

Содержание

Введение	4
1 Аналитический раздел	6
1.1 Формализация данных	6
1.2 Описание ролевой модели	6
1.3 Обзор существующих СУБД	7
1.3.1 Классификация СУБД по модели данных	7
1.3.2 Обзор объектно-реляционных СУБД	9
1.3.3 Выбор СУБД для решения задачи	11
1.4 Методы построения линии тренда	11
1.4.1 Линейная линия тренда	12
1.4.2 Логарифмическая линия тренда	12
1.4.3 Полиномиальная линия тренда	12
1.4.4 Экспоненциальная линия тренда	13
1.4.5 Степенная линия тренда	13
1.4.6 Линейная фильтрация	13
1.4.7 Выбор метода построения линии тренда	13
Вывод к аналитическому разделу	14
2 Конструкторский раздел	15
2.1 Проектирование базы данных	15
Вывод к конструкторскому разделу	17
3 Технологический раздел	18
Вывод к технологическому разделу	18
Заключение	19
Список литературы	20

Введение

Методы прогнозирования многообразны, как и объекты, прогнозированием которых занимается человек. Ему всегда было интересно узнать будущее – свое, своих близких, государства, экономики, предугадать погоду, определить как изменятся в ближайшее время курсы валют и т. п.

Но прогнозировать ситуацию важно не только из-за того, что это интересно, но и потому, что от предвидения будущего зависят действия человека. Если прогнозируется скачок цен на какой-то товар, то можно заранее закупить его. Можно сказать, что жизнь современного человека невозможно себе представить без прогнозирования.

Прогноз – это результат индуктивного вывода, когда по характеру ограниченного множества значений показателей или взаимосвязи факторов делается вывод о том, что и остальные, еще не наблюдаемые значения этих показателей или взаимосвязи будут обладать аналогичными свойствами [1].

В мировой экономической науке накоплен и апробирован значительный арсенал методов прогнозирования, который дает возможность решать комплекс задач по обоснованию решений в различных областях [2].

Задача прогнозирования цен на потребительские товары актуальна для различных типов пользователей: во-первых, это конечные покупатели, которые принимают решение о покупке того или иного товара; прогноз цены может повлиять не только на выбор момента покупки, но и на сам факт покупки товара как таковой; во-вторых, это владельцы магазинов (необязательно онлайн), которые планируют закупки и ассортимент товаров; в-третьих, маркетологи, формирующие аналитику изменения рынка тех или иных товаров [3].

Цель работы – реализовать базу данных, хранящую информацию о покупателях, магазинах, ассортименте товаров в магазинах и историях цен товаров в них, и программное обеспечение для работы с информацией из этой базы данных, а также прогнозирования цен на товары в магазинах посредством построения линии тренда на основе истории цен.

Таким образом, необходимо решить следующие задачи:

- проанализировать варианты представления данных и выбрать из них подходящий для решения задачи;

- проанализировать системы управления базами данных и выбрать подходящую систему для хранения данных;
- проанализировать методы построения линии тренда и выбрать подходящий метод для прогнозирования цен;
- спроектировать базу данных, описать её сущности и связи;
- реализовать интерфейс для работы с базой данных;
- реализовать возможность построения линии тренда для прогнозирования цен на товары в магазинах.

1 | Аналитический раздел

В данном разделе формализованы данные, хранимые в базе данных, описана ролевая модель, описаны существующие системы управления базами данных и методы построения линии тренда.

1.1 Формализация данных

База данных должна хранить информацию о:

- магазинах;
- товарах
- ассортименте товаров в магазинах;
- историю цен на товары в магазинах за последние полтора года;
- товарные чеки.

Таблица 1.1: Категории и описание данных

Категория	Описание
Магазин	Название, описание магазина, ассортимент товаров
Товар	Название, тип товара, актуальная цена
Ассортимент товаров	Список товаров
История цен	Год, месяц, цена в этот период
Товарный чек	ФИО, дата покупки, информация о месте покупки(магазине), список купленных товаров, итоговая стоимость

1.2 Описание ролевой модели

Для управления системой введено три роли: пользователь, аналитик, администратор.

Таблица 1.2: Роли и описание их функционала

Роль	Функционал
Пользователь	Просмотр информации о магазинах, их ассортименте товаров и их актуальной стоимости
Аналитик	Просмотр информации о магазинах, их ассортименте товаров и их истории цен. Возможность прогнозирования цены на товар в магазине
Администратор	Просмотр, добавление и удаление информации о магазинах, их ассортименте товаров и их истории цен, товарных чеках. Возможность прогнозирования цены на товар в магазине

Для доступа к ролям «Аналитик» и «Администратор» требуется авторизация (ввод пароля).

1.3 Обзор существующих СУБД

Система управления базами данных (сокр. СУБД) – Программная система, предназначенная для создания и хранения базы данных на основе некоторой модели данных, обеспечения логической и физической целостности содержащихся в ней данных, надежного и эффективного использования ресурсов (данных, пространства памяти и вычислительных ресурсов), предоставления к ней санкционированного доступа для приложений и конечных пользователей, а также для поддержки функций администратора базы данных [4].

Функции СУБД:

- управление данными во внешней памяти;
- управление данными в оперативной памяти с использованием дискового кэша;
- журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД.

1.3.1 Классификация СУБД по модели данных

Модель данных – система типов данных, типов связей между ними и допустимых видов ограничений целостности, которые могут быть для них определены. Здесь имеется в виду современное понимание типа данных как носителя свойств, определяющих и состояние экземпляров типа, и их поведение [4].

По этому признаку СУБД делят на:

- иерархические;
- сетевые;
- реляционные;
- объектно-ориентированные;
- объектно-реляционные.

Иерархические СУБД

В иерархических СУБД используется представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок. Объекты, имеющие общего предка, называются близнецами [5].

Примеры: Caché, Google App Engine Datastore API.

Сетевые СУБД

Сетевые СУБД подобны иерархическим, за исключением того, что в них имеются указатели в обоих направлениях, которые соединяют родственную информацию.

Несмотря на то, что эта модель решает некоторые проблемы, связанные с иерархической моделью, выполнение простых запросов остается достаточно сложным процессом.

Также, поскольку логика процедуры выборки данных зависит от физической организации этих данных, то эта модель не является полностью независимой от приложения. Другими словами, если необходимо изменить структуру данных, то нужно изменить и приложение [5].

Примеры: Caché.

Реляционные СУБД

Реляционные СУБД ориентированы на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:

1. каждый элемент таблицы является одним элементом данных;

2. каждый столбец обладает своим уникальным именем;
3. одинаковые строки в таблице отсутствуют;
4. все столбцы в таблице однородные, то есть все элементы в столбце имеют одинаковый тип;
5. порядок следования строк и столбцов может быть произвольным.

Практически все разработчики современных приложений, предусматривающих связь с системами баз данных, ориентируются на реляционные СУБД. По оценке Gartner в 2013 году рынок реляционных СУБД составлял 26 млрд долларов с годовым приростом около 9%, а к 2018 году рынок реляционных СУБД достигнет 40 млрд долларов. В настоящее время абсолютными лидерами рынка СУБД являются компании Oracle, IBM и Microsoft, с общей совокупной долей рынка около 90%, поставляя такие системы как Oracle Database, IBM DB2 и Microsoft SQL Server [6].

Объектно-ориентированные СУБД

Объектно-ориентированные СУБД управляют базами данных, в которых данные моделируются в виде объектов, их атрибутов, методов и классов.

Этот вид СУБД позволяет работать с объектами баз данных так же, как с объектами в программировании в объектно-ориентированных языках программирования. ООСУБД расширяет языки программирования, прозрачно вводя долговременные данные, управление параллелизмом, восстановление данных, ассоциированные запросы и другие возможности [6].

Примеры: GemStone.

Объектно-реляционные СУБД

Объектно-реляционные СУБД поддерживают некоторые технологии, реализующие объектно-ориентированный подход: объекты, классы и наследование реализованы в структуре баз данных и языке запросов.

Зачастую все те СУБД, которые называются реляционными, являются, по факту, объектно-реляционными [6].

В данной работе будет рассмотрен именно этот класс СУБД.

Примеры: PostgreSQL, DB2, Oracle Database, Microsoft SQL Server.

1.3.2 Обзор объектно-реляционных СУБД

Oracle Database

Oracle Database [7] – объектно-реляционная система управления базами данных компании Oracle [8].

На данный момент у этого продукта блестящая репутация [9]. Кроме того, существует несколько версий этого продукта для удовлетворения потребностей конкретной организации.

Актуальная версия Oracle на данный момент предназначена для облачных сред и может быть размещена на одном или нескольких серверах, это позволяет управлять базами данных, которые содержат миллиарды записей. Некоторые из функций новейшей версии Oracle включают в себя grid framework и использования как физических, так и логических структур.

Это означает, что физическое управление данными не влияет на доступ к логическим структурам. Кроме того, безопасность в этой версии доведена до высочайшего уровня, потому что каждая транзакция изолирована от других.

Microsoft SQL Server

Microsoft SQL Server [10] – система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft.

Microsoft SQL Server является ещё одной из популярных СУБД [9]. Это система управления базами данных, движок которой работает на облачных серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно. Вскоре после выпуска Microsoft SQL Server 2016, Microsoft адаптировала продукт для операционной системы Linux, а на платформе Windows он работал изначально.

Одной из уникальных особенностей версии 2016 года является temporal data support (временная поддержка данных), которая позволяет отслеживать изменения данных с течением времени. Последняя версия Microsoft SQL-сервер поддерживает dynamic data masking (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные.

PostgreSQL

PostgreSQL [11] – свободная объектно-реляционная система управления базами данных (СУБД).

PostgreSQL является одним из нескольких бесплатных популярных вариантов СУБД, часто используется для ведения баз данных веб-сайтов [9]. Это весьма старая система, поэтому в настоящее время она хорошо развита, и позволяет пользователям управлять как структурированными, так и неструктурированными данными. Может быть использована на большинстве основных платформ, включая Linux (где особенно хорошо проявляется производительность). Прекрасно справляется с задачами импорта информации из других типов баз данных с помощью собственного инструментария.

Движок БД может быть размещен в ряде сред, в том числе виртуальных, физических и облачных. Последняя версия PostgreSQL предлагает об-

работку больших объемов данных и увеличение числа одновременно работающих пользователей. Безопасность была улучшена благодаря поддержке DBMS_SESSION.

DB2

DB2 [14] – семейство систем управления реляционными базами данных, выпускаемых корпорацией IBM. Чаще всего, ссылаясь на DB2, имеют в виду реляционную систему управления базами данных DB2 Universal Database (DB2 UDB).

DB2 представляет собой СУБД, которая имеет возможности NoSQL, и может читать JSON и XML-файлы [9]. Ввиду того, что система разрабатывалась для серверов компании IBM модельного ряда iSeries, логично, что система работает на Windows, Linux и MacOS.

Диалект языка SQL, используемый в DB2 за редкими исключениями строго декларативен, система снабжена многофазовым оптимизатором, строящим по этим декларативным конструкциям план выполнения запроса.

Оптимизатор DB2 широко использует статистику распределения данных в таблицах (если процесс её сбора был выполнен администратором базы данных), поэтому один и тот же запрос на языке SQL может быть оттранслирован в совершенно различные планы выполнения в зависимости от статистических характеристик данных, которые он обрабатывает.

Одно из улучшений DB2 – Blu Acceleration, которое предназначено для ускорения работы базы данных. Пропуск данных предназначен для повышения быстродействия системы с большим количеством данных, чем может она вместить в себя. Последняя версия DB2 также обеспечивает усовершенствованные функции аварийного восстановления, совместимости и аналитики.

1.3.3 Выбор СУБД для решения задачи

Для решения задачи была выбрана СУБД PostgreSQL, т. к. она открытая, кроссплатформенная, имеет поддержку БД неограниченного размера, мощные и надёжные механизмы транзакций и репликации, систему встроенных языков программирования (PL/pgSQL [12], plpython3u [13]), а также легкую расширяемость.

1.4 Методы построения линии тренда

Линия тренда – прямая или кривая линия, аппроксимирующая (приближающая) исходные данные на основе уравнения регрессии или скользящего

среднего [15]. Аппроксимация определяется по методу наименьших квадратов [16]. В зависимости от характера поведения исходных данных (убывают, возрастают и т.д.) выбирается метод интерполяции, который следует использовать для построения тренда.

1.4.1 Линейная линия тренда

Линейная линия тренда определяется функцией

$$y = ax + b, \quad (1.1)$$

где a – тангенс угла наклона прямой, b – смещение.

Прямая линия тренда (линейный тренд) наилучшим образом подходит для величин, изменяющихся с постоянной скоростью. Применяется в случаях, когда точки данных расположены близко к прямой [15].

1.4.2 Логарифмическая линия тренда

Логарифмическая линия тренда определяется функцией

$$y = a \ln x + b, \quad (1.2)$$

где a и b – константы.

Логарифмическая линия тренда соответствует ряду данных, значения которого вначале быстро растут или убывают, а затем постепенно стабилизируются. Может использоваться для положительных и отрицательных данных [15].

1.4.3 Полиномиальная линия тренда

Полиномиальная линия тренда определяется функцией

$$y = \sum_{i=0}^n a_i x^i, n \leq 6, \quad (1.3)$$

где a_i – коэффициенты полинома.

Полиномиальная линия тренда используется для описания попеременно возрастающих и убывающих данных. Степень полинома подбирают таким образом, чтобы она была на единицу больше количества экстремумов (максимумов и минимумов) кривой [15].

1.4.4 Экспоненциальная линия тренда

Экспоненциальная линия тренда определяется функцией

$$y = ae^{bx}, \quad (1.4)$$

где a и b – константы.

Экспоненциальный тренд используется в случае непрерывного возрастания изменения данных. Построение указанного тренда невозможно, если в множестве значений членов ряда присутствуют нулевые или отрицательные данные [15].

1.4.5 Степенная линия тренда

Степенная линия тренда определяется функцией

$$y = ax^b, \quad (1.5)$$

где a и b – константы.

Степенная линия тренда дает хорошие результаты для положительных данных с постоянным ускорением. Для рядов с нулевыми или отрицательными значениями построение указанной линии тренда невозможно [15].

1.4.6 Линейная фильтрация

Линейная фильтрация определяется формулой

$$F_t = \frac{A_t + A_{t-1} + \dots + A_{t-n+1}}{n}, \quad (2 \leq t < n), \quad (1.6)$$

где n – общее число членов ряда, t – параметр фильтра.

Тренд с линейной фильтрацией позволяет сгладить колебания данных, наглядно демонстрируя характер зависимостей [15]. Для построения указанной линии тренда пользователь должен задать число t .

Линейная фильтрация использует метод скользящего среднего, при котором каждое значение функции заменяется средним арифметическим A по t соседним точкам, расположенным симметрично относительно данной. Линия тренда в этом случае уравнения не имеет [17], поэтому для прогноза изменения величины она не подходит.

1.4.7 Выбор метода построения линии тренда

Цена товара как величина положительная и имеет непостоянный рост: она может как расти, так и падать. Таким образом была выбрана полиномиальная линия тренда для прогнозирования цен на товары.

Вывод к аналитическому разделу

В данном разделе были формализованы данные, хранимые в базе данных, описана ролевая модель, рассмотрены существующие СУБД и методы построения линии тренда.

Для решения задачи были выбраны PostgreSQL в качестве СУБД и полиномиальная линия тренда для прогнозирования цен на товары.

2 | Конструкторский раздел

2.1 Проектирование базы данных

ER-диаграмма сущностей базы данных представлена на рисунке 2.1.

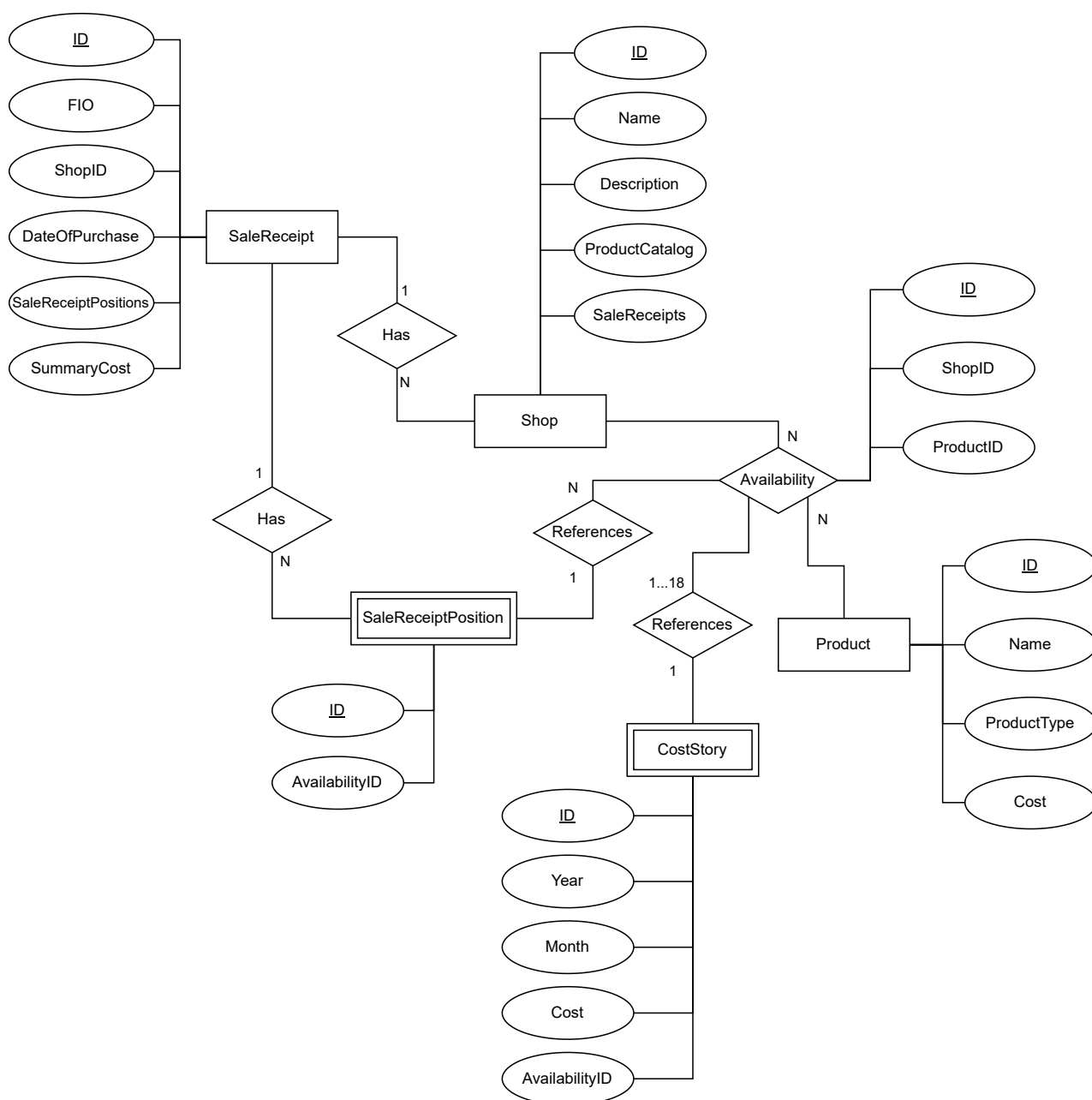


Рис. 2.1: ER-диаграмма сущностей базы данных в нотации Чена.

В базе данных будет существовать 6 таблиц, одна из которых является развязочной (Availability). Остальные таблицы на диаграмме представлены в виде сущностей.

Таблица **SaleReceipt** хранит данные о товарных чеках и содержит следующие поля:

- ID – идентификатор товарного чека;
- FIO – ФИО покупателя;
- ShopID – идентификатор магазина, в котором была совершена покупка;
- DateOfPurchase – дата покупки;
- SaleReceiptPositions – позиции товарного чека;
- SummaryCost – суммарная стоимость в чеке.

Таблица **Shop** хранит данные о магазинах и содержит следующие поля:

- ID – идентификатор магазина;
- Name – название магазина;
- Description – описание магазина;
- ProductCatalog – каталог продуктов;
- SaleReceipts – товарные чеки.

Таблица **Product** хранит данные о продуктах и содержит следующие поля:

- ID – идентификатор продукта;
- Name – название продукта;
- ProductType – тип продукта.

Таблица **Availability** хранит отношения наличия товара в каталоге магазина и содержит следующие поля:

- ID – идентификатор отношения;
- ShopID – идентификатор магазина;
- ProductID – идентификатор продукта.

Таблица **CostStory** хранит историю цен на товары в магазинах и содержит следующие поля:

- ID – идентификатор значения цены в конкретный год и месяц;
- Year – год;
- Month – месяц;
- Cost – значение цены в конкретный год и месяц;
- AvailabilityID – идентификатор отношения наличия товара в каталоге магазина.

Таблица **SaleReceiptPosition** хранит информацию о позиции в чеке и содержит следующие поля:

- ID – идентификатор позиции чека;
- AvailabilityID – идентификатор отношения наличия товара в каталоге магазина.

Таблицы **SaleReceiptPosition** и **CostStory** содержат поле идентификатора отношения наличия товара в каталоге магазина в целях борьбы с рассинхронизацией: товарный чек и история цен на товар в магазине не могут ссылаться на товары, которых нет в данном магазине.

Актуальная стоимость товара в магазине формируется как последнее по дате значение цены товара в истории цен.

На рисунке 2.2 представлена Use-case-диаграмма ПО.

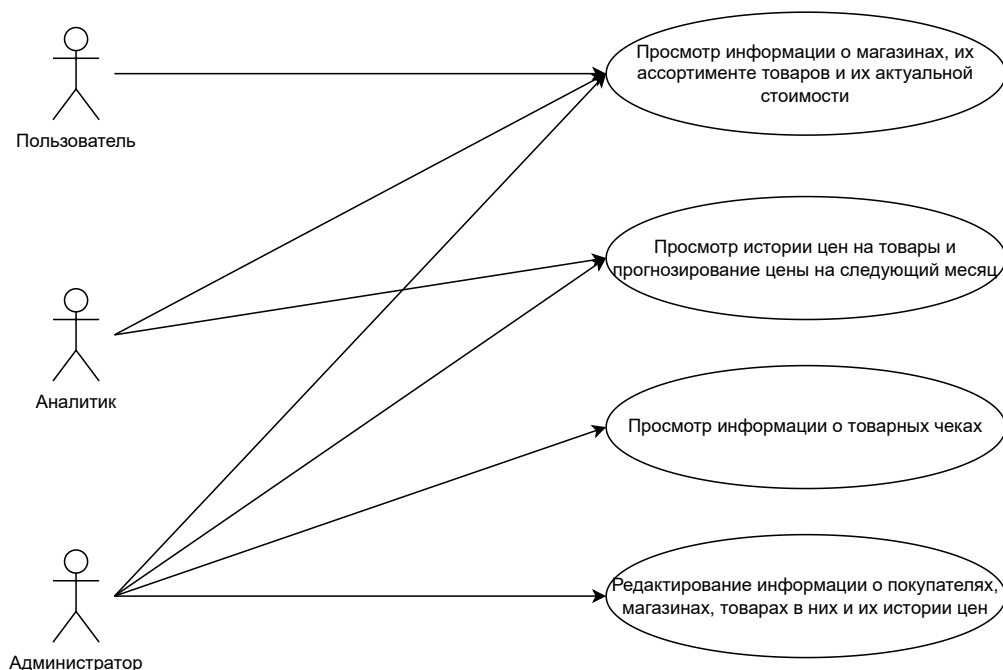


Рис. 2.2: Use-case-диаграмма ПО.

Вывод к конструкторскому разделу

3 | Технологический раздел

Вывод к технологическому разделу

Заключение

Список литературы

- [1] Методы и модели социально-экономического прогнозирования : учебник и практикум для академического бакалавриата. В 2-х т. Т. 1. Теория и методология прогнозирования / И. С. Светульников, С. Г. Светульников. — М. : Издательство Юрайт, 2014. — 351 с. — Серия : Бакалавр. Академический курс.
- [2] Касперович С.А. Прогнозирование и планирование экономики : курс лекций для студентов специальностей 1-25 01 07 «Экономика и управление предприятием», 1-25 01 08 «Бухгалтерский учет, анализ и аудит», 1-26 02 02 «Менеджмент», 1-26 02 03 «Маркетинг» / С. А. Касперович. - Минск. : БГТУ, 2007. - 172 с.
- [3] К. В. Богданов. Методика прогнозирования цен на товары в интернет-магазинах [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/metodika-prognozirovaniya-tsen-na-tovary-v-internet-magazinah/viewer> (дата обращения 4.04.2022)
- [4] Когаловский М.Р. Энциклопедия технологий баз данных. — М.: Финансы и статистика, 2002. — 800 с.
- [5] Классификация систем управления базами данных [Электронный ресурс]. — Режим доступа: <https://scienceforum.ru/2016/article/2016019197> (дата обращения 6.04.2022)
- [6] Системы управления базами данных [Электронный ресурс]. — Режим доступа: <https://lecturesdb.readthedocs.io/databases/dbms.html> (дата обращения 8.04.2022)
- [7] Oracle Database [Электронный ресурс]. — Режим доступа: <https://www.oracle.com/database/> (дата обращения 8.04.2022)
- [8] Oracle [Электронный ресурс]. — Режим доступа: <https://www.oracle.com/> (дата обращения 8.04.2022)
- [9] Сравнение современных СУБД [Электронный ресурс]. — Режим доступа: <https://drach.pro/blog/hi-tech/item/145-db-comparison> (дата обращения 8.04.2022)

- [10] Microsoft SQL-Server [Электронный ресурс]. – Режим доступа: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2019> (дата обращения 8.04.2022)
- [11] PostgreSQL [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/> (дата обращения 8.04.2022)
- [12] PL/pgSQL [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/docs/9.6/plpgsql.html> (дата обращения 8.04.2022)
- [13] PL/Python [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/docs/current/plpython-python23.html> (дата обращения 8.04.2022)
- [14] DB2 [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/analytics/db2> (дата обращения 8.04.2022)
- [15] Построение линии тренда в Excel [Электронный ресурс]. – Режим доступа: <https://ya-znau.ru/znaniya/zn/202> (дата обращения 9.04.2022)
- [16] Линник Ю. В. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений. — 2-е изд. — М., 1962. (математическая теория)
- [17] Линейная фильтрация [Электронный ресурс]. – Режим доступа: <https://poznayka.org/s23136t1.html> (дата обращения 9.04.2022)