

Лабораторная работа 1. Системы управления базами данных MongoDB и SQLite в Python

Цель лабораторной работы: освоение работы с системами управления базами данных MongoDB и SQLite в языке программирования Python для сбора, консолидации и аналитической обработки финансовой и экономической информации.

Общие требования к заданию:

1. Использовать Python 3 и библиотеки для работы с MongoDB и SQLite.
2. Собрать данные из одного или нескольких предложенных источников данных.
3. Сохранить данные в MongoDB и SQLite.
4. Провести анализ данных с использованием SQL-запросов и методов MongoDB.
5. Представить результаты анализа в виде графиков или таблиц.
6. Составить отчет, содержащий код, результаты и анализ.

Пример

Задание:

Сбор и анализ данных о дневных курсах валют с сайта Центрального банка РФ

Пошаговый алгоритм решения в SQLite

1. Установка необходимых библиотек:
`pip install requests beautifulsoup4 pymongo pandas matplotlib`
2. Импортирование библиотек:
`import requests
from bs4 import BeautifulSoup
from pymongo import MongoClient
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt`
3. Получение HTML-кода страницы:
`url = 'https://www.cbr.ru/currency_base/daily/'
response = requests.get(url)
page_content = response.content`
4. Парсинг HTML с помощью BeautifulSoup:
`soup = BeautifulSoup(page_content, 'html.parser')
table = soup.find('table', {'class': 'data'})
rows = table.find_all('tr')`
5. Извлечение данных и создание DataFrame:
`data = []
for row in rows[1:]: # Пропускаем заголовок
 cols = row.find_all('td')
 cols = [ele.text.strip() for ele in cols]
 data.append(cols)
df = pd.DataFrame(data, columns=['Num', 'CharCode', 'Unit', 'Currency', 'Value'])
df['Value'] = df['Value'].str.replace(',', '.').astype(float)`

6. Сохранение данных в SQLite:

```
conn = sqlite3.connect('financial_data.db')
df.to_sql('currency_rates', conn, if_exists='replace', index=False)
```
7. Анализ данных с использованием SQLite:

```
query = "SELECT * FROM currency_rates WHERE CharCode = 'USD'"
df_sqlite = pd.read_sql(query, conn)
print(df_sqlite)
```
8. Интерпретация результатов и подготовка отчета.
В отчете должны быть:
 - Описание задачи и источника данных.
 - Код парсинга данных.
 - Код сохранения данных в SQLite.
 - SQL-запросы для анализа данных.

Пошаговый алгоритм решения в MongoDB

1. Установка необходимых библиотек.

```
pip install requests beautifulsoup4 pymongo pandas matplotlib
```
2. Импортирование библиотек.

```
import requests
from bs4 import BeautifulSoup
from pymongo import MongoClient
import pandas as pd
import matplotlib.pyplot as plt
```
3. Получение HTML-кода страницы.

```
url = 'https://www.cbr.ru/currency_base/daily/'
response = requests.get(url)
page_content = response.content
```
4. Парсинг HTML с помощью BeautifulSoup.

```
soup = BeautifulSoup(page_content, 'html.parser')
table = soup.find('table', {'class': 'data'})
rows = table.find_all('tr')
```
5. Извлечение данных и создание DataFrame.

```
data = []
for row in rows[1:]: # Пропускаем заголовок
    cols = row.find_all('td')
    cols = [ele.text.strip() for ele in cols]
    data.append(cols)
df = pd.DataFrame(data, columns=['Num', 'CharCode', 'Unit', 'Currency', 'Value'])
df['Value'] = df['Value'].str.replace(',', '.').astype(float)
```
6. Подключение к MongoDB.

```
try:
```

```

# Подключение к MongoDB
client = MongoClient(mongo_uri)
# Проверка подключения
client.admin.command('ping')
print("Подключение к MongoDB установлено успешно!")
# Выбор базы данных
db = client['labs']
# Выбор коллекции
labs_collection = db['lab11']
except Exception as e:
    print(f"Ошибка подключения: {e}")

```

7. Сохранение данных в MongoDB.

```

db = client['financial_data']
collection = db['currency_rates']
collection.insert_many(df.to_dict('records'))

```

8. Анализ данных с использованием MongoDB.

- Пример запроса для получения всех данных о курсе USD.

```

usd_data = collection.find({'CharCode': 'USD'})
for item in usd_data:
    print(item)

```

- Пример запроса для получения всех данных с сортировкой по курсу.

```

sorted_data = collection.find().sort('Value', -1)
for item in sorted_data:
    print(item)

```

9. Визуализация данных.

```

df.plot(kind='bar', x='CharCode', y='Value', legend=False)
plt.title('Курсы валют к рублю')
plt.xlabel('Валюта')
plt.ylabel('Курс')
plt.show()

```

10. Интерпретация результатов и подготовка отчета.

В отчете должны быть:

- Описание задачи и источника данных.
- Код парсинга данных.
- Код сохранения данных в MongoDB.
- Примеры MongoDB-запросов для анализа данных.
- Графики и таблицы с анализом.

Варианты заданий

1. **Парсинг данных о дневных курсах валют с сайта Центрального банка РФ и их хранение в MongoDB и SQLite**
Ссылка: [ЦБ РФ](#)
2. **Извлечение исторических данных о ценах на нефть с портала Bloomberg и их хранение в MongoDB и SQLite**
Ссылка: [Bloomberg](#)
3. **Сбор данных о ежедневных индексах NASDAQ с официального сайта NASDAQ и их хранение в MongoDB и SQLite**
Ссылка: [NASDAQ](#)
4. **Анализ изменений цен на золото на сайте Kitco и их хранение в MongoDB и SQLite**
Ссылка: [Kitco](#)
5. **Выгрузка и анализ данных о доходности государственных облигаций с сайта Investing.com и их хранение в MongoDB и SQLite**
Ссылка: [Investing.com](#)
6. **Парсинг данных о рыночных капитализациях криптовалют с CoinMarketCap и их хранение в MongoDB и SQLite**
Ссылка: [CoinMarketCap](#)
7. **Сбор новостей экономического блока с CNN и их хранение в MongoDB и SQLite**
Ссылка: [CNN](#)
8. **Анализ статистики торгов на Шанхайской бирже и их хранение в MongoDB и SQLite**
Ссылка: [Шанхайская биржа](#)
9. **Выгрузка данных о кредитных ставках различных банков и их хранение в MongoDB и SQLite**
Ссылка: [Bankrate](#)
10. **Парсинг аналитических статей о мировой экономике с сайта Business Insider и их хранение в MongoDB и SQLite**
Ссылка: [Business Insider](#)
11. **Извлечение данных о рейтингах стран по ВВП с сайта МВФ и их хранение в MongoDB и SQLite**
Ссылка: [МВФ](#)
12. **Анализ данных о бюджетных поступлениях различных стран с сайта OECD и их хранение в MongoDB и SQLite**
Ссылка: [OECD](#)
13. **Сбор и анализ данных о безработице в различных странах с сайта Statista и их хранение в MongoDB и SQLite**
Ссылка: [Statista](#)
14. **Изучение динамики изменения цен на первичном жилищном рынке с сайта Zillow и их хранение в MongoDB и SQLite**
Ссылка: [Zillow](#)
15. **Анализ фондового рынка с использованием данных Yahoo Finance и их хранение в MongoDB и SQLite**
Ссылка: [Yahoo Finance](#)
16. **Извлечение и анализ данных о продажах автомобилей с сайта Statista и их хранение в MongoDB и SQLite**

Ссылка: [Statista](#)

17. **Сбор информации о финансовых результатах компаний с Forbes и их хранение в MongoDB и SQLite**

Ссылка: [Forbes](#)

18. **Анализ тенденций в изменении стоимости аренды недвижимости с сайта RentCafe и их хранение в MongoDB и SQLite**

Ссылка: [RentCafe](#)

19. **Сбор данных для анализа экономического влияния спортивных событий с сайта ESPN и их хранение в MongoDB и SQLite**

Ссылка: [ESPN](#)

20. **Изучение влияния политических событий на фондовые рынки с сайта Reuters и их хранение в MongoDB и SQLite**

Ссылка: [Reuters](#)

21. **Сбор данных о туристическом потоке и его влиянии на экономику стран с сайта UNWTO и их хранение в MongoDB и SQLite**

Ссылка: [UNWTO](#)

22. **Анализ влияния экологических изменений на экономику регионов с сайта World Bank и их хранение в MongoDB и SQLite**

Ссылка: [World Bank](#)

23. **Парсинг данных о государственных закупках для экономического анализа с сайта Федеральной антимонопольной службы и их хранение в MongoDB и SQLite**

Ссылка: [ФАС](#)

24. **Изучение инвестиционных трендов в развивающихся рынках с сайта MSCI и их хранение в MongoDB и SQLite**

Ссылка: [MSCI](#)

25. **Анализ статистики потребления интернет-услуг и его влияние на экономику с сайта ITU и их хранение в MongoDB и SQLite**

Ссылка: [ITU](#)