

Лабораторная работа 1.

Системы управления базами данных MongoDB и SQLite в Python

Цель лабораторной работы: Изучение работы с системами управления базами данных MongoDB и SQLite на языке программирования Python для сбора и преобразования статистической информации.

Задание: Сбор и анализ данных о продаваемых машинах в США за 2023-2024год.

Выполнил: Петров Евгений С. – БД231м.

Пошаговый алгоритм решения в SQLite

1. Установка необходимых библиотек:

```
In [1]: 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
import numpy as np
```

3. Получение HTML-кода страницы:

```
url = 'https://greenway.icnet.ru/cars-sales-actual-usa.html#null'
response = requests.get(url)
page_content = response.content
```

4. Парсинг HTML с помощью BeautifulSoup:

```
soup = BeautifulSoup(page_content, 'html.parser')
table = soup.find('table', {'id': 'unique_id'})
rows = table.find_all('tr')
```

5. Извлечение данных и создание DataFrame:

```
data = []
for row in rows[1:]:
    cols = row.find_all('td')
    cols = [ele.text.strip() for ele in cols]
    data.append(cols)

# Создаем DataFrame
df = pd.DataFrame(data, columns=['Rank', 'Brand', 'Sales_2024', 'Sales_2023', 'Change_percent_2024',
                                'Brand_2', 'Cumulative_2024', 'Cumulative_2023', 'Cumulative_Change_percent'])

df.replace('-', np.nan, inplace=True)

df['Sales_2024'] = df['Sales_2024'].str.replace(',', '.').astype(float)
df['Sales_2023'] = df['Sales_2023'].str.replace(',', '.').astype(float)
df['Change_percent_2024'] = df['Change_percent_2024'].str.replace(',', '.').astype(float)
df['Cumulative_2024'] = df['Cumulative_2024'].str.replace(',', '.').astype(float)
df['Cumulative_2023'] = df['Cumulative_2023'].str.replace(',', '.').astype(float)
df['Cumulative_Change_percent'] = df['Cumulative_Change_percent'].str.replace(',', '.').astype(float)
```

```
print(df)
```

	Rank	Brand	Sales_2024	Sales_2023	Change_percent_2024	Brand_2 \
0	1	Toyota	198.0	194.0	2.4	Toyota
1	2	Ford	176.0	160.0	10.1	Ford
2	3	Chevrolet	151.0	152.0	-0.7	Chevrolet
3	4	Honda	140.0	112.0	25.0	Honda
4	5	Hyundai	87.0	71.0	21.2	Nissan
5	6	Kia	75.0	72.0	4.3	Hyundai
6	7	Nissan	71.0	73.0	-1.4	Kia
7	8	Subaru	63.0	56.0	11.8	Subaru
8	9	Tesla	53.0	58.0	-9.3	Tesla
9	10	Mazda	41.0	30.0	36.7	Mazda

	Cumulative_2024	Cumulative_2023	Cumulative_Change_percent
0	1567.0	1425.0	10.0
1	1380.0	1333.0	3.6
2	1176.0	1190.0	-1.2
3	951.0	855.0	11.2
4	636.0	624.0	1.9
5	593.0	570.0	4.1
6	525.0	537.0	-2.3
7	437.0	410.0	6.4
8	394.0	453.0	-13.0
9	283.0	245.0	16.0

6. Сохранение данных в SQLite:

```
conn = sqlite3.connect('financial_data.db')
df.to_sql('car_sales', conn, if_exists='replace', index=False)
```

7. Анализ данных с использованием SQLite:

Rank	Brand	Sales_2024	Sales_2023	Change_percent_2024	Brand_2
9	Tesla	53.0	58.0	-9.3	Tesla
	Cumulative_2024	394.0	453.0	-13.0	

```
Empty DataFrame
Columns: [Rank, Brand, Sales_2024, Sales_2023, Change_percent_2024, Brand_2, Cumulative_2024, Cumulative_2023, Cumulative_Change_percent]
Index: []
```

Rank	Brand	Sales_2024	Sales_2023	Change_percent_2024	Brand_2	
0	1	Toyota	198.0	194.0	2.4	Toyota
1	2	Ford	176.0	160.0	10.1	Ford
2	4	Honda	140.0	112.0	25.0	Honda
3	5	Hyundai	87.0	71.0	21.2	Nissan
4	6	Kia	75.0	72.0	4.3	Hyundai
5	8	Subaru	63.0	56.0	11.8	Subaru
6	10	Mazda	41.0	30.0	36.7	Mazda
		Cumulative_2024	Cumulative_2023	Cumulative_Change_percent		
0		1567.0	1425.0	10.0		
1		1380.0	1333.0	3.6		
2		951.0	855.0	11.2		
3		636.0	624.0	1.9		
4		593.0	570.0	4.1		
5		437.0	410.0	6.4		
6		283.0	245.0	16.0		

Rank	Brand	Sales_2024	Sales_2023	Change_percent_2024	Brand_2
0	1 Toyota	198.0	194.0	2.4	Toyota
1	2 Ford	176.0	160.0	10.1	Ford
2	3 Chevrolet	151.0	152.0	-0.7	Chevrolet
3	4 Honda	140.0	112.0	25.0	Honda
	Cumulative_2024	Cumulative_2023	Cumulative_Change_percent		
0	1567.0	1425.0	10.0		
1	1380.0	1333.0	3.6		
2	1176.0	1190.0	-1.2		

```
url = 'https://greenway.icnet.ru/cars-sales-actual-russia.html#null'
response = requests.get(url)
page_content = response.content
```

4. Парсинг HTML с помощью BeautifulSoup.

```
soup = BeautifulSoup(page_content, 'html.parser')
table = soup.find('table', {'id': 'unique_id'})
rows = table.find_all('tr')
```

5. Извлечение данных и создание DataFrame.

```
data = []
for row in rows[1:]:
    cols = row.find_all('td')
    cols = [ele.text.strip() for ele in cols]
    data.append(cols)

# Создаем DataFrame
df = pd.DataFrame(data, columns=['Rank', 'Brand', 'Sales_2024', 'Sales_2023', 'Change_percent_2024',
                                'Brand_2', 'Cumulative_2024', 'Cumulative_2023', 'Cumulative_Change_percent'])

df.replace('-', np.nan, inplace=True)

df['Sales_2024'] = df['Sales_2024'].str.replace(',', '.').astype(float)
df['Sales_2023'] = df['Sales_2023'].str.replace(',', '.').astype(float)
df['Change_percent_2024'] = df['Change_percent_2024'].str.replace(',', '.').astype(float)
df['Cumulative_2024'] = df['Cumulative_2024'].str.replace(',', '.').astype(float)
df['Cumulative_2023'] = df['Cumulative_2023'].str.replace(',', '.').astype(float)
df['Cumulative_Change_percent'] = df['Cumulative_Change_percent'].str.replace(',', '.').astype(float)
```

```
print(df)
```

	Rank	Brand	Sales_2024	Sales_2023	Change_percent_2024	Brand_2 \
0	1	Toyota	198.0	194.0	2.4	Toyota
1	2	Ford	176.0	160.0	10.1	Ford
2	3	Chevrolet	151.0	152.0	-0.7	Chevrolet
3	4	Honda	140.0	112.0	25.0	Honda
4	5	Hyundai	87.0	71.0	21.2	Nissan
5	6	Kia	75.0	72.0	4.3	Hyundai
6	7	Nissan	71.0	73.0	-1.4	Kia
7	8	Subaru	63.0	56.0	11.8	Subaru
8	9	Tesla	53.0	58.0	-9.3	Tesla
9	10	Mazda	41.0	30.0	36.7	Mazda

	Cumulative_2024	Cumulative_2023	Cumulative_Change_percent
0	1567.0	1425.0	10.0
1	1380.0	1333.0	3.6
2	1176.0	1190.0	-1.2
3	951.0	855.0	11.2
4	636.0	624.0	1.9
5	593.0	570.0	4.1
6	525.0	537.0	-2.3
7	437.0	410.0	6.4
8	394.0	453.0	-13.0
9	283.0	245.0	16.0

6. Подключение к MongoDB.

```
mongo_uri = "mongodb://mongouser:mongopasswd@localhost:27017"
```

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

Подключение к MongoDB установлено успешно!

7. Сохранение данных в MongoDB.

```
db = client['economic2023']
collection = db['usacarsales2023-2024']
collection.insert_many(df.to_dict('record'))
```

```
] InsertManyResult([ObjectId('66f02a8aadae505bd964db8a'), ObjectId('66f02a8aadae505bd964db8b'), ObjectId('66f02a8aadae505bd964db8c'), ObjectId('66f02a8aadae505bd964db8d'), ObjectId('66f02a8aadae505bd964db8e'), ObjectId('66f02a8aadae505bd964db8f'), ObjectId('66f02a8aadae505bd964db90'), ObjectId('66f02a8aadae505bd964db91'), ObjectId('66f02a8aadae505bd964db92'), ObjectId('66f02a8aadae505bd964db93')], acknowledged=True)
```

8. Анализ данных с использованием MongoDB.

```
all_brands = collection.distinct('Brand')
print(all_brands)
```

```
['Chevrolet', 'Ford', 'Honda', 'Hyundai', 'Kia', 'Mazda', 'Nissan', 'Subaru', 'Tesla', 'Toyota']
```

```
usd_data = collection.find({'Brand': 'Tesla'})
for item in usd_data:
    print(item)
```

```
{'_id': ObjectId('66f02a8aadae505bd964db92'), 'Rank': '9', 'Brand': 'Tesla', 'Sales_2024': 53.0, 'Sales_2023': 58.0,
'Change_percent_2024': -9.3, 'Brand_2': 'Tesla', 'Cumulative_2024': 394.0, 'Cumulative_2023': 453.0, 'Cumulative_Change_percent': -13.0}
```

```
sorted_sales_data = collection.find().sort('Sales_2023', -1)
for item in sorted_sales_data:
    print(item)
```

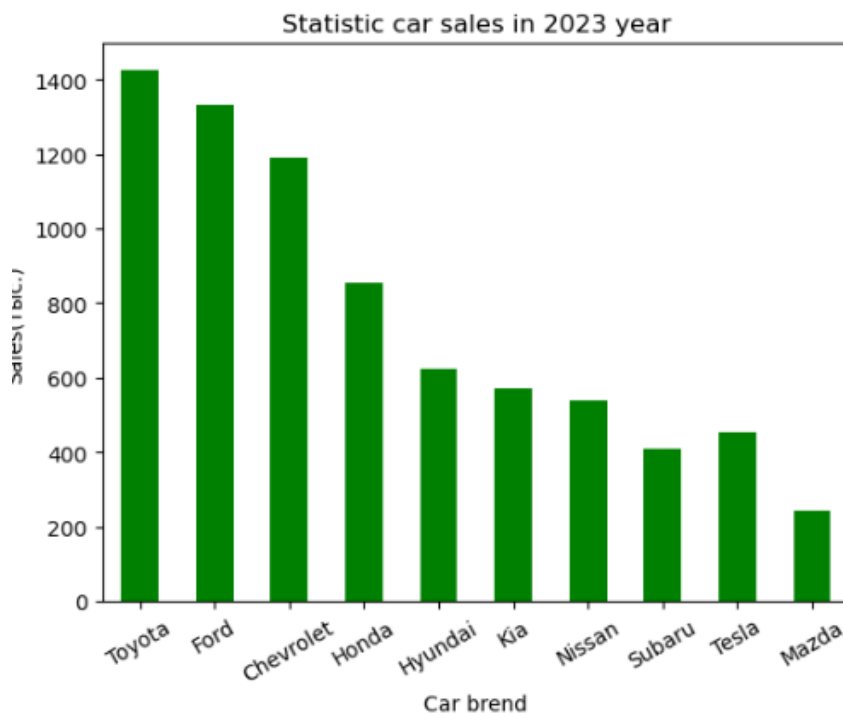
```
{'_id': ObjectId('66f02a8aadae505bd964db8a'), 'Rank': '1', 'Brand': 'Toyota', 'Sales_2024': 198.0, 'Sales_2023': 194.0, 'Change_percent_2024': 2.4, 'Brand_2': 'Toyota', 'Cumulative_2024': 1567.0, 'Cumulative_2023': 1425.0, 'Cumulative_Change_percent': 10.0}
{'_id': ObjectId('66f02a8aadae505bd964db8b'), 'Rank': '2', 'Brand': 'Ford', 'Sales_2024': 176.0, 'Sales_2023': 160.0, 'Change_percent_2024': 10.1, 'Brand_2': 'Ford', 'Cumulative_2024': 1380.0, 'Cumulative_2023': 1333.0, 'Cumulative_Change_percent': 3.6}
{'_id': ObjectId('66f02a8aadae505bd964db8c'), 'Rank': '3', 'Brand': 'Chevrolet', 'Sales_2024': 151.0, 'Sales_2023': 152.0, 'Change_percent_2024': -0.7, 'Brand_2': 'Chevrolet', 'Cumulative_2024': 1176.0, 'Cumulative_2023': 1190.0, 'Cumulative_Change_percent': -1.2}
{'_id': ObjectId('66f02a8aadae505bd964db8d'), 'Rank': '4', 'Brand': 'Honda', 'Sales_2024': 140.0, 'Sales_2023': 112.0, 'Change_percent_2024': 25.0, 'Brand_2': 'Honda', 'Cumulative_2024': 951.0, 'Cumulative_2023': 855.0, 'Cumulative_Change_percent': 11.2}
{'_id': ObjectId('66f02a8aadae505bd964db90'), 'Rank': '7', 'Brand': 'Nissan', 'Sales_2024': 71.0, 'Sales_2023': 73.0, 'Change_percent_2024': -1.4, 'Brand_2': 'Kia', 'Cumulative_2024': 525.0, 'Cumulative_2023': 537.0, 'Cumulative_Change_percent': -2.3}
{'_id': ObjectId('66f02a8aadae505bd964db8f'), 'Rank': '6', 'Brand': 'Kia', 'Sales_2024': 75.0, 'Sales_2023': 72.0, 'Change_percent_2024': 4.3, 'Brand_2': 'Hyundai', 'Cumulative_2024': 593.0, 'Cumulative_2023': 570.0, 'Cumulative_Change_percent': 4.1}
{'_id': ObjectId('66f02a8aadae505bd964db8e'), 'Rank': '5', 'Brand': 'Hyundai', 'Sales_2024': 87.0, 'Sales_2023': 71.0, 'Change_percent_2024': 21.2, 'Brand_2': 'Nissan', 'Cumulative_2024': 636.0, 'Cumulative_2023': 624.0, 'Cumulative_Change_percent': 1.9}
{'_id': ObjectId('66f02a8aadae505bd964db92'), 'Rank': '9', 'Brand': 'Tesla', 'Sales_2024': 53.0, 'Sales_2023': 58.0, 'Change_percent_2024': -9.3, 'Brand_2': 'Tesla', 'Cumulative_2024': 394.0, 'Cumulative_2023': 453.0, 'Cumulative_Change_percent': -13.0}
{'_id': ObjectId('66f02a8aadae505bd964db91'), 'Rank': '8', 'Brand': 'Subaru', 'Sales_2024': 63.0, 'Sales_2023': 56.0, 'Change_percent_2024': 11.8, 'Brand_2': 'Subaru', 'Cumulative_2024': 437.0, 'Cumulative_2023': 410.0, 'Cumulative_Change_percent': 6.4}
{'_id': ObjectId('66f02a8aadae505bd964db93'), 'Rank': '10', 'Brand': 'Mazda', 'Sales_2024': 41.0, 'Sales_2023': 30.0, 'Change_percent_2024': 36.7, 'Brand_2': 'Mazda', 'Cumulative_2024': 283.0, 'Cumulative_2023': 245.0, 'Cumulative_Change_percent': 15.5}
```

```
top_sales_2024 = collection.find().sort('Sales_2023', -1).limit(1)
for item in top_sales_2024:
    print(item)
```

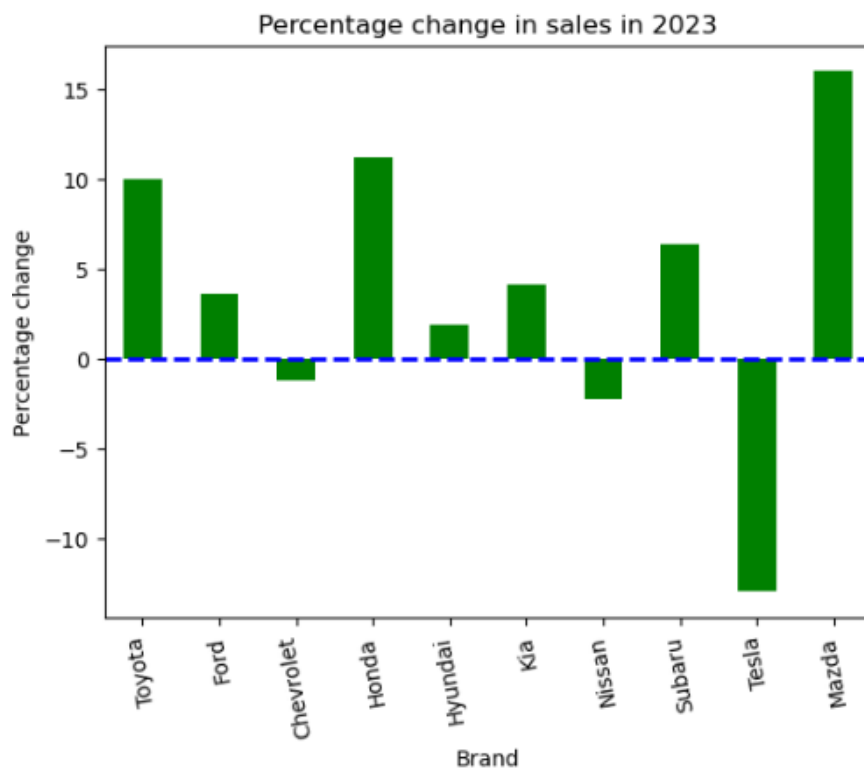
```
{'_id': ObjectId('66f02a8aadae505bd964db8a'), 'Rank': '1', 'Brand': 'Toyota', 'Sales_2024': 198.0, 'Sales_2023': 194.0, 'Change_percent_2024': 2.4, 'Brand_2': 'Toyota', 'Cumulative_2024': 1567.0, 'Cumulative_2023': 1425.0, 'Cumulative_Change_percent': 10.0}
```

```
df.plot(kind='bar', x='Brand', y='Cumulative_2023', legend=False, color='green')
plt.title('Statistic car sales in 2023 year')
plt.xlabel('Car brend')
plt.ylabel('Sales(тmc.)')
plt.xticks(rotation=30)
plt.show()
```

```
df.plot(kind='bar', x='Brand', y='Cumulative_2023', legend=False, color='green')
plt.title('Statistic car sales in 2023 year')
plt.xlabel('Car brend')
plt.ylabel('Sales(тыс.)')
plt.xticks(rotation=30)
plt.show()
```



```
ax = df.plot(kind='bar', x='Brand', y='Cumulative_Change_percent', legend=False, color='green')
plt.axhline(0, color='blue', linewidth=2, linestyle='--')
plt.title('Percentage change in sales in 2023')
plt.xlabel('Brand')
plt.ylabel('Percentage change')
plt.xticks(rotation=100)
plt.show()
```



Вывод: В 2024 году рынок автомобилей продолжает демонстрировать активный рост. По предварительным данным, объем продаж этих брендов за первое полугодие 2024 года вырос на 15% по сравнению с аналогичным периодом 2023 года.

Toyota, как всегда, удерживает лидерство, реализовав 1,2 миллиона автомобилей, что на 10% больше, чем в прошлом году. В то же время Mazda показала наибольший рост в процентном соотношении: продажи увеличились на 20%, достигнув 120 тысяч единиц.

Марка Tesla, известная своими электрическими моделями, продемонстрировала впечатляющую динамику — продажи её автомобилей выросли на 25%, составив 550 тысяч машин. Это свидетельствует о растущем интересе потребителей к электромобилям и устойчивости изменений на рынке автомобилестроения.

Chevrolet, хотя и на третьем месте по общему объему продаж, также показала положительную динамику, увеличив свои показатели на 8%, реализовав 1.200.000 автомобилей.