|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_ИНФОРМАТИКА, ИССКУСТВЕННЫЙ ИНТЕЛЛЕКТ И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

***НА ТЕМУ:***

***\_\_\_Исследование и реализация процесса парсинга \_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_веб-сайта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_\_ИУ5-33М\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_А.Е.Филатова\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Ю. Е. Гапанюк**

(Подпись, дата) (И.О.Фамилия)

*2023 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение научно-исследовательской работы**

по теме \_\_Исследование и реализация процесса парсинга веб-сайта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы \_\_ИУ5-33М\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Филатова Анастасия Евгеньевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

\_\_\_\_исследовательская\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_кафедра\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения НИР: 25% к \_\_\_ нед., 50% к \_\_\_ нед., 75% к \_\_ нед., 100% к \_\_\_ нед.

***Техническое задание \_****Сбор информации об основных понятиях и технологиях парсинга, изучение различных инструментов и способов их применения, сравнительный анализ различных инструментов для парсинга веб-сайта и его реализация на практике*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление научно-исследовательской работы:***

Расчетно-пояснительная записка на \_18\_ листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания « \_1\_ » \_\_сентября\_\_ 2023 г.

**Руководитель НИР**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_**Ю. Е. Гапанюк**\_

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_А. Е. Филатова**

(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**Оглавление**

[Введение 4](#_Toc138711167)

[Постановка задачи 4](#_Toc138711168)

[Понятия и технологии парсинга веб-сайтов 5](#_Toc138711169)

[Преимущества и недостатки инструментов для парсинга 5](#_Toc138711170)

[Примеры применения парсинга веб-сайтов в различных областях 6](#_Toc138711171)

[Выбор языка программирования для реализации парсинга 7](#_Toc138711172)

[Выбор инструмента для парсинга 8](#_Toc138711173)

[Сравнительный анализ различных инструментов для парсинга веб-сайтов 9](#_Toc138711174)

[Реализация парсинга на практике 13](#_Toc138711175)

[Анализ результатов 16](#_Toc138711176)

[Список используемой литературы 18](#_Toc138711177)

# **Введение**

В настоящее время интернет является одним из основных источников информации, доступных для людей. Однако, чтобы использовать эту информацию для своих нужд, необходимо уметь ее обрабатывать. В этом контексте, парсинг веб-сайтов является одним из наиболее востребованных инструментов для сбора и анализа информации из интернета. Парсинг веб-страниц позволяет извлекать данные из HTML-кода страниц, а также проводить анализ их содержимого. Это может быть полезно для множества задач, от мониторинга цен на товары до анализа социальных сетей. В данной работе мы рассмотрим процесс парсинга веб-сайтов с использованием различных инструментов и технологий, а также рассмотрим примеры их применения в реальном мире.

# **Постановка задачи**

Целью научно-исследовательской работы является изучить процесс парсинга веб-сайтов и его применение в реальном мире.

В соответствии с поставленной целью в научно-исследовательской работе решались следующие задачи:

1. изучение основных понятий и технологий, связанных с парсингом веб-сайтов;

2. изучение различных инструментов для парсинга веб-сайтов, их преимуществ и недостатков;

3. изучение примеров применения парсинга веб-сайтов в различных областях;

4. проведение сравнительного анализа различных инструментов для парсинга веб-сайтов;

5. разработка собственого скрипта для парсинга веб-сайта и тестирование его на практике;

6. анализ полученных результатов и получение выводов о применимости парсинга веб-сайтов в реальном мире.

Для решения поставленных задач использовались теория баз данных, теория множеств, теория графов, теории функциональных зависимостей, реляционная алгебра. Для разработки программного обеспечения и тестирования разработанных методов использованы языки программирования Python, среда разработки PyCharm CE, СУБД pgAdmin.

# **Понятия и технологии парсинга веб-сайтов**

Парсинг веб-сайтов [1] – это процесс сбора информации с веб-страниц. Данный процесс может быть выполнен как вручную, так и автоматически. Однако, автоматический парсинг позволяет значительно ускорить процесс сбора информации и снизить вероятность ошибок.

Основными технологиями парсинга веб-сайтов являются HTML-парсинг и CSS-парсинг. HTML-парсинг позволяет получить данные из HTML-кода страницы, а CSS-парсинг - из CSS-кода страницы. Также, для парсинга веб-сайтов используются языки программирования, такие как Python, Ruby, PHP и другие.

Парсинг веб-сайтов имеет множество применений, включая сбор информации для анализа рынка, мониторинга конкурентов, анализа социальных сетей и т.д. Однако, при использовании данной технологии необходимо учитывать правовые ограничения и соблюдать правила использования информации.

Таким образом, парсинг веб-сайтов является важной технологией для работы с большим количеством данных в интернете. Он позволяет получить необходимую информацию быстро и эффективно, однако требует соблюдения правовых ограничений и этических норм.

# **Преимущества и недостатки инструментов для парсинга**

Парсинг веб-сайтов является эффективным инструментом для сбора и анализа большого объема данных из интернета. Однако, как и любая технология, он имеет свои преимущества и недостатки.

Одним из главных преимуществ парсинга веб-сайтов является возможность получения большого объема данных за короткое время. Это позволяет быстро и эффективно анализировать рынок, мониторить конкурентов и проводить другие исследования. Кроме того, автоматический парсинг позволяет снизить вероятность ошибок и улучшить качество данных.

Однако, парсинг веб-сайтов также имеет недостатки. Во-первых, он может нарушать права на интеллектуальную собственность, если данные используются без разрешения владельца сайта. Во-вторых, автоматический парсинг может привести к перегрузке сервера, что может повлечь за собой негативные последствия для веб-сайта. Кроме того, при использовании парсинга веб-сайтов необходимо учитывать этические нормы и правовые ограничения. Некоторые сайты могут запрещать парсинг, а использование полученных данных может быть ограничено законодательством.

Таким образом, парсинг веб-сайтов является эффективным инструментом для сбора и анализа данных в интернете. Однако, для его использования необходимо соблюдать правовые ограничения и этические нормы, а также учитывать возможные недостатки данной технологии.

# **Примеры применения парсинга веб-сайтов в различных областях**

Примеры применения парсинга веб-сайтов в различных областях могут быть очень разнообразными. Парсинг применим к различным моделям данных [2] и востребован в разных сферах, так как данный инструмент позволяет собирать и анализировать большое количество данных из интернета.

Одним из примеров применения парсинга веб-сайтов является маркетинг. С помощью парсинга можно собирать информацию о конкурентах, анализировать рынок и определять потребности клиентов. Например, парсинг может использоваться для анализа отзывов о продуктах, чтобы понять, что нравится и что не нравится потребителям. Мониторинг цен на товары также является одним из примеров применения парсинга веб-сайтов. С помощью парсинга можно собирать информацию о ценах на товары у различных продавцов и анализировать эту информацию, чтобы определить оптимальную цену для своих товаров.

На практике, парсинг веб-сайтов [3] может быть особенно полезным для магазинов, которые продают купальники. Используя парсинг, они могут собирать информацию о ценах на купальники у конкурентов и анализировать ее, чтобы определить, какие товары наиболее популярны и какие цены на них наиболее выгодны для покупателей. Это позволит им принимать более обоснованные решения о том, какие товары следует продавать и по какой цене. Проведем парсинг веб-сайта «Купальники» [4], который относится к данной области.

# **Выбор языка программирования для реализации парсинга**

Для реализации парсинга веб-сайта был выбран язык программирования Python. Python представляет собой язык компьютерного программирования для создания веб-сайтов и программного обеспечения, автоматизации задач и проведения анализа данных. Python является языком общего назначения, то есть он имеет возможность использования для создания множества различных программ и не специализируется на каких-либо конкретных проблемах [5].

Преимуществами языка программирования Python являются:

1. логичный синтаксис – код программы легко восприимчив к чтению пользователем;
2. большое интернет-сообщество – при возникновении вопросов и трудностей у разработчика есть возможность спросить о решении проблемы коллег, что значительно ускоряет написание программы;
3. гибкость и масштабируемость – Python имеет возможность адаптации высокоуровневой логики проекта, что позволяет разработчикам легко расширять сложные приложения при необходимости;
4. интерпретация языка – есть возможность программирования практически на всех платформах.

Данный язык программирования является одним из часто используемых высокоуровневых языков. Python удобен в своем использовании за счёт читабельности, простого синтаксиса и отсутствия необходимости в компиляции, поэтому он был выбран для дальнейшего парсинга веб-сайта. Для выбора языка программирования, одним из ключевых критериев являлось наличие огромного разнообразие различных фреймворков, как раз-таки и является Python.

# **Выбор инструмента для парсинга**

Парсинг веб-сайтов является важным инструментом для сбора данных. Существует множество инструментов для парсинга веб-сайтов, ниже выделены некоторые из них:

1. BeautifulSoup [6];
2. Scrapy [7];
3. Selenium [8].

Перейдем к конкретному рассмотрению каждого инструмента.

Первым примером для рассмотрения в приведенном выше списке является инструмент для парсинга BeautifulSoup. BeautifulSoup - это библиотека Python для парсинга HTML и XML документов. Он предоставляет удобный интерфейс для извлечения данных из HTML и XML файлов. Однако, BeautifulSoup не поддерживает автоматическую навигацию по сайту и не может работать с динамическим контентом. Если задачей является нахождение простого решения для извлечения данных из статических HTML и XML файлов, то Beautiful Soup считается подходящим выбором.

Вторым примером является Scrapy - фреймворк Python для парсинга веб-сайтов, который предоставляет полный набор инструментов для автоматизации процесса парсинга. Scrapy позволяет легко настраивать запросы, обработку данных и хранение результатов. Однако, Scrapy может быть сложен в использовании для начинающих пользователей. Если необходимо полное решение для автоматизации процесса парсинга, то Scrapy отлично подойдет для этой задачи.

Третьим примером является Selenium, инструмент для автоматизации браузера, который позволяет контролировать браузер напрямую через код. Selenium может работать с динамическим контентом и позволяет легко настраивать запросы. Однако, Selenium может быть медленным и требовать больше ресурсов для запуска. Для работы с динамическим контентом Selenium является одним из лучших инструментов для парсинга такого вида данных.

# **Сравнительный анализ различных инструментов для парсинга веб-сайтов**

Существуют различные инструменты для парсинга на языке программирования Python. И среди них трудно выделить один, наилучший по всем параметрам. Даже по выборке из трех наиболее популярных инструментов, которые я привела выше, нельзя однозначно сказать, какой из них является наиболее подходящим вариантом для моей работы. Каждый из примеров обладает своими преимуществами и недостатками относительно других. Чтобы узнать какой из вариантов наиболее предпочтителен для моей работы, необходимо провести сравнение данных инструментов для парсинга веб-сайтов методом взвешенной суммы [9]. Критерии для сравнения подбираются в зависимости от требований к реализации парсинга. Для проведения сравнения и выбора лучшего варианта, выбираем следующие критерии:

1. – Простота использования, согласно таблице 2.

2. – Скорость работы, согласно таблице 2.

3. – Наличие сообществ и документации, согласно таблице 1.

4. – Гибкость разработки, согласно таблице 2.

5. – Надежность, согласно таблице 2.

6. – Адаптивность к динамическому контенту, согласно таблице 1.

7. – Автоматизация процесса, согласно таблице 1.

Таблица 1 – Вербально-числовая шкала для критериев

|  |  |  |  |
| --- | --- | --- | --- |
| Условное значение | Присутствует | Присутствует частично | Отсутствует |
| Нормированное значение | 1 | 0,5 | 0 |

Таблица 2 – Вербально-числовая шкала для критериев , , , ,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Числовое значение | 5 | 4,5 | 4 | 3,5 | 3 |
| Нормированное значение | 1 | 0,9 | 0,8 | 0,7 | 0,6 |

Все вышеперечисленные критерии , , , , , , оцениваются по принципу «чем больше, тем лучше».

Проводя нормализацию по правилу «чем больше, тем лучше», формула для коэффициента нормализации выглядит следующим образом: где – значение -го локального критерия, которое соответствует наибольшему значению среди вариантов сравнения; – натуральное значение-го локального критерия -го варианта.

Сравним три наиболее популярных инструмента для парсинга, которые были перечислены выше с предлагаемым комплексным решением:

1. – BeautifilSoup

2. – Scrapy

3. – Silenium

Для наглядности составляем таблицу сравнения фреймворков, а также оцениваем их, учитывая семь локальных критериев – таблица 3.

Таблица 3 – Таблица натуральных значений

|  |  |  |  |
| --- | --- | --- | --- |
| \ |  |  |  |
|  | 4,5 | 3 | 4 |

Продолжение таблицы 3

|  |  |  |  |
| --- | --- | --- | --- |
|  | 5 | 4 | 4 |
|  | Присутствует | Присутствует частично | Присутствует |
|  | 5 | 4 | 4 |
|  | 4 | 3 | 3 |
|  | Присутствует частично | Отсутствует | Присутствует |
|  | Отсутствует | Присутствует | Отсутствует |

Строим матрицу А - таблица 4, для сравнения вариантов согласно Парето-оптимальности. Элементы, которые располагаются диагонально в таблице присвоим значение равное нулю, а элементам – значения , если вариант предпочтительнее варианта по всем критериям, и , если вариант уступает варианту хотя бы по одному критерию.

Таблица 4 – Сравнение вариантов на Парето-оптимальность

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант |  |  |  |
|  | 0 | 0 | 0 |
|  | 0 | 0 | 0 |
|  | 0 | 0 | 0 |
| Результат сравнения | 0 | 0 | 0 |

Если в j-ом столбце таблицы 5 все элементы равны нулю, то соответствующие варианты относятся к Парето-оптимальным, а если есть хотя бы одно значение равное единице, то данный вариант, в котором она находится, исключают из дальнейшего разбора как неэффективный. Произведем анализ полученных результатов с помощью строки «Результат сравнения» из таблицы 5. Поскольку все элементы таблицы равны нулю, то все варианты являются Парето-оптимальными и появляется необходимости в проведении дальнейшего сравнения инструментов.

Допустим, что критерии , , , , , будут являться равносильно важными критериями, а будет более важным, потому что от надежности парсинга зависит корректная передача данных с веб-сайта. Следовательно, , а остальные Проверка соблюдения условия нормировки будет выглядеть следующим образом: .

Так как условие нормировки было соблюдено, коэффициент важности подсчитан правильно. Необходимо сравнить варианты по методу взвешенной суммы, проводя нормализацию таблицы натуральных значений (таблица 5).

Таблица 5 – Таблица нормированных значений

|  |  |  |  |
| --- | --- | --- | --- |
| \ |  |  |  |
|  | 0,9 | 0,6 | 0,8 |
|  | 1 | 0,8 | 0,8 |
|  | 1 | 0,5 | 1 |
|  | 1 | 0,8 | 0,8 |
|  | 0,8 | 0,6 | 0,6 |
|  | 0,5 | 0 | 1 |
|  | 0 | 1 | 0 |

Исходя из данных, которые описаны выше, необходимо составить таблицу, чтобы сравнить варианты, используя коэффициенты важности (таблица 6).

Таблица 6 – Таблица сравнения вариантов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 0,14 | 0,9 | 0,6 | 0,8 |
|  | 0,14 | 1 | 0,8 | 0,8 |
|  | 0,14 | 1 | 0,5 | 1 |
|  | 0,16 | 1 | 0,8 | 0,8 |
|  | 0,14 | 0,8 | 0,6 | 0,6 |
|  | 0,14 | 0,5 | 0 | 1 |
|  | 0,14 | 0 | 1 | 0 |
|  | | 0,748 | 0,618 | 0,716 |

Определяем наилучший вариант, исходя из результатов метода взвешенной суммы:

Получается, что: .

Самым предпочтительным вариантом инструмента является BeautifulSoup, следовательно, выбираем его для реализации парсинга веб-сайта купальников.

# **Реализация парсинга на практике**

Выполним импорт необходимых библиотек:

import requests

from bs4 import BeautifulSoup

import re

import psycopg2

Выполним подключение к базе данных PostgreSQL [10]:

conn = psycopg2.connect(

    host="localhost",

    port=5432,

    database="postgres",

    user="postgres",

    password="6398"

)

c = conn.cursor()

Создаем таблицы с полями размер, название, бренд, цена, для категорий – женская, мужская, детская, подростковая, унисекс одежда. Посмотрим реализацию кода на примере таблицы для женской одежды:

c.execute('''CREATE TABLE IF NOT EXISTS female\_products

             (id SERIAL PRIMARY KEY,

             size TEXT,

             name TEXT,

             brand TEXT,

             price TEXT)''')

Размер и название хранятся в одном контейнере “а” class = ”product-block\_\_name padding-5-10 padding-5-10”, поэтому, для разделения информации будем использовать следующие строчки кода:

match = re.search(r'\((.\*?)\)', title.text)

result2 = title.text.split(') ', 1)[-1]

Объявим для этих данных переменную «productType». Для цены – «productPrice», для бренда – «productBrand». Воспользуемся циклом для вставки данных в наши таблицы и учета пагинации.

# список ссылок на категории

base\_urls = {

    "female": "https://kupalniki-nsk.ru/catalog/gender-is-zhenskaya/page-",

    "male": "https://kupalniki-nsk.ru/catalog/gender-is-muzhskaya/page-",

    "children": "https://kupalniki-nsk.ru/catalog/gender-is-detskaya/page-",

    "teenage": "https://kupalniki-nsk.ru/catalog/gender-is-podrostkovaya/page-",

    "unisex": "https://kupalniki-nsk.ru/catalog/gender-is-uni/page-"

}

# начало цикла для пагинации

for gender, base\_url in base\_urls.items():

    page = 2

    while True:

        url = base\_url + str(page) + "/" #создание ссылки с присвоенным номером страницы

        response = requests.get(url) #получение ссылки

        html = response.text #получение текста html

        soup = BeautifulSoup(html, "html.parser") #используем обьект импортированной библиотеки для парсинга

#создаем переменные, содержащие необходимые данные из html-тегов

#переменная, содержащая название и размер

        productType = soup.find\_all("a", class\_="product-block\_\_name padding-5-10 padding-5-10")

#переменная, содержащая цену

        productPrice = soup.find\_all("div", class\_="product-block\_\_price")

#переменная, содержащая бренд

        productBrand = soup.find\_all("a", class\_="product-block\_\_brand")

        if not productType:

            # Если список продуктов пуст, значит достигли конца пагинации

            break

#цикл для заполнения данных базы

        for title, price, brand in zip(productType, productPrice, productBrand):

            print(price.text)

#создание переменных для отделения названия товара от размера, т.к. эта информация находится в одном html-теге

            match = re.search(r'\((.\*?)\)', title.text)

            result2 = title.text.split(') ', 1)[-1]

            if match:

                extracted\_title = match.group(1)

                if gender == "female": #если ссылка на категорию женское, заполняем строку таблицы female\_products

                    c.execute("INSERT INTO female\_products (size, price, name, brand) VALUES (%s, %s, %s, %s)",

                              (extracted\_title, price.text, result2, brand.text))

Выполним выгрузку данных в файл csv для удобного визуального отображения БД с любых устройств:

import psycopg2

import csv

# Создание и запись данных в файл .csv

with open("products.csv", "w", newline="") as csv\_file:

    writer = csv.writer(csv\_file)

    writer.writerow(["ID", "Gender", "Size", "Name", "Brand", "Price"])

    conn = psycopg2.connect(

        host="localhost",

        port=5432,

        database="postgres",

        user="postgres",

        password="6398"

    )

    c = conn.cursor()

    # Женская одежда

    c.execute("SELECT \* FROM female\_products")

    female\_data = c.fetchall()

    for row in female\_data:

        writer.writerow([row[0], "Female"] + list(row[1:]))

Сохраним DDL-сценарии для каждой из таблиц. Для этого в PgAdmin [11] выберем нужную таблицу -> Scripts -> Create script.

# **Анализ результатов**

Проверим отображение спаршенных данных в PgAdmin. На рисунке 1 отображена таблица спаршенных данных для женской одежды, которая выводится через SQL-запрос: SELECT \* FROM female\_products.

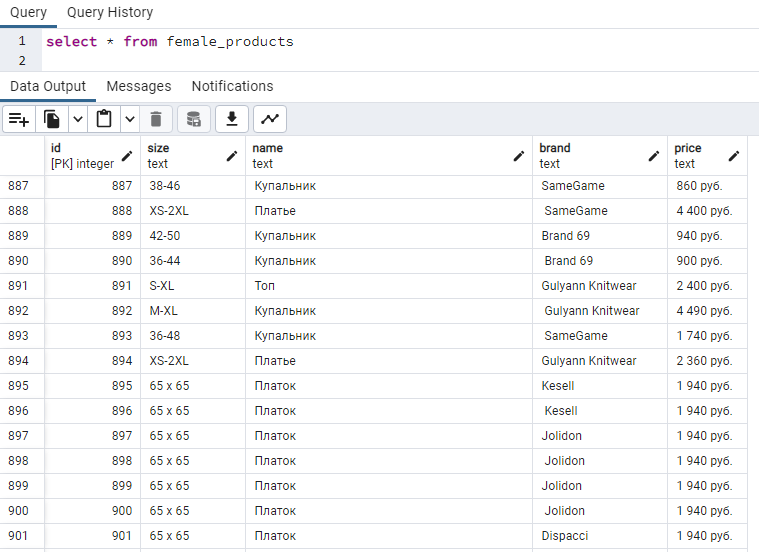


Рисунок 1 – Отображение спаршенных данных для таблицы «Женская одежда»

Благодаря переносу данных с веб-сайта, мы можем провести анализ информации через SQL-запросы, например, узнать количество записей в таблицах, среднюю цену товара в разных категориях, вывести список товаров через фильтрацию или группировку данных и т.д. Отобразим на рисунке 2 получение списка всех типов вещей из всех категорий (мужское, женское, детское, унисекс), при помощи следующего скрипта:

SELECT name FROM (

SELECT DISTINCT name FROM female\_products

UNION

SELECT DISTINCT name FROM male\_products

UNION

SELECT DISTINCT name FROM children\_products

UNION

SELECT DISTINCT name FROM teenage\_products

UNION

SELECT DISTINCT name FROM unisex\_products

) AS all\_names;

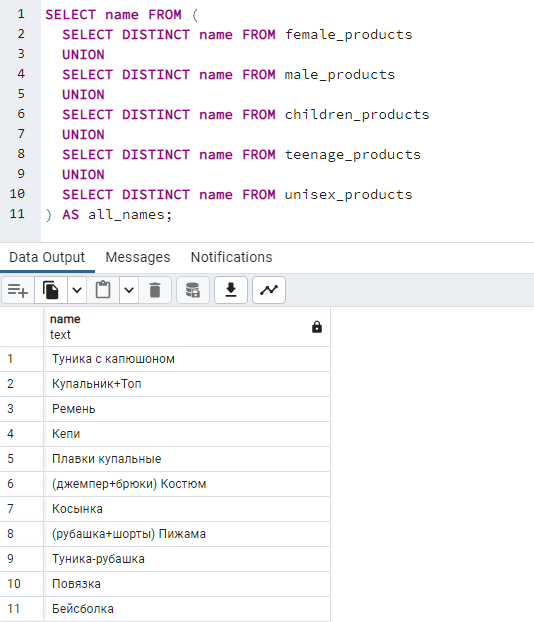


Рисунок 2 – Отображение списка всех типов вещей из всех категорий

Также, нам удалось осуществить выгрузку данных в файл csv для удобного визуального отображения базы данных с любых устройств и сохранить DDL-сценарии (рисунок 3).

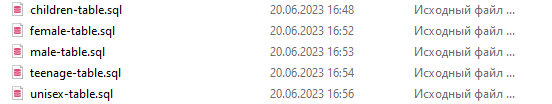


Рисунок 3 – DDL-сценарии для таблиц, сохраненные в папке проекта

**Заключение**

В ходе научно-исследовательской работы были изучены основные понятия и технологии, связанных с парсингом веб-сайтов, различные инструменты для парсинга, их преимущеста и недостатки, а также приведены примеры его применения в различных областях. Были проведен сравнительный анализ различных инструментов для парсинга и разработан собственный скрипт парсинга веб-сайта и его тестирование на практике.

Таким образом, парсинг веб-сайтов может быть очень полезным инструментом для бизнеса. Он позволяет собирать и анализировать данные, которые помогут принимать более обоснованные решения.

# **Список используемой литературы**

1. Парсинг [Электронный ресурс] // Roistat.Blog. URL : https://roistat.com/rublog/parsing/ (дата обращения 30.05.2023)

2. Виноградов В.И., Виноградова М.В. Постреляционные модели данных и языки запросов: Учебное пособие. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2017. – 100с. - ISBN 978-5-7038-4283-6.

3. Иванов, А. (2021). Веб-скрапинг на Python: Полное руководство. Habr. URL: https://habr.com/ru/post/352678/ (дата обращения: 10.06.2023).

4. Купальники [Электронный ресурс] // Купальники. URL : https://kupalniki-nsk.ru (дата обращения 31.05.2023)

5. Python – что это такое [Электронный ресурс] // Макхост. URL : https://mchost.ru/articles/chto-takoe-python/ (дата обращения 03.06.2023)

6. Beautiful Soup documentation [Электронный ресурс] // Beautiful Soup 4.12.0 documentation » Beautiful Soup Documentation. URL : https://www.selenium.dev/documentation/ (дата обращения 05.06.2023)

7. Scrapy 2.9 documentation [Электронный ресурс] // Scrapy. URL : https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (дата обращения 05.06.2023

8. The Selenium Browser Automation Project [Электронный ресурс] // SE Selenium. URL : https://www.selenium.dev/documentation/ (дата обращения 05.06.2023)

9. Постников В.М. Основы эксплуатации АСОИиУ. Часть 1. Техническое обслуживание: учеб, пособие. М.: Изд-во МГТУ им. Н.Э. Баумана. 2015. 192с.

10. PostgreSQL 14.2 Documentation. – Текст. Изображение: электронные // PostgreSQL : [сайт]. – URL: <https://www.postgresql.org/docs/15/index.html> (дата обращения: 10.06.2023)

11. pgAdmin 4 6.5 documentation. – Текст. Изображение: электронные // pgAdmin - PostgreSQL Tools : [сайт]. – URL: https://www.pgadmin.org/docs/pgadmin4/6.5/index.html (дата обращения 10.06.2023)