**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И   
МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ   
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ   
«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ   
СВЯЗИ И ИНФОРМАТИКИ» (МТУСИ)**

Изображение выглядит как эмблема, текст, символ, логотип

Автоматически созданное описание

**Кафедра «Информатика»**

**Дисциплина: Базы данных**

**Отчет по курсовой работе на тему:**

**Разработка корпоративного сайта организации  
«Молока долина»**

Выполнил: студент гр. БЭИ2202

Васильев В. А.

Проверил:

ст. пр. каф. «Информатика»

Юсков И.О.;

доц. каф. «Информатика»

Шакин В.Н.

Москва, 2023

Оглавление

[ВВЕДЕНИЕ 3](#_Toc153827788)

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 6](#_Toc153827789)

[ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 10](#_Toc153827790)

[1.1 Принципы создания базы данных в рамках организации веб-сайта компании «Молока долина» 10](#_Toc153827791)

[1.2 Выбор системы управления базами данных (СУБД) 11](#_Toc153827792)

[1.3 Разработка схемы базы данных 12](#_Toc153827793)

[1.4 Проблемы проектирования баз данных и способы их решения 13](#_Toc153827794)

[1.5 Аналитика данных в организации 15](#_Toc153827795)

[1.6 Использование данных для стратегического планирования 15](#_Toc153827796)

[1.7 Техническая поддержка и обслуживание базы данных 16](#_Toc153827797)

[ГЛАВА 2. СОЗДАНИЕ БАЗЫ ДАННЫХ 18](#_Toc153827798)

[2.1 ER модель базы данных 18](#_Toc153827799)

[2.2 Структура базы данных 18](#_Toc153827800)

[ГЛАВА 3. ПОДКЛЮЧЕНИЕ БД К САЙТУ. 22](#_Toc153827801)

[3.1 Структура проекта 22](#_Toc153827802)

[3.2 Программная реализация 23](#_Toc153827803)

[ЗАКЛЮЧЕНИЕ 32](#_Toc153827804)

[СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ 33](#_Toc153827805)

# ВВЕДЕНИЕ

На веб-сайтах базы данных являются основным хранилищем информации, необходимым для различных функциональных задач. В случае веб-сайтов, таких как «Молока долина», базы данных выполняют ряд важных функций:

Хранение информации: базы данных служат хранилищем для всех данных, которые использует сайт, будь то информация о пользователях, описание товаров, картинки и т. д.

Обеспечение доступа: базы данных обеспечивают доступ к нужным данным для того, чтобы отвечать на запросы пользователей и предоставлять информацию о продуктах и услугах.

Управление информацией: базы данных облегчают управление информацией, позволяя быстро и легко добавлять, изменять и удалять данные в случае необходимости.

Безопасность: базы данных используются для хранения данных пользователей и прав доступа, что обеспечивает безопасность данных клиентов и защиту операций.

Анализ данных: данные, хранящиеся в базах данных, можно анализировать, чтобы определить тенденции покупок и собрать статистику для развития и роста компании. Это помогает улучшить сайт и сделать его работу более эффективной.

С увеличением важности эффективного управления данными в сфере производства еды и напитков, использование баз данных на веб-сайтах становится все более актуальным.

**Цели исследования**

Целью данной работы является изучение архитектуры баз данных и программной реализации подключения базы данных для веб-сайта организации «Молока долина».

**Задачи исследования:**

В ходе исследования будут рассмотрены следующие аспекты:

* Изучение основных принципов архитектуры баз данных.
* Рассмотрение современных тенденций в области баз данных.
* Анализ существующих решений для веб-приложений в сфере производства молочных продуктов, выявление ключевых аспектов для разработчиков, инженеров и администраторов таких систем.
* Выделение ключевых аспектов для разработки и поддержки баз данных.

**Актуальность**

Тема баз данных актуальна для молочного завода, поскольку они являются важным инструментом для хранения и управления информацией о продуктах, клиентах и производственных процессах.

Актуальность базы данных для молочного завода обуславливается следующим:

* База данных позволяет хранить всю информацию о продукции, включая состав, сроки годности, сертификаты качества и т. д. Это облегчает процесс контроля качества продукции и предотвращает возможные ошибки.
* Базы данных позволяют отслеживать количество и состояние товаров на складе, а также контролировать процесс производства и доставки продукции.
* Базы данных могут содержать информацию о покупателях, их предпочтениях и истории покупок. Это позволяет проводить более точные маркетинговые кампании и улучшать качество обслуживания клиентов.
* Базы данных позволяют анализировать данные о продажах, маркетинговых активностях, эффективности производства и т. д., что помогает принимать обоснованные решения и оптимизировать бизнес-процессы.

**Объект исследования**

Объектом исследования являются информационная инфраструктура и база данных компании «Молока долина», которые поддерживают работу ее веб-сайта. Сюда входит структура данных, которые хранят информацию о продукции, клиентах, заказах, отзывах и других аспектах работы компании.

**Предмет исследования**

Предметом исследования является процесс создания, управления и улучшения архитектуры баз данных для сайта компании «Молока долина». Это включает в себя выбор оптимальной модели данных, создание структуры базы данных, написание запросов и хранимых процедур для эффективного взаимодействия с данными.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**1. Введение**

Данный программный продукт создан для проверки качества усвоения студентами базовых определений, принципов и аспектов дисциплины «базы данных». Оценка производится путем предварительного изучения электронного учебника, усвоения главных положений изучаемого материала, используемых терминов и выполнения тестовых заданий. Созданный программный продукт может быть использован в учебных заведениях, обучающих специалистов соответствующих областей народного хозяйства, или в рамках курсов повышения квалификации.

**2. Основание для разработки**

⎯ лист утверждения тем проектных работ, подписанный научным руководителем: Юсковым И. О.;

⎯ наименование разработки – Создание базы данных для организации «Молока долина».

**3. Назначение разработки**

Разрабатываемый программный продукт предназначен для изучения и, в последующем, контроля качества усвоения студентами основных определений, принципов и методов, используемых в создании баз данных.

**4. Требования к программе или программному изделию**

**4.1 Требования к функциональным характеристикам**

Программная разработка должна быть написана в четком соответствии с материалами занятий по дисциплине «базы данных».

Студенту, за счет часов самостоятельной работы, разрешается заниматься дополнительным поиском информации с целью расширения возможностей программной разработки.

**4.2 Требования к надежности**

Разрабатываемое программное обеспечение должно быть спроектировано таким образом, чтобы обеспечить защиту и надежную работу при наличии ошибок во входных данных и/или от некорректных действий пользователя. Предполагается, что программный продукт должен быть спроектирован таким образом, чтобы внутренняя или внешняя (некритическая для системы) ошибка не приводила к аварийной остановке.

**4.3 Условия эксплуатации**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

Программа будет работать в температурном режиме от – 5 до + 35 °C при относительной влажности 90 % и атмосферном давлении 462 мм.рт.ст., поскольку такие условия приблизительно соответствуют условиям эксплуатации современных компьютеров непромышленного исполнения.

**4.4 Требования к составу и параметрам технических средств**

Для корректной работы программного продукта вычислительная система должна обладать следующими характеристиками:

− процессор с тактовой частотой не ниже 1.0 ГГц;

− оперативная память объемом не менее 1 Гб;

− периферийные устройства: клавиатура, мышь;

− монитор с разрешающей способностью не ниже 800 х 600.

− жесткий диск объемом не менее 20 Гб; ­

− желательно наличие принтера для печати отчета итогов работы программы.

Для корректной работы вычислительной среды необходимо наличие системного программного обеспечения, основным элементом которого является операционная система. В связи с этим, целесообразнее использовать операционную систему семейства Windows не ниже Windows 10.

**4.5 Требования к информационной и программной совместимости**

Требования к информационным структурам (файлов) на входе и выходе не предъявляются.

Исходные коды программы должны быть реализованы на языке Python. В качестве интегрированной среды разработки программы должна быть использована среда PostgreSQL.

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы.

**4.6 Требования к маркировке и упаковке**

Продукт поставляется в виде сайта, выгруженного в сеть интернет.

Программное изделие должно иметь маркировку с обозначением товарного знака организации-разработчика, типа (наименования), номера версии, порядкового номера, даты изготовления и номера сертификата соответствия Госстандарта России (если таковой имеется). Маркировка должна быть нанесена на программное изделие в виде наклейки, выполненной полиграфическим способом с учетом требований ГОСТ 9181-74.

**4.7 Требования к транспортированию и хранению**

⎯ транспортировка разрабатываемого программного продукта должна осуществляться студентом в назначенный срок по указанию преподавателя, принимающего результаты работы по проекту;

⎯ ответственным за хранение программного продукта является студент;

**5. Требования к программной документации**

Предварительный перечень программной документации:

⎯ пояснительная записка к разработанному проекту, оформленная в соответствии с ГОСТ 7.32-2017;

⎯ разделы пояснительной записки должны соответствовать материалам занятий.

**6. Технико-экономические показатели**

В данной работе не рассчитываются.

**7. Стадии и этапы разработки**

Стадии и этапы разработки определены в план-графике.

**8. Порядок контроля и приемки**

⎯ контроль и приемка программного продукта осуществляется в течение семестра в соответствии с план-графиком.

# ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Принципы создания базы данных в рамках организации веб-сайта компании «Молока долина»

Создание базы данных для сайта «Молока долина», который специализируется на производстве молочных продуктов и их доставке, начинается с определения его целей и задач. Сайт такой компании представляет собой сложную информационную систему для эффективного хранения большого объема данных о продуктах, покупателях, заказах, логистике, сотрудниках и финансовой информации. При разработке такой системы важно учитывать масштабируемость, безопасность, производительность и надежность системы.

Важность масштабируемости базы данных заключается в том, что она играет ключевую роль в обеспечении эффективной работы сайта при увеличении объема данных и количества пользователей. Масштабируемость базы данных позволяет системе эффективно работать с увеличивающимся объемом данных и количеством пользователей. Это достигается за счет использования технологий, таких как кластеризация, репликация данных и горизонтальное масштабирование.

Безопасность данных является одной из ключевых задач при проектировании базы данных. Для обеспечения безопасности используются методы шифрования данных, аутентификации пользователей и ограничения доступа к конфиденциальной информации.

Производительность системы во многом зависит от правильного выбора СУБД и оптимизации запросов к базе данных. Выбор СУБД зависит от требований к системе, таких как объем данных, количество пользователей, требования к производительности и надежности. Для небольших проектов могут использоваться бесплатные СУБД, такие как MySQL или PostgreSQL, а для крупных проектов могут потребоваться коммерческие СУБД, такие как Oracle или Microsoft SQL Server.

Надежность системы обеспечивается использованием технологий репликации данных, автоматического резервного копирования и мониторинга состояния системы.

Основные элементы базы данных включают таблицы для хранения данных о разных объектах компании, таких как названия продуктов, цены, местоположения заводов, производящих этот продукт, и сотрудников этих заводов; связи между таблицами для формирования сложных запросов; индексы для ускоренной обработки данных в крупных таблицах; а также хранимые процедуры и триггера для автоматизации стандартных процессов и обеспечения бизнес-логики на уровне базы данных.

## 1.2 Выбор системы управления базами данных (СУБД)

Выбор СУБД для базы данных веб-сайта «Молока долина» является важным этапом проектирования информационной системы. Необходимо учесть множество факторов, таких как: тип данных, объем данных, требования к безопасности, стоимость и т.д.

Реляционные базы данных представляют собой классический тип баз данных, основанный на реляционной модели данных. В этой модели данные организованы в таблицы, состоящие из строк (записей) и столбцов (полей). Каждая запись в таблице представляет собой уникальный объект, а каждое поле содержит определенную характеристику этого объекта.

Реляционные СУБД предлагают высокую степень безопасности, нормализацию данных и интегрированы с другими бизнес-приложениями, однако они могут быть ограничены в масштабируемости и сложности запросов. Нереляционные СУБД более гибкие и масштабируемые, но могут иметь меньшую производительность и безопасность.

Нереляционные базы данных, также известные как NoSQL базы данных, представляют собой тип баз данных, которые не следуют реляционной модели данных. Вместо этого они используют другие модели данных, такие как документы, ключи-значения, графы и т. д., для хранения и доступа к данным.

NoSQL базы данных обычно используются в приложениях, где требуется высокая масштабируемость, гибкость и устойчивость к большим объемам данных. Они также могут быть более эффективными для доступа к неструктурированным данным, таким как текстовые документы или изображения.

Однако, нереляционные базы данных могут иметь некоторые недостатки по сравнению с реляционными базами данных, такие как меньшая стандартизация, меньшая поддержка со стороны инструментов и отсутствие некоторых функций, таких как транзакции и ссылочная целостность. В целом, выбор СУБД должен основываться на требованиях проекта, объеме данных и бюджете.

Таким образом, выбор СУБД для веб-сайта «Молока долина» должен быть тщательно обоснован, учитывая потребности бизнеса и технические требования проекта.

## 1.3 Разработка схемы базы данных

Процесс создания схемы базы данных сайта «Молока долина» представляет собой разработку структурированной модели данных, которая отражает все ключевые аспекты бизнеса компании и обеспечивает необходимую поддержку ее операций. Такая схема должна быть адаптируемой к будущим изменениям и масштабированию, а также оптимизированной для повышения производительности и эффективного выполнения запросов.

**Сущности:** В начале проектирования выделяются ключевые сущности, представляющие объекты бизнес-процессов «Молока долина». К ним относятся:

* Заводы (factory) – данные о заводах.
* Продукты (products) – данные о производимых молочных продуктах.
* Работники (employee) – данные о работниках.
* Коровы (cows) – данные о коровах.
* Клиенты (customers) – данные о клиентах.
* Заказы (orders) – данные о заказах клиентов.
* Отзывы (feedbacks) – данные об отзывах организации.

**Связи:** Связи между сущностями определяют, как данные в одной таблице соотносятся с данными в другой. Типы связей включают:

* *Один-ко-многим:* например, завод к нескольким продуктам, производящимся на нем.

**Проектирование схемы:** при проектировании схемы базы данных используется ER-диаграмма. Выполняются следующие шаги:

1. Создается ER-диаграмма для визуализации структуры базы данных.
2. Определяется нормализация данных для минимизации дублирования и упрощения структуры.
3. Планируется индексация для ускорения поиска и выборки данных.
4. Подготавливается документация схемы для понимания и поддержки базы данных.

## 1.4 Проблемы проектирования баз данных и способы их решения

Проектирование баз данных для веб-сайта организации «Молока долина» сопровождается рядом вызовов, обусловленных особенностями отрасли и требованиями к эффективности веб-ресурсов. Ниже представлены основные проблемы и соответствующие им решения:

* + 1. **Обеспечение производительности при высоких нагрузках:**

Проблема: Веб-сайты компаний могут испытывать пиковые нагрузки, требующие быстрого доступа к данным и способности обрабатывать большое количество запросов.

Решение:

* Использование кластеризация по индексу для ускорения поиска данных.
* Применение кеширования часто запрашиваемой информации.
* Реализация горизонтального масштабирования.

**1.4.2 Управление большими объемами данных:**

Проблема: Базы данных часто содержат большое количество данных о проектах, материалах, клиентах и транзакциях.

Решение:

* + Внедрение автоматизированных решений для архивации и удаления устаревших данных.
  + Организация эффективной структуры хранения данных, например, с помощью нормализации и партиционирования таблиц.

**1.4.3 Обслуживание базы данных:**

Проблема: Необходимость в постоянном мониторинге и обслуживании для обеспечения стабильности и производительности системы.

Решение:

* + Автоматизация мониторинга состояния базы данных.
  + Настройка оповещений и тревог для раннего выявления проблем.
  + Регулярное проведение аудита безопасности и проверок производительности.

**1.4.4 Использование кластеризации и репликации:**

Проблема: Обеспечение отказоустойчивости и повышенной производительности.

Решение:

* + Внедрение кластеризации для распределения нагрузки.
  + Применение репликации для улучшения доступности данных и распределения нагрузки чтения.

Решения этих проблем должны быть тщательно обоснованы и соответствовать конкретным потребностям и возможностям организации «Молока долина». Каждое решение имеет свои требования к ресурсам и может влиять на архитектуру системы в целом.

## 1.5 Аналитика данных в организации

Отчетность: Анализ данных в компании «Молока Долина» позволяет автоматизировать процесс создания отчетов о показателях производства, заказах, запасах и финансах. Визуализация данных в виде графиков и диаграмм упрощает понимание информации и представление отчетов заинтересованным сторонам.

Прогнозирование: Используя аналитические инструменты, компания может прогнозировать количество заказов на разные временные промежутки, управлять запасами и оптимизировать процессы доставки. Прогноз также позволяет предсказать возможные риски и принять меры предосторожности.

Следующие шаги: Внедрение инструментов анализа данных в компании «Молока Долина» помогает выйти за рамки текущего состояния и перейти к более стратегическому подходу к управлению бизнес-процессами, способствуя непрерывному улучшению и развитию.

## 1.6 Использование данных для стратегического планирования

В компании «Молока Долина» стратегическое планирование является ключевым элементом обеспечения долгосрочного роста и конкурентоспособности на рынке. База данных, включающая операционные, финансовые и клиентские данные, служит ключевым активом, обеспечивая основу для принятия обоснованных решений.

Определение стратегических направлений: Анализ данных помогает определить прибыльные сегменты рынка и направления для развития компании. Это включает исследование рыночных трендов, анализ спроса и предложения и оценку эффективности прошлых проектов. Решения о запуске новых продуктов, выходе на новые рынки и диверсификации предложений принимаются на основе анализа данных.

Управление рисками: Использование данных для выявления и управления рисками помогает снизить возможные негативные последствия. Анализ данных позволяет определить причины задержек, перерасхода бюджета и проблем с качеством. На основе этого разрабатываются стратегии управления рисками.

Распределение ресурсов: Точное понимание потребностей в ресурсах на основе данных позволяет эффективно планировать закупки, контролировать запасы и оптимизировать использование ресурсов, что способствует снижению затрат и повышению общей эффективности.

Вывод: Стратегическое планирование на основе данных делает сайт организации «Молока долина» более гибким, адаптивным к рыночным условиям и обеспечивает его конкурентоспособность. Данные становятся ценным ресурсом для развития стойкого и успешного бизнеса.

## 1.7 Техническая поддержка и обслуживание базы данных

Для обеспечения надежности и эффективности работы баз данных компания «Молока долина» применяет комплексный подход, включающий в себя регулярное резервное копирование данных, обновление и миграцию систем, а также обеспечение безопасности и технической поддержки.

Резервное копирование является ключевым элементом стратегии обеспечения непрерывности бизнес-процессов. Оно включает регулярное создание и тестирование резервных копий, а также их хранение в безопасных и надежных местах. Компания использует современные технологии и оборудование для обеспечения высокого качества резервных копий и их быстрого восстановления в случае сбоев.

Обновление и миграция систем также являются важными аспектами работы с базами данных. Компания регулярно обновляет свои системы управления базами данных, чтобы обеспечить их соответствие современным требованиям и стандартам. Миграция данных также требует тщательного планирования, тестирования и анализа рисков, чтобы минимизировать вероятность потери данных и нарушения работы системы.

Безопасность данных является одним из ключевых приоритетов компании. Она обеспечивается за счет применения мер физической безопасности серверов, использования современных программных средств защиты и регулярного обновления систем безопасности. Кроме того, компания проводит аудит прав доступа к данным и осуществляет их шифрование, чтобы обеспечить максимальную защиту от возможных угроз.

Мониторинг производительности базы данных необходим для контроля времени отклика, скорости транзакций и эффективности работы с данными. Он помогает оптимизировать структуру базы и выявлять узкие места, а также масштабировать систему при необходимости.

# ГЛАВА 2. СОЗДАНИЕ БАЗЫ ДАННЫХ

База данных для курсовой работы была выполнена с помощью PostgreSQL в оболочке pgAdmin4.

## 2.1 ER модель базы данных

На рисунке 1 представлена структурная схема данных моей базы данных.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рисунок 1 – ER-диаграмма базы данных сайта.

## 2.2 Структура базы данных

Данная база данных содержит 8 таблиц:

Таблица «factory» представляет собой хранилище информации о заводах, рисунок 2. В данной таблице содержатся следующие поля:

1. **factory\_id (первичный ключ):** Уникальный идентификатор заводов, который обеспечивает уникальность записи в таблице.
2. **location:** Данное поле предназначено для хранения адресов заводов.
3. **manager\_id:** Поле с идентификатором управляющего заводом.
4. **workers\_count:** Поле с количеством работников на заводе.
5. **description:** Поле с дополнительным описанием завода.

Изображение выглядит как текст, Шрифт, линия, число

Автоматически созданное описание

Рисунок 2 - Таблица «factory»

Вторая таблица «products» предназначена для хранения информации о продуктах, производимых на заводах, рисунок 3. В данной таблице содержатся следующие поля:

1. **product\_id (первичный ключ):** Уникальный идентификатор каждого продукта, который обеспечивает уникальность записи в таблице.
2. **Factory\_id:** Поле с идентификатором завода, на котором производится продукт.
3. **Prod\_name:** Данное поле предназначено для хранения названия продукта.
4. **Price:** Данное поле предназначено для хранения цены продукта за единицу товара.

Изображение выглядит как текст, число, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 3 - Таблица «products»

Третья таблица «employee» предназначена для хранения информации о работнике завода, рисунок 4. В данной таблице содержатся следующие поля:

1. **Employee\_id (первичный ключ):** Уникальный идентификатор каждого работника, который обеспечивает уникальность записи в таблице.
2. **Factory\_id:** Поле идентификатора завода, на котором работает человек.
3. **Full\_name:** Поле имени человека.
4. **Birthday:** Информация о дате рождения человека.
5. **Sex:** Информация о поле работника.
6. **Telephone:** Поле с номером телефона.

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

Рисунок 4 - Таблица «employee»

Четвертая таблица «cows» предназначена для учета коров на заводе, рисунок 5. В данной таблице содержатся следующие поля:

1. **Cow\_id (первичный ключ):** Уникальный идентификатор каждой коровы, который обеспечивает уникальность записи в таблице.
2. **Cow\_name:** Данное поле предназначено для имени коровы.
3. **Factory\_id:** Данное поле предназначено для хранения информации о заводе, к которому принадлежит корова.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 5 - Таблица «Price»

Пятая таблица «feedback» предназначена для отзывов покупателей о продукции, рисунок 6. В данной таблице содержатся следующие поля:

1. **Feedback\_id (первичный ключ):** Уникальный идентификатор каждого отзыва, который обеспечивает уникальность записи в таблице.
2. **Full\_name:** Это поле предназначено для имени человека, оставившего отзыв.
3. **Phone\_number:** Поле с телефонным номером человека, оставившего отзыв.
4. **Email:** Поле с электронной почтой человека, оставившего отзыв.
5. **Feedback\_text:** Поле для текста отзыва.
6. **Date\_of\_feedback:** Время, когда отзыв был отправлен.

Изображение выглядит как текст, число, меню, снимок экрана

Автоматически созданное описание

Рисунок 6 - Таблица «feedback»

Шестая таблица «customers» предназначена для данных о покупателях, рисунок 7. В данной таблице содержатся следующие поля:

1. **customer\_id (первичный ключ):** Уникальный идентификатор каждого клиента, который обеспечивает уникальность записи в таблице.
2. **First\_name:** Это поле предназначено для имени покупателя.
3. **Last\_name:** Это поле предназначено для фамилии покупателя.
4. **Phone\_number:** Поле с телефонным номером клиента.
5. **email:** Поле с электронной почтой клиента.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 7 – Таблица «customers»

# ГЛАВА 3. ПОДКЛЮЧЕНИЕ БД К САЙТУ.

Программа для курсовой работы была разработана в редакторе кода VS Code и реализована на языке программирования Python (фреймворк Django).

## 3.1 Структура проекта

Для того чтобы присоединить базу данных к сайту, понадобилось развернуть проект в Django и создать в нем приложения для групп страниц сайта (about, product, menu и homepage).

Структура программного проекта, рисунок 8:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 8 – Структура проекта

Далее были размещена верстка в файле templates, а также стили в файле static\_dev, рисунок 9.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 9 – Размещение верстки

## 3.2 Программная реализация

Код Django базовой страницы представлен ниже.

{% load static %}

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <link rel="stylesheet" href="{% static 'assets/css/bootstrap.css' %}">

    <link rel="stylesheet" href="{% static 'assets/css/style.css' %}">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Montserrat+Alternates:wght@400;700&family=Montserrat:wght@400;700&display=swap" rel="stylesheet">

    <title>Молока долина | Главная</title>

</head>

<body>

  {% include "includes/header.html" %}

  <main>

    {% block content %}{% endblock %}

  </main>

</body>

</html>

Все коды верстки приложений представлены ниже.

Код верстки приложения about.

{% extends "base.html" %}

{% load static %}

{% block content %}

    <div class="container">

        <div class="row">

            <div class="col-5">

                <div class="text">

                    <p><h2>О нас</h2></p>

                </div>

            </div>

        </div>

        <div class="row">

            <div class="col-lg-6 col-sm-12">

                <div class="text">

                    <p><h4>Мы стремимся к тому, чтобы бизнес можно было уверенно сравнивать с лучшими механизмами, где каждая деталь находится на своем месте и обеспечивает общую бесперебойную работу. </h4></p>

                    <p>Наше предприятие производит молочную продукцию на основе молока и растительного сырья. Мы производим продукты по собственной рецептуре с оригинальными вкусами, в том числе теми, которых пока нет на Российском рынке. В производстве мы используем сырье без консервантов, ГМО и усилителей вкуса. Более подробную информацию - каталог и презентацию запрашивайте в отделе по работе к клиентами.</p>

                </div>

            </div>

            <div class="col-lg-6 offset-lg-0 col-sm-8 offset-sm-2">

                <img src="{% static 'assets/images/farm-ranch-10.jpg' %}" alt="" width="100%">

            </div>

        </div>

        <div class="row">

            <div class="col-5">

                <div class="text">

                    <p><h2>Наши партнёры</h2></p>

                </div>

            </div>

        </div>

        <div class="row">

            <div class="col-lg-6 col-sm-12">

                <div class="text">

                    <p><h4><a class="nav\_p\_link" href="https://imuntouchable.github.io/#">Кафе "Гараж"</a></h4></p>

                    <p>Если вы привыкли двигаться в ускоренном темпе, то сама идея быстрой еды для вас не будет в новинку. Для тех, кто желает скорее перекусить в перерыве между делами, кафе быстрого питания Гараж предлагает ряд простых, но питательных блюд по умеренным ценам. Еда готовится таким образом, чтобы ее можно было взять с собой. Всех гостей ждут по адресу: Республика Северная Осетия — Алания, улица Морских Пехотинцев, 8. Для гостей заведение работает Пн-вс: 10:00 - 23:00.</p>

                </div>

            </div>

            <div class="col-lg-6 offset-lg-0 col-sm-8 offset-sm-2">

                <div class="garage-slider">

                    <div class="garage\_img">

                        <img src="{% static 'assets/images/Кафе гараж.jpg' %}" alt="Кафе гараж" width="50%" class="image">

                    </div>

                    <div class="garage\_img">

                        <img src="{% static 'assets/images/garage.jpg' %}" alt="Кафе гараж" width="50%" class="image">

                    </div>

                    <div class="garage\_img">

                        <img src="{% static 'assets/images/garage\_2.jpg' %}" alt="Кафе гараж" width="68%" class="image">

                    </div>

                    <div class="garage\_img">

                        <img src="{% static 'assets/images/garage\_3.jpg' %}" alt="Кафе гараж" width="75%" class="image">

                    </div>

                </div>

            </div>

        </div>

        <div class="row">

            <div class="col-lg-5 col-sm-12">

                <div class="text\_under">

                    <p><h5>Контактная информация:</h5></p>

                    <p><a class = "nav\_tel\_link" href="tel: +79104108877">+7 910 410 88 77</a> (WhatsApp, Telegram)</p>

                    <p>molokadolina@outlook.com</p>

                </div>

            </div>

            <div class="col-lg-6 offset-lg-1 col-sm-12">

                <div class="text\_under">

                    <p><h5>Юридический адрес</h5></p>

                    <p>Москва, 109429, Верхние Поля, 51А</p>

                </div>

            </div>

        </div>

    </div>

    <a href="#" class="top"></a>

    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>

    <!-- Подключаем стили slick -->

    <link rel="stylesheet" href="{% static 'slick/slick.css' %}">

    <link rel="stylesheet" href="{% static 'slick/slick-theme.css' %}">

    <!-- Подключаем саму библиотеку slick -->

    <script src="{% static 'slick/slick.min.js' %}"></script>

    <!-- Инициализируем слайдер -->

    <script>

        $(document).ready(function(){

            $('.garage-slider').slick({

                slidesToShow: 1,

                slidesToScroll: 1,

                autoplay: true,

                autoplaySpeed: 2000,

                infinite: true,

                speed: 500,

                adaptiveHeight: true

             });

        });

    </script>

    <script>

        function ToTop(){

            let button = $('.top');

            $(window).on('scroll', () => {

                if ($(this).scrollTop() >= 70){

                    button.fadeIn();

                } else {

                    button.fadeOut();

                }

            });

            button.on('click', (e) => {

                e.preventDefault();

                $('html').animate({scrollTop: 0}, 100);

            })

        }

        ToTop();

    </script>

{% endblock %}

Код верстки приложения contacts.

{% extends "base.html" %}

{% load static %}

{% block content %}

    <div class="container">

        <div class="row">

            <div class="col-sm-4 offset-sm-4">

                <img src="{% static 'assets/images/logo\_horizontal\_on-white\_\_1.png' %}" alt="" width="100%">

            </div>

        </div>

        <div class="row">

            <div class="col-5">

                <div class="text">

                    <p><h2>Контакты:</h2></p>

                </div>

            </div>

        </div>

        <div class="row">

            <div class="col-lg-6 col-md-12">

                <div class="text">

                    <div class="text\_p">

                        <h5>Коммерческий отдел</h5>

                        <p><a class = "nav\_tel\_link" href="tel: +79104108877">+7 910 410 88 77</a> (WhatsApp, Telegram)</p>

                        <p>Приём заявок с 9:00 до 19:00</p>

                    </div>

                    <div class="text\_p">

                        <h5>Адрес</h5>

                        <p>Солнечногорск, Красная улица, д. 186</p>

                    </div>

                    <div class="text\_p">

                        <h5>Реквизиты компании</h5>

                        <p>

                            ИНН: 111111 <br>

                            Банк: Сбербанк

                        </p>

                    </div>

                </div>

            </div>

            <div class="col-lg-5 offset-lg-1 col-md-8 offset-md-2">

                <div class="text">

                    <div class="text\_p">

                        <h5>Обратная связь</h5>

                    </div>

                </div>

                <div class="form">

                    <form action="#" method="post" id="message" width="70%">

                        <input type="text" name="name" class="form\_control input-lg" placeholder="ФИО\*" style="width: 100%;">

                        <input type="tel" name="tel" class="form\_control input-lg" placeholder="Телефон" style="width: 100%;">

                        <input type="email" name="email" class="form\_control input-lg" placeholder="e-mail\*" style="width: 100%;" required>

                        <input type="text" name="message" class="form\_control input-lg" placeholder="Ваше сообщение\*" style="width: 100%;" required>

                        <input name="submit" type="submit" class="btn btn-primary form\_btn" value="Отправить сообщение">

                    </form>

                </div>

            </div>

        </div>

    </div>

{% endblock %}

Код верстки приложения product.

{% extends "base.html" %}

{% load static %}

{% block content %}

    <div class="container">

        <div class="row">

            <div class="col-5">

                <div class="text">

                    <h2>Тан турецкий</h2>

                </div>

            </div>

        </div>

        <div class="row">

            <div class="col-lg-5 col-sm-12">

                <div class="text">

                    <p>Тан - кисломолочный напиток, приготовляемый из коровьего или козьего молока с добавлением закваски молочнокислых бактерий (болгарская палочка, термофильный стрептококк), молочных дрожжей, воды и поваренной соли. Главной технологической особенностью изготовления кисломолочных продуктов является сквашивание путём введения в него культур молочнокислых бактерий или дрожжей (самокваса или закваски). Часто перед производством кисломолочных продуктов используют предварительную пастеризацию или кипячение молока для исключения возможности развития жизнедеятельности находящихся в нём вредных микроорганизмов. Вы любите кисломолочные напитки? Тогда почему бы вам не пробовать наш ТАН.</p>

                </div>

            </div>

            <div class="col-lg-5 offset-lg-1 col-sm-8 offset-sm-2">

                <img src="{% static 'assets/images/Тан бутылка.jpg' %}" alt="Тан" class="image" width = 70%>

            </div>

        </div>

        <div class="row">

            <div class="col-lg-5 col-sm-12">

                <div class="text\_under">

                    <p><h5>Контактная информация:</h5></p>

                    <p><a class = "nav\_tel\_link" href="tel: +79104108877">+7 910 410 88 77</a> (WhatsApp, Telegram)</p>

                    <p>molokadolina@outlook.com</p>

                </div>

            </div>

            <div class="col-lg-6 offset-lg-1 col-sm-12">

                <div class="text\_under">

                    <p><h5>Юридический адрес</h5></p>

                    <p>Москва, 109429, Верхние Поля, 51А</p>

                </div>

            </div>

        </div>

    </div>

{% endblock %}

Для того, чтобы Django распознал данные приложения, необходимо зарегистрировать приложения, а также прописать настройки для них. Все настройки представлены в файле settings.py, код которого представлен ниже.

from pathlib import Path

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

SECRET\_KEY = 'django-insecure-(-tov\_^hyeclfhtje4$\*2((dc3gkvm77ty8o6n82)-ei4p#)3m'

DEBUG = True

ALLOWED\_HOSTS = []

INSTALLED\_APPS = [

    'about.apps.AboutConfig',

    'product.apps.ProductConfig',

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'Website.urls'

TEMPLATES\_DIR = BASE\_DIR / 'templates'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [TEMPLATES\_DIR],

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ]

        },

    }

]

STATICFILES\_DIRS = [

    BASE\_DIR / 'static\_dev',

]

WSGI\_APPLICATION = 'Website.wsgi.application'

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.postgresql',

        'NAME': 'milk\_factory',

        'USER': 'postgres',

        'PASSWORD': '123456',

        'HOST': 'localhost',

        'PORT': '5433',

    }

}

AUTH\_PASSWORD\_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

    },

]

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

STATIC\_URL = '/static/'

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

Подсоединение базы данных происходит через переменную DATABASES. Запросы пользователя осуществляются через Django ORM (Object Relational Mapping) – систему управления базами данных, которая позволяет работать с данными в объектно-ориентированном стиле, вместо того чтобы использовать SQL запросы напрямую. То есть сначала Django обрабатывает запрос через встроенную ORM, далее формирует автоматически SQL-запрос и получает данные.

Основное различие между моделями PostgreSQL и моделями Django заключается в том, что модели PostgreSQL используются для работы с базой данных напрямую, а модели Django используются для организации и доступа к данным через ORM.

Таким образом, база данных была подключена. Результат работы можно посмотреть на рисунке 9.

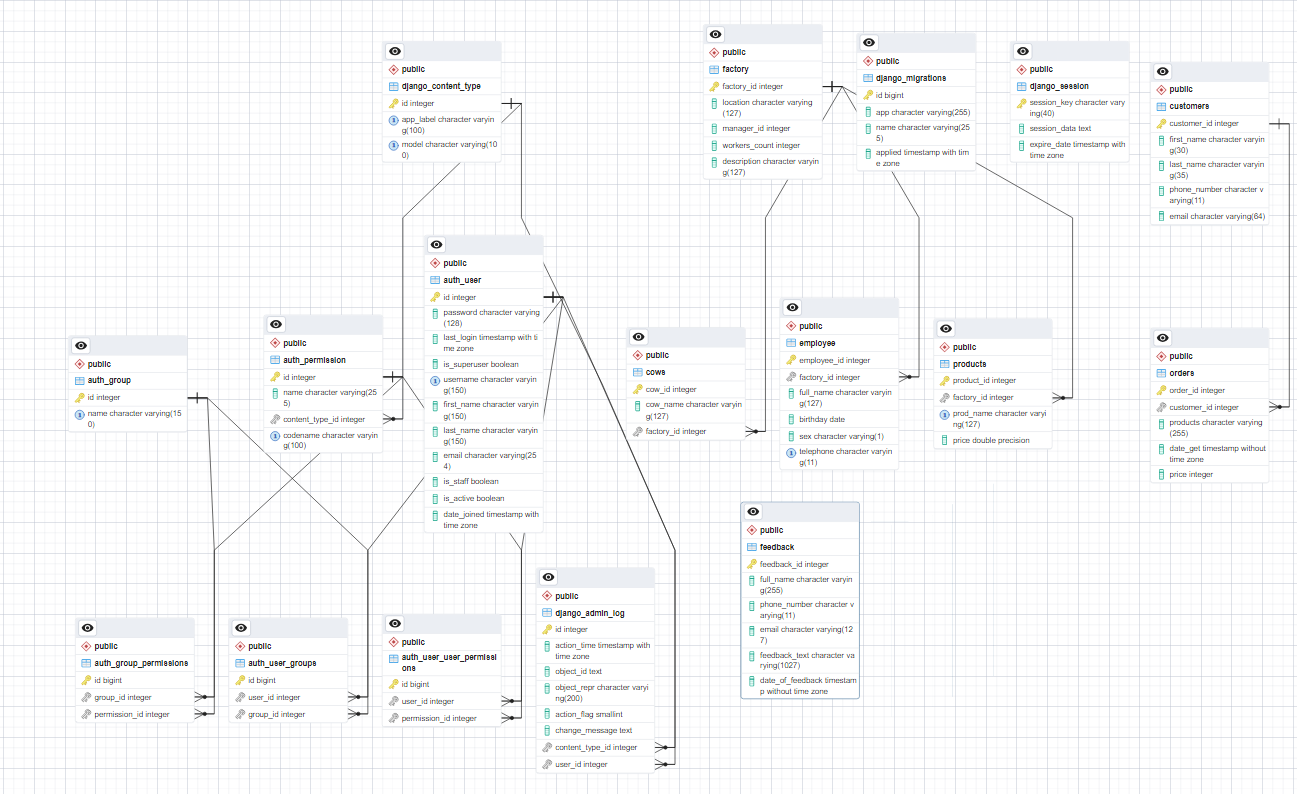


Рисунок 9 – Подключенные данные Django

# ЗАКЛЮЧЕНИЕ

Целью проекта было создание системы управления базами данных для компании «Молока долина» с применением PostgreSQL. В ходе работы мы изучили теоретические аспекты построения баз данных и систем управления, а также непосредственно использовали PostgreSQL для хранения и обработки информации.

Основные моменты, рассмотренные в работе:

1. **Теоретические основы:** Исследованы функции и возможности систем управления базами данных, опираясь на методологию PostgreSQL. Такой подход позволил нам лучше понять процессы создания и управления базами данных.
2. **Системный анализ и моделирование:** Был проведен системный анализ предметной области, созданы информационная и реляционная модели, определяющие структуру базы данных и её ключевые компоненты.
3. **Реализация в PostgreSQL:** База данных была успешно реализована в системе PostgreSQL. Создана реляционная база данных, содержащая 15 таблиц, предоставляя надежное хранение и обработку данных.
4. **Проектирование базы данных**: Была создана структура базы данных, включающая таблицы для наименований блюд, типов блюд, составов блюд, цен блюд, наличия постов на сайте и других важных элементов. Это обеспечило надежное хранение и упорядочивание данных.

Таким образом, разработанная система гарантирует эффективное управление данными «Молока долины», гарантируя надежность хранения информации и открывая возможности для будущего роста и интеграции с различными проектами.

# СПИСОК ИСПОЛЬЗОВАНЫХ ИСТОЧНИКОВ

1. kenwheeler.github.io: библиотека для слайдера: сайт. – URL: <https://kenwheeler.github.io/slick/> (дата обращения 04.12.2023). – Режим доступа: для всех пользователей. – Текст: электронный.
2. jquery.com: библиотека для JavaScript: сайт. – URL: [https://jquery.com/](https://jquery.com/%20) (дата обращения 01.12.2023). – Режим доступа: для всех пользователей. Текст: электронный.
3. getbootstrap.com: библиотека для верстки: сайт. URL: [https://getbootstrap.com/docs/5.3/](https://getbootstrap.com/docs/5.3/%20) (дата обращения 11.11.2023). – Режим доступа: для всех пользователей. – Текст: электронный.
4. github.com: бесплатный хостинг: сайт. – URL: [https://github.com/new](https://github.com/new%20) (дата обращения 09.11.2023). – Режим доступа: для всех пользователей. – Текст: электронный.
5. postgresql.com: СУБД для сайта: сайт. – URL: <https://www.postgresql.org/> (дата обращения 27.10.2023). – Режим доступа: для всех пользователей. – Текст: электронный.
6. djangoproject.com: Фреймворк для работы сайта с СУБД: сайт. – URL: <https://www.djangoproject.com/> (дата обращения 12.11.2023). – Режим доступа: для всех пользователей. – Текст: электронный.