемый подход можно формализовать при помощи множественных итераций с совместной сессией в конце для демонстрации прототипа и сбора обратной связи. Мы называем такие собрания *сессиями определения бизнес-дизайна* (business design sessions), что подчеркивает их назначение. Это не обычные сессии для сбора обратной связи или демонстрации первой версии продукта, здесь важен факт объединения усилий для достижения результата. В зависимости от ваших навыков в Power BI и DAX вы можете потратить два и более дней на разработку прототипа. Итоги сессии определения бизнес-дизайна используются впоследствии в качестве входных критериев для следующей итерации, как показано на рис. 1.1.8.



Рис. 1.1.8. Итеративный сбор требований

В результате этого совместного рабочего процесса получаются отчеты, в точности соответствующие всем актуальным бизнес-требованиям. Более того, отчеты и лежащая в их основе модель анализа используют конкретную спецификацию данных, которые должны предоставляться нижними уровнями пятислойной модели.

Наши данные некорректны

Второй принцип для вас, скорее всего, новостью не станет. Конечно же, наши данные некорректны! Причина этого в том, что реальные бизнес-процессы намного более сложны и запутаны в сравнении с нашей возможностью их смоделировать.

А если учитывать, что информационные системы разрабатываются с целью отразить то, как *должны* выглядеть бизнес-процессы в реальности, становится очевидно, что на пути автоматизации этих процессов непременно возникнет фундаментальная дилемма. Либо система будет реализовывать строгий контроль за качеством данных, в результате чего вы сможете вводить в нее только данные, в точности соответствующие разработанным процессам, либо будет допускать гибкость в отношении ввода данных, не отвечающих в точности идеальному течению процессов.

Чаще всего выбор делается в пользу второго варианта. Это обуславливает присутствие в типичных бизнес-системах возможности вводить пользовательские данные и наличие обходных путей. Как следствие данные из этих систем не всегда могут соответствовать вашим ожиданиям. И ситуация усложняется, когда ваши данные содержатся в электронных таблицах или других файлах!

В традиционных аналитических решениях ненадежные данные бывает очень сложно определить и исправить. Причина состоит в том, что такие системы хранят только агрегированные данные и не позволяют пользователям уточнять их при помощи детализированных сведений. И здесь на арене появляется Power BI, технологии моделирования в котором настолько мощны и совершенны, что в большинстве случаев позволяют загружать данные в модель без всякой предварительной агрегации. Визуальные и интерактивные отчеты позволяют делать аналитические выводы на основе пользовательской агрегации данных (при помощи DAX) и опускаться до требуемого уровня детализации.

При использовании итеративного подхода к разработке аналитических решений после первых нескольких итераций в результатах неизбежно будет много ошибок. Но опытные бизнес-пользователи обычно очень быстро способны их обнаружить. На первых стадиях недостатки в агрегированных данных устраняются именно так, но с продвижением по итерациям качество данных значительно улучшается. Возможность просматривать в отчетах Power BI детализированную информацию помогает повысить доверие к данным в новых аналитических решениях.

Цикл цифровой трансформации

До сих пор мы говорили о том, что необходимо для преодоления пути от сырых данных до аналитических выводов, и о роли в этом процессе моделей данных

Power BI и DAX. Как видите, для получения ценных сведений для бизнеса недостаточно просто подключиться к данным и подготовить их. Вся важная аналитика проистекает из визуальных и интерактивных отчетов.

В то же время финансовое положение компании не улучшится, если просто сидеть и смотреть на красивые отчеты. Важно то, как вы обращаетесь с полученными аналитическими выводами. Иными словами, вам необходимо предпринять какие-то действия для извлечения пользы из аналитических решений. Также вам не помешает наличие механизма оценки эффекта от этих действий – либо в автоматическом режиме, либо путем сбора обратной связи от пользователей посредством какой-то информационной системы. Результатом этого опять же будут данные.

Все это приводит нас к *циклу цифровой трансформации* (digital transformation cycle), или улучшения бизнеса на основе данных.

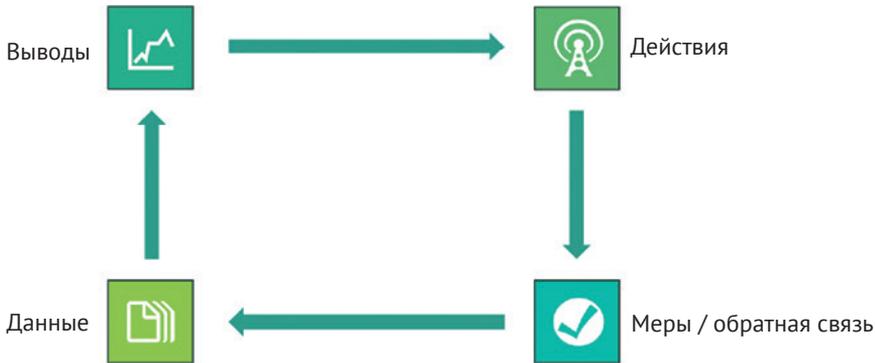


Рис. 1.1.9. Цикл цифровой трансформации

Power BI прекрасно проявляет себя на поприще преобразования исходных данных в аналитические выводы, но он не приспособлен для покрытия второй половины цикла трансформации, показанного на рис. 1.1.9. Здесь необходимы инструменты и технологии, обладающие возможностями ввода и обновления данных и объединяющие системы и людей.

Именно по этой причине компания Microsoft сделала Power BI частью объемной платформы под названием **Power Platform**. *Power Platform* охватывает полный цикл цифровой трансформации и опирается на единые принципы, включающие главенствующую роль бизнес-пользователя, легкий ввод данных и удовлетворение всех бизнес-требований.

Помимо Power BI, Power Platform состоит из следующих компонентов:

Power Apps предоставляет окружение для разработки бизнес-приложений с минимальным количеством кода для использования на мобильных устройствах и в браузерах. Эти приложения могут помочь при вводе и редактировании данных;

Power Automate позволяет автоматизировать обмен процессами между разными системами, службами и пользовательскими приложениями. К примеру, процессы могут запускаться в момент поступления письма по электронной почте, при этом может запрашиваться подтверждение пользователя и обновляться модель данных Power BI и связанные отчеты;

Power Virtual Agents предлагает платформу для интерактивного обмена информацией в рамках Power Platform посредством интеллектуальных чат-ботов. С их помощью пользователь может вводить данные или инициировать действия в интерактивном режиме, без необходимости запоминать какие-то сложные команды интерфейса.

Несмотря на то что описанные выше инструменты представляют собой отдельные приложения в рамках Power Platform, между ними существует тесная связь. К примеру, вы можете создать отчет в Power BI со встроенным приложением Power Apps, чтобы пользователи могли изменять данные в удобном виде прямо в момент получения аналитических выводов. Также и потоки Power Automate могут быть встроены в отчеты Power BI, что позволит пользователям предпринимать необходимые действия на основе полученных результатов.

Заключение

В этой главе мы обсудили область применения бизнес-аналитики и центральную роль, отведенную в современных аналитических решениях моделям данных. Модели Power BI идеально подходят для использования в аналитических решениях, не в последнюю очередь благодаря мощи DAX.

Вы узнали о двух возможностях, предоставляемых языком запросов DAX, которые позволяют ключевым образом влиять на концепцию и разработку решений:

- DAX может быть использован для создания сложных вычислений, связанных с агрегацией данных, для чего раньше приходилось долго и кропотливо подготавливать данные. Таким образом, DAX позволяет сместить фокус с данных и всех связанных с ними работ на логику получения аналитических выводов;
- с помощью DAX бизнес-пользователи, знакомые с Excel, могут самостоятельно работать с аналитическим решением в меру своей осведомленности и навыков. Это позволяет более правильно расставить бизнес-приоритеты и привлечь пользователей к работе.

Поскольку модели данных Power BI и DAX представляют две стороны одной монеты, очень важно уметь балансировать между ними для достижения оптимальных результатов. Проще говоря, DAX необходимо использовать только там, где это уместно, чтобы не углубляться в работу с данными и не обращаться к нему при подготовке и генерировании данных.

В следующих нескольких главах мы разовьем описанные здесь идеи. В главе 1.2 мы поговорим о том, что нужно и что не нужно делать при разработке моделей данных Power BI. Глава 1.3 будет посвящена использованию DAX для достижения наилучших результатов. А в главе 1.4 мы продолжим работать с DAX и обсудим очень важные концепции, применимые при написании запросов и вычислений. Во второй части книги будет содержаться множество примеров, по большей части из реальных проектов, с помощью которых мы продемонстрируем всю мощь языка DAX и покажем, как нужно балансировать между выражениями DAX и моделированием данных в Power BI.

Разработка модели

Эффективное использование языка DAX начинается с разработки хорошей во всех отношениях аналитической модели данных. В этой главе мы обсудим важные аспекты моделирования данных – ключевые для построения полноценной модели.

Темы, которые будут рассмотрены в этой главе:

- хранение данных в движке Power BI;
- правильный выбор типов данных;
- связи;
- оптимальная структура модели данных.

Чтобы модель данных отличалась своей эффективностью, вам необходимо особенным образом выстроить свое мышление. Такая смена образа мыслей требуется при переходе в Power BI из Excel или из реляционных баз данных. Разработчикам Excel бывает непросто понять концепцию связей между таблицами в модели данных Power BI. Но и для специалистов в области баз данных, где также есть связи, в Power BI найдется немало новых нюансов в этой области. Основной сложностью здесь является внешняя схожесть концепций, используемых в Power BI и базах данных, притом что они имеют фундаментальные отличия. Об этом, в частности, мы поговорим в данной главе.

Колоночное хранение данных

Своей мощью Power BI в основном обязан очень эффективному способу хранения данных. По своей сути модели данных Power BI представляют собой своеобразные базы данных по части организации и хранения информации. Но в то же время внутренне система хранения кардинально отличается от традиционных баз данных и других технологий, с которыми вы сталкивались в работе ранее.

Реляционные базы данных

Традиционно хранение информации в компаниях организуется при помощи *реляционных систем управления базами данных* (relational database management system – RDBMS) или *PCСУБД*, например Microsoft SQL Server. В реляционных базах данных информация хранится в виде *таблиц* (table) с фиксированным количеством столбцов. При этом каждый столбец характеризуется определенным *типом данных* (data type), таким как, например, *integer*, *text* или *decimal*, и информация о типах столбцов служит основой для подсчета места на диске,

необходимого для хранения одной строки или *записи* (record) в таблице, и количества записей, которые могут быть сохранены в файле данных на диске. Такая концепция хранения данных делает реляционные базы идеальным выбором для приложений, обрабатывающих транзакции, таких как система учета розничных продаж или финансовой отчетности.

Концепция *индексов* (index) позволяет достаточно быстро осуществлять поиск нужной записи в таблице, что обуславливает эффективную обработку существующих записей при работе с транзакциями. При этом таблицы могут быть объединены посредством *связей* (relationship), откуда и происходит название реляционной базы данных. Это позволяет обеспечить данным целостность. Например, вы не сможете добавить в таблицу с продажами запись с неизвестным покупателем.

Концепция реляционных баз данных появилась достаточно давно и за это время была значительно улучшена и оптимизирована. Этим объясняется тот факт, что многие традиционные аналитические платформы до сих пор полагаются на такое хранение информации. В то же время стоит отметить, что нужды аналитических решений значительно отличаются от требований к транзакционным системам. При работе с аналитикой вам обычно не требуется извлекать данные по строкам. Вместо этого вам чаще необходимо оперировать данными в виде блоков из большого количества записей, содержащих один или несколько столбцов. Для получения исходных данных из колонки реляционной базы данных все равно так или иначе приходится проходить по целым строкам в хранилище. Этим объясняется низкая эффективность реляционных баз при агрегировании данных на основании большого количества записей.

На рис. 1.2.1 схематически показан принцип хранения данных по строкам (помечены цифрами). Как видите, при таком хранении нет никакой возможности быстро выбрать сведения, относящиеся к нужной колонке (затемненные блоки).

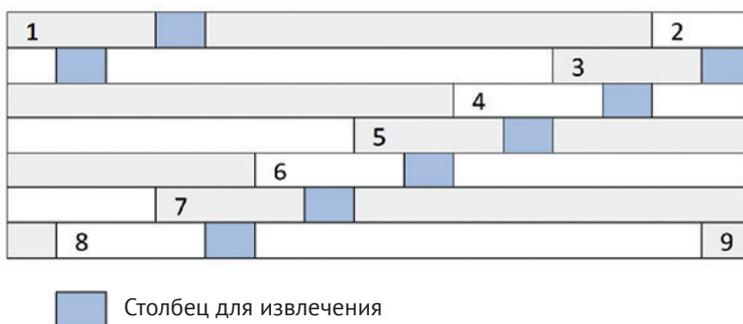


Рис. 1.2.1. Извлечение значений из столбца при построчном хранении данных

Колоночные базы данных

В моделях Power BI концепция хранения данных в реляционных базах перевернута на 90 градусов, так что информация в них аккумулируется не по строкам, а по столбцам. Одна из причин этого в том, что при работе с аналитическими системами нам зачастую требуется извлекать данные всего из одной

или нескольких колонок, но по всем существующим строкам. А эффективно это можно сделать только в одном случае – если данные из одного столбца физически хранятся вместе.

Еще одной причиной такого хранения данных является то, что в реальных системах значения в строках из одной колонки очень часто повторяются. К примеру, в таблице, содержащей миллионы строк, в одной из колонок может присутствовать всего несколько тысяч повторяющихся товаров. При колоночном хранении такие данные могут быть очень эффективно сжаты – по сути, храниться будут только уникальные значения и номера строк, в которых они присутствуют.

Высокая степень сжатия данных в колоночном хранилище дает возможность легко удерживать всю базу в памяти компьютера, вместо того чтобы обращаться за нужными данными к диску. Это, в свою очередь, позволяет существенно повысить скорость извлечения данных.

Кроме того, колоночная концепция хранения обеспечивает очень эффективное выполнение агрегации данных. К примеру, вместо того чтобы суммировать все значения из столбца построчно, движок может просто взять список уникальных значений из нужного столбца и перемножить на количество их вхождений. Иными словами, в Power BI реализован собственный движок хранения и обработки данных, позволяющий удовлетворить все основные требования при работе с аналитическими решениями, включая обработку больших массивов данных с определенными характеристиками, а также агрегирование и вычисления на их основе.

Есть, однако, у колоночного типа хранения данных и один недостаток. В конце концов, как вы понимаете, мы не можем полностью абстрагироваться от информации о том, какие значения из столбцов соседствуют друг с другом в одной и той же строке. Нам недостаточно знать, что товар с кодом 103 был продан, мы хотим также понимать, кому он был продан, когда именно и в каком количестве. Для предоставления этой информации движку необходимо хранить список *указателей* (pointer) с информацией о том, какие значения в каких строках размещаются. Очевидно, что накладные расходы, связанные с хранением этих указателей, растут с увеличением количества столбцов в таблице. Из этого следует, что «узкие» таблицы, состоящие из небольшого количества колонок, в моделях Power BI будут показывать лучшую эффективность по сравнению с «широкими».

Типы данных и кодирование

В моделях Power BI вы можете работать с ограниченным набором *типов данных* (data type). Выбор правильного типа данных для столбца имеет огромное значение, поскольку именно этот тип будет определять, как данные будут храниться и *кодироваться* (encoding) и насколько эффективно модель будет их обрабатывать. Ниже приведен список всех типов данных, поддерживаемых в Power BI:

- **Текст** (Text): это самый общий тип данных. Практически любые данные можно сохранить в виде текста. При загрузке данных посредством Power Query общий тип **Any** преобразуется в модели Power BI в тип **Текст**. Это может приводить к хранению числовых данных в виде текста, если за-

быть указать тип в Power Query в явном виде. Конечно, вы можете изменить тип данных в модели, что приведет к созданию соответствующего шага в Power Query;

- **Целое число** (Whole Number): как несложно догадаться, этот тип используется для хранения целочисленных данных. Это один из самых эффективных типов данных в Power BI с точки зрения хранения и сжатия;
- **Десятичное число** (Decimal Number): это наиболее универсальный тип данных для хранения числовой информации. Он позволяет хранить как целые числа, так и дробные. Максимальное количество знаков – 15;
- **Десятичное число с фиксированной запятой** (Fixed Decimal Number): этот тип данных, иногда называемый **Валюта** (Currency), используется для хранения десятичных чисел с четырьмя знаками. Максимальное количество знаков, включая десятичные, составляет 19. Таким образом, этот тип данных имеет меньшую точность по сравнению с предыдущим. Обычно он используется для хранения финансовой информации, но может быть использован и для любых других числовых данных, не требующих большого количества знаков после запятой;
- **Дата и время, Дата, Время** (Date/Time, Date, Time): в Power BI календарные значения хранятся в структурах, подобных тем, которые используются в Excel. Таким образом, для их хранения применяются десятичные числа, в которых целая часть отвечает за дату, а дробная – за время. Разница с Excel заключается в том, какая дата рассматривается в качестве точки отсчета: в Power BI число 1 соответствует дате 31 декабря 1899 года, тогда как в Excel – 1 января 1900 года (в обоих случаях отсчет ведется с полуночи). Десятичные знаки добавляются в виде доли от суток. Например, значение 2,5 в Power BI будет представлять полдень 1 января 1900 года. Вы можете хранить календарные данные в трех видах: тип **Дата и время** (Date/Time) позволяет хранить одновременно дату и время, тогда как типы **Дата** (Date) и **Время** (Time) предназначены для хранения отдельных составляющих, при этом тип **Дата** (Date) аналогичен целочисленному типу, а **Время** (Time) – десятичному;
- **Истина/ложь** (True/False): этот тип данных предназначен для хранения только двух значений: **Истина** (True) или **Ложь** (False). Очевидно, что это очень эффективный в плане хранения тип данных из-за ограниченности используемых значений;
- **Двоичный** (Binary): этот тип данных используется для хранения данных, которые не могут быть представлены в виде текста, таких как изображения или документы. К таким данным неприменимы агрегации и вычисления, но этот тип данных может применяться для хранения изображений, используемых в отчетах.

Выбор наиболее подходящих типов данных для колонок – это ключевой момент, влияющий на эффективность будущей модели. Движок Power BI оптимизирован таким образом, чтобы уникальные значения в столбцах хранились максимально эффективно, насколько это возможно. Давно прошли те времена, когда мы сражались за каждый байт и даже бит при хранении информации, но при разработке моделей данных по-прежнему очень важно понимать, как именно в компьютерах хранится информация. Выбор типов данных поможет

определить, сколько бит потребуется для хранения вашей модели. А поскольку модель размещается в оперативной памяти, борьба за каждый бит здесь по-прежнему актуальна.

Представьте, что в вашей модели данных есть колонка с целочисленными значениями в интервале от 0 до 10. В двоичном представлении число 10 выражается как 1010 и занимает 4 бита в памяти. Таким образом, значения в этой колонке могут быть закодированы словами размером 4 бита с явным представлением значений. Такой тип кодирования называется *кодированием на основе значений* (value encoding). В современных компьютерах с 64-битными процессорами использование для хранения значений четырех бит вместо шестидесяти четырех уже сулит неплохую прибавку в отношении эффективности.

Кодирование на основе значений может использоваться для типов данных, представляющих из себя «под капотом» целочисленные значения. К таким типам относятся **Целое число**, **Дата**, **Истина/ложь** и **Десятичное число с фиксированной запятой**. Последний из них по своей сути представляет целочисленный тип с делением на 10 000. Движок DAX способен выполнять базовые трансформации перед осуществлением кодирования на основе значений. Например, он может вычесть из всех значений одно и то же число.

Остальные типы данных не могут быть напрямую представлены в целочисленном виде, но движку все же необходимо находить экономичный способ их хранения. В таких случаях он нумерует уникальные значения в столбце и хранит только их цифровые эквиваленты. Такой подход получил название *кодирования на основе хеша* (hash encoding). Этот тип кодирования менее эффективен по сравнению с кодированием на основе значений, поскольку движку приходится выполнять сопоставление кодов и реальных значений при каждом обращении к столбцу.

Важно помнить, что Power BI автоматически выбирает оптимальный тип кодирования данных на основании типа и значений в столбцах. Это означает, что даже при хранении в колонке целочисленных значений движок может остановиться на кодировании на основе хеша. К примеру, если интервал значений в столбце не ограничивается диапазоном от 0 до 10, а может включать в себя и 1 000 000, для хранения значений в явном виде может потребоваться довольно много места, в результате чего движок может сделать выбор в пользу хеширования.

Мы достаточно регулярно наблюдаем подобную ситуацию в реальных проектах, особенно в случаях с хранением дат. Представьте, что у вас в модели есть таблица с сотрудниками, в которой учитываются не только даты приема сотрудников на работу, но и даты их увольнения. Для людей, которые работают в компании и пока никуда не планируют уходить, последняя дата, очевидно, должна оставаться незаполненной. Но зачастую вместо пустых значений в этом случае принято использовать какую-нибудь дату из далекого будущего. Это вполне допустимый подход, но если, к примеру, выбрать для этих целей дату 31 декабря 9999 года, то можно попрощаться с возможностью использовать для этой колонки кодирование на основе значений. Вместо этого всегда лучше использовать не столь далекую дату, например 31 декабря 2029 года, в зависимости от вашего конкретного сценария.