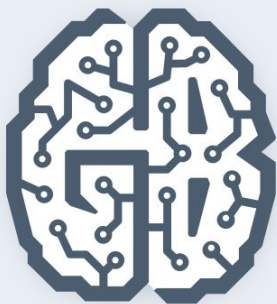


Методы сбора и обработки данных при помощи Python

Урок 5



СУБД MongoDB

Краткий обзор технологий для понимания сбора и обработки данных

План урока

1.
 - 1) SQL и NoSQL
 - 2) Что такое MongoDB и почему именно она
 - 3) Структура данных в MongoDB
 - 4) Установка MongoDB на примере Win 7
 - 5) Работа с MongoDB из консоли
 - 6) Работа с MongoDB в Python



SQL и NoSQL



SQL

- Atomicity – атомарность;
- Consistency – согласованность;
- Isolation – изолированность;
- Durability – устойчивость.



NoSQL

- Ключ-значение (Redis, Berkeley DB).
- Документоориентированные (MongoDB, CouchDB).
- Графовые (Giraph, Neo4j).
- BigTable (HBase, Cassandra).



NoSQL

```
{  _id: ObjectID('4bd9e8e17cefd644108961bb'),  
  title: 'Adventures in Databases',
```

← **Поле `_id` – первичный ключ**

```
    url: 'http://example.com/databases.txt',  
    author: 'msmith',  
    vote_count: 20,
```

```
    tags: ['databases', 'mongodb', 'indexing'],
```

① **Теги хранятся в виде массива строк**

```
    image: {  
      url: 'http://example.com/db.jpg',  
      caption: '',  
      type: 'jpg',  
      size: 75381,  
      data: "Binary"
```

← ② **Атрибут указывает на другой документ**

```
  },
```

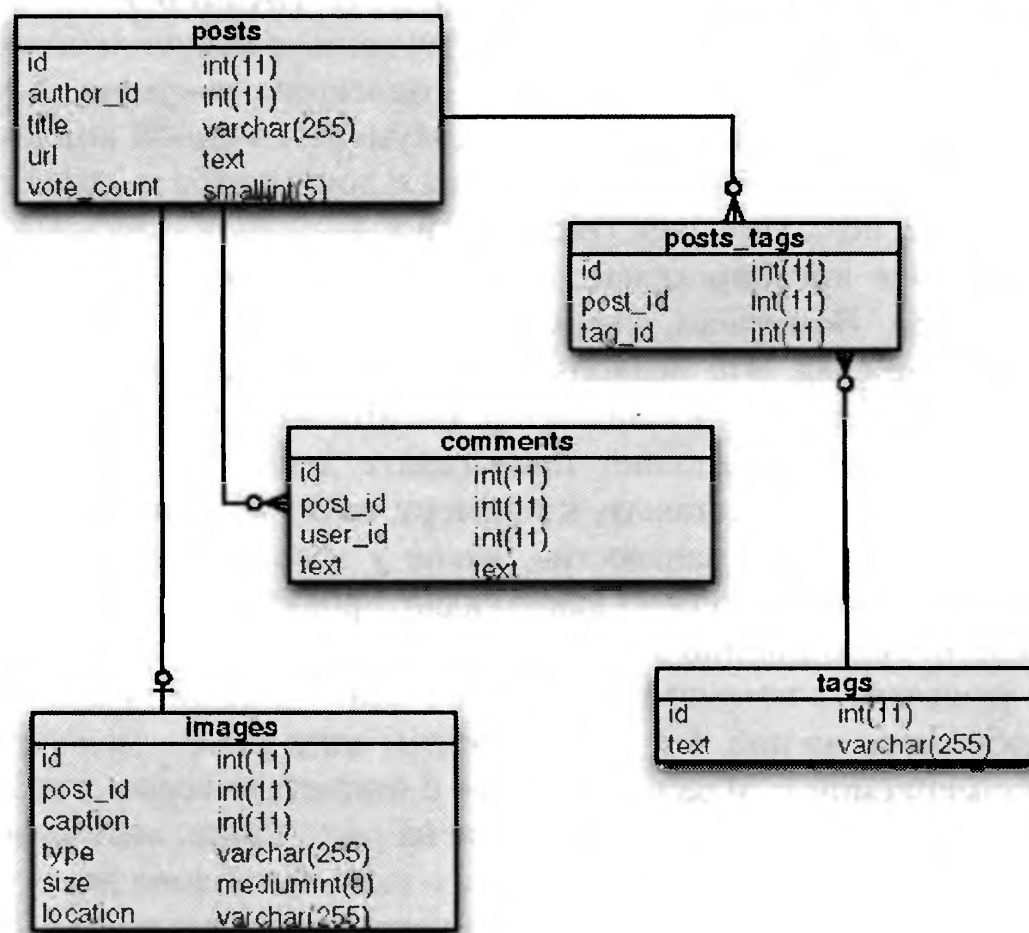
```
  comments: [  
    { user: 'bjones',  
      text: 'Interesting article!'  
    },  
    { user: 'blogger',  
      text: 'Another related article is at http://example.com/db/db.txt'  
    }  
  ]
```

← ③ **Комментарии хранятся в виде массива объектов, представляющих один комментарий**

```
}
```



SQL



Примеры запросов

Найти все статьи из таблицы posts с тегом politics, за которые проголосовало более 10 посетителей

SQL:

```
SELECT * FROM posts
  INNER JOIN posts_tags ON posts.id = posts_tags.post_id
  INNER JOIN tags ON posts_tags.tag_id == tags.id
 WHERE tags.text = 'politics' AND posts.vote_count > 10;
```

Mongo:

```
db.posts.find({'tags': 'politics', 'vote_count': {'$gt': 10}});
```



Что такое MongoDB и почему именно она?

- Скорость разработки.
- Нет необходимости в поддержке схемы и в коде, и в БД.
- Легкая масштабируемость.
- Гибкость при смене задачи.
- Удобство работы с денормализованными данными.



Что такое MongoDB и почему именно она?

- Данные быстро меняются (дополнительные данные из API, динамический контент в HTML-страницах).
- Меняя схему, надо менять и приложение, и БД.
- БД нужна лишь до тех пор, пока нужны данные.
- Данные постоянно обновляются.
- Нормализация не нужна.
- Задача не меняется.
- Одно приложение.



Структура данных MongoDB

```
{
  _id: DOC_ID,
  sid: 1,
  sources: [{
    name: "Source #1 Name",
    description: "Source #1 Description",
    uri: "https://source1.com"
  },
  {
    name: "Source #3 Name",
    description: "Source #3 Description",
    uri: "https://source1.com",
  }],
  children: [{
    sid: "2",
    level: 3,
    nest_level: 1
  }],
  group_name: "Group1",
  owner: {
    sid: 1,
    fullname: "Ivanov Ivan",
  }
}
```



Структура данных MongoDB

```
{  
  _id: DOC_ID,  
  sid: 1,  
  group_name: "Group1",  
  owner_id: 1  
}
```



Структура данных MongoDB

```
{
  _id: SOURCE_ID,
  doc_id: DOC_ID,
  name: "Source #1 Name",
  description: "Source #1 Description",
  uri: "https://source1.com"
},
{
  _id: SOURCE_ID,
  doc_id: DOC_ID,
  name: "Source #3 Name",
  description: "Source #3 Description",
  uri: "https://source1.com",
}
```



Структура данных MongoDB

```
{  
  _id: CHILD_ID,  
  doc_id: DOC_ID,  
  sid: "2",  
  level: 3,  
  nest_level: 1  
}
```



Структура данных MongoDB

```
{  
  _id: OWNER_ID,  
  doc_id: DOC_ID,  
  sid: 1,  
  fullname: "Ivanov Ivan",  
}
```



Домашнее задание

- 1) Развернуть у себя на компьютере/виртуальной машине/хостинге MongoDB и реализовать функцию, записывающую собранные объявления с avito.ru в созданную БД (xpath/BS для парсинга на выбор)
 - 2) Написать функцию, которая производит поиск и выводит на экран объявления с ценой меньше введенной суммы
- *Написать функцию, которая будет добавлять в вашу базу данных только новые объявления



Ваши вопросы?

