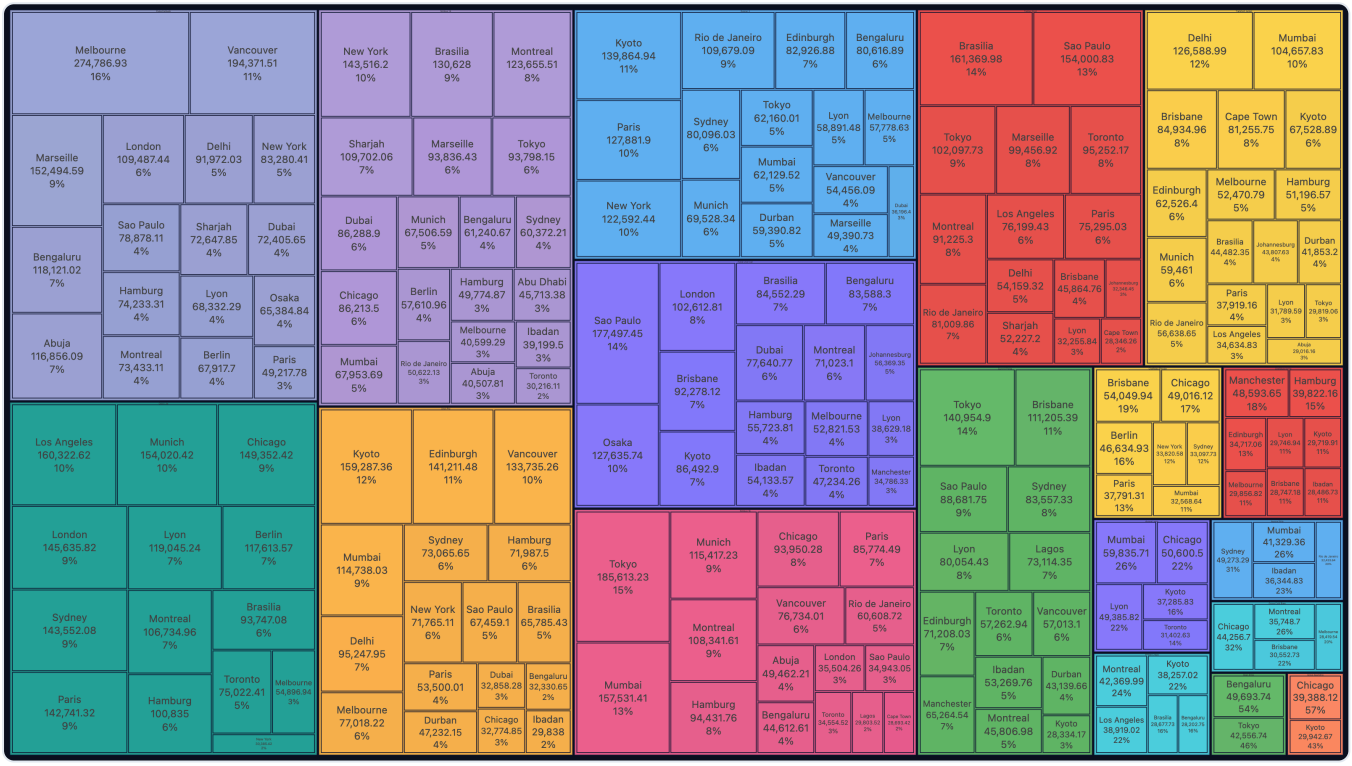
DataLad Insights

DataLad Insights — это модуль интерактивной визуализации данных, который превращает любой Parquet-датасет в набор профессиональных дашбордов. Аналитик указывает источник данных, выбирает колонки и агрегации — система возвращает готовый график без написания кода, SQL запросов в интерфейсе или создания собственных визуализаций.

Insights встроен в платформу DataLad и работает в паре с остальными сервисами (сбор событий, хранилище ключ-значение, MemTables). Визуализации доступны в отдельной вкладке «Аналитика» в левом меню.



Два уровня аналитики

Insights разделяет сценарии «построить график по любым данным» и «построить график по временным рядам». Первый сценарий работает с произвольным Parquet-датасетом; второй требует, чтобы данные соответствовали **канонической схеме временных рядов** (описана ниже).

УРОВЕНЬ	ТРЕБОВАНИЯ К ДАННЫМ	ДОСТУПНЫЕ ГРАФИКИ
Произвольные данные	Любой Parquet — колонки выбираются в интерфейсе	Treemap, Sunburst, Radial bar, Sankey
Временные ряды	Каноническая схема	Мультисерийный линейный график со слайдером и аннотациями

UI автоматически определяет, соответствует ли текущий датасет канонической схеме, и включает вкладку **«Время»** в списке семейств графиков без дополнительных настроек.

Каноническая схема временных рядов

Это восемь колонок в формате long-table, которые делают датасет пригодным для построения временных рядов. Шесть из них обязательные; `description` и `annotation` — необязательные и отвечают за разные задачи (см. ниже).

КОЛОНКА	ТИП	ОБЯЗАТЕЛЬНА?	НАЗНАЧЕНИЕ	ПРИМЕР
`stat_date`	Строка (`YYYY-MM-DD`) или `Date32` / `Date64`	да	Партиция по дате. Дублирует `ts`, но удобна для фильтров `WHERE stat_date = '…'`.	`"2026-02-25"`
`ts`	`Timestamp(*, *)` или ISO 8601-строка	да	Точный момент события. Строка и таймстамп обрабатываются единообразно — при запросе выполняется `CAST(ts AS TIMESTAMP)`.	`"2026-02-25T09:00:00"`
`granularity`	Строка (`minute` | `hour` | `day` | `week` | `month`)	да	Разрешение хранения. Интерфейс умеет укрупнять гранулярность «на лету» (`date_trunc`).	`"hour"`
`count_name`	Строка	да	Название метрики. В одной точке времени может быть несколько строк — по одной на каждую метрику.	`"total_events"`, `"resolved_events"`, `"distinct_item_count"`
`entity_name`	Строка	да	Объект, к которому относится метрика. Если фильтр по сущностям пуст — значения суммируются по всем объектам.	`"acme"`, `"globex"`
`value`	Численный (`Int*`, `UInt*`, `Float*`, `Decimal`)	да	Само измерение. При агрегациях приводится к `Float64`.	`2618`
`description`	Строка (необязательная)	нет	**Документация строки.** Поясняет, что измеряет метрика, историю данных, пользовательские заметки — тот текст, который аналитик видит в инспекторе	`"Поток входящих событий до дедупликации"`

КОЛОНКА	ТИП	ОБЯЗАТЕЛЬНА?	НАЗНАЧЕНИЕ	ПРИМЕР
			данных, но не на графике.	
`annotation`	Строка (необязательная; как правило пуста)	нет	**Текст маркера на графике.** Непустые значения превращаются в вертикальные янтарные линии на временном ряду. Используется точечно — обычно одна аннотация на событие.	`"релиз v2.4"`

Чем различаются description и annotation

Эти две строковые колонки похожи на первый взгляд, но играют противоположные роли и используются разными частями интерфейса:

- **`description` — история данных и контекст.** Как правило один и тот же текст повторяется для всех строк одной `count_name`, потому что он описывает саму метрику, а не конкретное событие. Это подпись, которую аналитик видит при просмотре строки; в качестве маркера на графике она не отображается.
- **`annotation` — маркер в точке времени.** Почти всегда пуста. Непустые значения появляются на считанных строках (релизы, начало/конец инцидента, запуск кампании), и именно эти строки превращаются в вертикальные линии на графике. На маркеры влияет только колонка `annotation`; к `description` система для этих целей не обращается.

Гибкость типов

Валидатор принимает любые строковые представления Arrow (`Utf8`, `LargeUtf8`, `Utf8View`), все разрешения таймстампов (секунды, миллисекунды, микросекунды, наносекунды) — с временной зоной или без — и любой числовой тип для `value`. Это значит, что датасеты, собранные разными инструментами (Arrow CSV, `csv-to-parquet` из состава DataLad, сервис сбора событий Ledger, pandas, Spark), попадают в один и тот же контракт без предварительного преобразования.

Проверка выполняется каждый раз при загрузке схемы: если датасет подходит — вкладка «Время» включается; если нет — доступны только «Иерархия» и «Поток».

Пример канонических строк

Каждая строка — одно измерение в тройке (время, метрика, сущность). `description` несёт документацию метрики для инспекции, а `annotation` пуста везде, кроме нескольких строк, которые должны стать маркерами на графике:

```
stat_date   ts                   granularity  count_name           entity_name   value     descripti
----------  -------------------  -----------  ------------------   ------------  -------   ---------
2026-01-01  2026-01-01T00:00:00  hour         total_events         acme          126.35    "Поток вх
2026-01-01  2026-01-01T00:00:00  hour         resolved_events      acme          103.46    "События,
2026-01-01  2026-01-01T00:00:00  hour         distinct_item_count  acme           42.00    "Уникальн
…
2026-02-25  2026-02-25T09:00:00  hour         total_events         initech       153.91    "Поток вх
```

Для одной сущности на горизонте 90 дней в почасовой гранулярности объём составляет примерно `90 × 24 × N_метрик` строк. Входящий в поставку демонстрационный датасет — 25 920 строк в Parquet-файле размером 96 КБ.

Семейства визуализаций

UI группирует типы графиков в три семейства. При переключении семейства меняется панель управления; при переключении типа внутри одного семейства (Treemap → Sunburst → Radial) перерисовка выполняется мгновенно без повторной выборки данных.

Иерархия

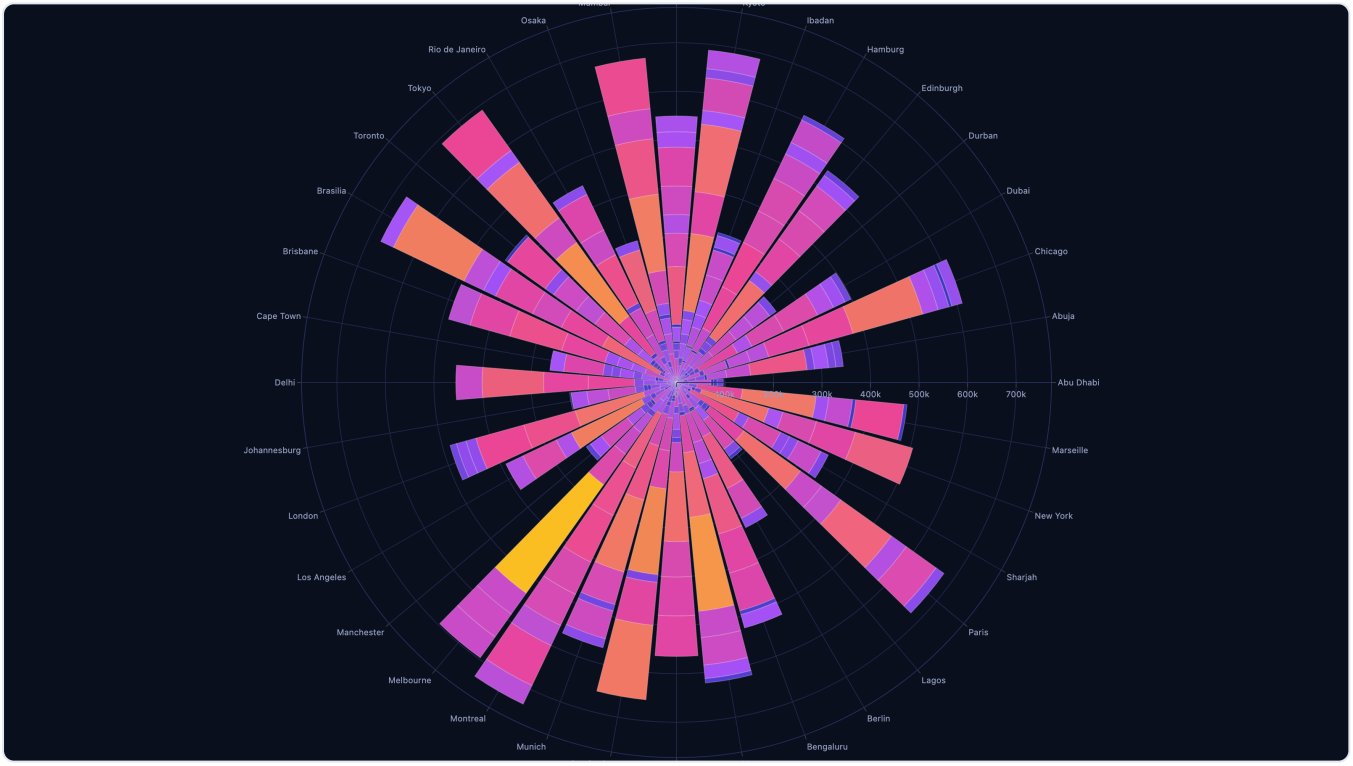
Пользователь выбирает колонку-метку (`label_col`), при необходимости — колонку-родителя (`parent_col`), а также агрегацию (`sum` , `avg` , `min` , `max` , `count` , `count_distinct`) и опциональный SQL-фильтр `WHERE` . Система возвращает подготовленные данные для Plotly:

```
SELECT
  CAST(<label> AS VARCHAR)  AS label,
  CAST(<parent> AS VARCHAR) AS parent,   -- NULL для плоского дерева
  <agg>(<value>)            AS value
FROM t [WHERE (<where>)]
GROUP BY label, parent
ORDER BY value DESC NULLS LAST
LIMIT <limit>
```

При указании родителя сервис автоматически синтезирует промежуточные узлы с агрегированными значениями, так что Plotly получает корректное дерево.

Три варианта отображения на одном и том же наборе данных:

- **Treemap** — прямоугольная упаковка плиток (используется по умолчанию).
- **Sunburst** — концентрические кольца; та же иерархия в радиальной форме.
- **Radial bar** — круговая столбчатая диаграмма. Если в данных есть иерархия, оставляются только листья, промежуточные узлы удаляются.

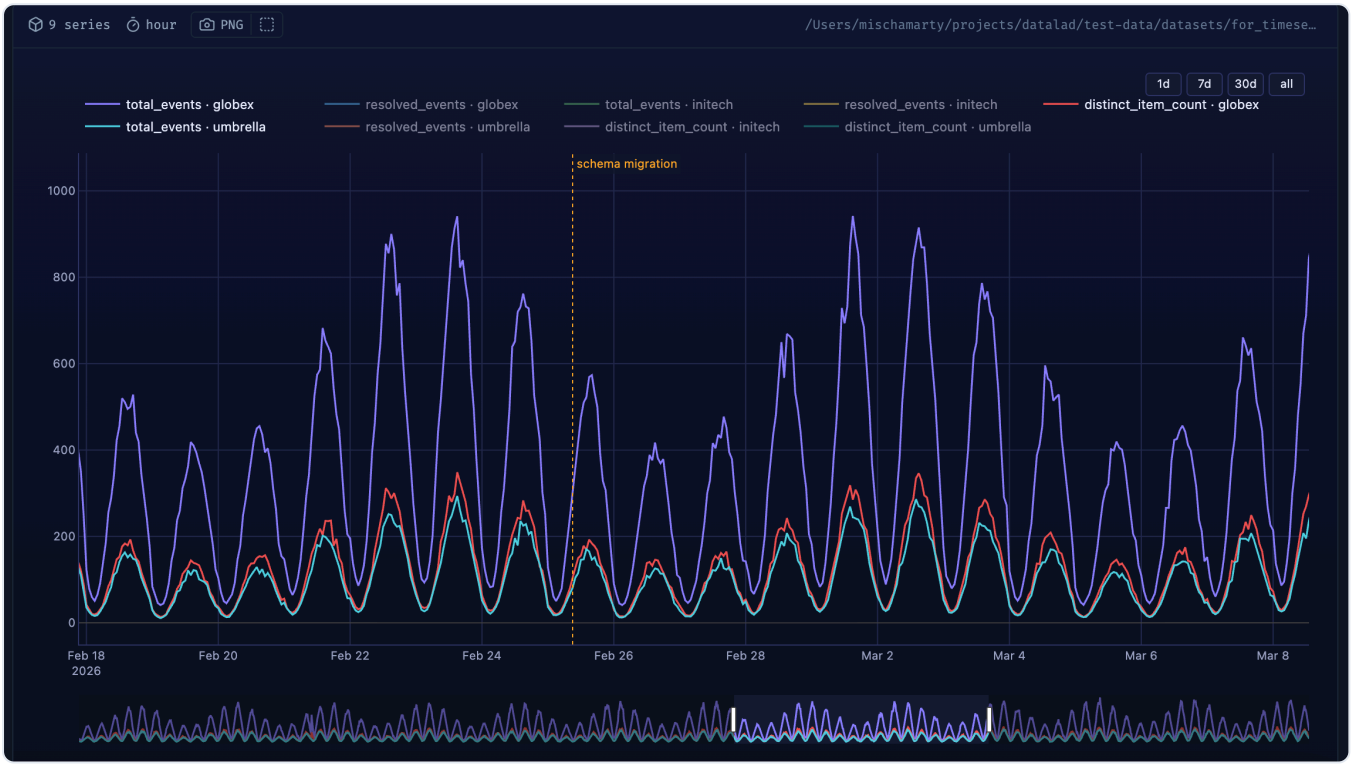


Поток (Sankey)

Пользователь выбирает колонку-источник (`source_col`), колонку-цель (`target_col`), колонку значений и агрегацию. Сервис формирует по одной связи на каждую пару `(source, target)` и приводит результат к формату Sankey-диаграммы Plotly: `{ nodes, links }`. Узлы дедуплицируются по значению — если «Europe» встречается и как источник, и как цель, он пройдёт через единственный узел.

```
SELECT CAST(<source> AS VARCHAR) AS source,
       CAST(<target> AS VARCHAR) AS target,
       <agg>(<value>) AS value
FROM t [WHERE (<where>)]
GROUP BY source, target
ORDER BY value DESC
LIMIT <limit>
```

Время (временные ряды)



Доступно только для датасетов с канонической схемой. Интерфейс подгружает уникальные значения `count_name` и `entity_name` и предлагает их в мультивыборе. Диапазон времени задаётся вручную или одной из пресет-кнопок (1 день, 7 дней, 30 дней, 90 дней). Гранулярность переключается независимо от того, как хранится датасет — система выполняет `date_trunc` на лету.

Шаблон запроса:

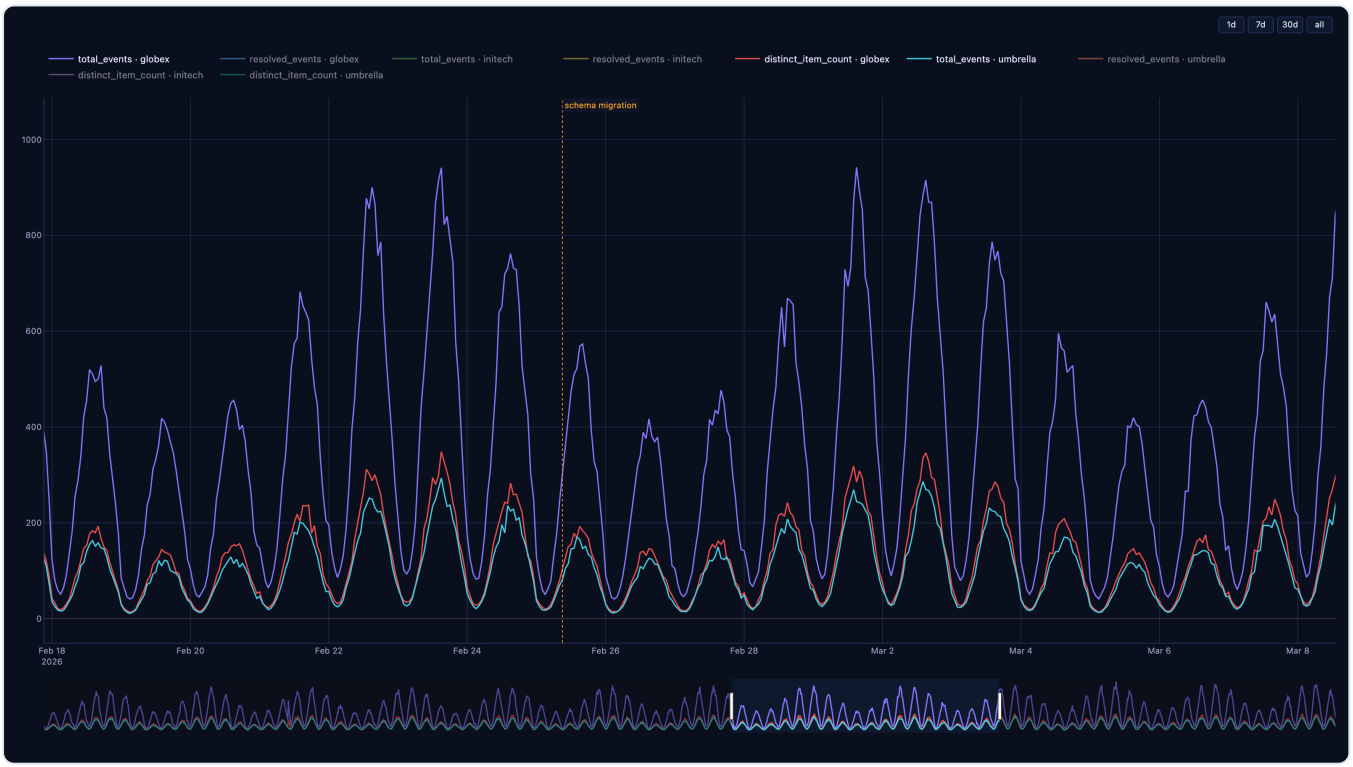
```
SELECT date_trunc('<gran>', CAST(ts AS TIMESTAMP)) AS bucket,
       CAST(count_name AS VARCHAR)   AS count_name,
       CAST(entity_name AS VARCHAR)  AS entity_name,  -- NULL при агрегации
       CAST(SUM(value) AS DOUBLE)    AS value
FROM t
WHERE count_name IN (…)
  [AND entity_name IN (…)]
  [AND CAST(ts AS TIMESTAMP) BETWEEN … AND …]
  [AND (<where>)]
GROUP BY bucket, count_name [, entity_name]
ORDER BY bucket, count_name
```

Формат ответа:

```
{
  "series": [
    {
      "name": "total_events · globex",
      "count_name": "total_events",
      "entity_name": "globex",
      "x": ["2026-02-25T09:00:00", …],
      "y": [912.4, …]
    }
  ],
  "annotations": [
    { "ts": "2026-02-25T09:00:00", "text": "schema migration",
      "count_name": "total_events", "entity_name": "initech" }
  ],
  "granularity": "hour",
  "truncated": false
}
```

Количество серий ограничено 24 (достаточно для любого осмысленного дашборда); они сортируются по пиковому значению, чтобы самая заметная серия отображалась первой в легенде.

Аннотации (колонка `annotation`)



Любая строка с непустым значением в колонке **`annotation`** превращается в маркер на графике. Система выполняет отдельный SQL-запрос параллельно с основным, применяет те же фильтры (по метрикам, сущностям, временному окну) и возвращает до 200 маркеров. UI рисует каждый как:

- Вертикальную пунктирную янтарную линию, проходящую через всю область графика.
- Небольшую текстовую метку над графиком, привязанную к соответствующему времени.

К колонке `description` система намеренно не обращается — это не визуальный маркер, а документация строки. Если датасет был создан ещё до разделения и в нём нет колонки `annotation`, сервис возвращает пустой список маркеров без ошибки.

Чтобы разметить собственные релизы, инциденты или маркетинговые кампании, достаточно записать значение в колонку `annotation` на строке соответствующей временной точки. Чтобы маркер не дублировался по числу метрик в точке, рекомендуется прикреплять его только к одной канонической `count_name` — например, `total_events`.

Цветовое оформление

Insights отходит от стандартной радужной палитры Plotly ради более сдержанного корпоративного стиля. Используются две стратегии в зависимости от того, содержит ли визуализация иерархию:

1. **Иерархическое отображение.** Каждый корневой узел (например, регион) получает отдельный цвет из кураторской палитры на 12 цветов. Все его потомки окрашены тем же цветом с прозрачностью от 0.55 до 0.85, пропорционально величине значения. Благодаря этому «регион» на treemap воспринимается как цельный блок, а отдельные плитки внутри остаются различимыми.
2. **Плоское отображение.** Применяется пятиступенчатый градиент `indigo-950 → indigo-700 → purple-500 → pink-500 → amber-400`, подобранный под тёмный фон панели. Значения предварительно пропускаются через `log(v + 1)`, чтобы длинный хвост мелких значений тоже использовал всю палитру, а не сжимался в одну тёмную зону.

Radial bar всегда использует плоскую стратегию (это по определению плоский график). Sankey использует прозрачный фиолетовый для связей и фирменный `#8a7cff` для узлов. Временные ряды циклически перебирают палитру из 15 цветов; порядок серий стабилен между перерисовками.

Запуск сервиса Insights

Каждый экземпляр Insights — отдельный сервис с собственным портом, собственной Swagger-документацией и собственной конфигурацией. Создание выполняется через интерфейс: на странице «Сервисы» нажмите **Add service**, выберите тип `insights`, заполните TOML-конфигурацию.

Минимальная конфигурация

```
[insights]
# scan_row_budget     = 2_000_000
# max_files_per_scan  = 5_000
# exec_timeout_secs   = 30
# max_treemap_slices  = 500

# Закреплённые датасеты — именованные ярлыки, которые появятся в списке
# «Pinned» в интерфейсе. Пользователи могут работать и с произвольными путями
# через поле «Custom path» — закрепление лишь добавляет удобные имена.
[[insights.datasets]]
id          = "sales_demo"
name        = "Sales (demo)"
path        = "./test-data/datasets/for_treemap"
description = "Демонстрация семейств Hierarchy и Flow (384 строки)"

[[insights.datasets]]
id          = "events_demo"
name        = "Events (demo)"
path        = "./test-data/datasets/for_timeseries"
description = "Демонстрация канонических временных рядов (25 920 строк, 90 дней)"
```

Ключевые параметры

ПАРАМЕТР	НАЗНАЧЕНИЕ	ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ
`bind`	Адрес и порт привязки сервиса	`0.0.0.0:8084`
`scan_row_budget`	Максимум строк, просматриваемых в одном запросе-графике	2 000 000
`max_files_per_scan`	Максимум parquet-файлов, включаемых в один запрос	5 000
`exec_timeout_secs`	Таймаут выполнения одного запроса (секунды)	30
`max_treemap_slices`	Максимальное число плиток в одном treemap/sankey	500

Эти пороги защищают сервис от случайного запуска гигантских выборок (например, от случайной подстановки корня продакшн-озера данных в поле пути).

Справочник HTTP API

Базовый URL внутри сервиса: `http://<bind>/api/v1/insights/`. Через UI-прокси доступ выполняется по `/api/svc/<port>/api/v1/insights/`. Все ответы в формате JSON; ошибки — в общем формате `{ error, code, details }`.

МЕТОД	ПУТЬ	НАЗНАЧЕНИЕ
`GET`	`/health`	Проверка готовности.
`GET`	`/status`	Текущая конфигурация и количество закреплённых датасетов.
`GET`	`/datasets`	Список закреплённых датасетов из TOML.
`POST`	`/schema`	`{ path }` → колонки, число файлов, флаг `timeseries_capable`.
`POST`	`/query/treemap`	`{ path, label_col, parent_col?, value_col?, agg, where?, limit? }` → `{ ids, labels, parents, values, total, truncated }`.
`POST`	`/query/sankey`	`{ path, source_col, target_col, value_col?, agg, where?, limit? }` → `{ nodes, links, total, truncated }`.
`POST`	`/query/distinct`	`{ path, column, limit? }` → `{ values, truncated }`. Используется UI для наполнения мультивыборов.
`POST`	`/query/timeseries`	`{ path, count_names[], entity_names[]?, granularity?, from?, to?, where? }` → `{ series, annotations, granularity, truncated }`.

Полная OpenAPI-спецификация доступна по адресу `/docs` на порту сервиса.

Демонстрационные данные

В поставку DataLad входят два демонстрационных датасета в каталоге `test-data/datasets/`.

`for_treemap/sales.parquet`

Глобальные продажи e-commerce, 384 строки, 11 колонок. Два независимых измерения иерархии и несколько числовых метрик:

- Географическое: `region → country → city` (4 × 3 × 3 = 36 городов)
- Товарное: `category → subcategory → product` (4 × 2 × 2 = 16 товаров)
- Метрики: `units`, `revenue`, `orders`, `customers`, `quarter`

Сценарии визуализации:

LABEL	PARENT	VALUE	AGG	ЧТО УВИДИТЕ
country	region	revenue	sum	«Карта выручки мира» в виде treemap
city	country	revenue	sum	Детализация городов в разрезе стран
product	category	units	sum	Проданные единицы по товарам
subcategory	category	revenue	avg	Средняя выручка по подкатегориям
(source=region, target=category)		revenue	sum	Sankey: потоки выручки из регионов в категории

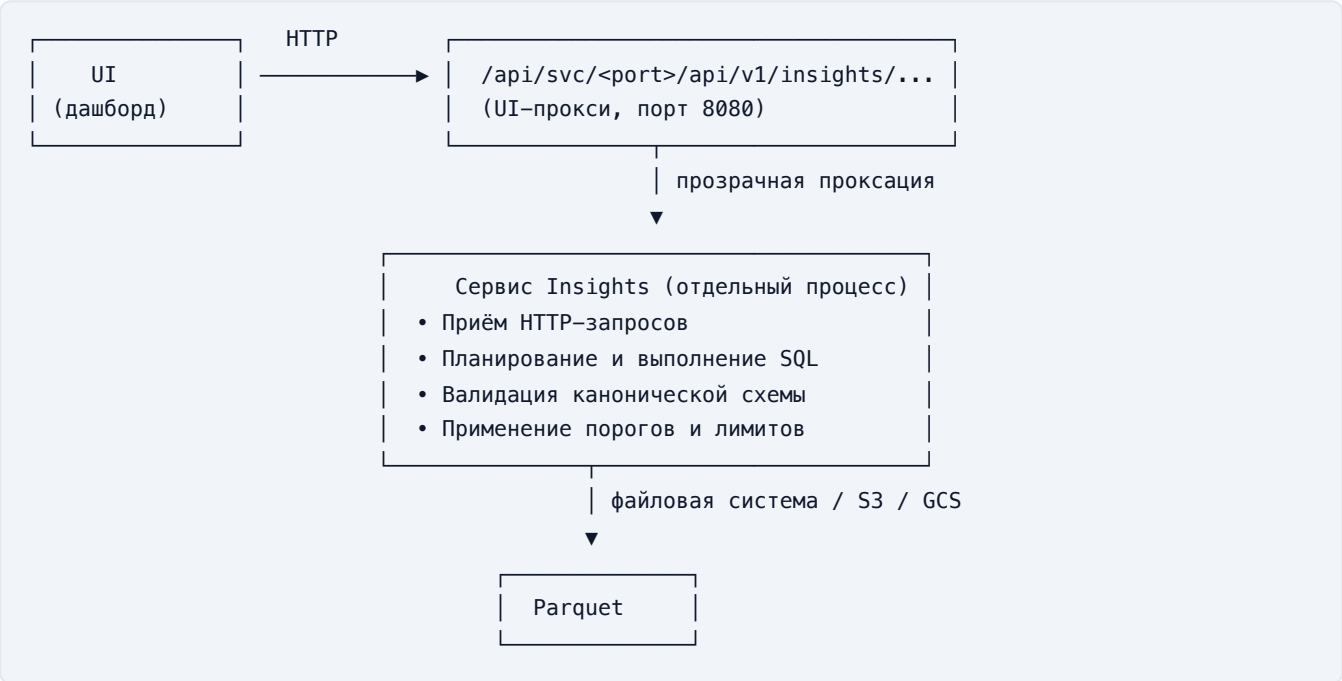
`for_timeseries/events.parquet`

Канонические временные ряды, 25 920 строк, 90 дней × 24 часа × 3 метрики × 4 сущности.

- Метрики: `total_events`, `resolved_events`, `distinct_item_count`
- Сущности: `acme`, `globex`, `initech`, `umbrella`
- Характер сигнала: суточный цикл + недельный цикл + постепенный рост на 15% + гауссовский шум + 5 заранее размеченных всплесков (два инцидента, маркетинговая кампания, перебой) + 7 маркеров в колонке `annotation` (при этом `description` несёт стабильную документацию по каждой `count_name`).

Генератор использует фиксированное зерно случайных чисел (`random.seed(42)`), поэтому все графики воспроизводятся один-в-один.

Архитектура (с точки зрения потока данных)



Сервис Insights работает только на чтение: у него нет собственного состояния и он не принимает данные. Все запросы выполняются SQL-движком в реальном времени по исходному Parquet. Пороги на количество строк, файлов и секунд защищают сервис от случайных больших выборок.

Для локальных путей выполняется рекурсивный обход каталогов — это гарантирует, что многоуровневые Hive-разметки (`schema=…/date=…/bucket=…/part.parquet`) полностью попадают в выборку.

Решение проблем

СИМПТОМ	ПРИЧИНА	ДЕЙСТВИЕ
Кнопка «Время» заблокирована	Датасет не соответствует канонической схеме	Загрузите датасет с колонками `stat_date, ts, granularity, count_name, entity_name, value` — при успехе в заголовке схемы появится зелёный маркер «time-series ready».
`unexpected label column type`	Сервис работает с устаревшей версией	Остановите и запустите сервис заново на странице «Сервисы».
`Parser error: Error parsing timestamp from 'YYYY-MM-DDTHH:MM'`	Поле `datetime-local` не включает секунды	Обрабатывается автоматически в UI; при вызове API напрямую дописывайте `:00`.
`no parquet files found under <path>`	Путь некорректен или файлы не имеют расширения `.parquet`	Проверьте, что путь существует и содержит файлы с расширением `.parquet` (регистр не важен). Можно указать и отдельный файл.
`too many parquet files (>N)`	Поддерево превышает `max_files_per_scan`	Увеличьте лимит в конфигурации или укажите более узкий путь.
404 на `/query/sankey` или `/query/timeseries`	Сервис обновлён, но процесс не перезапущен	Остановите и запустите сервис заново.

Интеграция с платформой

- **Сервисы** — Insights регистрируется наравне с другими сервисами DataLad, имеет собственный порт и собственную OpenAPI-документацию.
- **Закреплённые датасеты** — удобные именованные ссылки на источники данных, не ограничивают пользователя: любой аналитик может работать с произвольным путём через поле «Custom path».
- **Произвольные объектные хранилища** — поддерживаются протоколы `file://`, `s3://`, `gs://`; учётные данные настраиваются в TOML-конфигурации сервиса.
- **Экспорт** — любая визуализация экспортируется в PNG с двумя вариантами фона: тёмный (по умолчанию, подходит для слайдов) и прозрачный (для наложения на произвольный фон).