Как работают данные: Parquet против CSV

Почему формат важен в аналитике данных

Почему разговор о данных начинается с CSV?

В современной аналитике CSV остаётся точкой входа для большинства специалистов, несмотря на все его недостатки. Это не случайность — это реальность работы с данными.

Реальность

Первый контакт с данными — часто выгрузка из БД в CSV. Простота экспорта делает его универсальным форматом обмена.

Исторический долг

Легаси системы живут годами. Миграция не всегда окупается для небольших команд и дашбордов с ограниченным объёмом данных.

Контраст

Через недостатки строкового формата для аналитики лучше видна ценность колоночного подхода к хранению.

Эффективность

Формат задаёт потолок производительности — даже самый быстрый движок лишь приближается к этому пределу.

Ключевой инсайт

Формат данных — это архитектурное решение, которое определяет границы возможного для всей системы аналитики.

Что такое CSV на самом деле

CSV — это строковый формат без единого стандарта. Каждая строка хранится последовательно, одна за другой, как текст в обычном файле.

```
user_id,name,age,city,amount,category,timestamp,год,месяц
0,Даша,58,Казань,371.8,быт,2024-01-01 00:00:00,2024,01
1,Егор,65,Москва,753.41,быт,2024-01-01 00:01:00,2024,01
2,Артур,39,Екатеринбург,133.78,быт,2024-01-01 00:02:00,2024,01
3,Егор,34,Екатеринбург,393.37,развлечения,2024-01-01 00:03:00,2024,01
4,Егор,40,Москва,36.02,еда,2024-01-01 00:04:00,2024,01
5,Марат,45,Москва,30.89,еда,2024-01-01 00:05:00,2024,01
6,Артур,46,Краснодар,22.45,транспорт,2024-01-01 00:06:00,2024,01
7,Артур,54,Москва,588.86,еда,2024-01-01 00:07:00,2024,01
8,Артур,22,Москва,101.89,быт,2024-01-01 00:08:00,2024,01
9,Егор,63,Новосибирск,97.78,быт,2024-01-01 00:09:00,2024,01
10,Даша,63,Краснодар,356.99,развлечения,2024-01-01 00:10:00,2024,01
```

Полное чтение для частичного доступа

Чтобы прочитать одну колонку, нужно распарсить всю строку целиком. Нет способа пропустить ненужные данные.

Схема в голове, а не в файле

Типы данных — это не часть формата, а договорённость между системами. Файл не знает, что в колонке числа или даты.

Хрупкость по дизайну

Легко сломать: кавычки внутри значений, разделители в данных, пустые поля. Каждая система парсит по-своему.

Только текст

Нет поддержки бинарных данных и надёжной десериализации сложных типов. Всё приводится к строкам.

Колоночность как идея

Колоночное хранение — это прямой ответ на боль аналитических запросов. Аналитика по природе своей режет данные вертикально, по колонкам.

Типичные аналитические запросы

- Пользователи старше 30 лет
- Транзакции из Москвы
- Суммы больше 10 000 рублей
- Средний чек по категориям

В каждом случае нам нужны 1-2 колонки из десятков возможных.
Но строковый формат заставляет читать всё.

Строки (CSV)

Читаем все колонки последовательно, даже если нужна только одна. Полный парсинг каждой строки.

Колонки (Parquet)

Читаем только нужные колонки. Пропускаем остальное физически, без парсинга лишних данных.

Колоночное хранение превращает вертикальные срезы данных из дорогой операции в естественную и быструю.

Два сценария работы с данными

Формат хранения данных определяется не абстрактными предпочтениями, а конкретным сценарием использования. OLTP и OLAP — два противоположных подхода к работе с данными.

OLTP Online Transaction Processing

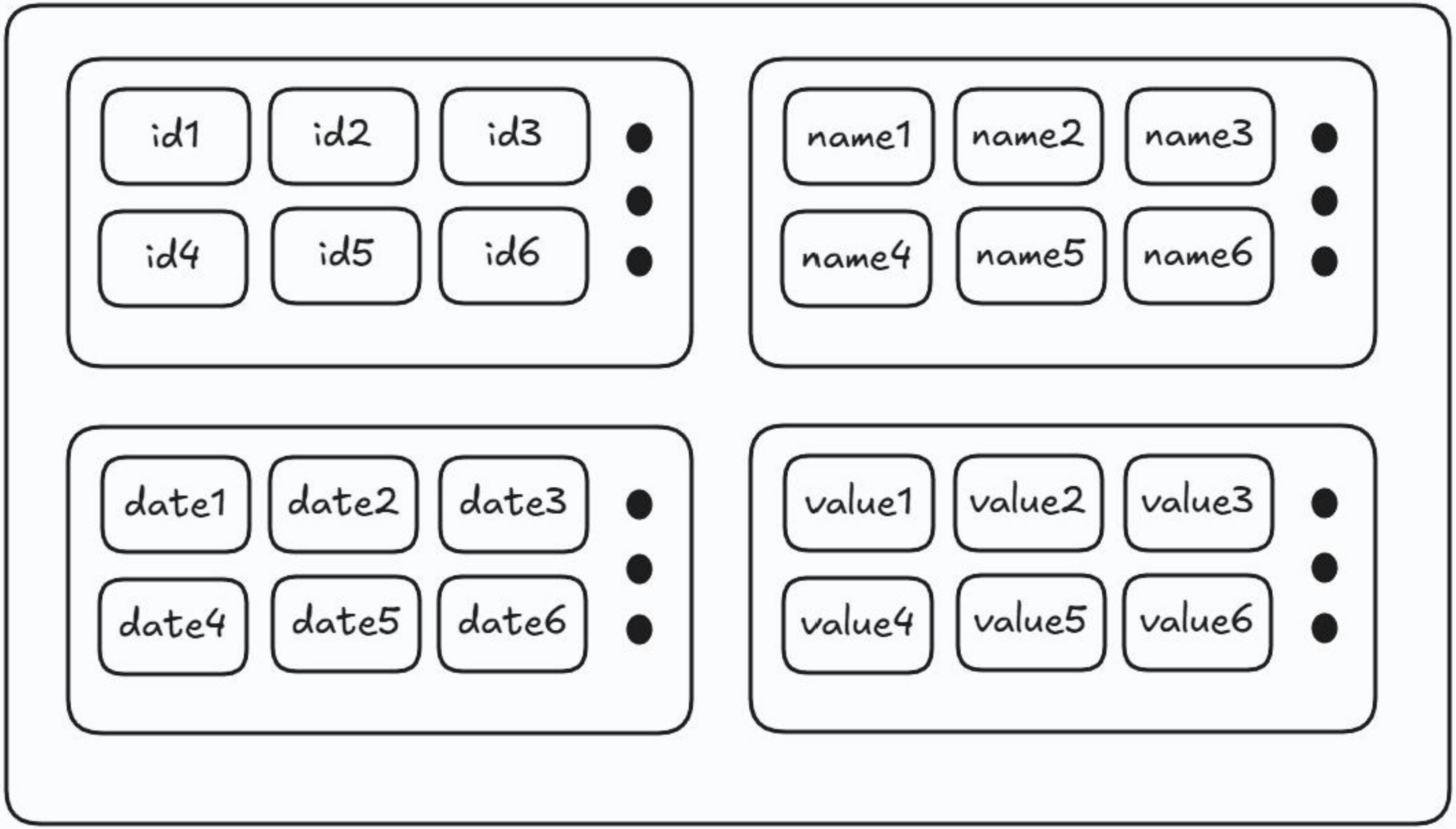
- Частая запись
- Обновления и удаления
- **Пример**: оформление заказа

OLAP Online Analytical Processing

- Частое чтение
- Работа с миллионами строк
- Только добавление (append)
- **Пример**: дашборд продаж за год

Формат хранения должен соответствовать сценарию: строковый — для транзакций, колоночный — для аналитики.

OLAP логика


id1
id2
id3
id4
id5
id6
name1
name2
name3
name4
name5
name6
date1
date2
date3
date4
date5
date6
value1
value2
value3
value4
value5
value6

Что такое Parquet

Parquet — это открытый колоночный формат от Apache Foundation, созданный специально для аналитики над большими данными.

1

2013

Создан в экосистеме Hadoop для эффективной работы с петабайтами данных в распределённых системах.

2

Стандарт де-факто

Стал стандарртом де-факто для аналтикии в Spark, Hive, Trino, Presto.

3

Встроенные возможности

Хранит данные по колонкам плюс полные метаданные внутри самого файла. Самодостаточный формат.

4

Часть формата

Схема, сжатие и энкодинги — не внешние слои абстракции, а неотъемлемая часть спецификации формата.

АРХИТЕКТУРА

Как устроен файл Parquet

Заголовок

Magic number и версия

Колонки

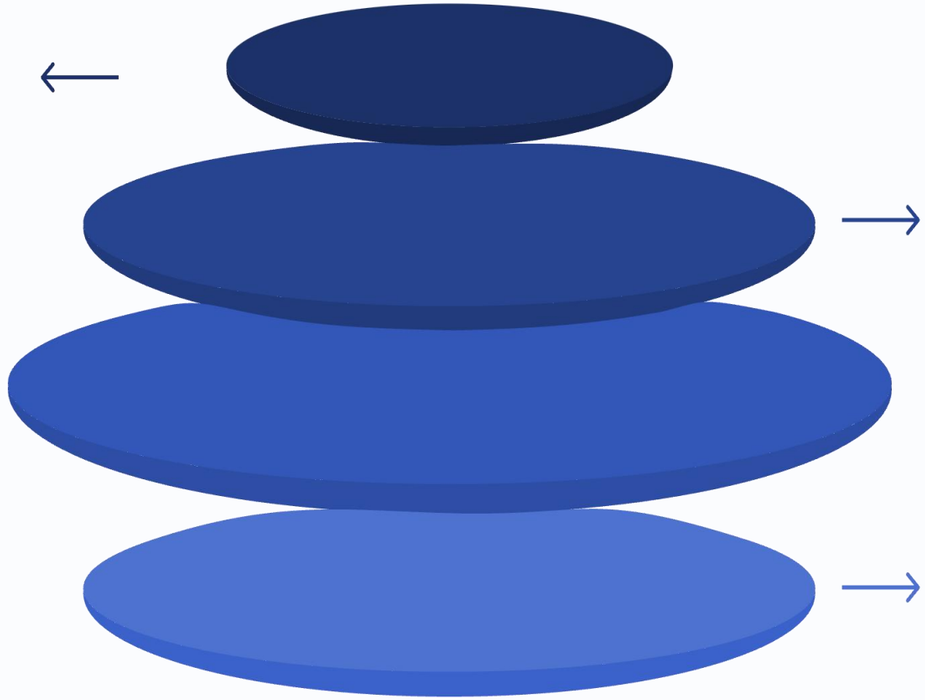
Chunks: id, name, age, city

Группы строк

Row groups с блоками данных

Футер

Схема и метаданные файла



Файл Parquet представляет собой самодостаточную структуру, где данные и метаданные объединены. Каждая колонка хранится отдельно, что позволяет читать только нужные данные.

Колоночное хранение

Данные каждой колонки (id, name, age, city) хранятся в отдельных блоках, независимо друг от друга.

Метаданные в файле

Схема с типами данных (int, string, timestamp), статистика по блокам (min/max/nulls), информация о сжатии — всё внутри.

Партиционирование для Parquet

Партиционирование — это не встроенная возможность формата Parquet, а организационный паттерн на уровне файловой системы.

```
данные/
  ├── год=2024/
  │   └── месяц=02/
  │       └── часть-001.parquet
  └── год=2025/
      └── месяц=01/
          └── часть-001.parquet
```

Два уровня отсечения

01

Партиции

Пропускаем целые директории (на уровне ФС)

02

Блоки

Пропускаем части файла (статистика в footer)

- Запрос WHERE год=2024 → читаем только одну папку
- Внутри — ещё отсекаем блоки по статистике

Схема, сжатие, энкодинги: CSV vs Parquet

Разница между форматами — это разница в том, что встроено в спецификацию, а что приходится делать снаружи.

Аспект	CSV	Parquet
Схема	Заголовок — просто текст. Типы данных — догадки парсера при чтении.	Строгие типы, опциональность полей, версионирование схемы без поломки обратной совместимости.
Сжатие	Файл целиком (gzip, bzip2). Распаковываем 100% данных, даже если нужна одна колонка.	Сжатие по колонкам и блокам. Распаковываем только нужное. Snappy, ZSTD, LZ4 на выбор.
Энкодинги	Отсутствуют. Данные хранятся «как есть» в текстовом виде.	Dictionary, RLE, Delta — семантическое сжатие на основе паттернов в данных.
Метаданные	Внешние. Схема и типы живут в коде или документации.	Встроенные. Файл содержит всю информацию о своей структуре и содержимом.

Реальные цифры: 10 млн строк

Датасет: 10 миллион строк с данными пользователей — город, возраст, история транзакций. Типичный кейс аналитики среднего масштаба.

Операция	CSV	Parquet	Ускорение
Размер	835.4 МБ	173.4 МБ	~5
Запись	37.3 с	7.8 с	~5
Чтение колонки	7.219 с	0.071 с	~100
Фильтр (город)	13.834 с	2.177 с	~6
Фильтр (партиции)	2.160 с	0.014 с	~150

Компромиссы Parquet

Parquet хорош для аналитики. Универсальных решений не существует, есть только компромиссы.

- **Не человекочитаем:** Откроете в блокноте или текстовом редакторе, то увидите бинарный мусор. Нужны специальные инструменты для просмотра.
- **Нет случайной записи:** Дописать одну строку в конец файла, означает переписать весь файл.
- **Требует буферизации:** Для эффективной записи нужны десятки тысяч строк в памяти. Построчная запись убивает производительность.
- **Сложно отлаживать:** Нужны специальные инструменты: parquet-tools, pandas, DuckDB. Быстро «заглянуть в файл» не получится.
- **Мелкие файлы:** Тысячи файлов по 1 МБ работают хуже одного на 1 ГБ. Накладные расходы на метаданные растут.
- **Не для частых обновлений:** Формат append-only, нет UPDATE или DELETE операций. Изменить данные, означает переписать файл или партицию.

Когда что выбирать

Выбор формата — это выбор архитектуры. Не существует лучшего формата в вакууме, есть только подходящий для конкретной задачи.

Критерий	CSV / NDJSON	Parquet
Объём данных	Открывается в редакторе, помещается в память	Важно регулярность запросов чтения (эффективность становится критичной)
Потребитель	Человек (Excel, визуальный просмотр, отладка)	Машина (Spark, Trino, аналитические движки)
Типичная задача	Разовая выгрузка, обмен данными, отладка, ручной анализ	Регулярная аналитика, фильтрация, агрегация, dashboard'ы
Паттерн записи	Поток событий, построчная запись, логи в реальном времени	Батч-загрузка, пакетная обработка, ETL пайплайны
Частота обновлений	Постоянное дописывание новых строк, append-only логи	Редкие обновления

Помните: формат — это не просто способ сохранить байты на диск. Это архитектурное решение, которое определяет эффективность всей системы работы с данными. Инструментов много — важно понимать трейдофы каждого и выбирать осознанно.

Подписывайтесь!

В следующей лекции:

Работа с данными через DataFrame API

Если вам понравилось – заходите на телеграм:

t.me/marat_notes

www.youtube.com/@marat_notes

https://vkvideo.ru/@club231048746

Все примеры кода и материалы доступны на GitHub:

github.com/MaratNotes/marat_notes

Перейти на YouTube

Открыть GitHub