KARPOV.COURSES >>>
КОНСПЕКТ

> Конспект > 4 урок > Подключение данных к Tableau

> Row/Column Security
> Подключение к Clickhouse

> Подключение и типы данных

Наша лекция и конспект рассказывают о возможностях платной версии Tableau, с которой вы можете столкнуться в своей дальнейшей работе. У вас есть возможность использовать триал Tableau, который ограничен периодом 14 дней.

Процесс подключение к данным в Tableau выглядит так же, как и в любой другой системе работы с данными. При запуске Tableau, в левом столбце вам будут указаны доступные типы подключений. Для подключения к данным можно использовать четыре способа:

Подключение к источнику данных на Tableau Server

Вы используете заранее созданные подключения к базе данных и опубликованные на сервере (Tableau server). Кликнув, вы переходите во внутренний интерфейс, где указаны доступные источники данных на сервере. Подключение к ним доступно по клику, доступы и авторизация вшиты внутрь, и вы можете выбрав таблицу, подключиться к ней.

Подключение к файлу

Пользователю доступны форматы и расширения (excel, csv, json, pdf и другие), которые поддерживает Tableau и встроенный инструмент парсинга. Такой тип подключения удобен, если вам необходимо совместить данные из внешней базы данных и локальной таблицей на вашем рабочем месте (ноутбуке).

Подключение к серверу

Это возможность подключения к различным базам данных. В начале этого списка будут показаны часто используемые вами виды подключений. По клику на строку ***More***, вы можете посмотреть весь список предлагаемых вариантов. В последней версии Tableau есть так называемые Installed Connector - драйвера, которые умеют соединять Tableau с какой-нибудь базой данных и Additional Connectors (дополнительные) - драйвера, представленные в галерее, не установленные на ваш компьютер и которые Tableau изначально не прописало коннектор.

Кроме баз данных, Tableau предоставляет возможность подключения к облачным хранилищам.

Выбирая необходимый коннектор, вы заполняете параметры доступа и подключаетесь. После этого у вас появляется база данных и таблицы, доступные пользователю. Необходимую таблицу можно перетащить из левого столбца в правый либо можно создать custom SQL или Union.

После подключения к БД, у вас отображается интерфейс, знакомый вам по Tableau Public:

- область сборки данных (сверху);
- данные и значения, отображаемые в таблице (нижняя часть)

Сохраненные источники

Если вы сделали какой-либо источник и используете его регулярно, он сохраняется у вас для быстрого доступа. Довольно удобно и экономно по времени.

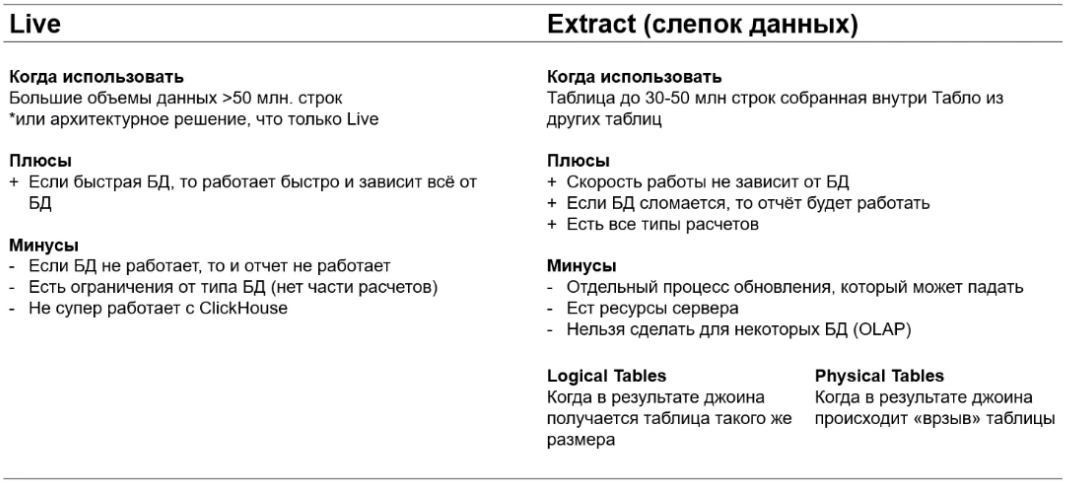
Типы подключений

В Tableau есть два способа подключения к БД:

- Live режим;
- Extract (слепок данных)

У каждого из типов подключения есть свои сильные и слабые стороны, использование которых зависит от ваших целей, навыков и использования архитектуры.

Типы подключения



Live	Extract (слепок данных)
Когда использовать Большие объемы данных >50 млн. строк *или архитектурное решение, что только Live	**Когда использовать** Таблица до 30-50 млн строк собранная внутри Табло из других таблиц
Плюсы + Если быстрая БД, то работает быстро и зависит всё от БД	**Плюсы** + Скорость работы не зависит от БД + Если БД сломается, то отчёт будет работать + Есть все типы расчетов
Минусы - Если БД не работает, то и отчет не работает - Есть ограничения от типа БД (нет части расчетов) - Не супер работает с ClickHouse	**Минусы** - Отдельный процесс обновления, который может падать - Ест ресурсы сервера - Нельзя сделать для некоторых БД (OLAP)
	Logical Tables Когда в результате джоина получается таблица такого же размера **Physical Tables** Когда в результате джоина происходит «врзыв» таблицы

> Объединения, настройки и типы подключения

После подключения к данным, у нас есть возможность выбрать тип соединения - найти его можно в правом верхнем углу.

При выборе Live режима, при каждом действии с листом будет происходить отправка запроса к базе данных.

При выборе Extract, вам необходимо будет указать Tableau куда вы хотите его (слепок данных) сохранить. При работе с большим объемом данных, например в течении дня, по умолчанию вам будет предложено сохранить это в папку Datasources в формате .hyper, который обеспечивает высокое качество сжатия.

В левой колонке, рядом с названием источника данных визуально показан режим подключения: Live или Extract (один или два бочонка).

На вкладке Data source, в правом верхнем углу, при выборе режима Extract у пользователя появляется кнопка Edit (редактировать) и Refresh (обновить).

В кнопке Edit живут настройки:

- логическая это или физическая таблица;

- фильтры, например по городу или гео;
- агрегация данных на уровне Extract - это оптимизирует скорость и объем отчета;
- Number of Rows;
- Incremental refresh - она позволяет делать Extract не полностью, а по инкременту;
- Hide all unused fields - при созданной визуализации, пользователь возможно использовал не все поля, эта настройка позволяет скрыть неиспользуемые данные, что оптимизирует работу Extract'a;
- History - история выполнения Extract'ов, вспомогательная информация, которая позволяет восстановить порядок совершения операций или логическую цепочку действий пользователя.

Всеми перечисленными настройками необходимо пользоваться для оптимизации размеров Extract'ов.

Еще одна классификация типов подключений в Tableau:

- Custom SQL - это возможность при подключении к БД использовать необходимую таблицу, а поверх этой таблицы использовать SQL код;
- View/Table

Типы подключения

Custom SQL	View/Table в БД
Когда использовать Делаете эдхок отчет и он будет на Экстракте **Плюсы** + Удобно ~~наговня...~~ быстро собрать данные **Минусы** - Доп. нагрузка на сервер - Сложно дебажить и искать ошибки - Нельзя переиспользовать	**Когда использовать** Понимаете какие точно данные нужны для дашборда. Может работать на лайве, но лучше материализованную таблицу. **Плюсы** + Можно переиспользовать в разных местах + Код храниться в гите + Не грузит Табло сервер **Минусы** - Лень делать)

Custom SQL настраивается на вкладке Data source, в левой колонке рядом с источником данных у вас будет строка New Custom SQL, которую вы перетаскиваете слева в верхнюю часть вкладки, в область создания данных.

У вас появляется редактор, куда вы можете вставить SQL код. Tableau дает возможность перенести часть трансформации на сторону BI-инструмента.

Минусы редактора:

- здесь нет подсветки кода, и это неудобно;
- эту трансформацию никто кроме вас не видит, это потенциальные риски.

Но основной плюс - можно быстро сделать какой-нибудь источник (ad-hoc).

В редактор можно вшить параметр, который по действию пользователя может модифицировать базу данных. При определенных навыках, это дает скорость и гибкость.

Минусы Custom SQL - Tableau его не оптимизирует при передаче в БД. Поэтому если вы делаете простую операцию или выбираете набор полей - делайте это в визуальном редакторе Tableau, код будет оптимизирован под решение вашей задачи.

> Типы объединений: Join, Union и Relation в Tableau

Tableau позволяет делать часть ETL процесса на стороне BI. Это можно использовать, когда аналитические отчеты нужно собрать быстро (ad-hoc).

> Join

По традиции, мы используем набор данных Sample superstore. В этом датасете есть таблица заказов (Orders), она является основной. На вкладке Data source, мы переносим ее из левой колонки в верхнюю часть вкладки, в область создания данных.

Для создания Join, необходимо зайти в таблицу заказов, в верхней части вкладки, двойным кликом. Затем мы объединяем заказы, перетаскивая таблицу people в верхнюю часть вкладки.

Если поля называются одинаково - Tableau автоматом создаст Join по этим полям. Если этого не произошло, пользователь может сам выбрать поля, по которым необходимо создать объединение.

Пользователь может выбрать тип Join'ов, можно донастроить на совпадение или не на совпадение. На этапе создания Join'ов вы можете задать предобработку данных.

Когда мы делаем Join, при выборе Extract'a у нас есть возможность настроить физического или логического Join'a. Как это работает:

- если мы делаем логические таблицы, Tableau сделает Join, и после этого сделаем Extract (данные сперва соединяются, а затем делается их слепок);
- физические таблицы работает по другому принципу - мы делаем слепок с обеих таблиц и после этого делаем Join, операция соединения происходит после и каждый раз при работе с этими таблицами.

> Union

Эта операция поможет вам, если стоит задача соединить несколько таблиц.

Первый способ - на вкладке Data source, мы переносим первую таблицу из левой колонки в верхнюю часть вкладки, в область создания данных, два раза кликаем и

кидаем вторую таблицу под первую. Так создается union таблиц

Второй способ - на вкладке Data source, в левой колонке строку New Union переместить в верхнюю часть, в область создания данных и он откроет вам окошко, куда можно накидать необходимые вам таблицы.

В новом окошке самая классная функция - создать автоматический Union по заранее заданному правилу. Например по названию листа, книги или по названию таблицы в БД.

> Relations

Relations (связи) - новый способ, который позволяет легко анализировать данные из нескольких таблиц на разных уровнях детализации.

Relations (связи) объединяют таблицы и агрегируют данные во время анализа, запрашивая данные на необходимом уровне детализации на основе полей и фильтров. Они позволяют уверенно комбинировать таблицы, разрешая многие сценарии дублирования данных и гарантируя, что вы получите точные результаты, не полагаясь на вычисления LOD.

Relations (связи) легко создавать. При подключении к данным в Tableau необходимо на вкладке Data source, перенести из левой колонки в верхнюю часть вкладки, в область создания данных таблицы и объединить их по необходимым вам параметрам (колонкам).

Здесь вы можете увидеть пример создания Relations

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c6c674fb-9517-4976-bd70-b5cd176188e6/.mp4

В примере с набором данных Sample superstore, для связки заказов и плана, мы создадим расчетные поля, для связки их по датам.

В таблице заказов даты указаны по дням, а в таблице планов - по месяцам. Чтобы между ними установить связь, нам понадобиться для таблицы заказов функция:

```
DATE(DATETRUNC ("month", [Order Date]))
```

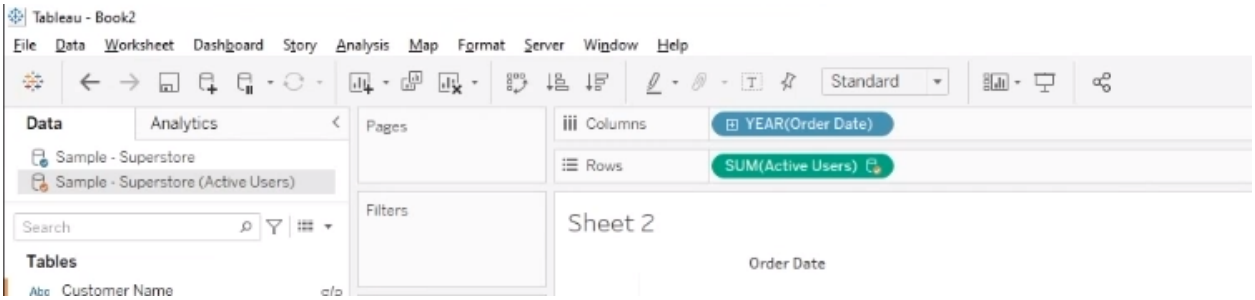
> Blending

В Blending логика очень похожа на Relations, но отличается сам подход и возможность настройки. Это подмешивание данных из разных источников в одни и те же дашборды.

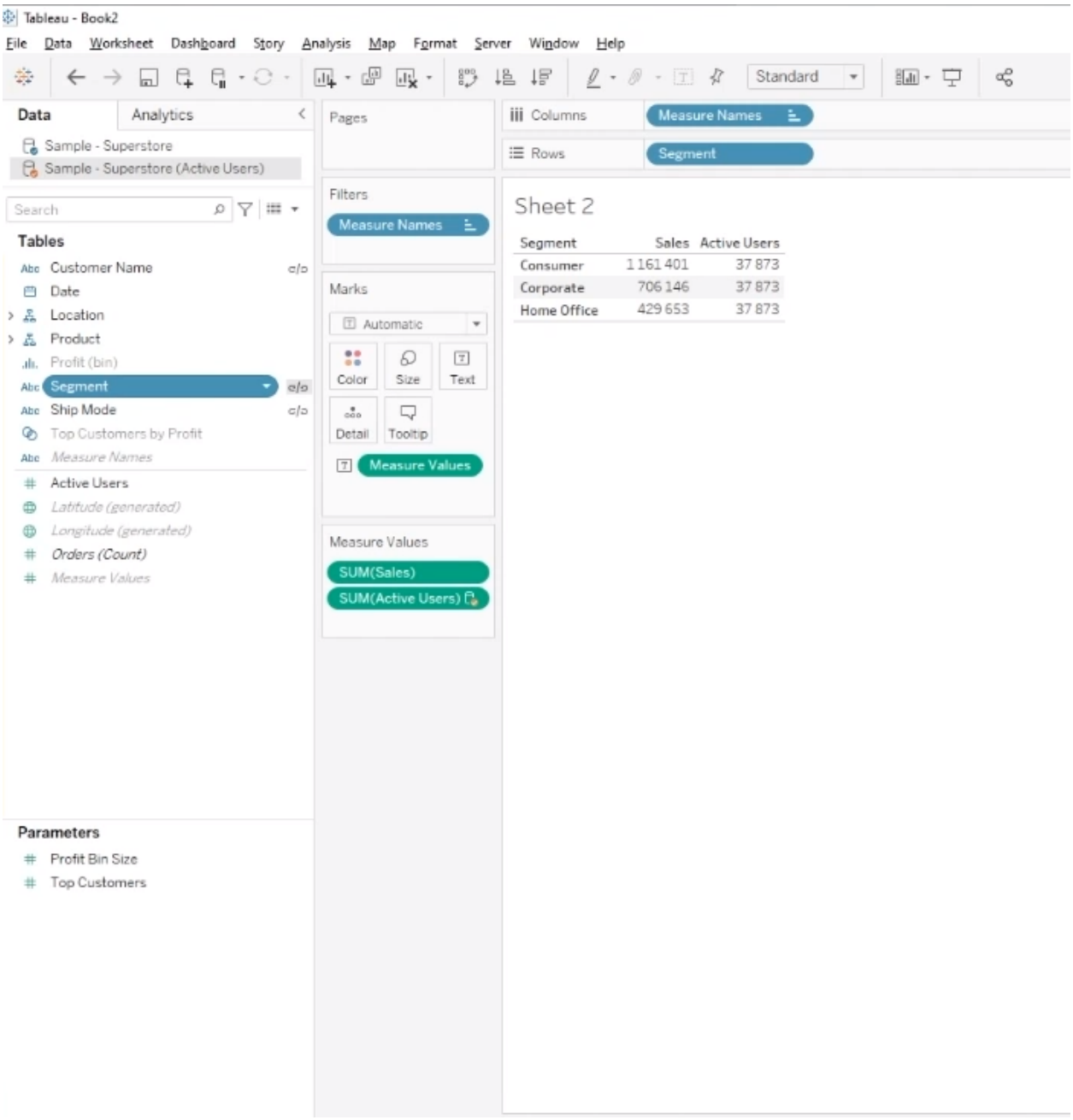
Предположим, у нас есть данные по продажам в разрезе регионов и в ассортименте. И есть данные по посещаемости нашего сайта, в каждой категории продукта. Эти данные между собой не связаны, но мы хотели бы их отображать и фильтровать одновременно.

Для правильного применения Blending'a важно, какой источник будет главным - это задается в начале вашей работы, когда вы забираете первое поле из имеющихся у вас источников данных. Второстепенный источник данных визуально помечается, чтобы пользователь видел иерархию при построении дашбордов.

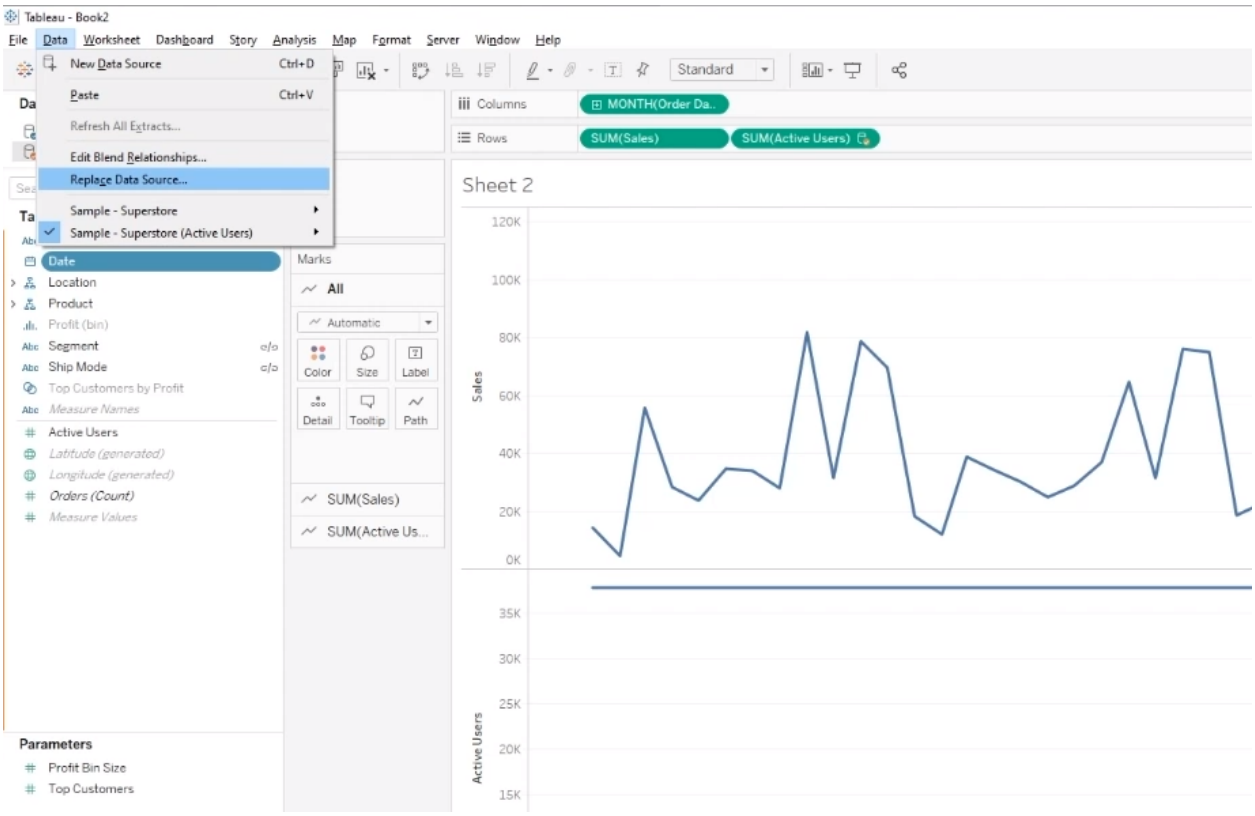
На примере ниже, синим в графе Columns выделен первостепенный (главный) источник, зеленым цветов в графе Rows выделен второстепенный источник данных.



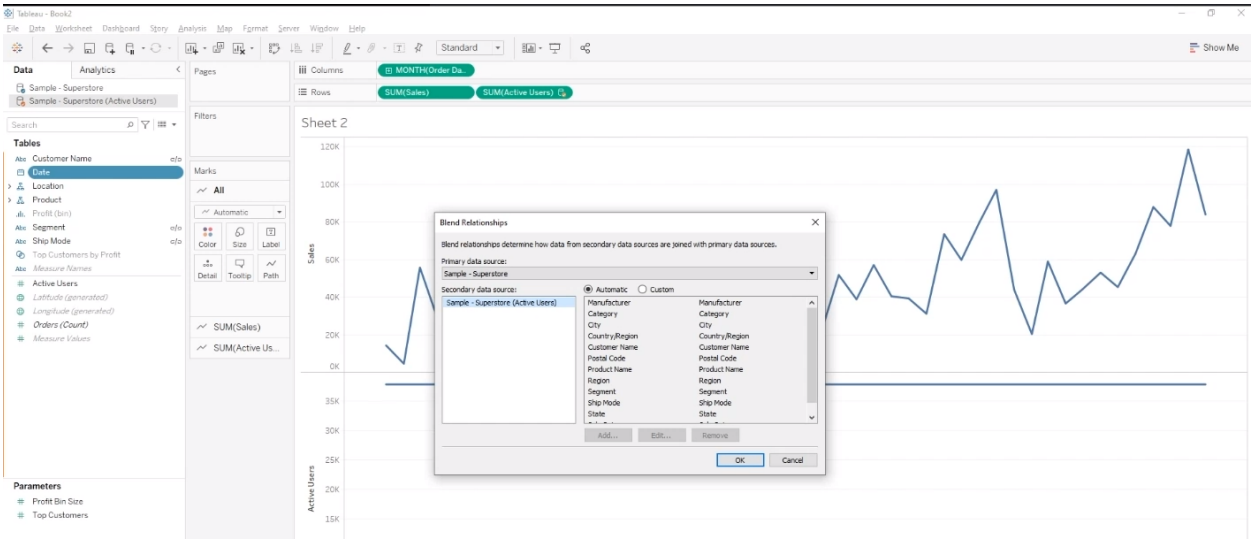
При построении дашбордов, в левой колонке Tables вы увидите иконку звеньев цепи (цепочку), которая обозначает связь между источниками данных.



Если стоит задача сравнить два равнозначных показателя из разных источников, у вас есть два способа. Первый - переименовать эти показатели, чтобы Tableau автоматически выстроил между ними связи (например сделать это сразу в Tableau). Второй способ - настроить эти значения через функционал **Data** > **Edit Blend Relationships ...**



В открывшемся окне можно выбрать, по каким полям ваши данные должны быть связаны. После настройки новой связи, в левой колонке **Tables** рядом со связанным полем появится иконка звеньев цепи (цепочка). Теперь это связанное поле вы можете использовать в своей работе.



В чем разница между Blending и Relations? Есть два кейса.

Первый: Blending настраивается для каждого листа (в Tableau) отдельно, Relations настраивается на уровне источника данных. Это дает гибкость в построении дашбордов.

Второй кейс - у вас есть два независимых источника данных, по ним есть графики и вам не нужно делать никаких операций между ними. Но при построении общего дашборда, пользователи хотят использовать один фильтр для двух разных графиков. Это дает удобство пользователям, Relations работал бы медленно и долго, создавая неудобство для пользователей, а Blending работает нормально.

> Особенности и преимущества различных типов объединения данных.

Типы объедения данных

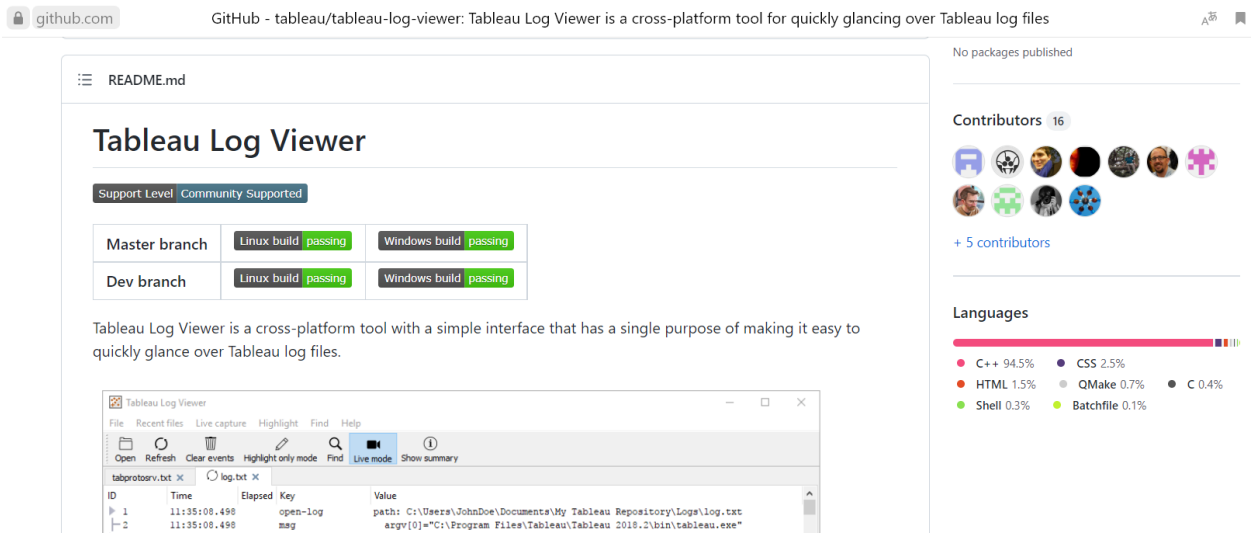
Join	Relations	Blending
Когда использовать В таблицах данные одной и той же гранулярности, джоин создаёт столько же строк и не взрывает данные. Данные из обоих таблиц используются почти во всех графиках или нужно делать сложные расчеты между полями обоих таблиц.	**Когда использовать** Таблицы с раной гранулярностью данных. Данные из обоих таблиц используются почти во всех графиках.	**Когда использовать** Таблицы с раной гранулярностью данных. Данные из одной таблицы используется только в небольшом количестве графиков. Между таблицами только самые простые расчеты или только фильтрация.
Плюсы + Понятно и железно работает + Работает быстро	**Плюсы** + Работает нормально по скорости	**Плюсы** + Очень гибко настраивается — для каждого графика
Минусы - Плохо работает при данных разной гранулярности	**Минусы** - Замедляет работу - Иногда странное поведение при работе	**Минусы** - Сильно замедляет работу - Много ограничений при расчетах - Лучше всего чтобы было полное совпадение по полям между обеими таблицами

Мы разобрались, какие соединения можно делать на стороне Tableau. Это гибкость BI инструмента предполагает, что пользователи могут делать изменения, которые повлияют и на вашу работу. Поэтому важно проговорить и обсудить порядок взаимодействия со всеми пользователями Tableau, чтобы разграничить ответственность и повысить уровень знания о data процессах в вашей компании.

> Как работает Tableau "под капотом"

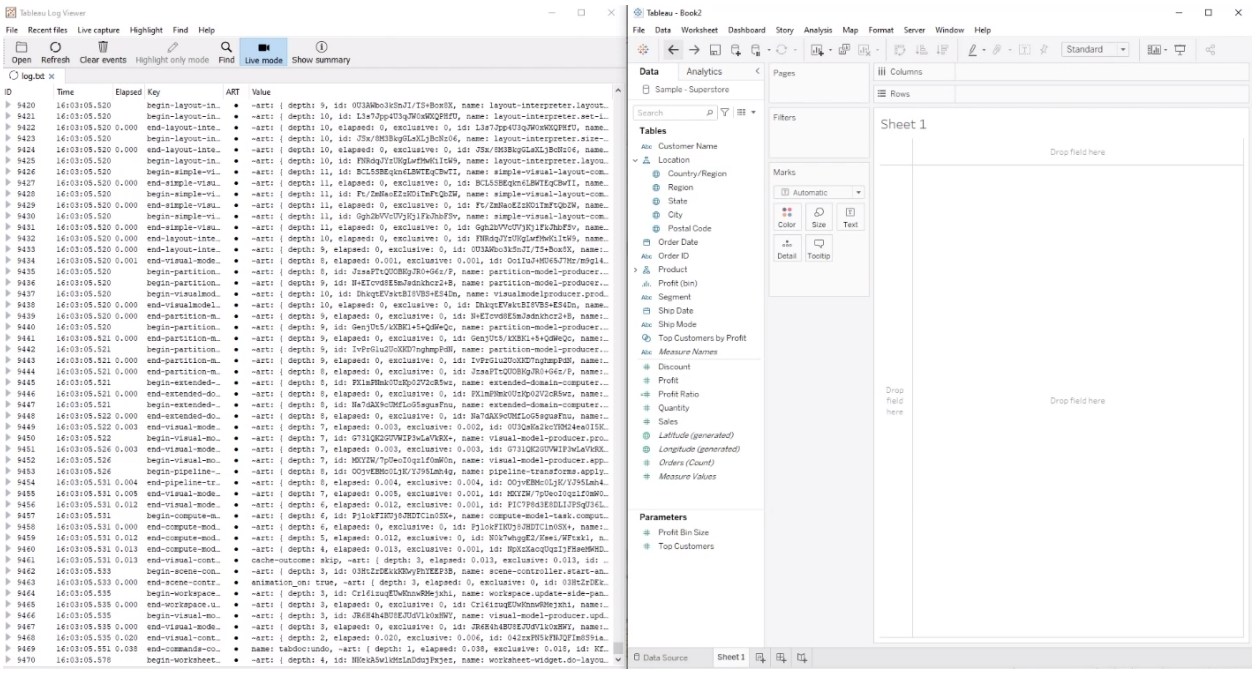
Для удобства рекомендую использовать утилиту Tableau log viewer

Это дистрибутив, который позволяет парсить логи Tableau Desktop.



Логи хранятся в папке My Tableau Repository > Logs. Здесь можно видеть все действия, которые производит программа во время работы.

При открытии https://github.com/tableau/tableau-log-viewer, вы подключаетесь к файлу, который был последним в работе. Дальше можно включить Live mode - это означает, что действия теперь будут логироваться.



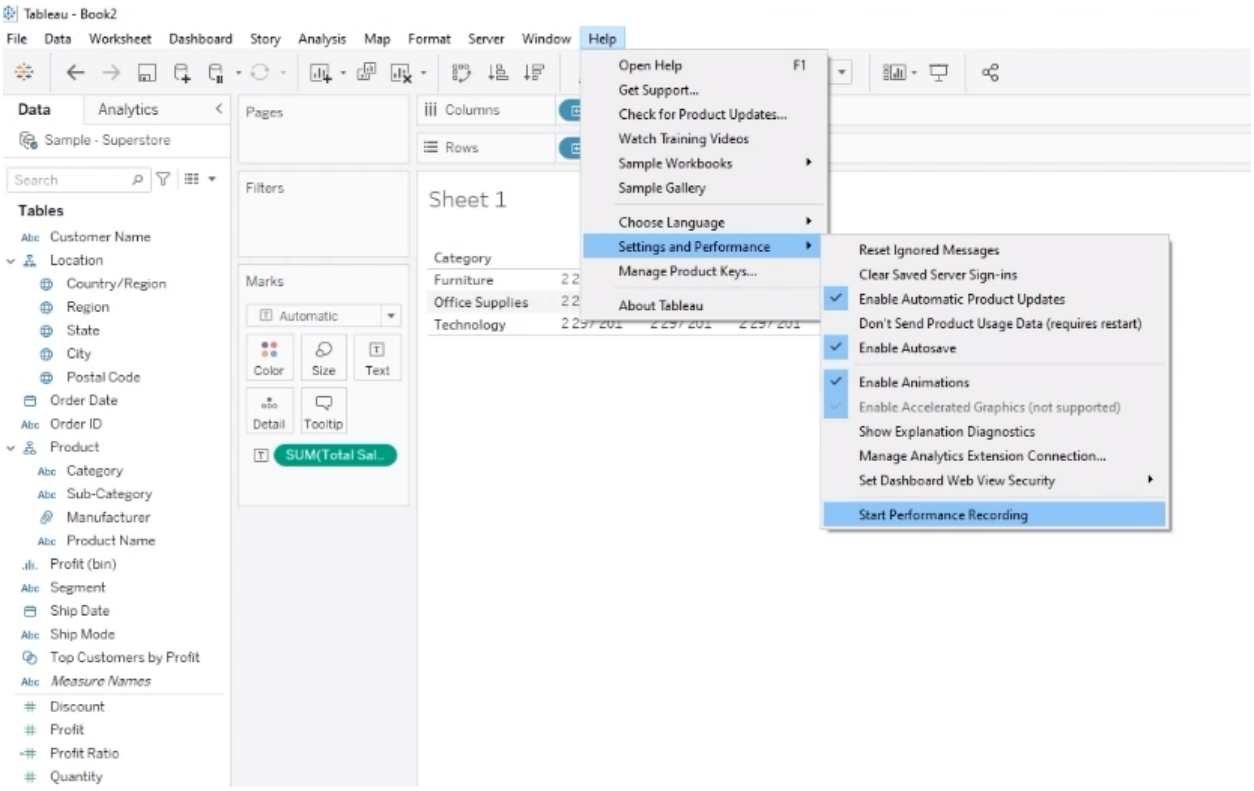
Вы делаете Extract и Refresh, удаляя кэш на стороне Tableau, очищаете ивенты и дальше можете смотреть, что происходит когда вы перетаскиваете какие-то поля. Здесь нам важно найти те запросы, которые формируются при формировании листа.

Для этого необходимо воспользоваться функцией Highlight... и найти строки с запросом `query compiled`.

Эти строки позволяет вам оценить, какие операции проходят с запросом к БД, а какие Tableau проводит на своей стороне, например Quick Table Calculation (быстрые табличные вычисления).

Такое логирование позволяет проверять, какие операции происходят "под капотом" Tableau. Этот инструмент показывает, что оптимизация возможна, а сам порядок операции и визуализации в данном BI компилируются в какой-то SQL код, который может работать не совсем очевидным образом.

Аналогичный инструмент доступен и в Tableau. Он называется Start Performance Recording, располагаясь в разделе Help > Settings and Performance.



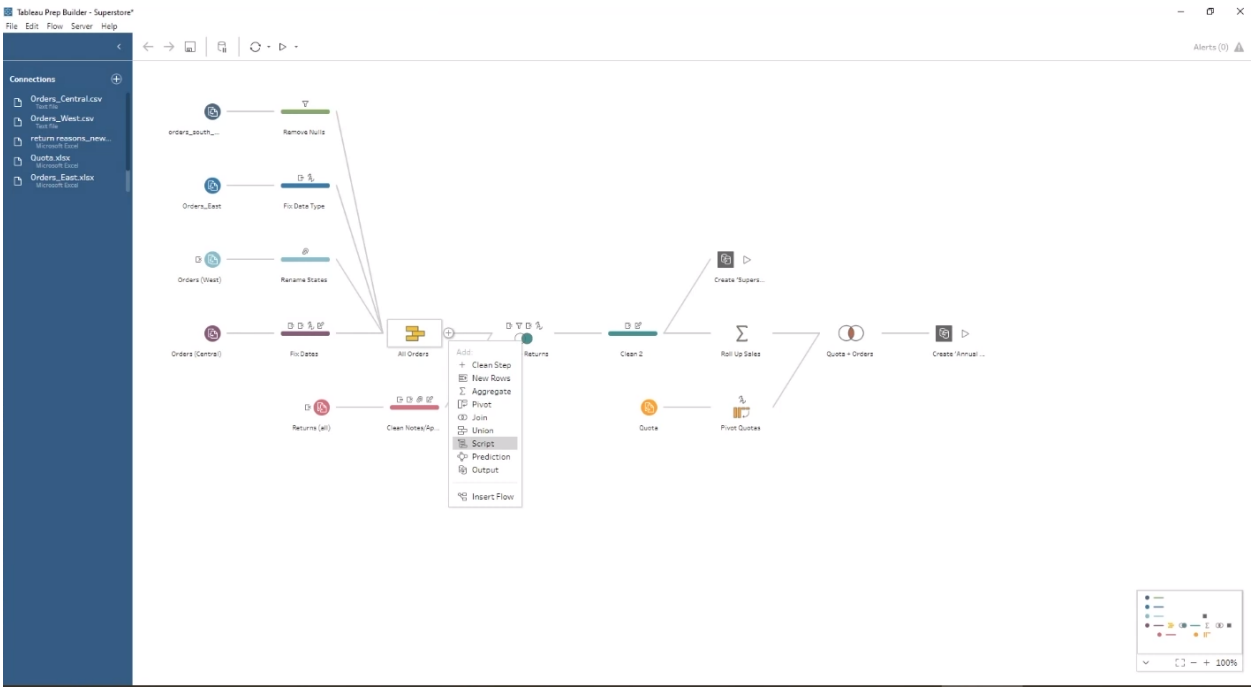
Он позволяет рассчитывать скорость загрузки визуализаций и строится на тех же логах, но не дает возможность посмотреть сами логи Tableau. После запуска Start Performance Recording появляется еще одна книга, в которой можно посмотреть временную шкалу ивентов и события.

> Tableau Prep и Data Management Add-on

Tableau Prep

это визуальный ETL инструмент для подготовки, очистки, слияния и загрузки данных в BI систему. Это отдельный инструмент на декстопе и на сервере.

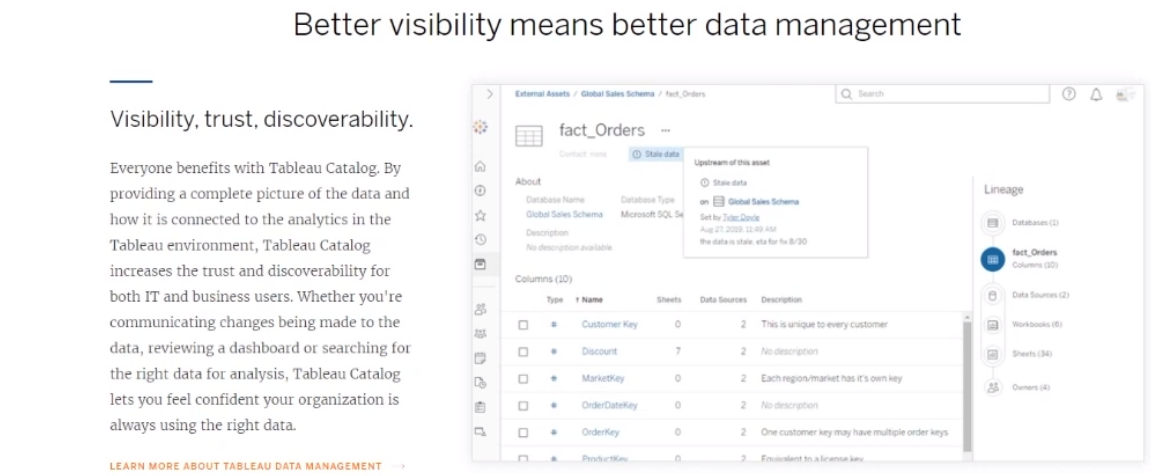
Вот так он выглядит и может быть знаком вам, если вы работали, например с Alteryx.



Это "молодой" инструмент, относительно ненадежный, активно развивается в экосистеме Tableau и подходит для небольших и среднего размера команд. Он умеет исполнять flow обработки данных и сразу их публиковать на сервер. что дает возможность обновлять Extract'ы.

Tableau Catalog

Он позволяет управлять данными на стороне сервера, т.е. делать Data Governance - выполнять стратегию для эффективного управления вашими (корпоративными) данными.



На Tableau Server появляется возможность посмотреть к какому источнику подключен ваш отчет и какие дальше идут соединения. Если вы работаете в связке Tableau Server, Tableau Prep и Tableau Catalog - это крутая альтернатива, которая дает вам комплексное решение.

Data Management Addon

Это набор функций и возможностей, которые помогают клиентам управлять содержимым Tableau и данными в их среде Tableau Server.

К этому относится возможность публиковать источники данных - встроенные в книгу или опубликованный на сервере.

Типы подключения

Встроенный в книгу	Опубликованный на сервер
Когда использовать Эдхок отчет с небольшим количеством данных. Нужна максимальная скорость работы.	**Когда использовать** Большой экстракт, который сложно скачивать каждый раз. Или подключение которое нужно переиспользовать для большого кол-ва книг.
Плюсы + Работает быстрее всего	**Плюсы** + Не надо скачивать гигабайты с сервера + Можно подключить к нескольким книгам
Минусы - Дублирование данных и ресурса сервера, если эти же данные используются в другом отчете - Много экстрактов на сервере	**Минусы** - Сложнее администрирование и работа с источником - +1-1.5 секунды к скорости загрузки отчета на каждый источник

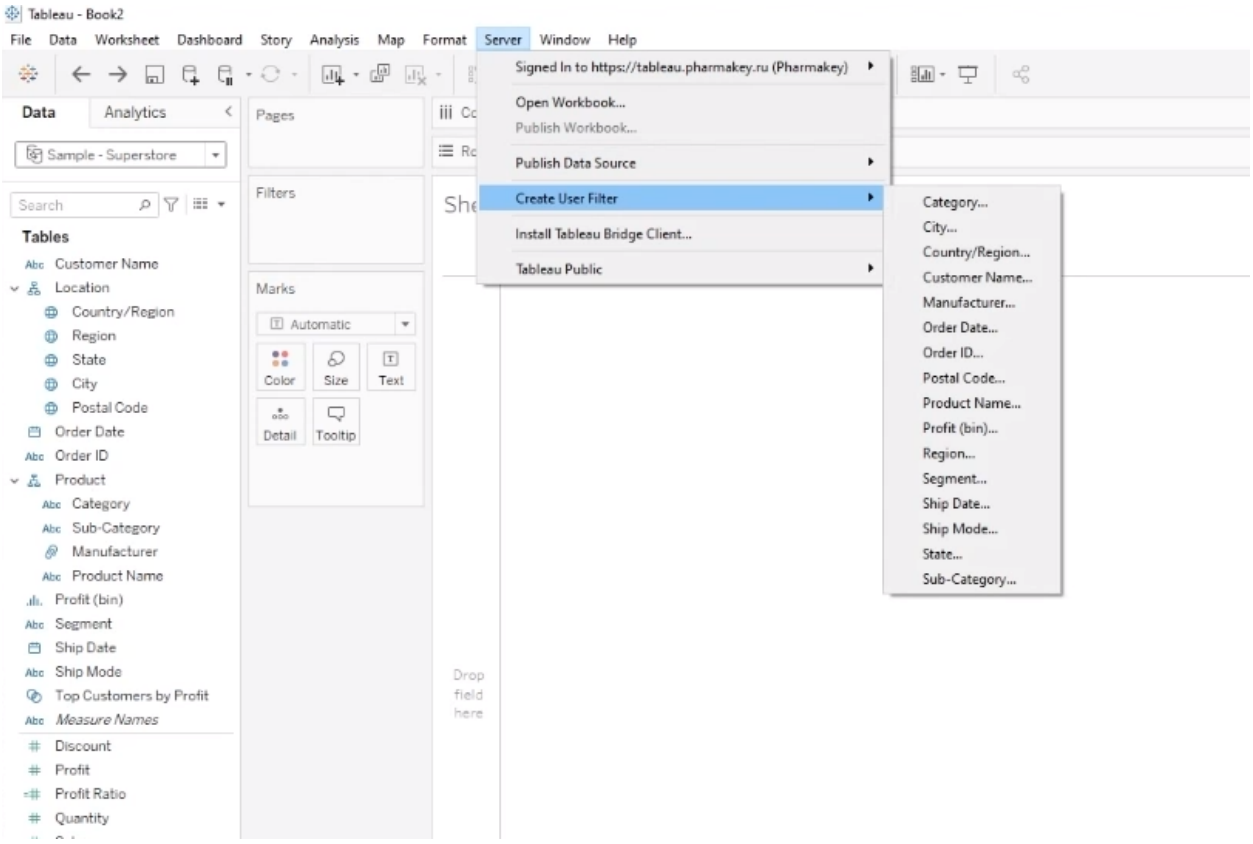
> Row/Column Security

Бывают такие задачи, что часть данных нужно показывать части пользователей. Это может быть региональное или продуктовое деление - это еще называется Row level security.

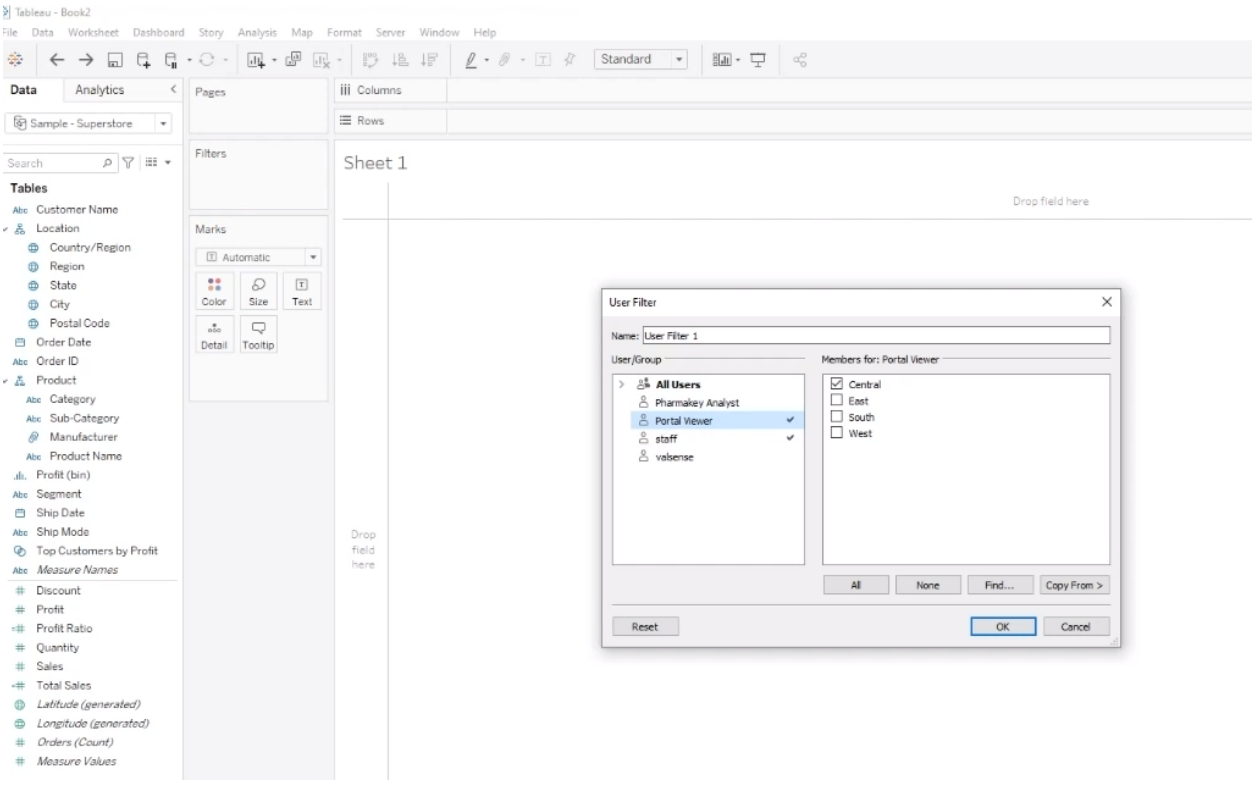
Ситуация, когда части пользователей надо показать один набор измерений, а другому - только часть этих измерений - это Column level security.

Для создания Row level security в Tableau есть как встроенные инструменты, так можно создавать отдельные таблицы управления доступами.

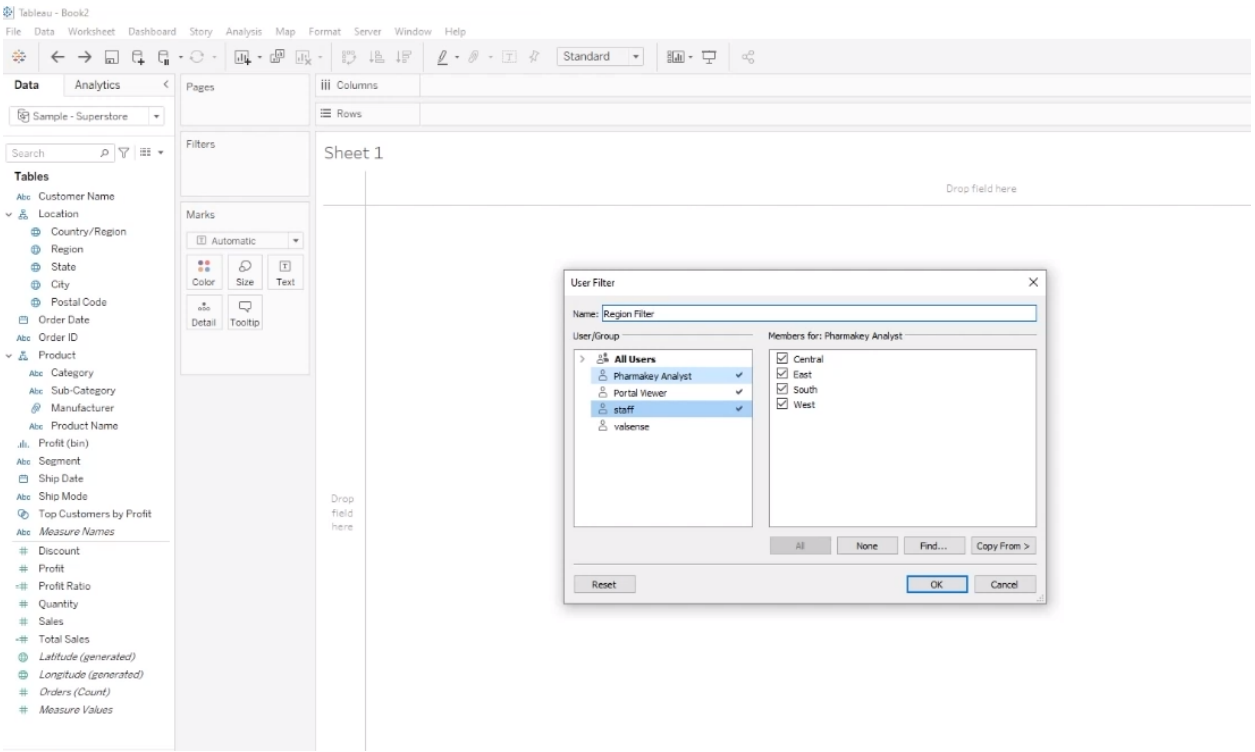
Во вкладке Server вы можете создать фильтры по пользователям, например одной части пользователей доступны определенные регионы.



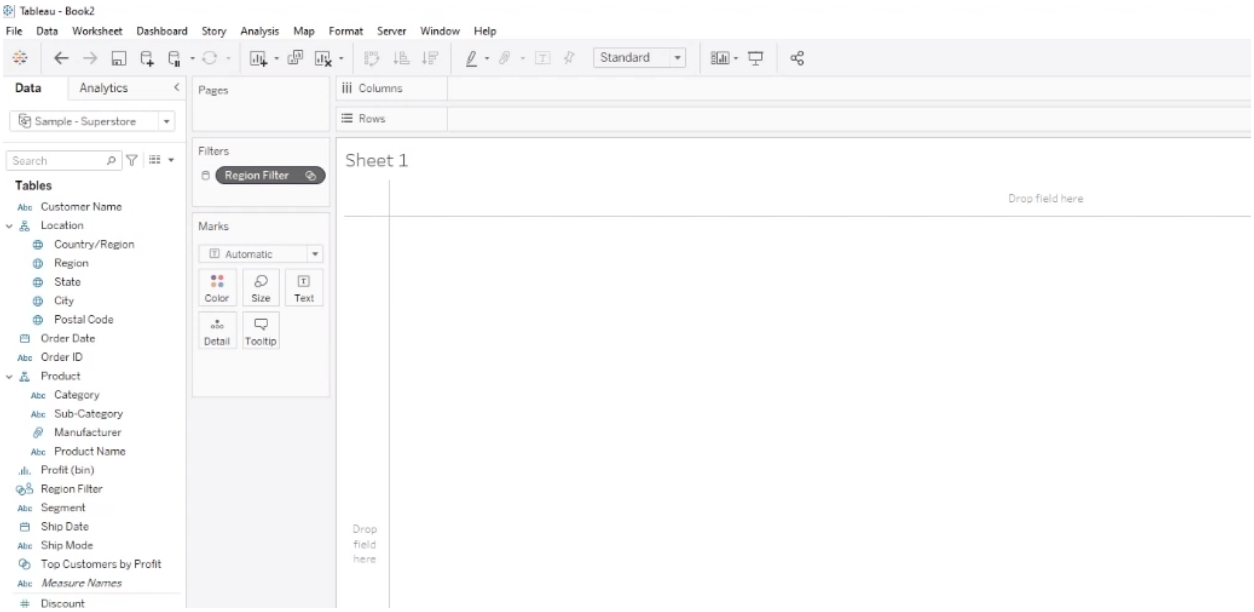
По клику вам доступно окно, со списком пользователей и регионов, в котором вы задаете необходимые параметры доступа.



Созданный пользовательский фильтр можно переименовать.



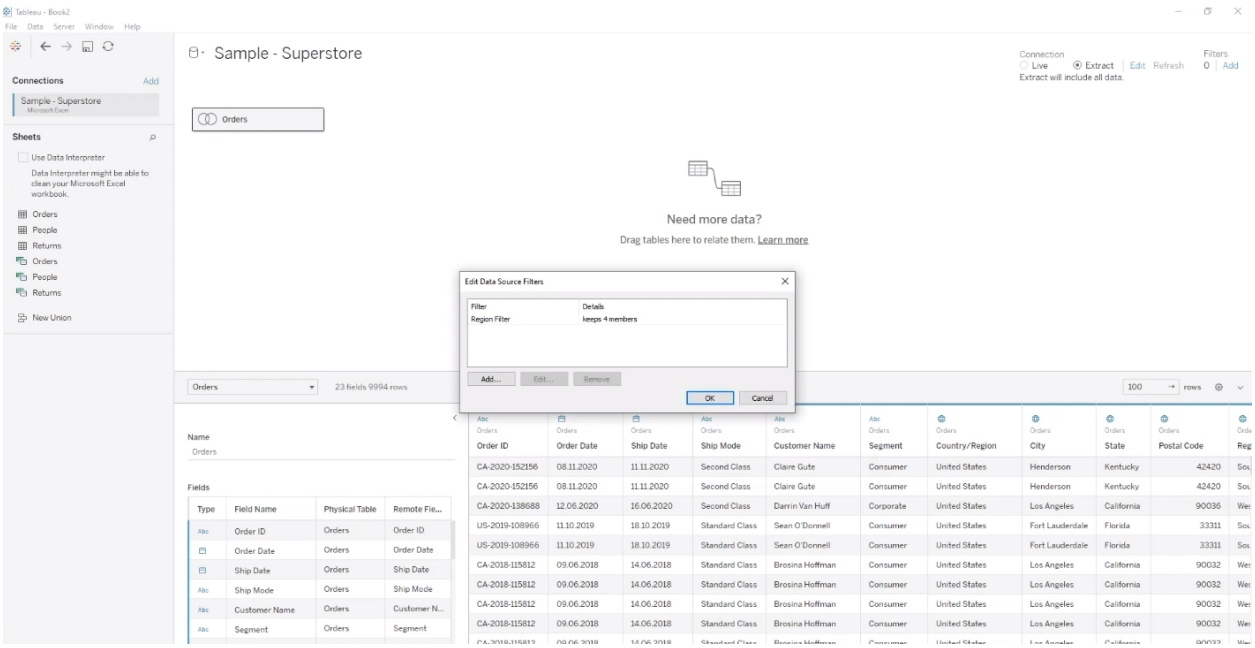
И затем использовать в построении дашборда.



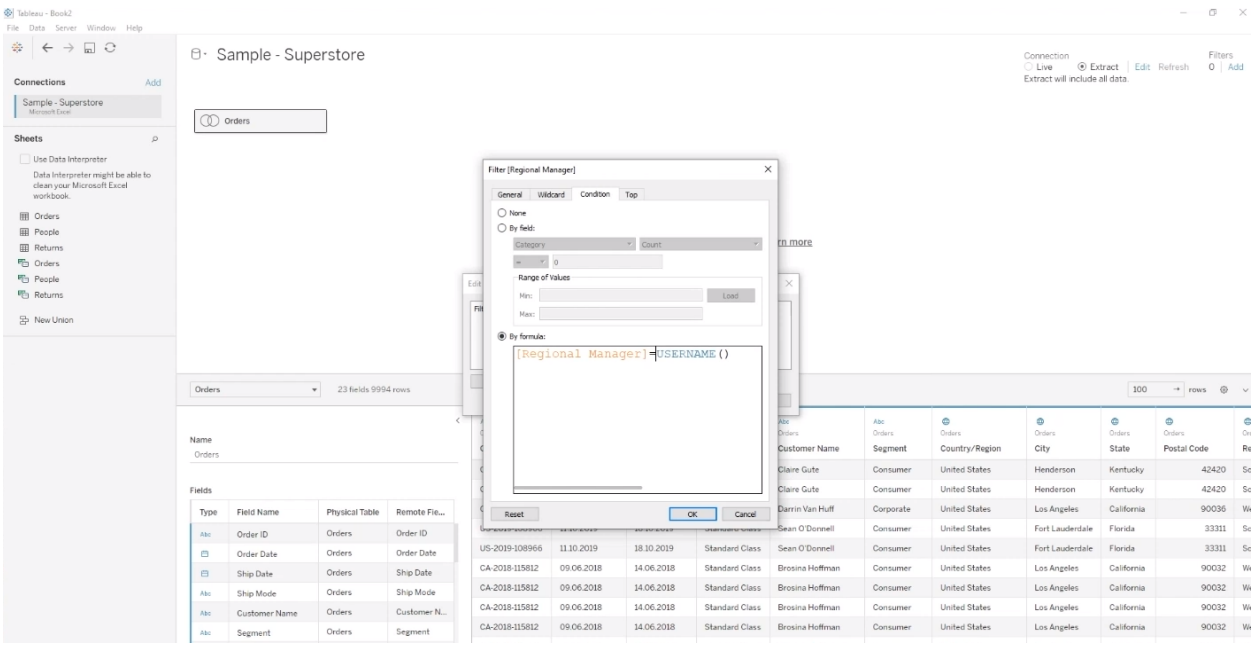
Этот инструмент можно быстро создать и гибко использовать. Созданные правила могут быть сложными, их может быть несколько и их действие может распространятся на отдельный лист или несколько листов в вашем дашборде.

Администрирование, при большом количестве пользователей и используемых фильтров может быть неэффективно. Решением служит создание таблицы фактов и таблицы фактов. В таблице фактов будет храниться пара: пользователь (сохраненный под тем же именем, что и на сервере) и регион доступа.

Делается это следующим образом: вы берете таблицу с заказами и добавляете к ней таблицу фактов.



Чтобы организовать Row level security, мы добавляем фильтры, которые будут действовать на весь дашборд, до его запуска. Мы выставляем эти условия по следующему условию - региональный менеджер должен совпадать с username пользователя.



Для такого решения нам обязательно нужно делать физический Join, потому что логическом Join наша таблица будет "взрываться" на количество менеджеров.

> Подключение к Clickhouse

Подключение к Clickhouse "из коробки" в Tableau пока отсутствует.

Из действующих решений на рынке рекомендую два решения:

- ODBC Driver for ClickHouse;
- ClickHouse JDBC driver

Инструкции по подключению доступны на страницах GitHub'a