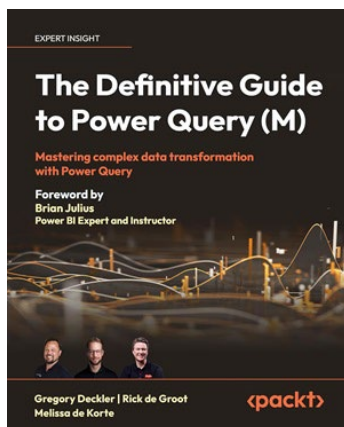


## Грегори Декер, Рик де Гроот, Мелисса де Корте. Полное руководство по языку M Power Query

Я использую Power Query на работе и в игре [Fantasy Premier League](#). При том что Power Query посвящено немало книг, языку M в каждой из них обычно уделяется одна глава. Я компенсировал этот пробел изучая англоязычные блоги: [Ben Gribaudo](#), [Chris Webb](#), [Gil Raviv](#), [Rick de Groot](#), а также [телеграмм-канал](#) Михаила Музыкина. Заметки в этих источниках были весьма поучительными, но целостную картину приходилось собирать по крупицам. Как обычно официальная [документация](#) Microsoft казалась слишком заумной для новичка. Я даже начал писать книгу об основах языка M. И тут такая удача – выход 760-страничного полного руководства по языку M!

Авторы подчеркивают, желательно, чтобы эта книга не была вашим первым знакомством с Power Query. Если это не так, рекомендую начать с Кен Пульс, Мигель Эскобар [Язык M для Power Query](#). Большинство примеров в книге основаны на Power BI Desktop. Перевод некоторых крупных текстовых фрагментов я доверил Chat GPT, но не самого важного в этой книге. Мои комментарии набраны с отступом. Буду признателен за замечания, если допустил неточности.

Gregory Deckler, Rick de Groot, Melissa de Korte. The Definitive Guide to Power Query (M). – UK: Packt Publishing Ltd., March 2024 – 760 p.



### Содержание

#### [Глава 1. Знакомство с M](#)

Глава 2. Работа с Power Query/M

Глава 3. Доступ к данным и их объединение

Глава 4. Общие сведения о значениях и выражениях

Глава 5. Общие сведения о типах данных

Глава 6. Структурированные данные

Глава 7. Концепция M

Глава 8. Работа с вложенными структурами

Глава 9. Параметры и пользовательские функции

Глава 10. Работа с датой, временем и длительностью

Глава 11. Сравнение, замена, комбинирование, разделение

Глава 12. Обработка ошибок и отладка

Глава 13. Итерация и рекурсия

Глава 14. Проблемные паттерны данных

Глава 15. Оптимизации производительности

Глава 16. Включение расширений

## *Вступительное слово*

Иногда звезды выстраиваются в одну линию в нужном направлении, и в результате получается нечто неожиданно великолепное. Это история книги, которую вы сейчас держите в руках. Но я забегаю вперед. Давайте вернемся в начало 2022 года, где и начинается история этой книги.

На тот момент я уже более 30 лет был специалистом по работе с данными в Excel, а также много занимался разработкой Power BI, и проводил онлайн-обучение. Я долгое время использовал Power Query для очистки и преобразования данных перед выполнением анализа и построением отчетов. Но почти вся эта подготовка выполнялась с помощью пользовательского интерфейса Power Query. Несмотря на то, что пользовательский интерфейс Power Query хорошо спроектирован, он довольно ограничен, предоставляя доступ только к части из 700+ функций M. Итак, в начале 2022 года я решил, что перестану баловаться M и сделаю его тщательное изучение своей главной тренировочной целью на год. В 2022 году это было непросто, поскольку, несмотря на то, что по Power Query были фантастические книги, в каждой из них лишь небольшой процент контента был посвящен M. В Интернете было много отличных ресурсов по M, но они предлагали глубокое понимание очень специализированных тем, что также не способствовало идеальному обучению.

Несмотря на эти препятствия, к концу 2022 года я обнаружила, что мое глубокое погружение в M было трансформационным. Мои отчеты Power BI работали намного быстрее, DAX, используемый в них, стал значительно проще, я смог очищать данные быстрее, чем когда-либо, и теперь я могу легко создавать многократно используемые пользовательские функции для автоматизации повторяющихся задач по работе с данными. Я также обнаружил, что изучение того, как визуализировать и манипулировать сложными структурами данных на M, значительно улучшило мои способности к программированию на других языках. Тем не менее, я также понял, что мой опыт был совершенно невозпроизводимым из-за того, что трое моих друзей (не случайно авторы этой книги) оказались одними из самых осведомленных экспертов по Power Query/M в мире, и я в значительной степени полагался на их опыт во время обучения. На самом деле, я часто шутил с Мелиссой и Риком о том, что каждый из них является «Избранным», которому суждено написать M-аналог «Полного руководства по DAX», но я думаю, что это казалось слишком монументальной задачей для них обоих.

В то время я был основным тестировщиком набора приложений, которые разрабатывал Грегори Деклер. Мы регулярно говорили о Power Query и M, когда обсуждали потенциальные будущие возможности этих приложений. Однажды Грег упомянул, что он работает над планом книги для того самого всеобъемлющего руководства по M, о котором я шутил (но в то же время искренне желал, чтобы оно существовало). Он спросил меня, кого бы я порекомендовал в качестве соавторов этой книги, и я очень рад сообщить, что мой ответ можно найти на обложке этой книги.

Освоение M не является ни быстрой, ни легкой задачей. Тем не менее, я полностью уверен, что независимо от того, являетесь ли вы аналитиком данных, разработчиком Power BI, специалистом по обработке и анализу данных, бизнес-аналитиком или пользователем Excel, желающим повысить уровень своих навыков, наличие тех же трех четких, терпеливых и блестящих инструкторов, что и у меня, сделает изучение M приятным и меняющим правила игры опытом и для вас.

*Брайан Джулиус, эксперт и инструктор по Power BI*

## *Об авторах*

Грегори Деклер (Gregory Deckler) – семикратный обладатель звания Microsoft MVP по Data Platform, активный [блоггер](#) и участник сообщества Power BI, автор более 6 000 ответов на вопросы сообщества. Грег является автором многочисленных книг по Power BI, в том числе первого и второго изданий Learn Power BI, вторых изданий Power BI Cookbook и Mastering Power BI, а также DAX Cookbook. Грег также создал несколько внешних инструментов для Power BI и регулярно публикует видеоконтент на своих YouTube-каналах – Microsoft Hates Greg и DAX For Humans.

*Я хотел бы выразить признательность моим замечательным соавторам за эту книгу. Я многому научился у них. Я благодарен сыну Ракете за то, что он поддерживал все мои начинания. Особая благодарность Брайану Джулиусу, который вдохновил меня на написание этой книги и без которого эта книга не существовала бы.*

Рик де Гроот (Rick de Groot) – консультант по Power BI, блоггер, ютубер и автор из Нидерландов. Имея опыт работы в сфере финансов, он посвятил 14+ лет накоплению опыта в области анализа данных, особенно на языке M Power Query. Он является независимым консультантом, который также проводит обучение по Power BI, Power Query и DAX. Он регулярно публикует статьи в своих блогах Power Query How и [BI Gorilla](#), которые стали важными источниками для статей и учебных пособий по M. Он также активен в YouTube-канале BI Gorilla. Стремление Рика делиться своим опытом принесло ему награду Microsoft Data Platform MVP в течение двух лет подряд.

*Моя глубочайшая признательность Брайану Джулиусу. Его вера в меня подтолкнула к тому, чтобы больше рассказывать о Power Query, и помогла мне вырасти как писателю. Без его поддержки я не думаю, что мое имя было бы на обложке этой книги. Он также сыграл важную роль в наших обсуждениях и помог сформировать содержание этой книги. Я глубоко благодарен за твою поддержку, Брайан. Моей семье – ваше терпение и поддержка во время написания этой книги значат для меня целый мир. Кроме того, огромное спасибо всем вам. Все обсуждения были невероятно ценными.*

Мелисса де Кортте (Melissa de Corte) – увлеченный специалист по решению проблем, известная тем, что упрощает сложные задачи с помощью Power Query. Обладая впечатляющим послужным списком в качестве эксперта, она стала ценным активом для сообщества, вовлекая, поддерживая и вдохновляя других. За ее профессиональной личностью скрывается искренняя преданность делу расширения прав и возможностей других людей путем обмена знаниями. Ее портфолио включает блоги, учебные пособия, курсы и вебинары, чтобы сделать Power Query и M более доступными.

*Марселю и Сэму за то, что они наполняют каждый момент, который мы разделяем, радостью. Я благодарна моим соавторам, а также Брайану Джулиусу за их роль в создании этой книги. Моей семье за их бесконечное терпение и поддержку. И всем, кто щедро делится своими знаниями, пробуждает любопытство и способствует созданию поддерживающего и инклюзивного сообщества. Ваш вклад неоценим!*

### *О рецензентах*

Эрин Островски – цифровой кочевник, который стремится построить здоровые, счастливые отношения, куда бы она ни пошла. Она обожает [STEAM](#) – не забывайте об искусстве, ребята! – и надеется вдохновить других присоединиться к веселому решению проблем, в котором нуждается этот мир. Эрин любит учиться, творить, заводить друзей, путешествовать и писать о многих любопытных моментах жизни. В настоящее время она руководит Fit 4 Duty Data, консалтинговой и учебной фирмой по данным и отчетности, которая поддерживает глобальный горнодобывающий и энергетический секторы.

*Я хотел бы поблагодарить Создателя, моих друзей и мою семью – вы все делаете жизнь прекрасной, особенно в трудные времена.*

Вахид Дустимайд – Microsoft MVP и сертифицированный тренер с более чем 15-летним опытом работы. Технофил, увлеченный управлением проектами и аналитикой данных. В настоящее время работает в Австралии менеджером по контролю и аналитике проектов. Активно участвует в сообществах Microsoft Fabric и Power BI, с удовольствием делится своим опытом в блогах и видео, а также организует мероприятия для персидской группы пользователей Power BI.

*Я хотел бы выразить сердечную благодарность моей жене за ее бесконечную поддержку и терпение. Ее поддержка была моей путеводной звездой.*

Ахмед Ойелоуо (Ahmed Oyelowo), управляющий партнер, ведущий тренер и консультант Foresight BI & Analytics Global Solutions, является сертифицированным специалистом Microsoft по анализу данных Power BI и Azure Enterprise Data Analyst, тренером Microsoft, автором бестселлера по Power BI на Udemy, который прошли 90 000+ студентов. Четырехкратный обладатель звания Microsoft MVP в категории «Платформа данных». Разрабатывал передовые решения для корпоративных клиентов.

*Моя искренняя признательность Создателю за дар всего, что необходимо для завершения этого обзора. Я также должен выразить благодарность дражайшей жене Кафаят и детям, Юсре и Ясиру, за их поддержку и понимание, пока я проводил некоторое время вдали от них, чтобы выполнить эту задачу. Благодарю авторов этой книги за замечательный ресурс!*

## *Подробнее в Discord*

Присоединяйтесь к пространству нашего сообщества в Discord для обсуждений с автором и другими читателями: <https://discord.gg/vCSG5GBbyS>

## *Предисловие*

За последнее десятилетие популярность языка M Power Query постоянно росла. Сейчас он работает в таких продуктах Майкрософт, как Power BI, Excel, Power Platform, Dynamics 365, SQL Server и Data Factory. Сегодня M и Power Query являются незаменимыми инструментами для специалистов по работе с данными.

Эта книга сделает вас специалистом языка M. Хотя M можно написать с помощью графического пользовательского интерфейса (GUI) редактора Power Query, этот подход сильно ограничивает вас лишь небольшой частью из 700+ основных функций M. По нашим оценкам, графический интерфейс для создания запросов позволяет решить только 50% задач, связанных с преобразованием данных. Владение M позволяет приблизить этот показатель к 100%. Начав с простых концепций и кода, эта книга постепенно продвигает вас по пути возрастающей сложности. Все понятия объясняются ясно и кратко с практическими примерами, демонстрирующими концепции в действии. К концу этой книги останется мало проблем трансформации данных, с которыми вы не сможете справиться.

Приготовьтесь к увлекательному и полезному путешествию к овладению языком M!

## *Для кого эта книга*

Если вы новичок в Power Query, то эта книга может быть не лучшим выбором для вас. Существуют книги, которые знакомят с основами пользовательского интерфейса и простыми действиями на основе кнопок, предлагая простой способ начать работу.

С другой стороны, если вы серьезно относитесь к полному пониманию языка M, то эта книга для вас. Если это ваш первый опыт работы с M, некоторые концепции могут показаться вам сложными. Чтение страниц в первый раз даст вам много информации, но некоторые темы могут быть слишком сложными. По мере того, как вы будете знакомиться с M, повторное прочтение книги, скорее всего, сделает эти сложные темы более понятными.

В целом, нашей целью было написать книгу, в которой теоретические знания сочетаются с практическими примерами. Первые несколько глав носят более теоретический характер и дают представление об основах языка M. По мере того, как книга продвигается вперед и ваше понимание углубляется, мы включаем проекты и файлы упражнений, которые вы можете выполнять вместе с нами.

Язык M Power Query полезен для самых разных пользователей. В Power BI можно использовать M для подготовки модели данных, в Excel – для преобразования данных и вывода их в электронные таблицы, а пользователи Data Factory могут использовать M для запросов API и передачи результата в базу данных. В этой книге мы стремимся угодить всем этим аудиториям.

Несмотря на то, что некоторые темы, такие как создание пользовательских коннекторов или оптимизация производительности, более интересны людям с продвинутыми знаниями из-за их технической природы, мы считаем, что всем пользователям полезно понимать эти аспекты M.

На освоение языка M у нас ушло несколько лет. Это сложное путешествие, требующее сосредоточенности и большого количества экспериментов. Однако, если вы проявите настойчивость, наша книга предложит беспрецедентную глубину всех аспектов M, что позволит вам стать настоящим экспертом.

## *О чем идет речь в этой книге*

Глава 1 знакомит с основами языка M, выражением let, историей M, формальными и неформальными характеристиками M. Обсуждается, кто должен изучать M, зачем изучать, где и как использовать.

Глава 2 знакомит с редактором Power Query – основным приложением для написания кода M. Вы сможете перейти от автоматического создания кода M в графическом интерфейсе редактора к написанию кода с помощью пользовательских столбцов и расширенного редактора.

Глава 3 исследует различные коннекторы данных для доступа к файлам, папкам, базам данных, кубам, двоичным данным; способы объединения данных из разных запросов.

Глава 4 знакомит с различными видами значений в языке M, а также с выражениями, операторами, управляющими структурами и типами (enumerations, [перечисления](#)).

Глава 5 дает общие сведения о типах данных, исследуется важность типов в языке M. Показана структура типов и их применении в реальных сценариях. Предлагаются методы автоматического определения типов данных и обсуждается, почему преобразование типов важно. Представлена концепция фасетов и атрибутирования типов, которые часто приводят к ошибкам в запросах.

Глава 6 изучает структурированные данные: списки, записи и таблицы. Такие данные могут хранить в себе несколько значений. Вы узнаете о методах их создания, типичном использовании и операторах, которые работают со структурированными данными. Объясняется, как получить доступ к отдельным значениям внутри структурированных данных.

Глава 7 приводит некоторые из наиболее абстрактных концепций M, которые имеют решающее значение для истинного понимания того, как работает язык. Рассматриваются: глобальная среда, создание собственной глобальной среды, разделы, замыкание, свертывание запросов и метаданные.

Глава 8 представляет методы преобразования и манипулирования структурированными данными, которые помогут вам преодолеть ограничения графического интерфейса.

Глава 9 начинается с обсуждения роли параметров в обеспечении гибкости и динамики запросов. Описано, как создавать пользовательские функции, их синтаксис, важность типов данных и методы отладки. Вы узнаете, что выражение *each* является синтаксическим сахаром для функции.

Глава 10 объясняет, почему временной анализ является ключом к получению многих аналитических сведений о данных и почему для выполнения такого анализа необходимо уметь работать с датами, временем и длительностью.

Глава 11 исследует методы манипулирования данными. Показано, как сравнивать, сортировать, заменять, объединять и разделять значения. Эти методы необходимы для широкого круга задач по преобразованию данных.

Глава 12 посвящена тому, что такое ошибки, и содержит рекомендации по их обработке на языке M и успешной отладке кода, что позволяет создавать надежные запросы. Исследуются методы сообщения об ошибках.

Глава 13 объясняет рекурсию с помощью оператора @. Основное внимание уделяется методам итераций с использованием List.Transform, List.Accumulate и List.Generate. Вы узнаете о требованиях к памяти и о том, какие аспекты следует учитывать для достижения наилучшей производительности.

Глава 14 иллюстрирует универсальность языка M, охватывая различные распространенные методы извлечения текста, а также предоставляя комплексный подход к созданию управляемого пользовательского решения для работы с большими файлами.

Глава 15 рассматривает факторы, влияющие на производительность запросов. В ней рассказывается о гибридных контейнерах и важности управления памятью. Мы изучим свертывание запросов, механику брандмауэра формул и представим различные методы повышения производительности запросов.

Глава 16 демонстрирует, как расширить язык M, создав собственную библиотеку функций, которую можно использовать повторно, включая подробный пример создания пользовательского коннектора данных.

### Чтобы получить максимальную отдачу от этой книги

Мы ожидаем, что у вас есть базовые знания Power Query и некоторый опыт анализа данных. Если у вас есть опыт работы с языком M, это полезно для более быстрого понимания концепций. Тем не менее, знание M не является обязательным требованием. В книге есть ссылки на код SQL и выражений анализа данных (DAX). Тем не менее, вам не нужно знать эти языки, потому что сравнения просто предназначены для того, чтобы отразить различные подходы в этих языках. Не



переживайте, если вы не понимаете фрагмент кода. Это значит, что сравнение не применимо к вам.

В более продвинутых разделах книги мы рассмотрим свертывание запросов, пользовательские коннекторы и использование памяти. Некоторые из вас, возможно, не знакомы с этими темами, и это нормально. Тем не менее, эта информация является важным элементом того, для чего используется язык M, и мы думаем, что с ней полезно ознакомиться всем.

#### [Скачать файлы с примерами кода](#)

Пакет кода для книги размещен на GitHub по адресу <https://github.com/PacktPublishing/The-Definitive-Guide-to-Power-Query-M-/>. У нас также есть другие пакеты кода из нашего богатого каталога книг и видео, доступных на <https://github.com/PacktPublishing/>. Ознакомьтесь с ними!

#### [Скачать цветные изображения](#)

Мы также предоставляем PDF-файл с цветными изображениями скриншотов/диаграмм, использованных в этой книге. Скачать его можно здесь: [https://static.packt-cdn.com/downloads/9781835089729\\_ColorImages.pdf](https://static.packt-cdn.com/downloads/9781835089729_ColorImages.pdf).

## Глава 1. Знакомство с M

M – мощный и универсальный язык формул, специально разработанный для извлечения и преобразования данных. Термин M является неформальным обозначением. Официальное название M – язык формул Power Query. M – это язык, лежащий в основе Power Query, который используется во многих приложениях, таких как Microsoft Excel, Power BI, Power Platform и Microsoft Fabric, для преобразования и подготовки данных.

Популярность языка M постоянно росла в течение последнего десятилетия, и язык был интегрирован в набор инструментов и платформ Microsoft. Сегодня M и Power Query являются незаменимыми инструментами для специалистов по работе с данными.

Эта глава – начало увлекательного путешествия, кульминацией которого станет овладение языком M. Мы начнем с краткой истории M, а затем рассмотрим того, кто, где, почему и как должны изучать M. Далее мы познакомимся с основами языка, формальными и неформальными характеристиками M. Эта глава обеспечит прочную основу для более глубокого изучения языка M, которое можно найти в остальной части книги.

### *История M*

Процессу извлечения, преобразования и загрузки данных столько же лет, сколько и самим информационным технологиям. Исторически сложилось так, что и бизнес-пользователи, и ИТ-специалисты боролись с этой проблемой, и на протяжении многих лет для ее решения было разработано множество программных средств, таких как SQL Server Integration Services (SSIS) и Alteryx.

Многие из этих инструментов были сложными и язык M, и Power Query были созданы, чтобы упростить обработку данных.

Несмотря на то, что существует более спекулятивная история относительно происхождения M, мы можем, по крайней мере, определенно проследить происхождение M до проекта, первоначально носившего кодовое название Data Explorer. Data Explorer – это проект Azure SQL Labs примерно в 2011 году, направленный на упрощение процесса доступа, очистки и подготовки данных из различных источников. Язык запросов (Mashup language) поэтому и назван M.

В 2013 году корпорация Майкрософт выпустила Power Query в качестве надстройки Excel. В Power Query появился удобный интерфейс, позволяющий бизнес-пользователям выполнять преобразования данных с помощью визуального редактора. За кулисами Power Query использовал язык M в качестве базового языка формул для управления преобразованиями данных, и поэтому эти преобразования данных стали воспроизводимыми. Вместо того, чтобы бизнес-пользователям постоянно выполнять одни и те же ручные преобразования исходных данных, теперь этот процесс можно автоматизировать.

После успеха Power Query в Excel корпорация Майкрософт включила Power Query в свой новый продукт Power BI Designer, который был переименован в Power BI Desktop. По мере того, как Power Query набирал популярность, возникла необходимость в стандартизации базового языка

формул. В 2016 году корпорация Майкрософт представила спецификацию языка формул Power Query в Европейскую ассоциацию производителей компьютеров (ЕСМА), международную организацию по стандартизации. В результате была создана формальная спецификация языка, обеспечивающая совместимость и взаимодействие между различными реализациями.

Хотя язык формально назывался языком формул Power Query, в сообществе пользователей он стал широко известен как M. Неофициальное название M получило широкое признание и в настоящее время широко используется для обозначения языка.

Корпорация Майкрософт продолжает совершенствовать язык M в рамках своих постоянных инвестиций в технологии интеграции и преобразования данных. Периодически внедряются новые функции, возможности и улучшения, чтобы предоставить пользователям более мощные и эффективные способы обработки и подготовки данных. Кроме того, корпорация Майкрософт продолжает внедрять M в программные средства и платформы, такие как интеграция данных в Microsoft Power Platform и потоки данных в Power BI и Fabric.

Сегодня язык M является ключевым компонентом набора инструментов Microsoft для преобразования и интеграции данных. Распространение языка M, а также его универсальность и расширяемость делают его бесценным для специалистов по работе с данными.

### *Кому следует изучать M?*

M – это мощный инструмент для людей, которые работают с данными на регулярной основе:

*Аналитики данных*, которые занимаются извлечением, преобразованием и подготовкой данных. M предоставляет полный набор функций и операторов, которые позволяют аналитикам данных эффективно формировать данные из различных источников и управлять ими. Освоив M, аналитики могут автоматизировать повторяющиеся задачи, обрабатывать сложные преобразования данных и обеспечивать качество данных, что приводит к более точному и надежному анализу данных.

*Профессионалы в области бизнес-аналитики (BI)* могут значительно улучшить свои навыки, изучая M, основной компонент Power BI. Понимая M, специалисты по бизнес-аналитике получают возможность подключаться к различным источникам данных, выполнять сложные преобразования и создавать многократно используемые сценарии подготовки данных.

*Инженеры данных*, участвующие в проектировании и реализации конвейеров данных и процессах интеграции данных. M позволяет эффективно извлекать, преобразовывать и загружать данные из различных источников в хранилища данных, в частности, в Power BI и Microsoft Fabric. M также обеспечивает гибкость и мощь для работы со сложными форматами данных, определения пользовательских преобразований и создания эффективных рабочих процессов обработки данных. Освоив M, инженеры данных могут оптимизировать процессы интеграции данных и обеспечить согласованность и качество данных.

*Специалисты по обработке и анализу данных*, выполняющие исследовательский анализ данных, разработку моделей и расширенную аналитику. M предоставляет надежный набор функций для очистки, формирования и агрегирования данных, позволяя специалистам по обработке и анализу данных сосредоточиться на аналитических аспектах своей работы. Включив M в свои рабочие процессы подготовки данных, специалисты по обработке и анализу данных могут оптимизировать процесс превращения необработанных данных в аналитические сведения, тратя меньше времени на очистку и подготовку данных и больше времени на моделирование и анализ данных.

*Опытные пользователи Excel и Microsoft Power Platform*, которые активно работают с данными и выполняют сложные манипуляции с данными. M интегрирован в Excel с помощью Power Query, что позволяет выполнять расширенные преобразования данных в знакомом интерфейсе Excel. Кроме того, M интегрирован в Microsoft Power Platform, что позволяет преобразовывать данные и сопоставлять их между системами. Освоив M, опытные пользователи могут расширить свои возможности по работе с данными, автоматизировать повторяющиеся задачи, а также повысить точность и надежность анализа.

*Иные специалисты, связанные с данными* (помимо ролей, упомянутых выше), такие как менеджеры проектов, консультанты, исследователи и эксперты в предметной области. Владение

M дает возможность людям самостоятельно решать задачи, связанные с данными, извлекать информацию и принимать решения на основе надежных данных.

### *Где и как используется M?*

M – это универсальный язык, включенный в различные инструменты и платформы, где преобразование и манипулирование данными имеют важное значение. Его интеграция в экосистему Power Query позволяет пользователям использовать возможности M в различных средах. В этом разделе мы рассмотрим некоторые ключевые области, в которых M широко используется.

Для разработки M доступны два интерфейса: для локального использования, для облачных приложений:

- Power Query Desktop – это интерфейс для Power Query, который можно найти в классических приложениях, таких как Power BI Desktop и Microsoft Excel. Несмотря на то, что большинство функций аналогичны, есть и различия. Интеграция с искусственным интеллектом (AI), машинным обучением (ML), R и Python, присутствуют в Power BI Desktop и отсутствуют в Microsoft Excel. И наоборот, параметры структурированного столбца, доступные в Excel, недоступны в Power BI Desktop.
- Power Query Online – облачная служба, позволяет пользователям создавать и управлять преобразованиями данных в веб-браузере. M широко используется в Power Query Online для преобразований данных, подключения к источникам и выполнения сложных манипуляций с данными. Пользователи могут получать доступ к запросам M и редактировать их непосредственно в веб-интерфейсе, что делает их удобными для совместной работы и работы над задачами преобразования данных из любого места, где есть подключение к Интернету. Power Query Online интегрирован в различные продукты Майкрософт, включая службу Power BI, Power Apps, Power Automate и др.

Важно отметить, что, несмотря на существование двух различных интерфейсов для разработки M, оба они обеспечивают почти одинаковый пользовательский интерфейс. Более того, оба предоставляют возможность редактировать базовый код M, что является основной темой книги.

### *Продукты и услуги*

M повсеместно используется в экосистеме Майкрософт, включая следующее программное обеспечение и службы:

*Потоки данных (Dataflows)* – это независимые от продукта облачные запросы M, которые можно повторно использовать в нескольких различных продуктах. Потоки данных позволяют пользователям создавать процессы подготовки и преобразования данных, пригодные для многократного использования, и управлять ими. Потоки данных используют интерфейс Power Query Online.

*Power BI Desktop.* M – это фундаментальный компонент Power BI Desktop, ведущего средства бизнес-аналитики. M позволяет подключаться к различным источникам данных, выполнять преобразования данных и создавать интерактивные визуализации и отчеты. M позволяет извлекать, очищать и формировать данные из различных источников, таких как базы данных, файлы Excel, веб-службы и многое другое. С помощью M можно определять шаги преобразования данных и создавать многократно используемые запросы, которые автоматически обновляются при изменении данных в источнике.

В Power BI Desktop M используется в редакторе Power Query, подпрограмме, запускаемой из Power BI Desktop. Редактор Power Query предоставляет графический интерфейс (GUI) для работы с языком формул M:



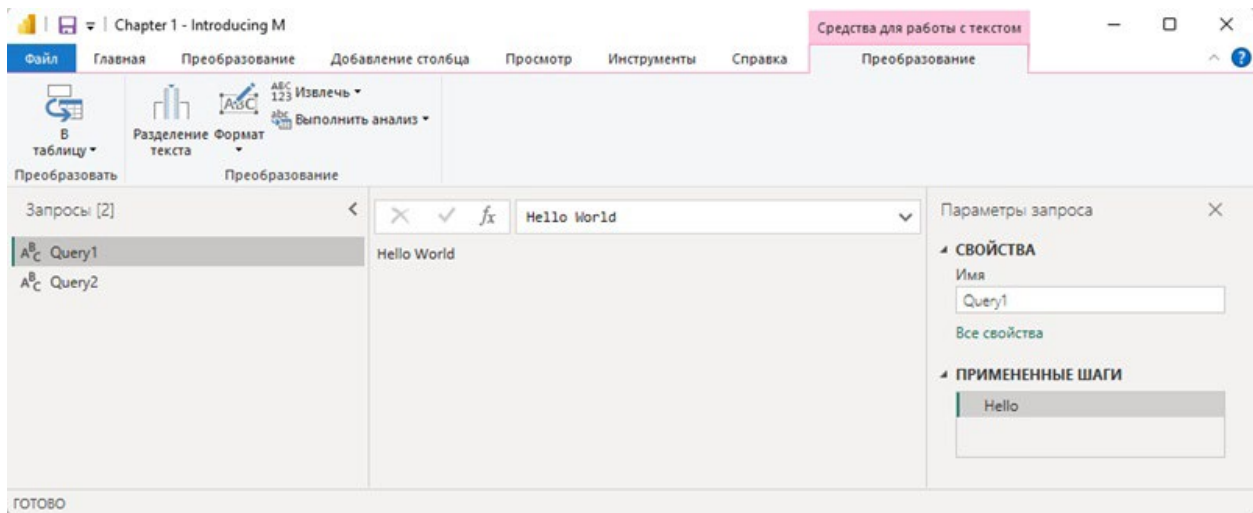


Рис. 1.1. Редактор Power Query в Power BI Desktop

Редактор Power Query более подробно описан в главе 2 Работа с Power Query/M. Power BI Desktop также поддерживает использование потоков данных.

Служба *Power BI/Fabric (powerbi.com)* – это облачный компонент Power BI, который позволяет предоставлять общий доступ к отчетам, панелям мониторинга и другому содержимому. Служба поддерживает использование кода M путем создания потоков данных с помощью интерфейса Power Query Online. Чтобы создать поток данных в службе Power BI, перейдите в любую рабочую область, отличную от My Workspace, и выберите *Создать*, а затем *Поток данных*:

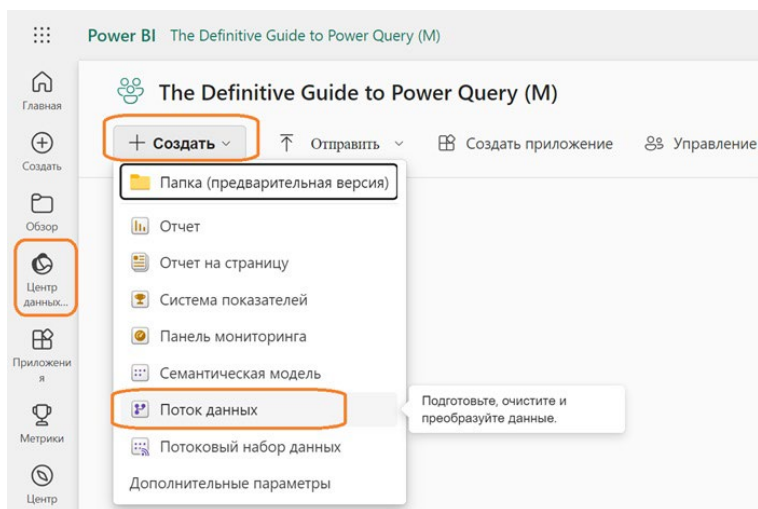


Рис. 1.2. Создание потока данных в службе Power BI

Сервер отчетов Power BI (PBRs) поддерживает интерфейс Power Query Desktop, позволяя создавать расширенные преобразования данных с помощью M.

*Excel (Windows u Macintosh)*. M интегрирован в Excel, позволяя выполнять расширенные преобразования данных в знакомом интерфейсе Excel. Power Query – механизм, лежащий в основе возможностей преобразования данных Excel, работает на базе M. Пользователи могут получить доступ к редактору Power Query в Excel для применения преобразований, фильтрации и сортировки данных, удаления дубликатов, слияния и добавления таблиц, а также выполнения других задач подготовки данных. M позволяет очищать, изменять форму и обогащать данные в Excel, повышая точность и надежность анализа.

В Excel для доступа к редактору Power Query перейдите *Данные* → *Получить данные*:

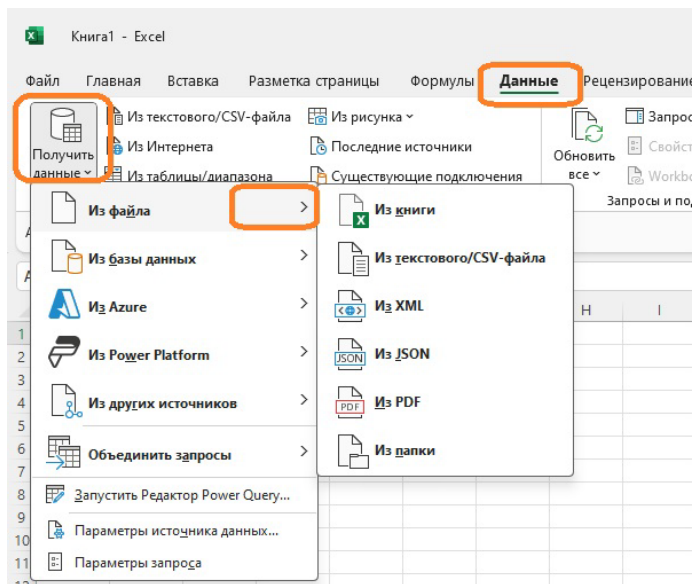


Рис. 1.3. Кнопка *Получить данные* в Microsoft Excel

После выбора источника данных доступ к интерфейсу редактора Power Query можно получить, нажав кнопку *Преобразовать данные*:

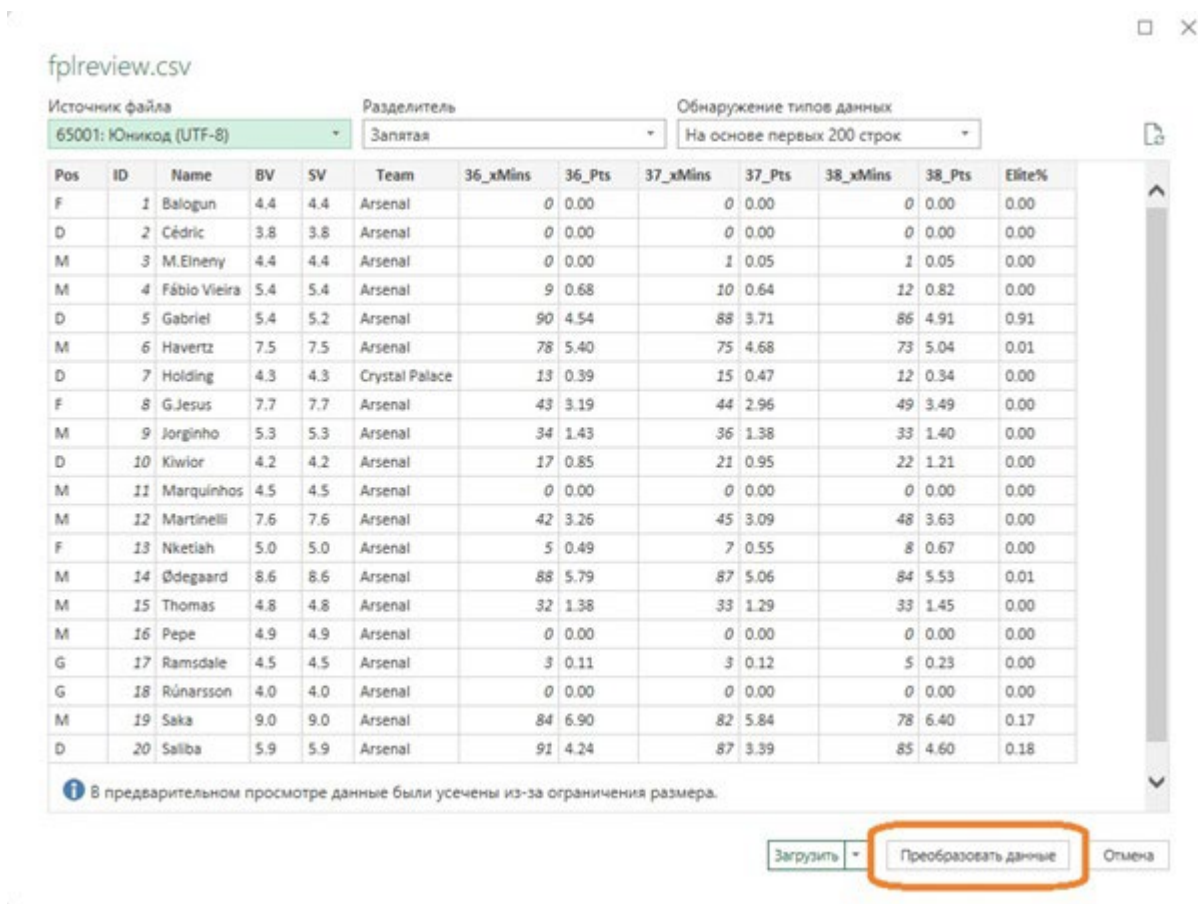


Рис. 1.4. Кнопка *Преобразовать данные* в Microsoft Excel

Версии Excel для Windows и Macintosh также поддерживают доступ к потокам данных.

*Power Apps* – это платформа Майкрософт с минимумом программирования для создания приложений. Поддерживается как интерфейс Power Query Online, так и использование потоков данных. Можно использовать M через интерфейс Power Query Online, либо через потоки данных, что позволяет переносить данные в Dataverse (формально Common Data Service).

*Power Automate* – это платформа Майкрософт с минимумом программирования для автоматизации рабочих процессов. Power Automate позволяет автоматизировать повторяющиеся

рабочие процессы и процессы, которые могут включать в себя операции с данными и задачи интеграции. M можно использовать в Power Automate для выполнения преобразований данных и обработки сложных сценариев в рамках автоматизированных рабочих процессов с помощью интерфейса Power Query Online. Включив M в Power Automate, могут создавать сложные решения для интеграции данных и автоматизации. Кроме того, потоки данных можно использовать в Power Automate с помощью коннектора потоков данных Power Query. Это позволяет выполнить действия после завершения потока данных (dataflow), а также обеспечивает возможность запуска потока данных в качестве действия в рамках потока Power Automate

*Фабрика данных* – это управляемая облачная служба, специально созданная для сложных extract-transform-load (ETL) и extract-load-transform (ELT) проектов интеграции. Фабрика данных позволяет создавать и координировать рабочие процессы на основе данных, перемещать и преобразовывать данные в большом масштабе. Фабрика данных Azure и фабрика данных в Microsoft Fabric поддерживают код M как через интерфейс Power Query Online, так и через потоки данных.

*SQL Server*: службы SSIS поддерживают ядро M, а SQL Server Analysis Services (SSAS) – Power Query Desktop.

*Customer Insights в Dynamics 365* – это платформа данных о клиентах (CDP) Майкрософт, которая обеспечивает целостное персонализированное обслуживание клиентов. Customer Insights поддерживает как потоки данных, так и интерфейс Power Query Online.

*Visual Studio* позволяет интегрировать M. Это делается с помощью Power Query Language Service for Visual Studio Code, доступной в Visual Studio Code Marketplace. Эта языковая служба предоставляет нечеткое автозаполнение, наведение, подсказки функций и другие опции для написания кода M в Visual Studio. Существует также пакет средств разработки программного обеспечения (SDK) Visual Studio Power Query. Этот пакет SDK состоит из набора средств, предназначенных для создания пользовательских коннекторов для источников данных Power Query. Пакет SDK для Visual Studio Power Query более подробно описан в главе 16 Включение расширений.

*Другие сценарии интеграции данных.* M не ограничивается вышеупомянутым программным обеспечением и сервисами. M также можно использовать в пользовательских приложениях и средах программирования, использующих библиотеки Power Query.

Навыки, полученные в этой книге, относятся к самому языку M и, таким образом, выходят за рамки использования конкретного продукта или услуги. Освоив M, вы получите возможность создавать многократно используемые процессы преобразования данных и расширите свои возможности по манипулированию данными в широком спектре сценариев, связанных с данными, а также в любом интерфейсе, продукте или сервисе, использующих M в качестве базового языка преобразования данных.

### *Зачем изучать M?*

В современном мире, управляемом данными, эффективное преобразование и анализ данных является ценным навыком. Power Query, мощный инструмент для преобразования и подготовки данных, завоевал огромную популярность благодаря бесшовной интеграции со многими популярными программными системами, а также простоте использования. В основе Power Query лежит M, язык формул Power Query. Но вы можете спросить себя, почему вы должны тратить время на то, чтобы учить M? Вот семь причин, по которым мы считаем, что специалисты по работе с данными и другие люди должны изучать M.

*Использование всей мощи Power Query.* В книге [Сбор, объединение и преобразование данных](#) Гил Равив пишет, что графический интерфейс для создания запросов M позволяет решить только 40% задач, связанных с преобразованием данных, но владение M позволяет приблизить этот показатель к 99,99%. В последующих главах демонстрируются конкретные примеры решения задач преобразования данных, которые не могут быть решены в графическом интерфейсе. Поскольку M служит основой возможностей преобразования данных Power Query, освоив M, вы получаете полный контроль над процессом преобразования данных.

Одна из основных причин, по которой вы должны изучить M, – *автоматизация повторяющихся задач* преобразования данных. Перед бизнес- и ИТ-специалистами часто стоит задача получать данные на регулярной основе, а затем создавать отчеты на основе этих данных. Вместо того, чтобы каждый раз вручную преобразовывать данные (часто в Excel), использование M позволяет реализовать логику преобразования данных один раз, а затем автоматически запускать ее каждый раз при получении новых данных.

*Гибкость и настройка.* Power Query предоставляет удобный интерфейс для работы с данными, но у него есть и ограничения. Изучив M, вы можете расширить возможности Power Query и преодолеть эти ограничения. M позволяет писать пользовательские функции, выполнять расширенные преобразования и применять сложную логику, выходящую за рамки встроенных возможностей интерфейса Power Query. Такая гибкость позволяет точно адаптировать преобразование данных к уникальным требованиям источников данных и их анализа.

*Оптимизация производительности.* M является высокоэффективным и оптимизированным языком для преобразования данных. Подсистема Power Query интеллектуально обрабатывает выражения M, оптимизируя производительность за счет сокращения ненужных загрузок данных и преобразований. При работе с большими наборами данных или сложными преобразованиями знание M позволяет писать эффективный код, который значительно ускоряет обработку данных. Понимая принципы M и его соображения по производительности, вы можете оптимизировать рабочие процессы обработки данных и сэкономить драгоценное время. Наконец, использование M может значительно сократить и упростить код формул и выражений анализа данных (DAX) в приложениях, таких как Excel и Power BI Desktop.

M предоставляет полный набор средств для *очистки и преобразования данных*. Функции M выходят далеко за рамки базовых операций, доступных в традиционных приложениях для работы с электронными таблицами. С помощью M можно легко решать проблемы качества данных, такие как удаление дубликатов, обработка отсутствующих значений, разделение столбцов, слияние наборов данных и выполнение расширенных вычислений.

M является не только самостоятельным языком, но и хорошо *интегрируется* с другими языками программирования, такими как SQL, R и Python. Такая интеграция позволяет использовать возможности этих языков в рабочих процессах Power Query. Вы можете комбинировать код M с собственными SQL-запросами, вызывать скрипты R или Python и включать внешние библиотеки и функции в процесс преобразования данных.

*Знание M может открыть новые карьерные возможности.* Владение M и Power Query стало востребованным навыком в индустрии данных. По мере того, как организации все больше полагаются на данные для принятия решений, люди, обладающие способностью эффективно преобразовывать, очищать и анализировать данные, пользуются большим спросом. Инвестируя время и усилия в изучение M, вы позиционируете себя как ценный актив для организаций, которые полагаются на аналитику на основе данных.

### *Основы языка M*

*Выражения и функции.* В M выражения образуют строительные блоки преобразования данных. Выражение представляет вычисление или операцию, результатом вычисления которой является значение. M предоставляет широкий спектр встроенных функций, которые можно использовать для выполнения операций с данными. Функции в M вызываются с использованием синтаксиса, где за именем функции следуют аргументы в круглых скобках. Например, функция [Text.Start](#)("Hello, World!", 5) возвращает подстроку *Hello* из входного текста.

*Типы данных.* M поддерживает различные типы данных: текст, число, дата, время, список, таблица, запись и др. Понимание типов данных в M имеет решающее значение для выполнения преобразований. M предоставляет функции для преобразования между различными типами данных и манипулирования данными на основе присущих им характеристик. Например, функция `Text.From` преобразует значение в текст, а функция `Date.Year` извлекает компонент `year` из значения `date` или `datetime`.

*Переменные и константы.* M позволяет определять переменные и константы для хранения и повторного использования значений во время преобразования данных. Переменные создаются в

выражении `let`, за которым следует список назначений переменных, разделенных запятыми. Переменные и константы помогают улучшить читаемость кода, обеспечить повторное использование и сделать сложные преобразования более управляемыми.

**Операторы.** M поддерживает множество операторов для выполнения математических вычислений, логических сравнений и манипуляций с текстом. Арифметические операторы (+, -, \*, / и т. д.) используются для числовых вычислений, операторы сравнения (>, <, = и т. д.) вычисляют логические условия. Оператор конкатенации & используется для объединения текстовых значений, добавления списков и таблиц или слияния записей.

**Пошаговый процесс преобразования.** M следует пошаговому процессу преобразования, где каждый шаг определяет операцию преобразования данных. Редактор Power Query предоставляет визуальный интерфейс для определения шагов и создает соответствующий код M. Шаги могут включать такие операции, как фильтрация строк, удаление дубликатов, разделение столбцов, слияние таблиц и агрегирование данных.

**Свертывание запросов** – это метод оптимизации в Power Query, который отправляет данные преобразования в источник данных, когда это возможно. При использовании M важно помнить о свертывании запросов, чтобы обеспечить эффективную обработку данных. Свертывание запросов может повысить производительность за счет сокращения передачи данных между источником данных и Power Query. Однако не все преобразования могут быть свернуты, поэтому важно понимать, какие операции можно свернуть, а какие нет. Например, при использовании режимов хранения таблиц Direct Query и Dual все запросы M должны быть свернуты, что может ограничить некоторые операции преобразования.

**Обработка и отладка ошибок.** M предоставляет механизмы обработки ошибок для перехвата и обработки исключений при преобразовании данных. С помощью таких функций, как `try`, `else` и `error`, можно управлять потоком выполнения и корректно обрабатывать потенциальные ошибки. Кроме того, M поддерживает возможности отладки, такие как возможность пошагового выполнения кода для выявления и устранения проблем в сложных преобразованиях.

**Чувствительность к регистру.** M чувствителен к регистру. Это относится ко всем функциям, выражениям, переменным и другим аспектам языка.

**Комментарии** в M соответствуют стилю комментирования языка C. Комментарии в строке кода идут за кодом и начинаются двойной косой чертой `//`. Комментарий следует до конца строки. Блоки комментариев используют `/*` при открытии и `*/` при закрытии. Они могут занимать одну или несколько строк.

### *Выражение let*

В основе языка M лежит выражение `let`, которое должно быть в паре с выражением `in`. Проще говоря, выражение `let` содержит входные данные и преобразования, в то время как выражение `in` содержит выходные данные. Простой *Hello World* для M выглядит следующим образом:

```
let
    Hello = "Hello World"
in
    Hello
```

Этот код вернет текст *Hello World*.

За каждым выражением в операторе `let` должна следовать запятая (,), за исключением последнего выражения перед выражением `in`. Таким образом, если выражение `let` состоит из нескольких подвыражений, то код может выглядеть следующим образом:

```
let
    Hello = "Hello",
    World = "World",
    Return = Hello & " " & World
in
    Return
```

Этот код также возвращает *Hello World*.



Понимание основ M, включая выражения, функции, типы данных, переменные, операторы и процесс пошагового преобразования, жизненно важно для эффективного манипулирования данными и их подготовки. Овладев этими основополагающими концепциями, вы получите возможность выполнять сложные преобразования, оптимизировать рабочие процессы с данными и раскрыть весь потенциал языка M. Остальная часть этой книги посвящена тому, чтобы помочь вам освоить эти основополагающие понятия и способы их применения.

### Характеристики M

M – это язык программирования, который служит основой Power Query, позволяя извлекать, очищать и изменять данные из различных источников, таких как базы данных, электронные таблицы, веб-страницы и многое другое. Однако, в отличие от языков программирования общего назначения, таких как C, C#, Java и Python, которые предназначены для широкого спектра приложений, M является предметно-ориентированным языком, специально разработанным для извлечения и обработки данных. Таким образом, M предоставляет богатый набор функций, операторов и выражений, которые позволяют выполнять сложные преобразования, вычисления и агрегирование данных. Чтобы лучше это понять рассмотрим характеристики M как с формальной, так и с неформальной точек зрения.

### Формальная классификация

Языки программирования классифицируются в соответствии с рядом свойств, таких как чистые/нечистые, низшего/высшего порядка, статически/динамически типизированные, сильно/слабо типизированные, с предварительной/ленивой оценкой и императивные/функциональные.

Microsoft описала M следующим образом.

Язык программирования считается *чистым*, если он обеспечивает ссылочную целостность. Другими словами, любое выражение может быть заменено значением этого выражения без изменения поведения или смысла программы.

Нечистый язык программирования допускает побочные эффекты, то есть действия, вызывающие изменения, выходящие за рамки возвращаемого значения функции. Язык M обычно используется для преобразования и извлечения данных, которые включают обращение к внешними источникам данных, выполнение операций с данными и создание выходных данных. Эти действия являются побочными эффектами, так как они влияют на состояние источника данных или создают выходные данные, выходящие за пределы возвращаемого значения функции. В то время как M предоставляет конструкции функционального программирования и поддерживает неизменяемость, позволяя создавать чистые функции, язык не является чисто функциональным из-за своей нечистой природы. Он включает в себя комбинацию функциональных и императивных парадигм программирования.

Для языков более низкого уровня, таких как машинный код или язык ассемблера, каждый оператор программирования соответствует одной инструкции для компьютера, в то время как в языках более высокого уровня каждый оператор соответствует нескольким инструкциям для компьютера. Языки более высокого уровня обычно позволяют использовать такие вещи, как функции, объекты и модули, в качестве значений в программе. Языки более высокого уровня часто рассматривают функции как граждан первого класса. В частности, это означает, что функции могут быть присвоены переменным, переданы в качестве аргументов другим функциям и возвращены в виде значений функций.

M поддерживает программирование высокого уровня, позволяя определять функции и управлять ими как значениями. Функции можно присваивать переменным, передавать в качестве аргументов другим функциям и возвращать функции в качестве результатов. Это позволяет создавать модульный и гибкий код за счет абстрагирования и повторного использования логики функций.

Обладая возможностями программирования более высокого уровня, M позволяет динамически применять преобразования и вычисления на основе входных параметров, потока управления и характеристик данных. Вы можете писать функции, которые работают с другими функциями, обеспечивая мощные сценарии манипулирования данными и преобразования.

Например, функции высшего уровня в М можно использовать для динамического применения серии преобразований к набору данных на основе определяемых пользователем критериев или для создания конвейеров функций для многократного использования для обработки данных.

Предоставляя возможности программирования более высокого уровня, М позволяет разработчикам писать выразительный и модульный код, упрощая работу со сложными преобразованиями данных и настраивая поведение функций в соответствии с конкретными требованиями.

Языки с динамической типизацией выполняют проверку типов во время выполнения, а не во время компиляции, как в случае со статически типизированными языками. Проверка типов – это проверка того, что аргументы функции имеют правильный тип, например текст, число или дату.

Языки со строгой типизацией чрезвычайно чувствительны к совместимости типов и требуют явных определений типов для переменных перед использованием. И наоборот, слабо типизированные языки, такие как М, не требуют явных определений типов, а некоторые даже выполняют автоматическое преобразование типов. М не так слабо типизирован, как, например, Python, поскольку переменные неизменяемы после вычисления. Таким образом, определение слабого типа для М обычно относится к возможности использования переменных, типы данных которых не были явно определены.

Рассмотрим следующий код:

```
a = 42
a = "Hello World"
print(a)
```

Этот код не выдаст ошибку в Python, даже если переменной *a* присвоены два разных типа данных. Однако подобный код невозможен в М, так как переменные после вычисления являются неизменяемыми (не могут быть изменены).

Гибкость типизации данных позволяет М работать с широким спектром источников данных и эффективно выполнять различные преобразования данных. Это упрощает процесс работы с разнородными наборами данных, которые могут содержать разные типы и структуры данных.

Важно отметить, что, несмотря на то, что М слабо типизирован, он все равно выполняет проверку типов во время выполнения, чтобы обеспечить согласованность операций. Если определенная операция несовместима с типом значения, во время выполнения может возникнуть ошибка. В целом, слабо типизированный характер М обеспечивает баланс между гибкостью и целостностью данных, предоставляя пользователям универсальный язык для задач преобразования данных.

Как правило, М следует стратегии активного вычисления (*eager evaluation*). Это означает, что при определении преобразований или вычислений эти преобразования и вычисления выполняются сразу после их применения к данным. Такой подход к оценке гарантирует, что преобразование данных происходит быстро, а результаты легко доступны для дальнейшей обработки или анализа. М предназначен для эффективной обработки задач обработки и извлечения данных, уделяя особое внимание немедленной оценке для предоставления обратной связи о преобразованиях в режиме реального времени.

Термин «частично ленивый» относится к специфической функции в М, называемой отложенным вычислением, которая отличается от общей стратегии оценки, присущей языку. В М отложенное вычисление применяется к выражениям в некоторых конкретных конструкциях, в частности, к выражениям *List*, *Record* и *Table*, а также к выражению *let*. Эти конструкции позволяют определять выражения, которые вычисляются только при необходимости, обеспечивая форму вычисления по требованию или отложенного вычисления.

Оценка выражений начинается с конца – с выражения, следующего за *in*. Далее по цепочке оцениваются выражения, внесшие вклад в выражение, следующее за *in*. Если строка кода не участвует в оценке выражения, следующего за *in*, то и вычисляться она не будет.

М позволяет определять необязательные аргументы для функций. Они вычисляются лениво, то есть они не вычисляются, если явно не используются в теле функции. Отложенное вычисление

необязательных аргументов помогает оптимизировать производительность, избегая ненужных вычислений для необязательных значений, которые на самом деле не используются в функции. Это гарантирует, что вычисления для необязательных аргументов будут отложены до тех пор, пока их значения не потребуются в ходе выполнения функции.

М также поддерживает конструкцию условного ветвления на основе операторов if-then-else. Вычисляется только та ветвь, которая соответствует условию, в то время как другая ветвь не вычисляется, что приводит к отложенному вычислению. Это отличается от стратегии активной (предварительной) оценки, где обе ветви оцениваются независимо от исхода состояния. Эта форма отложенного вычисления в конструкции if-then-else позволяет выполнять эффективные вычисления, избегая ненужных вычислений выражений в несовпадающей ветви.

М является *функциональным* языком программирования. Он поддерживает неизменяемость, функции высшего порядка, композицию функций, чистые функции и рекурсию.

*Неизменяемость* означает, что значения данных не изменяются на месте, а преобразуются в новые значения. Это позволяет избегать побочных эффектов.

М поддерживает функции более высокого уровня, рассматривая функции как значения первого класса. Вы можете передавать функцию в качестве аргументов другим функциям, возвращать функции из функций и сохранять функции в переменных.

М упрощает компоновку функций, позволяя комбинировать несколько функций для создания более сложных преобразований. Такая компоновка характерна для функционального программирования.

М способствует использованию чистых функций, которые не имеют побочных эффектов и выдают один и тот же выход для одного и того же входа. Чистые функции делают код более предсказуемым и упрощают устранение неполадок.

М поддерживает рекурсию, хотя и не так широко, как в некоторых других функциональных языках.

С точки зрения сравнения с другими языками, М, пожалуй, больше всего похож на F#, язык программирования, разработанный и реализованный Доном Саймом (Don Syme) из Microsoft Research, Кембридж, Великобритания.

### Неформальные характеристики М

Если говорить более неформально, то вот некоторые ключевые характеристики и особенности М.

М является *функциональным языком*, что означает, что он основан на концепции функций как основных строительных блоков для преобразования данных. Функции в М можно комбинировать, вкладывать и компоновать для выполнения сложных манипуляций с данными. М предоставляет более 700 встроенных функций, а также возможность создания пользовательских функций, адаптированных к вашим конкретным потребностям.

Обширный набор встроенных функций в М предоставляет пользователям мощные инструменты для обработки различных сценариев преобразования данных. Эти функции предназначены для упрощения типичных задач по обработке данных.

*Выразительный и удобочитаемый синтаксис.* Синтаксис М разработан так, чтобы быть интуитивно понятным и легким для чтения, что делает его доступным как для начинающих, так и для опытных программистов. Выражения М написаны в четкой и лаконичной манере, что облегчает создание сложных преобразований данных без ущерба для удобочитаемости. Синтаксис следует пошаговому подходу, позволяющему определить серию последовательных преобразований, применяемых к данным.

*Типы данных и значения.* М поддерживает различные типы данных, включая примитивные типы данных, такие как текст, числа, даты и длительность, а также структурированные типы данных, такие как списки, таблицы и записи. М также позволяет определять переменные, константы и параметры и работать с ними для хранения и повторного использования промежуточных результатов в процессе преобразования данных.

*Интеграция с редактором Power Query.* M легко интегрируется с редактором Power Query, предоставляя удобный интерфейс для взаимодействия с кодом M и его разработки. Редактор Power Query позволяет визуально создавать шаги преобразования данных, просматривать результаты и автоматически создавать код M. Он предоставляет надежную среду разработки, в которой можно писать и вести отладку кода.

*Расширяемость и кастомизация.* Одной из отличительных особенностей M является его расширяемость. Помимо того, что M предлагает широкий спектр встроенных преобразований, также можно создавать пользовательские преобразования в соответствии с вашими конкретными потребностями. С помощью M можно определять собственные функции, писать многократно используемые фрагменты кода и создавать расширенную логику манипулирования данными. Этот уровень настройки позволяет обрабатывать сложные сценарии, которые не охватываются стандартными преобразованиями, доступными через пользовательский интерфейс.

*Оптимизация производительности.* M оптимизирован для повышения производительности, обеспечивая эффективную обработку данных даже при работе с большими наборами данных. Подсистема Power Query интеллектуально оценивает и оптимизирует выражения M, чтобы свести к минимуму загрузку данных и преобразования.

Одним из конкретных методов оптимизации производительности является *семантическое потоковое выполнение* (streaming semantically), которое является свойством выражений *List* и *Table*. Семантическое потоковое выполнение включает повторное перечисление строк таблицы или элементов списка. Вместо того чтобы проходить через таблицу или список для каждого преобразования данных, при семантическом потоковом выполнении каждая строка таблицы или элемент списка оценивается для всех преобразований данных, и результаты собираются в качестве части вывода для выражения. Семантическое потоковое выполнение позволяет преобразовывать наборы данных, которые не помещаются в доступную память.

Еще один метод оптимизации производительности называется *свертыванием запросов*. Однако свертывание запросов не является свойством самого языка M. Свертывание запросов используется в Power Query для отправки или свертывания преобразований данных обратно в исходные базы данных. По сути, выражения в M преобразуются в эквивалентные операторы преобразования, доступные в исходных системах, таких как SQL Server. Это возвращает обработку преобразований в исходные системы из клиентской системы, выполняющей запрос M. Это может повысить производительность и эффективность за счет минимизации передачи данных из источника в Power Query.

### *Саммари*

Язык M является основой возможностей преобразования данных Microsoft Power Query. M широко используется в более широкой экосистеме Microsoft, включая различные программные приложения и веб-службы. Изучение M может быть полезно широкому кругу специалистов по работе с данными. Освоив M, вы получите возможность эффективно обрабатывать сложные преобразования данных, настраивать рабочие процессы обработки данных и оптимизировать производительность, тем самым используя весь потенциал Power Query для обработки данных.

В этой главе мы рассмотрели основные вопросы о том, кто, что, где, почему и как использует M, а также дали краткое введение в фундаментальные аспекты языка M. В следующей главе мы познакомим вас с основным интерфейсом, в котором вы будете писать код M.