МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Национальный Исследовательский Ядерный Университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра «Кибернетика» (№22)

Отчёт о работе по курсу

**Базы данных (теоретические основы баз данных)**

Подготовил

студент группы Б20-514

Фонов Н.Д.

Москва, 2022

Оглавление

[1. Предметная область 3](#_Toc122633820)

[2. Концептуальное проектирование. 5](#_Toc122633821)

[3. Физическая схема 8](#_Toc122633822)

[4. Создание таблиц базы данных 9](#_Toc122633823)

[5. Заполнение базы данных 10](#_Toc122633824)

[5.1. Парсинг данных 10](#_Toc122633825)

[5.2. Итоговое заполнение базы данных 13](#_Toc122633826)

[6. Запросы к базе данных 15](#_Toc122633827)

1. Предметная область

На данный момент оригинального сайта E-Katalog не существует, однако недавно появилась версия для Казахстана. Именно она была взята за основу для проектирования базы данных.

E-Katalog представлял собой сайт, в котором можно было найти определенный товар с определенными характеристиками. Зайдя на страницу, товара пользователь видел модификации данного товара. Нажав на одну из них, пользователю высвечивался список магазинов и цен в данных магазинах (+ ссылки на каждый магазин). Таким образом находилось оптимальное место покупки товара.

E-Katalog не имел функционала покупки товара. Все взаимодействие с ним заканчивалось на моменте выбора товара и магазина, в котором данный товар присутствует.

Пользователь может просматривать товары, сортировать их, смотреть цены на товары, а также магазины, которые предоставляют такие предложения. Также пользователь может добавлять товары в список favorites.

У E-Katalog была рекомендательная система, опирающаяся на то, какие товары и когда пользователь просматривал.

Так как в E-Katalog было огромное множество товаров, у каждого из них были специфические характеристики. Т.е. нельзя создать одну большую таблицу со всеми товарами и характеристиками (иначе получится огромное число столбцов). Для решения этой проблемы был выбран способ EAV (Entity Attribute Value). Создается таблица, в которой для каждой характеристики товара есть отдельная строка с id товара, id характеристики и ее значением.

Поскольку у товара есть модификации (например, цвет телефона, кол-во встроенной памяти), то для товара и для модификации товара необходимо создать таблицы. Для каждой из них также по методу EAV создается по таблице с характеристиками.

Для списка магазинов, продающих товар создается отдельная таблица, в которой находится id товара, id магазина и ссылка со стоимостью (по тому же методу EAV).

1. Концептуальное проектирование.
2. User.

Описание: Таблица с пользователями.

Содержит атрибуты: nickname, email, password.

1. Brand.

Описание: Таблица, в которой содержатся брэнды и описание данных брендов.

Содержит атрибуты: brand\_name и brand\_info.

1. Categories.

Описание: Таблица, в которой содержатся категории товаров и описание данных категорий.

Содержит атрибуты: category и description.

1. Properties.

Описание: Таблица, в которой прописываются возможные характеристики для различных типов товаров (например, встроенная память, экран и т.д.) и пояснения данных характеристик для пользователя (подобное описание было реализовано на сайте).

Содержит атрибуты: property, description.

1. ModProperty.

Описание: Таблица, в которой прописываются возможные характеристики для каждой модификации товара (например, цвет, кол-во памяти для конкретной модификации модели) и описание данных характеристик.

Содержит атрибуты: mod\_property, description.

1. Shop.

Описание: Таблица магазинов. Для упрощенного просчета средней оценки конкретного магазина вводится дополнительное поле count\_reviews, которое при добавлении новой оценки инкрементируется на

Содержит атрибуты: shop\_name, rating, home\_page, count\_reviews.

1. SearchHistory.

Описание: История просмотров товаров пользователем (нужна для рекомендаций). Дата записывается для отображения релевантных на данный момент предложений пользователю.

Содержит атрибуты: customer\_id [FK], model\_id [FK], search\_date.

1. ShopReview.

Описание: Обзор магазина пользователем.

Содержит атрибуты: customer\_id [FK], shop\_id [FK], review, score.

1. ProductReview.

Описание: Обзор продукта пользователем.

Содержит атрибуты: customer\_id [FK], product\_mod\_id [FK], review, score.

1. Product.

Описание: Таблица с конкретным товаром.

Содержит атрибуты: product\_name, brand\_id [FK], category\_id [FK].

1. ProductProperties.

Описание: Таблица, в которой связываются конкретный продукт и характеристики со значениями. Реализуется по методу EAV.

Содержит атрибуты: product\_id [FK], property\_id [FK], property.

1. ProductMod.

Описание: Таблица с определенной модификацией конкретного товара. Под модификацией понимается товар с конкретным набором изменяемых характеристик. Например, для телефона модификацией может быть кол-во встроенной памяти.

Содержит атрибуты: product\_id [FK], product\_mod\_name

1. ShopProductMod.

Описание: Таблица с продуктом (его модификацией) и магазином, который продает его. В данной таблице также присутствует цена и ссылка на данный товар в данном магазине.

Содержит атрибуты: product\_mod\_id, shop\_id, cost, link.

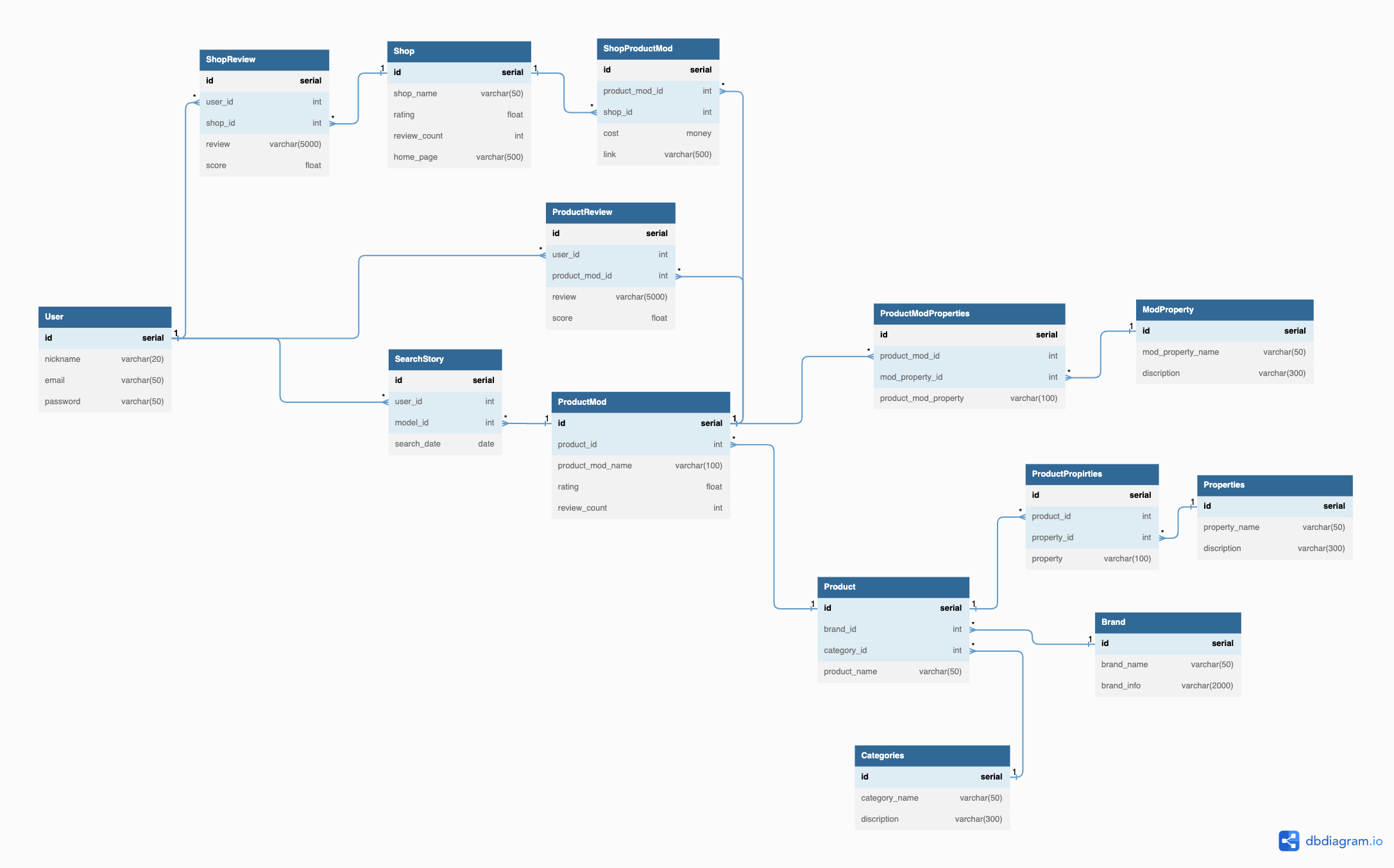
1. ProductModProperties.

Описание: Таблица, в которой связываются конкретная модификация продукта и характеристики со значениями. Реализуется по методу EAV.

Содержит атрибуты: product\_mod\_id [FK], mod\_property\_id [FK], value.

1. Физическая схема

Была проработана физическая схема БД для сайта-агрегатора товаров E-Katalog.



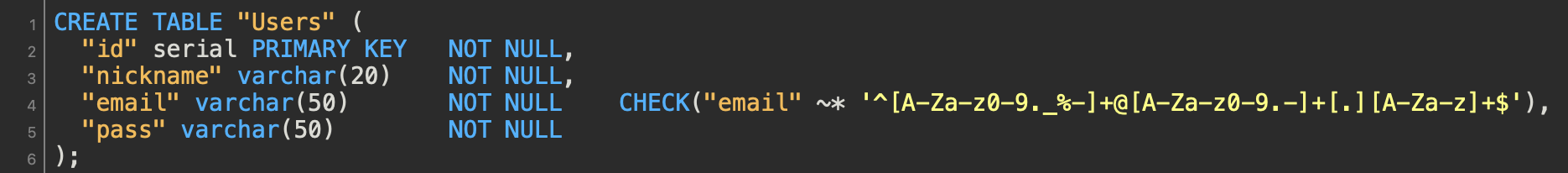
На данной схеме отображены сущности с атрибутами, их связи, типы атрибутов (напротив названия атрибутов) и ключи (жирный текст).

1. Создание таблиц базы данных

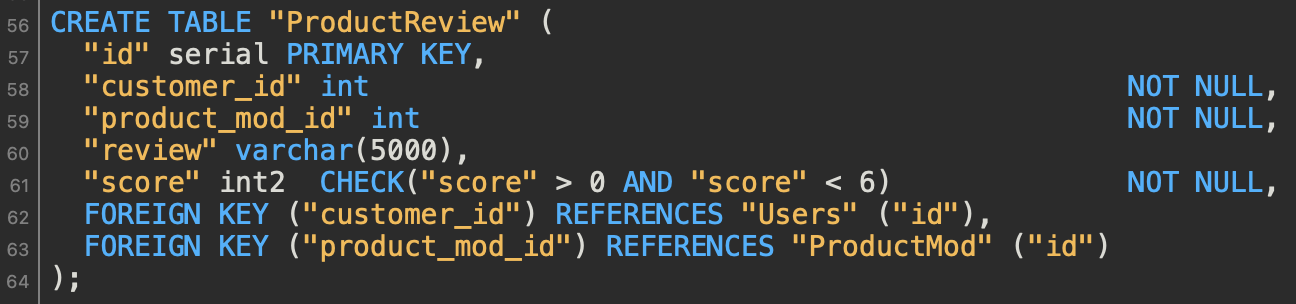
Для реализации базы данных в качестве СУБД был выбран PostgreSQL. Код на языке SQL выполнялся в графическом клиенте Postico 2.

Ниже приведен код на языке SQL для создания таблиц для некоторых сущностей.

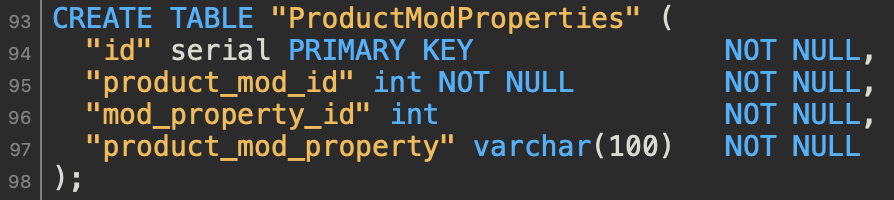
* Код для создания таблицы Users



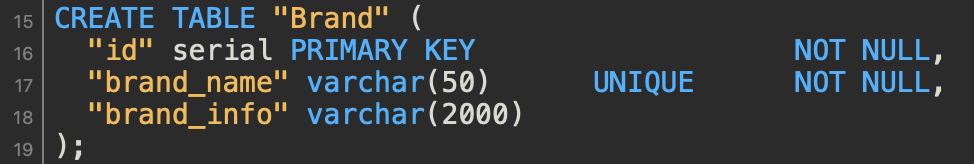
* Код для создания таблицы ProductReview



* Код для создания таблицы ProductModProperties



* Код для создания таблицы Brand



1. Заполнение базы данных
   1. Парсинг данных

На данный момент не существует русской версия E-Katalog, поэтому заполнение базы данных происходило из казахстанской версии сайта E-Katalog (kz.e-katalog.com).

Для сбора необходимой информации с данного сайта был написан парсер на языке программирования Python. В качестве библиотек для парсинга были выбраны такие библиотеки как:

* BeautifulSoup
* Requests
* Pandas (использовалась исключительно ввиду удобных контейнеров информации, таких как Pandas.DataFrame)

Для парсинга данных был написан класс Parser, который позволяет получать необходимое количество товаров заданного вида. В итоговый датафрейм записывались такие параметры, как:

* название товара
* оценки пользователей
* основные характеристики (7 полей)
* модификации товара

После окончания работы алгоритма, парсер сохранял полученные данные в .csv формат, откуда в дальнейшем и считывались данные для заполнения базы данных.

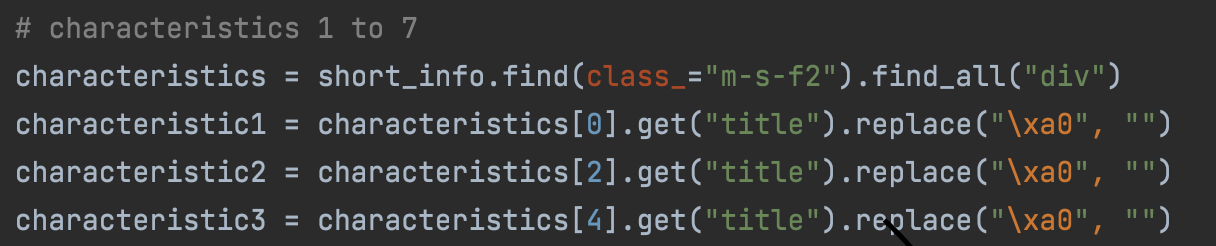
Стоит отметить, что далеко не все данные получилось достать с сайта. Например, получить список товар-магазин-цена не удалось, так как Javascript загружает подобные данные уже после того, как загрузилась основная страница.

В связи с этим, данные о магазинах был взяты с других конкретных сайтов (в основном с ЯндексМаркета).

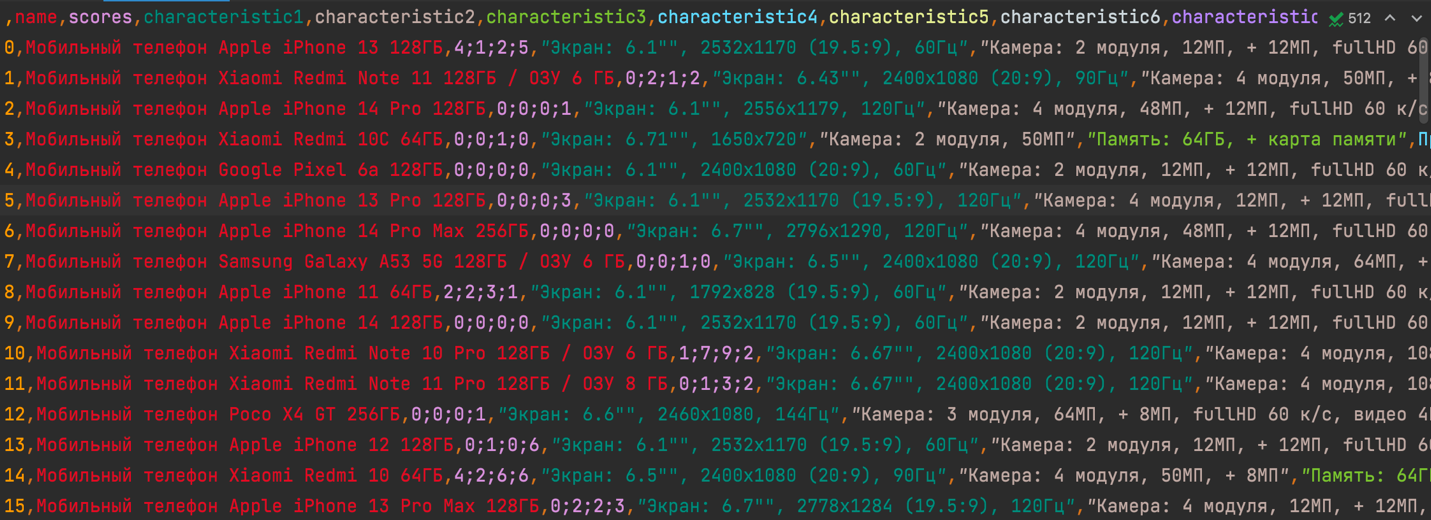
Ниже приведены скриншоты кода парсера и итогового .csv файла.



Парсинг характеристик:



Итоговый .csv файл



* 1. Итоговое заполнение базы данных

Для более удобной работы с базой данных был написан на языке программирования Python класс Database. Он позволяет открывать соединение с базой данных, закрывать его, записывать строки в нужную таблицу конкретной базы данных, а также автоматически заполнять БД данными, которые были получены в предыдущей главе.

Такие таблицы, как Users, ShopReviews и ProductReviews были заполнены рандомными значениями, удовлетворяющими ограничениям, прописанным в базе данных.

Ниже приведен кусок кода с данным классом.



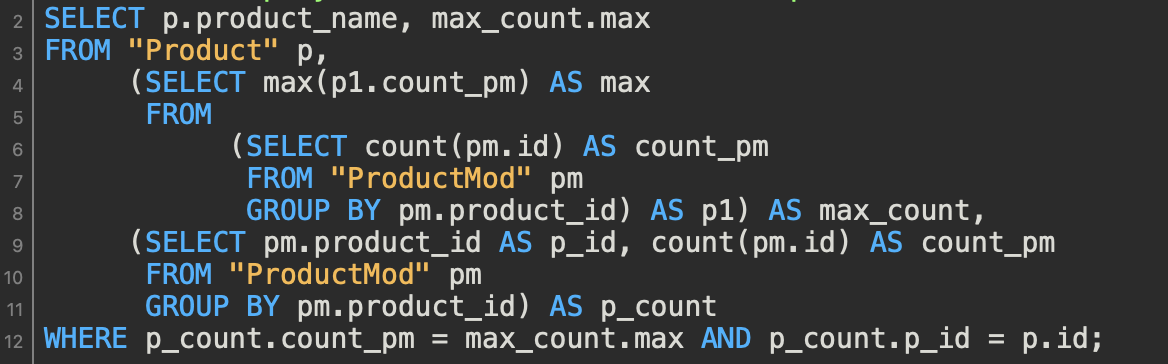
Функция записи всех магазинов в таблицу Shops.

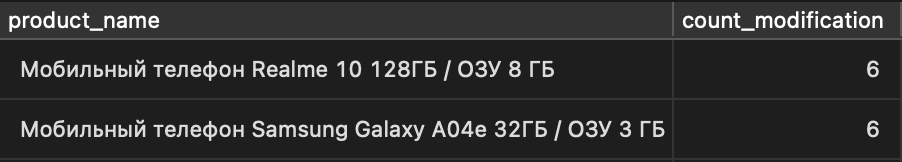


Функция записи всех брендов



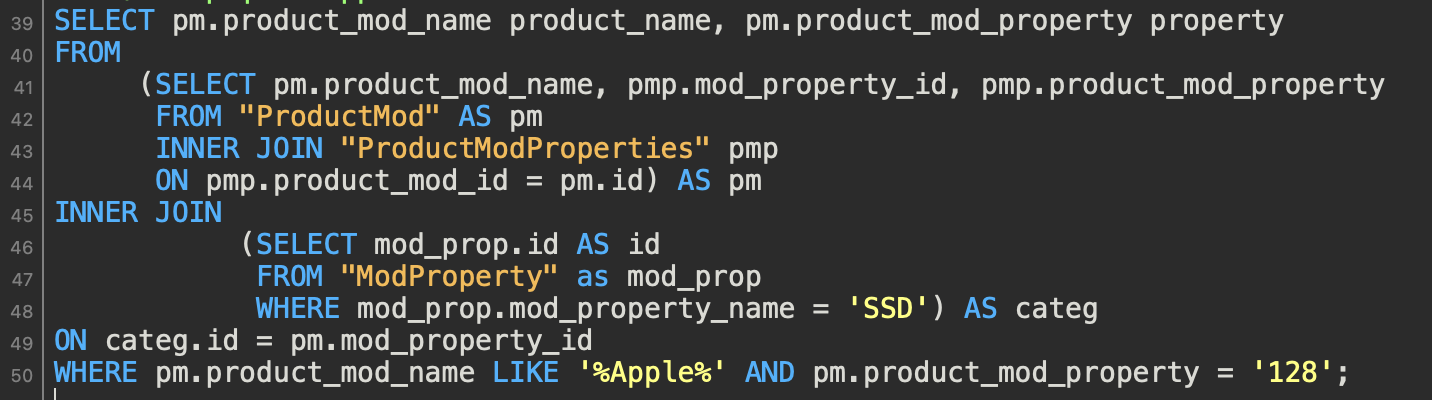
1. Запросы к базе данных
2. Задача: вывести товары, которые имеют наибольшее число модификаций.



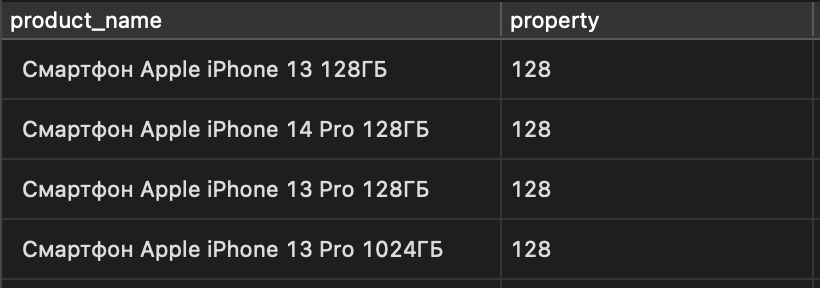
Результат выполнения:

1. Задача: вывести товары Apple с определенным объемом SSD (в данном случае 128 гб.)

Необходима для поиска товара с определенной характеристикой. Запрос легко расширяется для поиска по нескольким характеристикам, что позволяет сделать модель представления характеристик EAV.

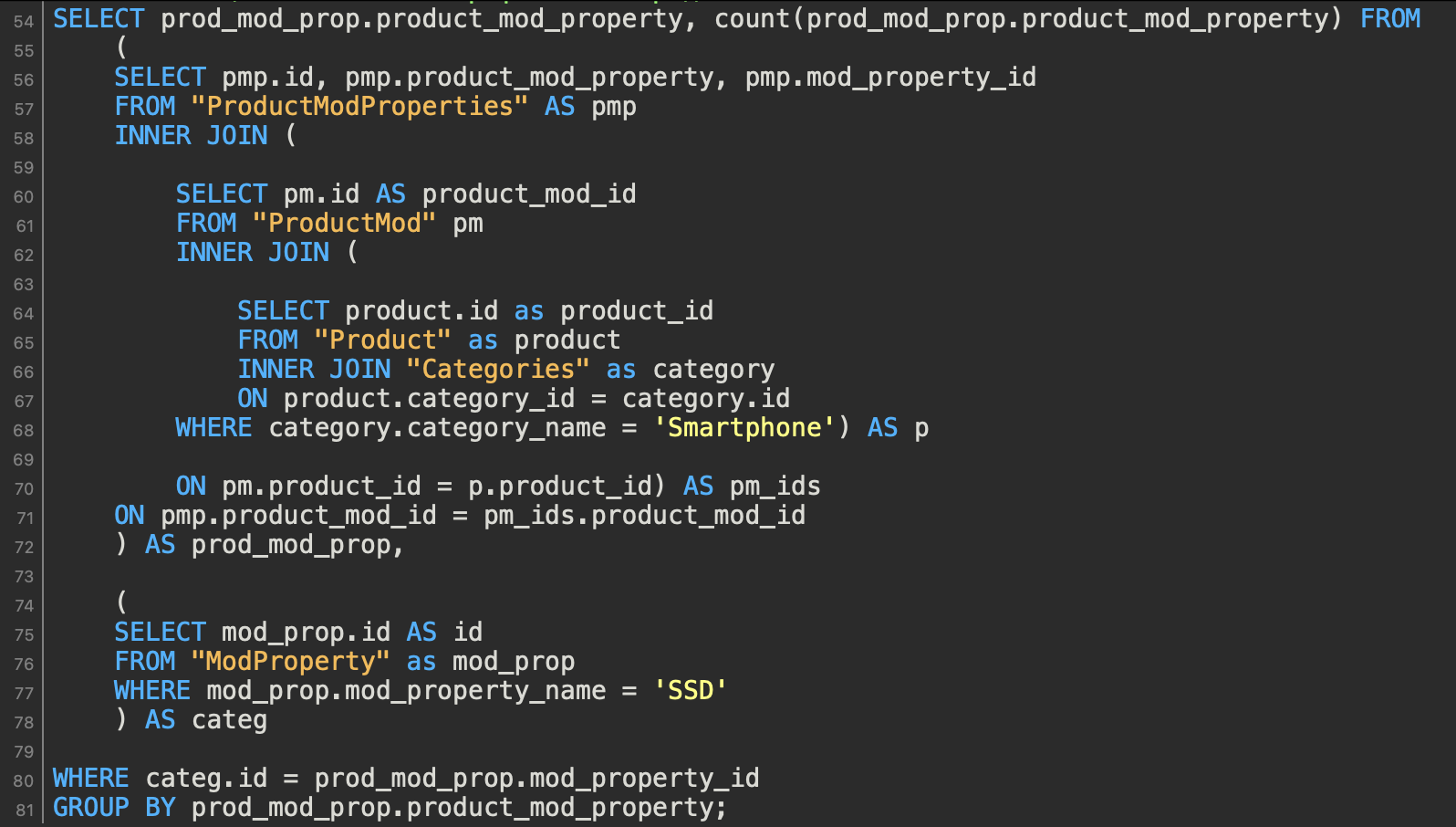


Результат выполнения:



1. Задача: вывести все товары категории “Smartphone” с определенным объемом SSD (в данном случае 128 гб.)

Необходима для поиска товара с определенной характеристикой. Запрос легко расширяется для поиска по нескольким характеристикам, что позволяет сделать модель представления характеристик EAV.

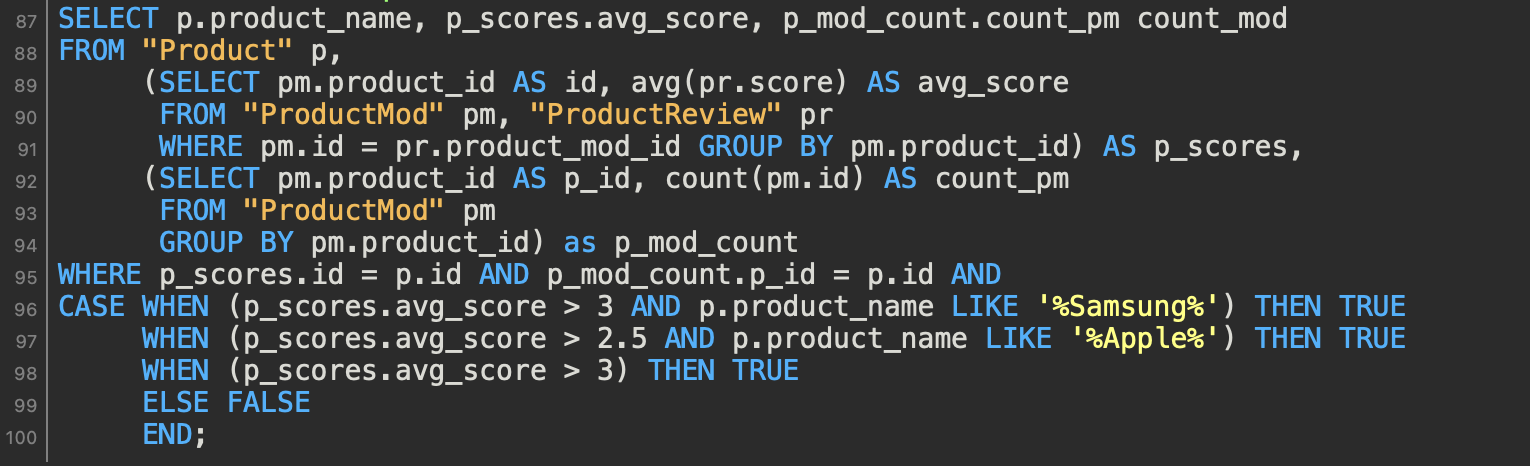


Результат выполнения:

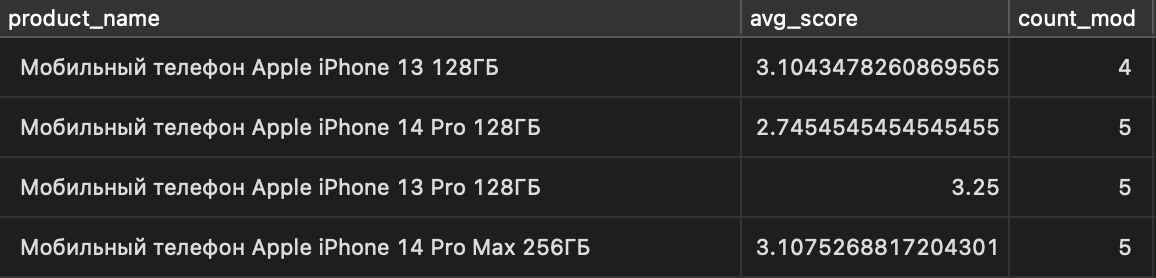


1. Задача: вывести все “удачные” модели товаров, а также кол-во модификаций. Под для каждого бренда “удачная” модель понимается по-разному: для Apple оценка пользователей должна быть, например, больше 3.0, а для какой-нибудь Sony оценка 2.5 уже является показателем удачной модели.

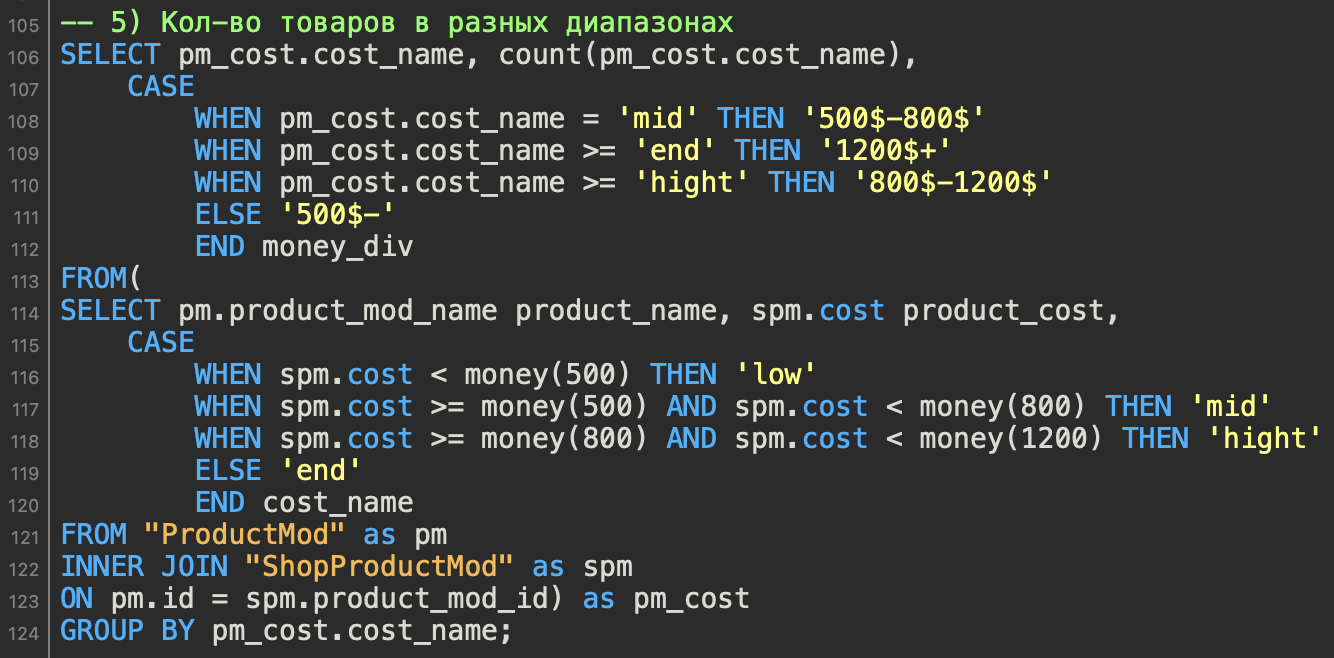
Может применяться для сбора статистики по товарам для каждого бренда, а также оценивать, как бренд подходить к вопросу обновления своих культовых моделей



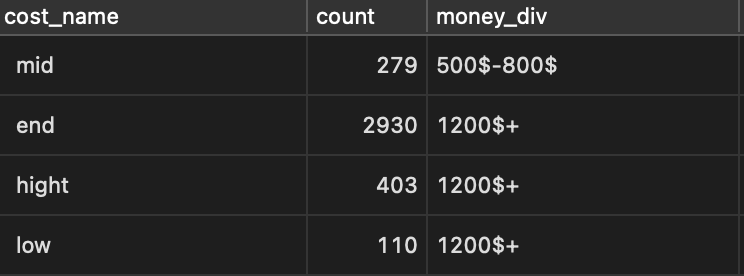
Результат выполнения:



1. Задача: оценить кол-во товаров в разных ценовых категориях.

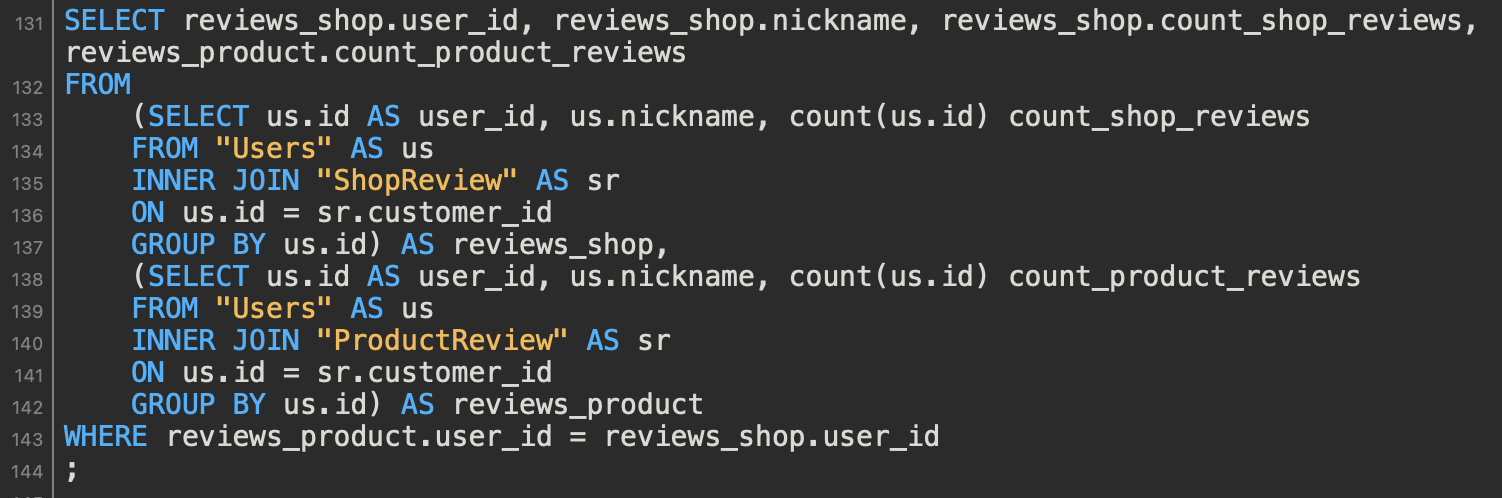
Может понадобится для оценки, какой ценовой сектор перенасыщен или наоборот – товаров слишком мало.

Результат выполнения:



1. Задача: найти для каждого пользователя кол-во его оценок для магазинов и для товаров.

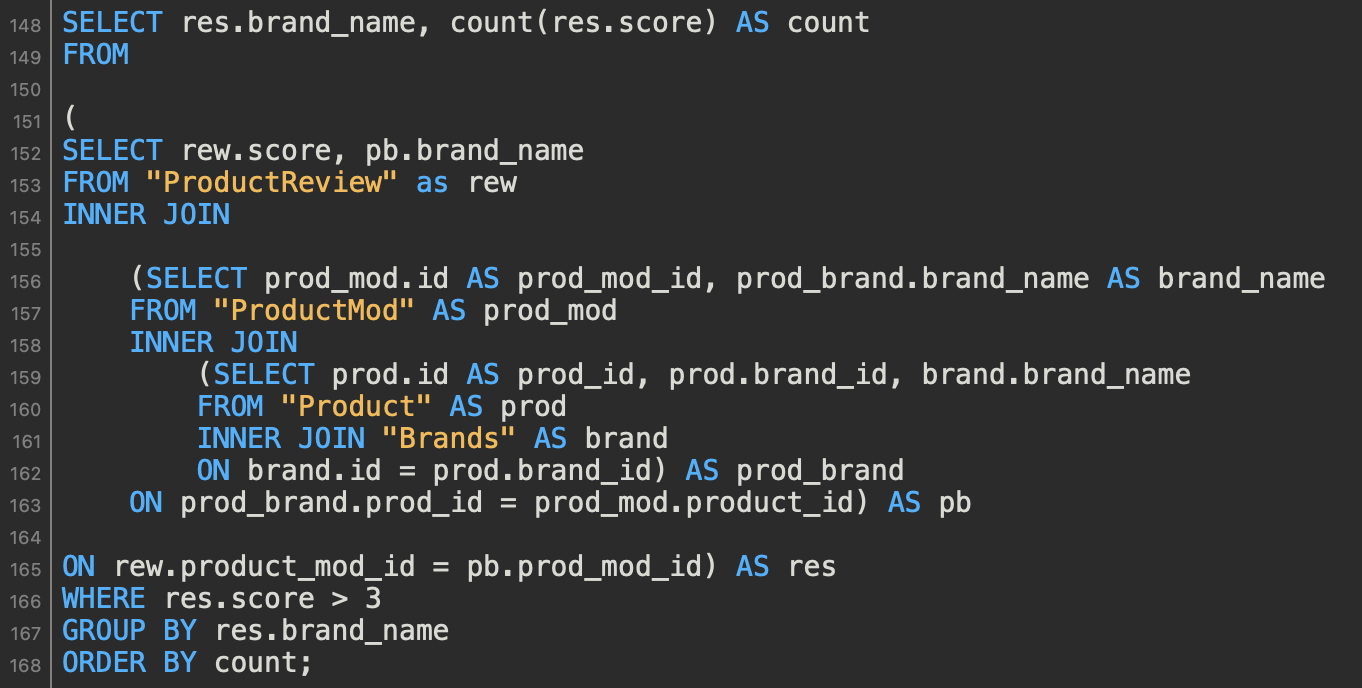
Необходимо для оценки активности профиля



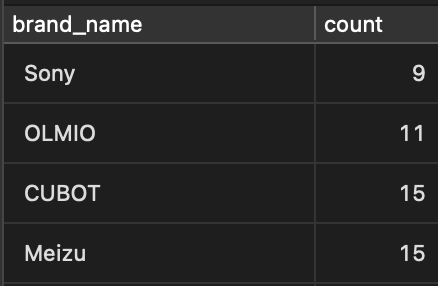
Результат выполнения:



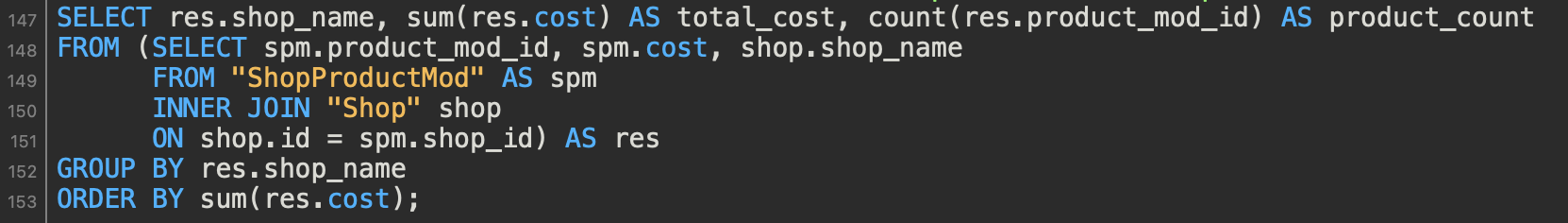
1. Задача: найти популярность брендов по кол-ву хороших товаров (т.е. по кол-ву товаров, рейтинг которых больше конкретного значения)



Результат выполнения:



1. Задача: для каждого магазина найти общую стоимость его товаров и кол-во товаров, представленных в нем.



Результат выполнения:

