**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет «Высшая школа экономики» Факультет компьютерных наук**

**О Т Ч Е Т**

**по дисциплине**

**«Теория Баз Данных»**

**Выполнил студент гр.**

**Копчев Владислав Олегович, Жестянников Андрей**

**(ФИО)**

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

***(подпись)***

**Проверил:**

*(должность, ФИО руководителя практики)*

*20.12.2021*

*(дата)*

**2021 год**

Содержание

Содержание 2

1. Описание предметной области 3
   1. Цель 3
   2. Внешние данные 3
   3. Основные сценарии использования 3
2. Концептуальная модель 3
   1. Диаграмма «Сущность-связь» 3
   2. Описание сущностей и связей 4
3. Инфологическая модель 5
   1. Диаграмма «Таблица-связь» 5
   2. Словарь данных 5
4. Даталогическая модель 6
   1. Используемая СУБД и диалект SQL 6
   2. Описание механизмов обеспечения целостности данных 6
   3. DDL-скрипты 7
5. Скрейпинг 7
6. Клиентское приложения 7
   1. Архитектура 7
   2. Сценарии использования 7
   3. Организация доступа к данным 7
   4. Интерфейс с пользователем 7
   5. Отчёты 7
7. Гайд по XPath 10
8. Заключение 11
   1. Объёмные характеристики разработки 11
   2. Авторский вклад и комментарии по выполнению проекта 11
9. Источники 11

# Описание предметной области

## Цель

Требуется разработать систему сбора данных с сайтов о продуктах. Программа должна пройти по сайту http://[vons.com](http://von.com) и собрать категории первого, второго и третьего уровня, притом для категорий третьего и второго уровней должны быть указаны их родительские категории. По данным, собранным данной программой, необходимо построить базу данных, содержащую таблицу с категориями и подкатегориями. Также необходимо разработать веб-сервис, который может получать список категорий из базы данных и добавлять свои.

## Внешние данные

1. Сайт [vons.com](http://vons.com)
2. Категории и их субкатегории на сайте

## Основные сценарии использования

* 1. Манипуляция базой данных с категориями
  2. Построение рекомендательной системы на основе данной базы данных.

# Концептуальная модель

## Диаграмма «Сущность-связь»Image

Рисунок 2.1. Диаграмма в нотации Чена

## Описание сущностей и связей

Описание сущностей:

* **Site - сайты, с которых надо собирать данные**
  + id
  + Title - название  сайта для сбора данных
  + url
  + Comment - комментарии
  + Options - особенности для сбора данных
* **Experiments\_run** - запуски краулера
  + id
  + Date\_start - время начала сбора
  + Date\_stop - время завершения сбора
  + Options - параметры запуска клаулера
* **Product Type (бананы)**
  + Id
* **Product (бананы в Магните)**
  + Description - Описание (текст в произвольной форме)
  + Specification - Спецификация данные с сайта
  + nutritional - Ингридиенты
  + category\_id -> FK supplier\_category\_id
  + app\_category\_id -> FK app\_category
  + catalog\_product\_id  - FK к каталогу продуктов
  + price

* **Сategory** - Категории (таксономия)
  + Id - int
  + Title - название катерогии

* **Supplier\_category** - Таксономия продавца
  + supplier\_category\_id - категории на сайте
  + supplier\_parent\_category\_id
  + app\_category\_id -> FK category

* **Сatalog\_product**- Каталог продуктов на сайте
  + Title - название типа
  + Upc - код
  + Description - описание
  + Brand - Производитель
  + Model - модель
  + Images - Ссылка на картинки

Описание связей:

* + Product appears in the catalog\_product: продукт находится в некотором каталоге
  + Product has product\_type: продукт имеет некоторый тип в таксономии рекомендательной системы
  + Product on the site: продукт находится на некотором сайте
  + Product has category: продукт относится к категории в таксономии рекомендательной системы
  + сайта
  + Experiment\_run gathers data from site: запуск программы собирает данные с некоторого сайта
  + Category appears in suppliers\_category: продукт относится к категории в таксономии сайта

Примеры категорий: Coconut Water, Bread & Bakery

Примеры типов: одежда, еда

Зачем разделять категории рекомендательной системы и сайта онлайн-магазина? Поскольку один товар на разных сайтах может иметь разные названия категорий ('Desserts & Cheesecakes’ и просто ‘Десерты’), поэтому названия категорий на сайтах надо будет переводить в категории рекомендательной системы. Зачем нужны типы? Поскольку для одежды и напитков сравнивать надо данные с разных сайтов — соответственно, необходимо в зависимости от типа подбирать разные сайты.

Данная диаграмма отражает базовую идею рекомендательной системы и в будущем будет видоизменяться.

# Инфологическая модель

## Диаграмма «Таблица-связь»Image

Рисунок 3.1. Диаграмма в нотации Чена

## Словарь данных

Составлю таблицу для тех столбцов в таблицах, где информация о них не очевидна из словаря сущностей.

| **Источник данных** | **Идентификатор данных** | **Назначение данных** | **Диапазон значений** | **Тип данных** |
| --- | --- | --- | --- | --- |
| **[vons.com](http://vons.com)** | Product.Description | Описание продукта на сайте (К нему будут применены алгоритмы NLP) | - | Text |
| **Администратор БД** | Category.Title | Различать категории не только по id | - | Text |
| **Администратор БД** | Experiments\_run.Options | Параметры запуска программы для сбора данных | - | Text |
| **Администратор БД** | Site.options | С какими параметрами запускать сбор данных с этого сайта | - | Text |
| **vons.com** | Product.price | Цена товара на сайте | > 0 | Int |

# Даталогическая модель

## Используемая СУБД и диалект *SQL*

Использована СУБД PostgreSQL, диалект PL/pgSQL.

Поддерживается третья нормальная форма. В силу простоты БД решено было не добавлять хранимые процедуры и триггеры

## Описание механизмов обеспечения целостности данных

1. Каскадное удаление и обновление идентификаторов категорий: app\_category\_id int REFERENCES Categories(Id) ON DELETE CASCADE ON UPDATE RESTRICT. Если мы удалили категорию (ее id), то ее нужно везде удалить, иначе потом будут несостыковки. А вот обновление категории может привести к коллизиям и вообще непонятно, зачем нужно, так что его надо запретить!
2. Аналогично сделано со всеми id, что вполне логично. Поскольку связи есть только между id, ничего другого делать и не надо.

## *DDL*-скрипты

1. Создание таблиц в БД: <https://github.com/aefrt/database-theory/blob/aefrt-patch-2/DDL-CREATE.sql>
2. Заполнение таблицы с категориями продавца: <https://github.com/aefrt/database-theory/blob/aefrt-patch-2/DDL-INSERT.sql>

# Скрейпинг

* 1. Код для скрейпинга: <https://github.com/aefrt/database-theory/blob/aefrt-patch-2/scrapping.py>
  2. Код для создания DDL вставки (DDL для создания написан вручную, результат работы кода для заполнения таблицы): <https://github.com/aefrt/database-theory/blob/aefrt-patch-2/DDL-INSERT.py>
  3. Результат скрейпинга: [https://github.com/aefrt/database-theory/blob/aefrt-patch-2/итог%20работы.rtf](https://github.com/aefrt/database-theory/blob/aefrt-patch-2/%25D0%25B8%25D1%2582%25D0%25BE%25D0%25B3%2520%25D1%2580%25D0%25B0%25D0%25B1%25D0%25BE%25D1%2582%25D1%258B.rtf)

# Клиентское приложения

## Архитектура

Приложение на Flask.

## Сценарии использования

1. Получение списка категорий
2. Добавление категорий

## Организация доступа к данным

Методы Post, Get, sqlite3 connect().

## Интерфейс с пользователем

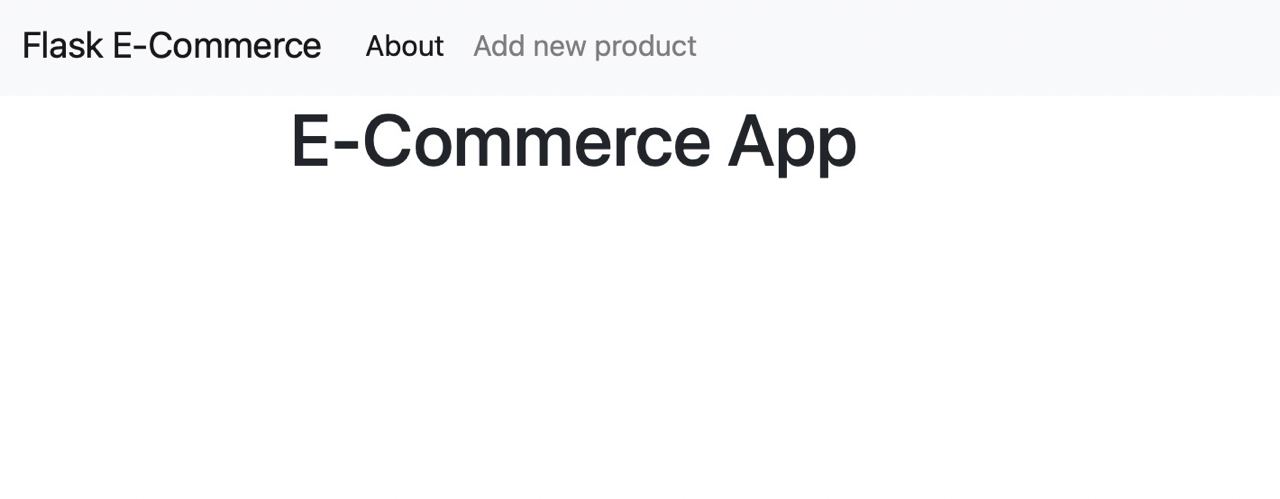
Веб-приложение

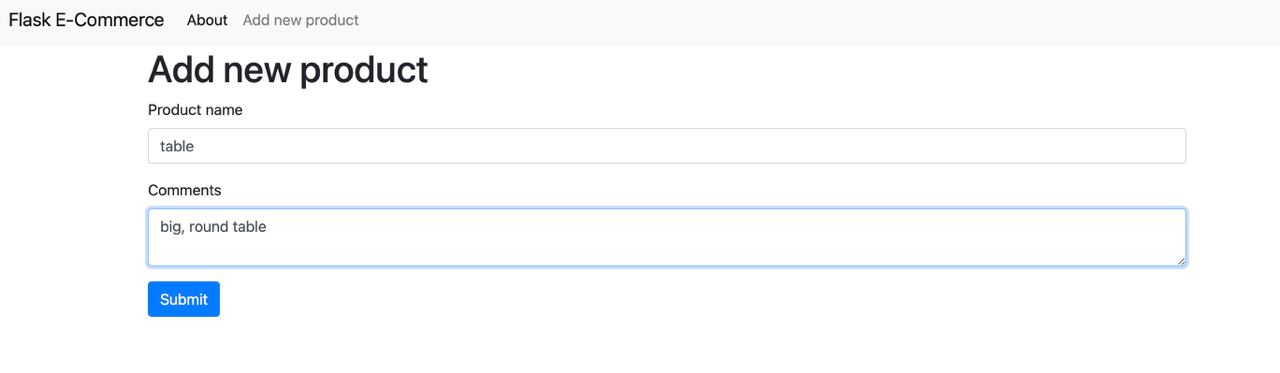
## Отчёты

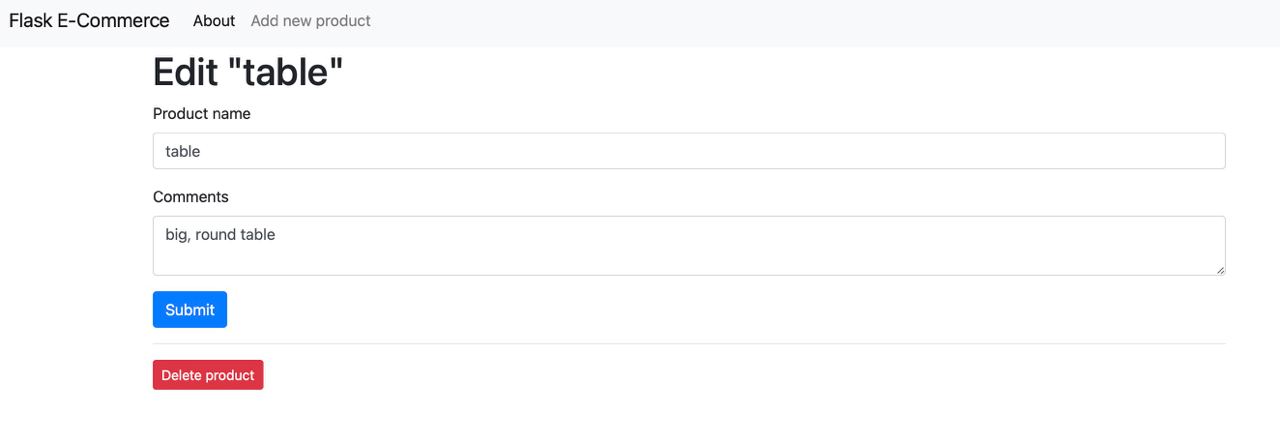
Были проведены операции по добавлению, изменению, удалению данных. Ошибок обнаружено не было

Демонстрация интерфейса:

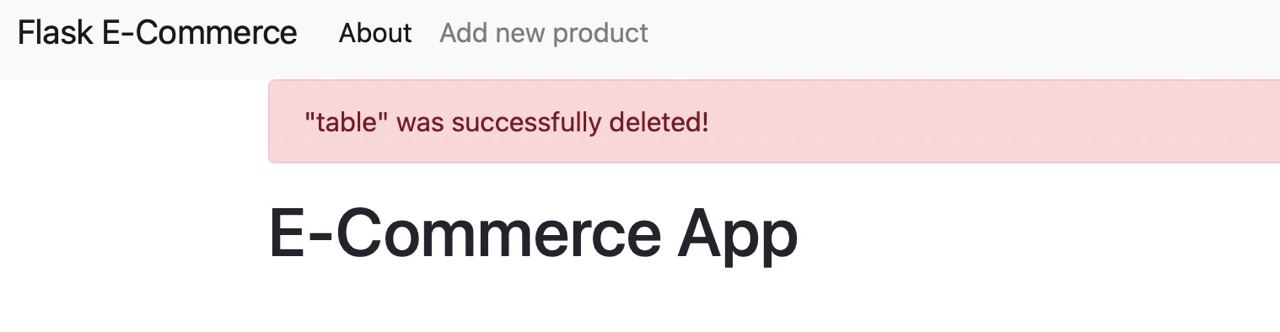
Главная страница:



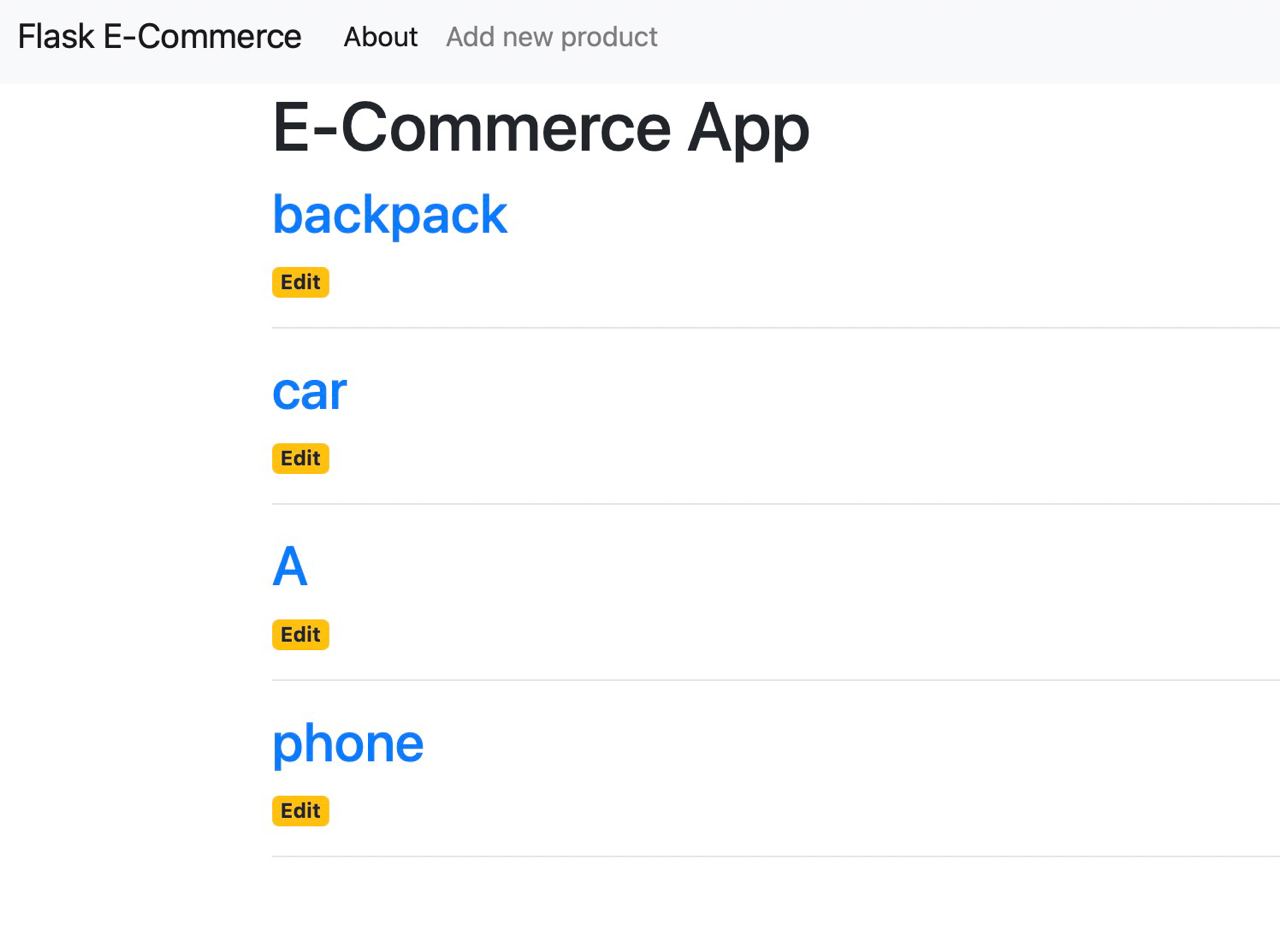
Добавление продукта:

Изменение данных:

Удаление данных:



Просмотр данных:



# Гайд по XPath

В будущем в проекте планируется использование XPath, поэтому напишу здесь Гайд по нему.

XPath (XML Path) — это язык запросов, для навигации по XML-документу и работы с его частями. Данный язык был создан W3C в 1999 году и используется в случаях, когда необходимо обработать XML-документ с помощью Python, Java или C\#.

библиотека BeautifulSoup не поддерживает XPath, однако с ним очень удобно работать, используя другие библиотеки. Немотря на то, что XPath создан для работы с XML-документами, его также можно использовать и для работы с HTML-документами

В основе синтаксиса XPath есть 4 основных концепта:

- Различия в работе с корневыми и некорневыми вершинами

- `/div` выбирает вершину `div` только если она является корнем документа

- `//div` позволяет обращаться ко всем `div`'ам где угодно в любом месте в документе

- Возможность выбора атрибутов

- `//@href` выберет все вершины с атрибутом `href`

- `//a[@href='http://google.com']` выберет все ссылки в документы, которые ведут на сайт Google

- Возможность выбора вершин по позиции

- `//a[3]` выбирает третью ссылку в документе

- `//table[last()]` выбирает последнюю таблицу в документе

- `//a[position() < 3]` выбирает первые три ссылки в документе

- Звездочка обозначает любой набор вершин или символов и может быть использована в различных ситуациях

- `//table/tr/\*` выбирает всех потомков тега `tr` во всех таблицах

- `//div[@\*]` выбирает все теги `div`, у которых етсь хоть какие-то атрибуты

У XPath огромное число и более интересных возможностей. За годы развития XPath стал сложным языком запросов с логическими операциями, функциями и т. д. Хорошая документация по XPath есть у [Microsoft](<https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-4.0/ms256471(v=vs.100)?redirectedfrom=MSDN)>

# Заключение

Код проекта находится в <https://github.com/aefrt/database-theory>

## Объёмные характеристики разработки

7 сущностей, 6 связей между ними, есть связи “многие-ко-многим”.

## Авторский вклад и комментарии по выполнению проекта

В дальнейшем планируется собрать категории с большего числа сайтов, собрать данные о товарах, написать код на XPath, разработать большую рекомендательную систему на основе алгоритмов NLP.

# Источники

**Database Design and Relational Theory. Normal Forms and All That Jazz** [Book] / auth. Date C.J.. - [б.м.] : O’Reilly Media, 2012. - p. 260.

**SQL Server technical documentation** [Online] / auth. Microsoft. - 10 10, 2020. - https://docs.microsoft.com/en-us/sql/sql-server/.

**Код проекта** [В Интернете]. - 20 09 2021 г.. - http://github.com/aefrt/database-theory