

Системы управления базами данных MongoDB и ClickHouse в Python.

Урок 3



План курса

1

Основы клиент-серверного взаимодействия. Парсинг API.

2

Парсинг HTML. BeautifulSoup.

3

СУБД MongoDB и ClickHouse в Python

4

Парсинг HTML. XPath.

5

Scrapy.

6

Scrapy. Парсинг фото и файлов.

7

Selenium в Python.


8

Работа с данными.

9

Инструменты разметки наборов данных.

Пример сайта, содержащего информацию



ВЕДЬМАК
ДИКАЯ ОХОТА

Вы — Геральт из Ривии, наемный убийца чудовищ. Вы путешествуете по миру, в котором бушует война и на каждом шагу подстерегают чудовища. Вам предстоит выполнить заказ и найти Цири — Дитя Предназначения, живое оружие, способное изменить облик этого мира.

ВСЕ ОБЗОРЫ: **Крайне положительные** (545,650) *

ДАТА ВЫХОДА: 18 мая. 2015

РАЗРАБОТЧИК: CD PROJEKT RED
ИЗДАТЕЛЬ: CD PROJEKT RED

Популярные метки для этого продукта:

[Открытый мир](#) [Ролевая игра](#) [Глубокий сюжет](#) [+](#)

Языки:

	Интерфейс	Озвучка	Субтитры
русский	✓	✓	✓
английский	✓	✓	✓
французский	✓	✓	✓
итальянский	✓		✓
немецкий	✓	✓	✓

[Просмотреть все поддерживаемые языки \(16\)](#)

20000 Jan 2016 Jan 2017 Jan 2018 Jan 2019 Jan 2020

ТИП ОБЗОРА ▼ ТИП ПОКУПКИ ▼ ЯЗЫК ▼ ПРОМЕЖУТОК ▼

- ☒ Все (707,829)
- ☐ Положительные (679,163)
- ☐ Отрицательные (28,666)

[Исключать обзоры не по теме](#) ✕

Ассоциативный массив

```
{  
  "appid": 292030,  
  "positive": 632627,  
  "negative": 25245,  
  
  "name" : "The Witcher 3: Wild Hunt",  
  "developer" : "CD PROJEKT RED",  
  "publisher" : "CD PROJEKT RED",  
  "genre" : "RPG",  
  "release_date" : "2015/05/18",  
  
  "tags" : {  
    "Open World" : 11677,  
    "RPG" : 10024,  
    "Story Rich" : 9219,  
    "Atmospheric" : 6478,  
    "Mature" : 6234,  
    "Fantasy" : 6057  
  }  
}
```



BSON (Binary JSON)

— это формат двоичной сериализации, используемый MongoDB для хранения и передачи данных.



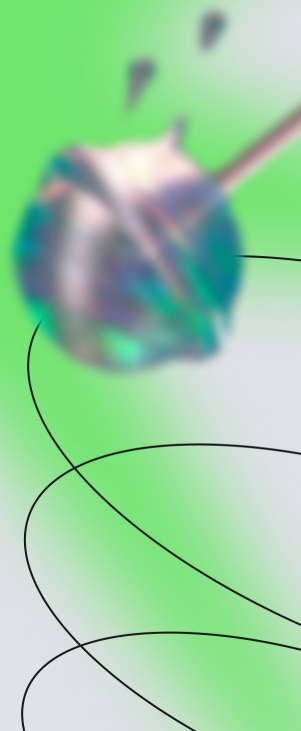


Преимущества Mongo DB

- 💡 имеет гибкую схему
- 💡 ориентирована на программистов
- 💡 имеет клиентские библиотеки (драйверы)
- 💡 возможность развертывания различными способами



Подключение к MongoDB





Выполнение операций CRUD в MongoDB





CRUD операции в Mongo DB



Create

```
db.mycollection.insert_one({"name": "John", "age": 30})
```



Read

```
cursor = db.mycollection.find({"name": "John"})  
for document in cursor:  
    print(document)
```



Update

```
db.mycollection.update_one({"name": "John"}, {"$set": {"age": 40}})
```



Delete

```
db.mycollection.delete_one({"name": "John"})
```



Операторы Mongo DB

- 💡 **\$eq** (равно)
- 💡 **\$ne** (не равно)
- 💡 **\$gt** (больше чем)
- 💡 **\$lt** (меньше чем)
- 💡 **\$gte** (больше или равно)
- 💡 **\$lte** (меньше или равно)
- 💡 **\$in** определяет массив значений, одно из которых должно иметь поле документа
- 💡 **\$nin** определяет массив значений, которые не должно иметь поле документа



Основы работы в ClickHouse





Table engines в Clickhouse



MergeTree Family



Log Family



Integrations



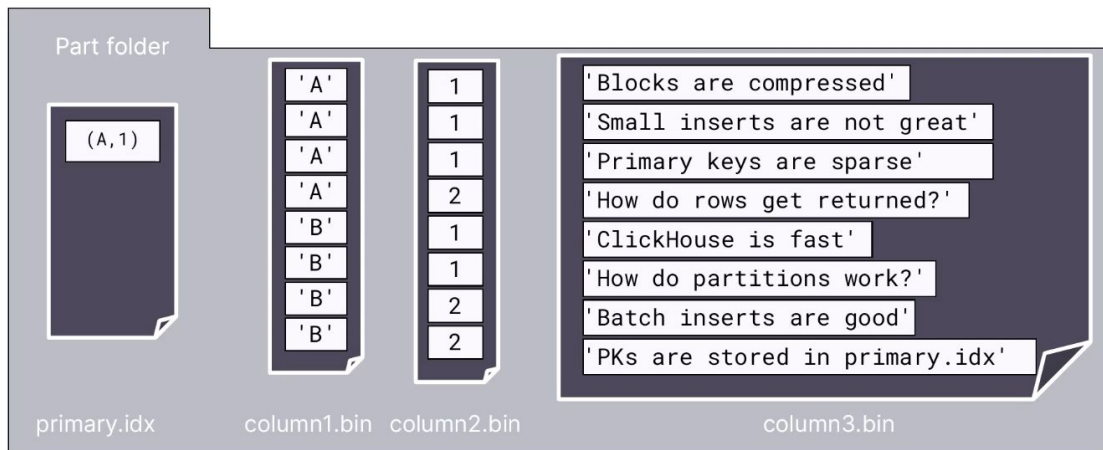
Special

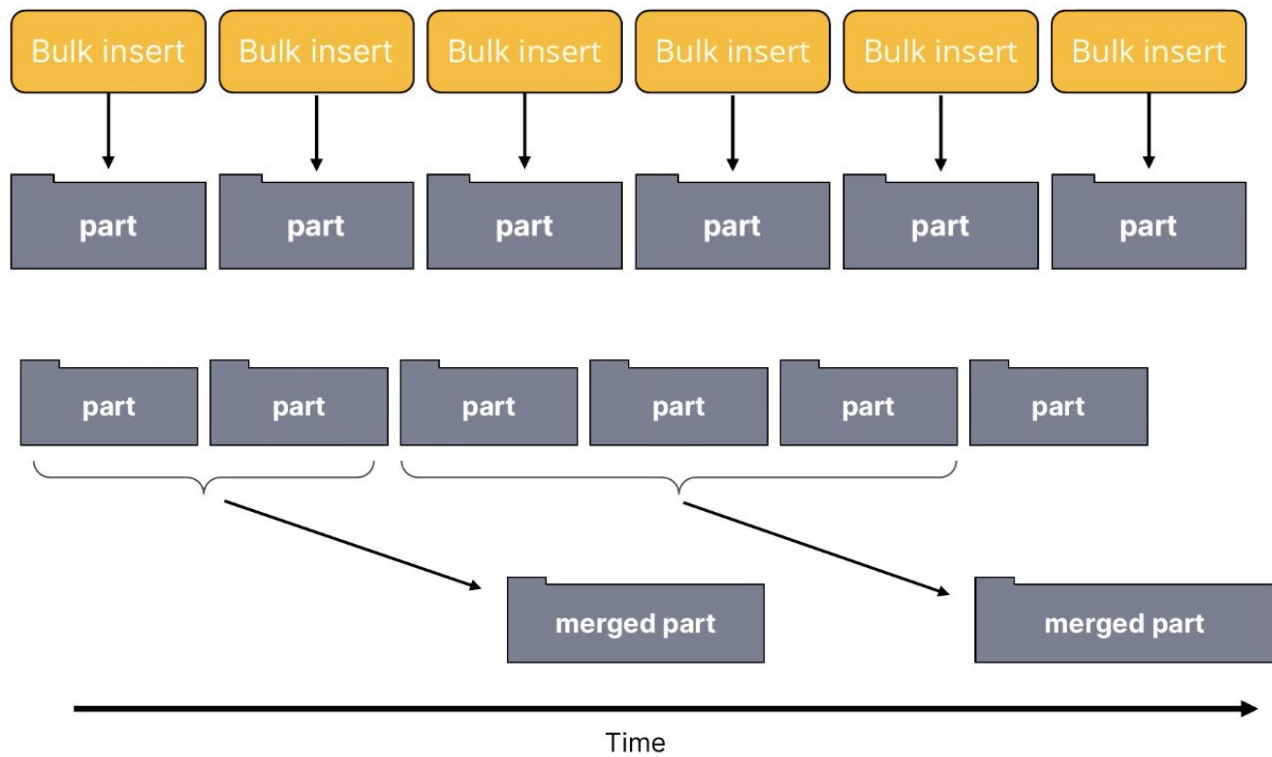
The **ORDER BY** tuple
becomes the **PRIMARY KEY**

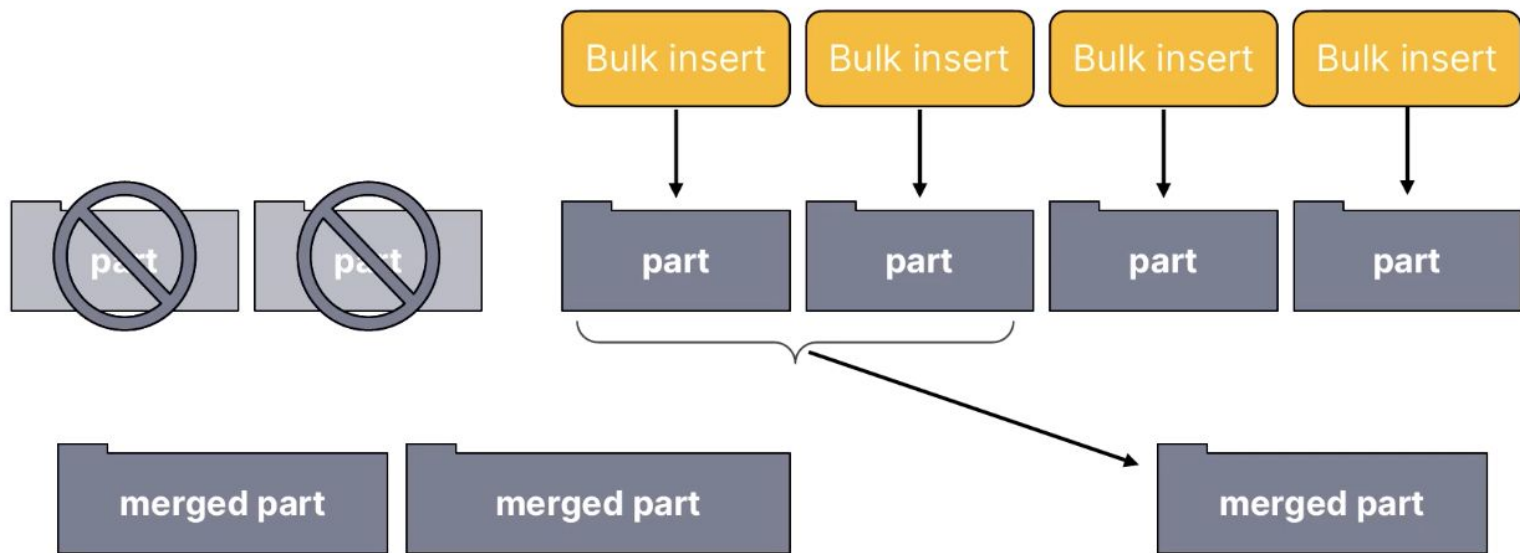


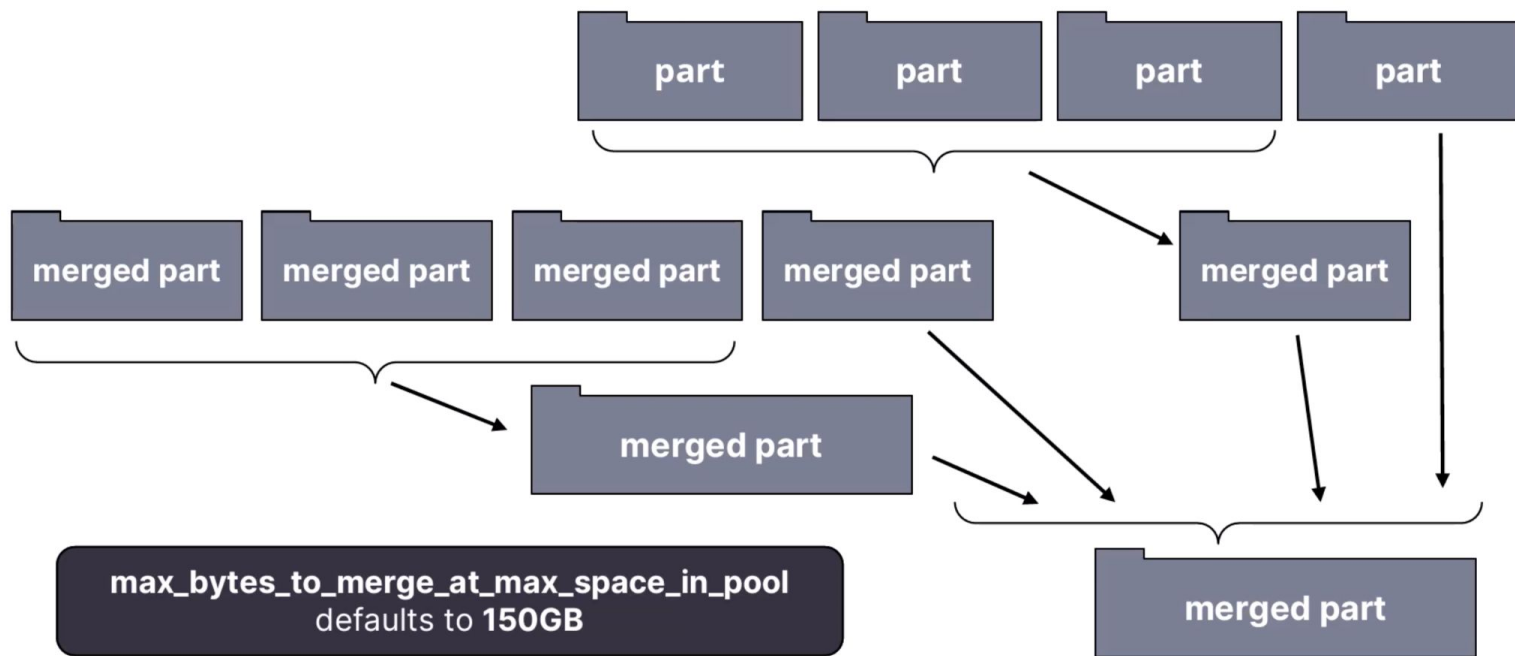
```
CREATE TABLE my_table
(
    column1    FixedString(1),
    column2    UInt32,
    column3    String
)
ENGINE = MergeTree()
ORDER BY (column1, column2)
```

```
INSERT INTO my_table (column1, column2, column3) VALUES
('B', 1, 'ClickHouse is fast'),
('A', 1, 'Blocks are compressed'),
('B', 2, 'Batch inserts are good'),
('A', 1, 'Small inserts are not great'),
('B', 1, 'How do partitions work?'),
('B', 2, 'PKs are stored in primary.idx'),
('A', 2, 'How do rows get returned?'),
('A', 1, 'Primary keys are sparse')
```







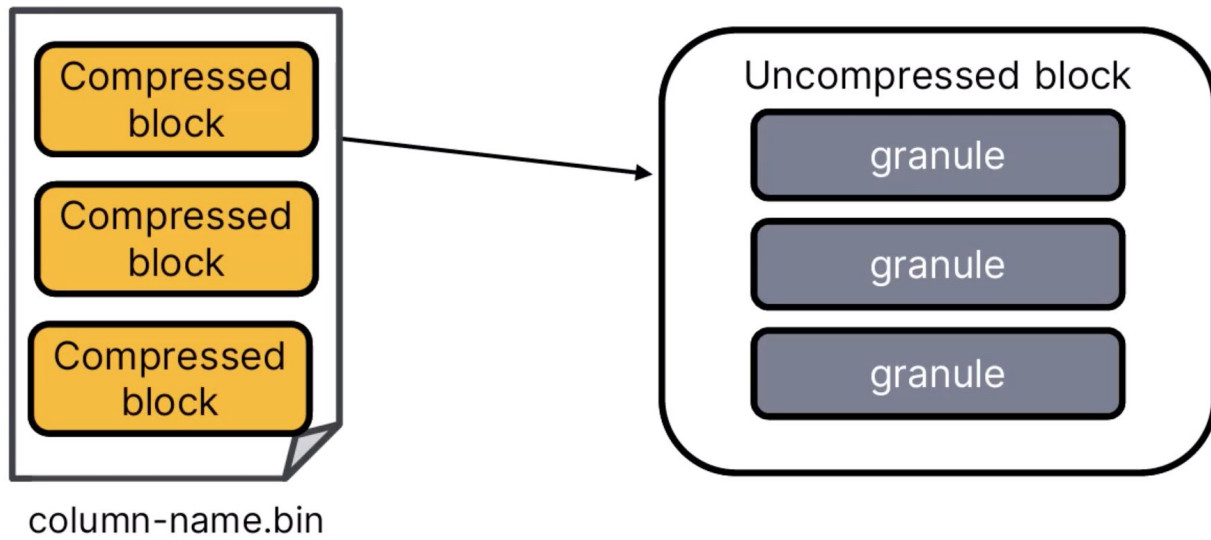


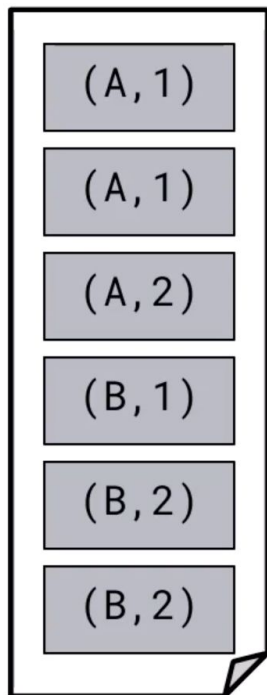
column-name.bin

Compressed block

Compressed block

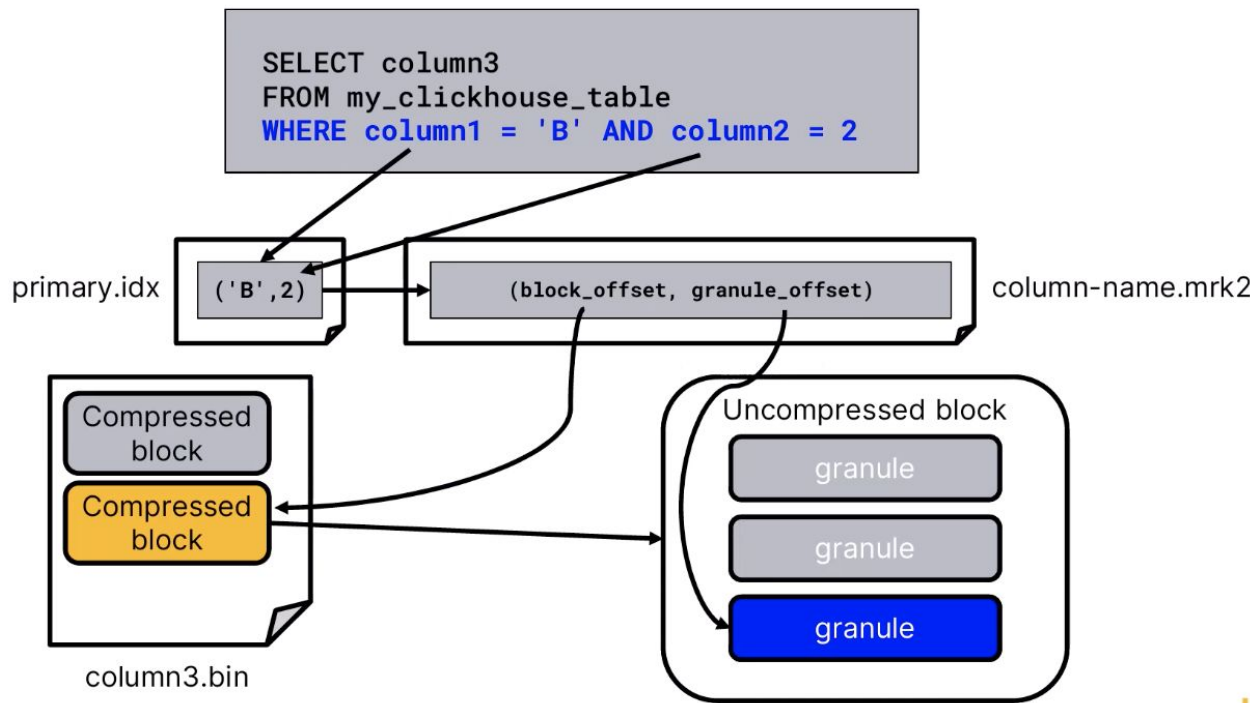
Compressed block



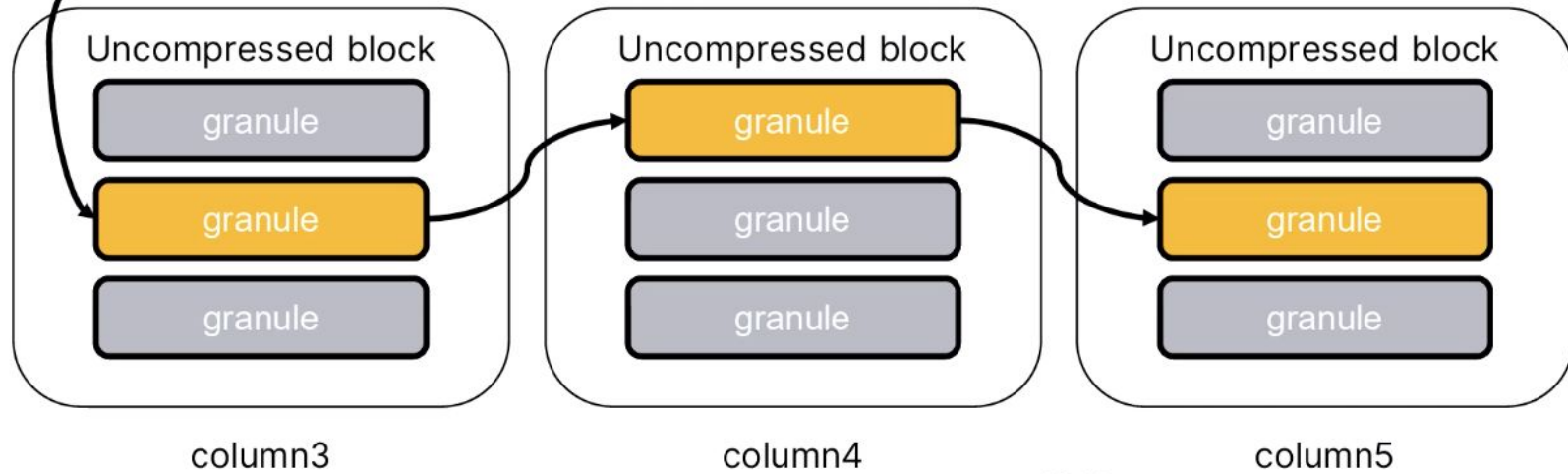


(A, 1)
(A, 1)
(A, 2)
(B, 1)
(B, 2)
(B, 2)

You can have *millions* of rows
but only *hundreds* of keys



```
SELECT * FROM my_clickhouse_table  
WHERE column1 = 'B'  
AND column2 = 2
```





Особенности применения Clickhouse

- 💡 работа с огромными объемами данных (измеряемыми в терабайтах), которые постоянно записываются и считываются;
- 💡 использование таблицы с огромным количеством столбцов, но значения столбцов достаточно короткие;
- 💡 данные хорошо структурированы, но еще не агрегированы;
- 💡 данные вставляются большими партиями в тысячи-миллионы строк;
- 💡 подавляющее большинство операций — это чтение с агрегацией;
- 💡 при чтении обрабатывается большое количество строк, но довольно малое количество столбцов;
- 💡 данные позже не потребуют изменения;
- 💡 нет потребности извлекать отдельные строки;