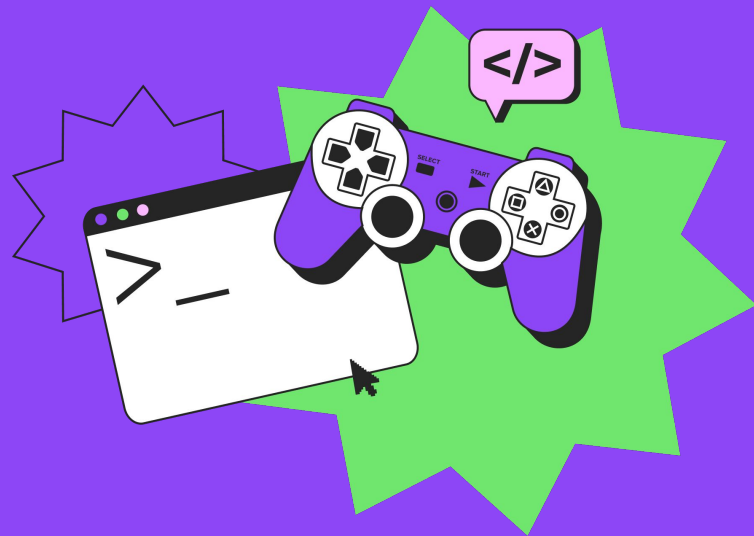


# Системы управления базами данных MongoDB и ClickHouse

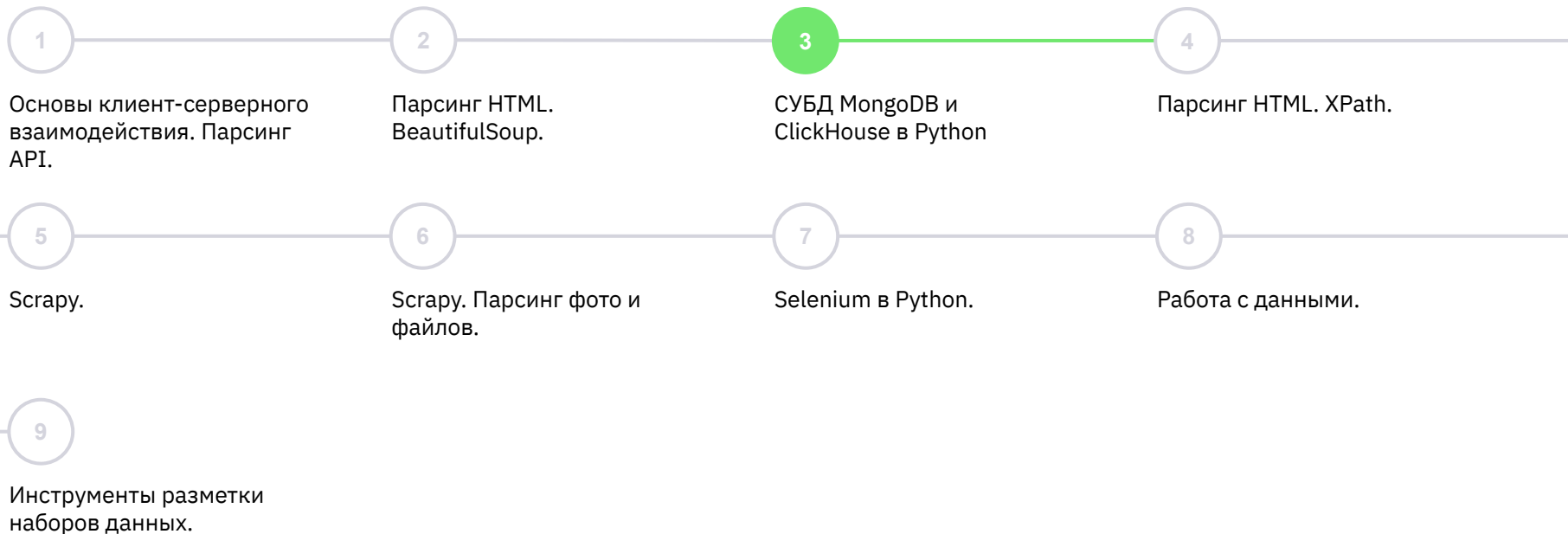
Семинар 3

Сбор и разметка данных








## Сбор и разметка данных





## Что будет на уроке сегодня

-  Создание скриптов Python для взаимодействия с базами данных MongoDB и Clickhouse, включая вставку, обновление и запрос данных.
-  Загрузка данные из файлов JSON и других источников в базы данных MongoDB и Clickhouse.
-  Различные типы запросов и их использование в MongoDB и Clickhouse.





# Викторина



## В чем разница между MongoDB и традиционными реляционными базами данных?

1. MongoDB предназначена для структурированных данных, тогда как традиционные базы данных предназначены для неструктурированных данных
2. MongoDB поддерживает только SQL, в то время как традиционные базы данных поддерживают NoSQL
3. MongoDB является документо-ориентированной базой данных, в то время как традиционные базы данных используют таблицы и строки
4. MongoDB не имеет возможности индексирования, в то время как традиционные базы данных имеют надежные функции индексирования



## В чем разница между MongoDB и традиционными реляционными базами данных?

1. MongoDB предназначена для структурированных данных, тогда как традиционные базы данных предназначены для неструктурированных данных
2. MongoDB поддерживает только SQL, в то время как традиционные базы данных поддерживают NoSQL
3. MongoDB является документо-ориентированной базой данных, в то время как традиционные базы данных используют таблицы и строки
4. MongoDB не имеет возможности индексирования, в то время как традиционные базы данных имеют надежные функции индексирования



## Как вставить данные в коллекцию MongoDB с помощью Python?

1. Использовать оператор CREATE
2. Использовать оператор INSERT
3. Использовать оператор UPDATE
4. Использовать оператор DELETE



## Как вставить данные в коллекцию MongoDB с помощью Python?

1. Использовать оператор CREATE
2. **Использовать оператор INSERT**
3. Использовать оператор UPDATE
4. Использовать оператор DELETE





## Что такое первичный ключ в ClickHouse?

1. Столбец или набор столбцов, который однозначно идентифицирует каждую строку в таблице.
2. Столбец, в котором хранятся числовые значения
3. Столбец, в котором хранятся строки
4. Столбец, в котором хранятся даты и время



## Что такое первичный ключ в ClickHouse?

1. Столбец или набор столбцов, который однозначно идентифицирует каждую строку в таблице.
2. Столбец, в котором хранятся числовые значения
3. Столбец, в котором хранятся строки
4. Столбец, в котором хранятся даты и время



## Как создать таблицу в ClickHouse?

1. С помощью оператора `CREATE TABLE`
2. Использовать оператор `INSERT INTO`
3. Использовать оператор `SELECT`
4. Использовать оператор `UPDATE`



## Как создать таблицу в ClickHouse?

1. С помощью оператора `CREATE TABLE`
2. Использовать оператор `INSERT INTO`
3. Использовать оператор `SELECT`
4. Использовать оператор `UPDATE`



## Как запросить данные из коллекции MongoDB с помощью Python?

1. Использовать оператор SELECT
2. Использовать метод FIND
3. Использовать метод QUERY
4. Использовать метод FILTER



## Как запросить данные из коллекции MongoDB с помощью Python?

1. Использовать оператор SELECT
2. Использовать метод FIND
3. Использовать метод QUERY
4. Использовать метод FILTER



## Какова цель сжатия данных в ClickHouse?

1. Для шифрования данных в целях безопасности
2. Для оптимизации доступа к данным и производительности запросов
3. Для хранения данных в определенном порядке
4. Группировать связанные данные вместе



## Какова цель сжатия данных в ClickHouse?

1. Для шифрования данных в целях безопасности
2. Для оптимизации доступа к данным и производительности запросов
3. Для хранения данных в определенном порядке
4. Группировать связанные данные вместе





## Как запросить данные из таблицы ClickHouse с помощью SQL?

1. Использовать оператор FIND
2. Использовать оператор SELECT
3. Использовать оператор QUERY
4. Использовать оператор FILTER



## Как запросить данные из таблицы ClickHouse с помощью SQL?

1. Использовать оператор FIND
2. Использовать оператор SELECT
3. Использовать оператор QUERY
4. Использовать оператор FILTER



Вопросы?

Вопросы?



Вопросы?





# Практика



## Задание 1

- Установите пакет PyMongo и импортируйте MongoClient и json.
- Установите Compass MongoDB
- Подключитесь к серверу MongoDB по адресу 'mongodb://localhost:27017/'.
- Создайте базу данных 'town\_cary' и коллекцию 'crashes'.
- Выполните чтение файла JSON 'crash-data.json'.
- Напишите функцию chunk\_data, которая принимает два аргумента: список данных и размер фрагмента. Функция должна разделить данные на более мелкие фрагменты указанного размера и вернуть генератор.
- Разделите данные JSON на фрагменты по 5000 записей в каждом.
- Переберите все фрагменты и вставьте каждый фрагмент в коллекцию MongoDB с помощью функции insert\_many().
- Выведите финальное сообщение, указывающее на то, что данные были успешно вставлены.



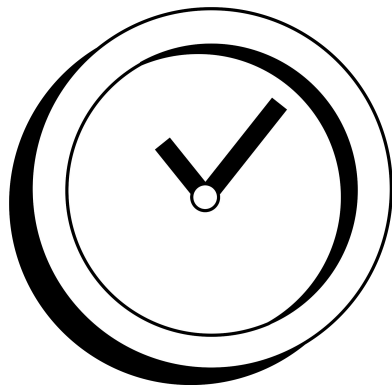
25 минут



## Задание 1

- Установите пакет PyMongo и импортируйте MongoClient и json.
- Установите Compass MongoDB
- Подключитесь к серверу MongoDB по адресу 'mongodb://localhost:27017/'.
- Создайте базу данных 'town\_cary' и коллекцию 'crashes'.
- Выполните чтение файла JSON 'crash-data.json'.
- Напишите функцию chunk\_data, которая принимает два аргумента: список данных и размер фрагмента. Функция должна разделить данные на более мелкие фрагменты указанного размера и вернуть генератор.
- Разделите данные JSON на фрагменты по 5000 записей в каждом.
- Переберите все фрагменты и вставьте каждый фрагмент в коллекцию MongoDB с помощью функции insert\_many().
- Выведите финальное сообщение, указывающее на то, что данные были успешно вставлены.

<<10:00-





## Задание 2

- Импортируйте MongoClient и json.
- Создайте экземпляр клиента для подключения к MongoDB.
- Подключитесь к базе данных 'town\_cary' и коллекции 'crashes'.
- Найдите первый документ в коллекции и распечатайте его в формате JSON.
- Используйте функцию count\_documents(), чтобы получить общее количество документов в коллекции.
- Отфильтруйте документы по критерию "properties.fatalities", равному "Yes", и подсчитайте количество совпадающих документов.
- Используйте проекцию для отображения только полей "properties.lightcond" и "properties.weather" для документов с "properties.fatalities" равным "Yes".



20 минут



## Задание 2

- Используйте операторы `$lt` и `$gte` для подсчета количества документов с "properties.month" меньше 6 и больше или равно 6, соответственно.
- Используйте оператор `$regex` для подсчета количества документов, содержащих слово "rain" в поле "properties.weather", игнорируя регистр.
- Используйте оператор `$in` для подсчета количества документов, в которых "properties.rdclass" является либо "US ROUTE", либо "STATE SECONDARY ROUTE".
- Используйте оператор `$all` для подсчета количества документов, в которых "properties.rdconfigur" содержит как "TWO-WAY", так и "DIVIDED".
- Используйте оператор `$ne` для подсчета количества документов, у которых "properties.rdcondition" не равно "DRY".

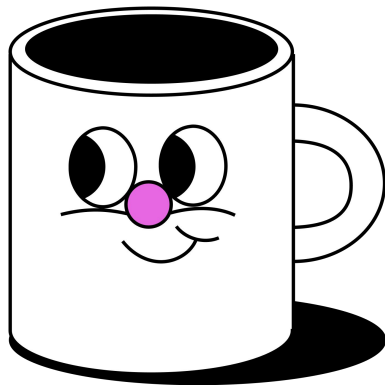


20 минут





## Перерыв



<<5:00->>



## Задание 3

- Установите базу данных ClickHouse на локальной машине и библиотеку clickhouse-driver Python.

В терминале выполните следующие команды:

```
curl https://clickhouse.com/ | sh
```

```
./clickhouse server
```

Откройте новое окно терминала и используйте clickhouse-client для подключения к вашему сервису:

```
./clickhouse client
```

- Подключитесь к серверу ClickHouse с помощью библиотеки clickhouse\_driver.
- Создайте новую базу данных с именем town\_cary, если она еще не существует.



40 минут



## Задание 3

- Разработайте схему для таблицы `crashes` со столбцами, полученными у преподавателя
- Используйте механизм MergeTree для таблицы `crashes` и упорядочьте таблицу по столбцу `id`.
- Прочитайте данные JSON из файла `crash-data.json` и извлеките ключ `features`.
- Пройдитесь по извлеченным данным и вставьте каждую запись об аварии в таблицу `crashes`. Обязательно обработайте значения `None` в данных, заменив их пустыми строками или другими подходящими значениями по умолчанию.
- После вставки данных проверьте успешность вставки, выполнив оператор `SELECT` и выведя первую вставленную запись.



40 минут



## Задание 3 - Hints

- Используйте класс `clickhouse_driver.Client` для подключения к серверу ClickHouse и выполнения запросов.
- Используйте операторы `CREATE DATABASE IF NOT EXISTS` и `CREATE TABLE IF NOT EXISTS` для создания базы данных и таблицы.
- Используйте библиотеку `json` для загрузки данных JSON и извлечения необходимой информации.
- Используйте условные выражения с оператором `or` для обработки значений `None` в данных при вставке записей, например `properties['rdfeature'] or ""`



40 минут



## Задание 4

- Используйте для работы Jupyter Notebook
- Использовать библиотеку `clickhouse_driver` для выполнения базовых запросов к таблице `'town_cary.crashes'`.
- Отфильтровать записи на основе таких критериев, как равенство, неравенство и диапазон.
- Отсортировать записи на основе одного или нескольких полей.
- Агрегировать записи с помощью таких функций, как `count`, `sum` и `avg`.
- Отображать результаты запроса в `pandas DataFrames` для лучшей наглядности.



40 минут



## Домашнее задание

1. Установите MongoDB на локальной машине, а также зарегистрируйтесь в онлайн-сервисе.
  2. Загрузите данные который вы получили на предыдущем уроке путем скрейпинга сайта с помощью BeautifulSoup в MongoDB и создайте базу данных и коллекции для их хранения.
  3. Поэкспериментируйте с различными методами запросов.
  4. Зарегистрируйтесь в ClickHouse.
  5. Загрузите данные в ClickHouse и создайте таблицу для их хранения.
- Чтобы загрузить данные из файла JSON в базу данных ClickHouse, можно использовать формат JSONEachRow и оператор INSERT. Вот пример того, как это сделать с помощью клиента ClickHouse:
1. Подключитесь к своей базе данных ClickHouse с помощью клиента.



## Домашнее задание

1. Создайте таблицу в вашей базе данных, соответствующую структуре ваших данных JSON.

```
CREATE TABLE my_table (  
    id UInt64,  
    name String,  
    age UInt8,  
    address String  
) ENGINE = MergeTree ORDER BY id;
```

1. Загрузите данные из файла JSON в таблицу с помощью оператора INSERT.

```
INSERT INTO my_table FORMAT JSONEachRow < my_data.json;
```

my\_data.json - это имя вашего файла JSON. Формат JSONEachRow указывает ClickHouse, что каждая строка в файле должна рассматриваться как отдельный объект JSON, который будет вставлен в таблицу.



## Домашнее задание

1. Убедитесь, что данные загружены в таблицу, выполнив любой запрос.

```
SELECT * FROM my_table;
```

Обратите внимание, что при использовании формата `JSONEachRow` необходимо убедиться, что каждая строка в вашем JSON-файле представляет собой один JSON-объект с той же структурой, что и таблица, в которую вы его загружаете. Если ваши данные JSON имеют другой формат, вам может потребоваться их предварительная обработка перед загрузкой в ClickHouse. Кроме того, вам может понадобиться настроить схему таблицы и типы данных в соответствии со структурой ваших данных.

6. Поэкспериментируйте с различными запросами к базе данных.

Выполните документирование и объяснение выполненных шагов.





Вопросы?

Вопросы?



Вопросы?





Спасибо за внимание!