РЕФЕРАТ

Отчет 187 страниц, 8 рисунков, 2 таблицы, 6 источников, 7 приложений.

БАЗА ДАННЫХ, SUSHI SHOP, ВЕБ-ПРИЛОЖЕНИЕ, POSTGREESQL, СУБД,

Объект исследования: Процесс управления данными в сети кафе-ресторанов.

Предмет исследования: Методы и средства проектирования, реализации и эксплуатации реляционной базы данных для задач учета и анализа в ресторанном бизнесе.

Цель работы: Разработка информационной системы для автоматизации учета продуктов, расчета себестоимости и маржинальности блюд на примере сети ресторанов "Sushi-shop".

Задачи:

* Проанализировать предметную область и сформулировать требования к базе данных.
* Разработать концептуальную, логическую и физическую модели данных.
* Обосновать выбор СУБД.
* Реализовать базу данных в выбранной СУБД (таблицы, связи, представления, права доступа).
* Разработать и протестировать SQL-запросы для извлечения данных и формирования отчетов.

Итогом работы стала разработанная база данных и веб-приложение для работы с ней.

Практическая значимость работы заключается в создании прототипа базы данных, который может служить основой для разработки полнофункциональной информационной системы управления ресторанами.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc198068552)

[Раздел 1. Анализ предметной области 5](#_Toc198068553)

[1.1 Определение границ исследования 5](#_Toc198068554)

[1.2 Анализ существующих подходов к автоматизации учета 6](#_Toc198068555)

[1.3 Требования к разрабатываемой базе данных 8](#_Toc198068556)

[1.3.1 Функциональные требования к базе данных 8](#_Toc198068557)

[1.3.2 Нефункциональные требования к базе данных 9](#_Toc198068558)

[1.3.3 Требования к данным 10](#_Toc198068559)

[Раздел 2. Проектирование базы данных 11](#_Toc198068560)

[2.1 Разработка концептуальной модели (ER-диаграмма) 11](#_Toc198068561)

[2.2 Разработка логической модели 13](#_Toc198068562)

[2.3 Выбор системы управления базами данных (СУБД) 15](#_Toc198068563)

[2.4 Разработка физической модели 18](#_Toc198068564)

[2.5 Проведение нормализации 20](#_Toc198068565)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 22](#_Toc198068566)

ВВЕДЕНИЕ

Актуальность автоматизации бизнес-процессов в сфере общественного питания неоспорима. В условиях высокой конкуренции и постоянно растущих требований к качеству обслуживания, эффективное управление ресурсами становится ключевым фактором успеха. Особое значение приобретает точный учет продуктов на складах, калькуляция себестоимости блюд и анализ маржинальности, поскольку эти показатели напрямую влияют на рентабельность заведения. Использование баз данных (БД) предоставляет мощный инструментарий для решения этих задач, обеспечивая структурированное хранение информации, быстрый доступ к ней и возможности для комплексного анализа.

Постановка задачи данной курсовой работы заключается в разработке базы данных для информационной поддержки основных процессов учета и анализа в сети кафе-ресторанов "Sushi-shop". Основной акцент будет сделан на управлении складскими запасами, учете состава блюд, расчете их себестоимости и последующем анализе маржинальности.

Целью работы является создание эффективной и надежной реляционной базы данных, обеспечивающей автоматизацию учета продуктов на складах, точный расчет себестоимости блюд и их маржинальности для сети ресторанов "Sushi-shop".

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области и сформулировать детальные требования к разрабатываемой базе данных.
2. Разработать концептуальную, логическую и физическую модели данных, отражающие структуру информации в предметной области.
3. Обосновать выбор системы управления базами данных (СУБД) для реализации проекта.
4. Реализовать спроектированную базу данных в выбранной СУБД, включая создание таблиц, связей, представлений (view), а также настройку прав доступа для различных категорий пользователей.
5. Разработать и протестировать набор SQL-запросов для извлечения необходимой информации и формирования отчетов.

Объектом исследования является процесс управления данными в сети кафе-ресторанов.

Предметом исследования выступают методы и средства проектирования, реализации и эксплуатации реляционной базы данных для решения задач учета и анализа в указанной предметной области.

Практическая значимость работы заключается в создании прототипа базы данных, который может быть использован как основа для дальнейшей разработки полнофункциональной информационной системы управления сетью ресторанов "Sushi-shop". Разработанные модели и SQL-запросы могут служить методическим материалом при решении аналогичных задач.

# Раздел 1. Анализ предметной области

Для успешного проектирования и реализации эффективной базы данных необходимо провести тщательный анализ предметной области, определить границы исследования и сформулировать четкие требования к будущей системе.

## 1.1 Определение границ исследования

Предметом исследования данной курсовой работы является разработка базы данных для сети кафе-ресторанов "Sushi-shop". Информационная система, основанная на этой БД, призвана автоматизировать и оптимизировать ключевые процессы, связанные с учетом продуктов, приготовлением блюд и анализом экономической эффективности.

Границы исследования охватывают следующие аспекты деятельности сети "Sushi-shop":

1. Учет продуктов на складах: поступление продуктов от поставщиков, хранение информации о продуктах (наименование, единицы измерения, сроки годности и т.д.), внутреннее перемещение продуктов между складами (если предполагается несколько складов в рамках одной точки или между точками), списание продуктов (по причине порчи, использования в приготовлении блюд).
2. Управление рецептурами (технологическими картами): хранение информации о составе каждого блюда, включая перечень ингредиентов и их точное количество, необходимое для приготовления одной порции.
3. Учет продаж и заказов: фиксация данных о проданных блюдах для последующего расчета расхода продуктов и анализа популярности позиций меню.
4. Расчет себестоимости блюд: автоматический расчет себестоимости каждого блюда на основе актуальных закупочных цен на ингредиенты и их количества согласно рецептуре.
5. Анализ маржинальности: расчет и анализ разницы между ценой продажи блюда и его себестоимостью.
6. Управление сотрудниками в контексте доступа к данным: фиксация информации о сотрудниках для разграничения прав доступа к различным функциям и данным в системе.
7. За рамками данного исследования остаются такие аспекты деятельности ресторанного бизнеса, как управление персоналом (графики работы, начисление заработной платы, кроме как в контексте должности для определения ставки и доступа), маркетинговые кампании, программы лояльности, финансовый учет предприятия в целом, бухгалтерский учет, а также разработка полнофункционального пользовательского интерфейса приложения. Основной упор делается на структуру и логику самой базы данных.

## 1.2 Анализ существующих подходов к автоматизации учета

Автоматизация учета в ресторанном бизнесе является распространенной практикой, и на рынке существует множество программных решений, предлагающих различные функциональные возможности. Среди них можно выделить несколько основных категорий:

* POS-системы (Point of Sale): Основная функция – автоматизация процесса продаж, прием заказов, печать чеков. Многие современные POS-системы также включают базовые функции складского учета, списывая ингредиенты согласно проданным блюдам.
* Системы управления складом (Inventory Management Systems): Специализированные решения для детального учета товарно-материальных ценностей, контроля остатков, управления закупками и инвентаризациями.
* ERP-системы (Enterprise Resource Planning) для ресторанов: Комплексные системы, охватывающие большинство бизнес-процессов предприятия, включая финансы, закупки, склад, производство, продажи и управление персоналом. Примерами могут служить такие системы, как iiko или r\_keeper, которые предоставляют широкие возможности по управлению ресторанным бизнесом, включая сложные схемы учета и анализа[1][2].

Анализ существующих решений показывает, что ключевым элементом любой такой системы является хорошо структурированная база данных. Она позволяет хранить информацию о продуктах, рецептах, продажах, поставщиках и т.д. Однако, готовые решения могут быть избыточны по функционалу для небольшой сети или стартапа, требовать значительных финансовых вложений или не обладать достаточной гибкостью для адаптации под уникальные бизнес-процессы. Разработка собственной базы данных, сфокусированной на конкретных потребностях "Sushi-shop", позволит создать более легковесное, адаптированное и потенциально менее затратное решение для ключевых учетных задач.

## 1.3 Требования к разрабатываемой базе данных

На основе анализа предметной области и существующих подходов можно сформулировать следующие требования к разрабатываемой базе данных для сети "Sushi-shop".

1.3.1 Функциональные требования к базе данных

База данных должна обеспечивать хранение и обработку информации для выполнения следующих функций:

Учет продуктов:

* Хранение информации о продуктах (наименование, единица измерения, категория, описание, поставщик по умолчанию (опционально)).
* Учет поступления продуктов на склад (дата, количество, закупочная цена).
* Учет остатков продуктов на складах (с возможностью ведения учета по нескольким складам).
* Учет списания продуктов (по различным причинам: приготовление блюд, порча).

Управление блюдами и рецептурами:

* Хранение информации о блюдах (наименование, категория, цена продажи, описание, фотография (ссылка)).
* Хранение рецептур (технологических карт) для каждого блюда с указанием ингредиентов и их количества.

Учет продаж:

* Фиксация информации о проданных блюдах (дата, время, количество, блюдо, итоговая сумма по позиции).
* Расчеты и аналитика:
* Возможность расчета себестоимости каждого блюда на основе актуальных цен ингредиентов.
* Возможность расчета маржинальности по каждому блюду и по группам блюд.
* Формирование отчетов об остатках продуктов, движении товаров, продажах.

Управление сотрудниками и доступом:

* Хранение информации о сотрудниках (ФИО, должность).
* Реализация механизма разграничения прав доступа к данным на основе ролей (например, администратор, управляющий, повар).

Справочная информация:

* Хранение справочников (категории продуктов, категории блюд, единицы измерения, должности, причины списания и т.д.).

### 1.3.2 Нефункциональные требования к базе данных

Надежность: Обеспечение сохранности данных при сбоях. Использование механизмов транзакций для атомарности операций.

Целостность данных: Поддержание логической согласованности и достоверности данных с помощью первичных и внешних ключей, ограничений (constraints).

Производительность: Обеспечение приемлемого времени отклика на типовые запросы пользователей.

Безопасность: Защита данных от несанкционированного доступа.

Масштабируемость: Возможность увеличения объема хранимых данных и интенсивности запросов без значительного падения производительности (в разумных пределах для курсового проекта).

Простота администрирования: Относительная простота обслуживания и внесения изменений в структуру БД (в рамках возможностей выбранной СУБД).

### 1.3.3 Требования к данным

Актуальность: Данные, хранящиеся в БД (особенно цены на продукты, остатки), должны отражать текущее состояние дел.

Непротиворечивость: Данные в различных частях БД не должны противоречить друг другу.

Полнота: БД должна содержать все необходимые данные для выполнения заявленных функций.

Точность: Данные должны быть корректными и точно отражать реальные значения (например, количество ингредиентов, цены).

Вывод:

В данной главе был проведен анализ предметной области – деятельности сети кафе-ресторанов "Sushi-shop" – и определены границы исследования для разработки базы данных. Рассмотрены существующие подходы к автоматизации учета в ресторанном бизнесе, что позволило выявить как преимущества, так и потенциальные ограничения готовых решений. На основе этого анализа были сформулированы подробные функциональные и нефункциональные требования к разрабатываемой базе данных, а также требования к самим данным. Эти требования послужат основой для последующих этапов проектирования и реализации базы данных "Sushi-shop".

# Раздел 2. Проектирование базы данных

После проведения анализа предметной области и формулировки требований к базе данных, следующим ключевым этапом является ее проектирование. Процесс проектирования включает разработку концептуальной, логической и физической моделей данных, выбор системы управления базами данных (СУБД) и проведение нормализации для обеспечения целостности и эффективности хранения информации.

## 2.1 Разработка концептуальной модели (ER-диаграмма)

Концептуальное моделирование является первым шагом в проектировании базы данных и направлено на создание высокоуровневого представления предметной области. На этом этапе определяются основные сущности (информационные объекты), их ключевые атрибуты и типы связей между ними, без учета деталей реализации и специфики конкретной СУБД[3].

Для информационной системы сети кафе-ресторанов "Sushi-shop" были выделены следующие основные сущности:

* Сотрудник: Информация о работниках заведения.
* Должность: Справочник должностей и связанных с ними ставок.
* Касса: Информация о кассовых аппаратах.
* Смена: Данные о рабочих сменах сотрудников.
* Склад: Информация о местах хранения продуктов.
* Продукт: Детализированная информация об ингредиентах и товарах.
* Категория Продукта: Классификатор продуктов.
* Единица Измерения: Справочник единиц измерения продуктов.
* Поступление: Документы, фиксирующие приход продуктов на склад.
* Списание: Документы, фиксирующие расход или утилизацию продуктов и блюд.
* Причина Списания: Справочник причин списания.
* Блюдо: Информация о позициях меню.
* Категория Блюда: Классификатор блюд.
* Состав Блюда (Рецептура): Связь между блюдами и продуктами, определяющая ингредиенты.
* Заказ: Информация о заказах клиентов.
* Позиция Заказа: Детализация заказа по конкретным блюдам.
* Остаток Продукта на Складе: Учет текущего количества каждого продукта на каждом складе.

Атрибуты для каждой сущности на концептуальном уровне отражают их основные характеристики. Например, для сущности "Продукт" это могут быть "Наименование", "Категория", "Единица измерения". Для сущности "Заказ" – "Дата заказа", "Сотрудник оформивший", "Общая сумма".

Связи между сущностями отражают бизнес-правила. Например, между "Сотрудник" и "Должность" существует связь "один-ко-многим" (одна должность может быть у многих сотрудников, но каждый сотрудник занимает одну должность). Связь между "Заказ" и "Блюдо" является "многие-ко-многим" и реализуется через промежуточную сущность "Позиция Заказа".

Результатом концептуального моделирования является ER-диаграмма (Entity-Relationship Diagram), которая наглядно представляет структуру данных. ER-диаграмма для разрабатываемой базы данных приведена в Приложении А (или на Рисунке 2.1, если ты решишь вставить ее прямо в текст).

## 2.2 Разработка логической модели

Логическая модель данных является следующим этапом проектирования и представляет собой преобразование концептуальной модели в реляционную структуру, не зависящую от физических аспектов хранения, но уже ориентированную на использование реляционной СУБД. На этом этапе определяются таблицы, их атрибуты, первичные и внешние ключи, а также детально прорабатываются связи между таблицами с указанием их кардинальности и опциональности. Для представления логической модели была использована нотация "вороньи лапы" (Crow's Foot).

В ходе разработки логической модели сущности из концептуальной модели были преобразованы в таблицы, а атрибуты сущностей – в поля таблиц. Связи "многие-ко-многим", такие как между "Заказом" и "Блюдом" или "Блюдом" и "Продуктом" (в контексте рецептуры), были реализованы через создание промежуточных (ассоциативных) таблиц: "ПозицияЗаказа" и "СоставБлюда" соответственно.

Ниже приведено описание основных таблиц логической модели (полная схема представлена на Рисунке 2.1 [ссылка на твое изображение физической модели, которое по сути является детализированной логической моделью в контексте таблиц и ключей]):

Таблица "Сотрудник": Хранит данные о сотрудниках. Первичный ключ – ID\_сотрудника. Внешний ключ ID\_должности ссылается на таблицу "Должность".

Таблица "Должность": Справочник должностей. Первичный ключ – ID\_должности.

Таблица "Продукт": Информация об ингредиентах. Первичный ключ – ID\_продукта. Внешние ключи ID\_категории\_продукта и ID\_единицы\_измерения ссылаются на соответствующие справочники.

Таблица "Блюдо": Позиции меню. Первичный ключ – ID\_блюда. Внешний ключ ID\_категории\_блюда ссылается на справочник категорий блюд.

Таблица "СоставБлюда": Промежуточная таблица для связи "многие-ко-многим" между "Блюдо" и "Продукт". Составной первичный ключ (ID\_блюда, ID\_продукта).

Таблица "Заказ": Информация о заказах. Первичный ключ – ID\_заказа. Внешние ключи ID\_кассы и ID\_сотрудника\_оформившего.

Таблица "ПозицияЗаказа": Промежуточная таблица для связи "многие-ко-многим" между "Заказ" и "Блюдо". Составной первичный ключ (ID\_заказа, ID\_блюда).

Таблица "Поступление": Учет прихода товаров. Первичный ключ – ID\_поступления. Внешние ключи ID\_склада и ID\_сотрудника\_принявшего.

Таблица "ПозицияПоступления": Детализация поступления. Составной первичный ключ (ID\_поступления, ID\_продукта).

Таблица "Списание": Базовая таблица для операций списания. Первичный ключ – ID\_списания.

Таблицы "СписаниеПродуктов" и "СписаниеБлюд": Реализуют механизм наследования от таблицы "Списание", детализируя списание конкретных объектов. Первичный ключ ID\_списания является также внешним ключом, ссылающимся на таблицу "Списание".

Таблица "ОстатокПродуктаНаСкладе": Учет текущих остатков. Составной первичный ключ (ID\_склада, ID\_продукта).

Связи между таблицами определены с указанием кардинальности. Например, связь между "Сотрудник" и "Смена" – один ко многим (один сотрудник может иметь много смен, каждая смена принадлежит одному сотруднику), обязательная со стороны "Смены". Связь между "Заказ" и "ПозицияЗаказа" – один ко многим, где один заказ должен содержать как минимум одну позицию, и каждая позиция принадлежит ровно одному заказу.

Логическая модель, представленная на Рисунке 2.1 [снова ссылка на твое изображение], обеспечивает структурированное представление данных, минимизирует избыточность и создает основу для физической реализации базы данных.

## 2.3 Выбор системы управления базами данных (СУБД)

Выбор подходящей СУБД является важным шагом, влияющим на производительность, надежность, масштабируемость и стоимость разработки и эксплуатации информационной системы. Для проекта "Sushi-shop" рассматривались реляционные СУБД, так как структура данных предметной области хорошо ложится на реляционную модель.

Основные кандидаты для выбора и их сравнительная характеристика по ключевым для проекта критериям представлены в Таблице 2.1.

Таблица 2.1 – Сравнительная характеристика реляционных СУБД

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | PostgreSQL | MySQL | MS SQL Server Express Edition |
| Тип лицензии | Открытое ПО (лицензия PostgreSQL, аналогична MIT) | Открытое ПО (GPL) / Коммерческая | Бесплатная версия коммерческого ПО |
| Поддержка стандартов SQL | Высокая, одно из лучших соответствий | Хорошая, с некоторыми особенностями | Хорошая, с T-SQL как диалектом |
| Объектно-реляционные возможности | Расширенные (наследование таблиц, пользовательские типы данных и операторы) | Ограниченные (в основном реляционная) | Присутствуют, но менее выражены, чем в PostgreSQL |
| Сложные запросы и аналитика | Очень хорошо подходит, мощный оптимизатор | Хорошо, но может уступать PostgreSQL на очень сложных запросах | Хорошо, особенно с инструментами аналитики Microsoft (в платных версиях) |
| Транзакции (ACID) | Полная поддержка | Полная поддержка (для InnoDB) | Полная поддержка |
| Масштабируемость | Высокая (вертикальная и горизонтальная) | Хорошая, особенно для операций чтения | Ограничена в Express Edition (по ресурсам) |
| Надежность и стабильность | Высокая, проверена временем | Высокая (особенно с InnoDB) | Высокая |
| Сообщество и поддержка | Активное и большое международное сообщество | Очень большое сообщество, обширная документация | Хорошая поддержка от Microsoft, активное сообщество |
| Простота использования и администрирования | Средняя, требует определенных знаний | Относительно простая, особенно для стандартных задач | Относительно простая, особенно с GUI-инструментами от Microsoft |
| Специфичные функции для проекта (наследование таблиц) | Полная поддержка | Отсутствует на уровне таблиц (можно эмулировать) | Отсутствует на уровне таблиц (можно эмулировать) |
| Стоимость | Бесплатно | Бесплатно (Community Edition) / Платно (Enterprise Edition) | Бесплатно (Express Edition) / Платно (Standard, Enterprise) |

Анализ сравнительной таблицы:

Из таблицы видно, что все три рассматриваемые СУБД являются зрелыми и надежными решениями. Однако, для проекта "Sushi-shop" PostgreSQL выделяется по нескольким ключевым параметрам:

Объектно-реляционные возможности и наследование таблиц: PostgreSQL нативно поддерживает наследование таблиц, что является важным аспектом для реализации механизма учета списаний (таблицы "СписаниеПродуктов" и "СписаниеБлюд" наследуют от "Списание"), как это было запланировано на этапе логического проектирования. В MySQL и MS SQL Server подобную функциональность пришлось бы эмулировать, что усложнило бы структуру и запросы.

Поддержка сложных запросов и аналитики: Для задач расчета себестоимости, маржинальности и формирования различных аналитических отчетов PostgreSQL предоставляет мощный инструментарий и эффективный оптимизатор запросов.

Открытый исходный код и отсутствие лицензионных ограничений: Это делает PostgreSQL привлекательным выбором с точки зрения затрат и гибкости использования.

Высокая степень соответствия стандартам SQL: Обеспечивает лучшую переносимость знаний и кода.

MySQL, будучи очень популярной и быстрой для многих веб-приложений, уступает PostgreSQL в поддержке расширенных реляционных и объектных функций, которые могут быть полезны для данного проекта. MS SQL Server Express Edition, хотя и является качественным продуктом, имеет ограничения по ресурсам (размер базы данных, использование процессора и памяти), что может стать препятствием при росте системы "Sushi-shop".

Таким образом, для реализации базы данных информационной системы "Sushi-shop" была выбрана СУБД PostgreSQL. Данный выбор обусловлен ее богатой функциональностью, включая поддержку наследования таблиц, высокой надежностью, соответствием стандартам SQL, отсутствием лицензионных затрат и хорошей масштабируемостью, что наилучшим образом соответствует требованиям проекта.

## 2.4 Разработка физической модели

Физическая модель данных описывает конкретную реализацию логической модели в выбранной СУБД, в данном случае – PostgreSQL. На этом этапе определяются точные типы данных для каждого поля, создаются индексы для оптимизации запросов, определяются правила ссылочной целостности и другие специфичные для СУБД параметры.

Структура таблиц, их поля и первичные/внешние ключи в физической модели соответствуют ранее разработанной логической модели (Рисунок 2.1). Ниже приведено описание ключевых аспектов физической реализации для некоторых таблиц с указанием типов данных PostgreSQL:

Таблица "Сотрудник":

* ID\_сотрудника: SERIAL PRIMARY KEY (автоинкрементный целочисленный идентификатор).
* ID\_должности: INTEGER NOT NULL REFERENCES "Должность"(ID\_должности).
* Фамилия, Имя, Отчество: VARCHAR(100).
* Телефон: VARCHAR(20) UNIQUE.
* Дата\_приема: DATE NOT NULL DEFAULT CURRENT\_DATE.
* Активен: BOOLEAN NOT NULL DEFAULT TRUE.

Таблица "Продукт":

* ID\_продукта: SERIAL PRIMARY KEY.
* ID\_категории\_продукта: INTEGER NOT NULL REFERENCES "КатегорияПродукта"(ID\_категории\_продукта).
* ID\_единицы\_измерения: INTEGER NOT NULL REFERENCES "ЕдиницаИзмерения"(ID\_единицы\_измерения).
* Название: VARCHAR(255) NOT NULL UNIQUE.
* Срок\_годности\_дни: INTEGER CHECK("Срок\_годности\_дни" > 0).

Таблица "ПозицияЗаказа":

* ID\_заказа: INTEGER NOT NULL REFERENCES "Заказ"(ID\_заказа) ON DELETE CASCADE.
* ID\_блюда: INTEGER NOT NULL REFERENCES "Блюдо"(ID\_блюда).
* Количество: INTEGER NOT NULL CHECK("Количество" > 0).
* Цена\_на\_момент\_заказа: NUMERIC(10, 2) NOT NULL.
* PRIMARY KEY (ID\_заказа, ID\_блюда).

Для обеспечения ссылочной целостности активно используются внешние ключи с указанием соответствующих действий (ON DELETE CASCADE в таблицах "ПозицияЗаказа", "СоставБлюда", "ОстатокПродуктаНаСкладе" для автоматического удаления связанных записей при удалении основной). Ограничения NOT NULL, UNIQUE и CHECK применяются для гарантии корректности и непротиворечивости данных.

Для повышения производительности запросов создаются **индексы**. Помимо индексов, автоматически создаваемых для первичных и уникальных ключей, дополнительные индексы создаются для всех внешних ключей, а также для полей, часто используемых в условиях выборки (WHERE) или сортировки (ORDER BY), например, для полей с датами (Дата\_создания в таблице "Заказ", Дата\_поступления в таблице "Поступление").

Физическая модель также включает определение таких объектов базы данных, как пользователи (роли), представления (views), триггеры и хранимые процедуры. Их назначение и логика работы будут рассмотрены в Разделе 3, а полный SQL-код для их создания представлен в Приложениях.

## 2.5 Проведение нормализации

Нормализация – это процесс организации данных в базе данных с целью уменьшения избыточности и улучшения целостности данных. Он включает применение набора правил, называемых нормальными формами. Для разрабатываемой базы данных "Sushi-shop" был проведен анализ на соответствие основным нормальным формам.

Первая нормальная форма (1НФ): Требует, чтобы все атрибуты таблиц были атомарными, то есть не содержали повторяющихся групп или составных значений в одной ячейке. Все таблицы в спроектированной базе данных соответствуют этому требованию, так как каждое поле хранит только одно значение.

Вторая нормальная форма (2НФ): Требует, чтобы таблица находилась в 1НФ и все неключевые атрибуты полностью функционально зависели от всего первичного ключа. Это требование особенно актуально для таблиц с составными первичными ключами.

Например, в таблице "ПозицияЗаказа" первичный ключ составной (ID\_заказа, ID\_блюда). Атрибуты Количество и Цена\_на\_момент\_заказа полностью зависят от комбинации конкретного заказа и конкретного блюда в этом заказе, а не от какой-либо части ключа по отдельности. Аналогично для таблиц "СоставБлюда","ПозицияПоступления", "ОстатокПродуктаНаСкладе".

Третья нормальная форма (3НФ): Требует, чтобы таблица находилась в 2НФ и не содержала транзитивных зависимостей неключевых атрибутов от первичного ключа (то есть, неключевой атрибут не должен зависеть от другого неключевого атрибута).

Например, в таблице "Сотрудник" атрибут Ставка\_в\_час не хранится напрямую, так как он зависит от должности сотрудника. Вместо этого, ID\_должности является внешним ключом, а сама ставка хранится в таблице "Должность". Это предотвращает транзитивную зависимость (Сотрудник -> Должность -> Ставка).

Аналогично, наименования категорий, единиц измерения вынесены в отдельные справочные таблицы ("КатегорияПродукта", "ЕдиницаИзмерения", "КатегорияБлюда", "ПричинаСписания"), и в основных таблицах используются только их идентификаторы, что устраняет транзитивные зависимости и избыточность.

Нормальная форма Бойса-Кодда (БКНФ): Является усиленной версией 3НФ. В большинстве случаев, если база данных находится в 3НФ и каждый детерминант (атрибут или набор атрибутов, однозначно определяющий другие атрибуты) является потенциальным ключом, то она находится и в БКНФ. Спроектированная структура стремится к соответствию БКНФ[4].

Вывод:

В рамках данного раздела была проделана работа по проектированию базы данных для информационной системы "Sushi-shop". Разработана концептуальная модель, отражающая основные сущности и связи предметной области. На ее основе создана логическая модель в нотации "вороньи лапы", детализирующая структуру таблиц, атрибутов и ключей. Был обоснован выбор СУБД PostgreSQL для реализации проекта. Далее была разработана физическая модель, описывающая реализацию таблиц с указанием типов данных PostgreSQL, ограничений и индексов. Проведенный анализ нормализации подтвердил, что структура базы данных соответствует как минимум третьей нормальной форме, что обеспечивает минимизацию избыточности и поддержку целостности данных. Созданная таким образом структура базы данных является надежной основой для последующей реализации и эксплуатации информационной системы.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Чен П. П.–Ш. Модель "сущность–связь" — шаг к единому представлению данных // СУБД. — 1995. — № 3. — С. 137–158.

2. Мартин Дж. Проектирование баз данных. — М.: Вильямс, 2018. — 45 с.

3. Дейт К. Дж. Введение в системы баз данных. — 8-е изд. — М.: Вильямс, 2010. — 1328 с.

4. Старкс Д. MySQL Security Handbook. — 2018. — 320 с.

5. Официальный сайт OpenServer [Электронный ресурс]. — URL: https://ospanel.io/ (дата обращения: 04.05.2025).

6. Документация по Windows Forms для .NET 7 | Microsoft Learn [Электронный ресурс]. — URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-8.0 (дата обращения: 04.05.2024).

Дейт К. Дж. Введение в системы баз данных. – 8-е изд. – М.: Вильямс, 2019. – 1328 с.

Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. – 3-е изд. – М.: Вильямс, 2018. – 1436 с.

Иванов А.А. Автоматизация ресторанного бизнеса: современные подходы и технологии // Управление предприятием. – 2022. – №3. – С. 45-52.

Петров В.С. Проектирование информационных систем в сфере общественного питания: учебное пособие. – СПб.: Питер, 2021. – 240 с.

PostgreSQL Global Development Group. PostgreSQL: Документация. [Электронный ресурс]. URL: https://www.postgresql.org/docs/ (дата обращения: 10.05.2025).

1. POS vs ERP Program System: Which One is Better? [Электронный ресурс]. — URL: [https://www.hashmicro.com/blog/pos-erp-program/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fwww.hashmicro.com%2Fblog%2Fpos-erp-program%2F) (дата обращения: 10.05.2025).
2. Системы управления запасами (Inventory Management Systems, IMS) [Электронный ресурс]. — URL: [https://soware.ru/categories/inventory-management-systems#definition](https://www.google.com/url?sa=E&q=https%3A%2F%2Fsoware.ru%2Fcategories%2Finventory-management-systems%23definition) (дата обращения: 10.05.2025).
3. Кара-Ушанов, В. Ю. Базы данных : учебное пособие / В. Ю. Кара-Ушанов ; М-во науки и высш. образования Рос. Федерации, Урал. федер. ун-т. – Екатеринбург : Изд-во Урал. ун-та, 2018. – 108 с. – ISBN 978-5-7996-2553-3. – URL: https://study.urfu.ru/Aid/Publication/13604/1/Kara-Ushanov.pdf (дата обращения: 13.05.2025).
4. Кузнецов, С. Д. Основы баз данных: учебное пособие / С. Д. Кузнецов. – 2-е изд., испр. – Москва: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2007. – 484 с. – (Основы информационных технологий). – ISBN 978-5-94774-736-2.