

eazyBI - простой инструмент Business Intelligence

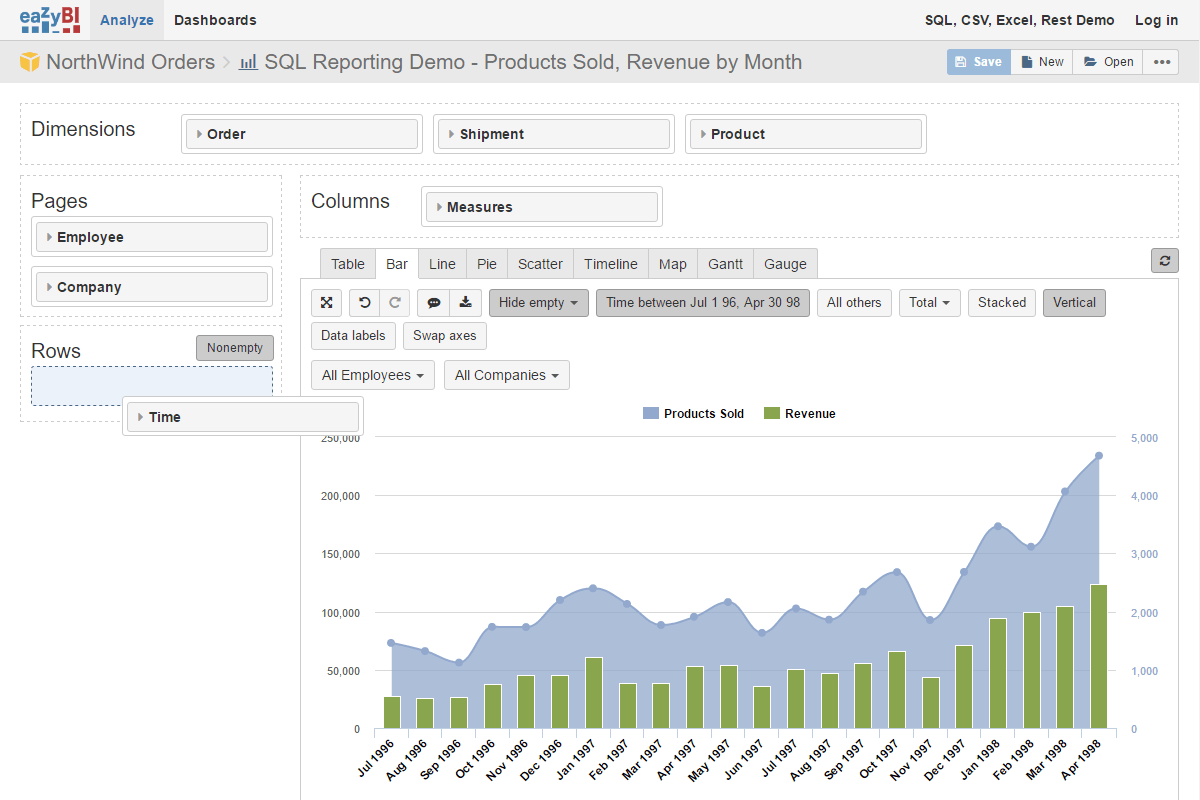
Размещенный в web, простой в использовании набор программного обеспечения для визуализации данных, отчетности и анализа, позволяющий заниматься бизнес-аналитикой.

Перевод с сайта https://eazybi.com/features

Фичи:

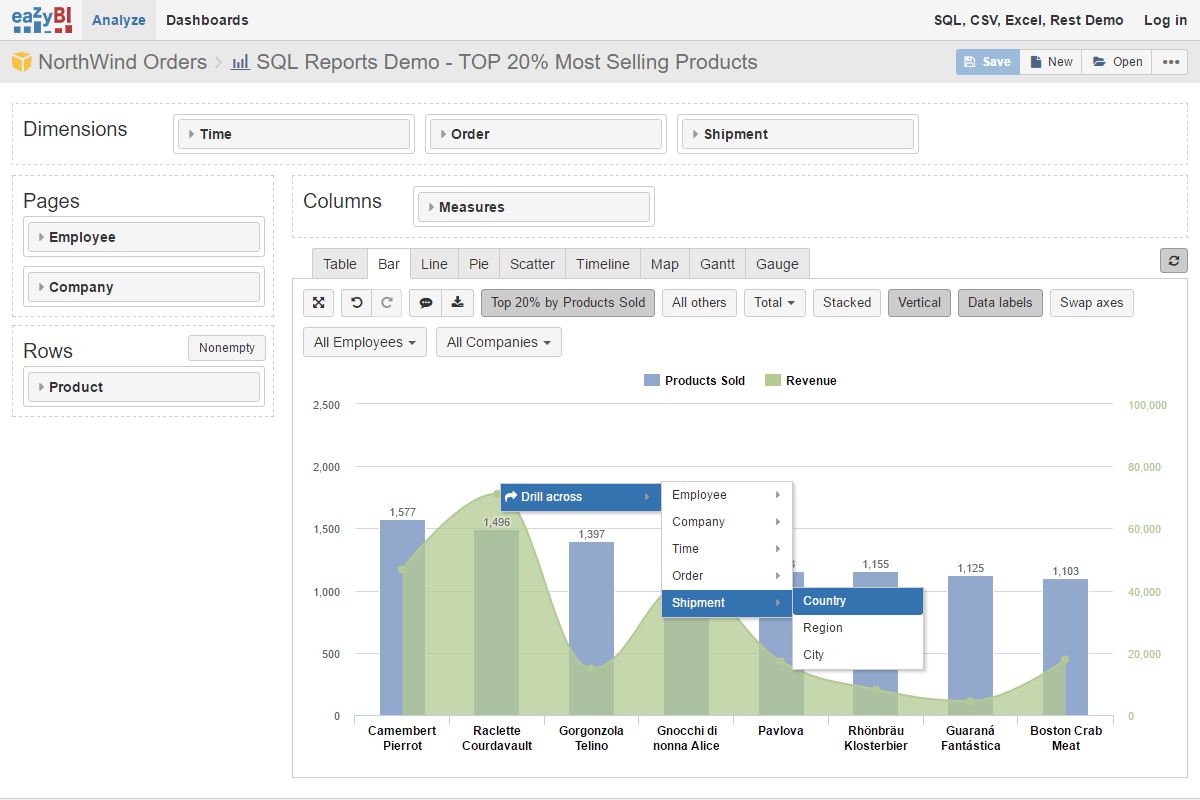
1. Простой Drag-and-Drop интерфейс

Позволяет быстро находить новые идеи для обработки данных. Просто перетащите данные в строки и столбцы и мгновенно появится результат.



1. Быстрое Drill Into Details или Across Dimensions

Быстрый и простой анализ данных с технологией OLAP в реальном времени. Начните с небольшой выгрузки и затем копайте глубже чтобы изучить ваши бизнес-данные с многомерным анализом (dimensional analysis). Погружайтесь в детали (Drill into details) или рассматривайте в нескольких измерениях (multiple dimensions interactively). Сортировка результатов по любым мерам (measure).

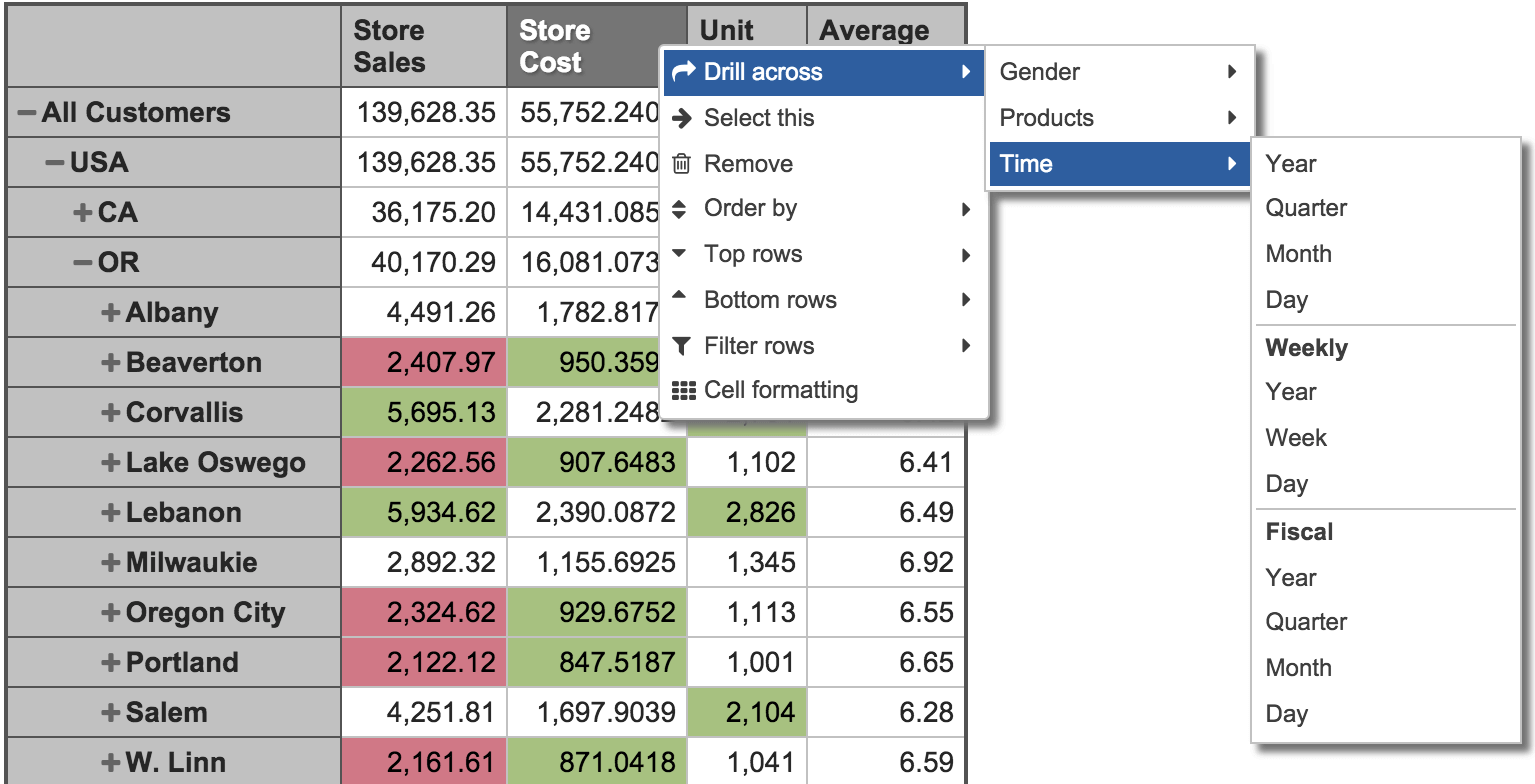


1. Многие интерактивные типы диаграмм

Быстрое переключение между таблицей, колонкой, гистограммой, линией, областью, пирогом, пузырями, графиком, картой, диаграммой Ганта или смешанными диаграммами позволит вам выбрать интерактивный тип диаграммы, который соответствует вашим данным лучше всего.

Таблица (Table Chart)

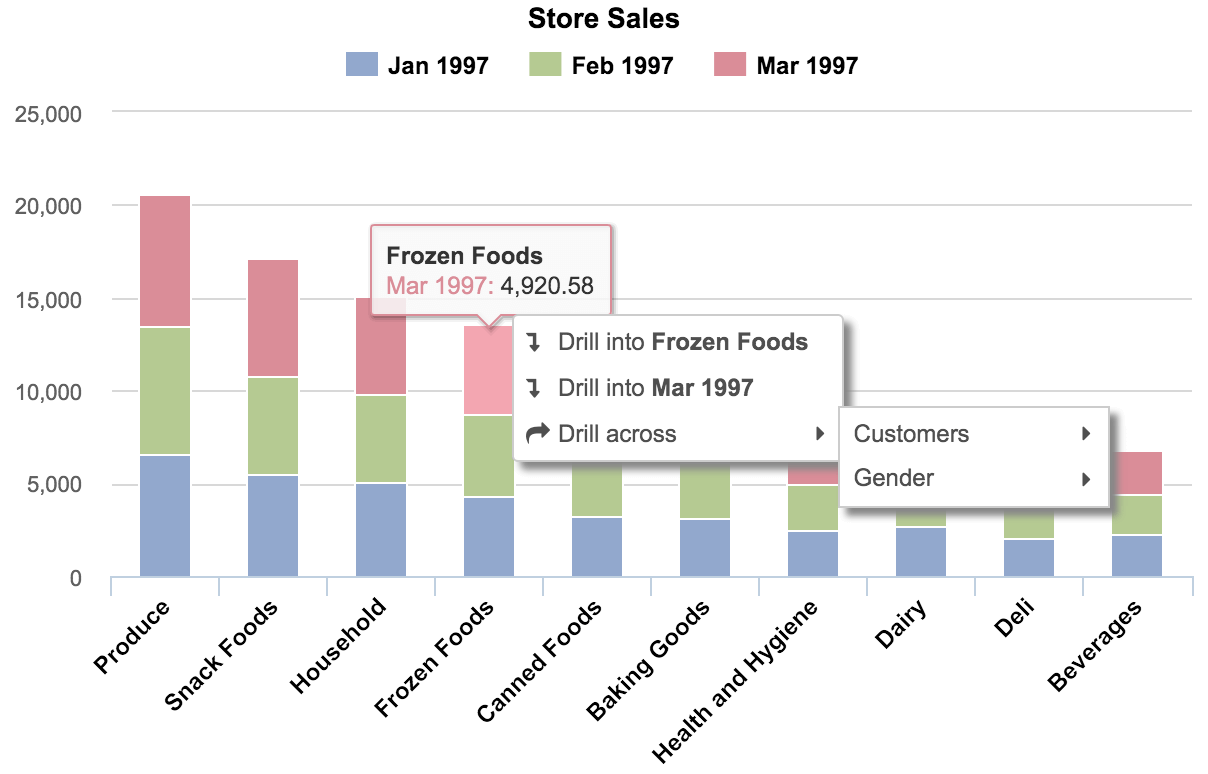
Все ваши цифры бок о бок



Перетаскивайте столбцы, погружайтесь в детали или по измерениям, добавляйте условное форматирование, создавайте фильтры или устанавливайте границы.

Гистограмма (Bar Chart)

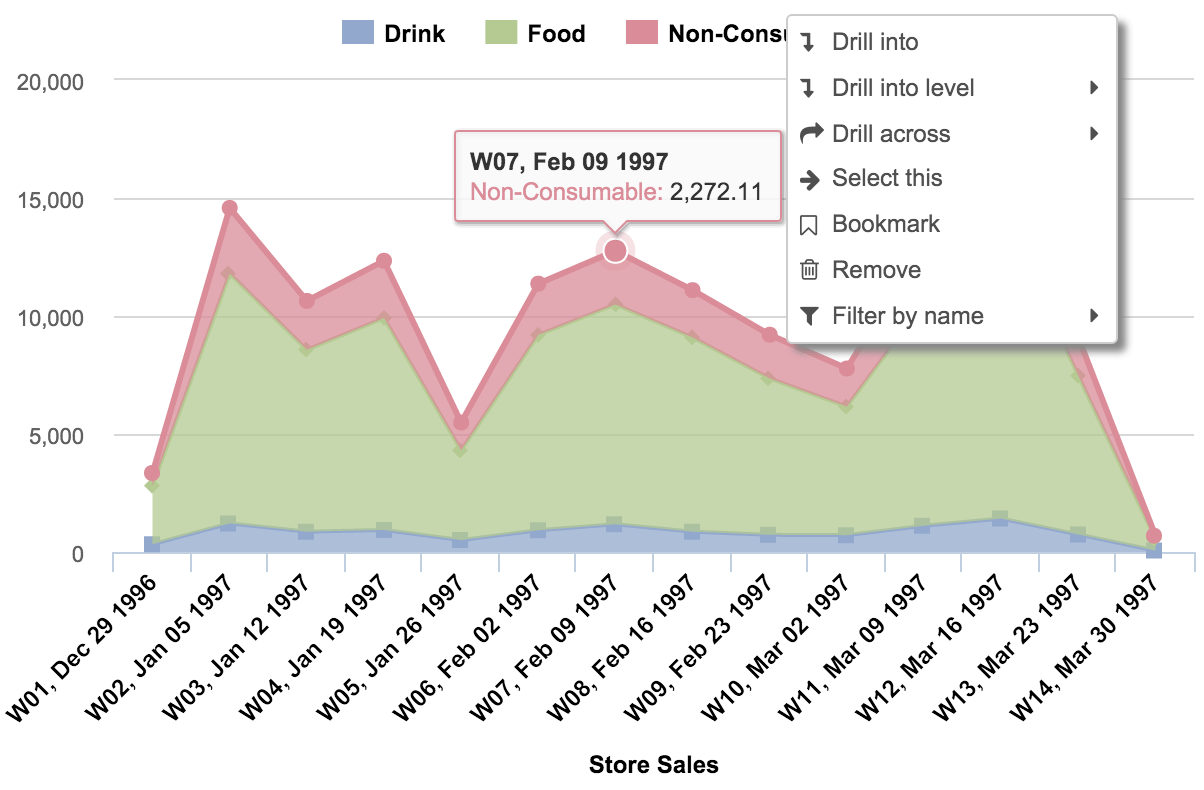
Классический способ представления данных сравнения различных наборов данных



Меняйте цвета, надписи и отдельные оси, чтобы выделить необходимые вам данные. Комбинируйте различные типы диаграмм.

График (Line Chart)

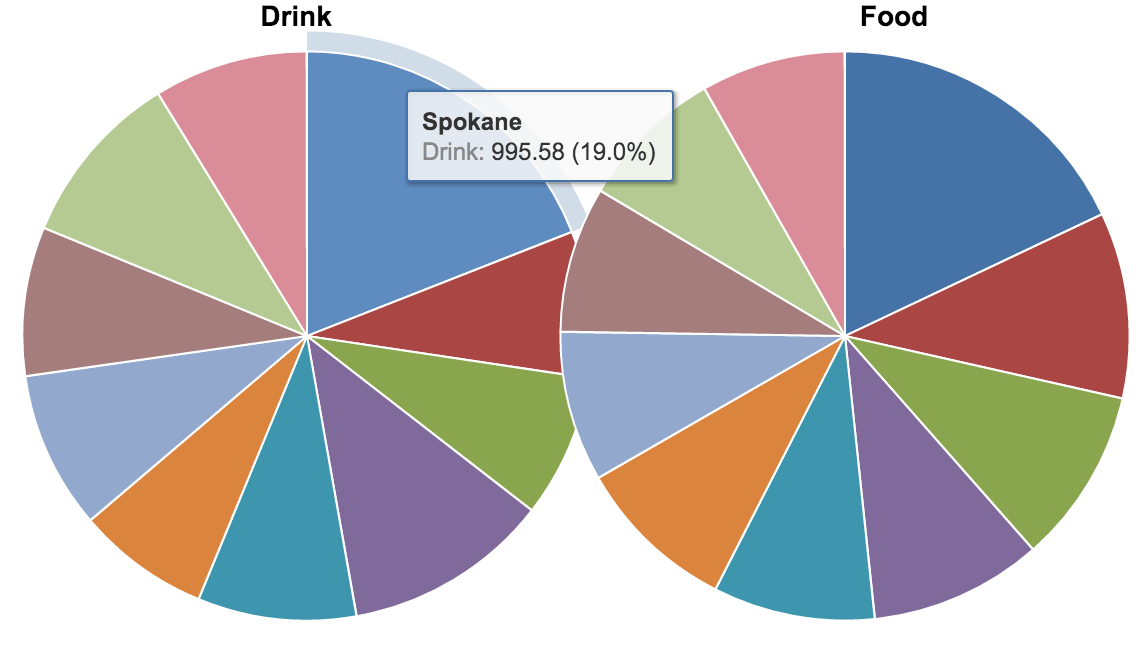
Лучший способ чтобы проиллюстрировать тенденции с течением времени.



Сравнивайте несколько категорий. Выберайте между линией и площадью чтобы выделить данные или создавайте диаграмму области (area chart) для отображения состава и детализации тенденции.

Круговая диаграмма (Pie Chart) // Пирог

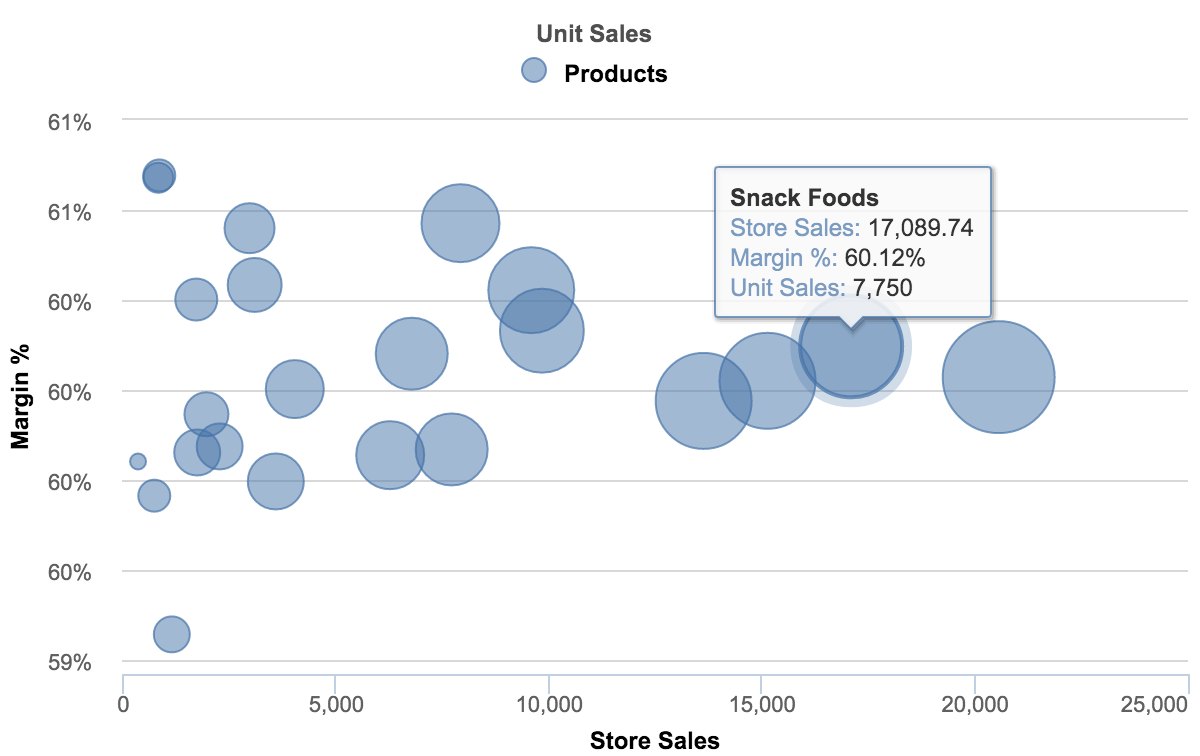
Отображение относительной доли от общего набора данных



Проиллюстрируйте простое погружение вниз в одном измерении, или добавьте больше измерений, создав диаграмму пончика (donut chart).

Диаграмма рассеивания и диаграмма-пузырь (Scatter and Bubble Chart) // точечная и пузырьковая диаграмма

Сравните два или более параметра в одном графике.

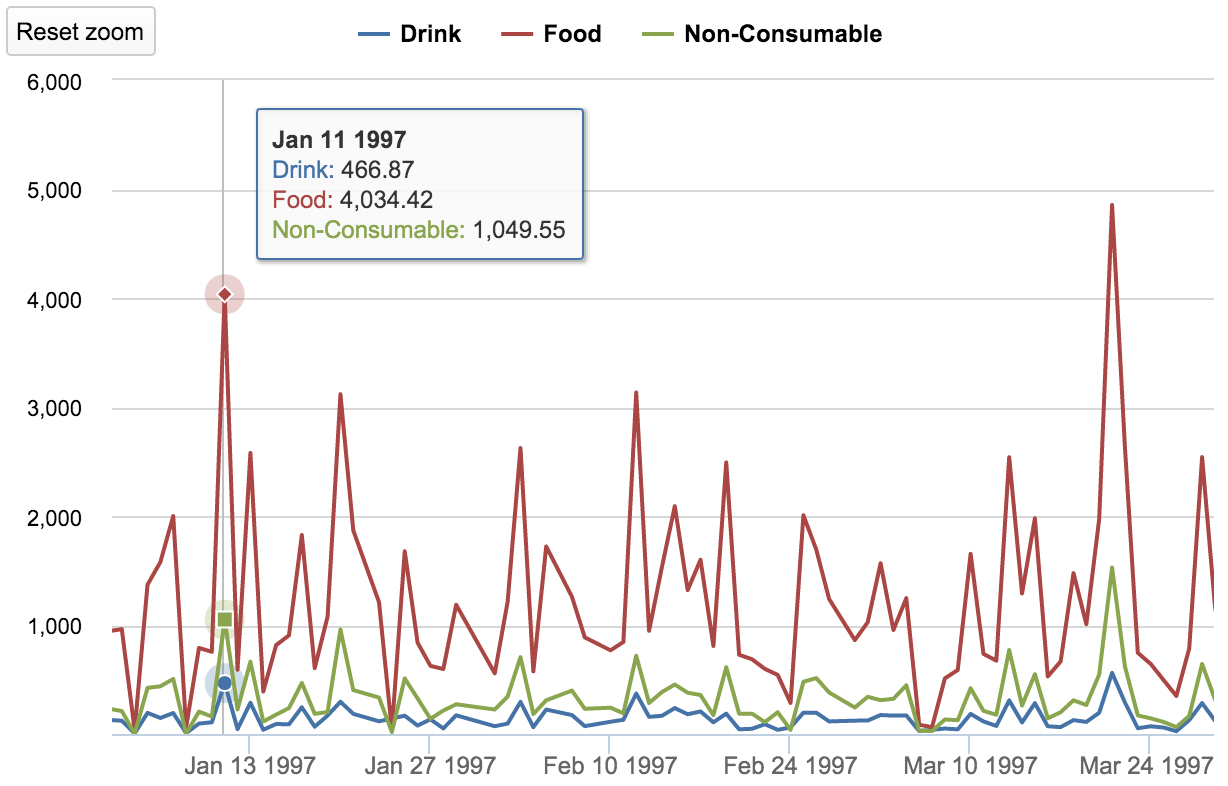


Отображайте отношения между элементами на основе двух переменных параметров, используя диаграмму рассеивания, или добавляйте ещё измерений и создавайте пузырьковую диаграмму.

Быстро и легко погружаться в детали или искать по измерениям, чтобы увидеть распределение данных с разных точек зрения.

Специальная временная ось (Special Timeline Chart)

Позволяет проиллюстрировать тенденции с течением времени.



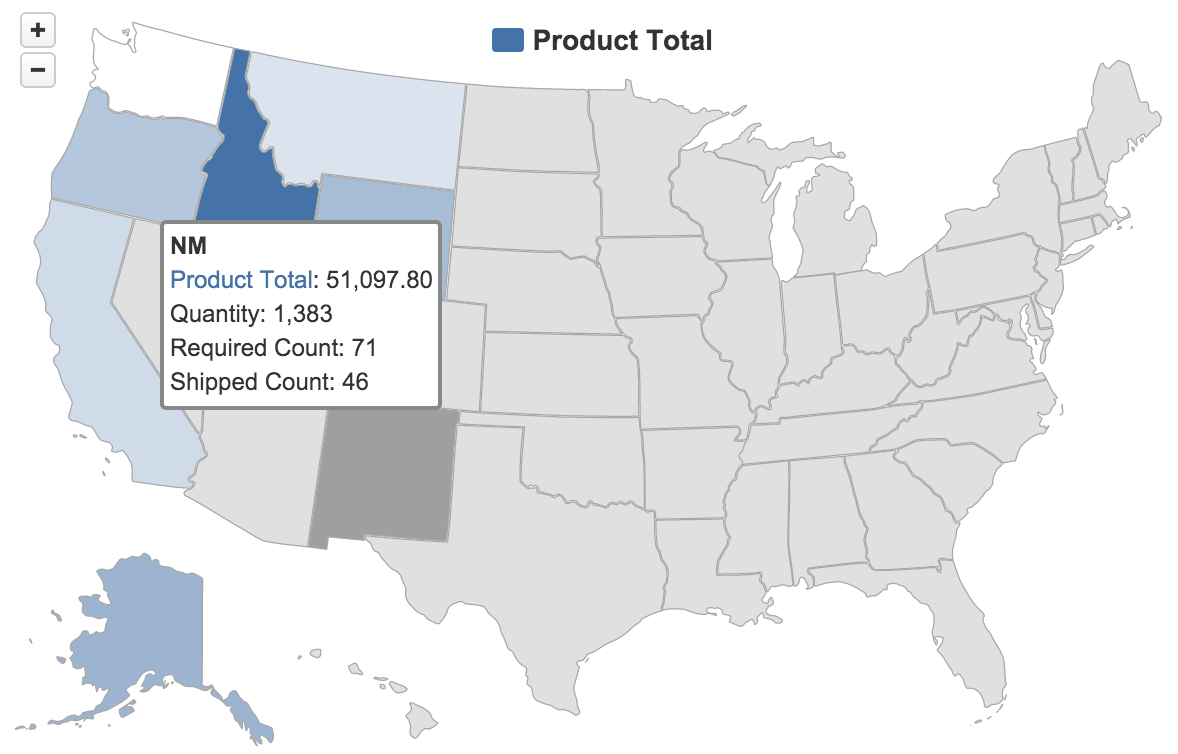
Временная шкала – специальный тип диаграммы, позволяющий увеличить или уменьшить масштаб с помощью простого щелчка и перетаскивания, выбирая точные рамки времени. Устанавливает конкретный временной фильтр, используя общий язык, как “30 days ago”.

Выбирайте любую точку данных и погружайтесь в детали с помощью щелчка.

Выбирайте между линией, сплайном, колонкой или различными типами диаграмм области, а также переключайтесь между столбцами для сравнения и оценки.

Карта (Map Chart)

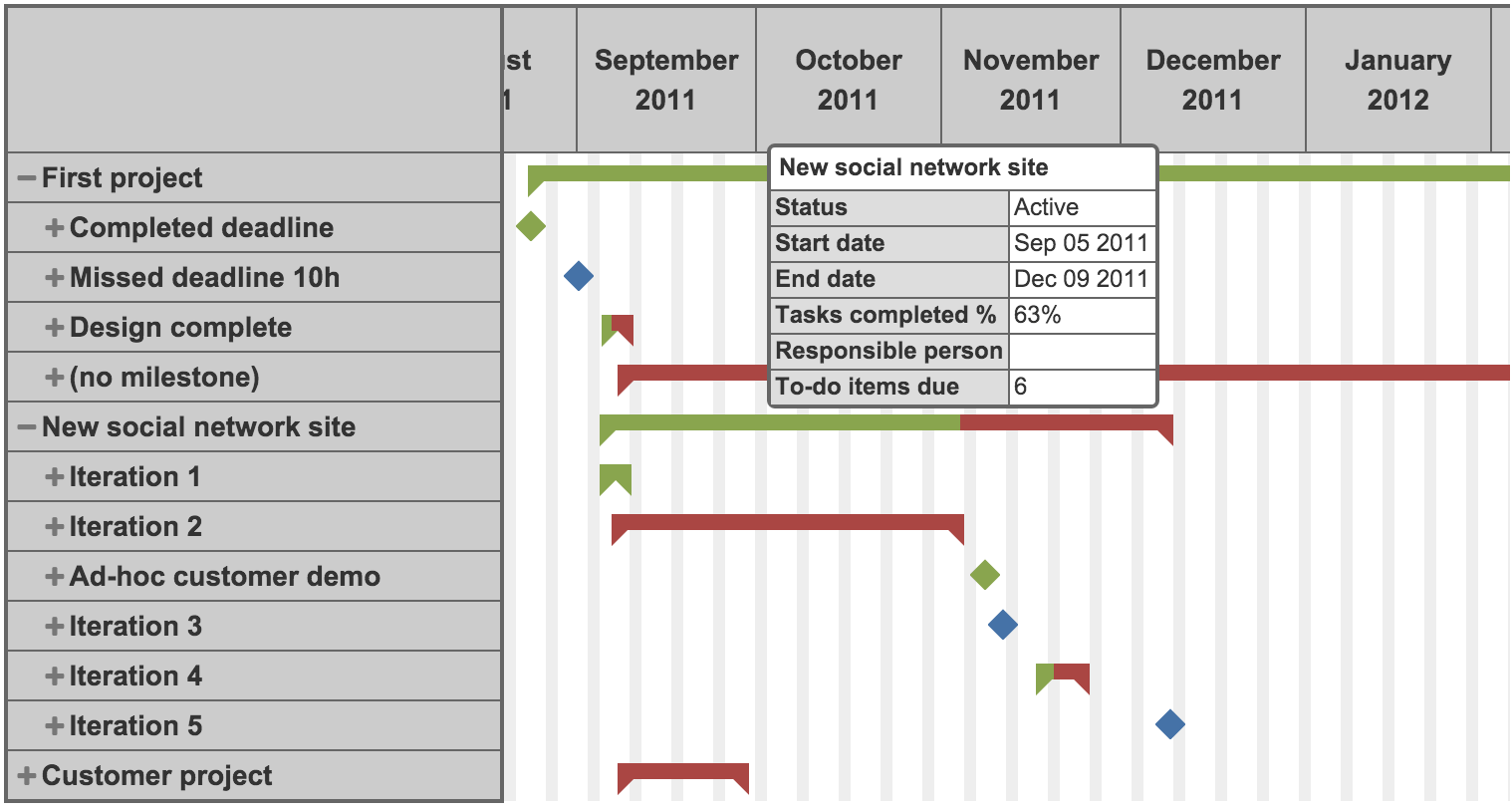
Позволяет мгновенно увидеть данные на основе гео-меток, таких как страны, регионы и города.



Переключайтесь между картой мира и Европы или выберайте конкретный округ, чтобы увидеть региональное распределение. Наведите курсор мыши на нужную область, чтобы увидеть соответствующую информацию. Выбирайте между статическими или другими типами карт, таких как OpenStreetMap или MapQuest, чтобы увидеть более визуальное распределение в диаграмме на основе географического местоположения.

Диаграмма Ганта (Gantt Chart)

Позволяет быстро увидеть прогресс ваших проектов



Переключайтесь между ежемесячным и ежедневным отображением, чтобы увидеть последние детали и наведите мышь на любую задачу, чтобы увидеть всю соответствующую информацию.

Устанавливайте фильтры и вы сможете быстро проверить индивидуальные проекты, задачи и статусы, чтобы идентифицировать всё текущие и просроченные задачи, а также увидеть индивидуальную производительность.

Gauge Chart

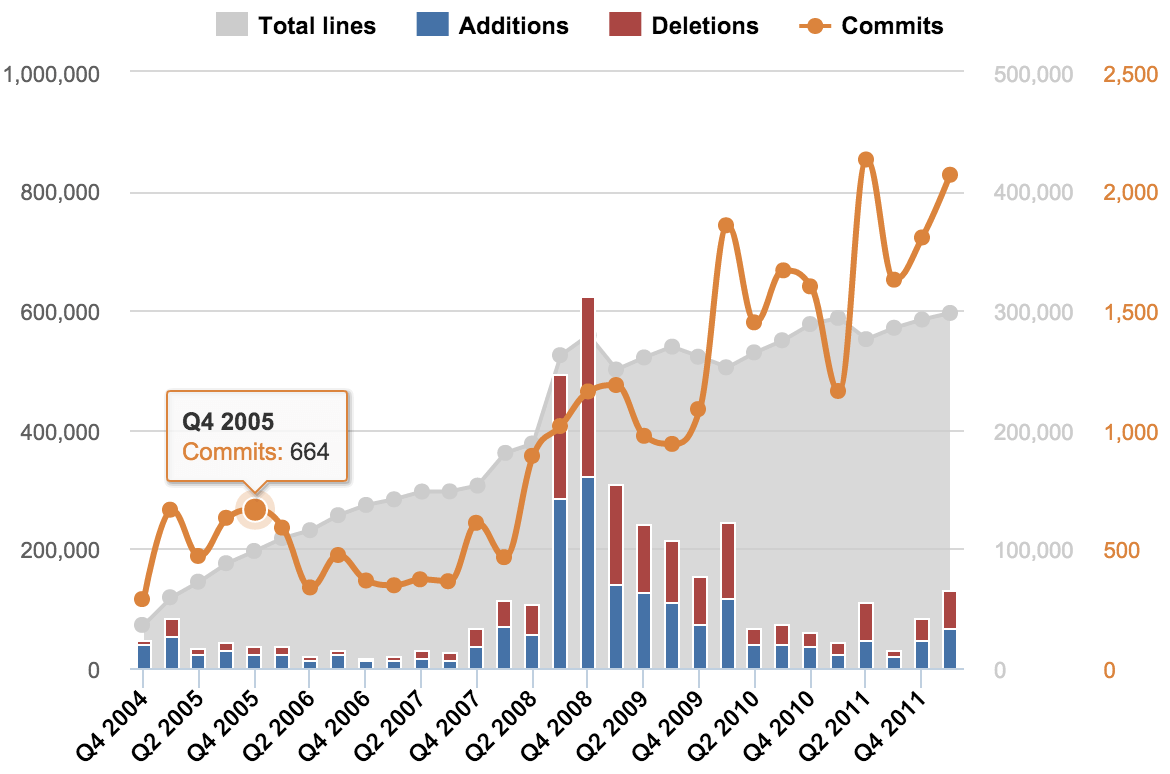
Все ваши KPI в одном месте



Диаграмма Gauge Chart – быстрый способ держать руку на пульсе своего бизнеса. Устанавливайте собственные пределы или добавляйте мощные фильтры. Погружайтесь в детали или в различные категории.

Смешанная диаграмма (Mixed Chart)

Когда простой график просто не может рассказать всю историю.

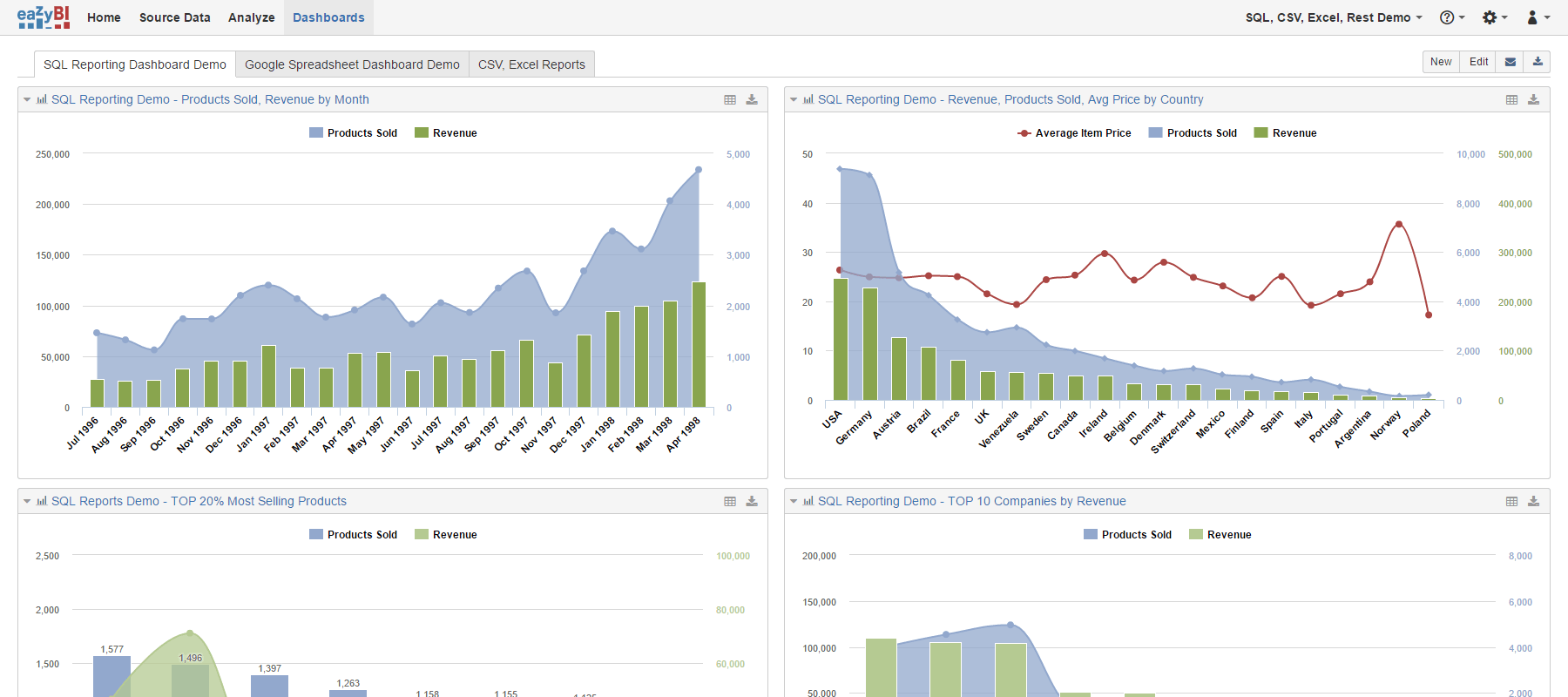


Создавайте визуально привлекательные и понятные диаграммы путём смешивания различных типов диаграмм. Начните с базовой гистограммы и добавьте различные типы диаграмм. Выбирайте между различными цветами или символами, чтобы выделить ключевые моменты.

1. Персонализированные отчеты и доски (Dashboards)

Создавайте мощные, персонализированные отчёты и доски в течение нескольких минут, и наслаждайтесь регулярным автоматическим обновлением данных с простой конфигурацией и установкой.

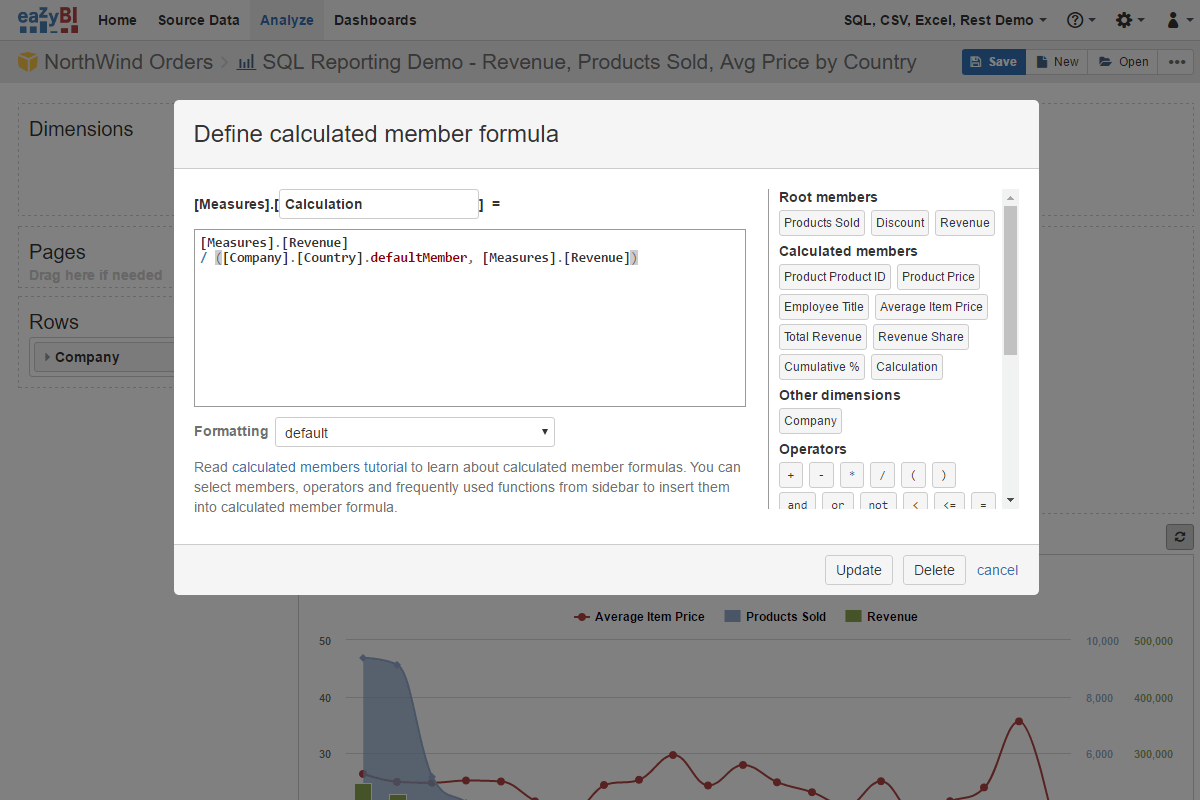
Позволяет экспортировать любые результаты в CSV или Excel, а также делиться ими и публиковать их.



1. Мощные расчёты для опытных пользователей

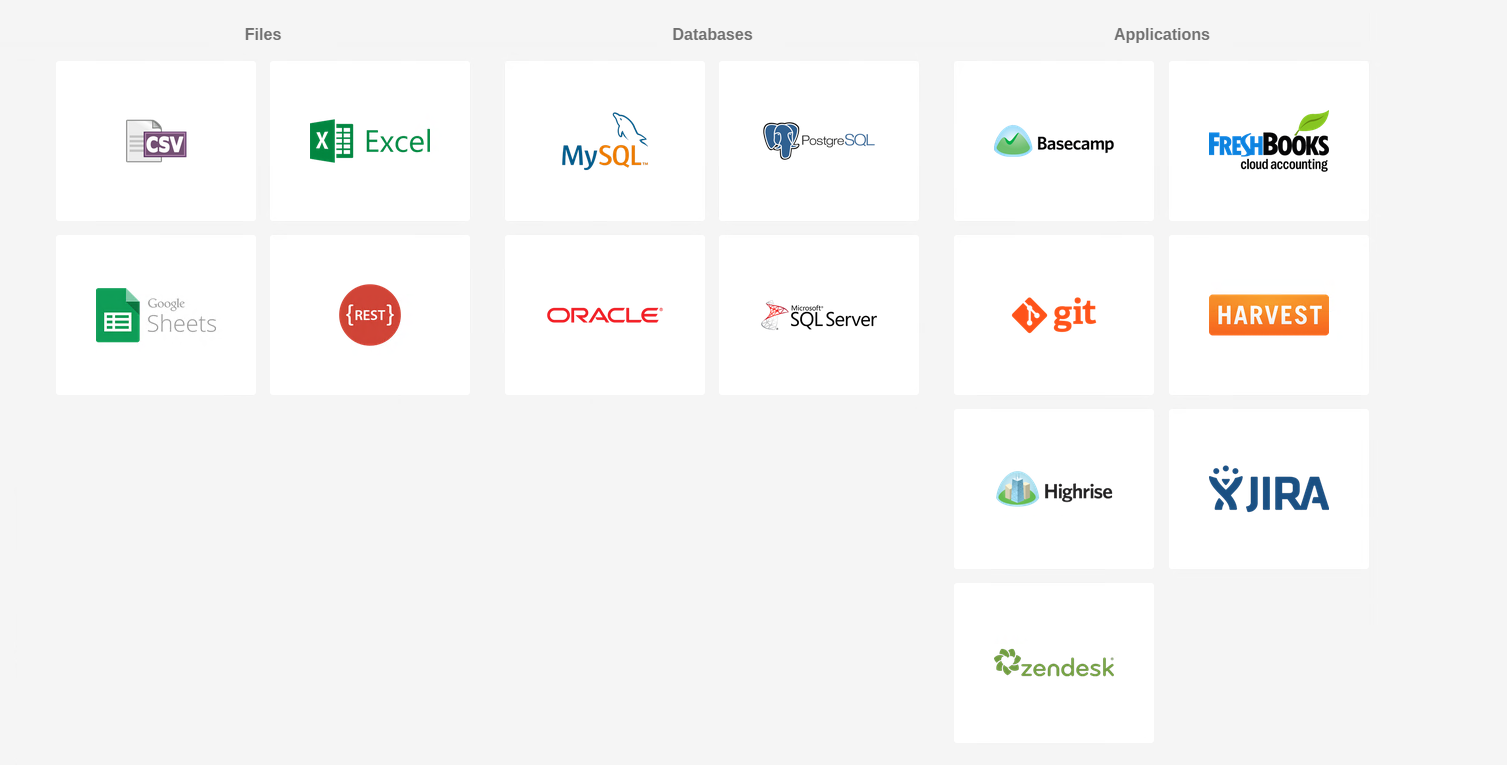
Выполните многомерный анализ и исследуйте бизнес-данные практически любым способом, например, с помощью языка запросов MDX.

Используйте предварительно определенные вычисления или создайте свои собственные. С MDX, вы можете сделать практически любой расчет, который можно себе представить! Воспользуйтесь дополнительными функциями MDX, предоставляемыми eazyBI для выполнения быстрых вычислений с общими типами данных.



1. Импорт любых источников данных

Импортируйте данные из популярных веб-приложений и файлы с неструктурированными данными или подключайтесь к существующему хранилищу данных.



Постоянно добавляются новые, популярные источники данных в библиотеку, так что можно иметь один централизованный источник информации, доступном в одном месте, для всех бизнес-идей, отчётов и анализа.



Наша архитектура

Краткая схема выполнения запросов отчётов, размещённых в Confluence

Том Кукуруз – обычный пользователь JIRA и Confluence

Пустые прямоугольнички в прямоугольнике Confluence – странички-отчёты.

Верхний прямоугольничек – новые отчёты в eazyBI. Один раз в заранее выбранный промежуток времени из базы JIRA запрос будет доставать данные для отчёта и сохранять в куб. Один отчёт – один куб.

Время старта автоматического обновления содержимого куба (от ‘раз в 10 минут’ до ‘раз в сутки’). Это помогает разгрузить базу, в разы ускорить время выполнения запросов, а также улучшить отображение результатов (см. скриншоты выше).

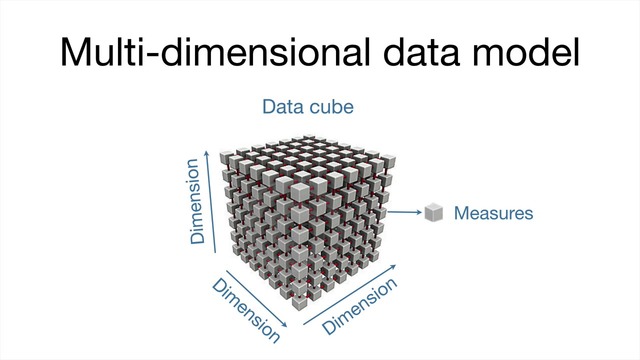
Нижний работает как раньше – с помощью макроса SQL-Query и общается напрямую с базой JIRA. Такая схема кажется более простой, чем с eazyBI, но ограничена по функционалу и грузит базу.

Кубик из кубиков – многомерный куб. Состоит из измерений и как минимум одной меры. Подробнее о кубах ниже.



Кубы (Cubes)

Импортированные данные хранятся в многомерных кубах данных. Кубы содержат слепки данных (fact data), которые разделены по размерам и каждый подобный факт “ячейка” содержит значения меры об этом факте.



Легко показать куб с тремя измерениями, но можно добавлять столько измерений и мер, сколько необходимо в одном кубе данных.

Все меры в одном кубе имеют одни и те же измерения. Если в одном аккаунте вы хотите хранить различные типы мер, которые не разделяют те же измерения, то можно создать несколько различных кубов.

Измерения (Dimensions)

Типичные измерения куба: время, клиенты, продукты, сотрудники, проекты. Каждое измерение может иметь либо один уровень, либо иерархию с несколькими уровнями.

Меры, как правило, являются целыми числами или десятичными значениями. Типичные примеры: количество, сумма затрат.

Вычисляемые элементы (Calculated members)

Иногда необходимо вычислить новые меры от других мер существующего куба. Можно также определить вычисляемые элементы в других измерениях. Расчётные формулы определяются с помощью языка MDX. Основные арифметические формулы могут быть созданы очень легко, но, изучая другие функции MDX вы сможете создавать любые вычисляемые элементы. (подробнее см. ниже)

Анализ данных и создание отчётов

После импорта данных из исходных файлов или приложений вы можете перейти на вкладку Анализ и начать исследовать данные в созданном кубе. Вы можете начать создавать отчёты-таблицы путём перетаскивания необходимых измерений (dimensions) для столбцов, строк и страниц и просматривать данные на различных уровнях измерения.

После выбора необходимых данных в отчёте-таблице вы можете переключиться на различные графические отчёты, чтобы изучить ваши данные в более наглядном виде. После того как вы создали макет отчёта, который вы хотите часто использовать, вы можете сохранить этот отчёт с данным именем. Когда позже вы будете открывать сохранённый отчёт, то будете получать последние результаты из куба данных с использованием сохранённого макета отчёта.

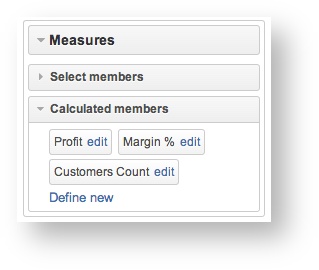
Вычисляемые элементы ([Calculated members](https://docs.eazybi.com/display/EAZYBI/Calculated+members))

**На этой странице:**

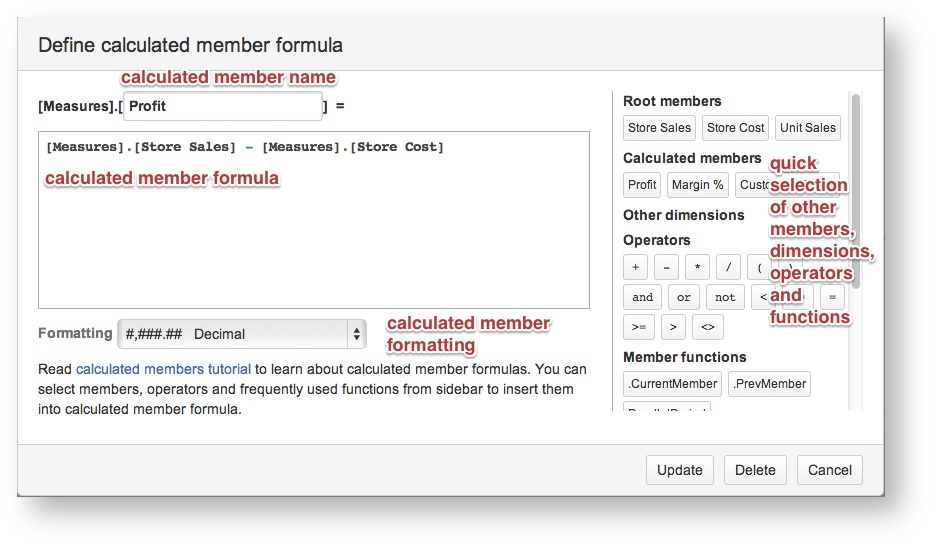
* [Определение](https://docs.eazybi.com/display/EAZYBI/Calculated+members#Calculatedmembers-Definenewcalculatedmember) нового вычисляемого элемента
* [Измерения, иерархия, уровни и имена](https://docs.eazybi.com/display/EAZYBI/Calculated+members#Calculatedmembers-Dimension,hierarchy,levelandmembernames) элементов
* [Простые](https://docs.eazybi.com/display/EAZYBI/Calculated+members#Calculatedmembers-Simplearithmeticcalculations) арифметические вычисления
* Кортежи
* [Перемещение](https://docs.eazybi.com/display/EAZYBI/Calculated+members#Calculatedmembers-Movingintime) во времени
* Множества
* Множества периодов времени
* Операции над множествами
* Объединение
* Объединение элементов в разных измерениях
* Условия
* Свойства элементов
* Преобразование типа Дата
* Расчёты разницы во времени
* [Значения](https://docs.eazybi.com/display/EAZYBI/Calculated+members#Calculatedmembers-Defaultvalues) по умолчанию
* [Комментарии](https://docs.eazybi.com/display/EAZYBI/Calculated+members#Calculatedmembers-Comments)
* Специальные комментарии с аннотациями
* [Список всех функций MDX](https://docs.eazybi.com/display/EAZYBI/Calculated+members#Calculatedmembers-ListofallMDXfunctions)

Определение нового вычисляемого элемента

Если вы хотите определить новый вычисляемый элемент – разверните соответствующее измерение (например, меры) и расширьте список вычисляемых элементов:



Если вы хотите определить новый вычисляемый элемент – нажмите **Define new**. Если вы хотите изменить (или удалить) существующий вычисляемый элемент – нажмите на редактирование. Если у вас нет прав для редактирования вычисляемых элементов в текущем аккаунте, то вы увидите только ссылку **отображения**, которая покажет определение существующего вычисляемого элемента. При определении нового вычисляемого элемента или изменении существующего – вы увидите следующую форму определения вычисляемого элемента:



Каждый вычисляемый элемент должен иметь имя (уникальное в пределах измерения) и расчётную формулу (далее туториал по написанию формул). Кроме того, вы можете указать, как расчётное значение элемента должно быть отформатировано (например, как целое, десятичное, дата или с помощью форматирования по умолчанию). С правой боковой панели вы можете быстро выбирать другие элементы, измерения, операторы или функции для использования в расчётной формуле.

После определения формулы вычисляемого элемента нажмите **Обновить**, чтобы сохранить его или **Удалить** для удаления существующего вычисляемого элемента или **Отменить**, чтобы отменить изменения. Если формула вычисляемого элемента будет недействительной, то будет выведено соответствующее сообщение об ошибке.

**Не изменяйте и не удаляйте заранее определённые вычисляемые элементы, которые были созданы за счёт импорта приложения источника** – их первоначальное определение будет воссоздано в следующий раз, когда вы будете выполнять импорт приложения источника. Если вам необходимо создать модифицированную версию существующего предварительно определенного вычисляемого элемента -- скопируйте его формулу расчета и создайте новый вычисляемый элемент с другим именем и модифицируйте скопированную формулу вычисления для нового вычисляемого элемента.

Следующие примеры будут использовать [Sales demo cube](https://eazybi.com/accounts/1/cubes/Sales) измерения и меры, чтобы проиллюстрировать создание расчётных формул.

Изменения, иерархия, уровни и имена элементов

При написании формул расчета вам нужно будет ссылаться на иерархии измерений, уровни иерархии и другие существующие элементы измерения. В MDX языке запросов все имена заключены в квадратные скобки [].

Для ссылки на **измерение** нужно просто заключить его имя в квадратные скобки, например, [Customers], [Time] или [Measures].

Если измерение имеет только одну **иерархию**, то можно ссылаться на основную иерархию таким же образом, как и на измерение, например [Customers] или [Measures]. Когда вы импортируете измерение **Time**, то он автоматически создаёт основную иерархию (с Year, Quarter, Month и Day уровнями) а также **Weekly** иерархию (с Year, Week и Day уровнями). [Time] будет ссылаться на основную иерархию измерения **Time**, но [Time.Weekly] будет ссылаться на измерение **Time** иерархии **Weekly**.

Когда вы расширяете конкретное измерение и **All hierarchy level members (все элементы иерархии уровней)**, то вы видите имена всех основных **уровней иерархии**. Вы можете ссылаться на определённый уровень иерархии с [dimension or hierarchy name].[level name]. Например, [Customers].[City] ссылается на уровне **City**основной иерархии измерения **Customers**.

В случае измерения Time (времени) используйте [Time].[Year], [Time].[Quarter], [Time].[Month], [Time].[Day], чтобы ссылаться на основные уровни иерархии [Time.Weekly].[Year], [Time.Weekly].[Week], [Time.Weekly].[Day], чтобы ссылаться на Weekly уровни иерархии.

Каждое измерение, как правило, будет иметь значение по умолчанию **All member**, которые могут быть использованы для получения итоговых мер по этому измерению. Если **Customers**измерение имеет элемент **All** member, названный **All Customers**, то вы можете сослаться на него с помощью [Customers].[All Customers]. По умолчанию **All**member могут быть переименованы, безопасно использовать [Customers].DefaultMember для получения того же результата.

В случае измерения **Time** существует два дефолтных **All** members для каждой иерархии – [Time].DefaultMember и [Time.Weekly].DefaultMember. В расчётных формулах можно использовать выражение [Time].CurrentHierarchy.DefaultMember,чтобы получить [Time].DefaultMember или [Time.Weekly].DefaultMember в зависимости от типа **Time** иерархии, который использовался в отчёте.

Если вы хотите ссылаться на **элементы верхних уровней измерений**, то используйте [dimension or hierarchy name].[member name]. Например, [Customers].[USA] будет ссылкой на **USA** элемент верхнего уровня **Country**. Для ссылки на **подробный уровень измерения элемента** необходимо указать полный путь “иерархии” этого элемента, например,[Customers].[USA].[CA]для ссылки на**CA** элемент в **State Province** уровне под родительским элементом **USA**. Или используйте элементы [Customers].[USA].[CA].[San Francisco], чтобы ссылаться на город и так далее.

Все **меры** находятся на верхнем уровне измерения **Measures** и вы можете ссылаться на них с помощью [Measures].[Store Sales], [Measures].[Store Cost] и т.д.

Простые арифметические вычисления

Теперь, когда вы знаете, как ссылаться на другие существующие меры и другие элементы измерения, мы можем создать простые арифметические расчёты.

Если у нас есть меры [Measures].[Store Sales] и [Measures].[Store Cost], то мы можем определить новую вычисляемую меру Profit (которая будет иметь полное название [Measures].[Profit]) с формулой:

|  |
| --- |
| [Measures].[Store Sales] - [Measures].[Store Cost] |

Когда вы будете использовать новую меру Profit в отчётах eazyBI, то для любой комбинации со значениями других измерений эта формула будет вычислять разность между значением меры **Store Sales** и **Store Cost**.

Вы также можете использовать уже определённые вычисляемые меры в других вычисляемых мерах, которые вы определили позже. Например, вы можете определить [Measures].[Margin %] формулой:

|  |
| --- |
| [Measures].[Profit] / [Measures].[Store Sales] |

и он будет рассчитывать прибыль как число от 0 до 1. Если вы хотите отображать число в виде процентов, то измените **Formatting** этой вычисляемой меры на **целое** или **десятичное процентное** форматирование.

Вы можете выполнять арифметические расчёты и для вычисляемых элементов в другом измерении. Например, вы можете определить [Customers].[West coast] вычисляемые элементы в измерении **Customers** с формулой:

|  |
| --- |
| [Customers].[USA].[CA] + [Customers].[USA].[OR] + [Customers].[USA].[WA] |

и теперь, когда вы будете сочетать в отчётах [Customers].[West coast] с [Measures].[Store Sales] вы получите общий объём продаж для всех этих трёх штатов вместе. Если вы будете сочетать его с [Measures].[Profit] вы получите общую прибыль для всех трёх штатов.

Существует по умолчанию скрытая [Measures].[Fact Count] мера, которая возвращает фактическое число строк в таблице базы данных куба. Вы можете использовать её чтобы вычислить среднее арифметическое значение, например, определить [Measures].[Average Store Sales] как

|  |
| --- |
| [Measures].[Store Sales] / [Measures].[Fact Count] |

Кортежи

Если вы используете [Measures].[Store Sales] в формуле, то она будет вычислять значение меры **Store Sales** для **текущего контекста** каждой ячейки отчёта. Текущий отчёт включает в себя соответствующую строку и столбец элементов измерения ячейки отчёта, а также отдельные элементы измерения страницы.

Если вы хотите переопределить при расчёте значение меры некоторого текущего элемента контекста измерения другим элементом, то вам нужно использовать **кортеж** этой меры и другого элемента измерения (или нескольких элементов измерения). MDX синтакс для кортежей (member\_1, member\_2, ..., member\_n). Например, если вы хотели бы получить значение меры **Store Sales** для всех клиентов, то в формуле следует использовать кортеж:

|  |
| --- |
| ( [Measures].[Store Sales], [Customers].DefaultMember ) |

Эта формула означает, что **Store Sales** должны рассчитываться с использованием всех элементов измерений из текущего контекста за исключением измерения **Customers,** для которого следует использовать элемент по умолчанию (**All Customers**).

Кортежи часто используются для расчёта процента значения измерения от некоторой общей стоимости. Давайте определим расчётную меру [Measures].[Sales / customers total %] с формулой (и процентом форматирования)

|  |
| --- |
| [Measures].[Store Sales] / ([Measures].[Store Sales], [Customers].DefaultMember) |

Когда в отчёте вы будете сочетать [Measures].[Sales / customers total %] и, например, [Customers].[USA].[CA], то вы увидите процент от продаж Калифорнии от общего объёма продаж клиента.

Возможно, вы не хотите видеть процент от общего объёма продаж, но хотите видеть процент от иерархии Customers прямого родителя Sales (например, для уровня City показать процент от соответствующих State sales). В этом случае вы можете использовать CurrentMember измерение или иерархию свойства для доступа к текущему элементу контекста измерения. Например, [Customers].CurrentMember возвращает текущий элемент в измерении **Customers**, для которых вычисляется формула. Или, другими словами, если измерение **Customers** размещается в строках отчёта, то при использовании [Customers].CurrentMember мы знаем, для какой строки формула была вычислена.

Если у нас есть текущий элемент, то мы можем перейти к другим членам измерения относительно этого элемента. [Customers].CurrentMember.Parent вернёт родительский элемент в иерархии **Customers** для текущего элемента. Ancestor([Customers].CurrentMember, [Customers].[Country]) будет переходить от текущего элемента до “предка” в уровне **Country**.

Таким образом, мы можем определить расчётную меру [Measures].[Sales / parent customer %] с формулой:

|  |
| --- |
| [Measures].[Store Sales] / ([Measures].[Store Sales], [Customers].CurrentMember.Parent) |

Перемещение во времени

Также как вы можете перейти на другой уровень иерархии измерения вы можете перейти к другому элементу измерения в том же измерении. Обычно используется в измерении **Time**, когда вы хотите сравнить меры между различными периодами времени или агрегировать (объединить) диапазон периода времени.

Например, расчётная мера [Measures].[Sales monthly growth] с формулой:

|  |
| --- |
| [Measures].[Store Sales] - ([Measures].[Store Sales], [Time].CurrentMember.PrevMember) |

рассчитает рост [Measures].[Store Sales] по сравнению с предыдущим периодом времени элемента измерения **Time**. Если эта расчётная мера будет объединена в отчёте с месяцем в измерении **Time**, то она покажет рост по сравнению с предыдущим годом. Если она будет объединена с годом, то покажет рост по сравнению с предыдущим годом.

Есть несколько функций, которые помогут вам “перемещаться во времени”:

* [Time].CurrentMember.PrevMember возвращает предыдущий элемент, который содержит данные, в том же самом уровне иерархии (возвращает пустой элемент для первого элемента)
* [Time].CurrentMember.NextMember возвращает следующий элемент
* [Time].CurrentMember.Lag(2) возвращает предыдущий элемент с шагом 2 (использует любое число для аргумента, Lag(1) такой же как PrevMember)
* [Time].CurrentMember.Lead(2) возвращает следующий элемент с шагом 2 (Lead(1) такой же как NextMember)
* ParallelPeriod([Time].[Year], 1, [Time].CurrentMember) вернёт “параллельное” Time элемента один год назад (например, для элемента уровня Day элемент  Jan 01 2012 это будет день Jan 01 2011, но для уровня Month элемент Jan 2012 станет Jan 2011)
* OpeningPeriod([Time].[Day], [Time].CurrentMember) вернёт первого потомка текущего элемента **Time** на уровне **Day** (первый день года, четверть или месяц, который определён в измерении **Time**)
* ClosingPeriod([Time].[Day], [Time].CurrentMember) вернёт последнего потомка текущего элемента **Time** на уровне **Day** (в последний день года, четверть или месяц, который определён в измерении **Time**)

Если куб имеет только одно измерение Time, то [Time].CurrentMember аргумент может быть опущен, так как он будет использоваться по умолчанию. Таким образом, вы можете использовать более короткое выражение ParallelPeriod([Time].[Year], 1)**,**OpeningPeriod([Time].[Day]), ClosingPeriod([Time].[Day])**.**

Иногда вы хотите проверить, работает ли ваше выражение навигации по времени как вы ожидаете (прежде чем использовать его в дальнейших расчётах). Вы можете использовать свойство элемента.Name, чтобы получить имя элемента как результат вычисления формулы. Например, определить расчётную меру [Measures].[test opening day] с формулой:

|  |
| --- |
| OpeningPeriod([Time].[Day], [Time].CurrentMember).Name |

и использовать его в отчёте вместе с элементами измерения **Time** и посмотреть, будет ли получен ожидаемый результат для элементов измерения **Time** на разных уровнях.

Множества (Sets)

Теперь вы знаете как перейти к отдельным элементам измерения. Но довольно часто необходимо выполнять операции на **множестве элементов измерения.** Например, вы хотели бы получить значение некоторой меры над выбранным множеством элементов измерения.

Самым простым способом использования множеств в расчетных формулах является использование списка элементов, заключенных в фигурные скобки, например:

|  |
| --- |
| {[Customers].[USA].[CA], [Customers].[USA].[OR], [Customers].[USA].[WA]} {[Time].[2011], [Time].[2012]} |

Если вы хотите выбрать множество в качестве диапазона последовательных уровней элементов измерения, то вы можете указать первый и последний элемент и использовать : между ними. Например, так выглядит множество дат с Jan 01 2012 по Jan 15 2012:

|  |
| --- |
| [Time].[2012].[Q1 2012].[Jan 2012].[Jan 01 2012]:[Time].[2012].[Q1 2012].[Jan 2012].[Jan 15 2012] |

Довольно часто бывает, что необходимо указать точный диапазон элементов, но хотелось бы получить all dimension hierarchy level members (все элементы уровня иерархии измерения?). Вы можете сделать это с помощью метода Members. Это позволит получить все месяцы в измерении **Time**:

|  |
| --- |
| [Time].[Month].Members |

Если вы хотите проверить, какие элементы возвращаются некоторым набором выражений, то вы можете использовать функцию SetToStr для создания строки конкатенированных имён всех элементов. Например, определить расчётную меру [Measures].[test all months] с формулой:

|  |
| --- |
| SetToStr([Time].[Month].Members) |

Есть несколько других полезных функций для работы с множествами. Несколько примеров их использования:

* [Customers].[USA].[CA].Children возвращает множество детей с использованием иерархии измерения (в данном примере все города в California)
* Descendants([Customers].[USA], [Customers].[City]) возвращает множество потомков на заданном уровне иерархии (в данном примере все города в USA)
* [Customers].[USA].[CA].[San Francisco].Siblings возвращает все элементы, имеющие одного и того же родителя как элемент (в этом примере все города в California), тоже самое если использовать выражение [Customers].[USA].[CA].[San Francisco].Parent.Children

Существуют дополнительные методы, которые можно использовать, чтобы получить только первый и последний элемент этих множеств – FirstChild, LastChild, FirstSibling, LastSibling.

Множества периодов времени

Есть дополнительные полезные методы выбора множества для периодов **Time**

* PeriodsToDate([Time].[Year], [Time].CurrentMember) возвращает все периоды с начала текущего элемента в заданном уровне (в данном примере с начала текущего года) до указанного элемента.
* YTD([Time].CurrentMember) сокращённый вариант той же функции (аббревиатура от начала года до даты).   
  Обратите внимание, что даже несмотря на иерархию Weekly также имеет уровень Year, по умолчанию эта функция будет работать только в иерархии Time. Для еженедельной иерархии используйте PeriodsToDate([Time.Weekly].[Year]).
* QTD([Time].CurrentMember) возвращает множество элементов от начала квартала до даты
* MTD([Time].CurrentMember) возвращает множество элементов от начала месяца до даты
* WTD([Time.Weekly].CurrentMember) возвращает множество элементов от начала недели до даты

Как уже говорилось ранее, если у вас есть только одно измерение **Time** в кубе, то вы можете использовать более короткие выражения YTD(), QTD(),MTD() и WTD() так как по умолчанию они будут получать в качестве аргумента текущий элемент измерения времени – [Time].CurrentMember.Но иногда вам будет нужно передавать другой элемент для этих функций. К примеру, если вы хотите сравнить данные с начала года до даты и с начала года до даты предыдущего год, то вы можете использовать следующее выражение, чтобы получить множества с начала года до даты элемента измерения Time год назад:

|  |
| --- |
| YTD(ParallelPeriod([Time].[Year], 1, [Time].CurrentMember)) |

Если вы хотите использовать более короткое выражение, то можете опустить текущий аргумент элемента – YTD(ParallelPeriod([Time].[Year], 1)). Большинство связанных со временем функций будут использовать [Time].CurrentMember как аргумент по умолчанию.

Если вы хотите получить несколько последних периодов от измерения **Time**, то вы можете использовать функцию LastPeriods. Например:

|  |
| --- |
| LastPeriods(3, [Time].CurrentMember) |

Будет возвращать множество с текущим элементом **Time** и 2 предыдущих элемента. Если же вы хотите получить текущий и два последующих периода **Time**, то используйте отрицательное значение -3 вместо 3. Вы также можете опустить последний аргумент этой функции, а также использовать LastPeriods(3).

Если вы хотите получить текущий элемент измерения **Time** в иерархии **Weekly**, то вам необходимо использовать выражение [Time.Weekly].CurrentMember. Иногда вам нужно будет написать формулу, которая должна использовать либо [Time].CurrentMember (например, при создании ежемесячного отчёта), либо [Time.Weekly].CurrentMember (при создании еженедельного отчёта). В этом случае вы можете использовать

|  |
| --- |
| [Time].CurrentHierarchyMember |

Которая будет либо возвращать основную иерархию **Time** или иерархию **Weekly** текущего элемента, зависящую от типа иерархии **Time**, который вы используете в отчёте.

Операции над множествами (Set operations)

Существуют несколько операций, которые можно выполнять над множествами:

* {set1, set2, ..., setn} возвращает объединение двух и более наборов
* Except(set1, set2) возвращает элементы *set1*, но удаляет любые элементы, которые находятся в *set2*
* Head(set, number) возвращает множество с первым *числом* элементов из исходного множества (если *номер* не указан, то возвращается множество из первого множества элементов)
* Tail(set, number) возвращает последнее *число* элементов из множества
* set.Item(position) возвращает один элемент из набора с указанной *позиции* (начиная с нуля). Т.е. если вы хотели бы получить первый элемент множества, вы можете использовать выражение Head(set).Item(0)

Довольно часто необходимо фильтровать множество элементов. Вы можете сделать это с помощью Filter(set, condition). Например, таким образом, вы можете фильтровать все города с продажами больше, чем 1000:

|  |
| --- |
| Filter([Customers].[City].Members, [Measures].[Store Sales] > 1000) |

В выражении условия [Customers].CurrentMember ссылается на текущий элемент множества, для которого условие вычисляется.. Например, это будет возвращать все города, которые имя начинается *San* (с помощью оператора *MATCHES* с [регулярным](http://en.wikipedia.org/wiki/Regular_expression) выражением)

|  |
| --- |
| Filter([Customers].[City].Members, [Customers].CurrentMember.Name MATCHES 'San .\*') |

Другая типичная функция, которая используется в условиях -- IsEmpty. Это выражение вернёт все города, которые имеют непустой объём продаж:

|  |
| --- |
| Filter(    [Customers].[City].Members,    NOT IsEmpty([Measures].[Store Sales])  ) |

Ранее уже было отмечено, что простое множество в строковой функции SetToStr полезно для тестирования выражения. Но если вы хотите отформатировать множество результатов, то вы можете использовать Generate(set, string\_expression, separator\_string). Например, следующее выражение будет возвращать названия городов, конкатенированных через запятую, где нет записей об объёме продаж:

|  |
| --- |
| Generate(    Filter(      [Customers].[City].Members,      IsEmpty([Measures].[Store Sales])    ),    [Customers].CurrentMember.Name,    ', '  ) |

Объединение (Aggregates)

Теперь, когда вы знаете, как выбрать различные множества элементов измерения, можно рассчитать различные значения из этих множеств:

* Sum(set, numeric\_expression) вычисляет *numeric\_expression* для каждого члена множества и выдаёт сумму всех результатов. Например, Sum(LastPeriods(3), [Measures].[Store Sales])  рассчитает общий объём продаж в течение последних трёх периодов **Time**, начиная с текущего элемента измерения **Time**.
* Count(set) возвращает число элементов множества. Count(set, ExcludeEmpty) будет возвращать количество элементов множества, для которых соответствующие значения меры не пусты.
* Avg(set, numeric\_expression) вычисляет среднее значение *numeric\_expression* во множестве
* Max(set, numeric\_expression) возвращает максимальное значение *numeric\_expression* во множестве
* Min(set, numeric\_expression) rвозвращает минимальное число *numeric\_expression* во множестве
* Median(set, numeric\_expression) возвращает среднее число *numeric\_expression* во множестве

Объединение элементов из разных измерений

В начале этого туториала мы определили [Customers].[West coast] вычисляемый элемент в измерении **Customers** с формулой:

|  |
| --- |
| [Customers].[USA].[CA] + [Customers].[USA].[OR] + [Customers].[USA].[WA] |

Но было бы лучше, если бы мы могли бы расширить или провалиться в West coast при использовании его в отчетах. Для этого нам необходимо определить вычисляемый элемент с помощью функции Aggregate(set). Когда вы будете сочетать такой вычисляемый элемент в отчёте вместе с другими мерами, то вы получите агрегированный результат (по умолчанию сумма) этой меры в указанном множестве элементов. Так что если вы определите [Customers].[West coast] вычисляемый элемент с формулой

|  |
| --- |
| Aggregate({    [Customers].[USA].[CA],    [Customers].[USA].[OR],    [Customers].[USA].[WA]  }) |

то вы получите тот же результат, как и раньше, но в дополнение вы будете иметь возможность расширить или погружаться в элементы West coast и подробные результаты для CA, OR and WA.

Аналогичным образом можно объединить более сложное множество выражений. Например, вы могли бы определить [Customers].[US without west coast]вычисляемый элемент с формулой:

|  |
| --- |
| Aggregate(    Except(      [Customers].[USA].Members,      { [Customers].[USA].[CA],        [Customers].[USA].[OR],        [Customers].[USA].[WA] }    )  ) |

которая будет вычислять совокупность всех штатов US без штатов западного побережья, и вы также сможете погрузиться в них.

Условия (Conditions)

Вы можете написать формулы с использованием условных функций IIF и CASE. Например, IF / THEN / ELSE тип условий реализуется как IIF(condition, if\_true\_expression, if\_false\_expression) (обратите внимание, что есть две буквы I в IIF), например,

|  |
| --- |
| IIF([Measures].[Profit] > 0, 'Profit', 'Loss') |

Если есть много условий, то проще использовать функцию CASE, например

|  |
| --- |
| CASE  WHEN [Measures].[Profit] > 1000  THEN 'Big profit'  WHEN [Measures].[Profit] > 0  THEN 'Small profit'  ELSE 'Loss'  END |

Если все условия сравнения одного и того же выражения могут быть в разных значениях, то можно использовать другую форму оператора CASE, например

|  |
| --- |
| CASE [Time].CurrentMember.Level.Name  WHEN 'Month' THEN    Sum(LastPeriods(3), [Measures].[Store Sales])  WHEN 'Day' THEN    Sum(LastPeriods(90), [Measures].[Store Sales])  END |

В IIF и CASE можно использовать стандартные операторы сравнения (=, <, <=, <>, >, >=), а также операторы AND, OR и NOT и несколько специальных операторов:

* IS возвращает, одинаковы ли объекты, например, [Customers].CurrentMember IS [Customers].DefaultMember(что будет true, если **Customers** текущего элемента по умолчанию элемент **All Customers**)
* IN и NOT IN возвращает, присутствует ли элемент в множестве, например [Customers].CurrentMember IN [Customers].[USA].[CA].Children

Свойства элементов (Member properties)

Элементы измерений обладают некоторыми свойствами по умолчанию (например .Name и .Key) а также они могут иметь дополнительные настраиваемые свойства. eazyBI импортирует исходное приложение (например, из Basecamp, Highrise или JIRA), а также импортирует дополнительные поля измерений из исходных систем. MDX имеет стандартные Properties функции для доступа к свойствам элементов. eazyBI определяет дополнительный get (а также с более длинным именем getProperty) функцию, которая будет возвращать пустой результат вместо исключения, если ни одно из свойств не определено для текущего уровня измерения.

Например, JIRA добавляет **Created at** свойство для всех импортированных issues. Следующее выражение возвращает значение свойства **Created at** для текущего элемента измерения **Issue**:

|  |
| --- |
| [Issue].CurrentMember.get('Created at') |

Преобразование типа Дата (Date type conversion)

Иногда вам может понадобиться преобразование строкового числа в целое число, десятичное или дату (например, для преобразования значения строки пользовательского поля для соответствующего типа, чтобы иметь возможность использовать его в дальнейших расчётах). Есть несколько доступных функций для преобразования типов данных:

* CInt(value) возвращает значение, преобразованное в целое число
* CDbl(value) возвращает значение, преобразованное в *double floating number* тип (следует использовать, когда результаты должны быть в десятичной форме)
* eazyBI определяет дополнительную функцию DateParse(value),которая будет пытаться преобразовать значение в значение даты, используя различные форматы дат (например, как 2012-01-31 и Jan 31 2012 будут преобразованы в правильную дату). Существует стандартная MDX функция CDate, но она поддерживает меньше форматов дат.
* Функция DateParse  также поддерживает динамические выражения дат, как смещения с сегодняшнего дня. Например, вы можете использовать DateParse('today') или DateParse('30 days ago') или DateParse('1 week from now').

Вычисления разницы во времени

eazyBI определяет дополнительную функцию DateDiffDays(from\_date, to\_date) которая будет возвращать разницу в днях между двумя датами. Она может быть использована вместе с Now() функцией (которая возвращает текущее время), чтобы получить расстояние в днях между выбранным элементом измерения времени и текущей даты, например

|  |
| --- |
| DateDiffDays([Time].CurrentMember.StartDate, Now()) |

Кроме того, eazyBI определяет функцию DateAddDays(date, number\_of\_days), которая будет возвращать новую дату в прошлом (если *number\_of\_days* отрицательный) или в будущем (если *number\_of\_days* положителен). Например, это возвращает дату, которая составляет 5 дней от текущей даты элемента в измерении **Time**.

|  |
| --- |
| DateAddDays([Time].CurrentMember.StartDate, 5) |

Иногда может быть полезно получить элемент измерения **Time**, который соответствует текущей дате. Это может быть сделано в eazyBI со специальным свойством уровня иерархии измерения --  CurrentDateMember. Например, эти выражения будут возвращать элементы в течение месяца с текущей даты и недели с текущей даты:

|  |
| --- |
| [Time].[Month].CurrentDateMember  [Time.Weekly].[Week].CurrentDateMember |

Если нет элемента уровня измерения **Time**, то это соответствует текущей дате (например, если данные за текущий месяц или неделю ещё не импортировались), то CurrentDateMember вернёт последний период до текущей даты, который присутствует в измерении **Time** соответствующего уровня.

В eazyBI также существует также дополнительный метод конкретного уровня иерархии измерения DateMember, для которого вы можете обеспечить динамическое выражение аргумента даты и получить соответствующий элемент (или последний период до этой даты, которая присутствует в измерении **Time**). Например:

|  |
| --- |
| [Time].[Day].DateMember('7 days ago') |

Если вы хотите выбрать подмножество элементов уровня измерения между указанными датами, то вы можете использовать функцию DateBetween для фильтрации пользователей уровня, используя [date range](https://docs.eazybi.com/display/EAZYBI/Date+filters) выражения. Например, следующее выражение будет возвращать множество 7 дней из [Time].[Day] элементов уровня:

|  |
| --- |
| Filter(    [Time].[Day].Members,    DateBetween([Time].CurrentMember.StartDate,'7 days ago','today')  ) |

Значения по умолчанию (Default values)

Иногда необходимо вернуть значение по умолчанию, если какая-то мера или функция вернёт пустое значение. В этих случаях можно использовать функцию CoalesceEmpty(expression, default\_value) например:

|  |
| --- |
| CoalesceEmpty([Measures].[Store Sales], 0) |

Комментарии

Можно писать комментарии в расчётных формулах. Используйте комментарии либо для описания некоторых неочевидных сложных вычислений или при отладке расчётных формул.

|  |
| --- |
| -- one line comment  expression -- comment until end of line  /\* multi line     comment  \*/ |

Специальные комментарии с аннотациями

Доступно на eazyBI Cloud или начиная с версии 4.0 eazyBI дополнения для JIRA или Private eazyBI.

Вы можете добавлять специальные комментарии в формулах с аннотациями для вычисляемого элемента. Эти аннотации используются в пользовательском интерфейсе eazyBI для группировки вычисляемых элементов и разрешения или запрета определенных действиий:

-- annotations.group="..."  
Помещает вычисляемый элемент в указанной группе в разделе **Calculated members** измерения в report builder.

-- annotations.disable\_drill\_into=true  
Запрещает действие **Drill into** для вычисляемого элемента.

-- annotations.disable\_drill\_across=true   
Запрещает действие  **Drill across** для вычисляемого элемента.

-- annotations.disable\_drill\_through=true   
Запрещает действие  **Drill through** для вычисляемого элемента.

-- annotations.disable\_drill=true   
Запрещает все drill действия для вычисляемого элемента.

Ссылка на все MDX функции https://docs.eazybi.com/display/EAZYBI/MDX+Function+Reference